Ryutarou Ohbuchi and Takahiko Furuya, Scale-Weighted Dense Bag of Visual Features for 3D Model Retrieval from a Partial View 3D Model, *accepted*, *Proc. ICCV 2009 workshop on Search in 3D and Video* (*S3DV*) *2009*, Sept. 27, Kyoto, Japan.

Scale-Weighted Dense Bag of Visual Features for 3D Model Retrieval from a Partial View 3D Model

Ryutarou Ohbuchi
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi-ken, 400-8511, Japan
ohbuchiAT yamanashi.ac.jp

Takahiko Furuya
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi-ken, 400-8511, Japan
snc49925AT gmail.com

Abstract

This paper describes a 3D shape model retrieval method that accepts, as a query, a 3D mesh obtained by a range scan from a viewpoint. The proposed method visually compares single depth map of the query with depth maps of a 3D model rendered from multiple viewpoints. Comparison of the depth maps employs bag-of local visual features extracted by using a modified version of Lowe's Scale-Invariant Feature Transform (*SIFT*)*. The method is capable of retrieving 3D models having diverse shape representations and is robust against articulation and global deformation of 3D shapes thanks to location-free integration of local visual features. Two modifications to the SIFT are made to avoid ill effects of range scanning artifacts, such as jagged edges and cracks, that exist in the query mesh. The two modifications are; (1) dense and random feature placement, and (2) importance sampling of low-frequency images in the SIFT's Gaussian image pyramid. Our experimental evaluation showed that the proposed method significantly outperforms previous methods.*

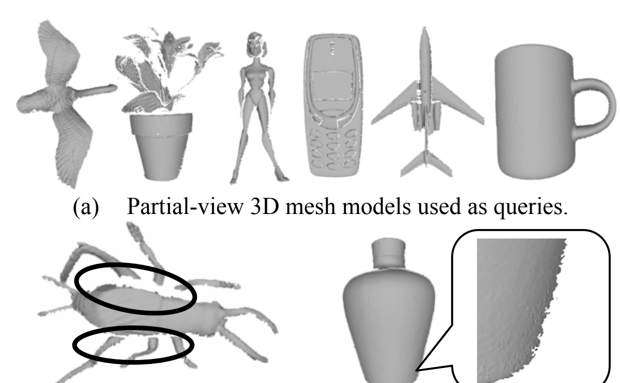
1. Introduction

Interest in content based 3D model retrieval has been increasing recently in the recent years due to popularity of 3D models used in game, movie special effect, mechanical design, medicine, architecture, and many other applications [6, 9]. Most of existing 3D model retrieval methods accepts a 3D model and returns a set of 3D models ranked by their similarity to the query. In some of the applications, however, the query is not a complete 3D model, but a range scan of a 3D object from a view (or from a few views). Examples of partial view retrieval include navigation of a robot using Laser range scanners and retrieval of archaeological artefacts. This paper proposes and evaluates a method for retrieving 3D objects from a partial view 3D mesh model obtained from a single range scan.

An international 3D model retrieval contest SHape REtrieval Contest was initiated in 2006, and its 2009 edition included, for the first time, a track on 3D model retrieval by using a partial 3D model generated from a viewpoint as a query. Figure 1 shows examples of the queries for the track. The track is organized by Afzal Godil of the National Institute of Standard Technology [1]. Two methods, the *Compact Multi-View Descriptor* (*CMVD*) by Axenopoulos and Daras [1, 4] and our *Bag-of-Features SIFT* (*BF-SIFT*) [1, 8], participated in the sub-track. Both methods are appearance-based, so that 3D shapes are compared based on the 2D renderings, e.g., binary and/or depth images, of the 3D models. View based methods (e.g., [3], [4], and [8]) have an advantage of being applicable to a diverse class of 3D shape representations, be it solid, manifold mesh, point set, or polygon soup. So far as a model can be rendered, its shape can be compared. In comparison, other 3D shape comparison methods often require the models to be defined by a specific 3D shape representation.

The result of the SHREC 2009 Partial View Retrieval Track showed that the best performing variations of the CMVD and the BF-SIFT virtually tied in various performance scores. Favourable classes are also split in half, each bettering the other on 10 out of 20 queries.

To compare between 3D models, the CMVD method by



(a) Partial-view 3D mesh models used as queries.

(b) Cracks. (c) Jagged edges.

Figure 1: Examples of the SHREC 2009 Partial Model Retrieval Track query models generated by Laser range scan (a). Cracks and jagged edges in query models ((b) and (c)).

Axenopoulos and Daras [1, 4] renders, after normalizing for size and position, a set of multi-view 2D images of a 3D model from a set of viewpoints located uniformly in the solid angle. For each view, the CMVD renders both binary and depth images. From each 2D image, the method computes three rotation invariant global 2D image descriptors, 2D Polar Fourier Transform, 2D Zernike Moments, and 2D Krawtchouk Moments. To compare a single view 3D model of the query with full 3D models, the method treat as if the binary and depth images of the query model is one of the many views of a full 3D model.

