Visual Semantic Based 3D Video Retrieval System Using HDFS

C.Ranjith Kumar[1] and S.Suguna[2]
[1]Research Scholar, Bharathiar University
Coimbatore, TN, India
[e-mail: ranjithnetm@gmail.com]
[2]Assistant Professor, Dept of Computer Science, Sri Meenakshi Govt Arts College
Madurai District, India
[e-mail: kt.suguna@gmail.com]
*Corresponding author: C.Ranjith Kumar

Received January 25, 2016; revised May 4, 2016; accepted May 23, 2016; published August 31, 2016

Abstract

This paper brings out a neoteric frame of reference for visual semantic based 3d video search and retrieval applications. Newfangled 3D retrieval application spotlight on shape analysis like object matching, classification and retrieval not only sticking up entirely with video retrieval. In this ambit, we delve into 3D-CBVR (Content Based Video Retrieval) concept for the first time. For this purpose we intent to hitch on BOVW and Mapreduce in 3D framework. Here, we tried to coalesce shape, color and texture for feature extraction. For this purpose, we have used combination of geometric & topological features for shape and 3D co-occurrence matrix for color and texture. After thriving extraction of local descriptors, TB-PCT (Threshold Based- Predictive Clustering Tree) algorithm is used to generate visual codebook. Further, matching is performed using soft weighting scheme with L_2 distance function. As a final step, retrieved results are ranked according to the Index value and produce results .In order to handle prodigious amount of data and Efficacious retrieval, we have incorporated HDFS in our Intellection. Using 3D video dataset, we fiture the performance of our proposed system which can pan out that the proposed work gives meticulous result and also reduce the time intricacy.

***Keywords*: BOVW, PCT, HDFS, Video Retrieval, Local Descriptors**

http://dx.doi.org/10.3837/tiis.2016.08.021 **ISSN : 1976-7277**

1. Introduction

Content based video retrieval (CBVR) plays an imperative role in the field of multimedia retrieval applications. In [1], the concept of STAR is being used for an efficient video retrieval. Our modern life application usage like face book, and many other social media involves large scale multimedia files sharing process like images, video, audio etc [9]. To cope up with such an enormous multimedia file sharing application in an efficient manner, creates a challenging issues to the mankind. To fulfill this ulterior motive, a new programming paradigm namely Map Reduce is required [6]. A series of operations is performed, each of which consists of Map and Reduce stage to execute an enormous number of independent data items [12]. Simultaneously, there is a rapid increase to improve the efficiency of 3D retrieval application [4]. Applications of the 3D models include medical imaging, films, industrial design and gaming. But it is basically focused on object matching, classification and retrieval [2]. For this purpose, various strategies and methodologies are implemented in [4] [2] [5]. In [17], there is a detail explanation about shape matching. Using frame conversion a tool, the video is divided into number of frames it has several redundant frames, to ignore those frame we performed process of key frame selection [20]. To enhance the performance, in terms of accuracy and time complexity of 3D model retrieval application, we need to bring in new mechanisms.

One of the effective data mining algorithms is BoF (Bag of Feature) extraction procedure mainly used for text classification and retrieval purposes. Compare with other approaches it is conceptually simple, computationally cheaper and bestow better results. This algorithm mostly used to generate visual vocabulary. Here we signify features as color, texture and shape. Combination of color and shape feature for 3D model is explained in [21]. Feature vector is used to map visual vocabulary to key point by counting the number of occurrences of the word in each sentence. For the purpose of vocabulary construction, algorithms like clustering [2], [3], [9] and [10] tree formation algorithms [15] can be used. To carry out matching and classification, supervised learning approaches [27], [2] are used. The proposed perception of Map Reduce based BOF retrieval, use soft weighting scheme [10] or pyramid matching methodologies to accomplish the purpose of matching. Conclusively, the notion is to implement BOVW (Bag of Visual Words) in 3D model retrieval application to get fruitful results [2], [4]. Search engines for 3D models are discussed in [7]. To attain scalability and handle huge amount of data in an efficient way combination of BoF with Map Reduce framework is introduced in [3], [9], [12] and [13]. Survey of the 3D model retrieval applications are presented in [25] and [26]. The aspiration of this article is to develop 3D CBVR (Content Based Video Retrieval) application using Bag of Visual Words (BoVW) in Map Reduce framework. Main contribution of our proposed framework includes,

- ➢ Newfangled approach of bringing 3D CBVR (Content Based Video Retrieval) application in Map reduce framework using Bag of Visual Words (BoVW) methodology.
- ➢ BOVW is applied on the selected key frames and local descriptors are used for extracting features. For shape, combined feature (topology & geometric) is used and for color & texture 3D co-occurrence matrix is being used.
- ➢ Fork Strategy based Medial Surface Extraction is a novel method used for shape extraction without affecting its geometry.

➢ To generate visual vocabulary, TB-PCT (Threshold Based – Predictive Clustering Tree) is used. To accomplish the purpose of matching, soft weighting scheme with L_2 distance function is considered as an innovative approach.

➢ Construction of Visual words sometimes causes noise due to the occurrence of repeated visual words, in order to overcome it, we have introduce a Numerical Semantic Analysis.

Above mentioned all information's are thrashed out in detail in the upcoming sections. Section 2 contains related works and its detailed explanation. Section 3 deals with the proposed Map Reduce framework with 3D CBVR (Content Based Video Retrieval) application. Section 4 details about the experimental result including comparisons with prior approaches. Finally, Section 5 concludes this paper.

2. Related Work

In past decades, several approaches have done for efficient retrieval of video and Images. But it provided less effective results. [27] Discuss about video instance retrieval (*for 2-dimension*) using q-MIL (q- Multiple Instance Learning) technique. Several multimedia applications for 3D shape retrieval, classification and matching processes are discussed in [2]. Survey of 3D shape retrieval methods are thrashed out in [25] [7], which considered feature based methods, graph based methods and other methods.

CBIR is brought into Hadoop MapReduce framework for reducing the processing time. In [32] feature extraction is performed by color indexing technique but ,here color is the only feature concentrated for CBIR. In [33] a detail about Auto Color Correlogram and Correlation (ACCC) technique which is used to reduce load balance. It works by splitting the input and each input is assigned with a map task. If the input is large, it works well and could provide better result but if the number of splits are less, it could lead to the situation of overhead.

For 3D model retrieval applications, many state-of-the-art techniques are illustrated in [4], [5] and [21]. Shape is used as a basic feature to produce an efficient result for Bag-of-View-Words. [4] Explains the generation of codebook using k-mean algorithm and performs matching using histogram. In [5] classification based 3D model retrieval is furnished by combining the concept of multiple SVM classifiers with D-S evidence theory. This approach does not focus on the way to establish reasonable measures to evaluate the classifiers. On the process of 3D model retrieval we had to perform combined feature extraction [21]. Shape & color features are extracted based on D2 shape descriptor and color similarity measurements.

Some other feature extraction procedures are discussed in [17], [18], [23], [28] and [29]. Geometric and topology features are combined in [17] and discussed the reason of combining both of them. Remaining papers [18], [23], [28] and [29] fully consider texture features and combination of color features. Authors of [22] discussed about codebook generation process in depth with the help of clustering. This paper portrays about k-means clustering algorithm, grouping similar values based on centroid value. Another superior form of clustering method includes randomized forests clustering [16]. It is appropriate only for image recognition, object detection and segmentation process.

