Multimed Tools Appl (2006) 31: 29–50
DOI 10.1007/s11042-006-0034-2

Content-based retrieval of 3D models through curvature maps: a CBR approach exploiting media conversion

Jürgen Assfalg · Alberto Del Bimbo · Pietro Pala

Published online: 5 August 2006

Abstract Multimedia digital libraries are gaining an increasing relevance, and are growing in both number and size. Moreover, advancements in acquisition and authoring technologies have also caused new media types to be added to these libraries. For instance, animations and 3D models have recently come alongside more traditional media such as text, images, audio and video. In particular, even if 3D objects were initially included in multimedia digital libraries in the form of images (i.e., views of the objects), they are now also available in the form of 3D models. These allow to overcome the inherent limitations of an image-based representation of a three-dimensional reality. In this paper we present a novel method for description and retrieval of 3D models that relies on curvature maps: 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 deformed into an ellipsoid, and is mapped onto a 2D image retaining curvature information of the original model. Through this media conversion from 3D model to image map, we enable application of analysis, description, and matching techniques which have been originally conceived for the image medium. In particular, the two approaches described in this paper rely on histograms for a coarse but efficient description of tiles of the curvature map, and on weighted walkthroughs for a more precise description of homogeneous regions of the map, taking into account their spatial arrangement and areas. Experimental results reveal that both techniques can be successfully applied to curvature maps. The two approaches have been compared against other solutions presented in literature, and have shown a good retrieval performance (particularly in the case of region-based description).

Keywords Content-based retrieval · 3D models · Curvature maps · Media conversion

J. Assfalg (✉) · A. Del Bimbo · P. Pala
Dipartimento di Sistemi e Informatica, Via S. Marta 3, 50139 Firenze, Italy
e-mail: assfalg@dsi.unifi.it

A. Del Bimbo
e-mail: delbimbo@dsi.unifi.it

P. Pala
e-mail: pala@dsi.unifi.it

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 fact, storage, indexing and retrieval by content of information from such collections represent issues that researchers have now been facing for some time. In recent years, as a result of the efforts spent in the attempt of finding solutions to these problems, many systems have been developed that enable effective retrieval from digital libraries, covering text, audio, images, and videos. For images, solutions have been proposed to support retrieval based on colour, texture, shape, and spatial relationships [12, 33]. In this field, researchers have long investigated content descriptors best suited to capture the properties of images so as to embody users' perception in the retrieval process. Also, in order to meet the users' expectations in terms of matching, different metrics have been experimented to evaluate similarity measures between descriptors of two different images [32]. For videos, additional clues, related to motion and video editing effects, have been exploited [10–12, 30, 44]. Along with the low-level cues, research in the area of videos has specifically addressed the problem of automatic semantic annotation, since querying by high-level concepts is a pervasive task in video retrieval [2].

Along with images and videos, 3D models have recently gained increasing attention for a number of reasons: advancements in 3D hardware and software technologies—in particular for acquisition, authoring and display [3, 9, 43]—their ever decreasing prices and increasing availability, and the establishment of open standards for 3D data interchange (e.g., VRML, X3D). Acquisition of a 3D model, in terms of geometry and visual appearance, can be achieved through many different techniques, ranging from CAD to 3D laser scanners and computer vision. Selection of a specific technique typically depends on application specific quality requirements. CAD yields noiseless models, but requires manual authoring, which implies a tradeoff between accuracy and cost. Three-dimensional scanners and computer vision allow for the automation of at least part of the acquisition process, but are affected by sensor noise. Also, acquisition through scanners initially produces a huge amount of data, which poses severe system requirements. Computer vision, instead, often requires a time consuming calibration in order to attain accurate results. Application of these techniques results in a large variety of models, differing in terms of their representation (e.g., point clouds, simplicial complexes, voxels, analytical functions), of their resolution (e.g., number of points/polygons) and size, of the presence, nature, and amount of noise and artifacts (e.g., sensor noise, occlusions).

As a consequence of the larger availability of enabling technologies, 3D models are being employed in a wider and wider range of application domains, including medicine, computer aided design and engineering, and cultural heritage. In medicine, 3D models are used to analyze the structure of molecules for preparing drugs, to catalogue bones and teeth, and to simulate surgical operations in virtual reality applications. CAD has long profited from 3D modelling, and is now extending its range of applications, from mechanical to civil engineering, through architecture and interior design. In these fields, 3D models are used to represent complex structures, often in terms of a composition of elementary modules. Reuse of existing models is here a current practice. Finally, cultural heritage is a field which is profitably exploiting 3D models both to preserve the artifacts mankind has produced over time and to allow for their unlimited availability to scholars and art enthusiasts [8, 25].

In this framework content-based retrieval assumes an ever increasing relevance, as it can support efficient and effective retrieval of target 3D models. 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 the essence of such objects, which is independent of the viewpoint. In other words, while a generic view can be constructed from a given 3D object, the opposite does not always hold. Moreover, the inherent differences between 3D models on the one side and images and videos on the other affect different aspects in their processing and management chain, including acquisition, archival, description, querying, and matching.

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 deformed 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. In the first approach we propose, the curvature map is subdivided into a grid of rectangular tiles; these tiles are described through histograms. In the second approach, the map is segmented into regions of homogenous curvature, and regions are described with weighted walkthroughs. The proposed approaches have been implemented in a prototype system supporting retrieval by content of 3D objects.

The paper is organised as follows: Section 2 surveys related work in the field of retrieval by content of 3D objects; Section 3 provides insight into preliminary steps of our method, describing the techniques used to evaluate the curvature map of a 3D model; Section 4 expounds on description and matching of curvature maps, based on both histograms of tiles and weighted walkthroughs of homogeneous curvature regions; then, Section 5 presents experimental results and a comparative assessment; finally, Section 6 comprises a final discussion and conclusions.

2 Related work

While some 3D data archives already exist,[1] they are expected to grow further in both size and number. Methods supporting retrieval of 3D models can be classified according to: the type of representation used for 3D model geometry, (i.e., volume or boundary representations—the former describing the whole volume of an object, and the latter only

[1] e.g., The Macromolecular Structure Database, http://research.bmn.com/msd and http://msd.ebi.ac.uk

its surface), the models' 2D visual appearance (i.e., its colour, projected shape, and texture), associated metadata and high-level concepts (either manually or automatically annotated).

Retrieval of 3D VRML object models based on textual metadata was early proposed in [15]. In some cases, image-based representations of 3D models have been exploited (e.g., multiple views or range images). In [5] retrieval of CAD object models is supported by describing the objects through a 2D representation of their contours, taken from different viewpoints.

Retrieval of 3D volumetric data based on shape similarity has been addressed in [37] and [1]. In [37] 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. Both approaches provide a global description of objects' geometries.

Some authors have investigated analytical representations of 3D models: in [22] retrieval of 3D objects based on similarity of surface segments is addressed. Surface segments model potential docking sites of molecular structures. The approach develops on the approximation error of the surface described by $z = f(x, y)$. Points on the surface are first grouped into segments, and then two features are computed for each segment: the parameter set that yields the minimum approximation error (given a set of approximating functions), and the 3D extension vectors (computed from the principal moments of inertia). Similarity between a query surface and a database surface is computed with a quadratic distance measure, which depends on the query surface. However, assumptions on the form of the function to be approximated limit the application of the approach to special contexts. In particular, since the model cannot be extended to represent the entire molecular surface, retrieval based on global similarity is not supported.

Retrieval of free-form meshes has been presented in [14, 15, 23, 26, 28]. In fact, an ever growing amount of 3D models is available in VRML format, as this fosters a common representation, independent of the acquisition or modelling techniques used to create the model.

The system developed within the Nefertiti project supports content-based retrieval of 3D models based on both geometry and visual appearance (i.e., colour and texture) [28]. For the purpose of describing geometry, the system relies on aspect ratio, on a wavelet decomposition of a voxel-based representation of the volume enclosed by the mesh, and on the distribution of *cords*' angles and lengths. Cords are vectors from the center of mass to the center of the polygons comprising the surface. Using cords rather than surface normals makes the system less sensitive to local variations of their orientations.

