Aggregating Viewpoints for Effective View-Based 3D Model Retrieval

Sou Watanabe*, Shigeo Takahashi*, and Luobin Wang*
*University of Aizu, Aizu-Wakamatsu, Japan

***Abstract*—The *bag-of-features* (*BoF*) model is the standard platform for image retrieval systems and successfully extended to systems for exploring 3D models through their projected views. However, we need a large number of views for each 3D model to achieve shape retrieval systems with high accuracy, which results in increased data storage and long computation time for shape comparison. This paper presents an approach for reducing projected images in such image-based shape retrieval by aggregating views of each 3D model. Our approach begins by discovering a proper metric for evaluating dissimilarity between 3D models by referring to their high-dimensional feature vectors obtained from the BoF model. We then introduce a variant of the k-means clustering method to identify the representative views of each 3D model, given the number of such essential views. Finally, we adjust the degree of such view aggregation by assessing the number of plane symmetries for each 3D model. We test our approach with a dataset containing 200 3D models and we learn that we can reduce the number of views to less than 10% while limiting the degradation of accuracy to approximately 5%.**

***Index Terms*—View-based 3D model retrieval, bag-of-features model, similarity measures, viewpoint aggregation, plane symmetries**

I. Introduction

Recent improvements in hardware have led to the emergence of computer graphics technologies that play an increasingly crucial role in synthesizing highly realistic and complicated 3D scenes. For that purpose, skill in modeling geometric shapes of 3D objects is essential to improving the visual quality of synthesized 3D scenes. This means that designing visually appealing 3D models directly influences the attractiveness of visual media, including graphics, animation files, video games, and advertisements.

Conventionally, 3D models were manually designed through trial and error using modeling software. However, commercially available cloud storage services facilitate shape databases consisting of a large number of ready-made and ready-to-use 3D models. This implies that properly retrieving requisite 3D models from such shape databases is an important technical concern in the computer graphics community. In practice, image retrieval techniques were successfully extended to implement effective shape retrieval systems [1].

Here, the *bag-of-features* (*BoF*) model has been successfully employed as the basis for finding 3D models similar to the input key. The 3D model search query is accomplished by first extracting image features from views of each 3D model projected from multiple viewpoints and then plotting the corresponding feature vectors in high-dimensional feature space. Each view is then vector quantized by decomposing the set of feature vectors into a specific number of clusters and then transformed to the histogram coordinates by counting the occurrence of the feature vectors in the respective clusters. This image-based retrieval technique also successfully facilitates an effective search query for a large set of 3D models that are ready to use.

However, this query system for 3D models often results in large data storage since a single 3D model requires multiple projected views to achieve high accuracy in retrieval performance. This is unlike an image query system in which each image is directly mapped to a single histogram through a one-to-one association mapping. Such large-sized storages also degrade the query performance since we need to investigate histograms obtained from multiple views to assess the similarity of a single 3D model with the input key model.

The objective of this study is to compose a compact representation of an image-based shape retrieval system employing a minimum number of views for each model. Our first task is to find a proper similarity metric between the histogram coordinates acquired from views of 3D models to maximally enhance the quality of similar shape queries. We then apply a variant of k-means clustering using the selected similarity metric to find a set of representative views of each 3D model. Finally, we detect significant plane symmetries to adjust the number of clusters to adaptively reduce the total number of projected views maintained in the image-based shape database. Our experiment demonstrates that we can successfully reduce the number of projected views to 10% of the original views while limiting the degradation of shape retrieval accuracy by around 5%.

This paper is organized as follows. Section II provides a brief survey of previous studies related to this work. Section III explains the algorithmic flow of our computational framework for retrieving 3D models from their projected views. We detail our primary contribution in Section IV, in which we take advantage of plane symmetries inherent in each 3D model to adaptively aggregate views projected from multiple viewpoints. Section V presents experimental results to demonstrate how we can adaptively minimize the views of 3D models stored in the image-based database system without significantly degrading the quality in shape retrieval. Finally, we present our conclusions and outline possible future work in Section VI.

II. Related Work

This section provides a summary of relevant studies by classifying them into the following three categories: image-based shape retrieval, viewpoint selection, and viewpoint aggregation.

A. Image-Based Shape Retrieval

The *BoF* model [2], [3] allows us to extract the underlying semantics inherent in a set of images. This model is an extended version of the bag-of-words model used in text mining techniques in the sense that the extracted image features are treated as words. For the image features, the *Scale-Invariant Feature Transform* (*SIFT*) [4] was commonly employed to facilitate the process of image categorization. Another possible formulation is *Speeded Up Robust Features* (*SURF*) [5], which has frequently been used as a descriptors of local image features. Although the SURF was partially inspired by the concept of SIFT, it successfully reduced the associated computational cost approximately threefold [6].

The BoF model promoted a variety of practical applications. Gao et al. [7] implemented a system for visually analyzing the bipartite relationships between images and their categories. Ohbuchi et al. [1] developed a shape retrieval system based on the BoF model and demonstrated its feasibility. More recently, multiple modalities including images have been also employed as the keys for shape retrieval, in which deep learning techniques were successfully incorporated [8].