Krisch descriptor based video retrieval process explained in [19], before this paper, key frame extraction method have been proposed in [24], [20] and [30]. [24] Details about the key frame extraction method for the purpose of video segmentation. It deals with a set of rules to haul out key frames from shot types. In [20], detail explanation about video segmentation is provided and [30] introduced the novel key frame extraction process based on similarity measure. Major drawback of that kind of process is to take more time for graph construction and key frame selection. Content based image retrieval based on Bag –of-feature is clearly explained in [13]. It provides adequate result but it does not ensure enough heterogeneity for large clusters. Survey paper [26], confer about several 3D CBOR methods.

In paper [34] author has thrashed out the technique of unsupervised feature selection using high dimensional data. For unsupervised feature selection process, combined version of structural analysis and cluster analysis named as clustering-guided sparse structural learning (CGSSL) was used. A technique like explicit feature correlation matching was not discussed in the paper and it does not provide efficient result. In [35], authors discussed about semantic based data representation, which uses the technique of robust structured subspace learning to guarantee the subspace using L2,1-norm. It also considers local and global structural consistencies and is robust against noise and outliers. Hence, the proposed subspace learning algorithm is slightly effective to uncover the latent subspace, which is used for the purpose of image understanding.

3 Proposed System

In this field of reference, we are going to capsulate our overall proposed work. **Fig. 1** presents the overview of the proposed work. We have partite our work into two modes namely offline and online mode. Admin does all works in offline mode (O_f) while user works in online mode (O_n). In order to handle huge amount of videos as well as make the result more efficient and efficacious, we have included the notion of Hadoop MapReduce. Overall working procedure of MapReduce in retrieval application and it also by dividing each process into Map and Reduce functions.

As an initial step to our notion, Videos uploaded by the admin is converted into frames via frame conversion tool and frames are considered as dataset. From the entire frame sequence, we need to select a particular key frame. For this purpose, we have used TMOF (Temporally Maximum Occurrence Frame) [20]. On the selected key frames, BOVW (Bag of Visual Word) is applied to extract the local descriptors like shape, color and texture. Integrating the concept of geometric and topological features for shape and 3D co-occurrence matrix is used for color & texture so that we have given a new direction to our proposed work.

As further step to our process, we need to construct the visual vocabulary. To the resultant, TB-PCT (Threshold Based-Predictive Clustering Tree) is used for visual codebook generation. Then count the number of occurrence of visual word from the vocabulary and compute the occurrence value. As a final step to the above process generate a histogram. On the other side, user provides a query image in the online mode and BOVW is computed.

Next, matching should be performed between the query image and resultant dataset of our proposed work. To make the matching process more Efficacious, we came out with an ingenious idea of soft weighting scheme with L_2 distance function. Finally, Index value is used to rank the matched image and is returned to the user. Further subsections are noted detailed information's of our proposed system.

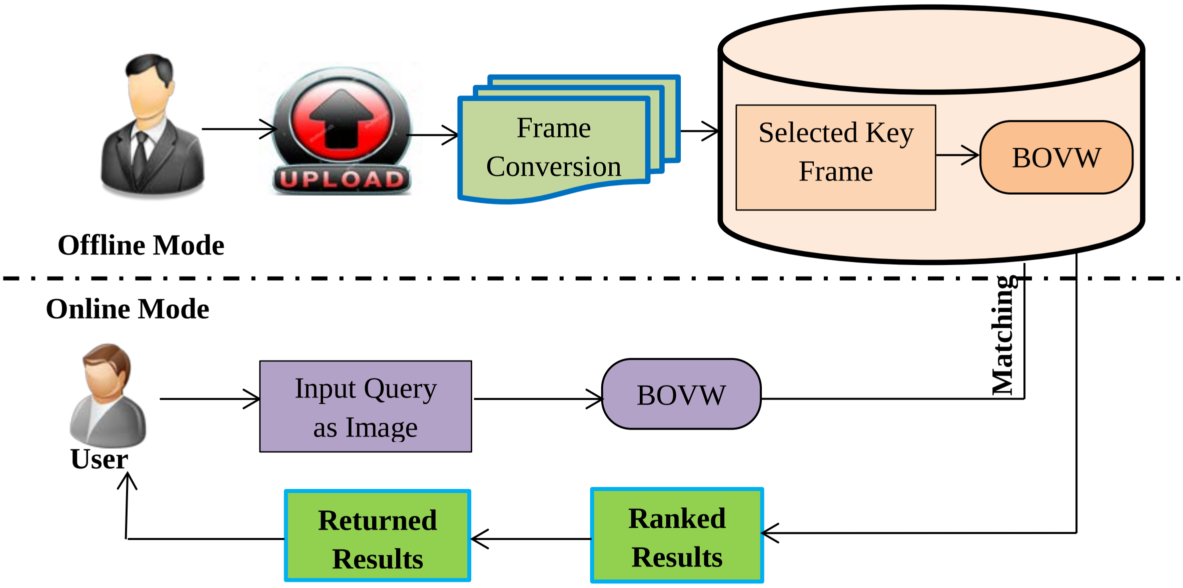

Fig. 1. Overview of proposed system

3.1 Key Frame Selection

Key frame selection plays a vital role in video retrieval application as it represents the whole video content. Several key frame selection approaches were discussed in the related work but these approaches overlook some temporal information. To overcome this problem, we have suggested optimal key frame selection procedure by TMOF in [20] in our proposed work. First step towards our work is to provide video as input to the computer nodes. Then this video is converted into frames using frame conversion tool and from the frames, we select key frame using TMOF to corresponding Map task. The output frames are saved in HDFS.TMOF is constructed based on the probability of occurrence of pixel value at each pixel position for all the frames. The outcome from TMOF is the maximum occurred frames which have multiple entities. From the Group of Frames (GoF), histogram is formed based on the pixel values at each corresponding pixel position. Gaussian function is applied to smooth the histogram. Pixels of the histograms are computed using the equation as follows,

$$\text{TMOF}\ (a,b,c) = b_{opt}, \quad 0 \leq a \leq I'-1 \text{ and } 0 \leq b \leq J'-1 \text{ and } 0 \leq c \leq K'-1 \qquad (1)$$

Where I'*J'*K' is the size of a TMOF, I, J, K represents the Width, Height, Depth and b_{opt} is chosen as follows:

$$b_{opt} = \arg\max \{K'_{a,b,c}(b)\}, \text{for } 0 \leq b \leq B \qquad (2)$$

By using Gaussian filter, smoothed histogram is determined as follows,

$$K'_{a,b,c}(b) = K_{a,b,c}(b) * G(\sigma,b) \qquad (3)$$

Where G(σ,b) is a Gaussian function with variance σ

$$K_{a,b,c}(b) = \sum_{n=0}^{N\text{-}1} \delta(f_n(a,b,c)\text{-}b), \quad \text{for } 0 \leq b \leq B$$
$$(m\text{-}n) = \begin{cases} 1, & for\ m = n, \\ 0, & for\ m \neq n. \end{cases} \tag{4}$$

$K_{a,b,c}$ represents the histogram formed by the corresponding pixel at the position of pixels in (a,b,c), $f_n(a,b,c)$ which represents the pixel level at coordinates (a,b,c) in frame n, total number of frames in GoF is represented as N and number of bins in the histogram is B. Generally, the intensity level of the pixel is equal to the number of bins in a histogram. Final values are the selected key frames which are represented as *Max Frame (a,b,c).*

3.2 BOVW (Bag of Visual Word) Generation

Bag of Visual Word is an extension of Bag of Words concept. It is an effective, potent and scalable approach for performing Content Based Video Retrieval (CBVR). Usually, BOVW is used to extract set of visual words from an enormous visual vocabulary. There are four basic steps in BOVW, which are as follow: (i) Feature extraction, (ii) Construction of visual vocabulary (iii) Quantization of each image features as discrete visual words, (iv) Construction of inverted-index using visual words and based on it matching is performed. The overall architecture of BOVW is presented in **Fig 2.** It envisions the sequence of work performed by Bag of Visual Word.

