ABSTRACT
Along with images and videos, 3D models have recently gained increasing attention for a number of reasons: advancements in 3D hardware and software technologies, their ever decreasing prices and increasing availability, affordable 3D authoring tools, and the establishment of open standards for 3D data interchange.

In this paper we address the problem of content-based retrieval of 3D models. The solution we present here relies on the description of each model by means a curvature map: After an initial pre-processing of the model, differential properties of points on the surface of the 3D object are evaluated; the model surface is then warped into an ellipsoid, and is mapped onto a 2D image retaining curvature information of the original model. Matching is performed by comparing the 2D map of the query against the 2D maps of the database models. This method has been implemented in a prototype system supporting retrieval by content of 3D objects through a web interface.

1. INTRODUCTION
Digital multimedia information is nowadays spreading through all sectors of society. Larger and larger collections of multimedia documents are being created at an increasing pace, enabling the exchange of information over computer networks and, consequently, the use of this information by a wide range of users. However, in order to exploit the valuable assets contained in these ever growing collections, users need to find information that matches their expectations—a notoriously hard problem, due to the inherent difficulties of managing multimedia documents. In recent years, many systems have been developed that enable effective retrieval from digital libraries, covering text, audio, images, and videos (refer to [2] for a detailed survey about images and videos).

Content-based retrieval of 3D information is a new challenge for researchers and practitioners. Development of techniques supporting archival, indexing and retrieval of 3D models is of paramount importance in a variety of application domains. For instance, this is particularly the case in the fields of cultural heritage and historical relics, where there is an increasing demand for solutions enabling preservation of relevant artworks (e.g. vases, sculptures, and handicrafts) as well as cataloguing and retrieval by content. In these fields, retrieval by content can be employed to detect commonalities between 3D objects (e.g. the “signature” of the artist) or to monitor the temporal evolution of a defect (e.g. the amount of bending for wooden tables). Tools supporting retrieval of 3D models are also expected to play a key role in educational programs, either traditional or computer-based.

Solutions proposed so far to support retrieval of images and videos cannot always be adapted to 3D models, as they do not account for the peculiar nature of 3D data. In fact, while images and videos are mostly views (static and dynamic, respectively) of real world objects and scenes, 3D models embody their essence of such objects.

In this paper we address the problems of description and matching of 3D objects. The basic idea underlying our approach is that the shape of a 3D object can be described through a curvature map of its surface. After an initial pre-processing of the model, differential properties of points on the surface of the 3D object are evaluated; the model surface is then warped into an ellipsoid, and is mapped onto a 2D image retaining curvature information of the original model. Matching is performed by comparing the 2D map of the query against the 2D maps of the database models.

The paper is organised as follows: Sec.2 surveys related work in the field of 3D CBR; Sec.3 expounds on our method, describing the techniques used for analysis, description, and retrieval of 3D models; then, in Sec.4 experimental results are presented.

2. RELATED WORK
Description and retrieval of information in the form of 3D data sets is recently receiving an increasing attention. Some 3D data archives already exist and they are expected to grow in both size and number. Methods addressing retrieval of 3D models can be classified according to different aspects, including the type of representation used for geometry, whether or not information on the models’ aspect (i.e. colour and/or texture) is also considered, the need for manual intervention.

Description of 3D volumetric data for the purpose of retrieval by shape similarity has been addressed in [10] and [1]. In [10] the shape of 3D objects is represented through algebraic moment invariants. Differently, in [1] 3D shape histograms are exploited to support retrieval of 3D protein structures.

Description of 3D surface data for the purpose of recognition or retrieval has been addressed in [5, 8]. Some authors have investigated analytical models, but this is not always a viable solution, as there are many limitations in providing parameterizations of arbitrary models. In [5] retrieval of 3D objects based on similarity of surface segments is addressed. Surface segments model potential docking sites of molecular structures. The proposed approach develops on the approximation error of the surface. However, assumptions on the form of the function to be approximated limit the application of the approach to special contexts.