The BF-SIFT method used for the contest is a variant of our previous view-based method for full 3D model to full 3D model comparison with the same name [8]. The original BF-SIFT renders, after normalizing for size and position, a set of depth images of a 3D model from 42 viewpoints distributed uniformly in solid angle. The BF-SIFT then computes, for each depth image, a set of multi-scale, local 2D image descriptors by using Lowe's *Scale Invariant Feature Transform* (*SIFT*) [7]. The SIFT first finds a few dozen *interest points* in its multiscale image pyramid, and then computes 128D features with scale and orientation information for each of the interest point. About a thousand SIFT features from 42 views per 3D model are integrated into a feature vector for the 3D model by using *bag-of-features* approach [2], through vector quantization into "*visual words*" of SIFT features followed by accumulation of the visual words into a histogram. A histogram per 3D model then becomes a feature vector for the model. An advantage of the BF-SIFT is its robustness against articulation and global deformation of 3D models to be compared, due to the use of local features and their location-free integration into a feature vector per 3D model by using the bag-of-features approach.

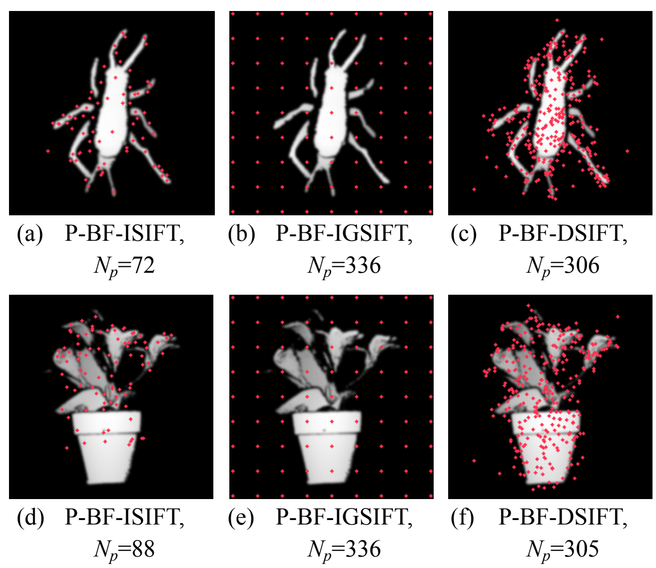
We entered the SHREC 2009 Partial View Retrieval track [1] with two variations of the original BF-SIFT. The first is the original BF-SIFT with its distance computation step modified. We call the method *Partial view Bag-of-Features Interest point SIFT* (*P-BF-ISIFT*) in this paper. The P-BF-ISIFT has 42 feature vectors per full 3D model, corresponding to 42 depth images. The 42 feature vectors are compared against a feature vector computed from the depth image of the single-view query 3D model. The comparison of a full 3D model with a partial view model thus requires 42 distance computations.

The second variation is created in an attempt to improve the retrieval performance of the P-BF-ISIFT above. The variation combines grid sampling and interest point sampling of local visual features; high frequency images are sampled using the original SIFT with its interest point detector while low frequency images are sampled at regular grid points. We call this variation *Partial view Bag-of-Feature Interest point and Grid sampling SIFT* (*P-BF-IGSIFT*) in this paper. Two reasons for the modification are; (1) to avoid effects of unwanted feature points at cracks and jagged edges found in range scanned 3D models, and (2) to increase number of samples in a depth image of a view. Experiments showed that the hybrid sampling P-BF-IGSIFT performed better than the interest point based P-BF-ISIFT.

Figure 2a and 2d show interest points of the P-BF-ISIFT, which are attracted to high frequency features such as the attachments of legs to the insect's body, or to the leaves of the potted plant. However, the body of the insect or the pot attracted only small number of interest points. Figure 2b and 2e show feature points on grid point of the P-BF-IGSIFT for low frequency images. Body and legs of the insect, or the pot and leaves of the potted plant are sampled more evenly in the grid sampling.

In this paper, we propose a method for partial-view to full 3D model comparison based also on our original BF-SIFT [8]. The method is multi-view based, and employs bag-of local visual features, which makes it robust against articulation and global deformation of 3D shapes. However, unlike the P-BF-ISIFT or P-BF-IGSIFT, the proposed method uses *dense and random placement* of a large number of SIFT features. This increases the number of features per view and avoids ill effects of falsely detected (e.g., at the cracks) interest points. Furthermore, the method *emphasizes low-frequency* or *larger scale features* in the images by placing more samples in low frequency images in the Gaussian image pyramid of the SIFT. This enables the method to suppress high-frequency noise introduced by range scan, and to capture more global shape features. We call the method *Partial view Bag-of-Features Dense SIFT*, or *P-BF-DSIFT* in this paper.

Our experimental evaluation using the SHREC 2009



(a) P-BF-ISIFT, N_p=72 (b) P-BF-IGSIFT, N_p=336 (c) P-BF-DSIFT, N_p=306

(d) P-BF-ISIFT, N_p=88 (e) P-BF-IGSIFT, N_p=336 (f) P-BF-DSIFT, N_p=305

Figure 2: SIFT sample points of the original P-BF-SIFT (a)(d), P-BF-IGSIFT for low frequency images (b)(e), and our proposed P-BF-DSIFT-L (c)(f). N_p indicates the number of SIFT features per image.

