

An invariant descriptor map for 3D objects matching

Abdallah El Chakik ^{1*}, Abdul Rahman El Sayed ², Hassan Alabboud ³, Amer Bakkach ⁴

¹ Beirut Arab university, Tripoli, Lebanon

² Doctoral School in Sciences and Technology, Laboratory of Informatics and Applications (LIA), Lebanese University, Lebanon

³ Business and Administration Faculty, Lebanese university, Tripoli, Lebanon

⁴ Beirut international University, Beirut, Lebanon

*Corresponding author E-mail: a.alshakik@bau.edu.lb

Abstract

Mesheres and point clouds are traditionally used to represent and match 3D shapes. The matching problem can be formulated as finding the best one-to-one correspondence between featured regions of two shapes. This paper presents an efficient and robust 3D matching method using vertices descriptors detection to define feature regions and an optimization approach for regions matching. To do so, we compute an invariant shape descriptor map based on 3D surface patches calculated using Zernike coefficients. Then, we propose a multi-scale descriptor map to improve the measured descriptor map quality and to deal with noise. In addition, we introduce a linear algorithm for feature regions segmentation according to the descriptor map. Finally, the matching problem is modelled as sub-graph isomorphism problem, which is a combinatorial optimization problem to match feature regions while preserving the geometric. Finally, we show the robustness and stability of our method through many experimental results with respect to scaling, noise, rotation, and translation.

Keywords: 3D Mesh; Descriptor Map Detection; Multi-Scale Computation; 3D Surface Segmentation; 3D Models Matching.

1. Introduction

The convergence between computer science, audiovisual, and internet leads to widen visual information. Many applications produce and use visual data, including digitized 3D objects due to the progress in acquisition hardware technics such as medical imaging devices, laser scanners, etc. The widespread distribution and storage of high quality 3D data brings the need of developing efficient algorithms to recognize, organize, and classify these data. In this context, matching these 3D objects is an important research field in computer vision with many applications in different fields such as medical imaging, computer graphics, molecular biology. 3D Objects matching problem can be formulated as finding a significant correspondence between object's surfaces.

In this paper, we present a robust method for finding a geometric correspondence between 3D shapes. We formulate the matching process as the maximum relational sub-graph matching. To do so, we compute an invariant multiscale vertices descriptors map to describe the signature of the shape. Then we define interest regions from the descriptors map and we apply an optimization algorithm to find the maximum relation between shapes featured regions. We construct firstly a Delaunay graph from the initial data to guarantee a strong neighboring relation modelling. Then, we compute an invariant 3D surface descriptor using 3D patches calculated based on Zernike coefficients. To define a patch for a vertex v_i we construct a fixed length rectangle centered at v_i with n cells. We compute then a height map by projecting the rectangle vertices onto a 2D plan and convert the height map into an image in order to calculate Zernike coefficients. Consequently, the vertex v_i can be described by a single scale descriptor computed based on Zernike coefficients. To enhance the quality of the vertices descriptors, we define multi-scale descriptor map by merging single scale descriptor maps calculated with different Delaunay neighboring level. A set of feature regions is defined using this multi-scale descriptor map. Finally, the 3D matching is formulated as a maximum relational matching of graphs representing the 3D objects and modeled as a combinatorial optimization problem.

1.1. Related work

In this subsection, we present the existing work on the 3D descriptors detection and shape matching.

Surface descriptors detection. Most of 3D descriptor methods focused on the 3D object shape, as given by the object's interior or its surface, rather than other characteristics such as color and texture, which are usually used in 2D image segmentation and recognition. This orientation in research is caused probably by the fact that the most information about the 3D objects similarity is computed and gathered by their shapes. Moreover, during recognition or segmentation processes, texture and color information is not always guaranteed to be provided. Many 3D descriptors models have been proposed. In [1], authors proposed to construct the patches to describe the surface by projecting the set neighbors of a given vertex onto an 2D plane oriented by principal directions and compute the patch degree based on the deviation factors between vertices. Authors in [2] proposed to build local adaptive patches in order to describe local vertices surfaces. The orientation and construction of proposed patch are computed according to a spherical neighborhood and 2D projection. Leifman et al. proposed in [3]

to construct a 2D histogram of spin images for each vertex. They measured then the disparity between these 2D histograms by diffusion discretization using the Gaussian pyramid. They computed the dissimilarity between vertices using a single scale computation and they proposed to decrease the noise and 3D texture effect. Finally, they have selected a set of focus vertices from the set of all vertices where the dissimilarity values is greater than 20%. The drawback of this selection is that the set of featured regions where the vertices dissimilarity value less than 20% is not considered and the multi scale phase used may not decrease the effects of noise. In contrast, we propose in our method an invariant descriptor dealing with noise based on Zernike Coefficients. Wu et al. in [4] have proposed to describe vertices by using the global rarity and local contrast. They calculated a square map of height projection to describe the shape that surrounds a given vertex. Zhao et al. [8] proposed to detect interest points using the Retinex theory [9] to estimate the invariant properties of the given surface and improve its local details. Tao et al. proposed in [11] a method to describe surface using the manifold-ranking algorithm in a descriptor space. They defined patches on the mesh and descriptors for these patches based on Zernike coefficients. They estimated then for each patch the local distinctness using the center-surround operator. They classified patches as background if they have a low local local distinctness and foreground otherwise. The method deals with noise but authors stated that it doesn't integrate priors with high levels.

