

Result Paper of Retrieval System of 3D Object

Bhupendra Namdev¹ Anamika Pyasi²

¹M. Tech Scholar

^{1,2}Department of Computer Science & Engineering

^{1,2}Babulal Tarabai Research and Technology, Sagar (M.P.), India

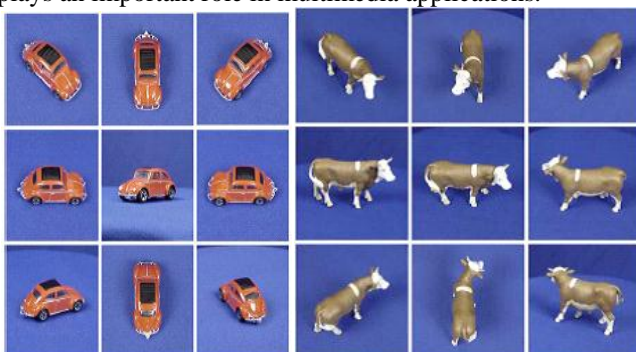
Abstract— On that time extensive research efforts have been dedicated to view-based methods for 3-D object retrieval in this paper 3-D object retrieval algorithm is proposed. 3-D object retrieval without the limitation of camera array restriction, A camera constraint-free view-based Method (CCFVM), The CCFVM removes the constraint of static camera array settings for view capturing and can be applied to any view-based 3-D object database The CCFVM is generated on the basis of the query Gaussian models by combining the positive matching model and the negative matching model. The CCFVM removes the constraint of static camera array settings for view capturing and can be applied to any view-based 3-D object ETH database and compare FCM and query Gaussian models.

Keywords: Camera Constraint-Free, Retrieval, 3-D Object, View-Based

I. INTRODUCTION

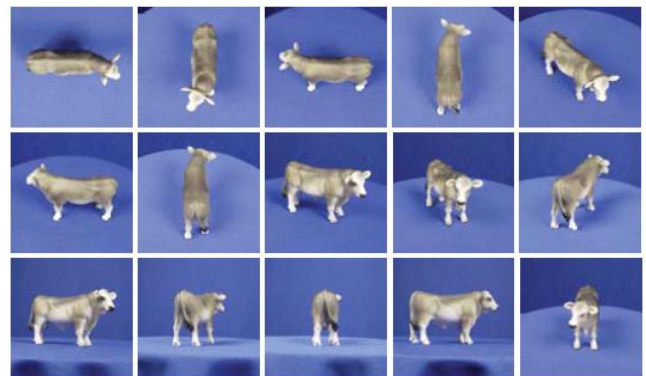
Today 3-D models have become very popular in many applications like manufacturing, medicine, entertainment etc. It leads to the urgent needs of efficient 3D object retrieval and recognition technologies [1]-[2].The advancement of modeling, visualizing and digitizing techniques for 3D shapes has led to an increasing amount of 3D models, both on the Internet and domain-specific databases. This has led to the development of the first experimental search engines for 3D shapes, Just like Ephesus search engine at the National Research Council of Canada.

3D models are not easily retrieved. Attempting to find a 3D model using conventional text-based and textual annotation search engine would not work in some cases. The annotations added by human beings depend on language, culture, age, sex, and other factors. They may be too limited or ambiguous. In contrast, content based 3D shape retrieval methods, which use shape properties of the 3D models to search for similar models, work better than text based methods. Recently, extensive research efforts have been dedicated to View-based 3-D object retrieval methods [3]–[4] because of the highly discriminative property of multiviews for 3-D object Representation [5], [6].Visual analysis also plays an important role in multimedia applications.



(a)

(b)



(c)

Fig. 1.1: Three Object examples described by multiple views.

Object examples, which are described by multiple views. Three-dimensional object representation by views can reduce computational the significantly complexity matching of 3-D object and also avoids the need for a model parameter estimation and 3-D model reconstruction. Different view-based 3-D object retrieval methods, e.g., light-field descriptors (LFDs) [7], elevation Descriptors (EDs) [8], bag of visual features (BoVF) [5], [9], and compact multiview descriptors (CMVDs) [10], have been proposed these years. These view-based 3-D object retrieval methods share a common advantage that is invariant to articulation and global deformation of the 3-D models. However, due to the different 3-D object analysis methods, most of the art-of-the- state approaches have their own constraints of camera array settings for capturing views of 3-D objects. For that time, the views for the LFD were captured by a group of cameras set on the vertices of a dodecahedron over a hemisphere. For the CMVD, the camera arrays were set at the 18 vertices of a 32-hedron to capture views. The camera array restrictions limit the application of these approaches when the view sets for 3-D objects cannot satisfy these requirements. In order to move toward a general framework for 3-D object retrieval without the limitation of camera array constraints, we propose a camera constraint-free view-based Method (CCFVM) 3-D Object retrieval algorithm, which requires no camera constraint for view capturing of all 3-D objects. In this framework, all 3-D Query object can be described by a set of views from any direction. For the query object, we cluster the entire query object (view) to generate the view clusters and then use these view clusters to build the query model. For a more accurate 3-D object comparison, a negative matching model and a positive Matching model are individually trained using Negative and positive matched samples, respectively. The query model generated the CCFVM model are work on combining the positive Matching model and the negative matching model.