B. Viewpoint Selection

In computer graphics, research has been intensively conducted to select optimal viewpoints for better acquisition of shape characteristics of 3D objects. Kamada and Kawai [9] and Roberts and Marshall [10] explored viewpoints that minimized the invisible areas of the 3D objects in their projected views. Barral et al. [11] modified the approach by Kamada and Kawai to incorporate perspective projections as well. Vázquez et al. [12] proposed viewpoint entropy based on the formulation of Shannon entropy to assess the amount of visual information available in the view projected from the corresponding viewpoint. Later, they developed this formulation to accommodate the concepts of view stability and depth maps of the 3D scenes [13]. More sophisticated approaches based on machine learning techniques [14] and perceptual studies [15] were also introduced to investigate the viewpoint selection problem. Techniques for selecting optimal viewpoints in the context of volume visualization were also investigated [16], [17].

C. Viewpoint Aggregation

It is often important to pursue a plausible set of representative viewpoints through an adaptive aggregation process. In the early stages of the research, Yamauchi et al. [18] proposed several different ideas, including clustering similar views on the viewing sphere. Tulsiani et al. [19] computed a symmetry-aware mapping from pixels to an object-centric canonical 3D coordinate system using *Convolution Neural Network* (*CNN*) to better propagate information over the projected view. Sridhar et al. [20] trained a 2D CNN to predict these representative viewpoints from multiple views of a 3D object. Research on effective viewpoint sampling for image-based 3D model retrieval has also been conducted using an approach different from that of this paper. Li et al. [21] presented a study on C-means-based view clustering from viewpoint entropy values to retrieve relevant 3D models based on hand-drawn sketches. This was done by measuring visual complexity using viewpoint entropy distributions and adaptively determining the number of representative views based on their complexity values.

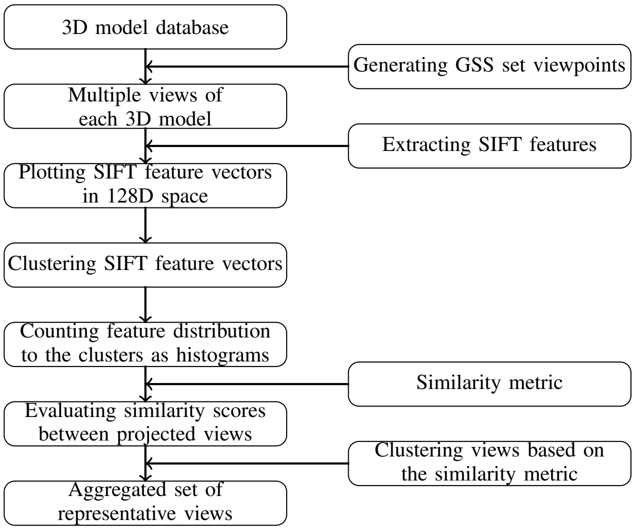

3D model database
Generating GSS set viewpoints
Multiple views of each 3D model
Extracting SIFT features
Plotting SIFT feature vectors in 128D space
Clustering SIFT feature vectors
Counting feature distribution to the clusters as histograms
Similarity metric
Evaluating similarity scores between projected views
Clustering views based on the similarity metric
Aggregated set of representative views

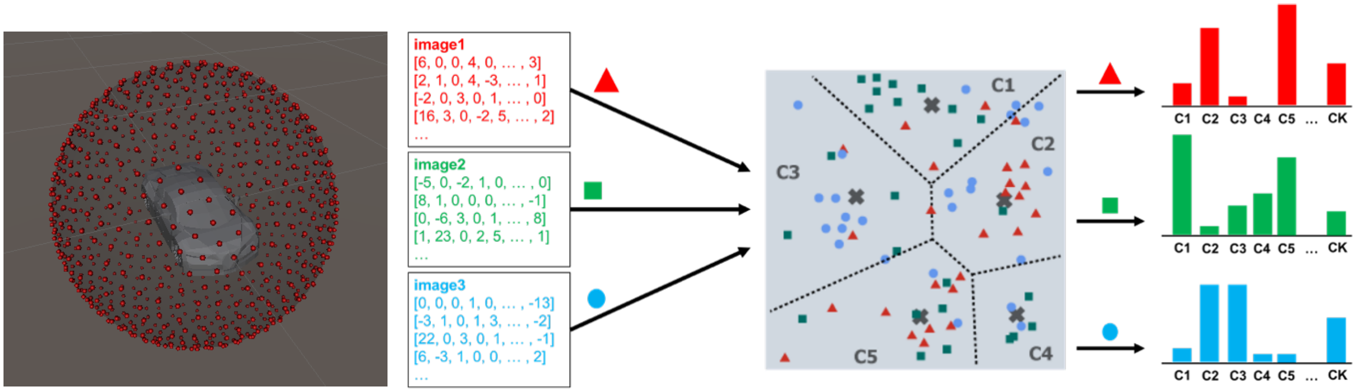
Fig. 1: Flow of the algorithm.

III. Algorithms

In this section, we describe the flow of the algorithm for retrieving 3D models through their projected views based on the BoF model (Fig. 1). Although this is a variant of the previous approach developed by Ohbuchi et al. [1], we strive to make the data storage required for the shape database as compact as possible by adaptively aggregating the original viewpoints in this study. In our setup, we first collect 200 3D models as our running example for 3D shape retrieval. We then store multiple views for each model by projecting them from 1,024 viewpoint samples that are uniformly distributed over the viewing sphere around the model. Our prototype retrieval system searches for a set of 3D models similar to a projected view of a 3D model taken as input. This is accomplished by comparing the SIFT features of the input view with those contained in the multiple views of 3D models stored in the database.

