Efficient 3D Object Retrieval Using Depth Images

N. Vajramushti
Dept. of Computer Science
University of Houston
Houston TX 77204-3010
nikhilv@cs.uh.edu

I. Kakadiaris
Visual Computing Lab
Dept. of Computer Science
University of Houston
Houston TX 77204-3010
ioannisk@uh.edu

T. Theoharis, G. Papaioannou
Dept. of Informatics
University of Athens
Athens Greece
theotheo@di.uoa.gr
georgep@di.uoa.gr

ABSTRACT

In this paper, we present a new three-dimensional object retrieval method. This method employs depth buffers (computed in hardware) for representing and comparing the objects. Specifically, multiple depth buffers per object (computed from different points of view) are compared for surface and volume similarity. Our method makes efficient use of hardware, it is easily extensible for hierarchical comparisons at multiple resolutions, and is highly parallelizable. We have employed this method for both inter-class and intra-class retrieval tasks on a gallery of over 3,000 three-dimensional objects of vehicles with very encouraging results. The accuracy of the method depends on the number of depth buffers and the depth buffer resolution.

1. INTRODUCTION

Three-dimensional (3D) objects are proliferating and are accessible to a common user from the internet. However, users need a way to find the object they want. A web search engine to search 3D objects is an example of such an application.

Many approaches have been proposed for 3D object retrieval with the usual trade-off between retrieval accuracy and efficiency. In this paper, an accurate 3D object retrieval method is presented that uses standard graphics hardware for acceleration. This method exploits depth images of objects derived by hardware depth buffers. Multiple views of the same object cover more details of the object.

Specifically, the proposed method compares a probe 3D object with all 3D gallery objects and retrieves the closest match to the probe 3D object. The matching error computation is based on the similarity of a number of depth images of the two objects being compared. The effect of the depth buffer resolution and multiple views is investigated. The effectiveness of the object retrieval method is assessed using traditional information retrieval effectiveness metrics.

This method can be applied to various specialized object retrieval problems. For our purposes, we define a class of objects to be a category of objects that share similar surface geometry. Passenger cars can be considered as a single class. In this paper, we test the proposed method by applying it to the recognition of objects within a class (e.g., passenger cars) and between multiple classes (e.g., small cars, full size cars, and trucks)

The rest of this paper is organized into the following sections: Section 2 covers related work. Section 3 describes our approach in detail. Section 4 summarizes the statistics of the gallery objects and Section 5 illustrates experiments and results. Section 6 draws the conclusions of this paper.

2. RELATED WORK

There are many projects ([6], [31], [8], [27], [23], [16], [4], [1], [30], [17], [22], [10], [2], [11]) investigating 3D object retrieval systems. All these retrieval systems contain a representation and indexing components, and a similarity model and sorting rules component.

Hlavaty *et al* [9] and Loncaric *et al* [15] survey the representation techniques. These representations can be categorized as follows: (a) Two-dimensional view-based representations [2], [12], (b) Shape-based representations [31], [4], [17], [16], [24], [25], [29], [30], (c) Histogram-based representations [18], [1], [11], and (d) Topology-based representations [8], [22]. Our method uses a representation that is two-dimensional view-based. The other techniques that employ two-dimensional view-based representations lose 3D information. Our method represents the objects using their depth images and hence retains the 3D information.

Veltkamp *et al* [26] survey the contemporary similarity models. The similarity models can be categorized as follows: (a) Two-dimensional view-based [2], (b) Feature-based similarity [31], [3], [28], [16],[25], [30], (c) Histogram-based similarity [18], [1], [11], (d) Topology-based similarity [8], [22]. Our method employs a combination of the volume descriptor (feature-based), and depth buffer-based surface matching technique (two-dimensional view-based) similarity.

In the past, the depth images have been used in the object recognition methods [14]. Few other computer vision techniques also employ the Z-buffer. Labsik *et al* [13] register freeform surfaces using depth buffers of the surfaces. Their method used *Iterative Closest Point* or *ICP* algorithm [7] to find least sum of square distances between depth buffer points.

Papaioannou *et al* [19] proposed a method to match complimentary 3D objects. It also uses depth images of the 3D objects to extract surface geometry and exploits the hardware acceleration to speed up matching process. Their method uses gradients of the depth buffer in X and in Y directions as error metric. They find the complementarity of the objects being compared. This paper extends their idea to object retrieval, in which we find the similarity instead of complementarity of the objects being compared.

3. THREE-DIMENSIONAL OBJECT RETRIEVAL USING Z-BUFFERS

In this section, we describe the proposed method for the estimation of the degree of similarity between two 3D objects and the estimation of the best relative pose. This method can very well be applied to any type of 3D object representations for which depth images can be produced. In this paper, we concentrate on polygonal objects for which we can take advantage of the graphics hardware pipeline, available on most modern graphics cards, in order to compute the depth images efficiently.

3.1 Overview of the Algorithm

First, we assume a regular convex polyhedron with n faces that completely surrounds the probe object and the gallery object. The faces of the convex polyhedron define the planes for which the n depth buffer images will be computed. A typical choice would be the cube (see Section 3.3). The gallery object is stationary, while the probe object is continuously being transformed, trying to match the gallery object.

