EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2014)
R. Klein and P. Santos (Editors)

*Short Paper*

# Content-based Retrieval of 3D Models using Generative Modeling Techniques

Harald Grabner[1], Torsten Ullrich[1], and Dieter W. Fellner[1, 2]

[1]Technische Universität Graz & Fraunhofer Austria Research GmbH, Austria
[2]Technische Universität Darmstadt & Fraunhofer IGD, Germany

**Abstract**

*In this paper we present a novel 3D model retrieval approach based on generative modeling techniques. In our approach generative models are created by domain experts in order to describe 3D model classes. These generative models span a shape space, of which a number of training samples is taken at random. The samples are used to train content-based retrieval methods. With a trained classifier, techniques based on semantic enrichment can be used to index a repository. Furthermore, as our method uses solely generative 3D models in the training phase, it eliminates the cold start problem. We demonstrate the effectiveness of our method by testing it against the Princeton shape benchmark.*

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Computer Graphics]: Information Systems—Information Search and Retrieval, I.2.4 [Computer Graphics]: Knowledge Representation Formalisms and Methods—Representations (procedural and rule-based), I.4.8 [Computer Graphics]: Scene Analysis—Object recognition

## 1. Motivation

According to the idea of generalized documents, multimedia data and in particular 3D data sets should be treated just like ordinary text documents, so that they can be inserted into a digital library. As a consequence, these media types must be integrated with the generic services that a library provides, namely markup, indexing, and retrieval.

With the integration of 3D models into digital cultural heritage libraries, new research challenges arise. The context of cultural heritage distinguishes itself by model complexity, model size, and imperfection to such an extent most approaches cannot handle [USF08]. As a consequence, our approach is to promote generative modeling techniques, which act as a link between computer science on the one hand and domain know-how on the other hand [SSUF10].

In this paper we present a novel training approach based on generative modeling techniques and a new retrieval technique. In the training phase, the generative models span a shape space, of which a number of training samples is taken at random. The big advantage of procedural modeling techniques is the included expert knowledge within an object description [UF11]; e.g. classification schemes used in architecture, archaeology, civil engineering, etc. can be mapped to procedures. For a specific object only its type and its instantiation parameters have to be identified. This identification is required by digital library services: markup, indexing, and retrieval. With generative models in the training phase, no “real” training data is needed a priori. The generative models themselves are represented as JavaScript code, which takes a number of parameters and returns a 3D model.

To illustrate the applicability of our approach two retrieval methods have been implemented: the established *salient local visual features* method [OOFB08] and our new algorithm called *histogram of inverted distances*.

## 2. Related Work

Our approach combines techniques of content-based retrieval and machine learning with shape description and generative modeling.

**Generative Modeling** With increasing complexity the manual creation of 3D models has become unfeasible. As a consequence, generative modeling has been developed in order to generate highly complex objects based on a set of formal construction rules. An overview on generative modeling techniques can be found in the survey by WATSON and WONKA [WW08], int he overview by VANEGAS et al. [VAW*10] and in the tutorial notes by KRISPEL et al. [KSU14].

DOI: 10.2312/gch.20141317

The first generative modeling approaches have always been text-based scripts exposing their algorithmic character. In combination with annotation techniques developed in the field of software engineering, a procedural model can be enriched semantically: a way to describe procedural knowledge and information about an object's inner structure, symmetry, and regularity [SSUF10]. Furthermore, annotations and human-readable meta data can be propagated easily; i.e. the human-readable description of an object class can be transferred to every identified class instance and to every sufficiently similar model.

**Content-Based Retrieval** Many content-based retrieval methods for 3D models have been proposed recently. Tangelder et al. [TV08] and Bustos et al. [BKSS07] have both surveyed literature on content-based retrieval methods. Tangelder et al. divide shape matching methods into three categories: feature-based, graph-based and geometry-based methods.

For the training phase, the above mentioned methods need a given sample set. This introduces a cold start problem. Ullrich and Fellner [UF11] circumvent this problem by fitting generative models to the test data, so only the generative models must be known in advance. We use this technique to span a shape space and to take a sample set by random. This randomized subset is the input of the training phase which uses histograms.

Shape histograms have also been used by Ankerst et al. [AKKS99] to classify molecules; however their approach uses one global histogram per molecule. Kriegel et al. [KBK*03] also split their voxeled models into a regular grid of cells and calculate features vectors per cell. In contrast to our approach, they do not use the histogram of inverted distances.