A. Sampling Viewpoints Using the Generalized Spiral Set

Our first step is to sample an initial set of viewpoints uniformly over the unit viewing sphere that encloses the 3D model. In this study, we use a *Generalized Spiral Set* (*GSS*) [22] to ensure uniformity in distributing the viewpoints over the viewing sphere. The GSS permits us to locate the



(a) (b)

Fig. 2: Image-based shape retrieval based on the BoF model. (a) Viewpoint position placed on the sphere by the GSS (red points are the viewpoint position). (b) Calculating a histogram for each projected view.

viewpoint samples by referring to the corresponding incidence and azimuth angles in the polar coordinate system. Let us denote the number of viewpoints by N. We can calculate the k-th ($k = 1, \ldots, N$) viewpoint using the GSS, formulated as the following incremental equations:

$$\begin{aligned} \theta_k &= \cos^{-1} h_k, \quad \text{where} \quad h_k = -1 + 2\frac{k-1}{N-1} \quad \text{and} \\ \phi_k &= \phi_{k-1} + \frac{3.6}{\sqrt{N}\sin\theta_k}, \quad \text{where} \quad \phi_1 = 0. \end{aligned}$$

Here, θ and ϕ indicate the incidence and azimuth angles in the polar coordinate system, respectively. Thus, we can calculate the 3D coordinates of the k-th viewpoint as (x_k, y_k, z_k), where $x_k = \sin\theta_k \cos\phi_k$, $y_k = \sin\theta_k \sin\phi_k$, and $z_k = \cos\theta_k$.

As for the choice of N, we first investigate the retrieval accuracy for a sufficiently large number of viewpoints and then learn that sampling $N = 1{,}024$ viewpoints is sufficient to retain high accuracy in the retrieval of 3D models. Fig. 2(a) shows the result of the GSS formulation used to sample viewpoints over the viewing sphere around a 3D model, where the red points represent the positions of sampled viewpoints. In the initial setup, we generate 1,024 views projected from these viewpoints for each 3D model and store the images in our prototype system for 3D shape retrieval. Note that we normalize the size of the input 3D model before projecting it from multiple viewpoints in such a way that the model will fit within the unit viewing sphere.

B. Extracting SIFT Feature Vectors

The next step is to construct the BoF model by taking as input multiple views of the respective 3D models, as described previously. Fig. 2(b) illustrates the entire process of calculating the histogram coordinates for each projected view. The construction process of the BoF model begins by extracting the SIFT features embedded in a projected image of each 3D model. Note that a single SIFT feature is represented as a 128-dimensional feature vector. Thus, we can plot a set of SIFT features extracted from each projected image in a 128-dimensional feature space, where the total number of projected views is 1,024 (viewpoints) multiplied by 200 (3D models).

We distribute all the SIFT feature vectors to a specific number of clusters by introducing the ordinary k-means clustering method. We then create a histogram for each projected view, where each bin represents the frequency of such feature vectors in the respective clusters. Finally, we normalize the histogram so that it represents a unit k-dimensional vector, for later convenience in evaluating the similarity scores between different views. It is crucial to properly select the number of clusters, k, which will be further discussed later.

C. Calculating Dissimilarity Scores

Having obtained histogram coordinates for each projected image, we want to compute the distance, i.e., dissimilarity, between every pair of histograms. Since a histogram can be thought of as a k-dimensional vector, the Euclidean distance metric in the k-dimensional space is the most natural tool for assessing the dissimilarity. However, we have other possible choices that can evaluate the distance between the histograms more faithfully. In practice, we introduced four options in our experiment and observed which option was the best for evaluating the dissimilarity between 3D models through statistical analysis. Suppose that we have two histograms, $\boldsymbol{x}$ and $\boldsymbol{y}$, to be compared, where x_i and y_i represent the i-th coordinate (i.e., the height of the i-th bin) of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The four metrics can be summarized as follows:

1) *Euclidean distance*:
This metric evaluates the dissimilarity between the two histograms $\boldsymbol{x}$ and $\boldsymbol{y}$ as the distance between the corresponding two vectors, $\{x_i\}$ and $\{y_i\}$, as follows:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

2) *Pearson product-moment correlation coefficient*:
This coefficient evaluates the strength of the linear relationship between the histograms $\{x_i\}$ and $\{y_i\}$, where the

two histograms are assumed to represent the sequences of probability values. The coefficient is given by

$$c(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{k}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{k}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{k}(y_i - \overline{y})^2}},$$

where $\overline{x}$ and $\overline{y}$ denote the average values of x_i's and y_i's, respectively. The coefficient ranges from -1 to 1 according to the degree of the positive correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus, the resulting dissimilarity metric can be defined as $d(\boldsymbol{x}, \boldsymbol{y}) = 1 - c(\boldsymbol{x}, \boldsymbol{y})$.

3) *Cosine similarity*:
This score calculates the cosine of the angle spanned by the two vectors $\boldsymbol{x} = \{x_i\}$ and $\boldsymbol{y} = \{y_i\}$. This implies that the score is expressed as

$$c(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{k} x_i y_i}{\sqrt{\sum_{i=1}^{k} {x_i}^2}\sqrt{\sum_{i=1}^{k} {y_i}^2}}$$

This value again ranges from -1 to 1 according to the degree of similarity between the two histograms $\boldsymbol{x}$ and $\boldsymbol{y}$, which implies that the associated dissimilarity metric is $d(\boldsymbol{x}, \boldsymbol{y}) = 1 - c(\boldsymbol{x}, \boldsymbol{y})$.

