Eurographics Workshop on 3D Object Retrieval (2015)
I. Pratikakis, M. Spagnuolo, T. Theoharis, L. Van Gool, and R. Veltkamp (Editors)

3D Object Retrieval with Parametric Templates

Roman Getto[1] and Dieter W. Fellner[1,2,3]

[1]Interactive Graphics Systems Group, TU Darmstadt, Germany.
[2]Fraunhofer IGD, Darmstadt, Germany.
[3]Institute of Computer Graphics and Knowledge Visualization, TU Graz, Austria.

Abstract

We propose a 3D object retrieval system which uses parametric templates as prior knowledge for the retrieval. A parametric template represents an object-domain and a semantic concept like 'chair' or 'plane' or a more specific concept like 'dining-char' or 'biplane'. The template can be specified at a general or specific level and can even equal actual retrieved objects. The parametric template is composed of several input parameters and an operation chain which constructs an object. Different parameter combinations lead to different object instances.
We combine and evaluate a paramteric template with different descriptors. Our results show that the usage of parametric templates can raise the retrieval performance significantly.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Computer Graphics]: Information Storage and Retrieval—Retrieval Models

1. Introduction

The classic content-based 3D object retrieval scenario consists of the retrieval of the most similar objects from a database, given a query object. For this purpose we usually want to compute a descriptor which should represent the proprieties of an object. The descriptor computation has experienced much attention in the last decades and the state-of-the-art descriptors have already reached an astonishing quality. The question raises if we can combine these descriptors with another concept to reach an even improved retrieval.

Several approaches did already tackle the 3D retrieval problem from another side. Not by improving the retrieval process and the descriptor itself but by improving the knowledge about the searched object. Some approaches include custom search queries, e.g. by drawing the searched object [FMK*03]. Other approaches improve the knowledge about the searched object by including prior knowledge about the searched object-class [LJ13, XL07, BGM*07]. This especially aims at retrieving object-classes which have a high diversity of forms and therefore are hard to retrieve consistently if we only have one query object.

Our approach also uses a kind of prior knowledge for enhancing the retrieval. Most prior knowledge based approaches integrate a learning phase using labeled objects. Our approach uses a predefined parametric template as prior knowledge. The parametric template is composed of several input parameters and an operation chain which constructs an object. Different parameter combinations lead to different object instances (See Figure 1).

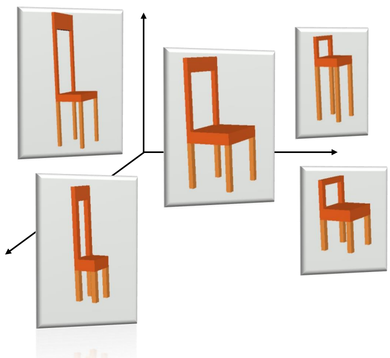


Figure 1: *The parametric template of a dining chair spans a multidimensional space of dining chair objects. In general, parametric templates can be very detailed and have any number of parameters. This makes the use for retrieval difficult, since the object space can be arbitrarly large and the number of dimensions of the object space equals the number of parameters of the template.*

A parametric template represents an object-domain,

DOI: 10.2312/3dor.20151054

which means that it corresponds to a semantic concept like 'chair' or 'plane', but it can also correspond to a more specific semantic concept like 'dining-chair' or 'biplane'. The level of specificity is defined with the template itself. If the template is very detailed, instances of the template can also equal exactly searched objects.

We define parametric templates on the basis of generative modeling using the Generative Modeling Language (GML: http://www.generative-modeling.org/). Generative modeling is a method to model an object without explicitly describing each component (e.g., polygon). The template is described in a generative way: as a chain of operations. When the operations are executed the object is constructed. Therefore the operations itself represent a description of an object.

For the retrieval of objects which are similar to a parametric template we generate instances of the template and compare them to the objects of a database using a descriptor. Our system is build in a way that it can use any descriptor for the comparison. We evaluate our system by combining a state-of-the-art descriptor with a predefined parametric template.

Contribution: To be clear, the descriptor is not developed by us nor did we invent the parametric templates. Our contribution is the definition of an appropriate parametric template space for the retrieval, as well as the development of a descriptor-independent retrieval system using these parametric templates and the combination and evaluation with different descriptors.

2. Related Work

We divide this section into three subsections. The first subsection gives a short overview of descriptors developed for 3D object retrieval, these are relevant as we use a descriptor in combination with a parametric template.

Since our work belongs to the 'prior-knowledge-based 3D object retrieval' systems, the second subsection discusses approaches similar to our own, which are based on templates or class-training. In the third subsection we review work related to generative modeling, which we use for the definition of parametric templates.

