Eurographics Workshop on 3D Object Retrieval (2008)
I. Pratikakis and T. Theoharis (Editors)

Part Analogies in Sets of Objects

S. Shalom[1] and L. Shapira[2] and A. Shamir[3] and D. Cohen-Or[2]

[1]Bar-Ilan University, Israel
[2]Tel Aviv University, Israel
[3]Interdisciplinary Center, Herzliya, Israel

Abstract

Shape retrieval can benefit from analogies among similar shapes and parts of different objects. By partitioning an object to meaningful parts and finding analogous parts in other objects, sub-parts and partial match queries can be utilized. First by searching for similar parts in the context of their shape, and second by finding similarities even among objects that differ in their general shape and topology. Moreover, analogies can create the basis for semantic text-based searches: for instance, in this paper we demonstrate a simple annotation tool that carries tags of object parts from one model to many others using analogies. We partition 3D objects based on the shape-diameter function (SDF), and use it to find corresponding parts in other objects. We present results on finding analogies among numerous objects from shape repositories, and demonstrate sub-part queries using an implementation of a simple search and retrieval application.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling curve, surface, solid and object representations

1. Introduction

Unlike documents search queries, it is still almost impossible to use text for 3D object searches. The main reason for this is that most digital 3D models are not annotated. Usually, searches for 3D objects are based on finding similar objects to a given query model. Moreover, searching for a specific part inside a 3D model is even more challenging since in most cases the models are not segmented and the different parts composing an object are not linked to textual tags.

In this paper we present a method that finds analogies among parts of digital 3D models by segmenting them and creating a signature for each part. We use such analogies to provide enhanced model-based queries of sub-parts, as well as a tool that can annotate a database of parts to support textual searches. Using analogous parts, model-based queries can support part-in-whole queries and partial matches. Using analogies many models can be annotated by carrying part-tags from one model to many others automatically. This can provide the basis for text-based searches of 3D models and model parts.

The dominating representation of digital 3D objects is a 2D surface mesh embedded in 3D, defining its boundary. Nevertheless, many of the analogies that exist among 3D shapes are *volumetric* in nature, and it is important to base the segmentation and analogies on volumetric attributes. Therefore, in this paper we use the shape-diameter function (SDF) both for segmentation and for part signature definition. The SDF provides a link between the object's volume and the mesh surface, mapping volumetric information onto the surface boundary mesh. It is defined by examining the diameter of the model in the neighborhood of each point on its boundary surface. This measure relates to the medial axis transform (MAT) [CCM97] which is extremely informative for shape analysis and partitioning. The SDF replaces the local shape-radius of the MAT by a measure of the local *shape-diameter*.

To find analogies we first partition all objects in a database into parts. Next, we define a signature for each part based on geometric attributes and its relation to the whole object. When a user specifies a model part query , we can retrieve the most similar parts from all models in the database based on the distance between the parts' signatures (Figure 1). Moreover, by annotating the parts of one model man-

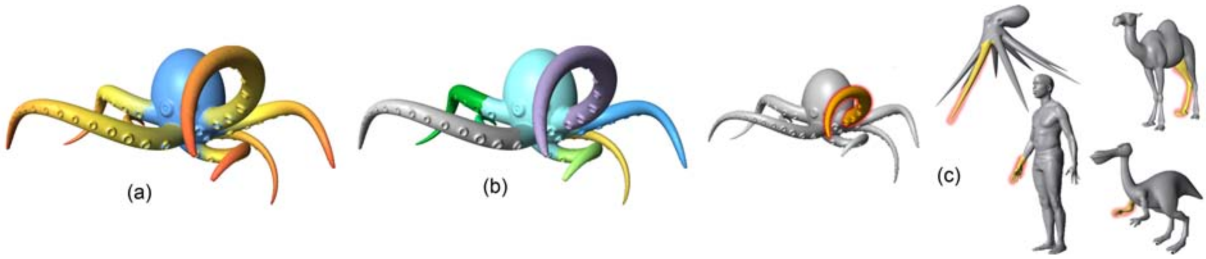


Figure 1: *Part analogies definition: The SDF is calculated on all objects in the database (colored from red (narrow diameter) to blue (wide diameter) in (a)), linking the mesh surface of the objects to their volumes. All objects are then partitioned (b) and the signatures of their parts define the analogies between similar parts of different meshes (c). Analogous parts to the octopus arm are found in different models even if they are dissimilar in shape.*

ually we carry the tags to all similar parts in the database. Later, these tags are used for text based retrieval from the database.