4) *Weighted Jaccard coefficient*:
The weighted Jaccard coefficient calculates the degree of similarity between two sets by dividing the total cardinality of the products of the two sets by that of their sums [23]. Thus, the coefficient is formulated as

$$c(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{k} \min\{x_i, y_i\}}{\sum_{i=1}^{k} \max\{x_i, y_i\}}$$

This coefficient ranges from 0 to 1 according to the degree of similarity. The corresponding distance metric can be defined as $d(\boldsymbol{x}, \boldsymbol{y}) = 1 - c(\boldsymbol{x}, \boldsymbol{y})$.

We conduct an experiment to assess which metric faithfully measures the similarity between the projected views of 3D models the best. The scenario for comparing the similarity metrics is summarized as follows.

We first classify the 200 3D models in the database into 29 categories according to the shape contents they represent, such as airplanes, cars, ships, humans, and fishes. We then randomly select one projected view out of $N =$ 1,024 views for each of the 3D models to use as a key image for the shape retrieval in the experiment. Using each of the four similarity metrics described above, we evaluate the degree of similarity of each 3D model to the input key by computing the similarity score of its N views in the database. With this approach, we find the top 10 models that have the highest degree of similarity in the database. Once we identify one of the N views as the most similar to the input key, we exclude other $N - 1$ views from our analysis.

In this way, for each 3D model, we can find the 10 most similar models in the shape database. We plotted these most similar models as small dots (in gray) in the similarity matrix, as shown in Fig. 3. Here, the rows and columns of the matrix correspond to the 3D models ordered according to the shape categorization. We then counted the total number of small dots in the submatrices along the diagonal of the entire similarity matrix as the number of retrieved 3D models that are sufficiently similar to the input key. This is because we consider that 3D models in the same category are considered as similar shapes. This implies that the accuracy of the similarity metric increases as the total number of small dots contained in the diagonal submatrices becomes larger. Of course, this total count varies according to the choice of the four similarity metrics (d) and the dimension of the histograms (k) that corresponds to the number of clusters in the feature space we described earlier. Actually, we want to employ the metric and the number of clusters that achieve the highest similarity count in our study.

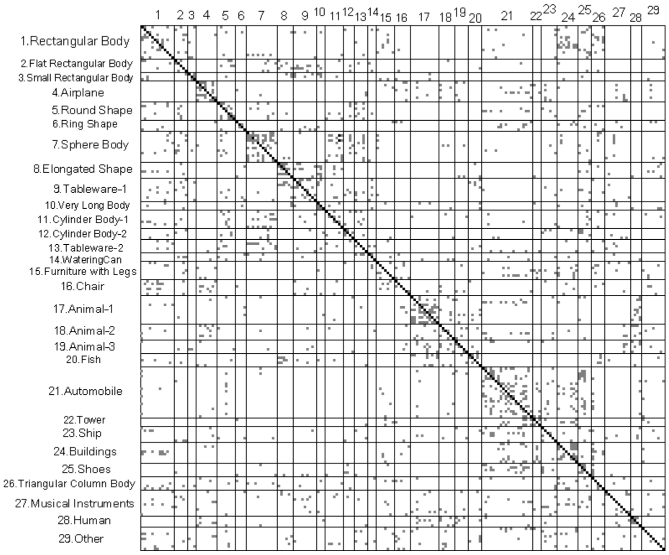


Fig. 3: Similarity matrix obtained using the weighted Jaccard coefficient when $k = 512$.

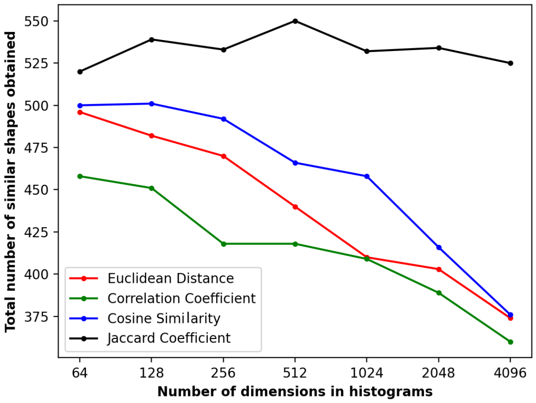
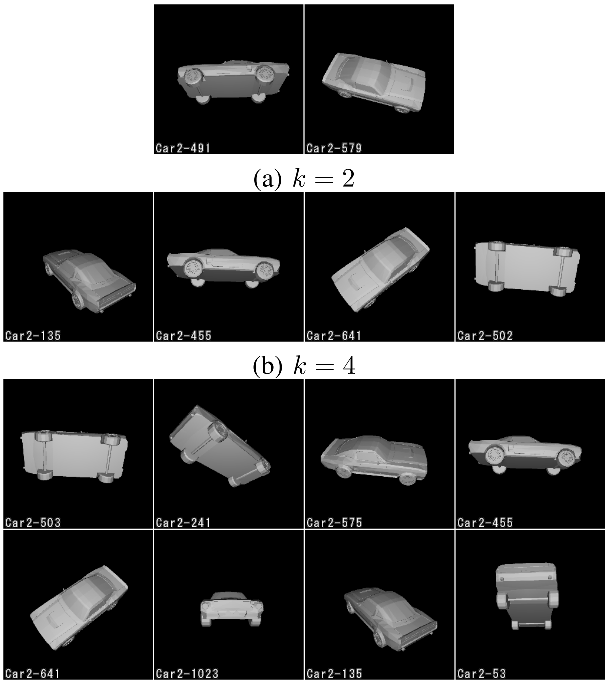


Fig. 4: Comparison between the four similarity metrics in terms of the histogram dimensions.