2.1. Descriptors

A descriptor describes an object in a (more or less) short manner, so that objects can be compared simply by comparing their descriptors. Descriptors can be based on several different proprieties of objects. They can be based on topological proprieties, geometrical features of the surface or just general appearance properties. The survey of Tangelder et al. [TV08] separates descriptors into feature based, graph based and geometry based descriptors.

Feature based descriptors include two different approaches. The first approach measures features over the whole surface and merge them to a global descriptor. The features can be based on, e.g., D2 distances [OFCD02], spherical harmonics [KFR03], oriented gradients [SWS10], 3D zernike polynomials [NK03] or density functions [ASYS09]. The second feature based approach only integrates (sparse) local features into the descriptor. This can happen in a bag-of-words manner [BBGO11, TCF09] or SIFT-based [DK12, MFK*10] or SURF-based [KPW*10].

Graph based descriptors describe the topology of an object as graph, e.g., as skeleton-graph [SSGD03]. Geometry based includes the residual, but the most relevant are the view-based descriptors which describes the object from 2D views, like the LightField descriptor [CTSO03]. There are also hybrid approaches combining view-based with feature-based, like the DESIRE descriptor [Vra05] and the PANORAMA descriptor [PPTP10].

In Section 5 we will evaluate several descriptors in combination with a parametric template and we will discuss the proprieties of the descriptors.

2.2. Prior Knowledge and Template-based Retrieval

Several approaches make use of prior knowledge. The knowledge can be present in different ways. The most straightforward way is using labeled objects for learning an object class in advance. Hou et al. [HLR05] use a support vector machine to learn a model and then retrieve objects similar to this model. Li et al. [LJ13] also computes a descriptor and uses prior known class information so that objects can be classified to their nearest class instead of just taking the nearest neighbor of the training set. Xu et al. [XL07] learn a neural network with the labeled data and classify new objects based on the neural network.

Van Kaick et al. [VKTS*11] analyze a set of labeled objects representing one class to learn a consistent segmentation and point-to-point correspondence over the training set. New objects of the same class can then be segmented accordingly and point-to-point correspondences can be established. Although no 3D object retrieval is present in this approach, this interesting approach could probably be adapted to 3D object retrieval.

Prior knowledge can also be present in the form of user-input before the retrieval. Patterson et al. [PIMD08] lets the user choose a part of an object which should be retrieved. The part is then searched and retrieved within the object itself. In the work of Sunkel et al. [SJWS13] the user labels the searched objects in a range image training scene and afterwards the same objects are retrieved in a new scene.

Several approaches [GWJ*14, LZYX15, LMT05, ASYS10] include relevance feedback to improve the retrieval performance. In relevance feedback the user labels results as correct or wrong, so that the learned model can be corrected. This is not a prior knowledge approach, since the knowledge is acquired afterwards, but in general this approach is similar to prior learning from labeled objects. The advantage of relevance feedback is that no pre-labeled data is needed.

The approaches most similar to ours, are those who use templates for the retrieval. Biasotti et al. [BGM*07] construct structural prototypes during a learning phase and use these to improve the retrieval.

However, several template based approaches are not developed for 3D object retrieval, but for the exploration of collections and creation of new combined objects. Averkiou et al. [AKZM14] extract part-based templates from collections of objects and synthesize them to new combinations of object parts. Jain et al. [JTRS12] co-segments two objects and produce different combinations of the two objects. Xu et al. [XLZ*10] generates correspondeces between parts of objects and then combines and scales these parts, deriving new objects.

Osjanikov et al. [OLGM11] construct very flexible part-based template models from a set of objects. These are developed for the exploration of a repository, but it would be interesting to use this approach for retrieval. Nevertheless, their template models only work if the variation within an object group is reasonable, which means that mostly the same parts differ in position and size (vases would not be possible).

Kim et al. [KLM*13] also construct part-based templates. They use simple manual templates as input and refine these with a given training set. Afterwards they use the templates to segment and construct point-to-point correspondences for new objects.

Part-based templates are typically more restricted than our approach. Parametric templates can represent a class in a general way but they can also be very specific even equaling the retrieved objects. Therefore, our approach can include a much higher variance than part-based templates. This is especially interesting for higher semantic concepts which can include a broad variety of different parts.

2.3. Generative Modeling

In the last century Ramamoorthi et al. [RA99] already created parametric templates by using generative modeling.

A template described with generative modeling can have any number of parameters which change the construction process of the object. Therefore we can create a parametric template which spans a continuous space of objects using generative modeling. Describing an object in operations can be a very cheap description as high-level-operations can also be constructed from low-level-operations and therefore be reused several times in a description. Berndt et al. [BFH05] have shown that this description of an object is also perfect for transmission as it uses less bandwith .