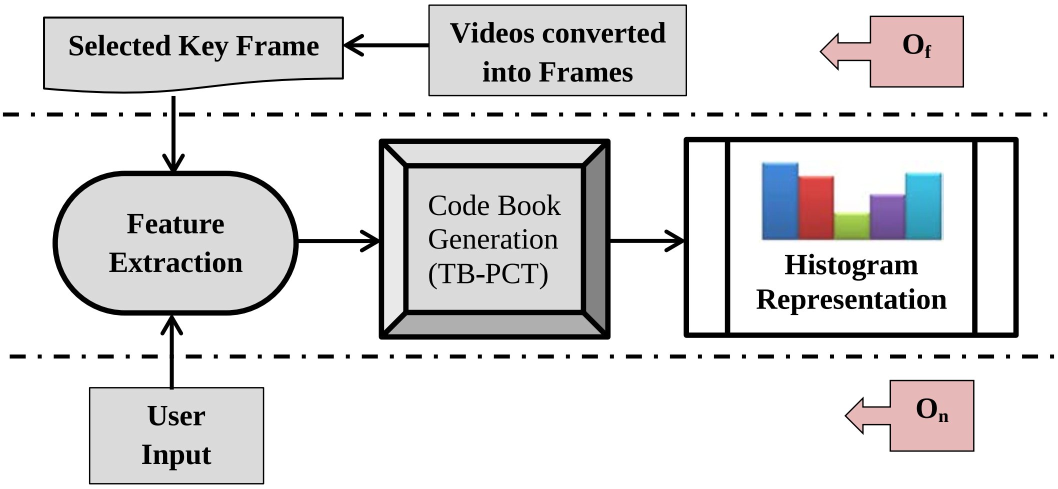

Fig. 2. BOVW procedure

Commonly, we use any one local descriptor to extract image feature in 3d model. But it is the first time we utilize three different image features (shape, color & texture) for feature extraction. In particular, we have considered combined feature (topological and geometric) for shape. Map reduce is used as a background for BOVW in implementation. Following section covers the above mentioned four steps.

3.2.1 Feature Extraction:

Initial stage of BOVW is feature extraction from the large dataset. For this motive, several local descriptors were used in the Conventional system. In most of the previous work [3], [7],

[9] SIFT is used as local descriptor. Feature extraction processes are described in sub chapters.

For feature extraction, we consider three different features of the image. In MapReduce, each task can be handled in parallel manner. So it makes feature extraction much simpler as one image does not depend on another and we can easily break down the tasks. Therefore, individual image is an input to the map task and extracted local feature of that image is the output. Map and Reduce functions states as follows,

$$Map(fl^{(n)}, I^{n}) \to \left[\langle fl^{(n)}, ft^{1 \ldots\ldots F(I^{n})}\rangle\right] \tag{5}$$

$$Reducer_{null}\left(fl^{n}, ft^{1 \ldots\ldots F(I^{n})}\right) \to \left[\langle fl^{(n)}, ft^{1 \ldots\ldots F(I^{n})}\rangle\right] \tag{6}$$

Where $fl^{(n)}$ is the key which represents the image filename or unique identifier. $I^{(n)}$ is n^{th} image of N images in the dataset. Key value assigned to the map tasks are unique identifiers. Each map tasks handles one image pair (i.e.) < $fl^{(n)}$, $I^{(n)}$>. Single map task will emit all local features of ($F(I^{n})$) as $ft^{1 \ldots F(I)}$ of $I^{(n)}$ image. To note, here the reducer is null reducer as the resultant output <key, value> pair is same as the input given to mapper.

In this manner, we can reduce the additional execution of reducer. Hence the output from mapper is given as input to the pipeline stage. As per our work shape feature is extracted first, followed by the color and texture features. The resultant feature extraction is a combination of all above mentioned features.

a. Shape Feature Extraction

Shape is considered as one of the prime feature of an image. In general, 3D model consider features like geometrical and topological for shape extraction. Previous works used any one of feature [25] or combination of two [17] for feature extraction. But previous work has many problems which we explained in related work. In order to overcome this, we have defined some rules to combine topological and highly biased geometrical feature. We make use for our proposed work. Overall step of shape feature extraction involves:

- Medial surface extraction & segmentation
- Re-adjustment of segmentation
- Use super ellipsoid to approximate the segmentation
- Use 3D DFD (Distance Field Descriptor)

To perform Medial Surface Extraction (M_S), we have used an appropriate Fork Strategy based Medial Surface Extraction. We start our process with 'M_S'. Medial surface extraction is performed using topology along with distance transform and Voronoi diagram method. In [17], M_S was performed using topology thinning which uses the concept of Average Outward Flux which in turn uses Euclidean distance as its parameter but it does not guarantee the overall geometry. In our proposed fork strategy, images are partitioned into cells, edges and vertices based on the points present on it. Each point should be equidistant from at least two edges. Further the corner cells are eliminated till, central point gets its immediate neighboring cells. Considering "n" points, we can say that every cell is the intersection of (n−1) half-planes which are convex in shape and have at least (n−1) edges in the boundary. To split 'n' points in the plane, required time is computed as,

$$T = O(n^{2})$$
$$Max\ Frame\ (a, b, c) \tag{7}$$

Next step is M_S segmentation which involves use of different classification methods to classify every voxel on medial surface. The classification methods are namely Simple, Surface Voxel, Line Voxel and Junction. After the completion of classification, apply rules to perform segmentation in M_S [17].