Also Kolonias et al. have used dimensions of the bounding box (i.e., its aspect ratios) and a binary voxel-based representation of geometry derived from the polygonal mesh [23]. They further relied on a third feature, namely a set of paths outlining the shape (*model routes*). Model routes are computed according to a set of heuristics, and combine a number of edges forming a more or less straight polyline. To perform retrieval, a distance is computed between the reference and database models; this is a weighted combination of distances between corresponding features. A recent study on the usage of shape distributions has been reported in [29] (*cords* fall into this category). Different shape functions (mapping geometric properties to a scalar value) have been assessed. Shape functions are typically computed on couples or triples of random points on the surface. The distribution (i.e., histogram) of such values is then evaluated, and used as a signature for the corresponding model.

In [26] 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.

The first step in matching develops on a geometric hashing of the aforementioned feature points; in a second step, preliminary results are refined through global verification. Geometric hashing techniques for model-based recognition of 3D objects have been previously used in [24]. Different orientations of the surface of the object are stored in a hash table mapping triplets of surface points into their orientations.

In [14], Elad et al. rely on moments (up to the 4th–7th 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. The retrieval engine exploits a Support Vector Machine to implement a relevance feedback mechanism. This usually allows queries to converge within 1–4 iterations.

3 From 3D models to curvature maps

Description of the geometric structure of a 3D object is accomplished through the following steps: *pre-processing, curvature estimation, deformation, 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, during the second step, curvature of the 3D surface is estimated for each vertex of the mesh. Information about curvature is used to annotate each vertex. In the third 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 collapsing the 3D object mesh onto a 3D sphere. At the end of the deformation process, the 3D object mesh roughly resembles a sphere and each vertex of the mesh keeps track of its curvature on the original 3D object. Once deformation 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.

These four steps are described in detail in the following subsections.

3.1 Smoothing and polygon simplification

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. Further, smoothing is also required when objects (and, therefore, the corresponding models) display textured surfaces, but the focus of attention is on their global shape, and not on their local surface properties. 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 (a known problem, which has been pointed out—for example—in [26, 40]. 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 neighboring vertices of x, and a parameter λ controls the strength of the filter. To avoid shrinking, we adopted the filter first proposed by Taubin [38]. This filter, also known as $\lambda|\mu$ filter, operates iteratively, and interleaves a Laplacian smoothing weighed by λ with a second smoothing weighed with a negative factor μ ($\lambda > 0$, $\mu < -\lambda < 0$). This second step is introduced to preserve the model's original shape. The quality of the Taubin filter, compared with the median and Laplacian, is shown in figure 1.

In many cases, some additional pre-processing is required to reduce the complexity of the model. To this end, 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

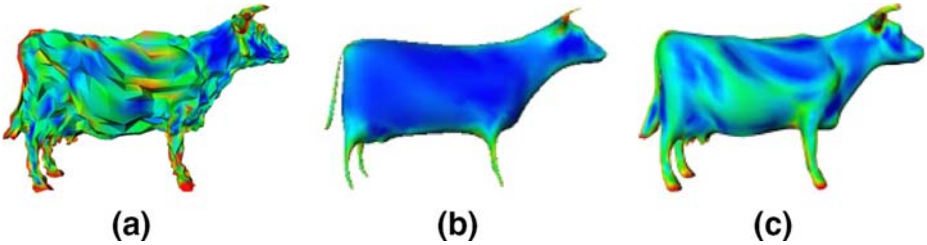


Fig. 1 Different smoothing filters behave differently on a 3D model: (a) the original (noisy) 3D model, (b) the model obtained by applying a mean smoothing filter, and (c) the model obtained through Taubin filtering. The Taubin filter prevents shrinkage of (parts of) the model, which, instead, occurs with the mean smoothing filter (see, for instance, the legs of the cow)

vertex pair is contracted; finally, the costs are updated [19] (see figure 2). The algorithm is iterated until a predefined stop criterion is met (e.g., desired number of polygons for the simplified model). This approach displays some advantages over others: it is simpler than vertex removal; it is well defined on any simplicial complex; as a by-product, it induces a hierarchy. In our experiments, we carried out supervised polygon reduction, where the stop criterion was set in terms of the number of polygons of the final model. Once the algorithm had been applied to the models in the database, the resulting models were inspected to assess their quality. If some model was over-simplified, the simplification process was carried out again, on the original model, by setting a higher number of polygons as the goal of the simplification.

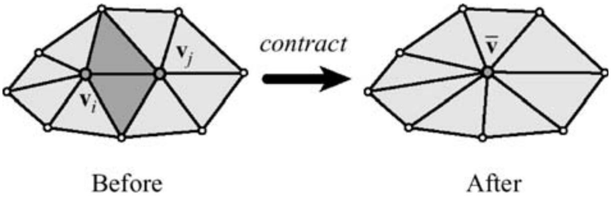
Smoothing is typically applied prior to polygon reduction. However, since this latter process may produce some noise, an additional smoothing iteration can be also applied afterwards.

3.2 Curvature estimation

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 [39]. A somewhat different approach is proposed in [27], where the estimation is not limited to adjacent triangles, but rather extends to a geodesic neighborhood. While this latter approach reduces sensitivity to noise, it requires an algorithm for evaluation of the neighborhood. 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 $v_i^{\perp}$, 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

Fig. 2 Polygonal simplification is obtained through edge contraction: iteratively, the edge displaying the lowest cost is contracted into a single vertex, until a given stop criterion is met



v_i. Given a generic vertex of the platelet $v_j \in V^{v_i}$ let $v_j^{\perp}$ be the normal to $\mathcal{M}$ at point v_j. Mesh curvature γ_{v_i} at vertex v_i is estimated as:

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

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

3.3 Deformation

The aim of the mesh deformation process is to transform the original mesh into a sphere or, more precisely, into a new mesh that can be described as a function on a 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:

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

being weights $\omega = \{w_j\}$ characteristic of each operator and μ 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 μ 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 two sometimes opposite requirements: mesh regularization and curvature minimization. The former requirement implies transformation of the mesh into a uniform mesh—i.e., a mesh for which the distance between a generic vertex and its first order neighbors is almost constant. Ideally, vertices of the mesh could be moved so as to minimize the variance of mesh edges. Whereas, to meet the latter 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 [4], 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.

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 smoothing operators are used in combination to achieve both mesh smoothing and regularization. Application of the two operators is iterated until the average value of vertex motion falls below a predefined threshold τ.

As an example, in figure 3 results of three deformation processes are shown, representing deformations of a bunny model. The first and second deformation processes (figures 3b–c) result from the application of one operator only (Laplacian and Gaussian, respectively). The third process, instead, results from the application, at each iteration step, of both Laplacian and Gaussian operators (figure 3d). It can be noticed that application of the sole Laplacian operator results in unnatural deformations. Differently, application of the sole Gaussian curvature operator results in a non-uniform mesh: especially of regions corresponding to the tips of the ears.