Much attention has been recently devoted to free-form (i.e. polygonal) meshes. While this representation of 3D models poses major hurdles to development and implementation of algorithms, it is indeed the most appealing field of application. The system
developed within the Nefertiti project supports content-based retrieval of 3D models based on both geometry and appearance (i.e. colour and texture) [8]. For the purpose of describing geometry the system relies on aspect ratio, on a wavelet decomposition of a voxel-based representation of the volume, and on the distribution of cords angles and lengths. Also Kolonias et al. have used dimensions of the bounding box (i.e. its aspect ratios) and a binary voxel-based representation of geometry [6]. They further relied on a third feature, namely a set of paths, outlining the shape (model routes). In [7] a method is proposed to select feature points which relies on the evaluation of Gaussian and median curvature maxima, as well as of torsion maxima on the surface. To reduce sensitivity to noise, a preliminary iterative smoothing is carried out on the models. In [3], Elad et al. rely on moments (up tp the 4-th order) of surface points as basic features to support retrieval of 3D models. Differently from the case of 2D images, evaluation moments is not affected by (self-)occlusions.

3. ANALYSIS, DESCRIPTION, AND RETRIEVAL OF 3D OBJECTS

Description of the geometric structure of a 3D object is accomplished through the following steps: Pre-processing and Curvature estimation, Warping, Curvature mapping. In the first step, the 3D object model—that is represented as a 3D mesh— is subject to polygonal reduction and smoothing. Then, curvature of the 3D surface is estimated for each vertex of the mesh. Information about curvature is used to annotate each vertex. In the second step, the 3D object mesh is subject to a deformation process. By acting on the (x, y, z) position of each mesh vertex, this process aims at transforming the 3D object mesh into a sphere. At the end of the deformation process, each vertex of the mesh keeps track of its curvature on the original 3D object. Once warping is complete, the annotated spherical mesh is projected onto an image that encodes position and curvature of points on the original object surface. This image is used as a descriptor of the geometric structure of the original 3D object.

3.1. Pre-processing and Curvature estimation

High resolution 3D models obtained through scanning of real world objects are often affected by high frequency noise, due to either the scanning device or the subsequent registration process. Hence, smoothing is mostly required when dealing with such models. Selection of a smoothing filter is a critical step, as application of some filters entails changes in the shape of the models. For instance, mean or Laplacian smoothing cause shrinking of the model. In Laplacian smoothing, every vertex $x$ is moved from its original location by an offset $\Delta(x)$; the offset is determined as a function of the neighbouring vertices of $x$, and a parameter $\lambda$ controls the strength of the filter. To avoid shrinking, we adopted the filter first proposed by Taubin [11]. This filter, also known as $\lambda|\mu$ filter, operates iteratively, and interleaves a Laplacian smoothing weighed by $\lambda$ with a second smoothing weighed with a negative factor $\mu$ ($\lambda > 0, \mu < -\lambda < 0$). This second step is introduced to preserve the model’s original shape.

Before features can be extracted to provide a description of a generic 3D model, some pre-processing is required both to reduce the complexity of the model, and to remove high frequency noise. To achieve the first goal, an algorithm performing an iterative contraction of vertex pairs (i.e. edges) is used: first, all edges are ranked according to a cost metric; then, the minimum cost vertex pair is contracted; finally, the costs are updated [4].

An approach to evaluation of curvature of polygonal meshes is described by Taubin, which uses a Laurent series approximation to estimate the tensor of curvature. In the proposed approach, estimation of surface curvature at a generic vertex $v_i$ of the mesh is accomplished by considering variations of surface normal over the platelet $V^{v_i}$ of vertex $v_i$. This guarantees less sensitivity to noise and acquisition errors.