Matching and object recognition. The dynamic programming technic can be used to find a matching between 3D shapes where the time complexity of the search all over the correspondence us cubic in shapes points number [5]. Thus, most of existing 3D matching methods rely on the technics of local optimization. Our proposed approach finds a maximum relational sub graph matching among the set of feature regions that represent an object. In [6], they proposed to map patches with tiny surfaces between two 3D objects instead of the point to point matching. They defined the matching problem as the diffeomorphisms between different 3D objects, which represented as the closed and continuous surfaces in the product space of shapes. This representation leads to a problem with the discrete formulation describing the space of search with linear constants. In [7], they proposed to find the correspondence that minimizes geodesic distortion using the framework of Gromov–Hausdorff. Authors in [8] calculated the correspondence between shapes using the coarse-to-fine strategy as optical flow approaches. Other 3D shapes matching methods as Riemannian [9] or conformal [10] geometries have proposed to use a local optimization technique to minimize the nonconvex energy. Thus, the quality of the solution depends on a well-designed coarse-to-fine strategy and an appropriate initialization. The solution in these methods doesn't guarantee the optimality. Authors in [12] recently proposed to formulate the matching problem as a graph-matching process and applying the QPBO algorithm [13]. The main drawbacks of this method are the continuous counterpart lack and the high computational complexity where it matches just the set of discrete points instead of patches that describe the surface. To overcome these problems, we propose to define 3D surface patches then computing an invariant multiscale saliency map to finally find the correspondence between featured regions from these maps.

1.2. Paper organization

The rest of this paper is organized as follows. Section 2 presents our 3D matching framework: model for 3D mesh surface, multiscale descriptor map computation, etc. In section 3, we show the efficiency of our proposed approach via many experimental results and we present the robustness and stability of our method according to rotation, noise, scaling, etc. Section 4 evaluate our method. Section 5 presents some applications. Finally, Section 6 concludes this work.

2. 3D matching framework

To find a matching between different 3D shapes, we define firstly the set of feature regions using a multi-scale descriptor map that describes the 3D object homogeneity. Then, we search for the maximum subgraph relational matching between these feature regions. As a 3D shape can be modeled by a weighted graph, the subgraph relational matching(isomorphism) ca be modeled as combinational optimization.

2.1. Descriptor map computation

We propose to define a set of descriptors to describe a given 3D surface. To do so, we start by modeling the 3D surface a weighted graph and estimating vertices normal vectors. Then, we define an invariant patch for each vertex that will be projected onto a 2D plane. A height map is then calculated and we compute Zernike coefficients for all vertices using the height map and associated patches. We compute finally the single-scale descriptors using the Zernike coefficients and we define the multi-scale descriptors by increasing the number of vertices neighbors.

2.1.1. Surface modeling and normal estimation

A 3D mesh surface can be represented as a weighted graph $G = (V, E, w)$ where V represents a finite set of vertices, E is a finite set of edges $E \subset V \times V$, and w the weight function that represents the similarity measure between vertices.

The set of neighbors $D_n(v_i)$ of a given vertex v_i is constructed using the concept of Delaunay triangulation [15]. To do so, we define the set of circumscribed spheres from all 3D surface points where a sphere doesn't contain any 3D point. A point P_j located on the borders of a circumscribed sphere from P_i is defined as P_i neighbor. In other word, The set of P_i neighbors is defined as the vertices existing on the triangles corners (faces) that share the set of P_i 's Neighbors as presented in Figure 1.

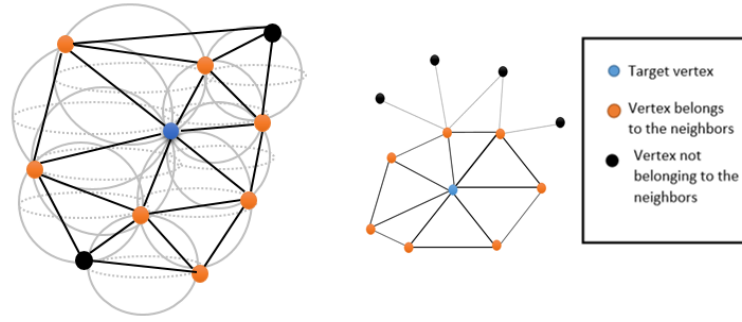


Fig. 1: Neighboring Function Construction.

The edge $(\mathbf{v}_i, \mathbf{v}_j)$ connects $\mathbf{v}_i$ and $\mathbf{v}_j$ and each vertex $\mathbf{v}_i$ is represented by the 3D coordinates $\mathbf{v}_i = (x_i, y_i, z_i)^T$, a normal vector $\mathbf{N}(\mathbf{v}_i)$, and the directional vectors $x(\mathbf{v}_i)$ and $y(\mathbf{v}_i)$ corresponding to the 2D tangent plan estimated at $\mathbf{v}_i$.

In order to compute the normal $\mathbf{N}(\mathbf{v}_i)$ of $\mathbf{v}_i$ and the 2-directional vectors that follow x- and y-axes we use the covariance matrix $\text{Cov}(\mathbf{v}_i)$. For this, we calculate the gravity center and the associated matrix of covariance at $\mathbf{v}_i$.

- The gravity center is computed as:

$$\check{\mathbf{v}}_i = \frac{1}{\text{Card}(D_n(\mathbf{v}_i))} \sum_{\mathbf{v}_j \in D_n(\mathbf{v}_i)} \mathbf{v}_j \quad (1)$$

- Then, we compute the covariance matrix at $\mathbf{v}_i$ as:

$$\text{Cov}(\mathbf{v}_i) = \sum_{\mathbf{v}_j \in D_n(\mathbf{v}_i)} (\mathbf{v}_j - \check{\mathbf{v}}_i)(\mathbf{v}_j - \check{\mathbf{v}}_i)^T \in \mathbb{R}^{3 \times 3} \quad (2)$$

Additionally, we use the covariance matrix to compute $\mathbf{C}_0(\mathbf{v}_i)$ and $\mathbf{C}_1(\mathbf{v}_i)$ estimating the minor and major principal directions.