Partial Models Retrieval benchmark showed that our method significantly outperforms both the CMVD and our previous method P-BF-IGSIFT. Using the SHREC 2009 Partial View Retrieval benchmark, our proposed method scored 37% in Mean First Tier. In comparison, both CMVD and P-BF-IGSIFT scored 22% in Mean First Tier in the same benchmark.

2. Method

Figure 3 shows the processing pipeline of the proposed P-BF-DSIFT method. The process starts with depth-image rendering; from 1 view for the partial view query 3D model and from N_i views for the full 3D model. The method then randomly and densely places sample points, and extract SIFT feature at each sample point. This gives a bag of SIFT features *per view*, that is, *per depth image*. For each view, the SIFT features are vector quantized using pre-learned codebook, and accumulated into a histogram for the view. Finally, a histogram for the partial view query and a set of N_i multiple histograms for the 3D model is compared, producing N_i distances. The minimum of the N_i distances becomes the distance between the query and the 3D model. We will explain the detail of each step in the following.

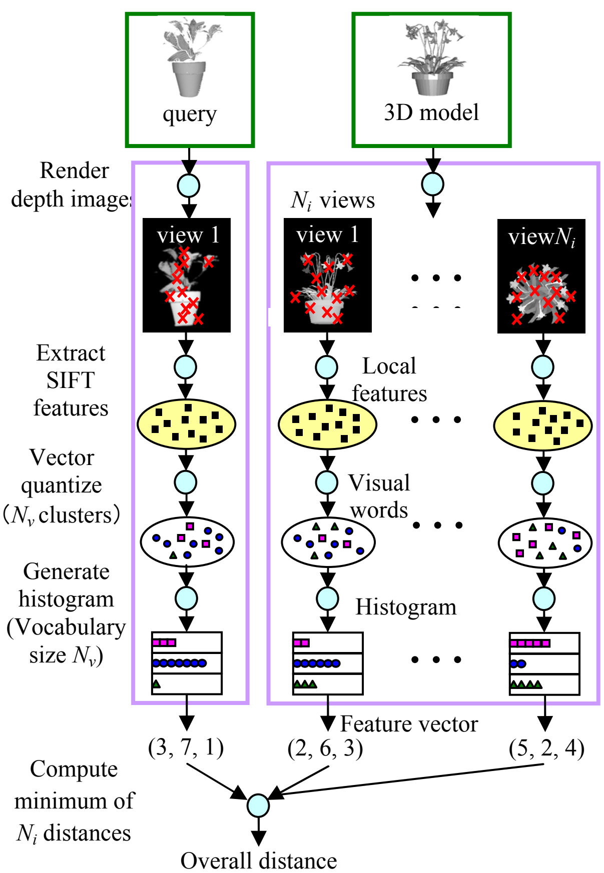


Figure 3: Overview of the proposed *Partial view Bag-of-Feature Dense SIFT* (*P-BF-DSIFT*) method.

2.1. Multi-View Depth Image Rendering

For a full 3D model, after normalizing for position and scale, the method renders the model from N_i viewpoints uniformly spaced in solid angle into N_i depth images per model. The method achieves approximate rotation invariance against two out of three rotational degrees of freedom via this multi-view rendering. Invariance to remaining one rotational degree of freedom is achieved via the rotation invariance of the SIFT. We will experimentally evaluate the effect of N_i on retrieval performance. For the experiment described in this paper, image size of 256×256 pixels is used to render a full 3D model.

For a partial view 3D model, we render the model into an image of size 1024×1024 pixels, which is then down sampled with low-pass filtering to 256×256 pixels in order to reduce aliasing artifacts. We use GPU for depth image rendering.

2.2. Scale-Weighted SIFT Feature Extraction

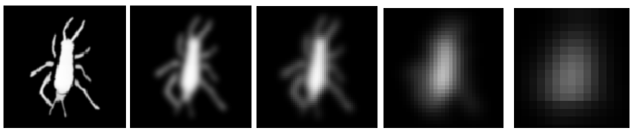
In this step, a set of SIFT features are extracted from each depth image rendered. The SIFT [7] is a scale and rotation invariant local visual feature having dimension of 128. By employing a multiresolution image pyramid, the SIFT feature capture multi-scale and multi-orientation image features. The original BF-SIFT [8] employed Lowe's SIFT without modification, which starts with *interest point* detector, followed by the feature extraction at these interest points. The SIFT places interest points at the points having high gray level gradient across the scale. As mentioned before, this causes problem if the rendered depth image of the query model contains cracks and/or jagged edges that are the result of Laser range scan.

On one hand, a large portion of the interest points are placed at these cracks and jagged edges. On the other hand, important yet smooth or feature-less parts of the model tend to be undersampled as they generate much smaller number of interest points. To counter these problems, our proposed method introduced the following two changes:

(1) **Dense and Random Sampling:** The proposed method discards the interest point detector, and places a large number of sample points at random locations for all the images in the multi-scale image pyramid.

(2) **Low-Frequency Emphasis:** Noise in range scanned query models, such as jagged perimeter edges and cracks are mostly of high-frequency, that is, smaller scale, features. To disregard the noise, our sampling method emphasize low frequency component in the image by importance sampling, that is, by allocating more samples in the low-frequency images. Low-frequency emphasis also enables the method to capture more of global shape features.

