Using Video for Multi-view Object Categorization in Security Systems

Humera Noor\textsuperscript{a}, Shaheena Noor\textsuperscript{b}, Shahid H. Mirza\textsuperscript{a}

\textsuperscript{a}Dept. of Computer & Information Systems Engg., NED University of Engineering & Technology, Karachi, Pakistan;
\textsuperscript{b}Dept. of Computer Systems Engineering., Sir Syed University of Engineering & Technology, Karachi, Pakistan.

ABSTRACT

Recognizing objects of interest in an environment is one of the most important aspects of security applications. Many techniques exist focusing on object categorization; however, most of them consider just a single viewpoint. This leads to increased false alarms as multiple objects look alike from one viewpoint and are totally different from other view. Hence, it is important to consider multiple views of the target simultaneously while categorizing. This paper presents a strategy for multi-view object categorization on the basis of video. The temporal and spatial information of videos is utilized to effectively categorize the objects from multiple views.

Given a set of images of an object category, independent graphical models are generated for each object using underlying geometry and pruned using morphing. Next, the model is evolved by combining the independent graphs, where each node represents different instances from same viewpoint and links exist between adjacent viewpoints. Finally, a classifier is trained for each node. The algorithm is strong in that it does not require camera calibration and doesn’t place any constraint on object and/or camera trajectory. Also, it eliminates the need of manually aligning the corresponding viewpoints across the different instances of the object. Finally, the motion continuity in video is exploited to perform matching based on video clip rather than the conventional single image. It is demonstrated using datasets from VIVID (DARPA), NASA and PASCAL 2006/2007 that the results thus obtained are much more robust with a precision of 85.73% as against 76.14% for single image.

Keywords: Object Categorization, Multi-view categorization, Video-based matching, Morphing.

1. INTRODUCTION

In current day environment, security is an issue that is becoming important day by day. For this purpose a variety of techniques are used, including sensors, surveillance cameras etc. These systems require human monitoring to detect and handle any intrusion. Meanwhile, Object Categorization (also referred to as object classification and generic object recognition) has emerged as an important and challenging field in computer vision. Its prime application is to identify objects of a category in an image. This work is focusing on using object categorization techniques to monitor the environment and generate alerts in case security-threatening equipment is identified.

Object categorization successfully handles the unseen versions of the modeled objects, by forming a comprehensive representation of class-specific features extracted from training images. However, these systems are usually capable of performing classification with respect to a single or pre-defined viewpoint only. Efforts are being applied to develop systems capable of performing object categorization from multiple viewpoints. Initially, Schneiderman et al. [4] proposed a statistical method for 3D object detection from multiple points of view and applied it to faces and cars. However, this work simply involves running of multiple detectors in parallel, where each detector is specialized for a particular viewpoint. Later on, Torralba et al. [24] focused on promoting feature sharing among the different object classes and used a boosting algorithm to learn the features of an object common in different categories or views. However, their main objective was the reduction of computational complexity and not multi-view object classification. Recently, Thomas et al. [25] attempted to combine the two state-of-the-arts, the image exploration algorithm [5] for object recognition and implicit shape model [11] for object class detection, and developed an algorithm to perform object categorization from multiple viewpoints. However, they need to arrange their input images manually by aligning...
the images taken from corresponding viewpoints. Hence, they constrained their adjacent viewpoints to be approximately 22.5° apart on a circle around the object. Moreover, they were required to isolate the target object in training images.

Most of the work focusing on object classification utilizes static images for training and testing the model. Recent efforts are being applied to explore the use of temporal information present in video sequences to carry out object categorization with reduced computational complexity and improved performance. Initial approaches to utilize video sequence were dedicated to specific object recognition [8, 14]. One of the first attempts to generic object recognition using video was made by Schweighofer et al. [20]. They segmented out the object of interest using the spatio-temporal information available in video and used the isolated object for training the classifier. This strategy reduced the computational complexity as well as improved the learning. Later on Opelt et al. [17] used tracking to extract temporal dependencies of object regions in videos and used this additional information to reduce the training data, thus getting a faster training speed. However, none of these techniques utilized the spatio-temporal information of videos during the test phase of categorization.

In this work, a strategy for multi-view object categorization using video is presented. Given a set of reference images, it is arranged in the form of a connectivity graph by identifying the underlying topological structure. The graph is configured in such a manner that it emphasizes the relative spatial spread-out of images, providing information about their proximity with respect to multiple viewpoints. The 2D model is generated using image measurables only and a refinement procedure is applied using a virtual view synthesis procedure to remove the redundant images in the data set. The strengths of the algorithm are that neither any knowledge of camera calibration is needed nor any constraint is placed on the placement or motion of object and/or camera. Also, this strategy eliminates the need of manually aligning the corresponding viewpoints across several instances of the object category as was desired in earlier work. This alignment is necessary because to develop a classifier, it is required to know the common features across the different instances of the object from a particular viewpoint. Typically, the research on classification focuses on single-view classifiers only, and this problem of aligning viewpoints arise because multiple-views are being considered. As mentioned above, in previous work, a manual alignment is required with constrained image capture. This work eliminates the need of manual intervention, thus allowing an unconstrained trajectory for image capture. Finally, a video sequence has been used instead of the conventional single image for testing, which provides information of multiple viewpoints; and utilizing its spatio-temporal information the confidence measure of results has been strengthened.

This paper is organized as follows: Section 2 introduces how to identify the topological structure present in the images and arrange them as a neighborhood graph. It also describes the merger of multiple graphs of objects of a category by identifying the clusters within a graph and their correspondences across them. Section 3 describes the training of a single-view classifier. Section 4 highlights the use of the image database in conjunction with a test video sequence for object classification. In Section 5, the conducted experiments are discussed along with the results obtained. Section 6 gives the conclusions and the future extensions.