A boundary voxel p[l] is assigned to the segment S_i,

$$p[l] \in\Leftrightarrow w_i . D_i[l] = argmin_{1\le k\le N_i}\{D_i(l)\} \tag{9}$$

Where w_i is a weight factor for given by the following equation:

$$W_i = 1 + \frac{\sigma_i - arg\ min_{1\le i\le n}\,\sigma_i}{arg\ max_{1\le i\le n}\,\sigma_i - arg\ min_{1\le i\le n}\,\sigma_i} \tag{10}$$

Let σ_i be the standard deviation of D_i[l] for all boundary voxels p_i[l] assigned to the segment S_i. Each and every extracted segment of the 3d-objects is approximated using super ellipsoids to extract the shape without any loss. By using the attributed graph representation, both topology and geometrical information's are combined. The result obtained from medial surface extraction and readjustment procedures are considered as topological information whereas the meaningful parts assigned to each surface segment are well thought out to be geometrical information.
The graph is represented as,

$$G = \{V, E, A\{\,B_i\}_{i=1}^{s}\} \tag{11}$$

Where V is the non-empty set of vertices, E is the set of edges and A is the binary symmetric adjacency matrix. According to the mapping definition, an undirected vertex-attributed graph is constructed. In equation below, the parameters specified within the function F is (x, y, z) which represents the co-ordinates of the 3D point.

$$\begin{aligned} \mathrm{F(x,y,z)} &= \left(((\frac{x}{a_1})^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}})^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} \right)^{\varepsilon_1} \\ &= \left(\frac{xa_2a_3 + ya_1a_3 + za_1a_2}{a_1a_2a_3}\right)^2 \\ &= 1. \end{aligned} \tag{12}$$

Function (10) is commonly called inside-outside function, since it a 3-D point with coordinates(x,y,z):

$$\begin{cases} F(x,y,z) > 1, & if\ (x,y,z) lies\ outside \\ & the\ surface\ of\ the\ object \\ F(x,y,z) \le 1, & if (x,y,z) lies\ inside\ or\ on \\ & the\ surface\ of\ the\ object \end{cases} \tag{13}$$

The issue of modeling 3-D object using super quadratic can be prevail over by reducing it to the least squares minimization of nonlinear inside-outside function F(x,y,z) with respect to several shape parameters,

$$F(x,y,z)=F(x,y,z;a_1,a_2,a_3,\varepsilon_1,\varepsilon_2,\varepsilon_3,\Phi,\theta,\chi,t_x,t_y,t_z) \tag{14}$$

Where t_x, t_y ,t_z and Φ, θ, χ are Euler angles and translation vector coefficients, a_1,a_2,a_3, $\varepsilon_1,\varepsilon_2,\varepsilon_3$, are the superquatratic shape parameters and (x,y,z) are the coordinates of 3-D objects. Using mean-square error, the above found parameters are minimized as,

$$MSE=\sum_{i=1}^{N}\sqrt{a_1a_2a_3}\ \left(F(x_i,y_i,z_i)-1\right)^2 \tag{15}$$

Where, N is the number of points of the 3-D object.

As mentioned in over segmentation criteria, all the parts of the image will be merged into a single part, if the number of resulted parts is comparable to the number of medial surface voxels. In case of merging, only geometry feature is considered ignoring the topology information. Coming up to the 3-D DFD, it is used to compute the difference between the surface of an ellipsoid and surface of an object. Using the following scaling procedure, the 3-D DFD is extorted from every segment of 3-D object,

$$d_{DF}^{S}=\frac{1}{\sum_i|d_{DF}(i)|}d_{DF} \tag{16}$$

$$d_{DF}=[\|F_{DF}(0,0)\|\|F_{DF}(0,1)\|\|F_{DF}(1,0)\|\|F_{DF}(0,2)\|$$
$$\|F_{DF}(1,1)\|\|F_{DF}(0,2)\|\ldots..] \tag{17}$$

$$F_{DF}=FT\ \{DF\} \tag{18}$$

$$DF=[d_{i,j}] \tag{19}$$

$d_{i,j}$ is the signed distance between the surface of an object and surface of the ellipsoid at the same(θ_i, Φ_j).

Equation for 3D-Distance Field Description is acquired as a result of shape based feature extraction,

$$(d_{DF}^{S})=\frac{1}{\sum_i|d_{DF}(i)|}d_{DF} \tag{20}$$

The above mentioned equation is 3D-Distance Field Descriptor which is obtained using maximum occurred frames. This value is further applied for color & feature extraction process.

b. Color & Texture Feature Extraction

The color and texture is also a significant feature of image. As we already extorted the shape feature, now there is a need to extract other features like color and texture which can be done using the approximated super ellipsoid 3-D object of an image. In reference to our article, we make use of 3d co-occurrence matrix as feature descriptor for both color and texture. As an initial step, RGB color component is converted into HIS color component in order to separate the color and intensity component. HIS color component is quantized into 128(8*4*4) bins. Next, the image is split into squared window. Our concept includes overlap regions as an added advantage. Nine orientations are employed in each window to define the neighborhood of each pixel along the HSI plane. As a result of 3D co-occurrence matrices, six textural features are calculated. In total 54 measures are generated as 9 matrices were calculated.

Formulation of 3D co-occurrence matrix makes use of orientation of angle ‘θ’ which occurs between the pair of gray levels and the axis. We consider four directions, they are θ = 0°, 45°, 90°, 135°. The corresponding Gray-level Co-occurrence matrix is given as,

$$p(i,j|d,0°) = \{(x_1,y_1),(x_2,y_2)\} \tag{21}$$

Where, $i = I(x_1,y_1), j = I(x_2y_2), |\, x_1 - x_2| = 0°, |\, y_1 - y_2| = d$

The above mentioned equation is reconstructed by modifying ‘θ’ orientation for the remaining three angles. Next, the gray level co-occurrence is defined as $p(i,j|d,\theta)$ with respect to the distance $'d'$ and angle ‘θ’. Further the probability value of gray level co-occurrence matrix is given as,

$$\mathrm{p(i,j|d,\theta)} = \frac{p(i,j|d,\theta)}{\sum_{i=1}^{256}\sum_{j=1}^{256} p(i,j|d,\theta)} \tag{22}$$

From the estimated $p(i,j|d,\theta)$ component, we get features which can be further utilized for extraction process. The texture features are calculated using,

Angular second moment (ASM):

It measures the number of repeated pairs. Only limited number of gray levels is presented in the homogeneous scene. ASM can be calculated as follow,

$$ASM = \sum_i^U P^2[i.j] \tag{23}$$

Contrast (CN):

Local intensity variation of an image and it will favor contributions from P(i,j) away from the diagonal, i.e. i≠ j. contrast can be calculated using the equation,

$$Contrast = \sum_i^U \sum_j^V (i-j)^2 P[i,j] \tag{24}$$

Entropy (ET):

This parameter calculates the randomness of gray-level distribution.

$$Entropy = -\sum_i^U \sum_j^V P[i,j] \log P[i,j] \tag{25}$$

Correlation (CR):

It provides the correlation between two pixels present in the pixel pair.

$$Correlation = \sum_i^U \sum_j^V \frac{(i-\mu)(j-\mu)P[i,j]}{\sigma^2} \tag{26}$$

Where μ and σ indicate mean and standard deviation.

Mean (MN):

Parameter is used to calculate the mean of gray level in an image. It can be manipulated as,

$$Mean = \frac{1}{2} \sum_i^U \sum_j^V (iP[i,j] + jP[i,j]) \tag{27}$$

Variance (VN):

Variance is used to explain the overall distribution of gray level. The following equation is used to calculate the variance,

$$Variance = \frac{1}{2}\sum_i^U \sum_j^V ((i-\mu)^2 p[i,j] + (j-\mu)^2 P[i,j]) \tag{28}$$

The above mentioned equations are applied to nine texture matrices and resulted as a single entity. The extracted feature is the combination of all the features mentioned in our work, which given as,

$$Etd_f = [ASM, CN, ET, CR, MN, VN] \tag{29}$$

Equation (27) gives the feature extracted (Etd_f) from all the three features taken into account (Shape, Color, Texture).

3.2.2 Visual vocabulary Generation using TB-PCT

Construction of Visual vocabulary (V_c) is a central part of the CBVR system. It is the next stage of BOVW. In the previous work, several clustering algorithms were used for vocabulary generation process [2], [4], [9]. K-means clustering algorithm is frequently used technique but it limits its merit in case of handling large amount of datasets. To overcome this problem, we used PCT (predictive clustering Tree) [15] in our work by modifying it as per our need.