Figure 4 shows examples of depth images in the multiresolution image pyramid. In this example, legs of the insect model are visibly detached from the body in the image having size 256×256. However, the discontinuity become less noticeable as the image is low-pass filtered and down sampled. For example, at the 64×64 image, while the legs are still recognizable, they appear to be connected to the body. At each scale, sample points for the dense sampling are placed randomly at pixels having non-zero pixel values to concentrate samples on or near the foreground object.



256×256 128×128 64×64 32×32 16×16

Figure 4: Example of images in the multiresolution image pyramid. ("$N \times N$" indicates the image size in pixels.)

Allocation of samples across the scale is controlled by a *scale weighting parameter* W. If we have M randomly placed samples at a scale having image size $N \times N$pixels, the next lower (i.e., larger scale) frequency image of image size $N/2 \times N/2$ has M/W randomly located samples. By altering W, either high frequency or low frequency images is emphasized. At $W = 4.0$ the number of samples is even across the scale. High frequency is emphasized if $W > 4.0$, and low frequency is sampled more if $W < 4.0$.

We compare the P-BF-DSIFT with our previous methods P-BF-ISIFT and P-BF-IGSIFT. Additionally, we implemented a variation called *Partial view Bag-of Grid SIFT* (*P-BF-GSIFT*) that uses grid sampling only. In comparison, the P-BF-IGSIFT switches from grid sampling at low-frequency to interest-point sampling at high frequency. In our P-BF-GSIFT, location of the grid points are the same across the scale; in other words, the same spatial location is sampled at various scale. In comparison, the random placement of the P-BF-DSIFT is so that the sample positions are different from one scale to the other. Figure 2c and 2f show examples of sample points for the proposed P-BF-DSIFT.

To speed up extraction of SIFT features, we use a GPU implementation of SIFT named *SiftGPU* by Wu [10] for all of the P-BF-DSIFT, P-BF-GSIFT, and P-BF-DSIFT. Since the SiftGPU is a SIFT with interest point detector, we modified it for the P-BF-DSIFT and P-BF-GSIFT so that SIFT sample locations can be specified.. For the P-BF-ISIFT, the original SiftGPU is used. With the increase in the number of sample point for the P-BF-IGSIFT and P-BF-DSIFT sampling, acceleration by using GPU brings significant speedup.

2.3. Vector quantization and histogram generation

Following the bag-of-features approach, the proposed method performs *Vector Quantization* (VQ) of the SIFT features generated from a range image into code vectors in the pre-computed codebook. After the VQ, the frequencies of code vectors, or "visual words", are counted to create a histogram. The histogram then becomes the feature vector for the range image.

The codebook for the VQ is learned by clustering local features generated from 3D models in the database. Vector quantization then is a process of finding a code vector closest to a feature to be quantized among N_v code vectors. The clustering for codebook learning often uses k-means clustering. However, k-means takes a long time especially if the number of features and the dimension of the feature increase. Furthermore, finding the closest code vector in high dimensional feature space for thousands of features per 3D model is quite time consuming. To speedup both codebook learning and the nearest neighbor search for VQ, we use the randomized decision tree algorithm by Guerts [5]. A tree node of the decision tree corresponds to a subdivision of feature space by a hyperplane perpendicular to a coordinate axis. The quality of cluster created by the randomized tree is less than that of the k-means. However, randomized tree is much faster than the k-means in learning codebook and performing vector quantization.

After the VQ, the visual words generated are accumulated into a histogram per range image (i.e., view) whose dimension is equal to vocabulary size, or number of clusters, N_v.

2.4. Distance computation

A feature vector of the partial view query 3D model is compared with N_i feature vectors of a 3D model in the database. The minimum of the N_i distances computed becomes the distance between the query and the 3D model. For the distance measure, our method uses a symmetric version of the *Kullback-Leibler Divergence*, or *KLD*. The distance $D(\mathbf{x}, \mathbf{y})$ between the N_v dimensional feature vectors $\mathbf{x} = (x_j)$ and $\mathbf{y} = (y_j)$ due to KLD is computed as follows;

$$D(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{n} (y_j - x_j) \ln \frac{y_j}{x_j}$$

Note that the distance computation above is performed per view, so that a larger number of N_i means a longer time to compare a full 3D model with a query.

To speed up the distance computation, we replace the function call to ln() with a table lookup. As the histogram is very sparse and a histogram bin has a small integer value, the table is quite compact, having only 100 entries. The table is small enough to easily fit in a cache memory of a modern CPU.

3. Experiments and Results

We evaluated the method through the following five experiments;

(1) **Sampling pattern and retrieval performance:** Compares the P-BF-DSIFT with P-BF-ISIFT and P-BF-GSIFT for their retrieval performance.
(2) **Scale-weighting and retrieval performance:** Quantifies effect of the scale-weighted sampling on retrieval performance for the P-BF-DSIFT.
(3) **Number of views and retrieval performance:** Quantifies effect of number of rendered views N_i per 3D model on retrieval performance for the P-BF-DSIFT.
(4) **Vocabulary size and retrieval performance:** Quantifies effect of visual word vocabulary size N_v on retrieval performance.
(5) **Performance comparison with other methods:** Compares the retrieval performance of the P-BF-DSIFT with other methods.
(6) **Computational cost:** Quantifies computational costs for the four processing steps of the P-BF-DSIFT.