We can plot the counts of retrieved similar 3D models in terms of the histogram dimension and the similarity count, as exhibited in Fig. 4. The resulting trend suggests that the best metric for seeking similar 3D models is the weighted Jaccard



(a) $k = 2$

(b) $k = 4$

(c) $k = 8$

Fig. 5: Views of a car projected from k representative viewpoints. (a) $k = 2$. (b) $k = 4$. (c) $k = 8$.

coefficient, and its corresponding proper histogram dimension is 512. We can observe that the Jaccard coefficient can keep a relatively stable accuracy regardless of the histogram dimension when compared with other similarity metrics.

D. Identifying the Representative Viewpoints

The last task is to adaptively select representative views from those projected from the original set of 1,024 viewpoints. Suppose that we have a means of computing the proper number of representative views as k for each 3D model. In this case, we can apply the conventional k-means clustering method to the 1,024 histogram coordinates and find the representative viewpoints that correspond to the cluster centers. Nonetheless, we have already decided to use the weighted Jaccard coefficient as our similarity metric, and thus, we cannot precisely identify the center of each cluster as its barycenter since we no longer use the Euclidean distance metric.

For this purpose, we introduce the *k-medoid* clustering method [24], which is a variant of the k-means clustering approach. In practice, the k-medoid clustering method is compatible with the non-Euclidean distance metric as it restricts the position of each cluster center on the data samples that belong to the cluster. In other words, we define a cluster center called a *medoid* to be the sample that minimizes the sum of distances from other samples in the same cluster. The medoid of the i-th cluster can be obtained by the following equation:

$$\mathop{\arg\min}_{\boldsymbol{x} \in G_i} \sum_{\boldsymbol{y} \in G_i - \{\boldsymbol{x}\}} d(\boldsymbol{x}, \boldsymbol{y}),$$

where G_i is the set of data samples in the i-th cluster and d is the distance metric.

In this study, we apply the k-medoid clustering method to the histograms of projected views to aggregate the viewpoints in each cluster to its center, which is expected to serve as the representative viewpoint. In the proposed approach, k-means++ [25] method is adapted to k-medoids clustering to minimize the influence of the initial conditions on the final clustering results.

Fig. 5 presents the selected views projected from the representative viewpoints aggregated using the k-medoid clustering method. As demonstrated in the figure, the set of selected views changes according to the number of clusters, k. Note that the label at the bottom left of each image contains the ID of the viewpoint generated using the GSS [22] formula. Based on the observation, we can claim that the set of views with a small number of viewpoint clusters is likely to survive in the set even when we raise the number of clusters. For example, the viewpoint IDs of a car model with $k = 4$ (Fig. 5(b)) can also be found in the set with $k = 8$ as identical IDs or immediately preceding/following IDs (Fig. 5(c)). This means that properly adjusting the number of viewpoint clusters while retaining high accuracy in shape retrieval will enhance the effectiveness of the image-based shape retrieval system. The next section describes our solution to this problem.

IV. Adjusting Viewpoint Aggregation

This section explores how we can assess the required number of representative viewpoints for generating projected views of 3D models in the image-based shape retrieval system.

A. Accuracy Degradation by Viewpoint Aggregation

As a preliminary experiment, we investigated how the accuracy in shape retrieval degrades as the number of viewpoint samples decreases. For this purpose, we counted the number of similar 3D models successfully retrieved from the database for numbers of viewpoints to the power of 2. We then computed the relative accuracy with respect to the initial number of viewpoints (1,024). Note that in this experiment, we employed the weighted Jaccard coefficient as the similarity metric and set the dimension of the histograms, k, to 512, as described earlier. Furthermore, we obtained the same number of similar 3D models as we did in Section III-C, in which we counted the number of dots in the submatrices along the diagonal of the entire similarity matrix (cf. Fig. 3).

Fig. 6 shows how the relative accuracy ratio changes according to the increase in the number of representative viewpoints to the power of 2. The figure indicates that the relative accuracy ratio does not drastically reduce as we aggregate the set of viewpoints by half. Actually, the accuracy ratio in retrieving similar 3D models decreases by 16%, even when we reduce the number of viewpoints significantly, from 1,024 to 2. This demonstrates the possibility of drastically reducing the number of viewpoints when preparing representative views of each 3D model in the database. However, we still need to assess the acceptable level of reduction in shape retrieval

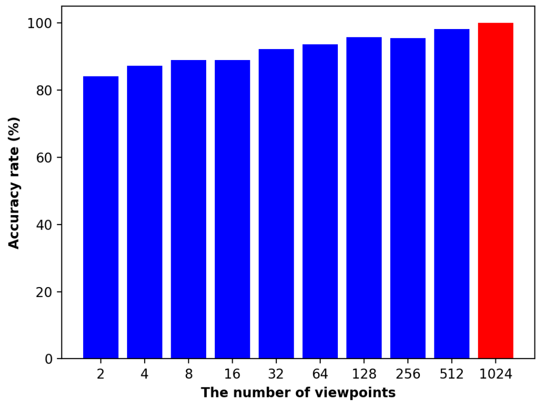


Fig. 6: Number of viewpoints aggregation and overall change in search accuracy.

accuracy when finding the optimal number of representative viewpoints in our approach.