To construct PCT, we combine the concept of predictive clustering and randomized tree construction. PCT can be provoked using a standard top-down induction of decision trees (TDIDT) algorithm. During the tree construction process, use local descriptors values in a random manner. Algorithm 1 shows the TDIDT algorithm.

Algorithm 1: TDIDT

```
procedure PCT(E) returns tree
(t*,h*,p*) = BestTest(E)
if t* ≠ none then
    for each $E_i \in p^*$ do
        $tree_i$ = PCT($E_i$)
    return node(t*,$\bigcup_i \{tree_i\}$)
else
    return leaf(Prototype(E))
procedure BestTest(E)
(t*,h*,p*) = (none, 0, Φ)
for each possible test t do
p= partition induced by t on E
h= $Var(E) - \sum_{E_i \in p} \frac{|E_i|}{|E|} Var(E_i)$
if(h>h*) ^ Acceptable (t,p) then
        (t*,h*,p*) = (t,h,p)
return (t*,h*,p*).
```

Algorithm 2: TB-PCT

```
Input: Set of subsets of local descriptors as
$S_i=\{s_1,s_2,......s_n\}$
output: Visual Vocabulary
Begin
Choose $S_i$ based on threshold (t)
Set TS = $S_i$
Assign $d_a = t_a$ & $C_a$ = 128 Vector
perform Cluster 'C' using PCT
Construct tree(T) using 'C' based on
 Randomized Tree
 leaf (l) = $V_w$
$V_c = \Sigma$ l;
End
```

In our proposed work, we introduced the concept of TB-PCT (Threshold Based –Predictive Clustering Tree) which is slight modification of PCT. Algorithm 2 depicts the overall working procedure for the construction of visual vocabulary using threshold.

Construction of visual words suffers from the problem of noise, which occurs when the same visual word exist more than once. These noisy visual words are referred as unusable visual words which can create difficulty during retrieval. To overcome this existing problem, we have introduced an innovative approach of numerical semantic analysis. Repeated visual words are considered as noise, since it takes more computation time to analyze multiple queries. To overcome this problem, we assign integers for each visual word. Unique integers are assigned to unique visual words and similar integers are assigned to similar visual words. By using this technique, we find similar visual word that occurs more than once as they will have similar integer numbers. Hence, we reduce the problem of noise which in turn reduces the computation time for the construction of visual words.

As a first step, local descriptors are gathered from section 3.2.1 and partitioned into subsets (S_i) using machine learning algorithm and predictive clustering [17]. Resultant subsets are given as input for vocabulary construction and output from this is used to construct $V_{c.}$ Subsets of local descriptors (s_i) is selected using threshold value (t). The selected values (s_i) constitute a training set (TS). For constructing tree (T), we set descriptive attribute (d_a) as target (t_a) and also clustering attribute (c_a). The value of descriptive attribute (d_a) is 128 dimensional vectors. This is one of the unique characteristics of the PCT. Root node maintains the overall information of the image and it recursively partition to construct tree. Apply pre-pruning techniques to control the size of visual vocabulary. Pre –pruning requires information regarding the minimum number of descriptor in each tree leaf (l). We can easily determine the number of instance required for a leaf to get the desired number of leaves for the tree, for the given dataset. Each leaf 'l' of the tree has a separate visual word (V_w). We need to combine all the leaves to form V_c i.e. visual codebook (V_{Cb}) for a frame.

These processes are managed by Map and Reduce functions in parallel manner [3]. The Map and Reduce functions is given as,

$$\mathrm{Map}(\mathrm{fl}^{(n)}, ft^{1.....F(I)^n} \rightarrow [<\mathrm{l}^{(1)},\mathrm{ft}^{(1)}>, \ldots, <l^{(F(I^n))}, ft^{F(I)^n} >],$$
$$\mathrm{Reduce}(\mathrm{l},\mathrm{ft}^{1\ldots.F(k)}) \rightarrow [<\mathrm{l},\mathrm{ft}^{\mathrm{avg}}>] \quad (30)$$

The map task holds the information of leaf node. TB-PCTs are efficient in enumeration. Here we resolve the problems of handling small amount of dataset by using random forest tree. By concatenating the individual V_{Cb} from each of the TB-PCT we can obtain final codebook.

The formula for evaluating TB-PCT is as given below,

$$V_{Cb} = \sum_{i=1}^{n} \left[V_{cb(i)} \right] \quad (31)$$

After completion of the V_{Cb} computation process, the next step is to constitute each 3D model (frames in dataset or query) as a histogram which is nothing but occurrences of the codebook element. For this purpose, we sort all the descriptors along the tree. As a Further step, we need to count the number of descriptors that is present in a given 'l' i.e. leaf , which accounts for the number of visual word. We have presented the frame as a histogram of descriptors per visual word.

3.2.3 Matching (soft-weighting schemes)

Matching is the final stage of CBVR where we have used the distance function and similarity schemes to retrieve proficient results. To measure the similarity between the two images, a distance function is needed. In order to accomplish this task, we have used L_2 (Euclidean) distance function and soft weighting scheme [10]. Map reduce task is partite as Map and reduce function.

a. Map stage:

In Map stage, measurement between one key point and its neighboring visual word is performed to compute the similarity function from this top-k nearest neighbors is found. Now the mapping process takes place by mapping the key point to its top-k nearest neighbor. From each mapper, we get result as the number of output pairs ($Ix^{(n)}$, $vt_k^{(n)}$) .Here, $Ix^{(n)}$ indicates the key($Ix_{(n)}$) which is the index of ith ranking visual word in the top-k results and $vt_k{}^{(}n^{)}$ indicates value($vt_k^{(n)}$)which represent the similarity score with partial weight according to the rule that forms proximity with visual word. Priority is given to the highest similarity score during the process.

b. Reduce stage:

In Reduce stage, histogram is computed by arranging the (key, value) ($Ix^{(n)}$, $vt_k^{(n)}$) pair to produce video representation.

In particular, reduce stage gathers the partial weight value of each key pair- value ($Ix^{(n)}$, $vt_k^{(n)}$) to its corresponding key index.

Soft-weighting scheme

Scheme plays an important role to find the weight of each visual code word. For each key point in the image (query image & dataset frame), we select the top-k nearest visual words instead of searching only for the nearest visual words. If suppose we have a visual vocabulary of k visual words, we use a k-dimensional vector VT = [vt_1, vt_2, …vt_k] with each component vt_k representing the weight of a visual word k in an image. It can be calculated as follows,

$$vt_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i}{}_{-1}} sim(j,k) \tag{32}$$

$$\text{sim(j,k)=dis}_{j,k} \tag{33}$$

Where M_i represents the number of key points whose ith nearest neighbor is visual word k. Estimating sim(j,k) constitutes the similarity between key point j and visual word k. In (25), key point is found based on its similarity to word k, weighted by $\frac{1}{2^{i}{}_{-1}}$ where the constituted word is its ith nearest neighbor. Usually, N=4 is reasonable to set. The component $dis_{j,k}$ represent the distance between two images. To predict the distance, we have used L_2 distance functions. Occurrence of visual word is constructed as a tree like index structure, which proves to be computationally efficient. We fetch weights from both i.e visual Vector of query image and, the image retrieved from nearest neighbors. Equation used to determine L_2 distance is given as,

$$dis = \sqrt{\sum_{x=1}^{k} \left(Vt_{xi} - Vt_{xj}\right)^2}$$

Where,

x – 1,2,3,……….k

Vt – Visual Vector of image

Based on the calculated visual vector similarity, ranking is performed using k-nearest neighboring visual vector images. It makes our proposed system efficient and effective. Finally the matched results are ranked according to the index value of an images and it is returned back to the queried users as a feedback.