## 3. Shape Description

In our approach each 3D model class is defined by one generative 3D model. A generative 3D model $M$ is an algorithm that takes an argument vector $x$ and produces 3D geometry $M(x)$. Each generative 3D model is used to generate training samples for the class it represents. Without loss of generality, the parameter domain $D(M)$ has a multidimensional, rectangular structure; i.e. the Cartesian product of closed intervals. If the generative shape is well-designed, a representative subset of the shape space is generated by randomly sampling a number of argument vectors $x_i$ from $D(M)$. These random models $M(x_i)$ are used in the training phase.

To illustrate our approach we use a generative model called "sedan car". The car model takes six parameters and generates 3D model with a fixed topology and varying geometry.

## 4. Histogram of Inverted Distances

After the generation of the training models, all models are scaled to a common size, aligned using Principal Component Analysis (PCA), voxelized into a grid of $R \times R \times R$ elements. The center of gravity of all models is $\left({}^{R}\!/_{2}, {}^{R}\!/_{2}, {}^{R}\!/_{2}\right)$. If $n$ is the number of training models, the family of aligned training models is noted $(T_i)_{\{i \in 1,\dots,n\}}$. After alignment, the value $\mathbf{v}(T_i,x,y,z)$ of a given element in the voxel grid is either 1, if the voxel contains a part of the surface of the model or zero, otherwise. Based on the aligned training models the inverse distance models are computed. The volume of the distance transformed training samples is defined by

$$\mathbf{v}(T_i,x,y,z) = \max\left\{\frac{cut - d(T_i,x,y,z)}{cut}, 0\right\}, \tag{1}$$

where $d(T_i,x,y,z)$ denotes the Euclidean distance of point $(x,y,z)$ to the model's surface. The value $cut$ adjusts the rate of diffusion of the inverse distance transformation. For the calculation of the inverted distance neither manifoldness nor watertightness is necessary. In fact, the inverse distance transformation could also be calculated for point sets. After the calculation of the inverse distance model, the model is split into a regular grid of cubic cells (which are larger and comprehend several grid elements). Let $p$ denote the number of cells along one axis ($p < R$), then the total number of cells is $p^3$. Each cell has a side length $s$ with $s = \frac{R}{p}$ and the family of the cells for model $i$ is denoted $C_i$. For each cell $C_i$ the normalized histogram of inverse distances with $k$ bins is calculated. This feature vector of cell $C_i$ is denoted as $h_i \in [0,1]^k$. Based on the feature vectors $h_i$ we estimate a non-parametric density function for each cell position $(a,b,c)$ using Gaussian kernel density estimation [Bis07]. The density function for a cell at position $(a,b,c)$ is

$$P(h'_{a,b,c}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{(2\pi\sigma^2)^{k/2}}\cdot\exp\left(-\frac{\|h'_{a,b,c} - h_{i,a,b,c}\|^2}{2\sigma^2}\right), \tag{2}$$

where $h'$ is the feature vector of a test model. $\sigma$ represents the standard deviation of the Gaussian kernel, which acts as a smoothing factor. Usually the standard deviation can be estimated easily using appropriate estimation methods [JMS96]. However, in this case, at some positions all the features are exactly the same. This situation often occurs at border cells, where the inverse distance to the surface is zero everywhere. To solve this problem, the standard deviation $\sigma$ has been set to an empirically determined value.

Matching a given test model is equivalent to approximating the probability of the test model belonging to a learned class. In analogy to the generative models during the learning phase, the test model needs to be voxelized and aligned using PCA. Based on the aligned test model $X$, the inverse distance transformation can be calculated. Like in the training phase, the inverse distance transformed model is partitioned into cells and the corresponding histograms are calculated. Using the density functions for each cell, the probability of a sample object belonging to a learned class can be approximated:

Let $X$ be a test model and $h'_{(a,b,c)}$ denote the feature vectors of the test model, then the joint probability of model $X$ belonging to the learned class is

$$\prod_{(a,b,c)\in(1\dots p)^3} P(h'_{(a,b,c)}, a, b, c). \tag{3}$$

We call this algorithm the histogram of inverted distances (HID) algorithm.

## 5. Salient Local Visual Features Method

To demonstrate that the generative training approach can be combined with different retrieval techniques, we implemented the salient local visual features method. It is a feature-based retrieval method, which operates on range images of a 3D model and has been introduced by OHBUCHI et al. [OOFB08]:

After the generation of the training models, the models are normalized. This is done by scaling them uniformly to a common size and centering them at the origin. For each normalized training model, range images are rendered from 42 viewpoints around the 3D model. The viewpoints are defined by the vertices of the polyhedron, generated by subdividing an icosahedron.