The advantages of the proposed method are twofold.

- 1) CCFVM removes the constraint of static camera array settings for view capturing and can be applied to any

view based.3-D object database. In CCFVM, any set of views for One 3-D object can be used for 3-D object retrieval.

- 2) CCFVM Improve the matching accuracy by using of positive matching model and the negative matching model for 3-D object analysis.

The rest of this paper is organized as follows. We first introduce the related work in Section II. The proposed

CCFVM framework for 3-D object retrieval is discussed in Section III. Section IV presents the experimental results and comparison with other methods to demonstrate the effectiveness of the proposed method. Finally, the conclusion and the future work are given in Section V.

II. SECTION II

In order to move toward a general framework for 3-D object retrieval without the limitation of camera array constraints, a camera constraint-free view-based (CCFVM) 3-D object retrieval algorithm is used, which requires no camera constraint for view capturing of all 3-D objects. The architecture of the previous CCFVM is shown in the figure below.

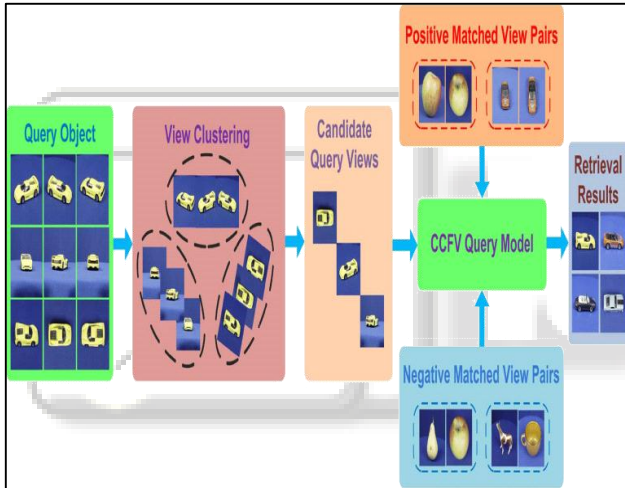


Fig. 2.1: Architecture of the previous CCFVM 3D object retrieval algorithm

A. Work Formulation

The work can be illustrated as follow. Each 3-D object is represented by its view set captured from any camera array. These views are used to convey the 3-D structure information through the relationships among such views. The view-based 3-D object retrieval depends on the analysis of these view sets. Given the query object, the retrieval task is to find the matched/similar objects from the 3-D object database. The key for 3-D.Object retrieval is to find the relationship between the query and the objects in the database.

Let the view set of query object are denoted $\mathcal{V}^Q = \{v_1^Q, v_2^Q, \dots, v_m^Q\}$

The view set of database object are denoted as:

$$v^m = \{v_1^m, v_2^m, \dots, v_n^m\}$$

Then the similarity between Q & M is defined as:

$$S(Q,M)=p(M|Q,\Delta=1)-p(M|Q,\Delta=0)$$

Where:

$p(M|Q,\Delta=1)$ denotes that the probability of M on Q when M is relevant to Q

$p(M|Q,\Delta=0)$ denotes the probability of M on Q when M is not relevant to Q .

B. View Clustering

As stated earlier, the view set of query object are denoted as:

$$\mathcal{V}^Q = \{v_1^Q, v_2^Q, \dots, v_m^Q\}$$

In this approach, the Zernike moment descriptor is employed to model the views. Zernike moments have been demonstrated to be robust to image translation, scaling, and rotation, and applied in many 3-D model/object analysis tasks. The Zernike moment descriptor contains 49 coefficients of Zernike moments of images. Here, each view is represented by a moment feature vector, and the corresponding feature set is:

$$\{\psi_1^Q, \psi_2^Q, \dots, \psi_m^Q\}$$

The view set of database object are denoted as:

$$\mathcal{V}^M = \{v_1^M, v_2^M, \dots, v_n^M\}$$

And the corresponding feature vector is:

$$\{\psi_1^m, \psi_2^m, \dots, \psi_n^m\}$$

A Gaussian model is learned to model the feature distribution in each cluster. Let X is the feature of the training view;

Then the model can be defined as:

$$p(x|\theta)=\sum_{i=1}^{n_{gm}} w_i g_i(x|u_i, \sigma_i^2)$$

Where:

$g_i(x|u_i, \sigma_i^2)$ denotes the i th Gaussian component, w_i indicates the weight of the i th Gaussian component, n_{gm} is the number of Gaussian models.