B. Selected Viewpoints and Viewpoint Entropy

One of the commonly used indicators for finding optimal viewpoints is the formulation of *viewpoint entropy* [12]. This indicator facilitates quantifying the visual information of a 3D model projected from the specific viewpoint position and can be formulated as follows:

$$E = -\sum_{i=0}^{M} \frac{A_i}{S} \log_2 \frac{A_i}{S},$$

where M is the number of faces that cover the 3D model, A_i is the visible area of the i-th $(i = 1, 2, \ldots, M)$ face on the 2D screen space, and A_0 corresponds to the area of the scene background. Suppose that S represents the total area of the screen space and is thus computed as $S = A_0 + \sum_{i=1}^{M} A_i$. This is just a variant of the Shannon entropy measure, which means that we can obtain more visual information in the projected view as the viewpoint entropy increases. We can identify the optimal viewpoints by computing the viewpoint entropy values of the views projected from viewpoints sampled with the GSS and identifying those with high entropy values.

For a more detailed investigation, we visualize the relationship between the distribution of the viewpoint entropy values and positions of representative viewpoints over the viewing sphere. To visualize this relationship, we compute the entropy values at the initial set of 1,024 viewpoints and then normalize them into the range $[0.0, 1.0]$ through an affine transformation. This normalization process allows us to consistently assign the colors according to the relative magnitude of the viewpoint entropy value for each viewpoint sample. In our implementation, we plot the viewpoints over the triangulated viewing sphere and render its wireframe representation. Here, the color of the wireframe representation changes from blue to green to red according to the magnitude of the corresponding viewpoint entropy value. To view the positions of representative viewpoints, we overlay each viewpoint as a small sphere on the viewing sphere.

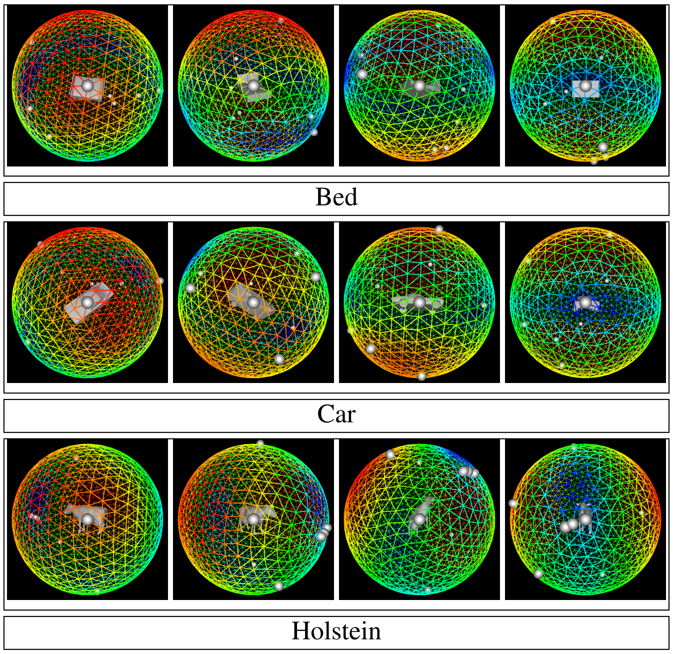


Fig. 7: Comparison between viewpoints entropy distribution and representative viewpoint positions.

Fig. 7 shows color-coded distributions of the normalized viewpoint entropy values for 3D models, together with the representative viewpoints obtained in this study. Here, we set the number of representative viewpoints to be eight. Our observation suggested that the representative viewpoints were likely to stay in the areas of high entropy values. Another interesting fact is, if the 3D model is almost symmetric about a plane, viewpoints in a pair are likely be located antipodal to each other on the viewing sphere. In this case, the distribution of the viewpoint entropy values is also nearly symmetric with respect to the plane. This fact actually inspired us to devise a new method of aggregating viewpoint samples while minimizing the degradation of accuracy in the shape retrieval process.

C. Viewpoint Aggregation Based on Plane Symmetries

The ultimate goal of this study is to adaptively adjust the number of representative viewpoints by referring to the intrinsic shape features of each 3D model. Our idea lies in the incorporation of the plane symmetries of 3D models for such adaptive aggregation of viewpoints. Suppose that a 3D model is nearly symmetric in terms of some plane. In this case, we obtain almost the same views of the 3D model if they are projected from a pair of antipodal viewpoints over the viewing sphere. This implies that we can skip half of the viewpoint samples if the 3D model is almost symmetric about the plane. Thus, we decided to assess the degree of plane symmetry for each 3D model to adaptively aggregate the initial set of viewpoints. Among the many methods currently available for this purpose, we employed one developed by Bo et al. [26], which facilitates the detection of such symmetry planes based on the distribution of viewpoint entropy over the viewing sphere. This method allows us to find a pair of viewpoints with similar distributions of entropy values and identify a symmetry plane if it bisects the line segment connecting the viewpoints.