Initially the two objects are aligned using a standard alignment techn-ique, such as Principal Component Analysis. This step reduces the range of the parameter values.

Algorithm: Three-Dimensional Object Retrieval Using Z-Buffers

Preprocessing: Pre-compute n depth buffers for each gallery object. Store these depth images of each object efficiently by compressing them. Also pre-compute the volume using these depth buffers. Store computed volume along with the depth buffers.

Step 1: Input probe object

Step 2: Compute the n depth buffers for the probe object and compute the volume of the probe object using the depth buffers.

Step 3: Select next object from the gallery and then retrieve n stored depth buffers for the gallery object.

Step 4: Determine whether the gallery object is a *Close Match* to the probe object. If not go to the Step 9.

Step 5: Perturb the pose vector that controls the orientation of the probe object according to an optimization technique.

Step 6: The transformations that correspond to the new pose vector are applied to the probe object, using graphics hardware to implement the transformations efficiently.

Step 7: Efficiently compute n depth buffers, corresponding to n orthographic projections of the probe object, using graphics hardware. Compute a similarity measure (or error) between the n depth buffers of the gallery and probe objects.

Step 8: Check whether the error has been minimized according to the optimization strategy used. If so, then output the optimum relative pose (giving the minimum error value) and the error value. Else repeat Steps 5-8 until the error is minimized.

Step 9: Repeat Steps 3-8 for all the objects in the gallery.

These steps are carried out for every gallery. The final ranking produced is the retrieved set. The first gallery model in the retrieved set is the (closest) match. Note that the depth buffers can be computed at a very coarse resolution first, in order to efficiently discard distant comparisons, and then at increasing resolutions as needed.

3.2 Step 1: Input Probe Object

Our method requires the user to specify a polygonal 3D object. The specified object can be in any format that the implementation understands.

3.3 Steps 2 & 7: Compute Z-Buffer Values

We access the graphics hardware and retrieve the Z-buffers of orthographic projections of the gallery and the probe objects. Different orthographic views produce different Z-buffer values.

Since parts of the two objects may be similar while the rest may not be (e.g., the underside of cars does not differ greatly between different cars) then, obviously, if the number of depth images is small the algorithm will not have enough discriminatory power. On the other hand, increasing the number of depth buffers raises proportionately the complexity of the algorithm. Also, the more concavities the objects have, the larger the number of views that will be necessary in order to take them into account as surface features (and of course some concavities will not be visible from any external angle). Figure 1 illustrates the different Z-buffers used for a 3D object.

We have chosen the cube (6 depth buffers) as it is the simplest closed convex polyhedron. Open polyhedra, such as 2- or 4-face parallelepipeds could miss a significant portion of surface detail. We expect that the discriminatory power of the algorithm would be reduced in such cases.

3.4 Steps 2: Compute Volume

The 3D object retrieval method discussed in this paper compares the objects based on their volumes. First, we threshold volume difference between the probe and the gallery object to generate index set A. The concept of volume for polygonal objects can be approximated to the volume enclosed in the polygons on the surface. For all practical concerns, cavities hidden inside the objects can be left unaccounted for.

The volume of the 3D object can be viewed as the number of voxels lying inside the object. Passalis *et al* [20] proposed a depth buffer-based voxelization method. Since use

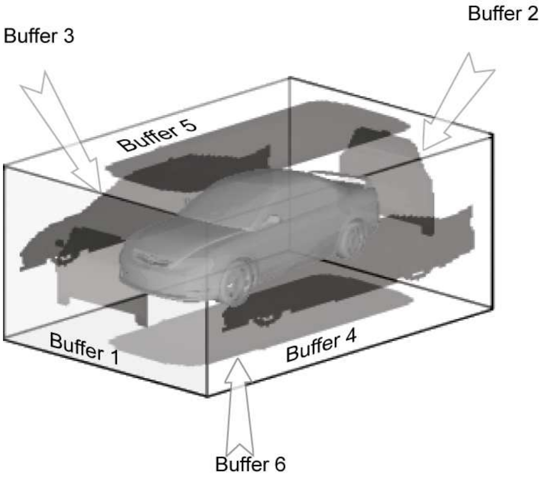


Figure 1: Illustration of different Z-buffers computed with respect to the 3D object.

of depth buffer is consistent with the other techniques in the 3D object retrieval method in this paper, we use the voxelization method mentioned above. This volume computation method is almost pose invariant, but is prone to aliasing, as voxelization is equivalent to sampling. But there could be a minor difference in the computed volumes, which could be attributed to a combination of pose and aliasing.