2. Related Work

There are numerous mesh partitioning techniques based on various mesh attributes. For a survey on different mesh partitioning technique we refer the reader to [Sha07]. Since we seek part-type partitioning, we employ the method in [SSCO07], which uses the SDF to partition sets of objects consistently. Partitioning based on the SDF is likely to create parts which are similar in their SDF signature and are correspondent among different objects (Figure 2). Nevertheless, other part-type partitioning methods such as [KT03, AFS06] could also be used in the first stage of our algorithm.

Shape matching is also an active field of research, and numerous shape signatures based on geometry and topology have been proposed [TV04, BKS*05, GSCO07]. In [GSCO07] the SDF was used alongside a centricity measure to define a global signature. We concentrate on matching sub-parts of objects which is related more to partial matching. Nevertheless, partial matching techniques [FS06, GCO06, NDK05, BMP01, JH99] are based more on feature correspondence and less on segmentation. Segmentation and shape analysis was also used to enrich models in semantic information in [ABM*06, ARSF07] by manually connecting parts through a user interface to an instance in a knowledge base, or by using ontology connecting form and functionality in [CGM07].

Finding correspondence and analogies between different shapes is recently becoming an active field of research. Many works focus on the need to define measures for similarity of shapes [SSGD03, CDS*05, FKMS05], alignment of shapes [GMGP05] and finding complete correspondence between two models for the purposes of deformations, morphing [Ale02], and cross-parameterizations [KS04a]. We focus on finding analogies between sub-parts.

Low level analogies to match vertices is used in [SAPH04, KS04b] for cross parametrization between two models. The matching is found using user supplied matching points. In [SP04] such matching points are used to define a many-to-many mapping of vertices between the models which is used to transfer deformations between the models. These works require a long time to process and non-trivial user input, they operate mostly on two models, and cannot work on partial models. Our work targets higher level analogies and can work also on objects with different topology and structure (Figure 1).

3. Partitioning to Parts

When examining 3D models, one can observe that the similarity of parts often stems from their functionality. For example in humans and animals parts are associated with organs, which are usually 3D *volumetric* sub-parts of the shape. Therefore, an automatic algorithm aimed at detecting such 3D shape analogies must first identify these sub-parts. We use a partitioning of the shapes guided by the *shape diameter function* (SDF) [SSCO07]. The SDF connects volumetric information of the shape onto the boundary mesh by measuring the local diameter of the object at points on its boundary. Hence, the SDF is suitable to guide volumetric part extraction, detect natural 3D shape partitioning, and to define the parts signatures (Figure 2).

The SDF is defined as the diameter of the object in the neighborhood of each point on it's surface. Given a point on the surface mesh a set of rays is sent inside a cone centered around its inward-normal direction (the opposite direction of its normal) to the other side of the mesh. The value of the SDF at the point is defined as the weighted average of all the lengths of the rays that fall within one standard deviation from the median of all lengths. The weights used are the inverse of the angle between the ray to the center of the cone. This is because rays with larger angles are more frequent, and therefore have smaller weights.

To maintain compatibility over different meshes, which may have different scales and resolutions, we normalize and smooth the values. We also perform the partitioning in logspace to enhance the importance of delicate parts, which tend to have low characteristic SDF values. The normalized SDF value $nsdf$ of facet f is defined:

$$nsdf(f) = \log\left(\frac{sdf(f) - \min(sdf)}{\max(sdf) - \min(sdf)} * \alpha + 1\right) / \log(\alpha + 1)$$

Where $sdf : F \rightarrow \mathbb{R}$ is the SDF value for each facet f. α is a normalizing parameter which is set to 4 in all our examples.

The SDF can be seen as a scalar function over the mesh domain. Specific iso-values of the SDF create iso-contours on the surface, which can be used to partition the mesh. The partitioning algorithm consists of two steps. In the first step we use a Gaussian Mixture Model (GMM) fitting k Gaussians to the histogram of SDF values of the faces. This is achieved using the Expectation-Maximization (EM) algorithm. The result of this clustering process for each face is a vector of length k signifying its probability to be assigned to one of the SDF clusters. Note that each SDF cluster may contain multiple mesh parts such as legs or fingers. Using more Gaussians will create more levels in the hierarchy of sub-parts.