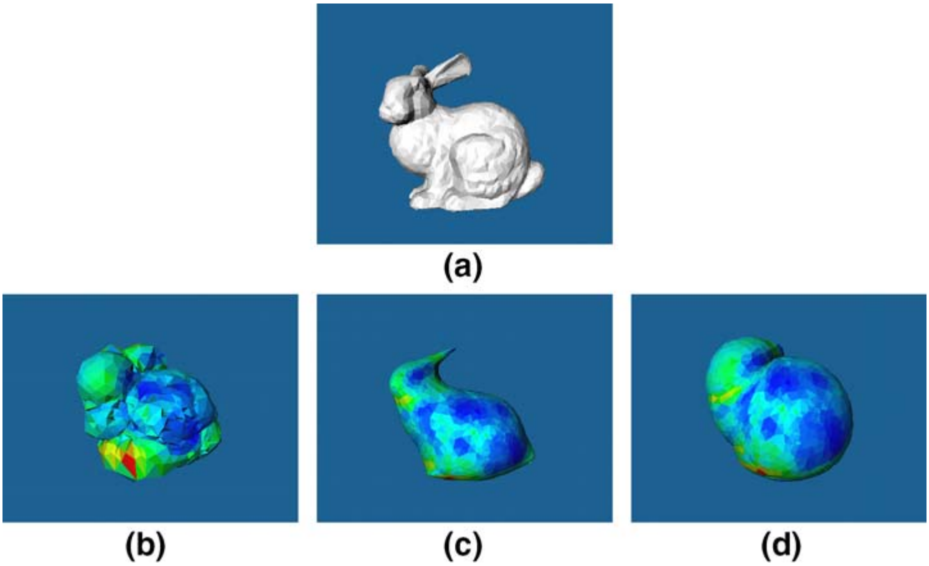


Fig. 3 Three deformation processes: **a** Laplacian operator only, **b** Gaussian operator only, and **c** Mesh deformation using both Laplacian and Gaussian curvature operators

3.4 Mapping

Projection of a curved surface is a well known problem in cartography [35]. 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.[2] Each one has its peculiar advantages and disadvantages. For instance, the gnomonic projection transforms geodesics (i.e., curves of shortest length connecting two points) into straight line segments; this is of paramount utility when searching for the shortest route on the globe. Instead, the Mercator projection transforms rhumb lines (i.e., curves on the globe that are traced when keeping a constant direction) into straight lines, easing the navigation task in that these lines can be followed by keeping the same direction. However, a distortion is inevitably associated with the projection process, whatever the projection is. Hence, it is never a matter of “if” but always a matter of “how much” and of “what kind”. In fact, it is not possible to map even a portion of a spherical surface onto a plane without introducing some distortion.

In our approach, we have selected the Archimedes’ projection (also known as the Lambert equal-area projection). Similarly to the Mercator projection, the Archimedes projection is a cylindrical projection (see figure 4). 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 Θ and longitude Φ, is mapped into the point on the cylinder with the same longitudinal angle Θ 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

[2] For a detailed analysis the interested reader can also refer to
http://thierry.hatt.gps.free.fr/projections/projections-cartes.htm
http://math.rice.edu/~polking/cartography/cart.pdf
http://www.mentorsoftwareinc.com/cc/gistips/TIPSarch.HTM

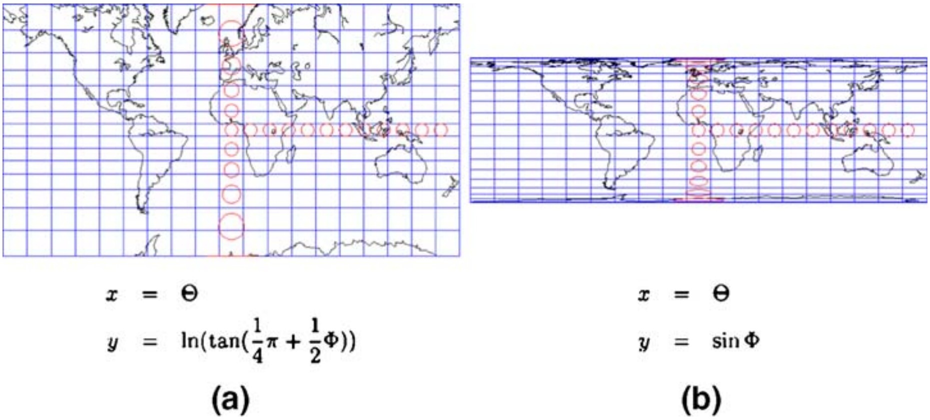


Fig. 4 Mapping a sphere onto a plane: (a) Mercator, and (b) Archimedes projections. The Mercator projection is a conformal projection, and rhumb lines (i.e., paths with a constant compass direction) are mapped into *straight lines*. The Archimedes projection is a non conformal projection preserving the area of any set

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.

A major advantage of the Archimedes' projection is that it is an area preserving projection: all regions on the surface of the sphere are mapped into regions on the map having the same area. This guarantees that, regardless of the position on the sphere, the relevance of any region is the same both on the sphere and on the map.

In general, projecting a spherical surface onto a 2D map entails one degree of freedom that limits map invariance to certain geometrical transforms (rotation and translation). For instance, the actual mapping that is obtained through the Archimedes' projection depends on the position of the poles. In order to cope with this problem, two alternative strategies can be adopted. One possible solution is used before the generation of the map and is based on the definition of a criterion that reduces the degree of freedom by defining a suitable constraint. Alternatively, invariance can be achieved later on, in the matching phase, by adopting a similarity measure that is not sensitive to the aforementioned transforms.

In our case, we decided to adopt the first strategy. In particular, the first two inertia axes are used to determine position of poles on the sphere and identification of the cutting plane, used to unroll the sphere.

4 From curvature maps to model descriptors

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

In practice, some care must be put in the choice of the combination of map projection and image description techniques. For instance, the Archimedes projection preserves regions' areas but 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). 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. Whereas, techniques that evaluate image similarity based on shape features may prove inadequate for this projection technique. Similar considerations apply to techniques relying on texture features, especially if these are based on some geometric primitive.

In the proposed approach, information about curvature maps is captured at two distinct levels: tiles obtained by a uniform tessellation of the map, and homogeneous regions obtained by segmenting the map. In the former case, we use histograms to capture global properties of map tiles, whereas in the latter case we rely on weighted walkthroughs to describe spatial arrangement and local properties of regions on the map. Details on the two techniques are provided hereafter.

4.1 Histogram-based description of map tiles

Several techniques have been proposed so far to represent information about the frequency distribution of features within an image [17, 20, 31, 34, 36]. Among these, histograms are probably the most commonly employed, both for their simplicity of use and for some convenient properties (colour histograms have been included in the basic features covered by the MPEG-7 standard). These include low storage requirements, invariance to image scaling and rotation, as well as ease of combination with metric indexing structures. Histograms provide a synthetic representation for content, and have been used for different features, such as colour and shape [17, 21]. The above properties make histograms very appealing for the problem at hand.

Histograms notation and properties A generic histogram H with n bins is an element of the histogram space $\mathcal{H}^n \subset \mathbb{R}^n$. Given an image and a quantization of a feature space, histogram bins count the number of occurrences of points of that quantized feature value in the image.

Histograms also support a multi-resolution description of image features. Given a partitioning of an image into n fine-grained tiles, histograms provide a representation for the content of each of these tiles. The representation of a wider tile, at a less fine-grained level, can be computed by merging the histograms of every tile $\mathcal{R}_k^i$ at level i that contribute to the tile $\mathcal{R}^{i+1}$ at level $i+1$ ($\mathcal{R}^{i+1} = \cup_{k=1}^{m} \mathcal{R}_k^i$). In this case, each j-th bin h_j^{i+1} at level $i+1$ is computed as follows:

$$h_j^{i+1} = \sum_{k=1}^{m} h_{j,k}^{i} \qquad \forall j = 1, \ldots, n \tag{3}$$

In order to compute the similarity between two histograms, a norm must be defined in the histogram space. Several distance measures have been proposed to evaluate the similarity between two histograms (for a detailed review refer to [7, 31]). In our experiments the *Kolmogorov-Smirnov* distance was adopted. Thus, the distance between two histograms H and H' is computed as follows:

$$\mathcal{D}_{KS}(H, H') = \max_i(\check{h}_i, \check{h}'_i) \tag{4}$$

being $\check{h}_i$ and $\check{h}'_i i$-th element of the cumulated histogram of H and H', respectively (i.e., $\check{h}_i = \sum_{k=1}^{i} h_k$).

To support description of models at different resolutions, we have chosen to work with three levels of detail: at the highest level of detail the map 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, 100 reference curvature values were selected to quantize the curvature space, and histograms with 100 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.