C. Positive and Negative Matching

Two matching views are used concerning two feature vectors from 3-D objects. When $\Delta=1$ or $\Delta=0$, the probability of these two views is matched or not matched and can be formulated using a probabilistic framework.

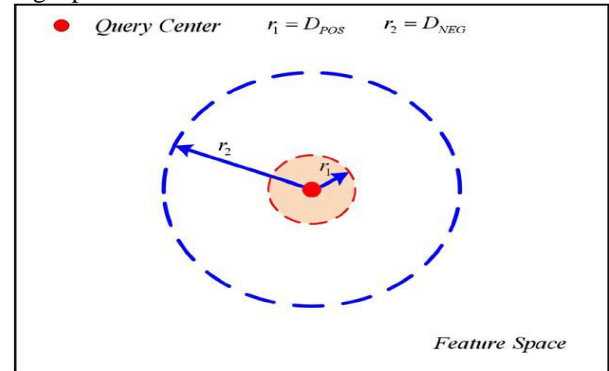


Fig. 2.2: Illustration for positive and negative matched samples in the feature space.

The central red point is the query center. When another view locates inside of the red circle (the radius is r_1), it must be a matched view of the query center. While when another view locates outside of the blue circle (the radius is r_2), it cannot be a matched view of the query center because of the large distance from the query center. The average distances in the two training sets (D_{POS}, D_{NEG}) can be calculated by:

$$D_{POS} = \frac{1}{n_{pos}} \sum_{k=1}^{n_{pos}} \{d[v_{pos}(k, 1), v_{pos}(k, 2)]\}$$

$$D_{NEG} = \frac{1}{n_{neg}} \sum_{k=1}^{n_{neg}} \{d[v_{neg}(k, 1), v_{neg}(k, 2)]\}$$

III. PROPOSED FRAMEWORK OF CCFVM

In order to improve the performance of the previous camera constraint free view based 3D retrieval, it is proposed to use an efficient clustering method called Fuzzy C-Means clustering. First the features of image are extracted and pdf is formed, cloning and KL distance will be used to find out better matching between the query and database image. The total process is divided into following steps as described below.

A. Query Processing

Each 3-D object is represented by its view set captured from any camera array. These views are used to convey the 3-D structure information through the relationships among such views. The view-based 3-D object retrieval depends on the analysis of these view sets. Given the query object Q, the retrieval task is to find the matched/similar objects from the 3-D object database. The key for 3-D object retrieval is to find the relationship between the query and the objects in the database. The algorithm architecture is given in Fig.

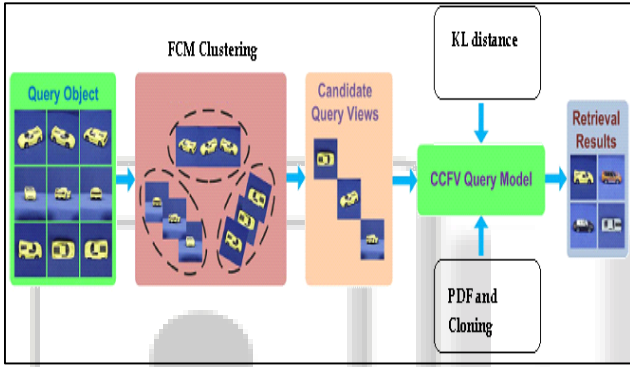


Fig. 3.1: Architecture of the proposed CCFVM 3D object retrieval algorithm

The proposed algorithm consists of the following procedures. First, the views of the query object are grouped into clusters, and query models are trained in these clusters. We also train a positive matching model and a negative matching model using positive matched samples and negative matched samples.

B. Fuzzy C-Means Clustering Fuzzy

C-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2 \quad 1 \leq m < \infty$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m}$$

This iteration will stop when

$\max_{ij} \left\{ |u_{ij}^{(k+1)} - u_{ij}^{(k)}| \right\} < \epsilon$ Where ϵ is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m . The algorithm is composed of the following steps: Initialize $U=[u_{ij}]$ matrix, $U^{(0)}$

At k-step: calculate the centers vectors $C^{(k)}=[c_j]$ with $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m}$$

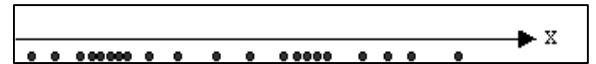
Update $U^{(k)}, U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

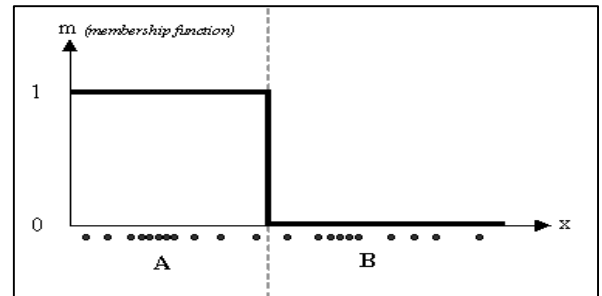
If $\|U^{(k+1)} - U^{(k)}\| < \epsilon$ then STOP; otherwise return to step 2.