In the second step we would like to smooth the boundaries between parts and adhere to local mesh features such as concave areas or creases. We employ an alpha expansion graph-cut algorithm [BVZ01, ZK04] to solve the k-way graph-cut problem which is known to be NP-hard. The alpha expansion algorithm utilizes a series of large moves, changing a large number of pixel labels at a time, to arrive at an approximate solution to the problem. The algorithm finds a labeling within a known factor of the optimal solution.

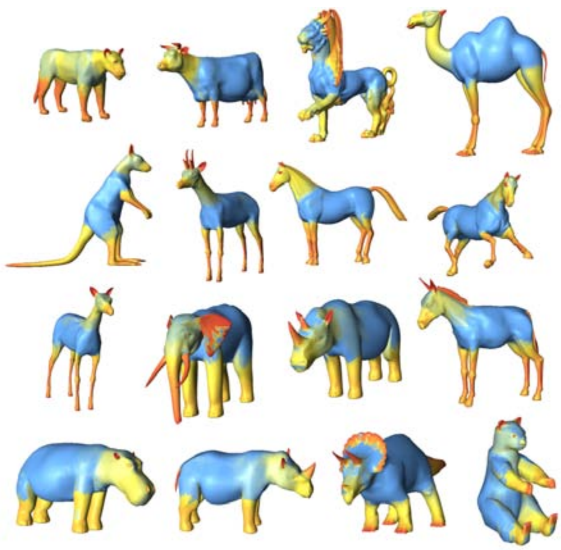


Figure 2: *Color mapping the SDF values on different models indicates similarity between their parts.*

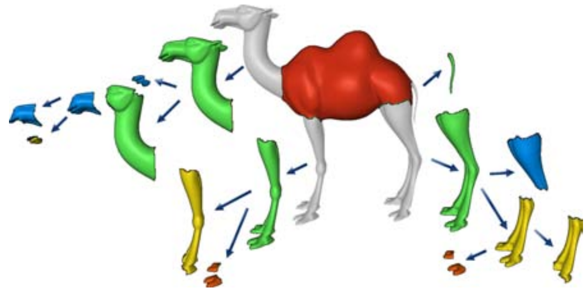
We wish to take into account both the probability vector from the EM step, and the quality of the boundaries. Therefore, the graph-cut optimization minimizes the following energy functional, which is built from e_1 the *data term*, and e_2, the *smoothness term*:

$$\begin{aligned} E(\bar{x}) &= \sum_{f \in F} e_1(f, x_f) + \lambda \sum_{\{f,g\} \in N} e_2(x_f, x_g) \\ e_1(f, x_p) &= -\log(P(f|x_f) + \varepsilon) \\ e_2(x_f, x_g) &= \begin{cases} l(f,g)(1 - \log(\theta(f,g)/\pi)) & x_f \neq x_g \\ 0 & x_f = x_g \end{cases} \end{aligned}$$

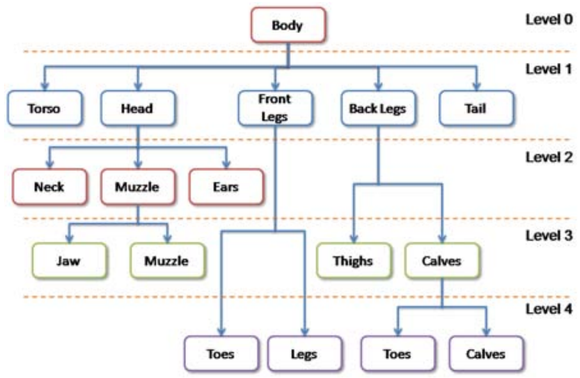
Where x_f is the cluster assigned to face f. $P(f|x_p)$ represents the probability of assigning face f to cluster p. These value are derived from the GMM fitted in the first step of the algorithm. $\theta(f,g)$ is the dihedral angle between facets f and g. N is a set of adjacent face pairs in the mesh, λ is a parameter defining the degree of smoothness. We've found that using $\lambda = 0.3$ gave good results for all models in the database, since we normalize smoothness by $l(f,g)$ - the length of the edge between f and g. Here and in subsequent equations, $\varepsilon = 10^{-3}$ is used to avoid numerical instability. The result of the graph-cut algorithm is a smooth partitioning of the mesh, clearly separating distinct parts (see examples in Figure 8).