3.5 Step 4: Determine a Close Match

The volume of the probe object, v_p is compared against the volume of the gallery object, v_g. For this comparison, the absolute difference between these two volumes, δ_v is computed. The gallery object for which δ_v is less than a certain threshold ε_d is considered to be a *close match.* The threshold ε_v is chosen so as to account for the minor volume computation differences due to pose and aliasing. Let S_G be the set of all the gallery objects. The retrieved set is thus:

$$S_R(p) = \{i : i \in S_G, |v_p - v_i| < \varepsilon_v\} \tag{1}$$

The retrieved set is represented by the formula given in the Eq. (1). In theory, for a request of an object belonging to a very large class like passenger cars, $S_R(q)$ can be very large. In practice, maximum number of elements in $S_R(p)$ is limited to moderate number, like 25. The lower limit to the cardinality of the retrieved set for a given request is zero. In case the gallery contains objects belonging to many classes, it is quite possible that volume of a object of a certain class can closely match volume of the probe object of some other class. To filter out such objects from the retrieved set, further matching process is necessary.

3.6 Steps 5 & 8: Optimization

To determine how similar two objects are, it is necessary to find the pose of the gallery object that gives the globally minimum error relative to the gallery object. Exhaustively searching the six dimensional continuous space (R^6) is extremely expensive. With 120 samples per rotational degree of freedom (i.e., every 3°) and 0.5 percent (of maximum) translational sampling, we need to sample a $120^3 \times 200^3 = 14 * 10^{12}$ points in space. Even with a very fast graphics card for the inner loop comparisons, it would take years.

This calls for a global optimization method to estimate the optimum matching error in reasonable time. In our method, enhanced simulated annealing (ESA) is used for the optimization. ESA [21] is a non-deterministic global optimization algorithm aimed at problems with high dimensionality. ESA uses space partitioning to improve convergence. At each step, only a subset of the complete parameter vector is perturbed. The ESA convergence algorithm has better probability of finding a global minima than does deterministic convergence algorithm.

3.7 Step 6: Transformations

As part of the comparison process, the gallery object is static, while the probe object is being transformed according to parameter vector. The center of masses of the objects are aligned with the origin. The probe object can freely rotate about any of its three axes by arbitrary angles θ_x, θ_y, and θ_z. It also has three translational degrees of freedom giving a translation vector $\vec{t} = [t_x, t_y, t_z]^T$. Thus, in all there are 6 degrees of freedom for the probe object. The pose of the probe object belongs to $\Re^6$. We obtain a pose vector containing 6 parameters for comparison:

$$\vec{\omega} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T \tag{2}$$

For every value of the pose vector $\vec{\omega}$ the probe object is transformed using standard graphics hardware (Figure 2). If T represents the translation matrix by $\vec{t}$ and $\mathbf{R}_{\hat{\mathbf{e}},\theta}$ is the rotation matrix by angle θ about axis $\hat{e}$ [5], then the transformation applied to the probe object is given by:

$$\mathbf{M} = \mathbf{T}\mathbf{R}_{\mathbf{z},\theta_\mathbf{z}}\mathbf{R}_{\mathbf{y},\theta_\mathbf{y}}\mathbf{R}_{\mathbf{x},\theta_\mathbf{x}} \tag{3}$$

We have assumed that the probe object and the gallery object are approximately aligned. Hence the rotational movement range is limited to ±15% of [0, 2π].

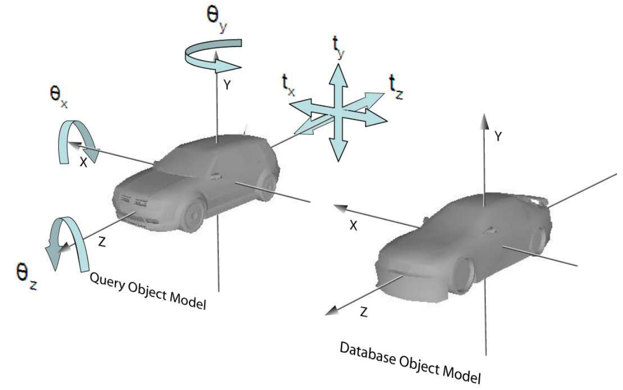


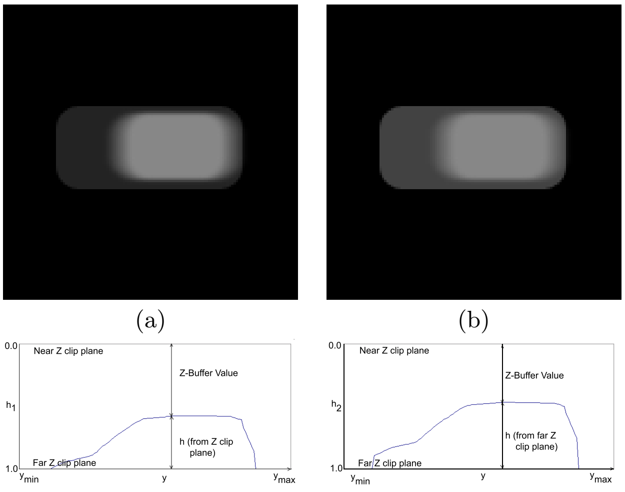
Figure 2: Illustration of the degrees of freedom of a probe object.

3.8 Step 7: Error Computation

We measure the similarity between the probe object and the objects belonging to the retrieved set given in Eq. (1) to accurately rank objects in the retrieved set. To measure the similarity of two objects at a specific pose, we estimate the difference between the n corresponding pairs of depth images. Since we treat all views equally, the final value is the average value. In this section, we describe how this difference (or error) value between two depth images is computed.