As already told, data are bound to each cluster by means of a Membership Function, which represents the fuzzy behavior of this algorithm. To do that, we simply have to build an appropriate matrix named U whose factors are numbers between 0 and 1, and represent the degree of membership between data and centers of clusters.

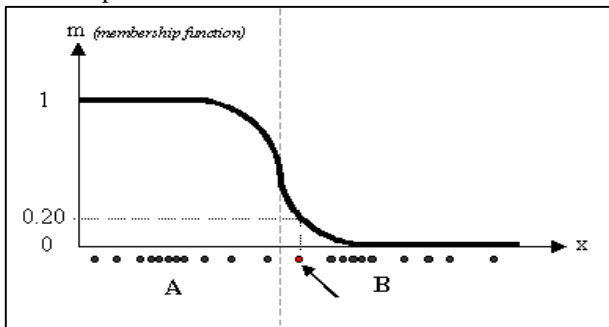
For a better understanding, we may consider this simple mono-dimensional example. Given a certain data set, suppose to represent it as distributed on an axis. The figure below shows this:



Looking at the picture, we may identify two clusters in proximity of the two data concentrations. We will refer to them using 'A' and 'B'. In the first approach shown in this tutorial - the k-means algorithm - we associated each datum to a specific centroid; therefore, this membership function looked like this:



In the FCM approach, instead, the same given datum does not belong exclusively to a well-defined cluster, but it can be placed in a middle way. In this case, the membership function follows a smoother line to indicate that every datum may belong to several clusters with different values of the membership coefficient.



In the figure above, the datum shown as a red marked spot belongs more to the B cluster rather than the A cluster. The value 0.2 of 'm' indicates the degree of membership to A for such datum. Now, instead of using a graphical representation, we introduce a matrix U whose factors are the ones taken from the membership functions:

$$U_{N \times C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \dots & \dots \\ 0 & 1 \end{bmatrix} \quad U_{N \times C} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ \dots & \dots \\ 0.9 & 0.1 \end{bmatrix}$$

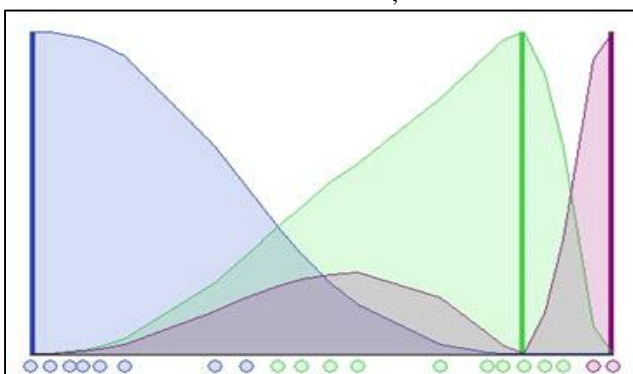
(a) (b)

The number of rows and columns depends on how many data and clusters we are considering. More exactly we have $C = 2$ columns ($C = 2$ clusters) and N rows, where C is the total number of clusters and N is the total number of data. The generic element is so indicated: u_{ij}

1) Example

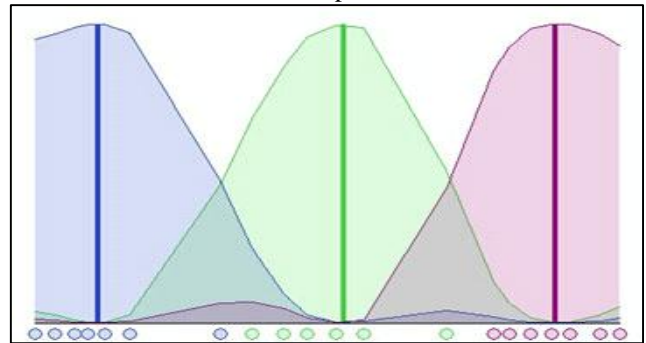
Here, we consider the simple case of a mono-dimensional application of the FCM. Twenty data and three clusters are used to initialize the algorithm and to compute the U matrix. Figures below (taken from our interactive demo) show the membership value for each datum and for each cluster. The color of the data is that of the nearest. Cluster according to the membership function.

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m < \infty$$

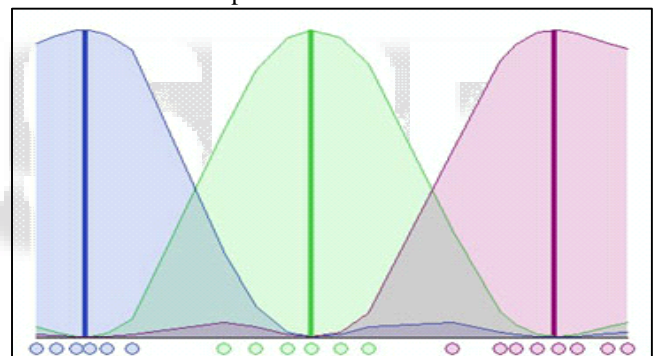


In the simulation shown in the figure above we have used fuzziness' coefficient $m = 2$ and we have also imposed to terminate the algorithm. When $\max_y \{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} < 0.3$.