Many times analogies between parts are based on the relation of the part to the whole object. Parts of different object that vary in their geometric shape or attributes individually, become analog when placed in the context of their whole shape. Therefore, we want to create a hierarchical representation of each shape's parts, and employ it later to find analogies. We sort the means of the GMM model from large to small, and define k iso-values that separate the Gaussians and separates the mesh into "levels". To create the hierarchal partitioning of the object we use a subset of these values, starting from the largest and add them in descending order, each one defining another level in the hierarchy. For example, the first iso-value on a human model would separate the arms, legs and head from the torso. Adding the second value would separate the hands from the arms, and the feet from the legs. Adding the third value would separate the fingers from the hands and feet.

We start with a root node representing the whole model. In each level we create child nodes from parts which are in a lower level of the partitioning and are also sub-parts of the parent node part (their sub-meshes are contained in its sub-meshes). For example, the five tier camel partitioning hierarchy (3(a)) induces a hierarchical part graph as can be seen in Figure 3(b). Additional hierarchical examples can



(a)



(b)

Figure 3: *(a) The camel model is partitioned using four iso-values, resulting in a five-tier hierarchy of partitioning (b) The partitioning induces a hierarchical graph of parts.*

be seen in Figure 4. This creates a tree structure defining the relation of parts inside the object, and assists to define a better distance metric to recognize similar object parts as described in the next Section.

4. Parts Signature & Distance

To find analogies between multiple models, we must define a way to measure similarity between parts of models. We contend that when seeking to compare two parts, the context from which they came is crucial to the comparison. A finger on a human model is just a capped cylinder. However, when taken in context of the hand, the arm and the entire body, its description is more complete, and better matches and analogies could be found.

Each segmented part in the model is assigned a local signature composed of its geometrical attributes. The geometrical attributes consist of the normalized histogram of SDF values within the part and the size of the part as a percentage of the whole model. The histogram represents the distribution of SDF values and provides a unique descriptor as to the shape of the specific part (Figure 5). We construct a robust histogram based on the original SDF measurements in the part, removing the top and bottom 5% to remove outliers. The part's relative size helps measure the scale of the part.

In many cases, the geometrical attributes are sufficient to define a good distance metric between the shape of parts (Figure 9), especially in distinct parts, such as a hand. However, analogies stem from the characteristic of the part in the whole as well as it's geometric attributes. Therefore, we also define shape context attributes for each part that are derived from the hierarchical partitioning of the whole model.

Using the hierarchy we define the context of a part as the path between the node representing the part, and the root of the tree. Each node along this path represents a part for which we can calculate the geometrical attributes as described above. The set of all of these geometric attributes define the context descriptor of the part. We use this context descriptors in a distance measure between two parts, that takes into account both the similarity of the parts themselves, and the similarity among the path nodes.

First, we define a local distance metric d between two parts p and p' and then we extend the definition to a distance measure D that includes also the shape context. $d(p,p')$ is defined as a weighted sum of the distance between the local part histograms, and the relative part sizes.

$$d_{histogram}(p,p') = \left\| {}^{H(p)}\!/_{\|H(p)\|} - {}^{H(p')}\!/_{\|H(p')\|} \right\|_2$$

$$d_{size}(p,p') = \frac{|size(p) - size(p')|}{size(p) + size(p')}$$

$$d(p,p') = \frac{1}{3} \cdot d_{size}(p,p') + \frac{2}{3} \cdot d_{histogram}(p,p')$$

H is a normalized robust histogram of the part's SDF values. We use the *L2* distance metric on the normalized histogram, treating it as a vector. We've experimented with various distance measures such as *Chi-Squared* and *Kullback-*