4 Experimental Evaluation

4.1 Dataset and Tools

Our proposed system is evaluated as per the data set which kept in the view for user convenience. Some of the experimental data sets are used in research papers such as text documents, images and audio files. Generally the sizes of the video files are larger when it compared with other files like images or audios. In our experiment, we take video files and convert it into frames using frame conversion tools, converted frames are considered as dataset. Whereas our experiments are considered 3D videos which are gathered from different web pages such as YouTube, Wikipedia, Flickr etc., which are also includes some other web pages that shows videos on its pages. Such web pages consist of more than thousands of video files in it. Here our proposed system brings out an effective retrieval of related videos as per the user's request.

Assessment of our proposed system includes a short description about the computer systems which is based on the storage capacity, speed, number of systems used, etc. Based upon the specifications, computer systems are to be participated and to provide estimated results regarding our progress. **Table 1** shows the hardware configurations of the implementation.

We take five computer systems which are named as Node 1 – Node 5 with different specifications. Here Node 1 acts as the master that leads alike a leader and other systems process under the supervising of Node 1. The nodes specifications are designed based on the hardware configurations. The architecture of the proposed running system is shown in **Fig. 3** which consists of a single master slave node with four slave nodes.

Table 1. Hardware Configuration of our Implementation

Specifications	**N_1**	**N_2**	**N_3**	**N_4**	**N_5**
Processor	Intel Core i3	Dual Core	Intel Pentium Quad Core Processor	AND Fx-6100	Intel Core i5
CPU Core	4	2	4	6	4
CPU Speed	2.53GHz	3GHz	2.2GHz	3.9GHz	3.7GHz
RAM	4GB	2GB	2GB	8GB	8GB

Master slave node is represented as N1 and other slave nodes are represented as N2 – N5. The user requests a 3d image and retrieves number of relevant 3d videos as per the user input query. Here single user is supposed to look for only one image at a time. In case of remote access, multiple users can give request to the search engine, query as 3d images from different systems. As, if the user request 3d image then the system starts run and retrieve the available number of related 3d videos.

4.2 Performance Evaluation

For the purpose of comparative analysis we broadly classified in three scenarios for each process that is takes place in our proposed mapreduce framework. The three scenarios are as follows (i) Scenario-1: Node 1 and Node 2 is used, (ii) Scenario-2: Node 1, Node 2 and Node 3 are used and at last in (iii) Scenario-3: All the five nodes from Node 1 – Node 5 are used. Our implementation work to be done on the software specifications of ubuntu 14.04 and hadoop2.7.0 and JDK 1.7 version. We consider parameters for evaluating 3D video retrieval of our proposed system. By taking the parameters such as Precision-Recall, E-Measure and Discounted Cumulative Gain and Mapreduce Time.

4.2.1 Precision-Recall

The parameter Precision is the ratio of retrieved 3D models which are related to the specified query. Then the parameter Recall is the ratio of the successfully retrieved 3D models which are related to the specified input query. Formula for precision and recall are as follows,

$$\text{Precision} = \frac{\text{Q} \cap \text{R}}{\text{R}}$$

$$\text{Recall} = \frac{\text{Q} \cap \text{R}}{\text{Q}}$$

By using the above mentioned formula, the Precision-recall estimation can be done. Where Q represents the relevant query that is retrieved from the 3D models and R is the retrieved 3D models.

4.2.2 E-Measure

E-Measure is defined as the measurement of precision and recall for a fixed number of retrieved 3D models. E-Measure is defined in the form of an equation as follows,

$$\text{E} - \text{Measure} = {}^{2}\!\Big/\!\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)$$

From the above equation, we can estimate E-Measure for entire proposed system

4.2.3 Discounted Cumulative Gain

The gain is the measure of the ranking list which is retrieved from 3D models. Here the correct results stands front of the ranking list with heavier weight than the corrected results at the end of the ranking list.

4.2.4 Map-Reduce

A Map-Reduce is collection of Map and Reduce procedure as discussed earlier in above sections. The map tasks run parallel by splitting the given input data's. Likewise reducer task also perform parallel working on different intermediate keys. Estimation of the overall processing time is calculated by Map-Reduce.

4.3 Experimental Results

As per the dataset, our system is composed of 12890 videos and 399015 key frames generated for the videos. Then a set of key frames are taken as training set and few are taken for testing.

The performance of Precision-Recall is experimentally analyzed by the graph. A curve is plotted for Precision-Recall parameter based on the three scenarios mentioned in the section 4.2. Overall precision –recall values displays the retrieval accuracy. From the Plot 6, we can observe that there is slight difference among the three scenarios. However scenario 3 shows better performance as compared to other two scenarios.

Table 2. Experimental results of training for three scenarios

Steps Involved	**Training**		
	Scenario 1	**Scenario 2**	**Scenario 3**
Shape Feature Extraction	702	421.2	210.6
Overall Feature Extraction	1367.5	820.25	407.65
Codebook Generation	1277	892.2	446.1

The resulted values are the processing time (unit mentioned in seconds) taken for three different scenarios. They are tabulated in Table 2 and Table 3. All the three scenarios are dissimilar as they differ in their processing time. The values tabulated in the tables are obtained for the above mentioned 399015 key frames.

Table 3. Overall Processing Time Vs Scenarios

PROCESSING STEPS	**TESTING**		
	Scenario 1	Scenario 2	Scenario 3
Overall Feature Extraction	1510.5	968.28	550.79
Shape Feature	836	555	345

Table 4. Overall Mapreduce Results

Overall Mapreduce		
Traning		
Scenario 1	Scenario 2	Scenario 3
2003	1202	601
Testing		
Scenario 1	Scenario 2	Scenario 3
3677	2206	1103

This analysis shows difference in the processing speed based on our scenarios. We have compared the scenarios and finally concluded that scenario 3 shows better performance than the other two scenarios.

Fig. 3 shows three different plots for three scenarios, considering the usage of hadoop. It implies that the presentation of scenario 3 is better in case of Precision-Recall rate when compared to other scenarios. These two parameters are described above with unique formulas. The entire E-measure percentage obtained for our proposed is around 64%. Whereas DCG obtained is of 69.4 % based on the list of ranking obtained from the retrieved 3d videos

Table 4 shows the resulted processing time taken for the overall Map-Reduce process. Here also the training and the testing time varies for each scenario..