For evaluation, we used the database and ground truth categories of the SHREC 2009 Partial 3D Models Track [1]. The database consists of 720 models divided into 40 classes, such as plant, furniture, and airplane. The query set is 20 partial view 3D models generated by range scanning real objects using a viewpoint per model. As the numerical performance index, we used *Mean First Tier* (MFT). First Tier is the ratio of 3D models retrieved from the correct class in the top q retrievals, where q is the size of the correct class. The number is averaged over all the queries to yield Mean First Tier.

Throughout the experiments, we used the following parameters. We fixed the number of viewpoint N_i for the multi-view rendering to $N_i = 42$, except for the third experiment in which N_i is varied. In rendering, 3D models are rendered into 256×256 pixels. Partial view 3D model is first rendered into 1024×1024 pixels, followed by smoothing and down sampling into 256×256 pixels before SIFT feature extraction. The codebook for vector quantization is learned from a set of 500k SIFT features. The set is a randomly selected subset of SIFT features extracted from 3D models in the retrieval target database.

3.1. Sampling Pattern and Retrieval Performance

Figure 5 shows, for P-BF-ISIFT, P-BF-GSIFT, and P-BF-DSIFT, the relationship between number of samples N_p and retrieval performance measured by MFT.

The plot for the P-BF-ISIFT is a point since its N_p is determined automatically by the interest point detector. For the P-BF-GSIFT and P-BF-DSIFT, we varied N_p from about 100 to 3,600. In this set of experiments, for all the sampling methods, scale weighting factor is set at W=4.0 for even sampling across the scale.

The best performing was the proposed P-BF-DSIFT, followed by the P-BF-GSIFT. The performance of P-BF-DSIFT increased with N_p, but is saturated at around $N_p = 1{,}200$ with MFT = 32.5%. The P-BF-GSIFT gained performance as N_p is increased, reaching MFT = 27.2% at $N_p = 3{,}600$. However, its performance never matched that of P-BF-DSIFT in the range of N_p we have experimented with. We suspect that the same regular grid points across the multi-scale image pyramid of the P-BF-GSIFT produced features having limited diversity. In comparison, P-BF-DSIFT places samples at different positions across the scale, presumably improving feature's diversity.

Original SIFT with its interest point detector performed the worst with its MFT = 9.2%. Two primary reasons for its low performance are the small sample counts per image ($N_p = 66$) and the misplacement of samples around range scanning artifacts such as cracks and jagged edges.

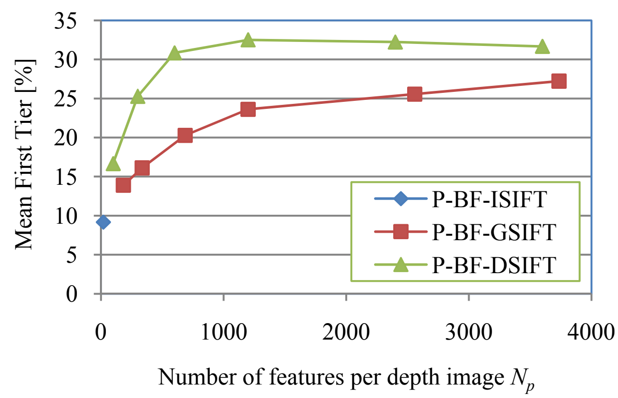


Figure 5: Number of samples N_p per depth image and retrieval performance plotted for three sampling methods.

3.2. Scale-Weighting and Retrieval Performance

In this experiment, we evaluated the relationship between the scale weighting parameter W and retrieval performance. We use the P-BF-DSIFT, and its number of samples is set at $N_p = 1{,}200$. We then tried various value of W in the range 1.0~12.0 to modify the allocation of samples over the scale space.

Figure 6 shows the result of the experiment. There is a performance peak at W=1.5 with MFT = 37.2% where low-frequency image has more samples than high-frequency images. The performance at W=1.5 is about 5% better than when even sampling density is employed across the scale.

Interestingly, there is little change in retrieval performance between even sampling (W=4.0) and high-frequency emphasized sampling (W>4.0) across the scales.

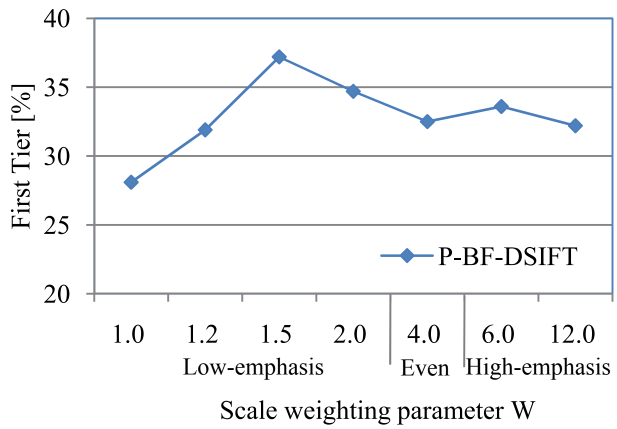


Figure 6: Scale weighting parameter W and retrieval performance. Importance sampling low-frequency images (W=1.5) improved retrieval performance.