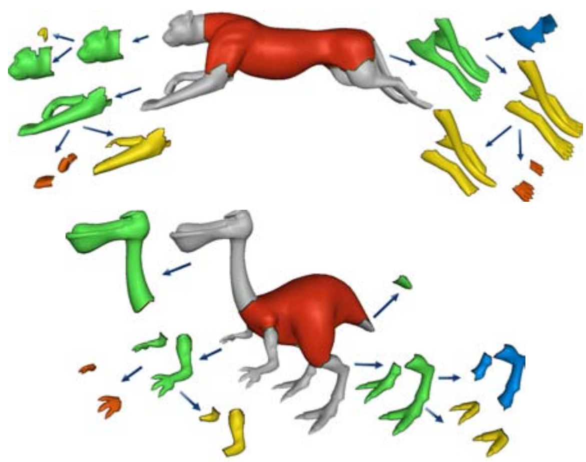


Figure 4: *Hierarchical partitioning of cheetah and dinopet model*

Leibler, but found that they provide little improvement over *L*2. $size(p)$ is the relative size of part p within its model.

The geometric distance $d(p,p')$ between parts p and p' is a symmetrized non-negative distance measure between parts. To define our shape context distance measure $D(p,p')$ we consider not only the $d(p,p')$ between the two parts, but the whole *paths* from the nodes of p and p' to the root of their partitioning hierarchy. Given two such paths on which we want to measure similarity, we build a bipartite graph G (Figure 6) such that each side represents all nodes in each path. The edges between the two sides contain one edge between p and p' (the two parts whose distance we want to measure), and an edge between each ancestor of p to each ancestor of p'. Note that the number of ancestors of p and p' may be different. The capacity of an edge between two nodes q and r is defined as

$$capacity(q,r) = \frac{1}{d(q,r)+\varepsilon} - 1$$

Lastly, we add two nodes source S and T sink and connect each one of them to the nodes in one side of the graph respectively, with capacity equal to $\alpha \cdot capacity(p,p')$, with α usually set to 1.5. This serves as an upper limit on the capacity of the flow in the graph G. We now define

$$D(p,p') = \frac{1}{flow(G)+1}$$

Where $flow(G)$ is the maximum flow in graph G.

The key motivation behind such a measure is on one side to match the part in context of the whole hierarchy, and on the other to achieve robustness against differences in partitioning. The measure will be higher as more parts in the path from the node to its root match the respective nodes in the compared part. However, it is hard to determine the exact matching of parts in two hierarchies, as we have no assurances that similar models will have similar partitioning. By connecting each ancestor of p to each ancestor of p' we are assured that the flow will represent the maximum similarity from possible different matchings.

For instance, given three geometrically similar parts p,p',p'' such that p and p' also come from similar models, $d(p,p')$ and $d(p,p'')$ will be similar and the flow on the edges (p,p') and (p,p'') will be similar. However, while comparing p and p's ancestors, more flow will be "added" to the graph, enhancing their similarity and arriving at $D(p,p') < D(p,p'')$. For example, figure 7 illustrates the distance measure against a variety of parts, some which are similar and some different from the query part.

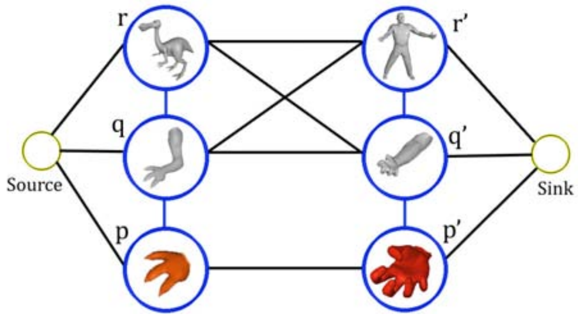


Figure 6: *In order to measure similarity between two parts, we build a bipartite graph. The first part hierarchy is represented by the nodes p,q,r while the second part hierarchy is represented by the nodes p',q',r'. The capacity of the edge (x,y) is defined to be $\frac{1}{d(x,y)+\varepsilon}$. The similarity between two parts is defined as the maximum flow through this graph.*

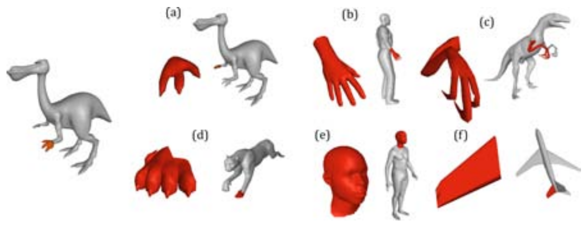


Figure 7: *We measure the distance from hand of the dinopet to six other parts. Parts (a) through (d) are similar in spite of their large geometric variability, while parts (e) and (f) are not. The distance measurements are listed in table 1.*