After the range images are rendered for a normalized training model, the Scale Invariant Feature Transform (SIFT) algorithm [Low04] is applied to the range images. The SIFT algorithm extracts salient visual features, which are invariant to position, scale and orientation. Each visual feature is described by a 128-dimensional feature vector. In our case 35 visual features were extracted per range image on average; consequently, the average number of visual features per 3D model was approximately 1500.

Using the visual word codebook (described below), each visual feature of a training model can be quantized into a visual word. This is done by assigning each feature to its closest cluster centroid. By accumulating the visual words into a histogram and normalizing the histogram, a feature vector for the 3D model is calculated. The size of the feature vector is equal to the number of visual words in the visual codebook. 3D Models can be compared by calculating the Manhattan distance between two feature vectors. Let $w$ be the number of words in the visual dictionary, then $l_{(i)} \in [0,1]^w$ denotes the visual word histogram of the $i$th training sample.

Like in the histogram of inverted distances algorithm, the visual word histogram is learned using kernel density estimation. However, in this case Manhattan distance $L_1$ is used as the kernel function. The similarity $S$ for the 3D model described by the visual word histogram $l'$ is given by:

$$S(l') = \frac{1}{n}\sum_{i=1}^{n} L_1\left(\frac{l' - l_{(i)}}{\sigma}\right), \tag{4}$$

where $n$ denotes the number of training samples and $\sigma$ represents the smoothing factor.

The visual codebook quantizes visual features into visual words. The visual codebook is learned unsupervised in a preprocessing step using *k-means++* clustering [AV07]. The set of visual features that have to be clustered is selected randomly from all views of the 3D models.

## 6. Results

We evaluated both retrieval methods with the Princeton shape benchmark [SMKF04] using the class "sedan car". The complete benchmark consists of 907 test samples including 10 sedan cars.

**Evaluation of the Histogram of Inverted Distances Method** The values for the retrieval parameters were evaluated empirically. The histogram of inverted distances method is able to find similar objects to the given generative models. The retrieval results for the class "sedan car" are almost perfect. Figure 1 shows the top 16 retrieval results for both retrieval methods.

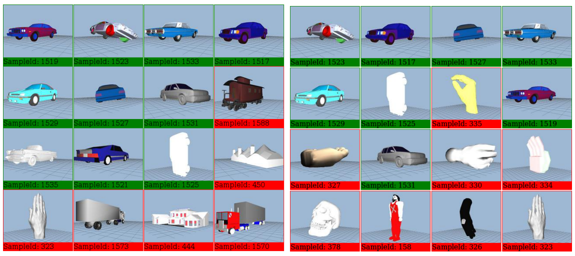

**Figure 1:** *The top 16 retrieval results for the class "sedan car" using the histogram of inverted distances method (left) are almost perfect. The ten car models of the benchmark are listed within the best 16 matches. The top 16 retrieval results for the classes "sedan car" (right) have been generated using the salient local visual features method.*

### Evaluation of the Salient Local Visual Features Method

Each each sample of the benchmark the range images have been rendered, visual features have been extracted and the visual word histogram has been calculated. The visual codebook consists of the extraction of 30000 visual features from a random subset of the benchmark and the clustering of the features into 1024 clusters.

The results in Figure 1 show that the salient local visual features method is able to find similar objects to the given generative models.

## 7. Conclusion

We presented two new approaches to perform content-based retrieval of 3D shapes: generative training and the histograms of inverted distances.

Generative training uses procedural models to describe 3D model classes, respectively, 3D shape spaces. In the training phase, the shape spaces are sampled randomly. In this way, no “real” training data is needed a priori. This technique can be combined with various retrieval algorithms. The big advantage of procedural modeling techniques is the included expert knowledge within an object description [UF11]; e.g. the knowledge of an expert about the inner structure and the semantics of an object class can be mapped to procedures [USSF13]. Within the Cultural Heritage (CH) project “‘ProFitS” we incorporate this technique to index a CH repository semantically using expert knowledge. The approach of a generative training set, which does not need any “real” data can be combined with various retrieval algorithms. We have presented two retrieval methods to illustrate this approach.

The first method is called the histogram of inverted distances method. Using a voxel representation, PCA alignment and inverse distance transformations on a grid, each grid cell’s histogram is the basis to learn a non-parametric density function. In the recognition phase, the test object is processed the same way, so that for each of its cells the similarity of a learned object class is estimated using the corresponding learned density function. The similarity of the whole model is given by the product of all cell similarities.