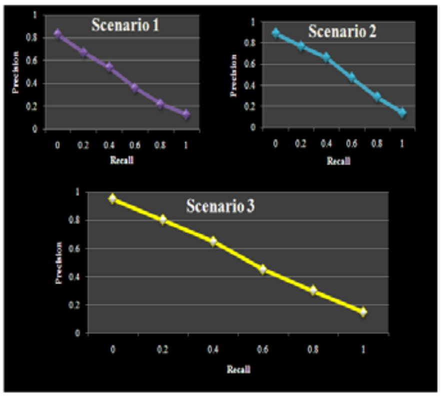

Fig. 3. Precision-Recall for three scenarios

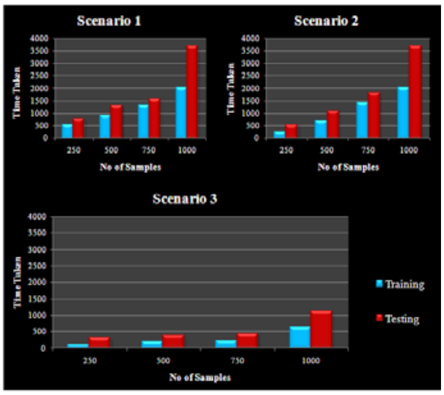

Fig. 4. Overall mapreduce time

Fig. 4 shows the plot overall Map-Reduce processing time for 1000 samples considered for our process. This resulted tentative result is drawn number of samples taken with respect to the entire processing time taken in Map-Reduce. The processing time differs for each input in query image. The processing time variance based on the videos available for the given input query. The total number of results in this form conveys the number of relevant 3d videos retrieved from the given 3d query.

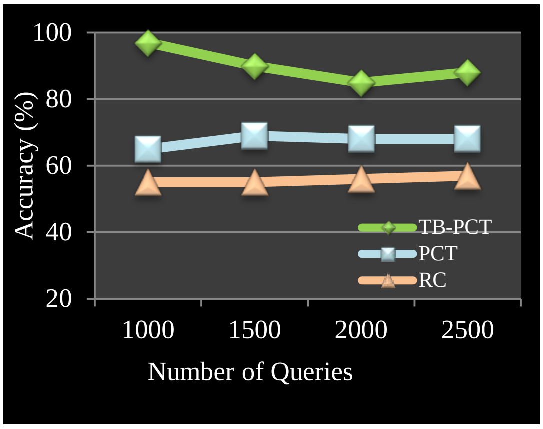

Fig. 5. Comparison on Codebook Generation

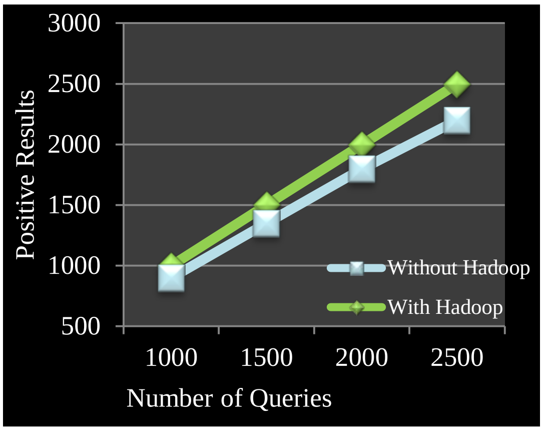

Fig. 6. Comparison on Results Retrieved

Fig. 5 illustrates the comparison between PCT [19], [31] Randomized Clustering [16] and TB-PCT. Here, accuracy is measured in terms of its efficiency to retrieve the relevant results for the given number of queries. In normal PCT process, the local descriptors are selected in random manner whereas in TB-PCT local descriptors are selected based on the threshold value.

Fig. 6 shows the plot between numbers of queries given by multiple users and its response. It also shows that queries by multiple users at the same time will provide positive result only. Above figure shows the comparison between the system with Hadoop and without Hadoop based on our proposed system. Finally, we conclude that the system using Hadoop gets positive response for all its queries.

5 Conclusion

We have included map reduce framework in our system in order to speed up the processing time of 3D video retrieval process. Our newfangled approach is mainly focused on scalability and fast processing of large dataset. Map reduce has following processing stage which includes key frame selection, feature extraction visual codebook generation and matching. All this process runs in the background of map reduce. From our experimental result, we can have proved that our system reduces the time complexity. Our 3D-CBVR system contains 399015 key frames for processing and hence produces efficient result. 3D-CBVR system has different precision & recall rate for different scenarios. Hadoop based 3D-CBVR system has an added advantage that it takes less time for clustering (generating visual codebook) because we have used TB-PCT method which proves to be efficient for large dataset. We have combined all the features (shape, color & texture) to acquire fruitful result and improve precision. Due to this, no dataset (videos) could be missed. We have affirmed that our system that is BOVW and Mapreduce based 3D-CBVR system take less time for video retrieval process. We have overcome the problem of time complexity and produced efficient result. In the future work, we will incorporate the concept of duplicate video detection to further reduce the processing time.

References

[1] C. Ranjith Kumar and Dr. S. Naga Nandhini Sujatha, "STAR: Semi-supervised-clustering Technique with Application for Retrieval of video," in *Proc. of International Conference on Intelligent Computing Applications*, pp. 223-227, 2014. Article (CrossRef Link)

[2] Hedi Tabia and Hamid Laga, "Covariance-Based Descriptors for Efficient 3D Shape Matching, Retrieval and Classification," *IEEE Transactions on Multimedia*, vol.17, no.9, pp.1591 – 1603, 2015. Article (CrossRef Link)

[3] Jonathon S. Hare, Sina Samangooei and Paul H. Lewis, "Practical scalable image analysis and indexing using Hadoop," *Multimedia Tools and Applications*, vol.71, pp.1215-1248, 2012. Article (CrossRef Link)

[4] Ke Ding, WeiWang and Yunhui Liu, "3D model retrieval using Bag-of-View-Words," *Multimedia Tools and Applications*, vol.72, pp. 2701-2722, 2013. Article (CrossRef Link)

[5] Zongmin Li, Zijian Wu, Zhenzhong Kuang, Kai Chen, Yongzhou Gan and Jianping Fan, "Evidence-based SVM fusion for 3D model retrieval," *Multimedia Tools and Applications*, vol. 72, pp. 1731-1749, 2013. Article (CrossRef Link)

[6] HanliWang, Fengkuangtian Zhu, Bo Xiao, LeiWang and Yu-Gang Jiang, "GPU-based MapReduce for large-scale near-duplicate video retrieval," *Multimedia Tools and Applications*, vol. 74, pp. 10515-10534, 2014. Article (CrossRef Link)

[7] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman and David Dobkin, "A Search Engine for 3D Models," *ACM Transactions on Graphics*, vol. 22, pp. 83-105, 2003. Article (CrossRef Link)

[8] Umesh K K and Suresha, "Web Image Retrieval Using Visual Dictionary," *International Journal on Web Service Computing*, vol. 3, No. 3, 2012. Article (CrossRef Link)

[9] Brandyn White, Tom Yeh, Jimmy Lin, and Larry Davis, "Web-Scale Computer Vision using MapReduce for Multimedia Data Mining," in *Proc. of 10th ACM International Workshop on Multimedia Data mining*, 2010. Article (CrossRef Link)

[10] Yu-Gang Jiang, Chong-Wah Ngo and Jun Yang, "Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval," in *Proc. of 6th ACM International Conference on Image and Video Retrieval*, pp.494-501, 2007. Article (CrossRef Link)

[11] Jialu Liu, "Image Retrieval based on Bag-of-Words model," *Information Retrieval*, 2013. Article (CrossRef Link)