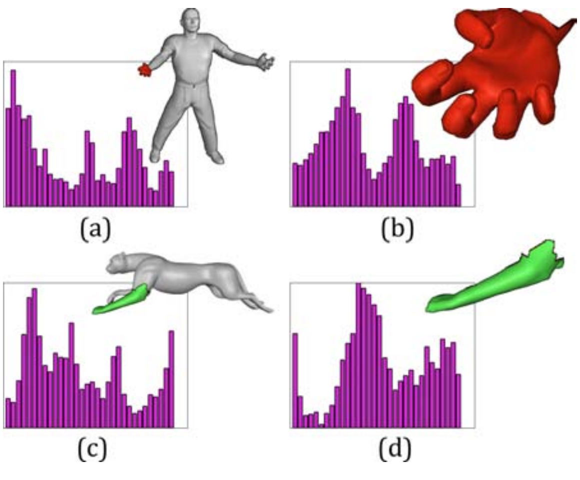


Figure 5: *Visualization of the robust and normalized SDF histogram used in a part's signature. We visualize the histograms for several models. (a) Whole man (b) Man's hand (c) Whole cheetah (d) Cheeta's front leg*

Part	$d(p,p')$	$D(p,p')$
a (dinopet other hand)	0.001	0.0033
b (human hand)	0.034	0.0147
c (dinosaur hand)	0.042	0.0263
d (cheetah paw)	0.07	0.0242
e (human head)	0.26	0.126
f (airplane wing)	0.373	0.192

Table 1: *Example of the two distance measures (local and contextual) between the dinopet's hand and six other parts (Figure 7).*

5. Applications and Results

We will demonstrate the usefulness of our analogies approach using two applications. The first is in the context of a search and retrieval application of 3D shapes. Using analogies, one can search for parts of shapes in the database that are similar to a give part, or for objects that contain similar parts to a given part query. The second, can be seen as a tool to enhance the meta-data in 3D objects. Once part analogies are found, any information linked with the query part can be carried automatically to other analogous parts, enriching the database with meta-data.

5.1. 3D Model Parts Retrieval

Using the distance measure defined in section 4 we developed a simple part retrieval application. The user loads a model, which is automatically partitioned. The user can select a part p and search for similar parts in the database. The database models are segmented to parts, each retaining its partitioning hierarchy, and pre-calculated geometric attributes. We scan the database, and for each part p', calculate the shape context distance measure $D(p,p')$. We sort the results and return the top matching parts. Several example queries can be seen in figure 10.

We also allow to search for analogies in a set of models (Figure 11). Given a source model, and k target models, we attempt to find an "as complete as possible" correspondence between the source and each of the target models. This is done in a greedy algorithm, which queries successively each part in the source model, over the subset of target models. The best match is selected, and the matching continues on the parts adjacent to it.

Our database includes 102 models that include interesting parts. The models are varied, consisting of animals, humans, cars, tables, chairs and planes. We did not use simple models such as cubes, cylinders, pipes etc., that cannot be partitioned to parts, and did not use non-volumetric models such as many plants models. The average number of faces is 10517 (standard deviation 8042). We partition each model using 5 iso-values, which results in up to five levels of partitioning. In total this produced 2200 parts, where the average number of parts in a model is 21 (including parts from all levels of the hierarchy).

In order to test the quality of our results, we performed a nearest neighbors test [SMKF04], querying each part in the database for its nearest neighbor (besides itself). We rated each query as success/failure based on the context of the shape (head, leg, surface etc.), as our database was not divided into partial classes. The percentage of correct queries was 79%. If we take into account defunct parts (which occur in non-manifold and low quality models) this result is improved to 89%.

All statistics were gathered on a 1.8*ghx* single processor Windows XP machine. Building the database consists of calculating SDF values for all the models, which took 10 minutes, and performing automatic partitioning which took 7 minutes. A query takes on average 600*ms* to go over all parts in the database, and return the relevant results.