Error Metric: Similarity of Depth Image Surfaces

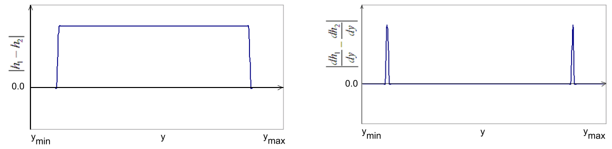
Figure 3 depicts a probe car object and its matching car object in the gallery and their depth buffers. The probe car object is translated by different amounts along Z axis. To compute the similarity measure, a naïve approach is to find the difference between the point-wise distances of the object from the reference plane. Unfortunately, in this simple approach, the error value depends on the distance of the object from the reference plane. Figure 4 (a) illustrates this approach. To overcome this limitation we take the derivatives of the object surface contour with respect to the axes of the reference plane as in Papaioannou *et al* [19]. The error is then estimated as the difference between corresponding derivatives as shown in Figure 5. This approach eliminates the dependency on the relative distance of the object from the reference plane. This is illustrated in Figure 4(b), where little variation in the measure will not make as much difference in the metric as compared to Figure 4(a).



(a) (b)

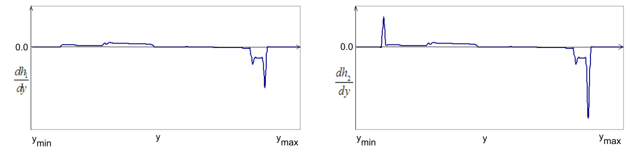
(c) (d)

Figure 3: Computation of matching error for identical objects displaced along the Z axis. (a) Object closer to the reference plane, (b) Object further from the reference plane, (c) Z-Buffer computation of the object closer to the reference plane, and (d) Z-Buffer computation of the object further from the reference plane.



(a) (b)

Figure 4: Computation of matching error (a) Difference between distances, and (b) Distance between derivatives.



(a) (b)

Figure 5: Computation of matching error using derivatives of object surfaces (a) Profile plot of the derivatives of Z-buffer of the object in Figure 3(c), and (b) Profile plot of the derivatives of Z-buffer of the object in Figure 3(d).

The error is computed for a particular pose of the two objects (pose vector value) over the entire region of overlap $O(\vec{\omega})$ of the two (projected) objects. Moreover, the error is normalized by dividing by the area A_o of $O(\vec{\omega})$:

$$\varepsilon_d(\vec{\omega}) = \frac{1}{A_O} \int\int_{O(\vec{\omega})} \Big(\Big| \frac{\partial d_1(x,y)}{\partial x} - \frac{\partial d_2(x,y)}{\partial x} \Big| + \Big| \frac{\partial d_1(x,y)}{\partial y} - \frac{\partial d_2(x,y)}{\partial y} \Big| \Big) dS \qquad (4)$$

where, $\vec{\omega}$ is the pose vector given in Eq. (2) that defines the pose and d_1, d_2 are distances of the two objects respectively from the reference plane, which is indexed by x and y.

In discrete form, samples are taken along both the x and y axes at locations $(i, j), i = 1, \cdots, N_x, j = 1, \cdots, N_y$, for both objects. Partial derivatives in Eq. (4) can be approximated by forward differences

$$\begin{aligned}
\Delta_x d_1(i,j) &= d_1(i+1,j) - d_1(i,j), \\
\Delta_y d_1(i,j) &= d_1(i,j+1) - d_1(i,j), \\
\Delta_x d_2(i,j) &= d_2(i+1,j) - d_2(i,j), \\
\Delta_y d_2(i,j) &= d_2(i,j+1) - d_2(i,j);
\end{aligned}$$

and integration can be replaced by summation:

$$\begin{aligned}
\varepsilon_d(\vec{\omega}) &\cong e_d(\vec{\omega}) \\
e_d(\vec{\omega}) &= \frac{1}{N_O} \sum_{(i,j) \in O(\vec{\omega})} (|\Delta_x d_1(i,j) - \Delta_x d_2(i,j)| \\
&\qquad + |\Delta_y d_1(i,j) - \Delta_y d_2(i,j)|) \qquad (5)
\end{aligned}$$

where N_O is the number of points in the region of overlap $O(\vec{\omega})$ of the projections of the two objects onto the reference plane.

We can replace d_1 and d_2 in Eq. (5) as follows:

$$d_1(i,j) = Z_1(i,j), \qquad (6)$$

$$d_2(i,j) = Z_2(i,j). \qquad (7)$$

where Z_1 and Z_2 are the depth buffers of the probe and gallery object respectively.

Error Metric: Similarity of Volume Extent

The depth buffer comparisons proposed so far take account of the surface geometry of the two objects. However, two dissimilar objects with identical geometry in the area of overlap of their projections can have quite different projection extents. To take this into account, the matching error function given by Eq. (5) should be modified. We assume that the rendered scene only contains the objects being compared. In Figure 6 (a, b), N_{S_1} and N_{S_2} denote the buffers corresponding to the two objects. We quantize the volume extent comparison into a matching error function, e_v, which ideally approaches zero when depth buffers of the objects compared overlap exactly.