The picture shows the initial condition where the fuzzy distribution depends on the particular position of the clusters. No step is performed yet so that clusters are not identified very well. Now we can run the algorithm until the stop condition is verified. The figure below shows the final condition reached at the 8th step with $m=2$ and $\epsilon = 0.3$:



Is it possible to do better? Certainly, we could use a higher accuracy but we would have also to pay for a bigger computational effort. In the next figure we can see a better result having used the same initial conditions and $\epsilon = 0.01$, but we needed 37 steps!



It is also important to notice that different initializations cause different evolutions of the algorithm. In fact it could converge to the same result but probably with a different number of iteration steps.

C. Data Cloning

We use a new statistical computing method, called data cloning, to calculate maximum likelihood estimates and their standard errors for complex ecological models [11]. The idea is simple: construct a full Bayesian model of the problem, complete with fully specified, proper prior distributions for unknown parameters, but instead of using the likelihood for the observed data, use the likelihood corresponding to k copies (clones) of the data, where k is large and the copies are assumed to be independent of each other. The posterior is then calculated and the mean of the resulting posterior distribution equals the ML estimate, and k times the variance of the posterior equals the asymptotic variance of the ML estimate. fig.3.3 shows the example of data cloning with discrete attributes [12].

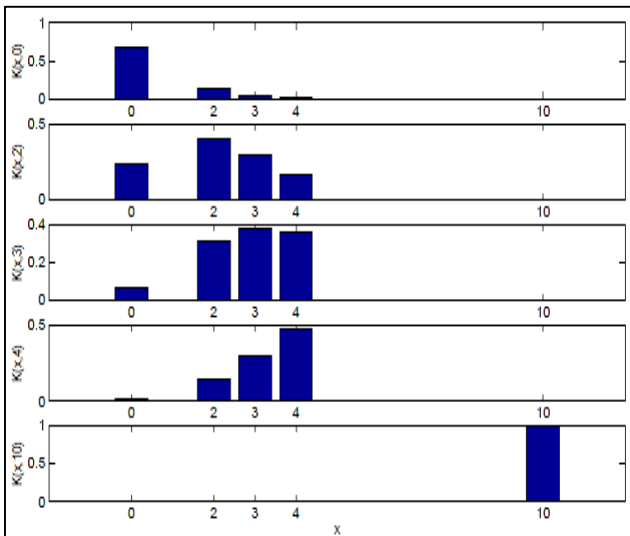


Fig. 3.3: a family of discrete smoothing kernels. The values of are shown for.

The cloning is used for the PDF of the query and the matching database models and best matching is obtained using KL distance concept.

D. KL Distance

The Kullback Leibler distance (KL-distance) is a natural distance function from a "true" probability distribution, p , to a "target" probability distribution, q . It can be interpreted as the expected extra message-length per datum due to using a code based on the wrong (target) distribution compared to using a code based on the true distribution. For discrete (not necessarily finite) probability distributions $p = \{P_1, \dots, P_n\}$ and $q = \{q_1, \dots, q_n\}$, the KL-distance is defined to be:

$$KL(p, q) = \sum_i p_i \cdot \log_2 \left(\frac{p_i}{q_i} \right)$$

For continuous probability densities, the sum is replaced by an integral.

$$KL(p, q) \geq 0$$

$$KL(p, q) \geq 0$$

Note that the KL-distance is not, in general, symmetric. Inwards, it is the expectation of the logarithmic difference between the probabilities P and Q , where the expectation is taken using the probabilities P . Equivalently, this can be written as.

$$D_{KL}(P||Q) = \int_X \ln \left(\frac{dP}{dQ} \right) \frac{dP}{dQ} dQ,$$

Which we recognize as the entropy of P relative to Q . Continuing in this case, if μ is any measure on X for which $P = \frac{dP}{d\mu}$ and $q = \frac{dQ}{d\mu}$ exist, then the Kullback-Leibler divergence from P to Q is given as.

$$D_{KL}(P||Q) = \int_X p \ln \frac{p}{q} d\mu.$$

The logarithms in these formulae are taken to base 2 if information is measured in units of bits, or to base e if information is measured in nats. Most formulae involving the KL divergence hold irrespective of log base.