2.1.2. Invariant patch construction

A vertex descriptor is computed depending on the vertex patch. These patches should be well defined to describe the vertex and constructed as:

- Firstly a rectangle with n cells is defined and centered at a given vertex $\mathbf{v}_i$ with a side length $L(\mathbf{v}_i)/n$ for each rectangle cell, where n is a constant depending on the application and $L(\mathbf{v}_i)$ is calculated according to the bounding box of $\mathbf{v}_i$ neighbors as:

$$L(\mathbf{v}_i) = \max_{\mathbf{v}_j \in D_n(\mathbf{v}_i)} (\|\mathbf{v}_j - \mathbf{v}_i\|_2^2). \quad (3)$$

Where $\|\cdot\|_2^2$ the Euclidean norm.

- We construct then the height map for every vertex $\mathbf{v}_i$ by projecting the set of $\mathbf{v}_i$ neighbors onto a 2D plane defined by the vectors $(x$ and $y)$ and passing by the vertex $\mathbf{v}_i$. We obtain as a result for any projected vertex $\mathbf{v}_j'$, the list of vertices $\mathbf{v}_j'$ defined as:

$$\mathbf{v}_j' = [(\mathbf{v}_j - \mathbf{v}_i) \cdot x(\mathbf{v}_i), (\mathbf{v}_j - \mathbf{v}_i) \cdot y(\mathbf{v}_i)]^T. \quad (4)$$

The position of cells is defined according to $\mathbf{v}_j'$ where $\mathbf{v}_j \in D_n(\mathbf{v}_i)$ as:

$$\text{position}^d(\mathbf{v}_j') = \left\lfloor \frac{\mathbf{v}_j'^c}{\frac{L(\mathbf{v}_i)}{n}} \right\rfloor, \quad (5)$$

Where c represents x or y coordinates and $\lfloor \cdot \rfloor$ is the rounded integer.

At this stage, the absolute value of projections sum is assigned to each cell of the patch. The projection height of $\mathbf{v}_i$ is defined as:

$$\text{He}(\mathbf{v}_i) = \|\mathbf{v}_i - \mathbf{v}_i'\|_2^2 \quad (3)$$

- Finally, we convert the height map into an image where its intensity is the cells value. Then, we compute the images Zernike coefficients [18] that yields to an invariant representation. Thus, a given vertex $\mathbf{v}_i$ is described by the list of Zernike coefficients.

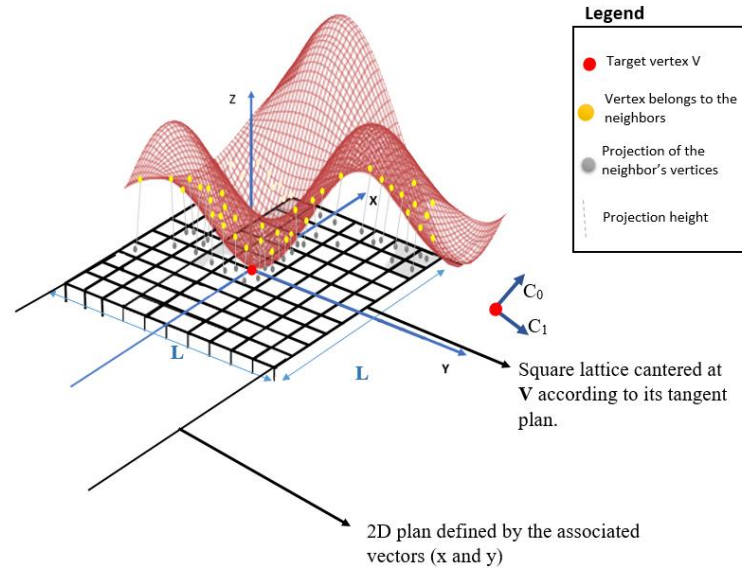


Fig. 2: Constuction of Patches with Vertices Projection. $C_0(V)$ and $C_1(V)$ Represent the Principal Directions at V and L The Length of the Given Patch.

2.1.3. Single-scale descriptor

The descriptor map of vertices is constructed based on Zernike coefficients. We firstly define the weight function using the sets of Zernike coefficients of vertices. Then the descriptor map is computed using the weight and the angle between normal of the given set of vertices. The distance function $dsit(Z_1, Z_2)$ between two Zernike coefficients Z_1 and Z_2 is computed similarly to the Euclidian distance as:

$$dsit(C_1, C_2) = \sqrt{(real(Z_2) - real(Z_1))^2 + (imaginary(Z_2) - imaginary(Z_1))^2} \quad (7)$$

Then, the distance between the Zernike coefficient sets describing two adjacent vertices v_i is defined as following:

$$D_z(v_i, v_j) = \frac{\sum_{k=1}^n dist(Z^k(v_i), Z^k(v_j))}{n}, \quad (8)$$

Where $Z^k(v_j)$ is the K^{ism} coefficient.

Thus, the weight function $\omega: V \times V \rightarrow [0, 1]$ can be calculated as:

$$\omega(v_i, v_j) = e^{-\frac{D_z(v_i, v_j)}{\sigma(v_i) * \sigma(v_j)}}, \quad (9)$$

Where $\sigma(v_i)$ represents the scale parameter and can be computed as:

$$\sigma(v_i) = \max_{v_k \in D_n(v_i)} (\|v_k - v_i\|_2) \quad (10)$$

Finally, the single-scale descriptor at v_i is defined as:

$$\text{Single-Scale-descriptor}(v_i) = \frac{\sum_{j \in |V|} \alpha(v_i, v_j) \times w(v_i, v_j)}{|V|} \quad (11)$$

Where $\alpha(v_i, v_j)$ the angle between v_i 's and v_j 's normal.

