3D Models Retrieval Based on Turning Angle Function

**O. AIT ZEMZAMI**
Equipe ASIA, Department of Computer Science, Faculty of Sciences and Technology, Moulai Ismail University, P.O. Box 509, Boutalamine, Errachidia, Morocco

**H. AKSASSE**
Equipe ASIA, Department of Computer Science, Faculty of Sciences and Technology, Moulai Ismail University, P.O. Box 509, Boutalamine, Errachidia, Morocco

**M. OUANAN**
Equipe ASIA, Department of Computer Science, Faculty of Sciences and Technology, Moulai Ismail University, P.O. Box 509, Boutalamine, Errachidia, Morocco

**B. AKSASSE**
Equipe ASIA, Department of Computer Science, Faculty of Sciences and Technology, Moulai Ismail University, P.O. Box 509, Boutalamine, Errachidia, Morocco

**Abstract**—Recent scanning technology and 3D modeling allowed having a large 3D meshes database. These models are widely used in several areas such as CAD, computer graphics and audiovisual production. Content based retrieval is a necessary solution to structure, to manage the multimedia data, and to navigate in these databases. In this paper, we propose a method to automatically search and retrieve 3D models visually similar to a query 3D model. This is based on the representation of a 3D model by a series of slices along a direction; the nearest models to the query are those which have cuts similar to it.

Keywords—3D Model retrieval, Turning Angle Function, 3D triangular mesh model

I. INTRODUCTION

The availability of 3D models database has greatly expanded in the recent years, thanks to recent advances and developments in techniques for modeling, digitizing and visualizing 3D shapes.

The number of 3D object databases has been growing bringing, therefore, difficult to navigate through the database, making mandatory the development of indexing methods. Indexing models is to extract meaningful information from models to facilitate for subsequent comparison. The nature of these multimedia documents requires the phase separation of description and research to enable that it occurs in user time, relatively short time during which the user expects the system response. It varies depending mainly on the choice of descriptors used and their extraction techniques.

3D models retrieval system consists of extracting the most similar models for a given query from a large database. Many methods in this way are developed. To characterize the shape of the 3D object Assfalg et al. [2] constructed a map of curvature of its surface. After an initial preprocessing step trying to smooth and to reduce the complexity of the mesh, the authors express the curvature of each vertex of the mesh. The map of curvatures in two dimensions is obtained by projecting, the measuring at each vertex, to a cylinder. Finally, the descriptor is the partition of card into region with the area and the average curvature for each of them. This approach gives good results, but requires a normalization of the object to ensure invariance to geometric transformations. In order to take account the larger deformations on the surface, it would be interesting to implement an elastic comparison of curves card.

In order to enrich the description given by Vranic [8] Ohbuchi et al. [6] propose instead to extract 42 depths images. The views associated with these images are uniformly distributed on the unit sphere and correspond to the 12 vertices and 30 midpoints of edges of an icosahedron. The depths images obtained are finally transformed from the system of cartesian coordinates to polar coordinates before to keep the low frequency coefficients of their 2D FFT Zhang and Lu [10]. One of the main weaknesses of techniques using depths images is the difficult to take into account the deformation of the mesh.

In the approach based on the contours and silhouettes, Chaouch [3] proposed to characterize the shape of a 3D object by studying the position of the points that make up the outline. In order to normalize the contour length, the author makes an angular sampling by selecting for each sector, N farthest points.

The descriptor is then composed of the low-frequency coefficients of the one-dimensional Fourier transform for N points in polar coordinates of the three views corresponding to the directions of the principal axes.

Funkhouser et al. [4] proposed to segment the 3D object in order to make partial matches on it. To partition the model, the authors have developed a tool to draw the path where the cut will be. To assist the user, the path trying to follow the natural "seam" of the object, thus making more effective segmentation. Each part of the object can then be described by a 3D descriptor.

In this paper we proposed a method of 3D models retrieval, for 3D triangular mesh models, based on polygons given by cutting each 3D model by paralleled, equidistant and perpendicular plans to X-axis.

The paper is organized as follow. In section 2, we present the 3D Zernike moment and 3D surface moment invariants descriptor. The distance metric between polygonal shapes are proposed in section 3. In section 4, we propose feature vectors based on the similarity measure between polygons. We give the
experimental results where tools used to evaluate our retrieval system, comparison with other 3D descriptors, in section 5. Finally conclusion is presented in section 6.