Various conventions exist for referring to $D_{KL}(P||Q)$ in words. Often it is referred to as the divergence between P and Q ; however this fails to convey the fundamental asymmetry in the relation. Sometimes it may be found described as the divergence of P from, or with respect

to Q (often in the context of relative entropy, or information gain). However, in the present article the divergence of Q from P will be the language used, as this best relates to the idea that it is P that is considered the underlying "true" or "best guess" distribution, that expectations will be calculated with reference to, while Q is some divergent, less good, approximate distribution.

IV. RESULTS

A. Experimentation

The thesis work is performed in MATLAB 7.10.0, following figures represents some of the screenshots of the experimentation. The basic graphical user interface (gui) environment is shown in fig 5.1. It consists of input panel, training database path, training panel and search panel.

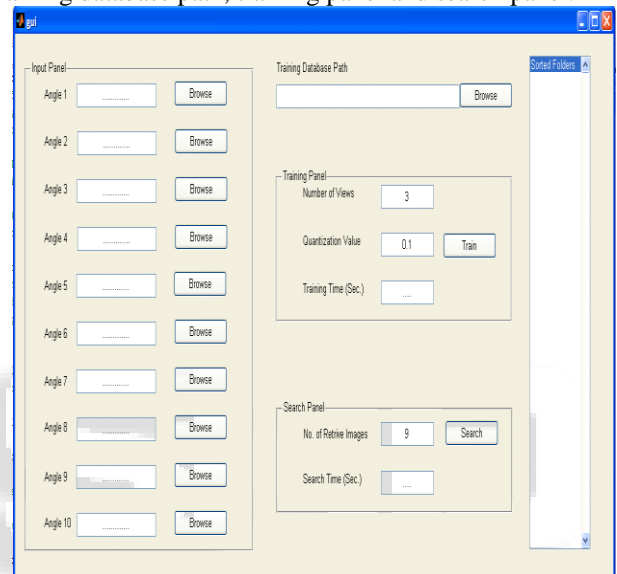


Fig. 5.1: Basic Graphical User Interface

The path for the database is required to be given through training database path by browsing the folder where the database is stored. The training panel is added to select the number of views and quantization value. Once these two parameters are provided and train is depressed, the training starts. The time required for training is indicates in training time after completion of training also the selected folders are made available for users' reference under title "selected folders". The input panel provides facility of providing different views. In the setup total 10 angles can be provided to select total ten views for the query. Once the query is browsed, the searching can be started. The search panel gives the information regarding number of total retrieved results; also the total time required for searching in indicated which can be further used for comparative study of the proposed model with other models. Fig. 4.2 gives the snapshot for the results when object "boat" was searched with three angled input.

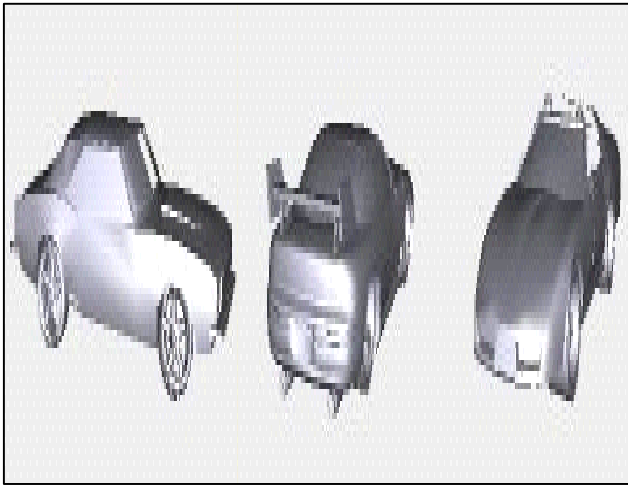


Fig. 4.2: Query object

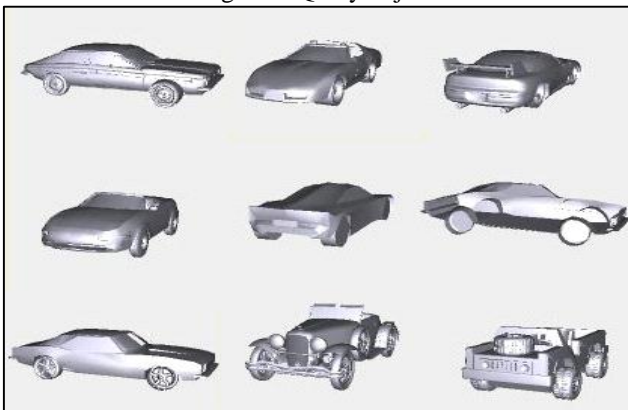


Fig. 5.2: results for object "car" with 3 view angles

It is clear from the results that depending upon the given inputs; appropriated results are made available at output. Some other images from the NTU database are as shown in fig below.