Since the descriptor degree of a vertex v_i depends on the Zernike coefficients that describes v_i and its neighbor, this will ensure that all the information of v_i 's neighbors are considered which lead to a robust and consistent descriptor degree.

2.1.4. Multi scale descriptor

In order to deal with noise, scaling and translation, we propose to improve the quality of the descriptor map by constructing descriptor maps at different scales and then finding a multi-scale descriptor map. To do so, we define each scale relatively to a vertices neighboring function where we consider four Delaunay neighboring levels which are 2, 4, 6 and 8. We calculate then the Pearson Correlation Coefficient to measure the strenghtens of the relation between different scales descriptors maps. Finally, we merge the descriptor map by weighting the degrees of each vertex Correlation Coefficient in order to obtain the multi-scale descriptor map.

Given vertex v_i , we compute its degree of single scale descriptor and the average of its neighbors descriptors degrees. Then, we calculate the Pearson Correlation coefficient as:

$$r = \frac{\sum_{k=1}^4 (S_k(v_i) - \bar{S}_k(v_i))(AS_k(v_i) - \overline{AS}_k(v_i))}{\sqrt{\sum_{k=1}^4 (S_k(v_i) - \bar{S}_k(v_i))^2} \sqrt{\sum_{k=1}^4 (AS_k(v_i) - \overline{AS}_k(v_i))^2}} \quad (12)$$

Where $S_k(v_i)$ represents the Single Scale descriptor of v_i at a given scale k and $AS_k(v_i)$ is the average Single Scale descriptor of v_i neighbors.

A robust multi scale descriptor map is obtained by weighting the descriptor degree of each vertex by the coefficient of correlation. This multi scale map takes into consideration the descriptor degree disparity at each scale and computed as:

$$\text{Multi - Scale - descriptor}(v_i) = r * \text{Single - Scale - descriptor}(v_i) \quad (13)$$

Figure 4 shows the results of our method and associated ground truth maps [16]. We can state that the results of our approach correspond well to the ground truth as we can detect the important regions in 3D objects colored in red. The highest vertices descriptor degrees are colored in red and decrease gradually to the blue color as seen in Figure 3 a.

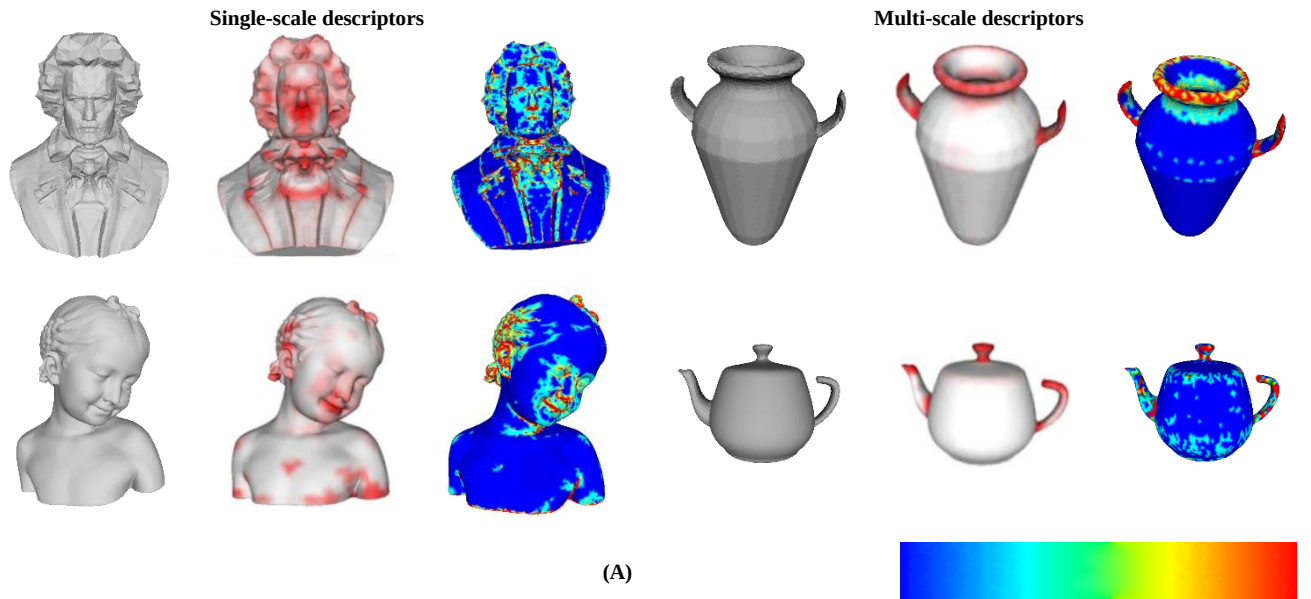


Fig. 3: Descriptor Maps. From Left to Right: Initial 3D Meshes, Ground Truth Maps, Descriptor Maps (A) Shows the Color Map).

2.1.5. Detection of feature regions

A 3D mesh can be segmented into feature regions using the multi-scale descriptor map. We segment firstly the 3D mesh according to vertices descriptor degree where a feature region will contain similar vertices with highest degrees.