Ullrich et al. [UF11] showed that a parametric template can be used to recognize and describe an object. Our approach is motivated by this proof of concept.

Generative modeling is also highly related to procedural modeling. In procedural modeling a set of rules can be arbitrary combined to create new objects. Haegler et al. [HMVG09] uses procedural modeling to describe different building which are equal in their kind of structure. Marvie et al. [MPB05] also construct different building with procedural modeling by using grammars.

The process of inverse procedural modeling constructs the rules from existing objects, enabling the possibility to combine the rules in a new way to produce new objects. Bokeloh et al. [BWS10] uses this approach to define castle-wall construction rules and shows that it is possible to arbitrary combine these rules to construct new wall-combinations. Fisher et al. [FRS*12] also uses inverse procedural modeling and finds rules for office-desk arrangements and combines them to new arrangements.

In sum, procedural and generative modeling approaches have shown that these are very powerful tools. The major advantage is that parametric templates describe a continuous space of objects, therefore containing many object variants which could not be covered with a database of example objects. It is not necessary to know every possible object in advance but rather we can just generate the objects we need for comparison. Still, the retrieval results are heavily dependent on the correct definition of a template. We are investigating possibilities of automatically or semi-automatically retrieving new parametric templates from known objects (i.e., inverse generative modeling). For this work we have developed the used parametric template manually.

3. Definition of a Parametric Template using Generative Modeling

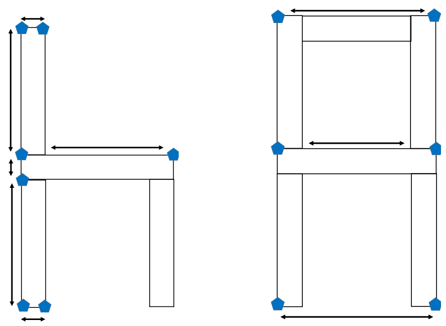
A parametric template consists of a chain of operations and a set of parameters. Executing the operations with the given parameters leads to a mesh representation of an object. Such an object is called an instance of the parametric template. Hence, a parametric template can be understood as the definition of an object construction algorithm. Dependent on the parameters different instances are generated by the algorithm, therefore a parametric template spans a multidimensional space of objects (illustrated in Figure 1).

The construction algorithm can be arbitrarily complex and process any number of parameters. Hence, in general, any object can be created with a parametric template. Nevertheless, an improperly complex high-dimensional representation is cumbersome to handle and especially unsuitable for 3D object retrieval.

Therefore, when using generative modeling to create a parametric template representing an object domain and a semantic concept like "chair","dining chair" or "biplane" one has to consider in which proprieties the objects should change and how much they should change, i.e. clearly defining and restricting the change axes of an object domain.

Indeed, if used correctly the resulting parametric template, i.e. object domain construction algorithm, can be a very compact representation and be very handy. The construction algorithm can consist of high-level-operations which are a reusable combinations of low-level-operations. Hence, the algorithm can be rather short. A representation of an object in operations can be more compact than a usual representation as a list of polygons [BFH05].

We use the Generative Modeling Language (GML: http://www.generative-modeling.org/) created by Sven Havemann [HF05]. The operations are based on Euler operations working on a half-edge structure of a boundary representation (B-Rep) of a mesh.



(a) Side view (b) Front view

Figure 2: *The basic concept of a simple chair construction. The chair is represented by 10 points (3 points are shown in both views). The distance of the points to each other define the size of each part of the chair.*

In Figure 2 we illustrate a basic concept for a parametric template of a chair. The parameters are 10 points represented by (x,y,z)-coordinates. The points define the size of all parts of the chair, but also where there chair is placed in Euclidean 3D space. Note, that this definition does not restrict how the points are related one to another. Therefore choosing bad values for the parameters could also result in a self-overlapping or degenerated mesh. Or the result could simply just not look like a chair.

For the use in 3D object retrieval we choose a restrictive space-invariant representation of a parametric template, which limits the possible change-axes of the chair.

As we want to measure the difference of objects in a orientation, translation and scaling invariant form we do not have to consider (x,y,z)-coordinates of the parameter points. We are much more interested in the ratios within the object, i.e. is the back of the chair much longer than the legs? Or are the legs of the chair thin or thick in relation to the height of the chair?