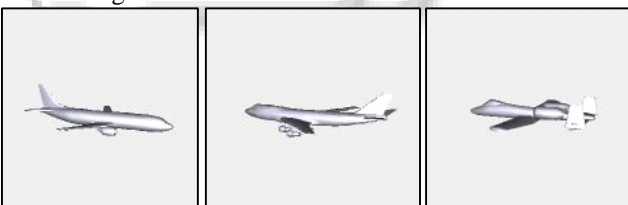


Fig. 5.3: (a) images for the objects in NTU database

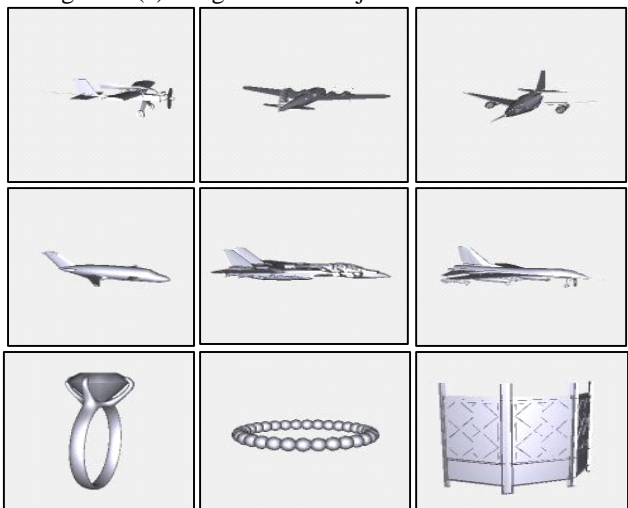


Fig. 5.3: (b) images for the objects in NTU database

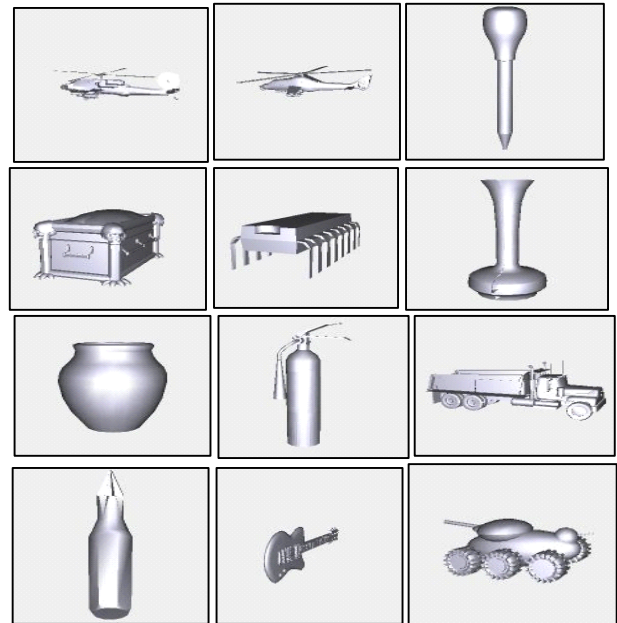


Fig. 5.3: (c) images for the objects in NTU database

The experimental results are made available here in the form of tables and graph, gives detail information about the processing time, accuracy and comparative idea for the future work. Table 5.1 gives the results for the training and the searching time required for the different number of views.

No. of Views	Training Time (Sec.)	Searching Time (Sec.)			
		4	9	16	25
2	21.096	0.01703	0.0183	0.0196	0.0208
3	18.737	0.02555	0.0264	0.0275	0.0283
4	13.577	0.03371	0.0345	0.0358	0.0372
5	12.696	0.04410	0.0450	0.0465	0.0488
6	10.799	0.05160	0.0520	0.0539	0.0548
7	9.991	0.06326	0.0649	0.0668	0.0690
8	9.897	0.07476	0.0756	0.0786	0.0826

Table 5.1: results of training and searching time required with the proposed method for different no. of views.

The training as well as searching time increases with the number of views. Results for the precision, recall and F-Measure are represented by tables 5.2, 5.3 and 5.4 respectively.

Number of Views	Precision			
	4	9	16	25
2	0.33	0.44	0.375	0.36
3	0.5	0.667	0.625	0.60
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1

Table 5.2: results of precision with the proposed method for different no. of views

Number of Views	Recall			
	4	9	16	25
2	0.016	0.065	0.098	0.147
3	0.008	0.098	0.164	0.245
4	0.065	0.147	0.262	0.410
5	0.065	0.147	0.262	0.410

6	0.065	0.147	0.262	0.410
7	0.065	0.147	0.262	0.410
8	0.065	0.147	0.262	0.410

Table 5.3: results of recall with the proposed method for different no. of views

Number of Views	F-Measure			
	4	9	16	25
2	0.031	0.113	0.155	0.209
3	0.016	0.196	0.260	0.347
4	0.128	0.256	0.415	0.582
5	0.128	0.256	0.415	0.582
6	0.128	0.256	0.415	0.582
7	0.128	0.256	0.415	0.582
8	0.128	0.256	0.415	0.582

Table 5.4: results of F-measure with the proposed method for different no. of views.