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 100-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:

$$\mathcal{D}^j_{maps}\left(M,\hat{M}\right)=\min_p\left\{\sum_{i=1}^{n}\mathcal{D}_{KS}(H^j_i,\hat{H}^j_{p(i)})\right\} \tag{5}$$

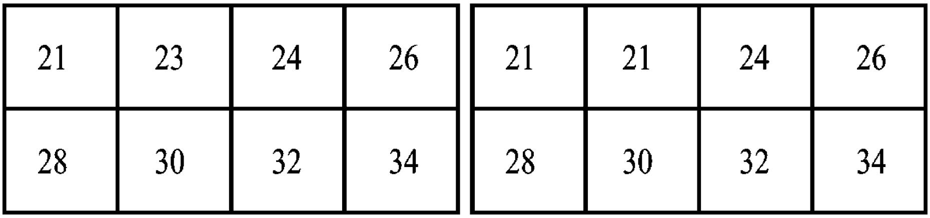
being H^j_i the histogram of the i-th tile in the first map and $\hat{H}^j_{p(i)}$ the histogram of the $p(i)$-th tile in the second map.

The solution p to Eq. 5 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.

As an example, in figure 5 the simplified models of two curvature maps are reported. Each map is partitioned into eight rectangular tiles. Each tile is associated with a histogram that, for the sake of simplicity, is assumed to be a 1-dimensional vector: a scalar number. The value of the histogram is reported inside each tile. Furthermore, within each map, tiles are ordered clockwise starting for the upper-left one. Solution of Eq. 5 requires the identification of the best permutation function $p:\{1,\ldots,8\}\rightarrow\{1,\ldots,8\}$ that minimizes the sum of distances between tiles' histograms.

Obviously, the best permutation function for this case is given by $p_{opt}(i)=i$, $\forall i\in\{1,\ldots,8\}$. In fact, with this permutation function the distance between histograms of corresponding tiles is always null apart from the association of the tile 2 in the first map with tile 2 in the second one. In this case, the distance is $|21-23|=2$.

It should be noticed that the use of a greedy search approach yields a suboptimal solution in this case. In fact, by adopting this approach, tiles in the first map are associated (clockwise, starting from the upper-left one) to the most similar tile in the second map. This



21	23	24	26
28	30	32	34

21	21	24	26
28	30	32	34

Fig. 5 Comparison of two sample maps. Each map tile is represented through a 1-dimensional histogram

yields the selection of the permutation function shown in Table 1. In this case, the distance between the two maps amounts to $|23-24|+|24-26|+|26-28|+|28-21|=12$.

4.2 Weighted walkthroughs of map regions

The use of curvature histograms for global description of map content has several advantages, including invariance to image scaling and rotation, as well as ease of combination with metric indexing structures. However, description of map content through histograms is not able to capture neither the spatial arrangement nor the local properties of individual regions of the map. In some cases this can be a limitation, since information about individual regions and their spatial arrangement in the map is strictly related to information about shape and structure of the original 3D mesh. To overcome these limitations, the coarse description of map content is complemented with a local approach capturing local properties of individual regions in the map as well as their spatial arrangement.

Local description of map content is based on weighted walkthroughs technique [13]. In particular, description of map content is accomplished by segmenting the map into regions characterized by uniform curvature values. For each region, information about region area and average curvature is retained. Furthermore, for each pair of regions, their relative position is captured through a 3×3 array corresponding to the weighted walkthroughs for the two regions, as explained in the following.

Weighted Walkthrough Definition and Notation Given two regions in the plane, there is an infinite set of paths leading from a generic point $a \in A$ to a point $b \in B$. This infinite set of paths can be reduced to a finite set of 9 *equivalence classes*. Equivalence classes abstract from the specific trajectory of the path connecting a and b and only encode the mutual displacement of a and b.

Each of these classes, called *walkthrough*, can be encoded by a pair of indices $\langle i,j \rangle$, taking values in $\{-1,0,1\}$. Index i (j) is equal to -1, 0 or 1 depending on whether the step along the horizontal (vertical) axis, has a negative, null or positive orientation, respectively (see figure 6).

In order to represent the relative position of two regions A and B each walkthrough $\langle i,j \rangle$ is associated with a weight $w_{i,j}(A,B)$ measuring the number of pairs of points of A and B that are connected through a Cartesian path equivalent to the walkthrough $\langle i,j \rangle$. This weight is evaluated as:

$$w_{i,j}(A,B) = K_{i,j}(A,B) \int_A \int_B C_i(x_b - x_a) C_j(y_b - y_a) dx_a dx_b dy_a dy_b \tag{6}$$

Table 1 Solution to Eq. 5 for maps shown in figure 5, using a greedy search approach

i	1	2	3	4	5	6	7	8
$p(i)$	1	3	4	8	5	6	7	2

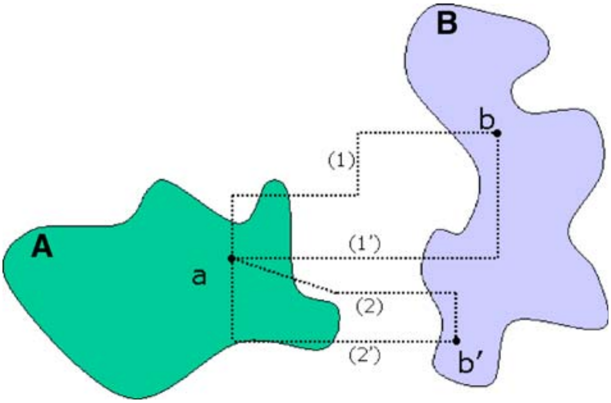


Fig. 6 Equivalence classes. Paths (1) and (1′)—connecting points a and b of regions A and B, respectively—belong to the same equivalence class (i.e., *walkthrough*) which encodes the mutual displacement between the two points; similarly, paths (2) and (2′) belong to a second walkthrough, encoding the displacement between a and b'

being:

- $C_{-1}(\cdot)$ and $C_1(\cdot)$ the characteristic function of negative and positive real numbers;
- $C_0(\cdot)$ the Dirac function;
- $K_{i,j}(A,B)$ a positive normalization factor used to guarantee that $w_{i,j} \in [0,1] \quad \forall i,j$.

The use of weighted walkthroughs enables description of map content in the form of an attributed relational graph. Graph vertices correspond to regions of the map and are labelled with the region's area and average curvature. Graph edges retain information about the relative position of regions they link and are labelled with the corresponding 3 × 3 weighted walkthroughs. It can be noticed that, due to the definition of weighted walkthrough, $w_{i,j}(A,B) = w_{-i,-j}(B,A)$. This permits to label the edge between vertices A and B only with $w_{i,j}(A,B)$, yet being able to derive from these values the values of $w_{i,j}(B,A)$.

The descriptor of content of a generic map can be represented as $\langle R,f,w \rangle$, being R the set of regions in the map, f the set of visual features capturing the appearance of each region (in our case region area and average curvature), and w the set of weighted walkthroughs capturing the relative position of each region pair.

Similarity computation Computation of the similarity between two descriptors of map local content is equivalent to an error correcting subgraph isomorphism problem [16], which is an NP-complete problem with exponential time solution algorithms [18].

In general, computation of the similarity between two graphs $\langle R,f,w \rangle$ and $\langle R',f',w' \rangle$ requires the association of the entities in the first graph with a subset of the entities in the second graph (without loss of generality it can be assumed that the number of entities on the first graph is less than or equal to the number of entities of the second graph). This association is equivalent to an injective function Γ such that

$$\Gamma : R \mapsto R'$$
$$r_1, r_2 \in R \text{ and } r_1 \neq r_2 \Rightarrow \Gamma(r_1) \neq \Gamma(r_2)$$

The similarity between two graphs under the association Γ is computed by combining *vertex* and *edge distances*. For this purpose, it is assumed that two metric functions D_f and

D_w are provided to compute the distance between two region descriptors (graph vertices) and two weighted walkthroughs sets (graph edges), respectively.

In the proposed approach, identification of the optimal association Γ_{opt} is accomplished through the technique presented in [6]. This is based on a look-ahead strategy that extends classical state-space search approaches. These latter derive the optimal solution by searching in a state-space including all possible assignments of the entities (vertices and edges) of the two graphs to be compared [42]. In order to avoid an exhaustive inspection of the space of solutions, search is done incrementally, by repeatedly incrementing the number of assigned entities in the two graphs.