For our parametric template of a dining chair we restrict one size c and introduce 3 parameters p_1, p_2, p_3, defining the ratios within the chair. The thickness t and the length l are defined as follows:

$$t_{legs} = \frac{3}{4}c \qquad l_{legs} = p_1 \cdot 10c$$
$$t_{back} = \frac{3}{4}c \qquad l_{back} = p_2 \cdot 10c$$
$$t_{seat} = c \qquad l_{seat} = p_3 \cdot \frac{3}{4} \cdot 10c$$

We additionally restrict the possible change to a closed range: $p_1, p_2, p_3 \in [0.5, 1.0]$

The seat is considered to be quadratic and therefore only has one size describing his side length. Note that the side length of the seat has a diminishing factor of $\frac{3}{4}$ since the size variance of the seat is smaller than the size variance of the legs and the back. The range $[0.5, 1.0]$ means that the length of the legs l_{legs} and the back l_{back} are 5 to 10 times the size of the thickness constant c. The side length of the seat l_{seat} is 3.75 to 7.5 times the size of the thickness constant c.

4. 3D Object Retrieval with Parametric Templates

The retrieval with parametric templates $T_{\{p\}}$ can be difficult as the template can contain many parameters $\{p\}$ and therefore the spanned space of objects can be very high dimensional.

Comparing an object of the database with every possible instance generated from the parametric template is impossible as these are infinite. Therefore we need to sample the space of possible objects and generate a set of instances $\{i : i \in T_{\{p\}}\}$. When we have a set of instances of the template we can simply compare each instance to a database-object.

To enable the generation of a reasonable small set of instances of the parametric template we sample the 3-parameter representation $T_{p_1 p_2 p_3} \subset T_{\{p\}}$ described in the previous section. For each dimension we take one sample each 0.05 of the restricted value range $p_i \in [0.5, 1.0]$. This results in 11 sample values for each dimension and 1331 instances of the parametric template.

Now we can compare each sampled instance i to a database-object o. For this purpose we compute a descriptor $d(..)$ for both. The difference of the descriptors $\delta[d(i), d(o)]$ represents the difference between the objects.

For the final retrieval result of a parametric template $T_{\{p\}}$ we aggregate the comparison of each instance. We define the difference of an object to the template as the distance to his nearest neighbor in the object space of the template:

$$\delta[T_{\{p\}}, o] = min\{\delta[d(i), d(o)] : i \in T_{\{p\}}\}$$

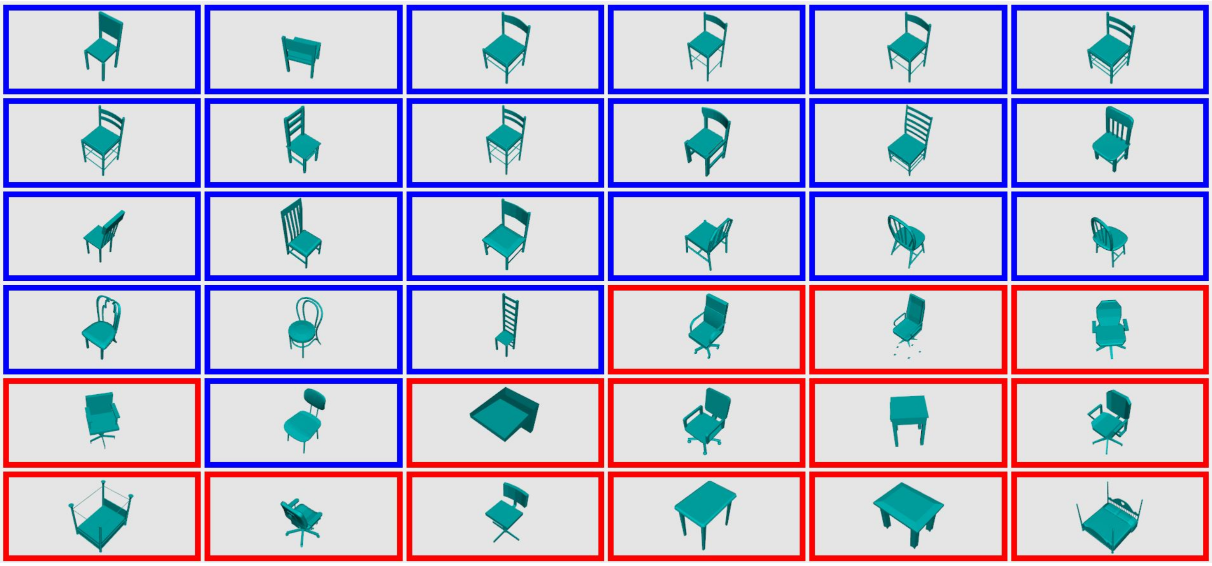


Figure 3: *We show the retrieval results using the parametric template "dining chair" in combination with the PANORAMA descriptor in the Princeton Shape Benchmark database. The first 36 retrieved objects are shown. All 22 members of the class "dining chair" are within the result. The results are ordered primarily from left to right and secondarily from top to bottom.*

5. Evaluation

We use the Princeton Shape Benchmark (PSB) [SMKF04] for the evaluation of our approach. We combine the parametric template of the dining chair with an appropriate descriptor and retrieve the most similar objects from the PSB database (1815 models). We consider every object classified as "dining chair" within the database as a correct retrieval.