Number of Query View	Proposed Method	Previous Method
2	0.020824	0.125
3	0.028359	0.130
4	0.037214	0.137
5	0.048882	0.150
6	0.054801	0.175
7	0.069063	0.200
8	0.082636	0.225

Table 5.5: Retrieval Time (in seconds) Comparison between Previous and Proposed System

The comparative results of the retrieval time in seconds for the proposed and the previous method is represented in table 5.5. The graphical representation of recall vs. precision of the proposed and the previous methods is plotted in fig5.2.

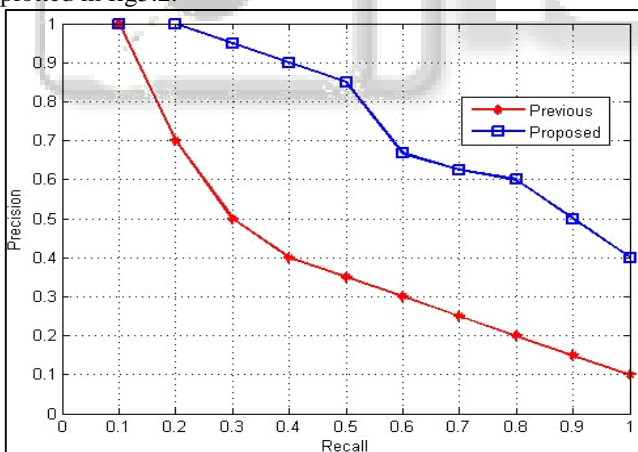


Fig. 5.2: Precision vs. Recall Comparison for Previous and Proposed System

V. CONCLUSION

The CCFVM can search 3-D objects with the query of any view set captured by any camera arrays. That is, no camera constraint is required. The speed of Proposed CCFVM is more than the existing methods of retrieval also accuracy is more due to used of extended hierarchical matching. The concept of cloning results in better matching of the query images; use of FCM clustering is helpful to identify the objects belonging to more than one set. Proposed CCFVM

still relies on the static query views. However, for real-world applications, users cannot capture too many query views for objects. Thus, In the future work, we can focus on designing a method to automatically suggest new best query views for users according to the existing query views.

REFERENCES

- [1] Bimbo and P. Pala, "Content-based retrieval of 3D objects," ACM Trans Multimedia Computing Communication and Appl., vol. 2no. 1, pp.20-43 2006.
- [2] Yue Gao, Jinhui Tang, Richang Hong, Shuicheng Yan, Qionghai Dai, Naiyao Zhang, and Tat-Seng Chua, "Camera Constraint-Free View-Based 3-D Object Retrieval," IEEE Transactions on Image Processing, vol. 21, no. 4, April 2012, pp.2269-2281.
- [3] F. Li, Q. Dai, W. Xu, and G. Er, "Weighted subspace Distance and its application to object recognition and Retrreval with image sets," IEEE Signal Process Lett., vol. 16, no. 3, pp. 227-230, Mar.2009.
- [4] Y. Gao, M. Wang, J. Shen, Q. Dai, and N. Zhang, "Intelligent query:Open another door to 3-D object retrieval," in Proc. ACM Conf. Multimedia,2010, pp. 1711-1714.
- [5] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno, "Salient local visual features for shapebased 3-D model retrieval," in Proc. IEEE Conf.Shape Model. Appl., 2008, pp. 93-102.
- [6] R. Ohbuchi and T. Furuya, "Accelerating bag-of-features sift algorithm for 3-D model retrieval," in Proc. SAMT Workshop Semantic 3-D Media, 2008, pp. 22-30.
- [7] D. Y. Chen, X. P. Tian, Y. T. Shen, and M. Ouhyoung, "On visual similarity based 3-D model retrieval,"Comput. Graph. Forum, vol. 22, no. 3, pp. 223-232, Sep. 2003.
- [8] J. L. Shih, C. H. Lee, and J. T. Wang, "A new 3-D model retrieval approach based on the elevation descriptor," Pattern Recognit., vol. 40, no. 1, pp. 283-295, Jan. 2007.
- [9] R. Ohbuchi and T. Furuya, "Scale-weighted dense bag of visual features for 3-D model retrieval from a partial view 3-D model," in ProcIEEE ICCV Workshop S3DV, 2008, pp. 63-70.
- [10] P. Daras and A. Axenopoulos, "A 3-D shape retrieval framework supporting multimodal queries," Int. J. Comput. Vis., vol. 89, no. 2/3, pp. 229-247, Sep. 2010.
- [11] M. Steinbach, G. Karypis, and V. Kumar, "Acomparison of document clustering techniques," in Proc. KDD Workshop TextMining, 2000, pp. 1-15.
- [12] J.C. Bezdek, R.T.Ehrlich,W.Full, "The Fuzzy C-Means Clustering Algorithm," computers and geosciences vol 10 no 2-3,pp.191-203,1984.