[12] Hanli Wang, Yun Shen, Lei Wang, Kuangtian Zhufeng, Wei Wang and Cheng Cheng, "Large-Scale Multimedia Data Mining Using MapReduce Framework," in *Proc. of IEEE 4th International Conference on Cloud Computing Technology and Science*, pp. 287 – 292, 2012. Article (CrossRef Link)

[13] Eric Brachmann, Marcel Spehr and Stefan Gumhold, "Feature Propagation on Image Webs for Enhanced Image Retrieval," in *Proc. of 3rd ACM International Conference on Multimedia Retrieval*, pp 25-32, 2013. Article (CrossRef Link)

[14]KehuaGuo, WeiPan, MingmingLu, XiaokeZhou and JianhuaMa, "An effective and economical architecture for semantic-based heterogeneous multimedia big data retrieval," *Journal of System and Software*, vol. 102, pp. 207–216, 2014. Article (CrossRef Link)

[15] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Saso Dzeroski, "Fast and Scalable Image Retrieval Using Predictive Clustering Trees," *Lecture Notes in Computer Science*, vol. 8140, pp. 33-48, 2013. Article (CrossRef Link)

[16] Frank Moosmann, Eric Nowak and Frederic Jurie, "Randomized Clustering Forests for Image Classification," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 30, no. 9, pp. 1632 – 1646, 2008. Article (CrossRef Link)

[17] Athanasios Mademlis, Petros Daras, Apostolos Axenopoulos, Dimitrios Tzovaras and Michael G. Strintzis, “Combining Topological and Geometrical Features for Global and Partial 3-D Shape Retrieval,” *IEEE Transactions On Multimedia*, vol. 10, no. 5, pp. 819 – 831,2008. Article (CrossRef Link)

[18] William Robson Schwartz and Hélio Pedrini, “Color Textured Image Segmentation Based on Spatial Dependence Using 3D Co-occurrence Matrices and Markov Random Fields,” *Journal of Science*, vol. 53, no. 3, pp. 693-702, 2012. Article (CrossRef Link)

[19] B H Shekar, K Raghurama Holla and M Sharmila Kumari, “Video Retrieval: An accurate approach based on Kirsch Descriptor,” in *Proc. of IEEE International Conference on Contemporary Computing and Informatics*, pp. 1203 – 1207, 2014. Article (CrossRef Link)

[20] Kin-Wai Sze, Kin-Man Lam, and Guoping Qiu, “A New Key Frame Representation for Video Segment Retrieval,” *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 15, no. 9, pp. 1148 - 1155, 2005. Article (CrossRef Link)

[21] Conrado R. Ruiz, Jr., Rafael Cabredo, Levi Jones Monteverde and Zhiyong Huang, “Combining Shape and Color for Retrieval of 3D Models,” in *Proc. of IEEE 5th International Joint Conference*, pp. 1295 – 1300, 2009. Article (CrossRef Link)

[22] Amirthalingam Ramanan and Mahesan Niranjan, “A Review of Codebook Models in Patch-Based Visual Object Recognition,” *Journal of Signal Processing System*, vol. 68, pp. 333-352, 2011. Article (CrossRef Link)

[23] Arati S. Kurani, Dong-Hui Xu, Jacob Furst and Daniela Stan Raicu, “Co-Occurrence Matrices for Volumetric Data,” in *Proc. of 7th International Conference on Computer Graphics and Imaging*, 2014. Article (CrossRef Link)

[24] J. Calic, B. T. Thomas, “Spatial Analysis In Key-Frame Extraction Using Video Segmentation,” in *Proc. of Workshop on Image Analysis for Multimedia Interactive Services*, 2004. Article (CrossRef Link)

[25] Johan W.H. Tangelder and Remco C. Veltkamp, “A Survey of Content Based 3D Shape Retrieval Methods,” *IEEE Proceedings of Shape Modeling Applications*, pp. 145 – 156, 2004. Article (CrossRef Link)

[26] Hanan ElNaghy, Safwat Hamad and M. Essam Khalifa “Taxonomy For 3d Content-Based Object Retrieval Methods,” *Journal of Research and Reviews in Applied Sciences*, pp. 412-446, 2013. Article (CrossRef Link)

[27] Ting-Chu Lin, Min-Chun Yang, Chia-Yin Tsai and Yu-Chiang Frank Wang, “Query-Adaptive Multiple Instance Learning for Video Instance Retrieval,” *IEEE Transactions On Image Processing*, vol. 24, no. 4, pp. 1330 - 1340, 2015. Article (CrossRef Link)

[28] Chunlai Yan, “Accurate Image Retrieval Algorithm Based on Color and Texture Feature,” *Journal Of Multimedia*, vol. 8, no. 3, pp. 277-283, 2013. Article (CrossRef Link)

[29] Md. Baharul Islam, Krishanu Kundu and Arif Ahmed, “Texture Feature based Image Retrieval Algorithms,” *International Journal of Engineering and Technical Research*, vol. 2, pp. 71 – 75, 2014. Article (CrossRef Link)

[30] Peng Huang, Adrian Hilton and Jonathan Starck, “Automatic 3D Video Summarization: Key Frame Extraction from Self-Similarity,” *Fourth International Symposium on 3D Data Processing*, Visualization and Transmission, 2008. Article (CrossRef Link)

[31] Bernard Ženko, , Sašo Džeroski and , Jan Struyf, “Learning Predictive Clustering Rules,” *Knowledge Discovery in Inductive Databases*, vol. 3933, pp. 234-250, 2006. Article (CrossRef Link)

[32] Prof.Deepti Chikmurge, “Implementation of CBIR Using MapReduce Over HADOOP,” in *Proc. of IEEE Sponsored International Conference On Empowering Emerging Trends In Computer, Information Technology & Bioinformatics International Journal of Computer, Information Technology & Bioinformatics*, vol. 2, Issue 2, pp 1 – 3, 2014. Article (CrossRef Link)

[33] Wichian Premchaiswadi, Anucha Tungkatsathan, Sarayut Intarasema, “Improving Performance of Content-Based Image Retrieval Schemes using Hadoop MapReduce,” in *Proc. of IEEE International Conference*, pp. 615 – 620, 2013. Article (CrossRef Link)

[34] Zechao Li, Jing Liu, Yi Yang, Xiaofang Zhou, Hanqing Lu, "Clustering-Guided Sparse Structural Learning for Unsupervised Feature Selection," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, vol. 26, no. 9, pp 2138 -2150, 2014. Article (CrossRef Link)

[35] Zechao Li, Jing Liu, Jinhui Tang, Hanqing Lu," Robust Structured Subspace Learning for Data Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, Issue 10, pp. 2085 -2098, 2015. Article (CrossRef Link)

Ranjith Kumar is working in Educational Institution of Madurai Kamaraj University, Tamilnadu. Passed 6 years he was working Lecturer as in computer science department. He was received his Master degree under Madurai Kamaraj University. He was interested to do his Ph.D degree in a domain of Data Mining. Currently his research interest includes image and video retrieval, hadoop and 3-D processing.

Suguna is working as an Assistant Professor in Sri Meenakshi Government Arts College belonging to the department of Computer Science in Madurai District. She has completed her Post-Graduate Degree in M.k. University and did her M.Phil in Mother Teresa Women's University. At present her interest is to do Ph.D in Computer Science department. Her current research interest includes Web Mining and Semantic concepts.