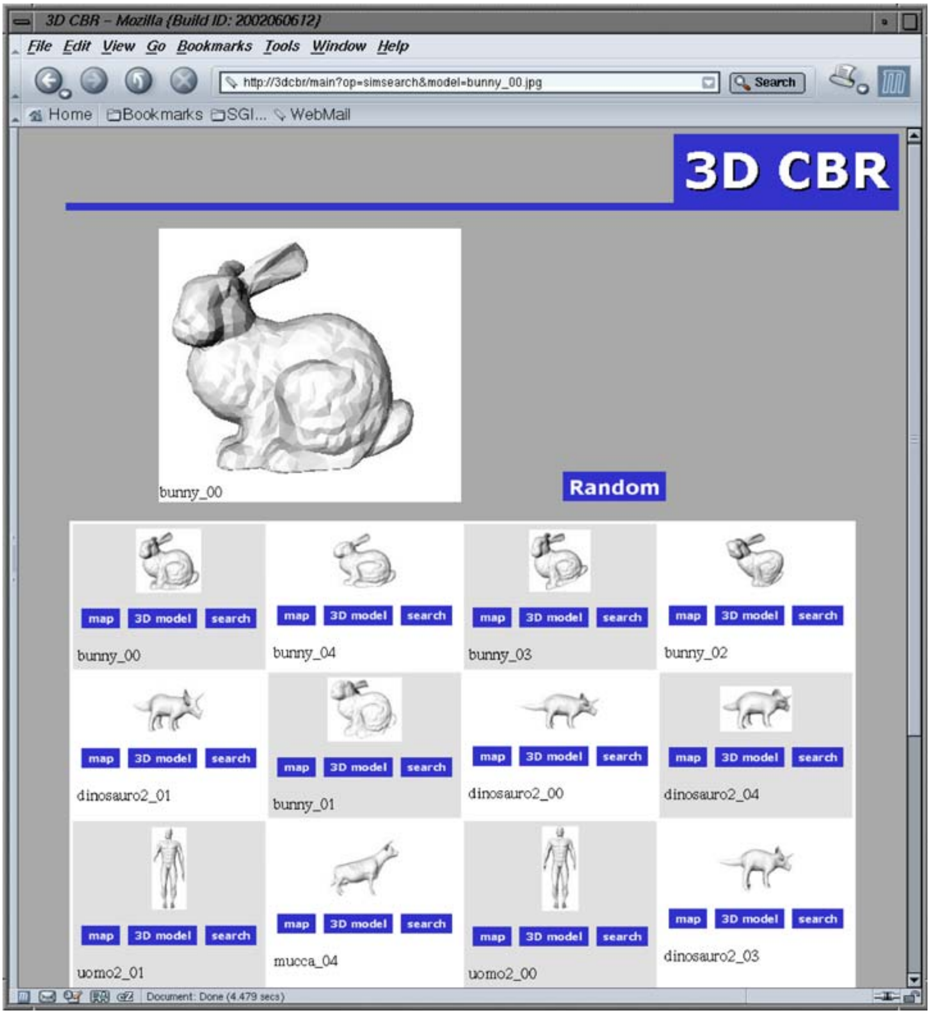

3D CBR – Mozilla {Build ID: 2002060612}
File Edit View Go Bookmarks Tools Window Help
http://3dcbr/main?op=simsearch&model=bunny_00.jpg
Search
Home Bookmarks SGI... WebMail
3D CBR
bunny_00
Random
map 3D model search
bunny_00
bunny_04
bunny_03
bunny_02
dinosauro2_01
bunny_01
dinosauro2_00
dinosauro2_04
uomo2_01
mucca_04
uomo2_00
dinosauro2_03
Document: Done (4.479 secs)

Fig. 7 The web interface of the 3D content-based retrieval system. The snapshot shows a query carried out on the histogram-based representation of tiles of the curvature map. In the *upper left corner* the query is shown. Retrieved models are *displayed below it*

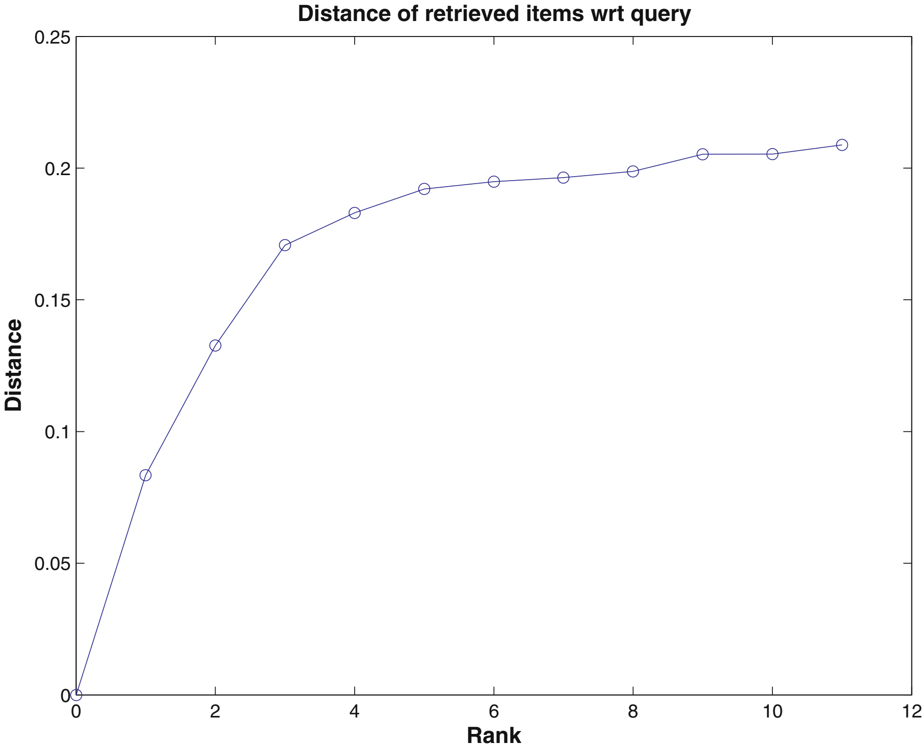


Fig. 8 Distance (with respect to the query) of items retrieved in figure 7

5 Experimental results and competitive assessment

Approximately 250 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

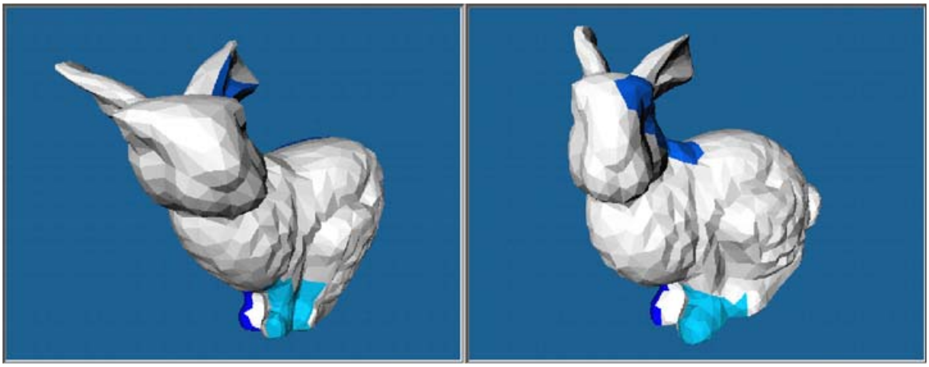


Fig. 9 The figure shows the best three correspondences of regions of the curvature maps of two bunny models, backprojected onto the original models. The correspondences were found following the heuristic search outlined in Section 4.1

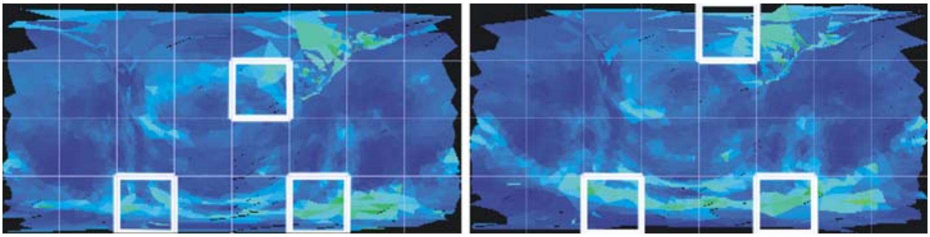


Fig. 10 Curvature maps of the two models of figure 9. The three best-matching tiles were *highlighted* on both maps

noise, which caused surface points to be moved from their original locations). Feature descriptors were then evaluated and added to the index.