To measure the similarity of two vertices v_i and v_j according to their descriptor degree, we define a descriptor similarity function as following:

$$F(v_i, v_j) = \begin{cases} 1 & \text{if } |SL(v_i) - SL(v_j)| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

Where $SL(v_i)$ represents the descriptor degree of v_i and σ is the vertices density. Two vertices v_i and v_j belong to the same segment if $F(v_i, v_j) = 1$.

In order to cluster the graph vertices, we propose to implement a linear programming model to aggregate the set of similar vertices by minimizing a cost function. We formulate our problem as a combinatorial problem of graph labelling where we search for the minimum number of labels to be assigned for similar vertices. We propose to use the integer linear problem ILP for vertices-clustering process where its optimal solution is a graph with minimum number of labels assigned to similar vertices.

We start by considering a set of n labels and vertices $V = \{v_1, \dots, v_n\}$ and we assign a unique label for each vertices. Then, our goal is to minimize the number of labels by merging two similar vertices using a unique label. The labeling ILP model can be established as

$$\text{Min } \sum_{l=1}^n y_l$$

Subject to:

- 1) $\sum_{l=1}^n x_{il} = 1 \quad \forall i=1, \dots, n$
- 2) $x_{il} - y_l \leq 0 \quad \forall i, l=1, \dots, n$
- 3) $x_{il} - x_{jl} \leq 0 \quad \forall l=1, \dots, n; \forall (v_i, v_j) \in E; F(v_i, v_j) = 1,$
- 4) $x_{il} \in \{0, 1\}; y_l \in \{0, 1\}$

Where y_l is a binary variable to indicate if a label l has been used ($y_l = 1$) and x_{il} is a binary variable that indicates whether a vertex v_i has been assigned with the label l .

Constraint (4) ensures that variables x_{il} and y_l are binary, Constraints (1,2,3) guarantee respectively that each vertex has been labelled; v_i can receive a label k if the label is already used, and non-similar adjacent vertices will have different labels.

To solve the ILP model, we present a heuristic algorithm similar to [17] with a score function $\text{Score}(v_i, v_j)$ as:

$$\text{Score}(v_i, v_j) = \begin{cases} 0 & \text{if } v_j \in C(v_i) \\ 1 & \text{otherwise,} \end{cases} \quad (15)$$

Where $C(v_i)$ is the set of vertices belonging to $D_n(v_i)$ and similar to v_i in term of descriptor defined as following:

$$C(v_i) = \{v_j \in D_n(v_i) \text{ such as } F(v_i, v_j) = 1\} \quad (16)$$

As a results, The set of vertices having same geometrical properties will have the same label as shown in Figure 4.

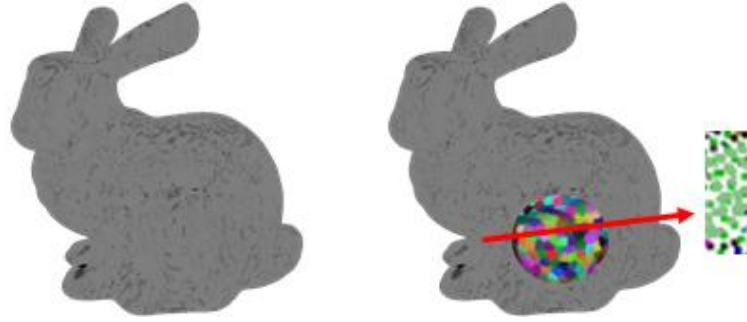


Fig. 4: Feature Regions Representing Vertices with Similar Geometrical Properties.

Figure 5 summarizes the regions segmentation process:

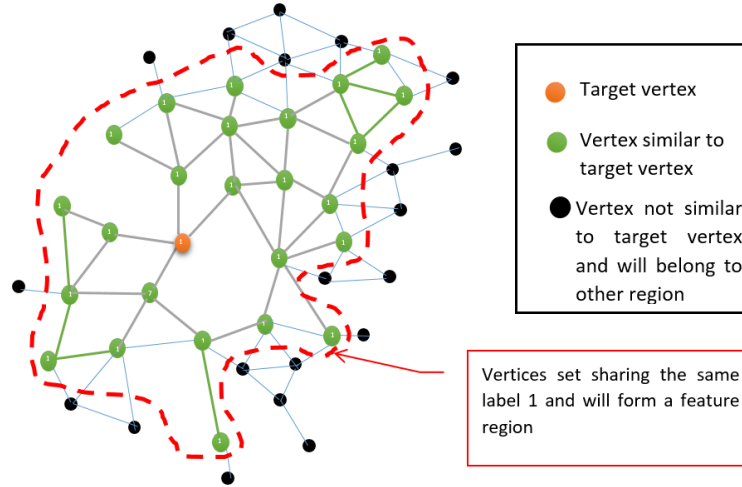


Fig. 5: Region Segmentation Illustration.

Finally, after defining the set of feature regions containing highest descriptor degree, the matching process consists to find a correspondence region for each feature region in the second 3D object.

2.2. 3D object matching

2.2.1. Introduction

3D objects matching problem is formulated as finding the relation between maximum number of featured common parts between two objects. We propose to model the matching problem as a combinational optimization problem using maximum relational subgraph matching and simulated annealing meta-heuristic algorithm. The relational subgraph matching notion consist of finding the maximum number of subgraphs isomorphic pairs. Let $G_A = (V_A, E_A, w_A)$ and $G_B = (V_B, E_B, w_B)$ be two graphs having N vertices each and two associated adjacency matrices A and B . The goal is to find the bi-continued injective correspondence Π that minimizes a graphs distance function that can be defined as:

$$J(\Pi) = \sum_{k=1}^N \sum_{l=1}^N (E_A(v_k, v_l) - E_B(\Pi(v_k), \Pi(v_l)))^2 \quad (17)$$

The traditional matching problem between two graphs can be formulated using the Euclidean matrix from equation (17) as

$$J(P) = \|A - PBP^t\|^2, \quad (18)$$

Where P represents the permutation matrix with the coefficient:

$$P_{ki} = \begin{cases} 1 & \text{if } \Pi(V_k) = V_i' \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

The graph isomorphism is hence reduced to the problem of finding the permutation matrix that minimizes equation (18).