3.3. Number of Views and Retrieval Performance

It is expected that an increased number of views N_i to render depth images will provide better approximation to rotational invariance, at the expense of higher cost of distance computation. Figure 7 shows the relationship between the number of views N_i to the retrieval performance. As the graph shows, performance jumps when N_i is increased from 20 to 42, while the performance change from 42 to 80 is negligible. In the following experiments, we fix the number of views $N_i = 42$.

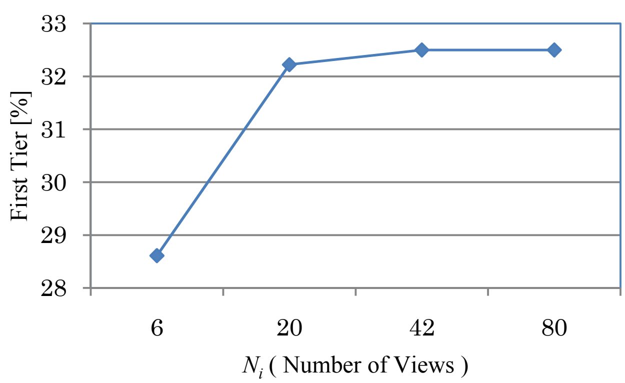


Figure 7: Number of views $\mathrm{N_i}$ and retrieval performance for the P-BF-DSIFT.

3.4. Vocabulary Size and Retrieval Performance

Figure 8 shows the effect of vocabulary size N_v on the retrieval performance. We compared four different method and parameter combinations;

(1) P-BF-ISIFT: P-BF-ISIFT with average $N_p = 66$.
(2) P-BF-GSIFT: P-BF-GSIFT with $N_p = 1{,}200$.
(3) P-BF-DSIFT (Even): P-BF-DSIFT with even sampling (W=4.0) and $N_p = 1{,}200$.
(4) P-BF-DSIFT (LFE): P-BF-DSIFT with low frequency emphasis (W=1.5) and $N_p = 1{,}200$.

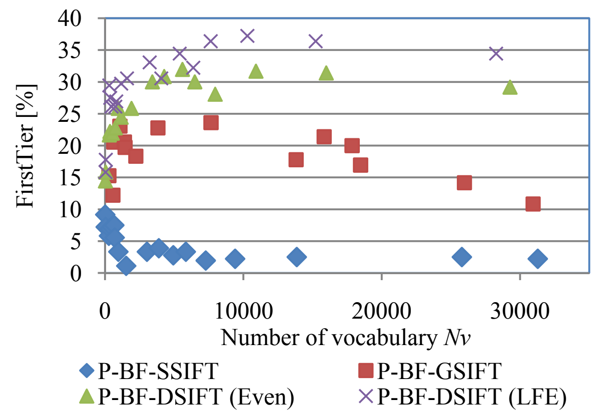


Figure 8: Vocabulary size and retrieval performance.

Note that each plot is not smooth due to the randomized nature of the clustering algorithm [5].

P-BF-SIFT peaks out at a small N_v for it has a small average number of sample points $N_p = 66$. P-BF-GSIFT has its peak at about $N_v = 10{,}000$, and the performance drops as the N_v increases past that point. P-BF-DSIFT also appears to have a peak past $N_v = 10{,}000$, but the peak is less evident than in the case of P-BF-GSIFT. Compared to the interest point sampling of the original P-BF-ISIFT, the dense sampling of the P-BF-DSIFT, and to some extent the grid sampling of the P-BF-GSIFT appear to improve the diversify SIFT features extracted, thereby increasing the vocabulary size having maximum retrieval performance. , At around $N_v = 10{,}000$, histograms for both P-BF-DSIFT and P-BF-GSIFT are very sparse, dominated by bins having 0 values. This is because the number of samples $N_p = 1{,}200$ is smaller than the number of bins $N_v = 10{,}000$

If we compare among peak retrieval performances of the methods, the P-BF-DSIFT with low emphasis has the highest retrieval performance, followed by the P-BF-DSIFT with even weighting.

3.5. Comparison with Other Methods

In this set of experiments, performances of the following six variations of methods are compared;

(1) **CMVD**: The method used by Axenopoulos and Daras [1, 4] for SHREC 2009. We used the best performing one that uses depth map image for feature extraction.
(2) **P-BF-IGSIFT**: The method used by Furuya and Ohbuchi [1] for the SHREC 2009 that combines grid points and SIFT interest points.
(3) **P-BF-SSIFT**: Our implementation of P-BF-SIFT, which uses interest-point only, i.e., original SIFT.

(4) **P-BF-GSIFT**: This method uses grid sampling only.
(5) **P-BF-DSIFT (Even)**: This method uses dense sampling and even weighting in scale space (W=4.0).
(6) **P-BF-DSIFT (LFE)**: This method uses dense sampling and low-frequency emphasis (W=1.5).

Figure 9 shows the recall-precision plots, and Table 1 shows the Mean First Tier (MFT) figures for the six methods. The exact numbers plotted in Figure 9 and tabulated in Table 1 for the CMVD and the P-BF-IGSIFT were obtained from the SHREC 2009 Partial View Model Retrieval web page. The availability of detailed performance evaluation results at the web site enabled us these direct performance comparisons.