II. 3D ZERNIKE MOMENT AND 3D SURFACE MOMENT INVARIANTS

A. 3D Zernike moment descriptor

The 3D Zernike functions \( Z^m_\ell(X) \) are written in cartesian coordinates [5] using the harmonic Polynomials \( e^m_\ell \):

\[
Z^m_\ell(X) = \sum_{p=0}^{l} q^p_{\ell,n} |X|^p e^m_\ell(X)
\]

While restricting \( l \) so that \( l \leq n \) and \( l \) be an even number, \( 2k-n-1 \). And the coefficients \( q^p_{\ell,n} \) are determined to guarantee the orthonormality. We are now able to define the 3D Zernike moments \( \Omega^m_{nl} \) of a 3D object defined by \( f \) as

\[
\Omega^m_{nl} = \frac{3}{4\pi} \int f(x) Z^m_{nl}(x) \, dx
\]

Note that the coefficients \( Z^m_{nl} \) can be written in a more compact form as a linear combination of monomials of order up to \( n \):

\[
Z^m_{nl} = \sum_{s+t+u=n} X^s_u \beta_s^l y^r z^t
\]

Finally, the 3D Zernike moments \( \Omega^m_{nl} \) of an object can be written as a linear combination of geometrical moments of order up to \( n \):

\[
\Omega^m_{nl} = \frac{3}{4\pi} \sum_{s+t+u=n} X^s_u \beta_s^l M_{rst}
\]

Where \( M_{rst} \) denotes the geometrical moment of the object scaled to fit in the unit ball:

\[
M_{rst} = \int f(X) x^r y^s z^t \, dx
\]

Where \( X \in \mathbb{R}^3 \) denotes the vector \( X = (x,y,z) \).

The collect of the moments into \((2l+1)\)-dimensional vectors \( \Omega^m_{nl} = (\Omega^m_{nl}, \Omega^m_{n+1l}, \ldots, \Omega^m_{n+l}) \) define the 3D Zernike descriptors \( F_{nl} \) as norms of vectors \( \Omega^m_{nl} \):

\[
F_{nl} = \| \Omega^m_{nl} \|
\]

B. 3D Surface moment invariants descriptor

Dong Xu and Hua Li [9] had used a 3D surface moment invariants as shape descriptors for the representation of free-form surfaces. We consider a 3D surface triangulation \( T = \bigcup_{t \in S} T^t \) consisting of triangles \( T^t \), \( t \in S \subseteq \mathbb{R}^3 \).

The \((k+l+m)\)th order 3D surface moments \( M^k_{lm} \) of \( T \) are the accumulated surface moments \( m^i_{klm} \) of the associated triangles \( T^i \) i.e

\[
M^k_{lm} = \sum_{t \in S} m^i_{klm}
\]

For a general triangle \( \Delta \) the surface moments are

\[
m^i_{klm} = \int_{\Delta} x^k y^l z^m \rho(x,y,z) \, ds
\]

with a surface density function \( \rho(x,y,z) \). Using a surface \( \rho(u,v) = (u,v) \) parameterization in \( \mathbb{R}^3 \) \( D \) is definition domain of \( (u,v) \) in \( \mathbb{R}^2 \).

The moment (8) can be rewritten as:

\[
m^i_{klm} = \int_{\Delta} (u^k v^l w^m) \rho(x,y,z) \, ds = \int_{\Delta} E \, ds
\]

Where \( E = x^2 + y^2 + z^2 \) , \( G = x^2 + y^2 + z^2 \) and \( F = xu, yu, zu \). Then, central moment are defined as

\[
M^k_{lm} = \frac{1}{2} \int (x-\bar{x})^k (y-\bar{y})^l (z-\bar{z})^m \rho(x,y,z) \, ds
\]

So the central surface moments are invariant under translation. Then, we normalize the surface moments by \( M^k_{lm} \), they also became invariant under scaling and can be defined as \( \mu^k_{klm} = \frac{M^k_{lm}}{M^0_{m}} \).

To construct the surface moments invariant under rotation, D. Xu [9] use four geometric primitives for constructing six invariants consist of 3 fourth order, 2 third order and 1 mixed order surface moment invariants.

III. DISTANCE METRIC BETWEEN POLYGONAL SHAPES

To efficiently compute a distance measure between polygonal shapes Esther M. Arkin et al. [1] present an algorithm whose principal idea is to converted a polygon to the turning angle function representation also called Turning function, which will compare the polygons by calculating the distance between their turning functions. The resulting method is invariant to
rotation, scale, and choice of reference point, and is robust to moderate amounts of uniform noise.