In particular, surface curvature in correspondence with the i-th vertex $v_i$ of the mesh $\mathcal{M}$ is estimated by considering versor $v_i^+$, that is, the normal to $\mathcal{M}$ at point $v_i$. Then, the platelet $V^{v_i}$ of vertex $v_i$ is considered. This is defined as the set of all mesh vertices around $v_i$. Given a generic vertex of the platelet $v_j \in V^{v_i}$ let $v_j^+$ be the normal to $\mathcal{M}$ at point $v_j$. Mesh curvature $\gamma_{v_i}$ at vertex $v_i$ is estimated as:

$$
\gamma_{v_i} = \frac{1}{2} \sum_{v_j \in V^{v_i}} |v_j^+ - v_j^-| \frac{|v_j^+|}{|v_j^-|} \tag{1}
$$

It can be shown that with this definition, the value of $\gamma_{v_i}$ is always in $[0, 1]$.

3.2. Deformation

The aim of the mesh deformation process is to transform the original mesh into a sphere. More precisely, it is not necessary for the transformed mesh to correspond exactly to a sphere. Rather, it suffices for the transformed mesh to be described as a function on the sphere. Mesh deformation is obtained by iteratively applying a smoothing operator to the mesh. In general, application of a smoothing operator is accomplished by updating the position of each vertex of the mesh according to the following formula:

$$
\mathcal{M}(v_i) \otimes \omega = \sum_{v_j \in V^{v_i}} \omega_j \sum_{v_j \in V^{v_i}} w_j \ast v_i - v_i \tag{2}
$$

being weights $\omega = \{w_j\}$ characteristic of each operator and $\mu$ a parameter used to control the amount of motion of each vertex and to guarantee stability and continuity of the smoothing process.

Under the assumption of low $\mu$ values, the iterative application of the smoothing operator to every vertex of the mesh is equivalent to an elastic deformation process. During the deformation process each vertex of the mesh should be moved in order to satisfy to sometimes opposite requirements: mesh regularization and curvature minimization. To meet the first requirement, positions of the vertices are to be moved so as obtain a uniform mesh. This is a mesh where the distance between a generic vertex and its first order neighbors is almost constant. To fulfill the second requirement, the position of a generic vertex should be updated so as to decrease the absolute value of the local curvature.

As demonstrated in previous work, application of Laplacian Smoothing, Taubin Smoothing, or Bilaplacian Flow operators increases mesh regularization but may result in unnatural deformations of the original mesh. Differently, application of Mean Curvature Flow operator doesn’t guarantee mesh regularization [13].

To achieve both regularization and smoothing of the original mesh, the proposed solution develops on the application of two distinct operators at each step of the iterative deformation process. In particular, Laplacian and Gaussian Curvature operators are used in combination to achieve both mesh smoothing and regularization.
Fig. 1. Mapping a sphere onto a plane: a) Mercator and b) Archimedes projections. The Mercator projection is a conformal projection, and rhumb lines (i.e. paths with a constant compass direction) are mapped into straight lines. The Archimedes projection is a non conformal projection preserving the area of any set.

Application of the two operators is iterated until the average value of vertex motion falls below a predefined threshold.

3.3. Mapping

Projection of a curved surface is a well known problem in cartography [9]. There are many different projections used to map (a part of) the globe onto a plane, but their description is far beyond the scope of this paper.

In our approach, we have elected an area preserving projection, the Archimedes projection (also known as the Lambert equal-area projection). Similarly to the Mercator projection, the Archimedes projection is a cylindrical projection (Fig.1). In particular, it is the projection along a line perpendicular to the axis connecting the poles and parallel to the equatorial plane. Thus, a point on the sphere with latitude $\Theta$ and longitude $\Phi$, is mapped into the point on the cylinder with the same longitudinal angle $\Theta$ and height $\sin(\Phi)$ above (or below) the equatorial plane. In other words, this map is created by wrapping a cylinder around the equator and then projecting along lines of constant latitude. When the cylinder is unrolled, a flat coordinate system is produced. A cylinder can be unrolled without creating distortion in the east-west directions, the distortion being limited to north/south only. Areas close to the equator exhibit little distortion either way.

3.4. Description and matching

Once a 3D model is represented through a 2D map, any approach supporting image retrieval by visual similarity can be used to evaluate the similarity between two 3D models. In fact, this can be achieved by computing the similarity of the corresponding maps.