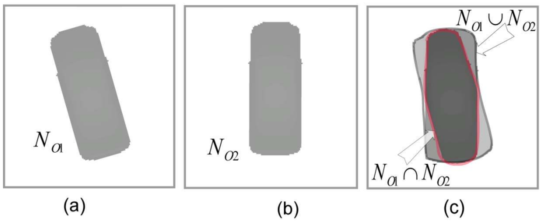


Figure 6: Volume extent error function. (a) Top view of probe object, (b) Top view of gallery object, and (c) Combination of the views.

$$e_v(\vec{\omega}) = \frac{N_{S_1} \bigcup N_{S_2}}{N_{S_1} \bigcap N_{S_2}} - 1 \tag{8}$$

where N_{S_1} is the number of non-background depth buffer pixels of the probe object and N_{S_2} is the number of non-background depth buffer pixels of the gallery object.

Combining error metrics given by Eq. (5) and Eq. (8), we derive a composite error metric, $e_c(\vec{\omega})$ as

$$\begin{aligned} e_c(\vec{\omega}) &= e_d(\vec{\omega}) \cdot e_v(\vec{\omega}) \\ &= (\frac{N_{S_1} \bigcup N_{S_2}}{N_{S_1} \bigcap N_{S_2}} - 1) \\ &\quad \cdot (\frac{1}{N_{S_1} \bigcap N_{S_2}} \\ &\quad \sum_{(i,j) \in O(\vec{\omega})} (|\Delta_x d_1(i,j) - \Delta_x d_2(i,j)| \\ &\quad\quad + |\Delta_y d_1(i,j) - \Delta_y d_2(i,j)|)) \end{aligned} \tag{9}$$

As it can be easily seen, for e_c to attain a minimum value, both e_d and e_v must converge to minima. This makes e_c dependent on both surface geometry and object extent.

4. GALLERY DESCRIPTION

The object retrieval method described in this paper was tested on a large gallery of 3D objects.

The objects are characterized by their volumes in voxels. Table 1 shows the statistics of the gallery objects. This table shows that the gallery contains a large population of compact passenger sedans (or other types of automobiles having similar volumes).

Class	Volume	X	Y	Z	Count
Small cars	40000.00	1.98	1.67	4.12	1246
Mid-size cars	44700.00	2.05	1.40	4.82	922
Full-size cars	50900.00	2.41	1.85	5.10	826
Trucks	100000.00	3.12	2.98	5.65	642

Table 2: Limits of the classification criteria in the object gallery.

The gallery contains many different types of cars. We partition these cars into different classes according to their size. These classes are: (a) Small cars, (b) Mid-size cars, (c) Full-size cars, and (d) Trucks.

The size is determined by the bounding box dimensions and the volume. Table 2 depicts the criteria used to classify the objects in the gallery used in this paper.

5. RESULTS

Three-dimensional object retrieval using depth images is tested within a class to verify that it responds predictably. The effect of number of depth buffers (Section 3.3) and the effect of depth buffer resolution on the accuracy of the method is tested. Moreover, we also test whether the method allows for an error in the initial pose estimation. Finally, we compare the performance of inter-class retrieval *vs* intra-class retrieval.

Figure 7 depicts the 3D objects used for testing. The gallery contains 3556 objects. All tests were performed on a Pentium 4 2.7GHz processor, 1GB RAM, and AGP card with 128 MB RAM.

To evaluate our method, the evaluation metric used is the Precision-Recall curve. At every recall level, the average precision of the retrieval is computed for each experiment. In this method, it is assumed that all the objects in the retrieved set are examined. These objects are first sorted according to a degree of relevance. We then examine the ranked list starting from the top document. As we proceed with the examination of the retrieved set, the recall and precision varies. These precision and recall values are plotted in a Precision-Recall curve.

Testing the Accuracy of the Algorithm Within Class

In this test, the classification task considered was the entire set of cars. We attempted to match a probe car object with all the car objects in the gallery. Probe objects illustrated in Figure 7 (a-c) were treated as probe objects.

Then these objects were displaced or perturbed by 0.1 radians along the Y axis to simulate pose estimation error. Each of these objects was then treated as a probe object and retrieved from the gallery. The test was carried out 10 times for each probe object.

When the original gallery object was used as a probe object for retrieval, its matching error with the original gallery object was always zero. This established that the metric

Characteristic	Mean	Std Dev	Max	Min
Object Volume (in voxels)	9648.14	4402.60446	67450	54
Vertex count	2731.97	2653.83	29987	146
Face count	3331.53	3002.98	40227	144
Bounding box X extent (in meters)	2.04	0.32	8.40	0.65
Bounding box Y extent (in meters)	1.50	0.40	9.14	0.17
Bounding box Z extent (in meters)	4.70	0.70	9.91	0.44
Bounding box volume (in $meters^3$)	15.23	11.79	262.27	0.17

Table 1: Gallery object statistics.