Figure 7 shows the web interface of the prototype system displaying a retrieval example for the histogram-based description of map tiles, where the model of a bunny was selected as a query template (upper left). Retrieved models are shown below. The database contains

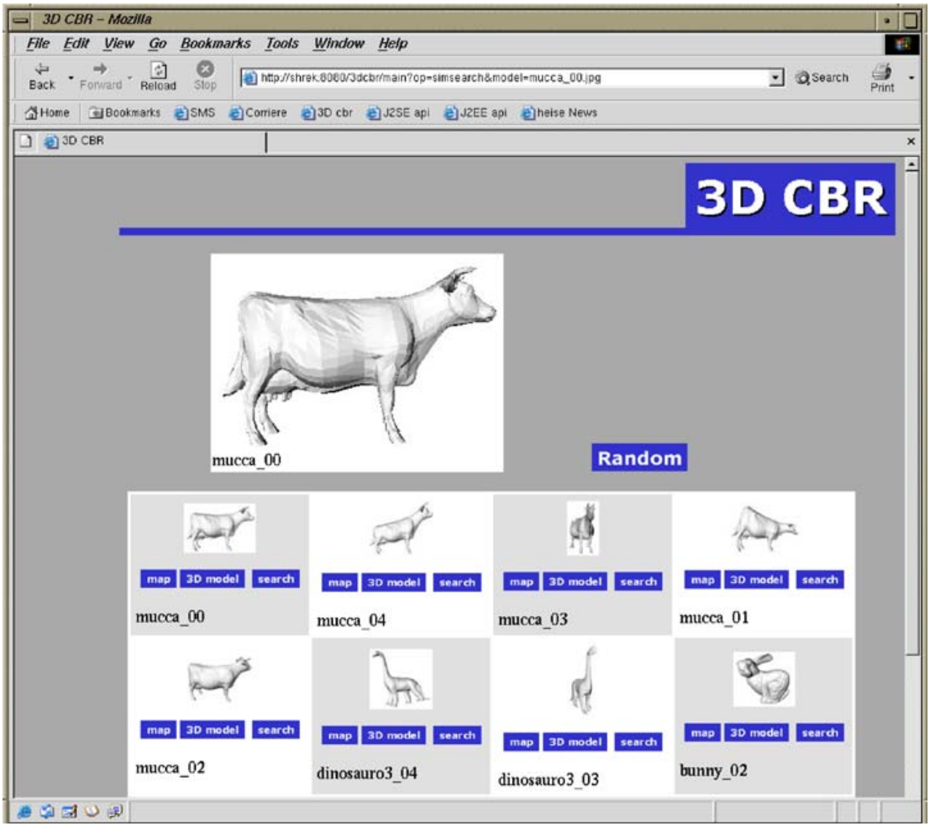


Fig. 11 Retrieval of 3D models by global similarity. In the *upper left corner* the query is shown. Retrieved models are sorted in decreasing order of similarity from *left to right* and from *top to bottom*

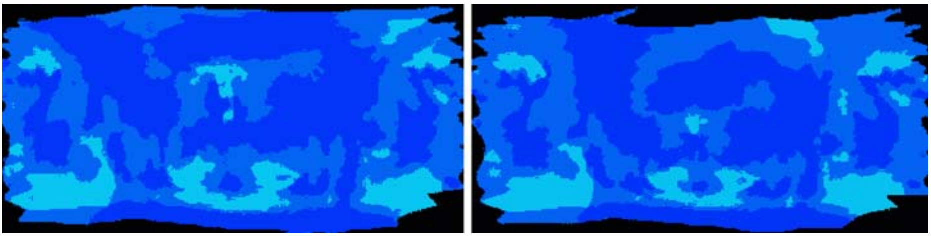


Fig. 12 Curvature maps of the first two models of figure 11

five bunny models, and all of them were retrieved. Figure 8 shows more information about this retrieval example. In particular, the distance value between the query and each retrieved item is plotted. Representation of distance values evidences that the first three retrieved items are remarkably more similar to the query than the other retrieved items. Actually, the fourth and sixth ranked models, still representing a bunny, are so much deformed that their curvature map is not much dissimilar to the map of a dinosaur model (fifth, seventh and eighth ranked models). Figure 9 provides additional insights into the matching process, showing the three best matches between regions of two different models (the two maps are reported in figure 10 for the sake of completeness).

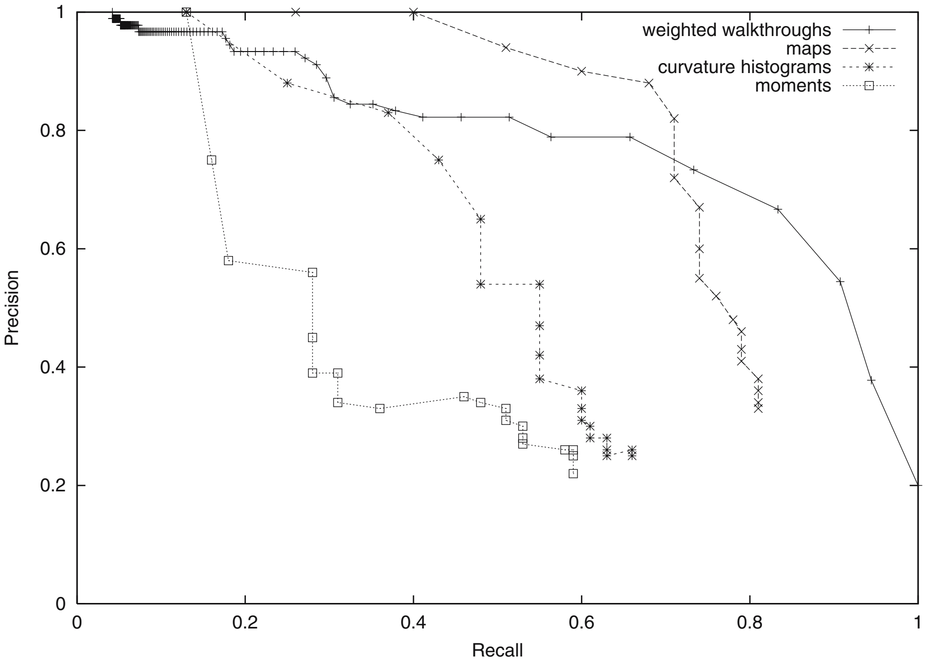


Fig. 13 Comparison of precision/recall figures for the three methods: curvature histograms, moments, and weighted walkthroughs of curvature maps. The graph shows the retrieval performance of curvature histograms, moments, and of the histogram-based description of curvature maps