5.2. Parts Annotation

Using the *shape context distance measure*, we can now transfer user supplied annotations from one part to others in our database automatically. We developed a simple interface, in which a user may select a part (of any level in the hierarchy of the model) and annotate it with one or more textual tags. The tag is then associated with the part, and kept in the database.

Given a part p which we wish to automatically annotate, we define it as a query and search the database, getting a set R of results. We discard all but the first 20 results from R and build a set of tags T containing all tags attached to parts in R. For each tag $t \in T$ we define a tag importance measure:

$$C(t) = \sum_{r \in R_t} \frac{1}{D(p,r)-1}$$

Where $R_t = \{r \in R | t \in r\}$, D is the *shape context distance measure* defined in section 4.

We associate part p with all tags t such that $C(t) > k$. We employed $k = 100$ as a fixed constant. The tags are attached to the parts and saved in the database.

We allow the user to perform an annotation transfer on all tags found in the database, or only on selected tags. Consequently we can perform text searches in the database, searching for specific tags, such as "ear", "head", "thin", "wide" etc. (Figure 12).

6. Conclusions

We have presented a method that automatically finds analogies among sets of objects. The method first partitions the objects to create a parts hierarchy, and then defines a signature for each part. This signatures draws not only from the properties of the part itself, but from the relations between the part and the whole object. These signatures are used to define an effective distance measure that can find analogous parts among sets of objects.

We have shown that such analogies can support part search queries in a shape retrieval application. We have also used them to add semantic information to the objects by carrying information defined on one part (e.g. tags) to analogous parts in other objects.

The current method strongly relies on the initial segmentation of the objects to parts and its hierarchy. A stronger approach would try and analyze or partition the object in various ways depending on the query context. This would allow more flexible analogies to be found and better support to partial matching which is not restricted to the given partitioning. Moreover, since both object partitioning and parts signatures are based on the SDF, the method is suitable to certain types of objects. In the future we would like to extend it's applicability to other types of objects such as CAD models.

References

[ABM*06] ATTENE M., BIASOTTI S., MORTARA M., PATANE G., SPAGNUOLO M., FALCIDIENO B.: Computational methods for understanding 3d shapes. *Computers & Graphics 30*, 3 (June 2006), 323–333.

[AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer 22*, 3 (2006), 181–193.

[Ale02] ALEXA M.: Recent advances in mesh morphing. *Comput. Graph. Forum 21*, 2 (2002), 173–196.

[ARSF07] ATTENE M., ROBBIANO F., SPAGNUOLO M., FALCIDIENO B.: Semantic annotation of 3d surface meshes based on feature characterization. In *Proceedings SAMT 2007* (2007), vol. Lecture Notes in Computer Science, Springer-Verlag, pp. 126–139.

[BKS*05] BUSTOS B., KEIM D. A., SAUPE D., SCHRECK T., VRANIĆ D. V.: Feature-based similarity search in 3d object databases. *ACM Computing Survey 37*, 4 (2005), 345–387.

[BMP01] BELONGIE S., MALIK J., PUZICHA J.: Matching shapes. In *Proceedings ICCV* (2001), pp. 454–463.

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 11 (Nov 2001), 1222–1239.

[CCM97] CHOI H., CHOI S., MOON H.: Mathematical theory of medial axis transform. *Pacific Journal of Mathematics 181*, 1 (1997), 57–88.

[CDS*05] CORNEA N. D., DEMIRCI M. F., SILVER D., SHOKOUFANDEH A., DICKINSON S., KANTOR P. B.: 3d object retrieval using many-to-many matching of curve skeletons. In *SMI '05: Proceedings, Shape Modelling and Applications Conference* (2005), IEEE Computer Society.

[CGM07] CAMOSSI E., GIANNINI F., MONTI M.: Deriving functionality from 3d shapes: Ontology driven annotation and retrieval. *Computer-Aided Design & Applications 4*, 6 (2007), 773–782.

[FKMS05] FUNKHOUSER T., KAZHDAN M., MIN P., SHILANE P.: Shape-based retrieval and analysis of 3d models. *Communications of the ACM 48*, 6 (June 2005), 58–64.

[FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3d shapes with priority-driven search. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 131–142.

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1 (2006), 130–150.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *Proc. Symp. Geom. Processing* (2005).

[GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose oblivious shape signature. *IEEE Transactions of Visualization and Computer Graphics 13*, 2 (2007), 261–Ű271.

[JH99] JOHNSON A., HEBERT M.: Using spin images for efficient object recognition in cluttered 3d scenes. 433–449.

[KS04a] KRAEVOY V., SHEFFER A.: Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graph. 23*, 3 (2004), 861–869.

[KS04b] KRAEVOY V., SHEFFER A.: Cross-parameterization and compatible remeshing of 3d models. *ACM Trans. Graph. 23*, 3 (2004), 861–869.

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2003) 22*, 3 (2003), 954–961.

[NDK05] NOVOTNI M., DEGENER P., KLEIN R.: *Correspondence generation and matching of 3d shape subparts*. Tech. Rep. CG-2005-2, Friedrich-Wilhelms-Universität Bonn, June 2005.

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. *ACM Trans. Graph. 23*, 3 (2004), 870–877.

[Sha07] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum* (2007), to appear.

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Shape Modeling International* (June 2004).

[SP04] SUMNER R. W., POPOVIC J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (2004), 399–405.

[SSCO07] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer* (2007), to appear.

[SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S.: Skeleton based shape matching and retrieval. In *SMI '03: Proceedings, Shape Modelling and Applications Conference* (2003), IEEE Computer Society, p. 130.

[TV04] TANGELDER J. W. H., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. In *SMI '04: Proceedings of the Shape Modeling International 2004 (SMI'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 145–156.

[ZK04] ZABIH R., KOLMOGOROV V.: Spatially coherent clustering using graph cuts. *CVPR'04 02* (2004), 437–444.

Figure 8: *Automatic partitioning examples*

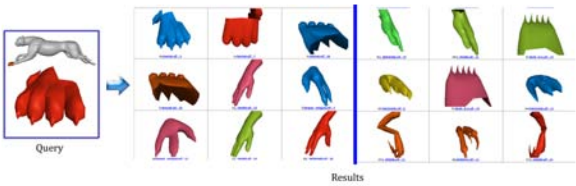


Figure 9: *In some cases, the geometrical attributes are sufficient to define a good distance metric, as evident here, searching for parts similar to the cheetah paw.*

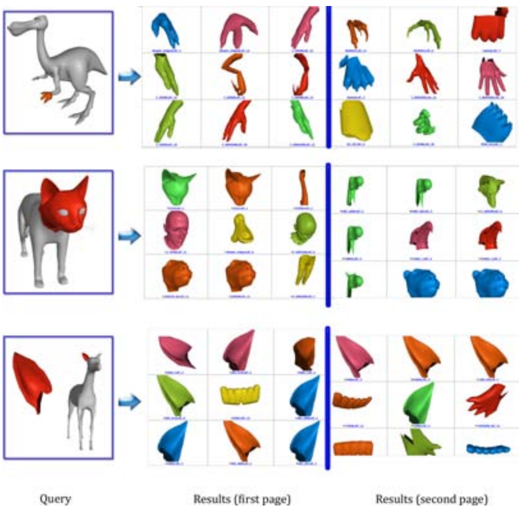


Figure 10: *Results of several example part queries. On the left is the query part and on the right the search results. The database used consists of 2200 parts partitioned from 102 models.*

Figure 11: *Analogies between parts of whole models.*

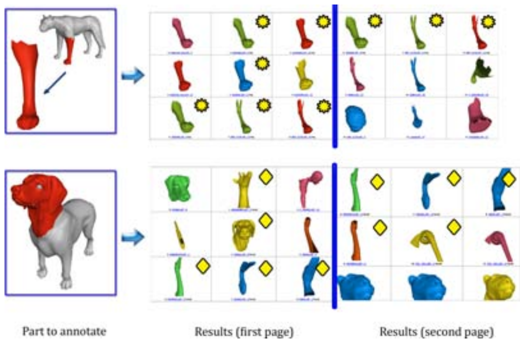


Figure 12: *Two examples of automatic annotation transfer. (top) We search for the cheetah's leg. All results marked with a yellow asterisk have already been tagged as* ***'leg'****. (bottom) We search for the dog's head. All results marked with a yellow diamond have already been tagged* ***'head'****. These tags are now transferred to the query parts.*