The relational sub-graph isomorphism criteria is defined as:

$$E = \sum_{s=1}^S w_s E^s, \quad (20)$$

Where w_s represents the weight for each relationship s and E^s is expressed by

$$E = \sum_{k=1}^N \sum_{l=1}^N (1 - 2a_{kl}) b_{\Pi(k)\Pi(l)} \quad (21)$$

Equation (21) is derived from equation (18) as the coefficient of p can be rewritten using kronecker symbol.

Finally, the pruning criteria for the maximum relational graph is computed as:

$$D_k = \sum_{s=1}^S (\sum_{l \neq k, l=1}^N (w_A^s (V_k, V_l) - w_B^s (\Pi(V_k), \Pi(V_l))))^2 \quad (22)$$

2.2.2. Matching by simulated annealing

Simulated annealing is an optimization algorithm that starts with an arbitrary state given a temperature T and generate random candidate state space transitions to optimize the system energy ΔE . A transition is associated with a probability (sigmoid function) and accepted if the probability is as follows:

$$\begin{cases} 1 & \text{if } \Delta E < 0 \\ e^{-(\Delta E/T)} & \text{otherwise} \end{cases} \quad (23)$$

To apply simulated annealing algorithm in the matching process, we should define the states and their transitions with associated energy changes.

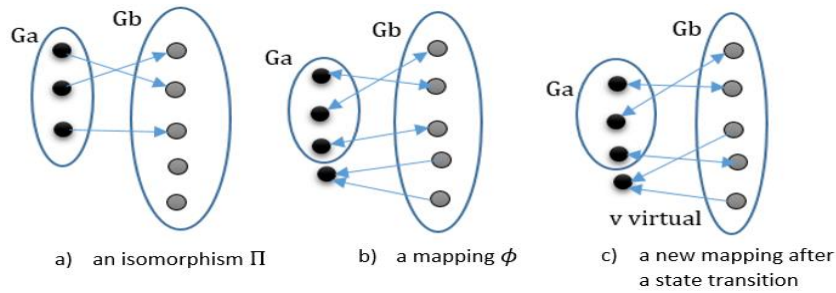


Fig. 6: Isomorphism and A Transition to A One.

The matching problem between two graphs G_A and G_B can be formulated as finding a subgraph of G_B isomorphic to G_A and minimizes the criteria in equation (21). Simulated annealing algorithm can be formulated using the property of graphs matching where we consider a state and the states spaces respectively as an isomorphism and the set of possible isomorphism. The mapping function ϕ can be defined as:

$$\begin{cases} \phi(v_B) = v_A & \text{if } \Pi(v_A) = v_B \\ \phi(v_B) = v_{virt} & \text{otherwise,} \end{cases} \quad (24)$$

Where Π is an isomorphism mapping G_A the a subgraph of G_B . This mapping is considered from G_B to $G_A \cup v_{virt}$ where a virtual node v_{virt} which is not connected to G_A nodes added to G_A as shown in figure 6 c).

The state transition can be done by changing two vertices correspondents within a mapping to produce a new one. To generate a random state,

A state transition is created by varying the of two vertices within a mapping to obtain a new one. The make random state changes, we should firstly randomly select a vertex v_A from G_A where its correspondent in G_B is $\Pi(v_A) = v_B$. Then, a random vertex v_B' is selected and we swap the mapping functions $\phi(v_B)$ and $\phi(v_B')$ in $G_A \cup v_{virt}$.

The system energy that controls the simulated annealing is defined as:

$$\Delta E = 2 \sum_{s=1}^S \lambda_s \sum_{m=1, m \neq k, l}^N (b_{i\Pi(m)}^s - b_{j\Pi(m)}^s) (a_{km}^s - a_{lm}^s) \quad (26)$$

3. Experimental results

To show the robustness of our approach, we made many experimentations using the dataset “MeshsegBenchmark” [14] that contains 400 objects with a variety of shapes and conditions. Figure 7 presents the matching of the same 3D object with changes in arms and legs positions. Once can see that our approach produces an exact matching of the shapes where we have mentioned that our proposed multi-scale descriptor map is invariant in term of movements.

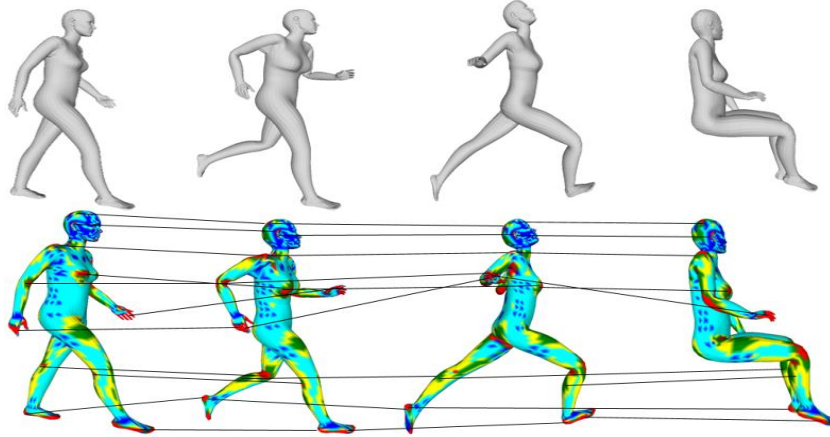


Fig. 7: Matching an Object with Different Movements. the Correspondences are Modeled as Lines That Connect Descriptors Map.