Clearly, P-BF-DSIFT(LFE) with its MFT=37.2% performed the best. If we look at Figure 9, P-BF-DSIFT(LFE) has significantly better precision at low-recall region compared to the runner-up P-BF-DSIFT(Even).

Three methods, the P-BF-IGSIFT (MFT=22.5%), the P-BF-GSIFT (MFT=23.6%), and the CMVD (MFT=21.7%) are about equal in terms of MFT. Our proposed method P-BF-DSIFT(LFE) with its low-frequency emphasis performed about 15% better, at MFT=37.2%, than these three methods.

Figure 10 shows retrieval examples for two queries from the class "potted plant" and "nonflying insects". In these examples, the proposed method with low-frequency emphasis P-BF-DSIFT (LFE) performed better than the other two methods shown, the CMVD and the P-BF-IGSIFT from the SHREC 2009.

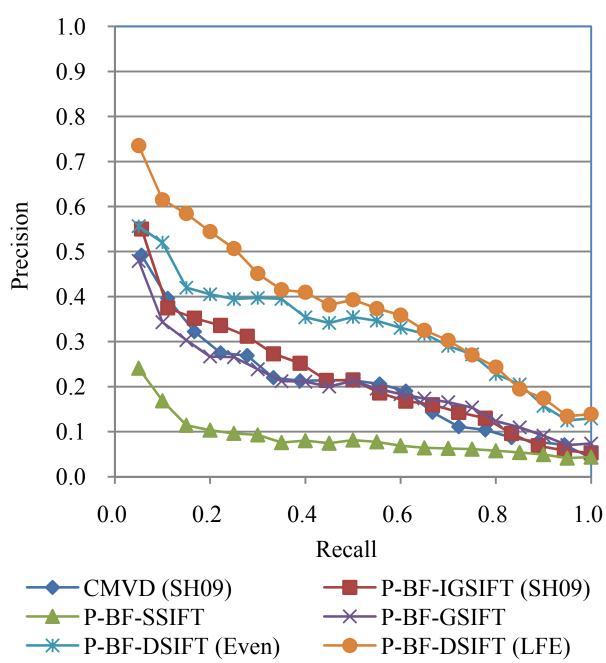


Figure 9: Comparison of retrieval performance among various methods using recall-precision plots.

Table 1: Comparison of Mean First Tier among various methods.

Methods	Mean First Tier [%]
CMVD (SH09)	21.7
P-BF-IGSIFT (SH09)	22.5
P-BF-ISIFT	9.2
P-BF-GSIFT	23.6
P-BF-DSIFT (Even)	32.5
P-BF-DSIFT (LFE)	37.2

3.6. Computational cost

In this experiment, we look into the computational cost breakdown for the proposed P-BF-DSIFT method. Computational cost of the method can be split into the codebook learning phase and query phase.

To learn a VQ codebook from 500k SIFT features, our implementation of the randomized decision tree algorithm took 63.2s. This is tolerable considering it is a one-time cost spent during pre-processing of the database.

Our method took 1.47s to process a query for the SHREC 2009 Partial View Retrieval benchmark. Table 2 list the cost breakdown for various steps of the method. To produce the table, $N_p = 1{,}200$ and $W = 1.5$ are used.

Talbe 2 shows that the cost of computing distance to search through the database containing 720 models dominated the overall computation time. This is because the number of comparison par 3D model is not 1 but the number of views N_i. The cost of distance computation would increase further as the number of object in the database increases.

Table 2: Computational cost in seconds at various steps during the query processing phase of the proposed method.

Render depth maps	Extract SIFT features	Quantize SIFT features	Compute distance	Total
0.074	0.150	0.037	1.209	1.470

4. Conclusion

In this paper, we proposed and experimentally evaluated a view-based algorithm for partial-view 3D model to full 3D model comparison named *Partial view Bag-of-Features Dense SIFT* (*P-BF-DSIFT*). The method compares a view of range image from the query model with multi-view range images of a full 3D model. To deal with artifacts due to range scan, such as cracks and jagged perimeter edges, the method employed *dense sampling* and *low-frequency weighted importance sampling* of local visual features. Experimental evaluation showed that the proposed method significantly outperformed our previous methods on which the proposed method is based. The proposed method outperformed the other methods we have compared against by 15% in Mean First Tier.

Future work would include further exploration of scale-space weighting and its effect on various shape retrieval benchmarks, e.g., those with and without articulation. For example, low frequency emphasis might impact the performance for articulated models. We also would like to improve computational efficiency of the method, especially by improving the distance computation step.

References

[1] A. Axenopoulos, P. Daras, H. Dutagaci, T. Furuya, A. Godil, R. Ohbuchi, SHREC 2009 - Shape Retrieval Contest of Partial 3D Models, *Eurographics Workshop on 3D object retrieval*, 2009.

[2] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual Categorization with Bags of Keypoints, *Proc. ECCV '04 workshop on Statistical Learning in Computer Vision*, 59-74, 2004

[3] D-Y. Chen, X.-P. Tian, Y-T. Shen, M. Ouh-young, On Visual Similarity Based 3D Model Retrieval, *Computer Graphics Forum*, 22(3):23-232, 2003.