A. The turning angle function

Let \( \{v_1, v_2, ..., v_n\} \) a list of vertices of a given polygon, each vertex is represented by their coordinates; to calculate the angle of rotation of the polygon function, we start with the first vertex \( v_1 \) in the representation. The turning angle function representation records the angle of each line segment \( v_i v_{i+1} \) of the polygon makes with the line segment that preceded it in counterclockwise order \( v_{i-1} v_i \), weighted by the length of the segment \( v_i v_{i+1} \), for all \( i = 1, ..., n \).

Because the polygon is convex all the angles considered in Figure 1 are positive (counterclockwise). Had there been reflex (concave) vertices in the polygon, the angles would be negative (clockwise).

The total sum of the angles in the turning angle function is \( \alpha + 2\pi \), where \( \alpha \) is the angle made by the positive x-axis with the first line segment.

The turning angle function of a polygon for each line segment is the cumulative turning angle weighted by the length of this line segment.

Each polygon is rescaled so that the total perimeter length is 1 which will provide an invariance to scale. The length of the polygon be parameterized by \( s \), \( s = 0 \) meaning having made one full revolution and \( s = 1 \) meaning having made one full revolution and come back around to the first vertex.

Let \( \Theta_A(s) \) be the turning angle function for polygon \( A \). For a convex polygon \( A \), \( \Theta_A(s) \) is a monotone function, starting at some value \( \alpha \) and increasing to \( \alpha + 2\pi \). For a non convex polygon, \( \Theta_A(s) \) may become arbitrarily large, since it accumulates the total amount of turn, which can grow as a polygon “spirals” inward. Note also that if we make more revolutions around the perimeter of the polygon, with each revolution the turning function grows by \( 2\pi \).

IV. PROPOSED METHOD

The method we propose consists on representation of 3D objects by polygons set. Initially, the object 3D has an arbitrary position in the space Figure 3(a), and then has been translated so that its center of mass coincides with the origin, as shown in Figure 3(b), then it is scaled to unit sphere and rotated with the PCA method Figure 3(c), to alleviate the problem of rotation invariance.

After the 3D model is aligned, it is cut out by 100 equidistant plane and perpendicular to the Z-axis, the intersection of the 3D mesh model, and this set of plane allows giving a set of polygons. Thus each 3D model is represented by a set of polygons.

To avoid the problem of redundancy and reduce the complexity of the program we select the polygons that best characterize the three-dimensional model by replacing similar polygons by a single representative.

As each 3D model is described by a set of polygons, for measuring the degree of similarity between models, it is sufficient to compare the sets of polygons for each model to others. For it we have chosen to use the Hausdorff distance to the simplicity of implementation and it gives good results in our approach.

Let \( M \) and \( N \) 3D models with their representation by a set of polygons such as \( M(A^1, A^2, ..., A^m) \) and \( N(B^1, B^2, ..., B^n) \). The similarity measure between these models by the Hausdorff distance is defined by:

\[
D_{H}(M, N) = \max_{A \in M} \min_{B \in N} d_2(A, B) \quad \text{or} \quad \max_{A \in M} \min_{B \in N} \left[ \max_{A \in M} \min_{B \in N} d_2(A, B) \right]
\]

IV. PROPOSED METHOD

The method we propose consists on representation of 3D objects by polygons set. Initially, the object 3D has an arbitrary position in the space Figure 3(a), and then has been translated so that its center of mass coincides with the origin, as shown in Figure 3(b), then it is scaled to unit sphere and rotated with the PCA method Figure 3(c), to alleviate the problem of rotation invariance.

After the 3D model is aligned, it is cut out by 100 equidistant plane and perpendicular to the Z-axis, the intersection of the 3D mesh model, and this set of plane allows giving a set of polygons. Thus each 3D model is represented by a set of polygons.

To avoid the problem of redundancy and reduce the complexity of the program we select the polygons that best characterize the three-dimensional model by replacing similar polygons by a single representative.

As each 3D model is described by a set of polygons, for measuring the degree of similarity between models, it is sufficient to compare the sets of polygons for each model to others. For it we have chosen to use the Hausdorff distance to the simplicity of implementation and it gives good results in our approach.

Let \( M \) and \( N \) 3D models with their representation by a set of polygons such as \( M(A^1, A^2, ..., A^m) \) and \( N(B^1, B^2, ..., B^n) \). The similarity measure between these models by the Hausdorff distance is defined by:

\[
D_{H}(M, N) = \max_{A \in M} \min_{B \in N} d_2(A, B) \quad \text{or} \quad \max_{A \in M} \min_{B \in N} \left[ \max_{A \in M} \min_{B \in N} d_2(A, B) \right]
\]

IV. PROPOSED METHOD

The method we propose consists on representation of 3D objects by polygons set. Initially, the object 3D has an arbitrary position in the space Figure 3(a), and then has been translated so that its center of mass coincides with the origin, as shown in Figure 3(b), then it is scaled to unit sphere and rotated with the PCA method Figure 3(c), to alleviate the problem of rotation invariance.