As our approach can be combined with any descriptor, We have chosen several descriptors for the evaluation. Nevertheless, to use the full advantage of the parametric templates, we considered different proprieties of the descriptors.

Every type of descriptor has different advantages and disadvantages. Graph based descriptors focus on the topology of an object and therefore are very good for objects which mainly are related by their topology (e.g., quadrupeds). They give poor results if the meshes are disconnected, with holes and include self-overlaps or if the searched objects have a high variety of possible topologies (e.g., vases).

Global feature based descriptors are mostly good at discriminating at a high level of difference, while local feature based descriptors are better at discriminating objects based on small (local) differences. View-based and hybrid descriptors are in between in respect to the level of discrimination. Considering our setting, which includes objects represented as arbitrary mesh soups, we looked for a state-of-the-art descriptor which can handle these meshes. Also, as our parametric template is less detailed than the retrieved objects, we excluded the local feature based methods. We preferred the view-based, hybrid and global feature descriptors.

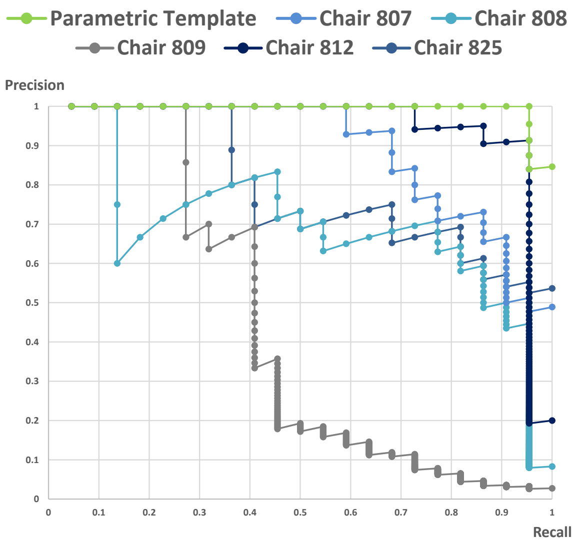


Figure 4: *Precision-Recall plot for the retrieval of the "dining chair" class (22 members) in the whole Princeton Shape Benchmark Database. The parametric template "dining chair" was combined with the PANORAMA descriptor. The single chair queries also used the PANORAMA descriptor. The 5 single queries plotted are chosen from the 22 possible queries and include the best and the worst results of the single queries.*