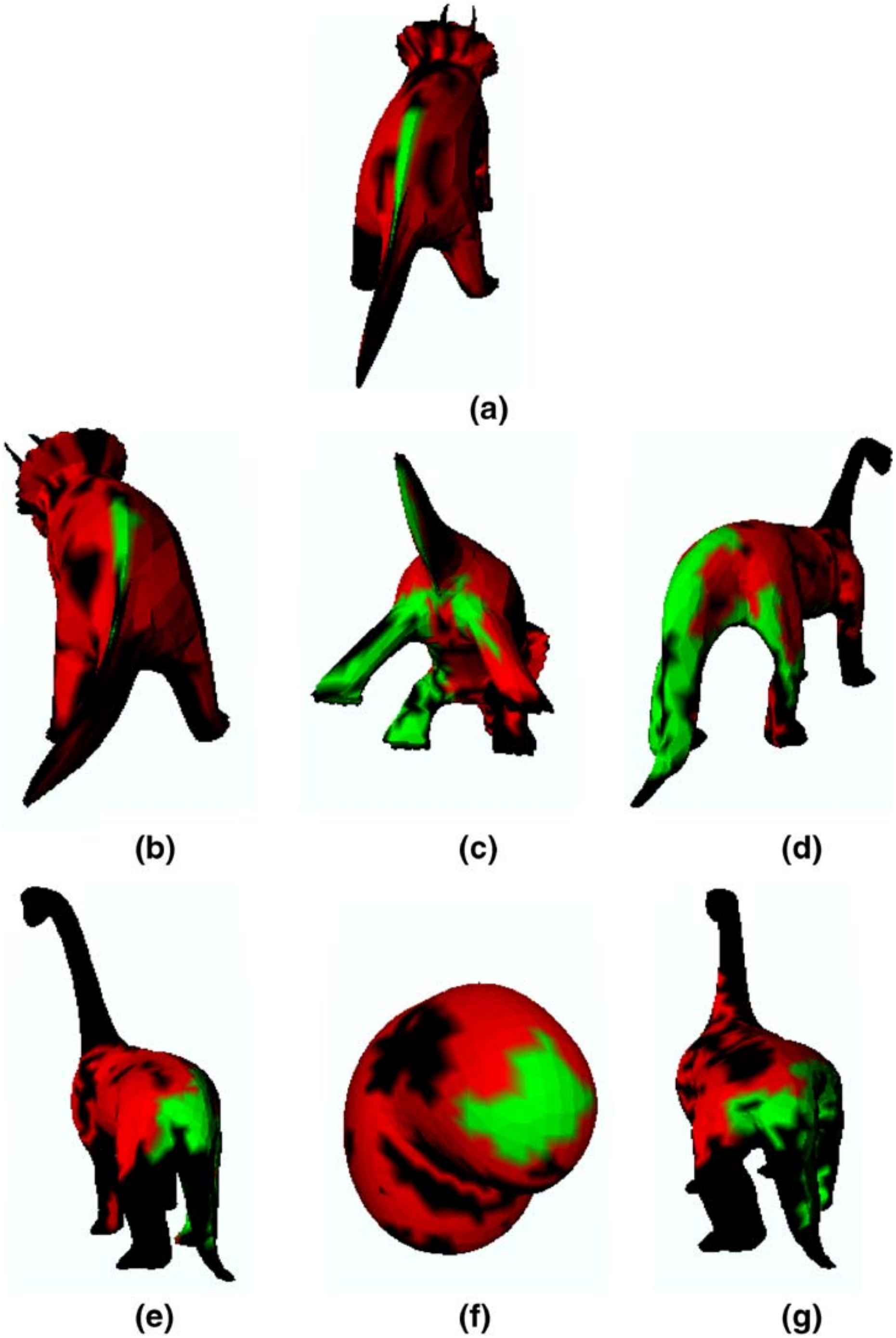


Fig. 14 Retrieval by similarity of parts. (a) The body and tail of the dinosaur are used as query. (b)–(g) retrieved models

Figure 11 shows a second retrieval example. In this case, the weighted walkthrough-based description of map regions was used. The model of a cow was selected as a query template (upper left). Retrieved models are shown next. The database contains five cow models, and all of them were retrieved.

Figure 12 provides insight into the matching process, showing the maps of the query model and the best matched retrieved model.

Comparative assessment Among the different techniques reviewed in Section 2, we selected the curvature histograms [41] and moments of surface points [14] for a comparative assessment, as enough implementation details were provided. Curvature histograms are constructed by evaluating the curvature of points of the mesh. This can be regarded as a special case of our approach: they can actually be evaluated by summing up the corresponding bins for all tiles of the curvature map. Differently, to evaluate the moments of a 3D object defined by a polygonal mesh, a limited set of points $\{P_i = (x_i, y_i, z_i)\}_{i=1}^{N}$ is considered:

$$m_{pqr} = \frac{1}{N} \sum_{i=1}^{N} w_i x_i^p y_i^q z_i^r$$

where w_i is a weight, which is usually proportional to the area of the portion of surface associated to the point P_i. To make the representation independent of the actual position of the model, the first order moments m_{100}, m_{010} and m_{001} are first evaluated, and higher order moments are then evaluated with respect to the first order moments. In our experiments, moments were computed for each model, up to the sixth order. This choice is motivated by the aim of attaining sufficient discriminatory power among different models.

Feature descriptors were extracted, to support all of the methods. Then, four sample queries were submitted to each of the three retrieval engines. Average precision vs. recall curves are shown in figure 13.

The comparative evaluation shows that our method performs well with both description techniques. This may be accounted to the fact that they allow for a local description of the models, while the other two methods only provide a global description. In fact, a global description is often inadequate also for 2D images, and 3D provides even more degrees of freedom, so that a global description may fail to discriminate among the different models. This hypothesis can be confirmed by the fact that the weighted walkthrough-based description of regions performs better that the histogram-based description of tiles, given that the former yields a better characterization of the maps in terms of their local properties.

In figure 14a–g an example of retrieval by part similarity is shown. In figure 14a two parts of the model of a dinosaur have been selected to be used as a query. The two parts (highlighted with red and green colours) roughly correspond to the body and the tail of the dinosaur. Retrieval results are shown in figure 14b–g. On each retrieved model, best matched regions are highlighted with red and green colours, so as to evidence their correspondence with the two query regions. On all retrieved models but one, best matched parts are also perceptually similar to the query parts. Only the fifth retrieved model, representing the model of a bean, is not perceptually similar to the query. Detailed inspection of the image map associated with the bean model shows that the reason why this model is retrieved is due to the close similarity of the dinosaur and bean models after the elastic deformation process. Due to this similarity, some regions on the curvature maps of the dinosaur and bean models are similar as well.

6 Conclusions and future work

In this paper, a model for representation and retrieval of 3D objects is presented. The model develops on the combined use of object deformation and planar projection to represent differential properties of the object surface through a 2D curvature map encoding the structure of the original object. Curvature map content is captured through histograms of tiles and weighted walkthroughs of regions.

Experimental results are presented to demonstrate the effectiveness of the proposed solution both for retrieval by global 3D model similarity and for retrieval by similarity of parts.

A comparative analysis is reported to demonstrate the potential and effectiveness of the proposed approach with respect to alternative solutions presented in the literature.

Future work will address investigation of the effects on retrieval effectiveness of using other features, in addition to area and curvature, to represent the structure of surface regions on the original 3D model.

Acknowledgment The authors thank Dr. Gianni Antini for his valuable contribution to part of this work.