Figures 8 and 9 demonstrate that our method is invariant in term of scaling, rotation and translation. to rotation, translation, and scaling. Figure 8 shows each shape with its matched one.

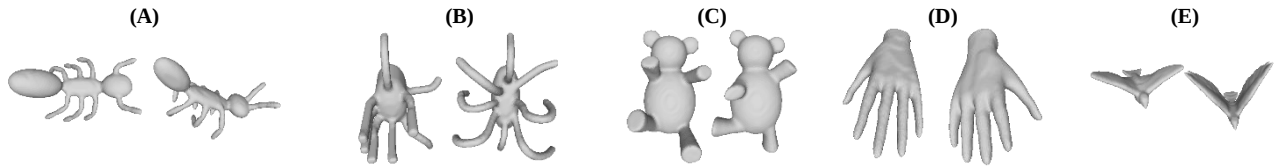


Fig. 8: Articulated Shapes Matching.

In Figure 9, we present the results of our descriptor map for the Gorilla 3D object with different transformations, mesh simplification, and holes. The results show that our method is invariant under all these factors.

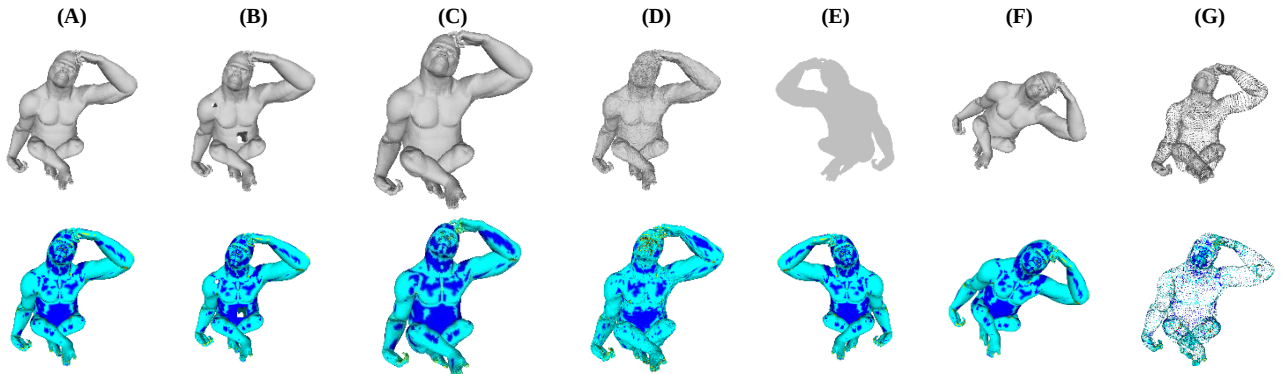


Fig. 9: Gorilla Mesh Matching with Different Transformations. (from Left to Right): (B) Object with Holes, (C) Scaled Object, (D) Noisy Objcey, (E,F) X Flip and Rotation, and (G) Mesh with Simplification.





Fig. 1: 3D Object Parts Detection.

Figure 10 shows that our method can also detect 3D object parts where the descriptor maps in the parts are similar to the original object. We have evaluated our approach in term of results accuracy. This accuracy is measured as:

$$\text{Accuracy} = 100 - (FPR + FNR), \quad (26)$$

Where the *False positive* and *false negative rates* *FPR* are respectively calculated as:

$$FPR = \frac{\text{number of false alarms}}{\text{number of correct matches}} \times 100, \quad (27)$$

$$FNR = \frac{\text{number of missed matching}}{\text{number of all models}} \times 100 \quad (28)$$

Table 1 shows the quantitative evaluation of our approach on 400 objects with different transformations.

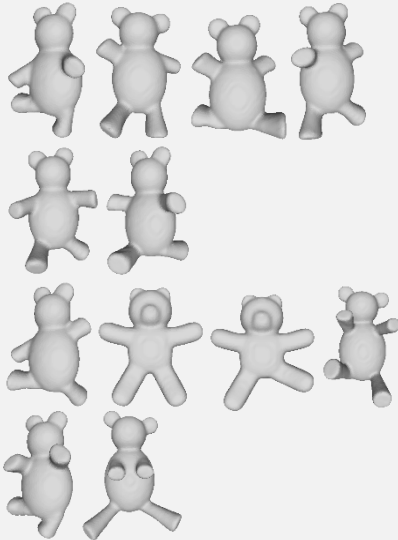
Table 1: Accuracy of Our Method

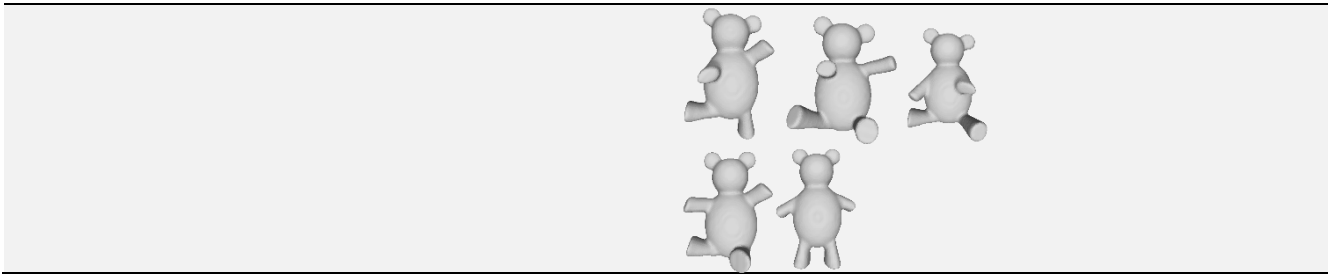
3D objects number	400
FPR	5.29
FNR	4.73
Accuracy	89.98

3.1. Classification

Our approach can be used to classify 3D objects. Table 2 shows the results of classification of two 3D objects with among 100 objects with different transformations. We calculated the multi-scale descriptor maps of these objects then we compute the similarity between the initial objects maps and all other ones and we select object with similarity more than 75%.