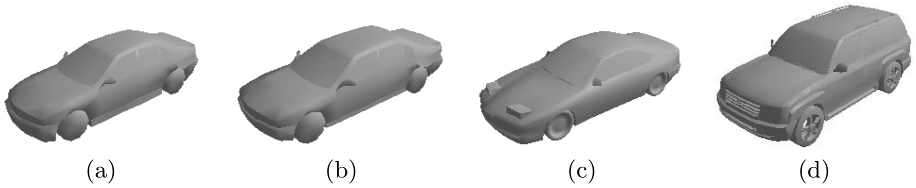


Figure 7: Examples of objects used for testing shape retrieval algorithm. (a) BMW5.CAR.FCE, (b) 750i.CAR.FCE, (c) 850csi.CAR.FCE, and (d) 4RUN.CAR.FCE.

used was a true metric. The retrieved set produced by this probe was considered as *relevant set* or R_q. The graph in Figure 8 illustrates the Precision-Recall curve for the three perturbed probe objects at the Z-buffer resolution of 128×128 pixels. The curve shows that the method retrieves 750i.CAR.FCE more accurately than 850csi.CAR.FCE and 850csi.CAR.FCE more accurately than BMW5.CAR.FCE. This disparity in the resolution power can be explained on the basis of the number of objects having similar volume. Since the gallery contains a large population of compact passenger sedans and BMW5.CAR.FCE is a compact passenger sedan, the retrieved set for this probe object contains many false positives having similar volume. 750i.CAR.FCE and 850csi.CAR.FCE are mid-size and full-size passenger sedans and hence there are not too many false positives in their retrieved set.

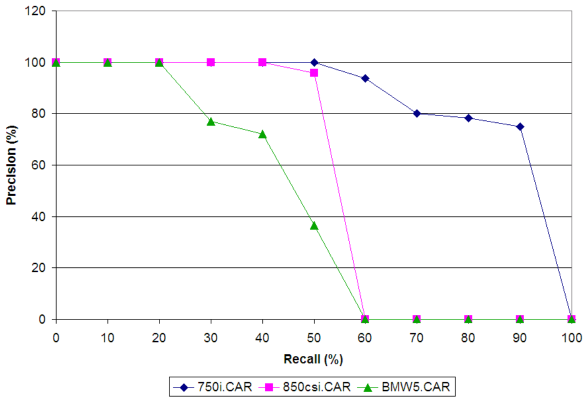


Figure 8: Precision-Recall curve for all probe objects.

Effect of Number of Buffers

In this experiment, the comparison task was carried out with different numbers of depth buffers as explained in Section 3.3. The following different configurations were used: (a) **One Buffer**: Buffer 1 in Figure 1, (b) **Two Buffers**: Buffer 1 and buffer 2 in Figure 1, (c) **Four Buffers** Buffers 1, 2, 5 and 6 in Figure 1, and (d) **Six Buffers** All 6 buffers in Figure 1. The Precision-Recall graph in Figure 9 depicts that precision is less for retrieval with one buffer at most recall levels. Retrieval with six buffers is slightly better than retrieval with two and four buffers at most recall levels. The lack of monotonic improvement in the precision can be attributed to the fact that the stochastic optimization process is random in nature.

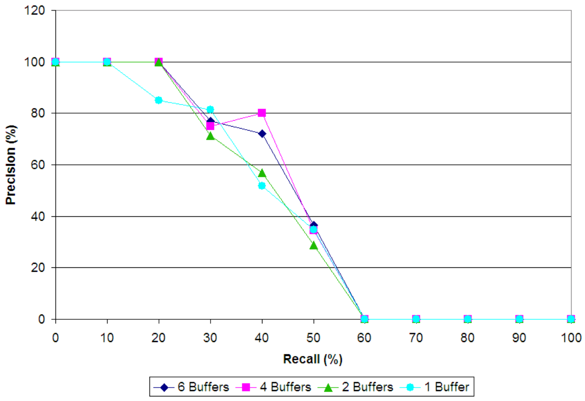


Figure 9: Precision-Recall curve for probe object BMW5.CAR.FCE with different number of depth buffers used in error computation.

Effect of Depth Buffer Resolution

The proposed method uses the depth buffer for the matching error computation. The resolution of the depth buffer is an important factor related to anti-aliasing. Hence, we test whether the resolution of the depth buffer is a factor in the accuracy of the method. In this experiment, we run the retrieval under the following resolutions: (a) 64 × 64 Pixels, (b) 128 × 128 Pixels, and (c) 256 × 256 Pixels The Precision-Recall graph in Figure 10 depicts that the average precision for retrieval under resolution of 64 × 64 pixels is less for most of the recall levels and the precision for retrieval under resolution of 128 × 128 pixels is approximately equal to precision for retrieval under resolution of 256 × 256 at all recall levels. This shows that the accuracy of the method is relatively unaffected at resolutions 128 × 128 and higher. Again, the lack of monolithic improvement in the precision can be attributed to the fact that the stochastic optimization process is random in nature.