The second method is called the salient local visual features method. Salient local visual features are extracted from range images of a 3D model by the SIFT algorithm. Using a precomputed visual codebook, the visual features are quantized into a histogram of visual words, which acts as a feature vector. Using the feature vectors a non-parametric density function is learned for each 3D model class. In the recognition phase, the feature vector is calculated for the test object and the similarity is estimated using the learned non-parametric density function.

Our contribution to 3D documents is a shape retrieval approach based on machine learning and generative modeling. In this way, we provide a classification technique, which uses generative modeling to encode expert knowledge in a way suitable for automatic classification and indexing of 3D repositories. We have shown that it is possible to train a retrieval method using generative models only. As a benefit (not only for users of our method), this technique eliminates the cold start problem in the training phase. A generative description implemented in a few lines of code is sufficient to generate a reasonable training set. Furthermore, we have shown that the histogram of inverted distances can be used as a feature vector for spatial data.

## Acknowledgements

The authors gratefully acknowledge the generous support of the European Commission within the DURAARK project founded by the program “ICT-2011-4.3-Digital Preservation”.

## References

[AKKS99] Ankerst M., Kastenmüller G., Kriegel H.-P., Seidl T.: 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. *Advances in Spatial Databases (Lecture Notes in Computer Science) 1651* (1999), 207–226. 2

[AV07] Arthur D., Vassilvitskii S.: k-means++: The Advantages of Careful Seeding. *Proceedings of the annual ACM-SIAM symposium on discrete algorithms 18* (2007), 1027–1035. 3

[Bis07] Bishop C. M.: *Pattern Recognition and Machine Learning*. Springer, 2007. 2

[BKSS07] Bustos B., Keim D., Saupe D., Schreck T.: Content-based 3D Object Retrieval. *IEEE Computer Graphics and Applications 27*, 4 (2007), 22–27. 2

[JMS96] Jones M. C., Marron J. S., Sheather S. J.: A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the American Statistical Association 91* (1996), 401–407. 2

[KBK*03] Kriegel H.-P., Brecheisen S., Kröger P., Pfeifle M., Schubert M.: Using sets of feature vectors for similarity search on voxelized CAD objects. *Proceedings of the ACM International Conference on Management of Data (SIGMOD) 29* (2003), 587–598. 2

[KSU14] Krispel U., Schinko C., Ullrich T.: The Rules Behind – Tutorial on Generative Modeling. *Proceedings of Symposium on Geometry Processing / Graduate School 12* (2014), 2:1–2:49. 1

[Low04] Lowe D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60* (2004), 91–110. 3

[OOFB08] Ohbuchi R., Osada K., Furuya T., Banno T.: Salient local visual features for shape-based 3D model retrieval. *Proceeding of the IEEE International Conference on Shape Modeling and Applications 8* (2008), 93–102. 1, 3

[SMKF04] Shilane P., Min P., Kazhdan M., Funkhouser T. A.: The Princeton Shape Benchmark. *Shape Modeling International 8* (2004), 1–12. 3

[SSUF10] Schinko C., Strobl M., Ullrich T., Fellner D. W.: Modeling Procedural Knowledge – a generative modeler for cultural heritage. *Proceedings of EUROMED 2010 - Lecture Notes on Computer Science 6436* (2010), 153–165. 1, 2

[TV08] Tangelder J. W. H., Veltkamp R. C.: A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications 39* (2008), 441–471. 2

[UF11] Ullrich T., Fellner D. W.: Generative Object Definition and Semantic Recognition. *Proceedings of the Eurographics Workshop on 3D Object Retrieval 4* (2011), 1–8. 1, 2, 4

[USF08] Ullrich T., Settgast V., Fellner D. W.: Semantic Fitting and Reconstruction. *Journal on Computing and Cultural Heritage 1*, 2 (2008), 1201–1220. 1

[USSF13] Ullrich T., Schinko C., Schiffer T., Fellner D. W.: Procedural Descriptions for Analyzing Digitized Artifacts. *Applied Geomatics 5*, 3 (2013), 185–192. 4

[VAW*10] Vanegas C. A., Aliaga D. G., Wonka P., Müller P., Waddell P., Watson B.: Modelling the Appearance and Behaviour of Urban Spaces. *Computer Graphics Forum 29* (2010), 25–42. 1

[WW08] Watson B., Wonka P.: Procedural Methods for Urban Modeling. *IEEE Computer Graphics and Applications 28*, 3 (2008), 16–17. 1