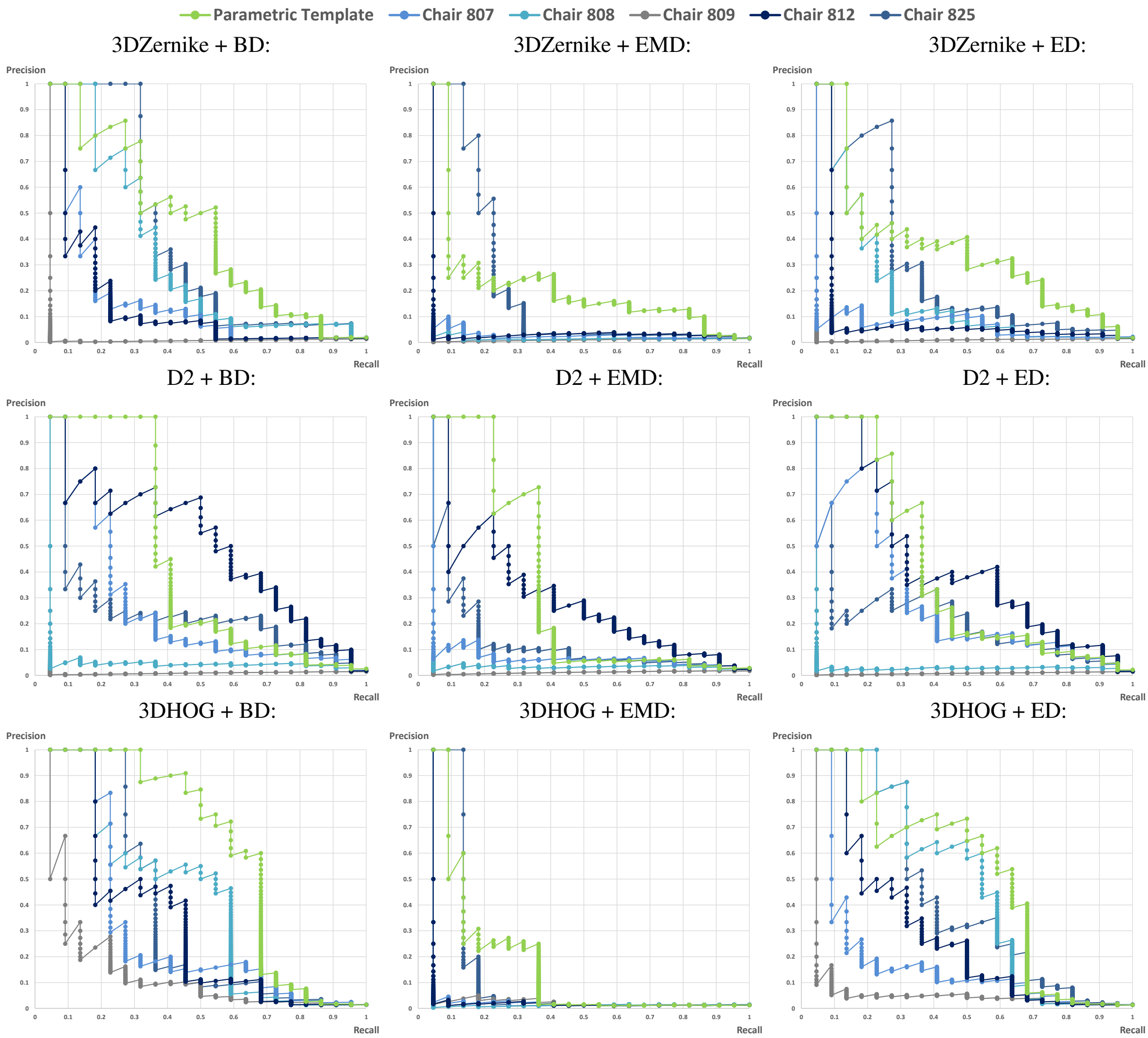


Figure 5: *Precision-Recall plots for the retrieval of the "dining chair" class (22 members) in the Princeton Shape Benchmark Database. The parametric template "dining chair" is combined with 3 different descriptors and 3 different distance measures. The descriptors are the 3D Zernike, the D2 Distribution and the 3D Histogramm of Oriented Gradients Descriptor. The distance measures are the Bhattacharyya distance (BD), Earth Mover's Distance (EMD) and Euclidean Distance (ED).*

We also considered that the descriptor has to be rotation-invariant, translation-invariant and scale-invariant for the usage with a parametric template. However, most state-of-the-art descriptors are either invariant by construction or include normalization for archiving invariances. Therefore this consideration does not exclude many descriptors.

The one state-of-the-art descriptor with the best results among these is currently the PANORAMA descriptor [PPTP10]. We also evaluated several descriptors which aggregate surface proprieties in a global (statistical) way into a feature vector. Namely we used the D2 distribution descriptor [OFCD02], the 3D zernike descriptor [NK03] and the 3D histogramm of oriented gradients descriptor [SWS10]. We compared the feature vectors with 3 well known different measures: The Euclidean distance, the Earth mover's distance and the Bhattacharyya distance.

5.1. Results

We show results of the PANORAMA descriptor in Figure 3 and 4 and the results for the other three descriptors with the three different distance measures in Figure 5. Each Precision-Recall plot shows the retrieval of the "dining-chair" class, which has 22 Members. Besides the retrieval with the parametric template we show 5 single queries in each plot. The retrieval queries use the (dining chair) models 807, 808, 809, 812 and 825 from the Princeton Shape Benchmark (PSB) Database. We evaluated all plots with all 22 dining chair single queries and (for diminishing overplotting) have chosen 5 Models which include the best and the worst results in all plots.

We evaluated 4 different Descriptors. We expected that the PANORAMA descriptor performs best as it is currently considered as one of the best descriptors. But we also considered the global feature based descriptors as these can be computed and compared faster and could probably lead to similar results in combination with a parametric model.

The results show that the PANORAMA descriptor indeed works perfectly well together with the parametric templates. A few single queries are already very good with the PANORAMA descriptor. The retrieval performance with the parametric template shows an even better result than any single query retrieval. Note that there is one chair which is hardly retrieved by any query as it differs much from the other chairs. Still, the parametric template retrieval retrieves this chair much better than any possible single query.

The best results with the other 3 descriptors are received with the Bhattacharyya distance. Surprisingly the Euclidean Distance also works reasonably well while the Earth Mover's Distance shows the worst results. The parametric template retrieval is always better or similar to the best single query retrieval, but the performance gain is not very high overall for these descriptors.