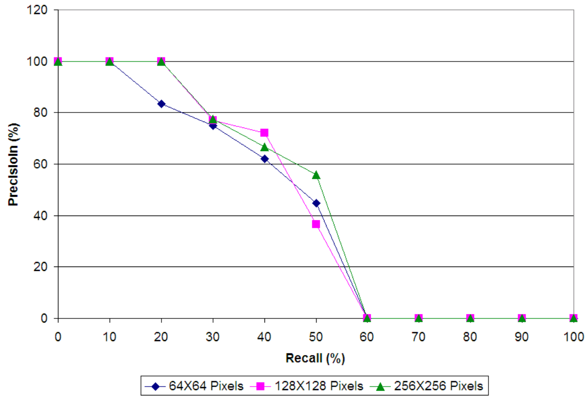


Figure 10: Precision-Recall curve for probe object BMW5.CAR.FCE at different Z-buffer resolutions.

Effect of Pose Variation in the Probe Object

The method assumes that a standard alignment technique, such as Principal Component Analysis, aligns the two objects being compared before the comparison begins. Ideally, a retrieval method should be unaffected by any error in the initial pose estimation. To test our method for pose estimation error invariance, we test the performance of the method when the poses are equal, when there is difference of 0.1 radian in the poses of the objects being compared and when there is difference of 0.2 radians in the poses of the objects being compared.

The Precision-Recall curves shown in Figure 11 depict the accuracies of the retrieval at different perturbation levels of the object BMW5.CAR.FCE. From the graph in Figure 11 it can easily be seen that without any perturbation, precision is very high at all the recall levels. Moreover, for other perturbation levels, retrieval precision is approximately equal and primarily independent of the perturbation. One can safely state that the method presented in this paper is pose estimation error invariant.

Intra-Class and Inter-Class Retrieval

In this experiment, objects belonging to different classes were chosen from the gallery. Here, we show the results of only one object. The chosen object is illustrated in Figure 7(d). This object was queried against unclassified gallery first and then against gallery classified according to the classifier function described in Section 4. The resulting precision was plotted against all recall levels. The graph in Figure 12 depicts the average precision of intra-class and inter-class retrieval experiments at all recall levels. We can see that the precision is 100% at all recall levels except the last level. This is expected as the ‘Truck’ class of objects is sparsely populated in the volume histogram. Moreover, at higher recall levels, the probability of correct retrieval should be higher in inter-class retrieval as the gallery is partitioned. The results support this intuition.

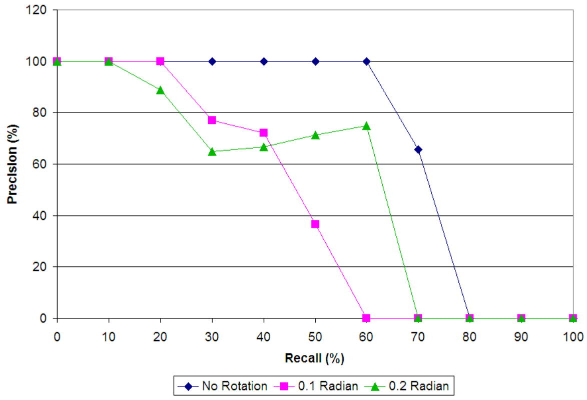


Figure 11: Precision-Recall curves at different poses of BMW5.CAR.FCE.

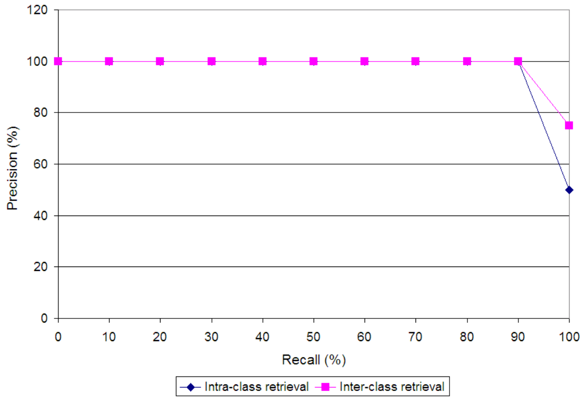


Figure 12: Precision-Recall curves for intra-class and inter-class retrieval.

6. CONCLUSIONS AND FURTHER WORK

In this paper, we have proposed a new object retrieval technique, which ex-ploits graphics hardware through the Z-buffer and coordinate transforms. Exhaustive search using this technique is impractical, hence a global optimization algorithm is employed. Though this technique is non-realtime, it is highly parallelizable as the probe object can be compared with every object in the gallery independently. Due to the non-realtime nature of this method, when searching in a large gallery, it is necessary to filter out distant matches. Voxel volume computation of a 3D object using Z-buffers is employed as a filter. Tests show that this algorithm works well for objects within a class. The accuracy of the method increases with the number of buffers used in comparison. Moreover, the higher the resolution of the depth buffers, the better is the accuracy. This method is initial pose estimation error invariant.

While this method can miss details hidden in invisible concavities of an object, in practice it gives encouraging results even for complex non-convex objects, such as cars. Occlusion affects the accuracy of recognition.

7. REFERENCES

[1] M. Ankerst, G. Kastenm̀uller, H.-P. Kriegel, , and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Proceedings of 6th International Symposium on Spatial Databases*, pages 207–226, Hong-Kong, China, July 1999.

[2] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. *EUROGRAPHICS 2003*, 22(3):223–232, 2003.