Our scenario is to adaptively aggregate the viewpoints by counting the number of such symmetry planes for each 3D model. However, this approach is computationally expensive as it requires an exhaustive search for symmetric pairs of viewpoints scattered over the viewing sphere. Consequently, we wanted to accelerate the computation by limiting the number of symmetry planes to be checked for viewpoint aggregation. This consideration leads us to the idea of aligning 3D models along the three principal axes. In this study, we introduce *Continuous Principal Component Analysis* (*CPCA*) [27] for this purpose and restrict our symmetry test to the three planes spanned by the principal axes. This sophistication considerably reduces the computation time that is initially required by the conventional approach for the exhaustive search for symmetry planes. Fig. 8(a) shows an example of the principal axes calculated by CPCA, and Figs. 8(b), (c) and (d) exhibit several 3D models with symmetry planes detected using this approach. Below, we demonstrate how the proposed approach can successfully aggregate the initial set of viewpoints while retaining high accuracy in retrieving similar 3D models in the shape database.

V. Experimental Results

We have implemented our prototype shape retrieval system on a laptop PC (MacBook Pro) with an Intel Core i5 processor with two cores (2.7 GHz), 8GB RAM, and an Intel Iris Graphics 6100 GPU (1536MB VRAM). The source code has been written in C++, OpenGL for rendering 3D models, and OpenCV for image feature extraction. Basically, it takes more time to retrieve similar 3D models as the number of projected views stored in the database increases. Our simple statistical analysis, together with linear regression, shows that the retrieval time is approximately proportional to the number of projected views to be compared.

In our experiments, we tested two approaches for viewpoint aggregation, the exhaustive plane symmetry search formulated by Bo et al. [26] and our restricted search with CPCA [27]. In both cases, we adaptively reduced the number of viewpoints by counting symmetry planes for each 3D model. Suppose that we start with an initial number of viewpoints, V and reduce the number by half once we can find a single symmetry plane. If we can extract the second symmetry plane, we can further reduce the number to a quarter of the initial value. However, we never lower the number of viewpoints, even when we have three or more symmetry planes in total, to avoid unexpected degradation in the accuracy of similar shape retrieval. We tested the two symmetry-based strategies for $V = 8, 128,$ and 512 and investigated how we can retain the accuracy in the search for the similar 3D models, as well as maintain the required time for shape retrieval. Table I shows the statistics resulting from our experiments on these two symmetry-based aggregations of viewpoint samples.

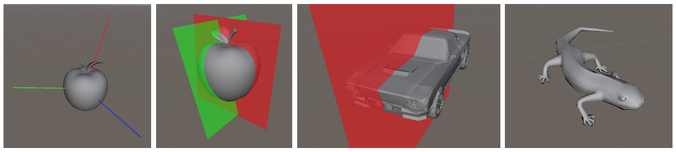

(a) (b) (c) (d)

Fig. 8: Symmetry planes detected using the algorithm by Bo et al. [26] and CPCA [27]. (a) Principal component axes calculated by CPCA. (b) Two symmetry planes. (c) One symmetry plane. (d) No symmetry planes.

The results in Table I demonstrate that with the exhaustive plane symmetry search, we could further reduce the number of viewpoints by almost 30% with little loss in the shape retrieval accuracy. This implies that the adaptive selection of viewpoint samples based on plane symmetries can effectively reduce the data size in image-based 3D shape retrieval systems.

Nonetheless, this simple approach for symmetry detection requires an exhaustive search for pairs of viewpoints over the viewpoint sphere and thus, results in a large amount of computation time. On the other hand, as described previously, our accelerated approach, based on CPCA [27], is effective since it aligns a 3D model to the principal three axes first and then tests the symmetry only in terms of three planes spanned by the principal axes. In our experiments, the conventional exhaustive search took 500.99 seconds to detect symmetry planes for each model, while the CPCA-based approach only required 0.698 seconds. This means that we accelerated the computation by a factor of approximately 700. Furthermore, the reduction rate of viewpoints has been further improved, as demonstrated in Table I. Simultaneously, this shape alignment enhancement again effectively suppresses the loss in the accuracy of the shape retrieval. This allows us to conclude that our accelerated symmetry search succeeded in selecting an effective number of viewpoints in shape retrieval and also reduces the required time for shape retrieval.

VI. Conclusion and Future Work

In computer graphics, the selection of good viewpoints has been an important area of study for many years. In this paper, these technical problems have been tackled in the context of image-based shape retrieval, in which 2D views of 3D models projected from multiple viewpoints are stored as images for finding similar 3D models. Our technical challenge includes effective adjustment of the number of projected images stored in the database to implement compact and accelerated shape retrieval systems. For this purpose, we try to aggregate the set of viewpoints using the k-medoids clustering while referring to a non-Euclidean distance metric.

In this study, we obtained several new findings. First, the Jaccard coefficient is the best similarity metric among several candidates, especially for retrieving similar shapes via image analysis based on the BoF model. Second, the accuracy in retrieving similar 3D models can be kept relatively high even when we drastically decrease the number of views through

TABLE I: Symmetry-based aggregation of viewpoints, shape retrieval accuracy, and computation times. O is the original number of viewpoints that produces projected views of 3D models (i.e., O = 1,024 × 200 = 204,800). V: selected number of viewpoints for each model before viewpoint aggregation. A: selected number of projected views (i.e., $A = N \times 200$). B: accuracy rate in the case of N viewpoints compared to the 1024 ones. C: total number of viewpoints after adaptive viewpoint aggregation. D: accuracy rate compared to the case before viewpoint aggregation (compared to the rate with N viewpoints.) T: time required for retrieving similar objects (in msec).