After the 3D model is aligned, it is cut out by 100 equidistant plane and perpendicular to the Z-axis, the intersection of the 3D mesh model, and this set of plane allows giving a set of polygons. Thus each 3D model is represented by a set of polygons.

To avoid the problem of redundancy and reduce the complexity of the program we select the polygons that best characterize the three-dimensional model by replacing similar polygons by a single representative.

As each 3D model is described by a set of polygons, for measuring the degree of similarity between models, it is sufficient to compare the sets of polygons for each model to others. For it we have chosen to use the Hausdorff distance to the simplicity of implementation and it gives good results in our approach.

Let \( M \) and \( N \) 3D models with their representation by a set of polygons such as \( M(A^1, A^2, ..., A^m) \) and \( N(B^1, B^2, ..., B^n) \). The similarity measure between these models by the Hausdorff distance is defined by:

\[
D_{H}(M, N) = \max_{A \in M} \min_{B \in N} d_2(A, B) \quad \text{or} \quad \max_{A \in M} \min_{B \in N} \left[ \max_{A \in M} \min_{B \in N} d_2(A, B) \right]
\]

IV. PROPOSED METHOD

The method we propose consists on representation of 3D objects by polygons set. Initially, the object 3D has an arbitrary position in the space Figure 3(a), and then has been translated so that its center of mass coincides with the origin, as shown in Figure 3(b), then it is scaled to unit sphere and rotated with the PCA method Figure 3(c), to alleviate the problem of rotation invariance.

After the 3D model is aligned, it is cut out by 100 equidistant plane and perpendicular to the Z-axis, the intersection of the 3D mesh model, and this set of plane allows giving a set of polygons. Thus each 3D model is represented by a set of polygons.

To avoid the problem of redundancy and reduce the complexity of the program we select the polygons that best characterize the three-dimensional model by replacing similar polygons by a single representative.

As each 3D model is described by a set of polygons, for measuring the degree of similarity between models, it is sufficient to compare the sets of polygons for each model to others. For it we have chosen to use the Hausdorff distance to the simplicity of implementation and it gives good results in our approach.

Let \( M \) and \( N \) 3D models with their representation by a set of polygons such as \( M(A^1, A^2, ..., A^m) \) and \( N(B^1, B^2, ..., B^n) \). The similarity measure between these models by the Hausdorff distance is defined by:

\[
D_{H}(M, N) = \max_{A \in M} \min_{B \in N} d_2(A, B) \quad \text{or} \quad \max_{A \in M} \min_{B \in N} \left[ \max_{A \in M} \min_{B \in N} d_2(A, B) \right]
\]
V. EXPERIMENTAL RESULTS

For experimentation, we have used 3D models extracted from the database Princeton Shape Benchmark [7], publicly available and well known. This database contains an 1814 3D mesh models. For each 3D model, there is an Object File Format (.off) file with the polygonal geometry of the model, a model information file, and a JPEG image file with a thumbnail view of the model. Along with the 3D models, a hierarchical classification of the models is provided.

Performance of our system is studied by using each 3D model in the database as the query and top 20 similar images are retrieved by Hausdorff distance. We have developed a graphic user interface that allows us to make a subjective quality assessment (see Figure 5). The query model is located in the upper left and the most similar objects are ordered from top to bottom and left to right.

The performance of each retrieval method is estimated by comparing the fused results in terms of precision and recall. This is a traditional and a common way for evaluating performance in documental and visual information retrieval. The recall measures the ability of the system to retrieve all models that are relevant; and the precision measures the ability of the system to retrieve only relevant models. They are defined as follows:

\[
\text{Recall} = \frac{\text{relevant correctly retrieved}}{\text{all relevant}}
\]

\[
\text{Precision} = \frac{\text{relevant correctly retrieved}}{\text{all retrieved}}
\]

To well show the effectiveness of the method, we compared our results with the Zernike moments descriptor [5] and surface moment invariants descriptor [9] (Figure 6).

VI. CONCLUSION

We have presented in this paper an indexing and 3D shape retrieval system. It consists to cut each 3D model into a set of polygons. On the first time, we compute the turning angle function of each polygon, in second we use the distance between turning functions and Hausdorff distance to measure the similarity between 3D models. To check the retrieval performance we have design experiments performing 3D model retrieval that has been shows good performance in comparison with the two methods Zernike moments descriptor and 3D surface moment invariants descriptor.

REFERENCES