References

1. Ankerst M, Kastenmuller G, Kriegel HP, Seidl T (1999, July) 3D space histograms for similarity search and classification in spatial databases. In Proc. 6th Int. Symposium on Spatial Databases (SSD'99). Hong Kong, China
2. Assfalg J, Bertini M, Colombo C, Del Bimbo A (2002) Semantic annotation of sports videos. IEEE Multimed 9(2):52–60
3. Balakrishnan R, Fitzmaurice GW, Kurtenbach G (2001, March) User interfaces for volumetric displays. IEEE Comput 34(3):37–45
4. Belyaev AG, Bogaevski IA, Ohtake Y (2000, April 10–12) Polyhedral surface smoothing with simultaneous mesh regularization. In: Proceedings of Geometric Modeling and Processing 2000. Theory and applications. Hong Kong, China, pp 229–237
5. Berchtold S, Kriegel HP (1997) S3: Similarity search in CAD database systems. Proceedings of the ACM SIGMOD International Conference on Management of Data, May, Tucson, Arizona, pp 564–567
6. Berretti S, Del Bimbo A, Vicario E (2001, October) Efficient matching and indexing of graph models in content-based retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(10):1089–1105
7. Brunelli, Mich (2001) Histogram analysis for image retrieval. Pattern Recognit 34(8):1625–1637
8. Chalmers A, Rushmeier H (eds) (2002, September/October) Special issue on Computer graphics in art history and archaeology. IEEE Comput Graph Appl 22(5)
9. Colombo C, Del Bimbo A, Pernici F (2002, June) Uncalibrated 3D metric reconstruction and flattened texture acquisition from a single view of a surface of revolution. In: Proceedings of 1st international symposium on 3D data processing visualization and transmission 3DPVT 2002, Padova, Italy, pp 277–284, IEEE
10. Courtney JD (1997) Automatic video indexing via object motion analysis. Pattern Recognit 30(4):607–625
11. Dagtals S, Al-Khatib W, Ghafoor A, Kashyap RL (2000) Models for motion-based video indexing and retrieval. IEEE Trans Image Processing 9(1):88–101
12. Del Bimbo A (1999) Visual information retrieval. Academic, New York
13. Del Bimbo A, Vicario E (1998, June) Using weighted spatial relationships in retrieval by visual contents. In: Proceedings of the IEEE workshop on content-based access of image and video libraries (CBAIVL'98, Santa Barbara, California, pp 35–39
14. Elad M, Tal A, Ar S (2001, September) Content based retrieval of VRML objects—An iterative and interactive approach. EG Multimed 97–108
15. Elvins TT, Jain R (1995, December) Web-based volumetric data retrieval. In: Proceedings of VRML 95, the first annual conference on the virtual reality modeling language

16. Eshera MA, Fu K-S (1984, May/June) A graph measure for image analysis. IEEE Trans Systems, Man and Cybernetics 14(3):398–407
17. Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D, Janker P (1995, September) Query by image and video content: the QBIC System. IEEE Comput 28(9):310–315
18. Garey MR, Johnson D (1979) Computer and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco
19. Garland M (1999, September) Multiresolution modeling: survey & future opportunities. In: Proceedings of Eurographics'99
20. Huang J, Ravi Kumar S, Mitra M, Zhu W-J, Zabih R (1997, June 17–19) Image indexing using color correlograms. In: Proceedings International Conference on Computer Vision and Pattern Recognition (CVPR'97), Puerto Rico
21. Jain AK, Vailaya A (1996, August) Image retrieval using color and shape. Pattern Recognit 29(8):1233–1244
22. Kriegel HP, Seidl T (1998) Approximation-based similarity search for 3D surface segments. GeoInformatica Journal 2(2):113–147, Kluwer, Norwell, Massachusetts
23. Kolonias I, Tzovaras D, Malassiotis S, Strintzis MG (2001, September 19–21) Content-based similarity search of VRML models using shape descriptors. In: Proceedings of International Workshop on Content-Based Multimedia Indexing, Brescia (I)
24. Lamdan Y, Wolfson HJ (1988) Geometric hashing: A general and efficient model-based recognition scheme. Proceedings IEEE International Conference on Computer Vision, Tampa, Florida, pp 238–249
25. Levoy M, Pulli K, Curless B, Rusinkiewicz S, Koller D, Pereira L, Ginzton M, Anderson S, Davis J, Ginsberg J, Shade J, Fulk D (2000, July) The digital Michelangelo project. In: Proceedings of the Conference on Computer Graphics (SIGGRAPH 2000)
26. Mokhtarian F, Khalili N, Yeun P (2001) Multi-scale free-form 3D object recognition using 3D models. Image Vis Comput 19(5):271–281
27. Page DL, Koschan A, Sun Y, Paik J, Abidi MA (2001, December 11–13) Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting. In Proceedings of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'01), Kauai, Hawaii 1:162–167
28. Paquet E, Rioux M (1999) Nefertiti: a query by content system for three-dimensional model and image database management. Image Vis Comput 17(2):157–166
29. Osada R, Funkhouser T, Chazelle B, Dobkin D (1999) Matching 3D models with shape distribution. In: Proceedings of shape modeling international, Genova (I)
30. Roth V (1999) Content-based retrieval from digital video. Image Vis Comput 17(7):531–540
31. Rubner Y, Tomasi C, Guibas LJ (1998) A metric for distributions with applications to image databases. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV'98), Bombay, India
32. Santini S, Jain R (1999) Similarity measures. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(9):871–883
33. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000, December) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Analysis and Machine Intelligence 22(12):1349–1380
34. Smith, JR, Chang SF (1996, November) VisualSEEk: a fully automated content-based image query system. In: Proceedings of ACM Multimedia '96. Boston, Massachusetts
35. Snyder JP, Bugayevski LM (1995) Map projections—a reference manual. Taylor & Francis
36. Swain MJ, Ballard DH (1991) Color indexing. Int J Comput Vis 7(1):11–32
37. Taubin G, Cooper DB (1991) Recognition and positioning of rigid objects using algebraic moment invariants. Geometric Methods in Computer Vision 1570:175–186, SPIE
38. Taubin G (1995) A signal processing approach to fair surface design. Comput Graph (Annual Conference Series) 29:351–358
39. Taubin G (1995) Estimating the tensor of curvature of a surface from a polyhedral approximation. Fifth International Conference on Computer Vision (ICCV'95)
40. Taubin G (2000) Geometric signal processing on polygonal meshes. In: Proceedings of Eurographics 2000
41. Vandeborre J-Ph, Couillet V, Daoudi M (2002) A practical approach for 3D model indexing by combining local and global invariants. In: Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'02)
42. Tsai W-H, Fu K-S (1979, November/December) Error-correcting isomorphism of attributed relational graphs for pattern analysis. IEEE Trans Systems, Man and Cybernetics 12(6):657–767
43. Wong K-YK, Mendonca PRS, Cipolla R (2003, February) Camera calibration from surfaces of revolution. IEEE Trans PAMI 25(2):147–161
44. Zhang HJ, Wu JH, Zhong D, Smoliar SW (1997) An integrated system for content-based video retrieval and browsing. Pattern Recognit 30(4):643–658

Jurgen Assfalg received the Laurea degree in electronics engineering from the Universityà degli Studi di Firenze, Italy, in 1998. In 2002 he received a Ph.D. in information and telecommunications engineering from the same University. He was then a research associate at the Visual Information Processing Lab until the end of 2003. He is now with the local government of the Provincia di Firenze. His research interests include multimedia retrieval systems, advanced user interfaces, and 3D graphics. He currently is a member of the ACM.

Alberto Del Bimbo is Full Professor of Computer Engineering and the Director of the Master in Multimedia of the University of Florence, Italy. At the present time, he is also Deputy Rector of the University of Florence, in charge of Research and Innovation Transfer. His scientific interests are Pattern Recognition, Image Databases, Human Computer Interaction and Multimedia applications. He has published over 170 publications in some of the most distinguished scientific journals and international conferences, and is the author of the “Visual Information Retrieval” monography on content-based retrieval from image and video databases. Prof. Del Bimbo is Member of IEEE (Institute of Electrical and Electronic Engineers) and Fellow of IAPR (International Association for Pattern Recognition). He was the President of the IAPR Italian Chapter, from 1996 to 2000 and Member at Large of the IEEE Publication Board from 1998 to 2000. He has been Associate Editor of Pattern Recognition, Journal of Visual Languages and Computing, Multimedia Tools and Applications, Pattern Analysis and Applications, IEEE Transactions on Multimedia, and IEEE Transactions on Pattern Analysis and Machine Intelligence. He was the Guest Editor of several special issues on Image databases and image analysis.

Pietro Pala graduated cum laude in Electronics Engineering at the University of Firenze, Italy in 1994. From the same University, he received the Ph.D. in Information and Telecommunications Engineering in 1997. Presently, he is Associate Professor of Computer Engineering at the University of Firenze where he teaches “Database Management Systems” and “Image and Video Processing and Analysis.” He also teaches “Multimedia Systems” at the degrees in “Internet Engineering” and “Content Design” of Master in Multimedia of the University of Firenze. Pietro Pala is vice-chair of the IAPR Technical Committee 12 on Multimedia and Visual Information Systems. His main scientific and research interests include Pattern Recognition, Image and Video Databases and Multimedia Information Retrieval.