[4] Petros Daras, Apostolos Axenopoulos, A Compact Multi-view Descriptor for 3D Object Retrieval, Proc. *Seventh International Workshop on Content Based Multimedia Indexing*, 115-119, 2009.

[5] P. Guerts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Machine Learning*, 36(1):3-42, 2006.

[6] M. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three Dimensional Shape Searching: State-of-the-art Review and Future Trends, *Computer Aided Design*, 5(15):509-530, 2005.

[7] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *Int'l Journal of Computer Vision*, 60(2):91-110, 2004.

[8] R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient local visual features for shape-based 3D model retrieval, *Proc. IEEE Shape Modeling International 2008* (*SMI 2008*), 93-102, 2008.

[9] J. Tangelder, R. C. Veltkamp, A Survey of Content Based 3D Shape Retrieval Methods, *Proc. SMI '04*, 145-156, 2004.

[10] C. Wu, SiftGPU: A GPU Implementation of David Lowe's Scale Invariant Feature Transform (SIFT), http://cs.unc.edu/~ccwu/siftgpu/

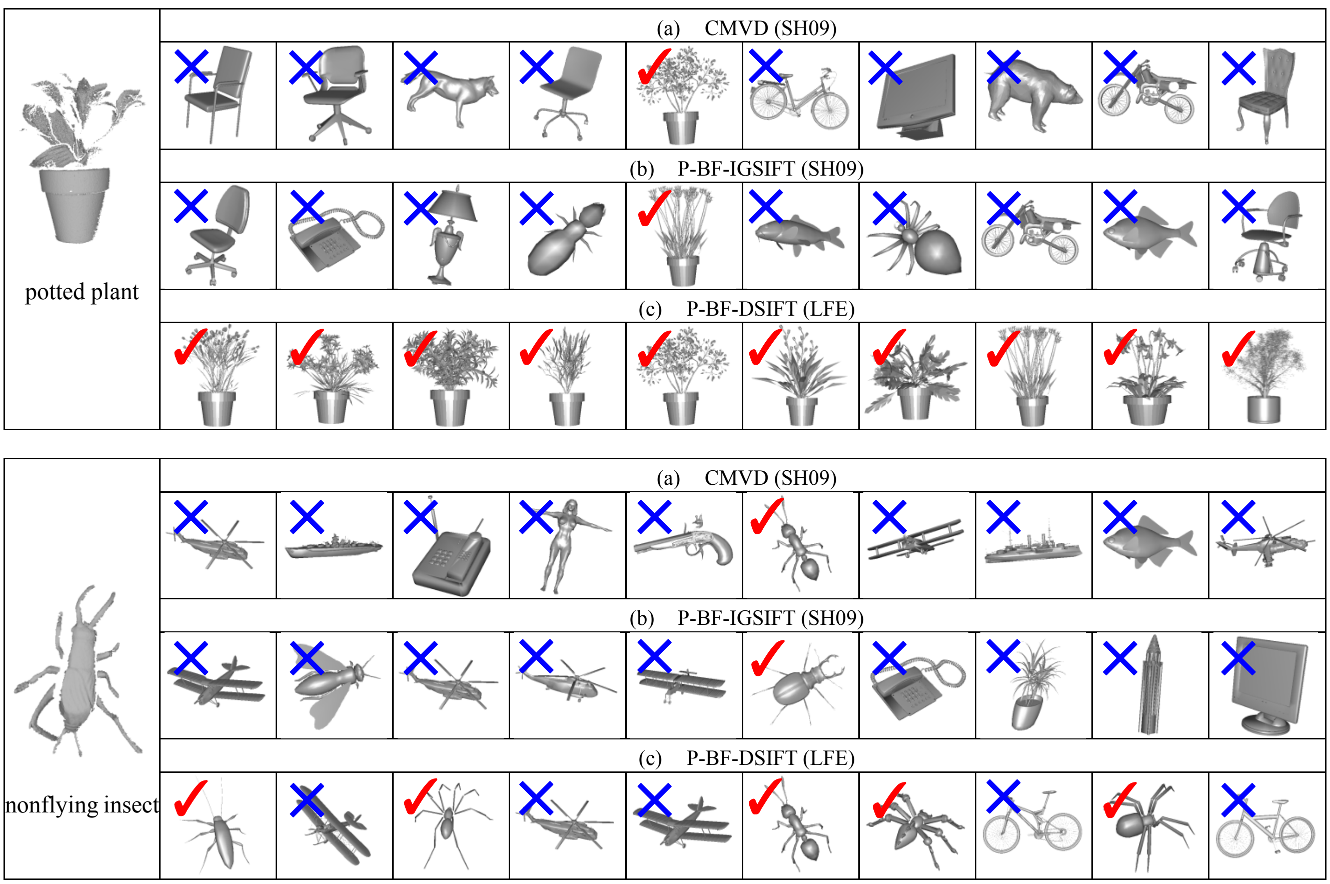


Figure 10. Retrieval examples for the two classes "potted plant" and "nonflying insect". The proposed method P-BF-DSIFT (LFE) performed better than the others in these examples.