As already mentioned, different projections display different properties, and introduce different types of distortion in the planar map. The Archimedes projection we have chosen preserves regions’ areas; however, it implies a severe distortion of regions’ shapes: the closer a region to one of the poles the more its shape on the map appears distorted. As a consequence, techniques that evaluate image similarity based on shape features may prove inadequate for the problem at hand. Similar considerations apply to techniques relying on texture features, especially if these are based on some geometric primitive. Hence, to take full advantage of properties of the Archimedes projection, the similarity between two image maps should be computed based on region area and their spatial arrangement.

Several techniques have been proposed so far to represent information about the frequency distribution of features within an image. Among these, histograms are probably the most commonly employed, both for their simplicity of use and for some convenient properties. These include low storage requirements invariance to image scaling and rotation, as well as ease of combination with indexing structures. Histograms provide a synthetic representation for content, and have been used for different features, such as color and shape. The above properties make histograms very appealing for the problem at hand.

To support local as well as global descriptions of the models we have chosen to work with three levels of detail: at the highest level of detail the image is partitioned according to a regular grid comprising 32 tiles, at the intermediate level the grid comprises 8 tiles, and at the lowest level a single tile covers the whole image. Following the above discussion, histograms at the lower levels can be computed from histograms at the highest level.

For the implementation of the system presented in this paper, 16 reference curvature values were selected to quantize the curvature space, and histograms with 16 bins are used to encode curvature information for each of the tiles images are partitioned into. Histograms are normalized with respect to the image size so as to provide scale invariance of the representation. A distance matrix $A$ has been defined for the computation of the Mahalonobis distance between curvature histograms.

3.4.1. Similarity Computation

Since the content of a map is represented at several resolution levels, the computation of the similarity between two maps relies on matching descriptors at equal resolution levels. At a generic resolution level, each map is partitioned into $n$ tiles and each tile is represented through a 16-bins curvature histogram.

Computing the distance between two maps requires to find the best tiles correspondence function. This is defined as the permutation $p: \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ that minimizes the sum of distances between corresponding tiles, that is:

$$D_{map}^i (M, M') = \min_p \left\{ \sum_{i=1}^{n} D_{KS}(H^i_{m}, H'_{p(i)}) \right\}$$

being $H^i_{m}$ the histogram of the i-th tile in the first map and $H'_{p(i)}$ the histogram of the $p(i)$-th tile in the second map.

The solution $p$ to Eq. (3) is approximated through a heuristic search approach that requires to scan all tiles in the first map in a predefined order and associate to each tile the most similar tile not yet associated in the second map. This pairwise NN association yields a suboptimal solution.

4. EXPERIMENTAL RESULTS AND COMPETITIVE ASSESSMENT

Approximately 120 models were collected to build the test database. These comprise three classes of models: taken from the web, manually authored (with a 3D CAD software), and variations of the previous two classes (obtained through deformation or application of noise, which caused surface to be moved points from their original locations). Feature descriptors were then evaluated and added to the index.

Fig.2 shows a retrieval example where the model of a bunny was selected as a query template (upper left). Retrieved models are shown below in decreasing order of similarity. The system interface shows the first 20 matched models. In the database there
are only 5 bunny models and they are all retrieved in the first 6 positions. Fig.3 provides insight into the matching process, showing the three best matches between regions of two different models. In Fig.4 values of precision and recall (averaged over 20 different queries) for the proposed system are compared with values of precision and recall for two alternative systems: one based on histograms of curvature (similar to [12]), the other based on moment descriptors (similar to [10]).

5. REFERENCES


Fig. 2. The web interface of the 3D content-based retrieval system. In the upper left corner the query is shown. Retrieved models are displayed below it.

Fig. 3. The figure shows the best three correspondences of regions of the curvature maps of two bunny models, backprojected onto the original models.

Fig. 4. Values of precision and recall (averaged over 20 different queries) for the proposed system ( ), the system based on histograms of curvature ( ) and system based on moment descriptors ( ).