				Symmetry detection only					Symmetry detection + CPCA				
N	A	A/O	B	C	C/O	D	$B \times D$	T	C	C/O	D	$B \times D$	T
8	1,600	0.78%	89.0%	1,178	0.58%	99.2%	88.2%	12.64	1,010	0.49%	97.7%	86.9%	7.53
128	25,600	12.5%	95.8%	18,188	8.89%	96.9%	92.8%	197.42	16,160	7.89%	98.3%	94.2%	132.51
512	102,400	50.5%	98.2%	72,688	35.49%	99.8%	98.0%	789.45	64,640	31.5%	99.8%	98.0%	727.30

viewpoint aggregation. The last and most important contribution lies in our new approach for adaptively reducing the number of viewpoints in the context of image-based shape retrieval, which is based on the plane symmetries of 3D models. By referring to the distribution of the viewpoint entropy values over the viewing sphere, we succeed in detecting the plane symmetries of 3D models, while retaining the accuracy in retrieving similar shapes even with a reduced number of viewpoints. We can reduce the number of views to less than 10% while limiting the degradation of accuracy to approximately 5%, especially when starting with 128 viewpoints for each 3D model as an initial set of samples over the viewing sphere.

Our future work is to justify these new findings quantitatively through additional experiments. In particular, we want to explore the meaningful relationship between the geometric features inherent in the 3D models and the number of viewpoints required for producing projected views. Exploring useful shape features other than symmetry through deep learning techniques is a future research theme.

References

[1] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno, "Salient local visual features for shape-based 3D model retrieval," in *Proc. Int. Conf. Shape Modeling and Applications (SMI2008)*, 2008, pp. 93–102.

[2] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Proc. 9th IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1470–1477.

[3] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. V. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop on Statistical Learning in Computer Vision*, 2004, pp. 59–74.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[6] P. Panchal, S. R. Panchal, and S. Shah, "A comparison of sift and surf," *Int. J. Innovative Research in Computer and Communication Engineering*, vol. 1, pp. 323–327, 2013.

[7] G. Yi, H.-Y. Wu, K. Misue, K. Mizuno, and S. Takahashi, "Visualizing bag-of-features image categorization using anchored maps," in *Proc. 7th Int. Symp. Visual Information Communication and Interaction (VINCI'14)*, 2014, pp. 39–48.

[8] H. Tabia and H. Laga, "Learning shape retrieval from different modalities," *Neurocomputing*, vol. 253, pp. 24–33, 2017.

[9] T. Kamada and S. Kawai, "A simple method for computing general position in displaying three-dimensional objects," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 1, pp. 43–56, 1998.

[10] D. R. Roberts and A. D. Marshall, "Viewpoint selection for complete surface coverage of three dimensional objects," in *Proc. Britsh Machine Vision Conference*, 1998, pp. 740–750.

[11] P. Barral, G. Dorme, and D. Plemenos, "Scene understanding techniques using a virtual camera," in *Proceedings of Eurographics 2000 - Short Presentations*, 2000, pp. 20–25.

[12] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Viewpoint selection using viewpoint entropy," in *Proc. Vision, Modeling, and Visualization (VMV) Conference*, 2001, pp. 273–280.

[13] P.-P. Vázquez, "Automatic view selection through depth-based view stability analysis," *The Visual Computer*, vol. 25, pp. 441–449, 2009.

[14] T. Vieira, A. Bordignon, A. Peixoto, G. Tavares, H. Lopes, L. Velho, and T. Lewiner, "Learning good views through intelligent galleries," *Computer Graphics Forum*, vol. 28, no. 2, pp. 717–726, 2009.

[15] A. Secord, J. Lu, A. Finkelstein, M. Singh, and A. Nealen, "Perceptual models of viewpoint preference," *ACM Transactions on Graphics*, vol. 30, no. 5, 2011.

[16] U. D. Bordoloi and H.-W. Shen, "View selection for volume rendering," in *Proc. IEEE Visualization*, 2005, pp. 487–494.

[17] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita, "A feature-driven approach to locating optimal viewpoints for volume visualization," in *Proc. IEEE Visualization*, 2005, pp. 495–502.

[18] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. Belyaev, and H.-P. Seidel, "Towards stable and salient multi-view representation of 3D shapes," in *Proc. IEEE Int. Conf. on Shape Modeling and Applications 2006 (SMI'06)*, 2006, pp. 40–40.

[19] S. Tulsiani, O. Litany, C. R. Qi, H. Wang, and L. J. Guibas, "Object-centric multi-view aggregation," 2020.

[20] S. Sridhar, D. Rempe, J. Valentin, B. Sofien, and L. J. Guibas, "Multiview aggregation for learning category-specific shape reconstruction," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] B. Li, Y. Lu, and H. Johan, "Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering," in *Proc. Eurographics Workshop on 3D Object Retrieval*, 2013, pp. 49–56.

[22] A. Yamaji, "Gss generator: A software to distribute many points with equal intervals on an unit sphere," *Geoinformatics (Joho Chishitsu)*, vol. 12, no. 1, pp. 3–12, 2001, (in Japanese).

[23] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*. Addison-Wesley, 2005.

[24] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3336–3341, 2009.

[25] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding." in *SODA*, N. Bansal, K. Pruhs, and C. Stein, Eds. SIAM, 2007, pp. 1027–1035.

[26] B. Li, H. Johan, Y. Ye, and Y. Lu, "Efficient 3D reflection symmetry detection: A view-based approach," in *Graphical Models*, vol. 83, 2016, pp. 2–14.

[27] D. V. Vranic, "3D model retrieval," Ph.D. dissertation, University of Leipzig, 2004.