Table 2: Classification Results. from Top to Down: Initial Objects, Selected and Missed Objects

Initial Objects			
			
Selected Objects	Missed Objects	Selected Objects	Missed Objects
			



4. Conclusion

In this work, we presented a robust method for 3D objects matching. The initial 3D object has been segmented into patches by projecting the 3D object onto a 2D plane. Then the similarity function between vertices is computed based on Zernike coefficients calculated on images produced by the projection height maps. A single-scale descriptor map is then computed and extended to a multi-scale map to deal with noise and different objects transformations. We have proposed then to segment the 3D objects using an Integer Linear programming algorithm. A matching approach on the segmented object is then implemented using the simulated annealing algorithm based on relational subgraph matching approach. Finally, we showed the efficiency and robustness of our method through different experimentations made on many 3D objects with different transformations with respect to translation, scaling, noise, and rotation. Furthermore, we have presented the classification of 3D objects as an application of our approach.

References

- [1] El Sayed, Abdul Rahman, et al. 'Efficient 3D mesh salient region detection using local homogeneity measure', IET Image Processing, 2018, 12,(9), pp.1663 – 1672. <https://doi.org/10.1049/iet-ipr.2017.0598>.
- [2] A. Nouri, C. Charrier, O. L  zoray, "Multi-scale mesh saliency with local adaptive patches for viewpoint selection", Signal Processing: Image Communication, 38, (2015), pp.151–166. <https://doi.org/10.1016/j.image.2015.08.002>.
- [3] Tal, A., Shtrom, E., Leifman, G.: 'Fast-match: fast affine template matching surface regions of interest for view-point selection'. IEEE Conf. on Computer Vision and Pattern Recognition, 2012, pp. 414–421.
- [4] Wu, J., Shen, X., Zhu, W., et al.: 'Mesh saliency with global rarity', Graph.Models, 2013, 75, (5), pp. 255–264. <https://doi.org/10.1016/j.gmod.2013.05.002>.
- [5] F. R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In IEEE Int. Conf. on Computer Vision, Rio de Janeiro, October 2007. 1, 3, 7. <https://doi.org/10.1109/ICCV.2007.4409018>.
- [6] T. Windheuser, U. Schlickewei, F. R. Schmidt, & D. Cremers, Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In Computer Vision (ICCV), 2011 IEEE International Conference. (2011, November), (pp. 2134-2141). <https://doi.org/10.1109/ICCV.2011.6126489>.
- [7] F. M  moli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. Foundations of Computational Mathematics, 5(3), (2005), PP. 313–347. <https://doi.org/10.1007/s10208-004-0145-y>.
- [8] A. Bronstein, M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. SIAM J. Sci. Comput., 28(5), (2006), PP. 1812–1836. <https://doi.org/10.1137/050639296>.
- [9] S. Kurttek, E. Klassen, Z. Ding, and A. Srivastava. A novel riemannian framework for shape analysis of 3d objects. In CVPR, (2010), pp. 1625–1632. <https://doi.org/10.1109/CVPR.2010.5539778>.
- [10] Y. Wang, X. Gu, K. Hayashi, T. Chan, P. Thompson, and S. Yau. Brain surface parameterization using riemann surface structure. In MICCAI (2), (2005), pp. 657–665. https://doi.org/10.1007/11566489_81.
- [11] P Tao, J Cao, S Li, X Liu, L Liu. *Mesh saliency via ranking unsalient patches in a descriptor space*, Comput Graph, 2015, 46(0): 264–274. <https://doi.org/10.1016/j.cag.2014.09.023>.
- [12] Y. Zeng, C.Wang, Y.Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. In CVPR, (2010), pp. 382–389. <https://doi.org/10.1109/CVPR.2010.5540189>.
- [13] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In ECCV (2), (2008), pp. 596–609. https://doi.org/10.1007/978-3-540-88688-4_44.
- [14] <https://github.com/forsythrosin/cs348aHW4/tree/master/models/MeshsegBenchmark-1.0>.
- [15] J. D. Boissonnat and B. Geiger. Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation. In Biomedical Image Processing and Biomedical Visualization, 1905, (1993), pp. 964–975. <https://doi.org/10.1117/12.148710>.
- [16] Chen, X., Sapiro, A., Pang, B., et al. Schelling points on 3D surface meshes. ACM Trans. Graph, 2012, 31, (4), pp.29. <https://doi.org/10.1145/2185520.2185525>.
- [17] A. El Sayed, A. el Chakik, H. Alabboud, Adnan, A. Yassine. Efficient 3D point clouds classification for face detection using linear programming and data mining. The Imaging Science Journal, 2018, 66, pp. 23-37. <https://doi.org/10.1080/13682199.2017.1376772>.
- [18] A. Maximo, R. Patro, A. Varshney, R. Farias A robust and rotationally invariant local surface descriptor with applications to non-local mesh processing Graph Models, 73 (5) (2011), pp. 231-242. <https://doi.org/10.1016/j.gmod.2011.05.002>.