[3] M. Elad, A. Tal, and S. Ar. Directed search in a 3D objects database using SVM. Technical report, HP Laboratories, Israel, 2000.

[4] M. Elad, A. Tal, and S. Ar. Content based retrieval of VRML objects - an iterative and interactive approach. In *The 6th Eurographics Workshop on Multimedia*, pages 97–108, Manchester, United Kingdom, Sept 2001.

[5] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics, Principles, and Practices.* Addison-Wesley Publishing Company, Reading, Mass., 1991.

[6] T. Funkhauser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22(1):83–105, January 2003.

[7] P.J. Besl H.D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.

[8] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH 2001*, pages 203–212, 2001.

[9] T. Hlavaty and V. Skala. A survey of methods for 3D model feature extraction. *Seminar on Geometry and Graphics in Teaching Contemporary Engineer*, 2003.

[10] I. Kolonias, D. Tzovaras, S. Malassiotis, and M. Strintzis. Fast content-based search of vrml models based on shape descriptors. In *Proceedings of International Conference on Image Processing 2001*, pages 133–136, Thessaloniki, Greece, October 2001.

[11] H. Kriegel, S. Brecheisen, P. Kröger, M. Pfeifle, and M. Schubert. Using sets of feature vectors for similarity search on voxelized cad objects. In *SIGMOD 2003*, pages 587–598, San Diego, CA, June 9-12 2003.

[12] T. Krüger, J. Wickel, P. Alvarado, and K. Kraiss. Feature extraction from vrml models for view-based object recognition. In *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services, 2003*, pages 391–394, London, April 9-11 2003.

[13] U. Labsik, R. Sturm, and G. Greiner. Depth buffer based registration of free-form surfaces. In *Proceedings of the 2000 Conference on Vision Modelling and Visualization (VMV-00), Saarbrücken, Germany*, pages 121–128, November 22-24 2000.

[14] S. Lao, M. Kawade, Y. Sumi, and F. Tomita. 3D template matching for pose invariant face recognition using 3D facial model built with isoluminance line based stereo vision. In *International Conference on Pattern Recognition*, pages 911–916, Barcelona, Spain, September 03-08 2000.

[15] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[16] M. Novotni and R. Klein. 3D Zernike descriptors for content based shape retrieval. In *The 8th ACM Symposium on Solid Modeling and Applications*, pages 216–225, Seattle, WA, June 16-20 2003.

[17] R. Ohbuchi, T. Minamitani, and T. Takei. Shape-similarity search of 3D models by using enhanced shape functions. In *Proceedings of Computer Graphics and Applications*, pages 293–302, 8-10 Oct 2003.

[18] R. Osada, T. Funkhauser, B. Chazalle, and D. Dobkins. Matching 3D models with shape distribution. *ACM Transaction on Graphics*, 21(4):807–832, October 2002.

[19] G. Papaioannou, E. Karabassi, and T. Theoharis. Reconstruction of three-dimensional objects through matching of their parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):114–124, January 2002.

[20] G. Passalis, I. A. Kakadiaris, and T. Theoharis. Efficient hardware voxelization. In *Proceedings of the Computer Graphics International*, Crete, Greece, June 16-19 2004.

[21] P. Siarry, G. Berthiau, F. Durbin, and J. Haussy. Enhanced simulated annealing for globally minimizing functions of many-continuous variables. *ACM Transactions on Mathematical Software*, 23(2):209–228, 1997.

[22] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings of Shape Modelling and Application Conference*, pages 130–142, Seoul, S. Korea, May 12-15 2003.

[23] M. T. Suzuki. A web-based retrieval system for 3D polygonal models. In *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pages 2271–2276, 2001.

[24] M. T. Suzuki, T. Kato, and N. Otsu. A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. In *Proceedings IEEE International Conference on Systems, Man, and Cybernatics (SMC2000)*, pages 2946–2952, 2000.

[25] J. Tangelder and R. Veltkamp. Polyhedral model retrieval using weighted point sets, 2003.

[26] R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999.

[27] D. Vraníc and D. Saupe. 3D model retrieval. In *Proceedings of Spring Conference on Computer Graphics and its Applications*, pages 89–93, Budmerice Manor, Slovakia, May 2000.

[28] D. Vranic and D. Saupe. Description of 3D-shape using a complex function on the sphere. In *Proceedings of International Conference on Multimedia Expo*, pages 177–180, Lausanne, Switzerland, August 2002.

[29] D. Vraníc, D. Saupe, and J. Richter. Tools for 3D object retrieval: Karhunen-loeve transform and spherical harmonics. In *Proceedings of the IEEE 2001 Workshop on Multimedia Signal Processing*, pages 293–298, Cannes, France, October 2001.

[30] T. Zaharia and F. Prteux. Shape-based retrieval of 3D mesh models. In *Proceedings of IEEE International Conference on Multimedia and Expo 2002*, pages 133–145, Lausanne, Switzerland, August 2002.

[31] C. Zhang and T. Chen. Indexing and retrieval of 3D models aided by active learning. In *Proceedings of ACM Multimedia 2001*, pages 615–616, Ottawa, Canada, 2001.