5.2. Discussion

Our Results show that the retrieval with parametric templates work very well with the right template representation and the right descriptor. Note that we excluded the local feature based descriptors as these are discriminant on a very detailed level and the parametric template of the chair is not detailed enough. However, our evaluation shows that the the global feature based descriptors are not discriminant enough in this case. One reason for this circumstance is that the PSB Database contains many objects which are similar to dining chairs (tables, beds, desk chairs, stools, etc.). Hence, the global feature based descriptors can hardly distinguish these. The hybrid descriptor PANORAMA works perfectly well with the parametric template we have chosen.

As we only evaluated the template of a chair we also have to investigate how much the retrieval gain changes with other classes and how the retrieval changes with more detailed templates. Probably different descriptors are better depending on the template. A very detailed template could probably work better with local feature (bag-of-word) based descriptors.

The obvious disadvantage within our approach is that the parametric templates are currently created manually. We are considering automatic or semi-automatic methods for the template creation to enable a more easy way of including parametric templates.

In sum, generative modeling and the parametric templates are a very powerful tool but hold many unsolved problems concerning their definition towards an easy retrieval.

6. Conclusion and Future Work

In this work we have defined a parametric template spanning an object space and representing an object domain "dining chair", in a way, that it can be easily used for 3D object retrieval. We have combined the parametric template with different descriptors and evaluated our approach with the Princeton Shape Benchmark, showing that the retrieval with a parametric template enhances the retrieval performance.

The definition of the parametric template and the limitation of the multidimensional object space it represents can be difficult. We will investigate possibilities of automatically or semi-automatically retrieving new parametric templates from known objects (i.e., inverse generative modeling). We also consider different ways of sampling an object space of a parametric model as these can be very high dimensional. The sampling could also happen during the retrieval process, dependent on the current retrieved object.

References

[AKZM14] Averkiou M., Kim V. G., Zheng Y., Mitra N. J.: Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 125–134. 3

[ASYS09] Akgül C. B., Sankur B., Yemez Y., Schmitt F.: 3d model retrieval using probability density-based shape descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 6 (2009), 1117–1133. 2

[ASYS10] Akgül C. B., Sankur B., Yemez Y., Schmitt F.: Similarity learning for 3d object retrieval using relevance feedback and risk minimization. *International Journal of Computer Vision 89*, 2-3 (2010), 392–407. 2

[BBGO11] Bronstein A. M., Bronstein M. M., Guibas L. J., Ovsjanikov M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG) 30*, 1 (2011), 1. 2

[BFH05] Berndt R., Fellner D. W., Havemann S.: Generative 3d models: a key to more information within less bandwidth at higher quality. In *Proceedings of the tenth international conference on 3D Web technology* (2005), ACM, pp. 111–121. 3, 4

[BGM*07] Biasotti S., Giorgi D., Marini S., Spagnuolo M., Falcidieno B.: 3d classification via structural prototypes. In *Semantic Multimedia*. Springer, 2007, pp. 140–143. 1, 3

[BWS10] Bokeloh M., Wand M., Seidel H.-P.: A connection between partial symmetry and inverse procedural modeling. In *ACM Transactions on Graphics (TOG)* (2010), vol. 29, ACM, p. 104. 3

[CTSO03] Chen D.-Y., Tian X.-P., Shen Y.-T., Ouhyoung M.: On visual similarity based 3d model retrieval. In *Computer* *graphics forum* (2003), vol. 22, Wiley Online Library, pp. 223–232. 2

[DK12] DAROM T., KELLER Y.: Scale-invariant features for 3-d mesh models. *Image Processing, IEEE Transactions on 21*, 5 (2012), 2758–2769. 2

[FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A search engine for 3d models. *ACM Transactions on Graphics (TOG) 22*, 1 (2003), 83–105. 1

[FRS*12] FISHER M., RITCHIE D., SAVVA M., FUNKHOUSER T., HANRAHAN P.: Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 135. 3

[GWJ*14] GAO Y., WANG M., JI R., WU X., DAI Q.: 3-d object retrieval with hausdorff distance learning. *Industrial Electronics, IEEE Transactions on 61*, 4 (2014), 2088–2098. 2

[HF05] HAVEMANN S., FELLNER D. W.: *Generative mesh modeling.* PhD thesis, University of Braunschweig-Institute of Technology, 2005. 4

[HLR05] HOU S., LOU K., RAMANI K.: Svm-based semantic clustering and retrieval of a 3d model database. *Computer-Aided Design and Applications 2*, 1-4 (2005), 155–164. 2

[HMVG09] HAEGLER S., MÜLLER P., VAN GOOL L.: Procedural modeling for digital cultural heritage. *Journal on Image and Video Processing 2009* (2009), 7. 3

[JTRS12] JAIN A., THORMÄHLEN T., RITSCHEL T., SEIDEL H.-P.: Exploring shape variations by 3d-model decomposition and part-based recombination. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 631–640. 3

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing* (2003), vol. 6. 2

[KLM*13] KIM V. G., LI W., MITRA N. J., CHAUDHURI S., DIVERDI S., FUNKHOUSER T.: Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 70. 3

[KPW*10] KNOPP J., PRASAD M., WILLEMS G., TIMOFTE R., VAN GOOL L.: Hough transform and 3d surf for robust three dimensional classification. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 589–602. 2

[LJ13] LI B., JOHAN H.: 3d model retrieval using hybrid features and class information. *Multimedia tools and applications 62*, 3 (2013), 821–846. 1, 2

[LMT05] LEIFMAN G., MEIR R., TAL A.: Semantic-oriented 3d shape retrieval using relevance feedback. *The Visual Computer 21*, 8-10 (2005), 865–875. 2

[LZYX15] LENG B., ZENG J., YAO M., XIONG Z.: 3d object retrieval with multi-topic model combining relevance feedback and lda model. In *Image Processing, IEEE Transactions on* (2015), IEEE, pp. 94–105. 2

[MFK*10] MAES C., FABRY T., KEUSTERMANS J., SMEETS D., SUETENS P., VANDERMEULEN D.: Feature detection on 3d face surfaces for pose normalisation and recognition. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on* (2010), IEEE, pp. 1–6. 2

[MPB05] MARVIE J.-E., PERRET J., BOUATOUCH K.: The fl-system: a functional l-system for procedural geometric modeling. *The Visual Computer 21*, 5 (2005), 329–339. 3

[NK03] NOVOTNI M., KLEIN R.: 3d zernike descriptors for content based shape retrieval. In *Proceedings of the eighth ACM symposium on Solid modeling and applications* (2003), ACM, pp. 216–225. 2, 6

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. *ACM Transactions on Graphics (TOG) 21*, 4 (2002), 807–832. 2, 6

[OLGM11] OVSJANIKOV M., LI W., GUIBAS L., MITRA N. J.: Exploration of continuous variability in collections of 3d shapes. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 33. 3

[PIMD08] PATTERSON IV A., MORDOHAI P., DANIILIDIS K.: Object detection from large-scale 3d datasets using bottom-up and top-down descriptors. In *Computer Vision–ECCV 2008*. Springer, 2008, pp. 553–566. 2

[PPTP10] PAPADAKIS P., PRATIKAKIS I., THEOHARIS T., PERANTONIS S.: Panorama: A 3d shape descriptor based on panoramic views for unsupervised 3d object retrieval. *International Journal of Computer Vision 89*, 2-3 (2010), 177–192. 2, 6

[RA99] RAMAMOORTHI R., ARVO J.: Creating generative models from range images. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 195–204. 3

[SJWS13] SUNKEL M., JANSEN S., WAND M., SEIDEL H.-P.: A correlated parts model for object detection in large 3d scans. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 205–214. 2

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Shape modeling applications, 2004. Proceedings* (2004), IEEE, pp. 167–178. 5

[SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S.: Skeleton based shape matching and retrieval. In *Shape Modeling International, 2003* (2003), IEEE, pp. 130–139. 2

[SWS10] SCHERER M., WALTER M., SCHRECK T.: Histograms of oriented gradients for 3d object retrieval. In *WSCG 2010, 18th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision. Full Papers Proceeding* (2010), pp. 41–48. 2, 6

[TCF09] TOLDO R., CASTELLANI U., FUSIELLO A.: A bag of words approach for 3d object categorization. In *Computer Vision/Computer Graphics CollaborationTechniques*. Springer, 2009, pp. 116–127. 2

[TV08] TANGELDER J. W., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. *Multimedia tools and applications 39*, 3 (2008), 441–471. 2

[UF11] ULLRICH T., FELLNER D. W.: Generative object definition and semantic recognition. In *Proceedings of the 4th Eurographics conference on 3D Object Retrieval* (2011), Eurographics Association, pp. 1–8. 3

[VKTS*11] VAN KAICK O., TAGLIASACCHI A., SIDI O., ZHANG H., COHEN-OR D., WOLF L., HAMARNEH G.: Prior knowledge for part correspondence. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 553–562. 2

[Vra05] VRANIC D. V.: Desire: a composite 3d-shape descriptor. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on* (2005), IEEE, pp. 4–pp. 2

[XL07] XU D., LI H.: 3d shape retrieval integrated with classification information. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on* (2007), IEEE, pp. 774–779. 1, 2

[XLZ*10] XU K., LI H., ZHANG H., COHEN-OR D., XIONG Y., CHENG Z.-Q.: Style-content separation by anisotropic part scales. In *ACM Transactions on Graphics (TOG)* (2010), vol. 29, ACM, p. 184. 3