2. MODEL GENERATION

Any object can be modeled using either an object-centered or a view-centered representation [19]. The object-centered representations use the features from the objects, like boundary curves, surfaces etc, to describe the volumes of space. View-centered representations, on the other hand, depend on the outlook of objects from different viewpoints. These involve the use of aspect graphs and silhouettes for modeling. In this work, the view-centered representation for generation of database has been used, which makes the task of matching simpler. This is because the need for projection of model to 3D is no longer there and the features that are to be compared are in 2D [19].

2.1 Development of Neighborhood Graph

The first step towards creation of model is to develop an isolated neighborhood graph (also hereby referred to as the connectivity graph) for each instance of object of the given category. Given a set of reference images, an approach is proposed to tessellate them around the viewing space of the object while ensuring a near-minimal size of the database [14].
The algorithm begins by identifying the feature points in all the images of the repository. A variety of features exist including color, texture, motion and shape primitives etc; and interest point and region detectors including edge, corners etc. In this work, the Scale-Invariant Feature Transform (SIFT) operator \[13\] has been used to extract the distinctive features in the image. This is because, the features are invariant to image scale and rotation; and robust to changes in viewpoint and illumination. This makes them efficient for use in subsequent steps of matching. Moreover, the key point locations are also returned, which are later on used for determining closeness of viewpoints by finding distances between images. Feature correspondences are then identified using the nearest-neighbor algorithm \[13\]. In this step, first, for each keypoint, the best candidate match is found by identifying its nearest neighbor in the image. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant vector. Next, this distance is compared with the distance of the second-closest neighbor and accepted only if the ratio of distance of closest neighbor to that of second-closest neighbor is less than a threshold. This allows removal of false matches while retaining the correct ones. Figure 1 shows SIFT points and matches identified for a pair of image. The feature correspondences identified in this step are used to decide the presence or absence of linkage between nodes.

For an image database having originally \(N\) images, an \(N \times N\) link matrix is formed. A link between image pair \((I_i, I_j)\) is marked if they are found to be neighbor. Two images are defined to be neighbors, if they are close to each other in viewpoint. It should be noted that till now the term neighbor implied neighboring "features", while from now on the term is used to mean neighboring "images".

The procedure for identification of neighbors is two-fold. In the first pass, the average Euclidean distance \(d\) for each image pair \((I_i, I_j)\) is found. Other distance metrics exist, for example, the Manhattan distance, the Chebychev distance etc; however the Euclidean formulation measures the distance following the most direct, logical and straightforward path between two points. For \(c\) corresponding points between two images, \(d\) can be computed as given by Equation 1:

\[
d(I_i, I_j) = \frac{\sum_{k=1}^{c} (\|x_{ik} - x_{jk}\|^2 + \|y_{ik} - y_{jk}\|^2)}{c}
\]

For each image, the pair having minimum distance is selected as the neighbor, and an edge is marked between them. Considering this attribute as the seed point, the region is expanded to include all those images in the neighborhood block, whose Euclidean distance fall within a particular threshold. This accounts for the out of plane images and handles arbitrary viewpoints. The threshold is defined relative to the Euclidean distance of the image from its existing neighbor as given in Equation 2.

\[
\text{threshold} = \min_{ED} (1 + \text{bound})
\]

where, \(\min_{ED}\) denotes the distance between an image and its closest neighbor and bound specifies the range of distances desired to be considered as "close viewpoints". The formation of linkage is directly derived from Equation 3.

\[
l(I_i, I_j) = \begin{cases} 
1 & \text{if } d(I_i, I_j) \leq \text{threshold} \\
0 & \text{otherwise}
\end{cases}
\]

The value of bound controls the compactness or spread of neighborhood and has to be adjusted manually depending on the distribution of input images; however a value of 25% has been observed to work for most cases.

In the second pass, the physical proximity constraint between successive video frames is applied. This implies that two consecutive frames of a video sequence represent two images in proximity and hence represent neighbors. Therefore:

\[
\text{Neighbor}(I_i, I_{i+1}) = 1, \forall i \in \text{Set of Frames}
\]
It may be noted that this second criterion simply improves the connectivity of the graph. In cases, where the image set is not from a true video sequence, and represents an arbitrary collection of images, only the first criterion would suffice. Figure 2 shows a graph that is generated for a car. Such a graph is generated for each object and stored as a model.

2.2 Graph Pruning using Virtual View Synthesis

As already pointed out, a view-centered approach leads to a space requirement that is larger than that of object-centered representation [19]. As may be noted in Figure 2, there is a large number of redundant images in the generated graph, requiring more storage capacity and computation power in subsequent steps. Hence the model has to be refined to keep the size of database as low as possible.

The procedure to reduce the size of dataset utilizes the virtual view synthesis technique described in [17]. The algorithm begins by analyzing for each image if it represents a morph of its neighbors or not. This means that it is checked if any image could be generated by interpolation or extrapolation of its neighbors or not. If it does, this implies that the information it contains could be represented by its neighbors and hence it could be removed without any loss of information. To test any image $I_i$, its SIFT features are extracted; in addition, virtual view synthesis is applied on its two adjacent images $I_j$ and $I_k$ to generate features and verify if they represent the features originally extracted from $I_i$ or not. It may be noted that instead of generating the complete virtual image, only the distinctive features are generated and used in further comparisons. This saves from retrieving the dense correspondences, which is computationally expensive [23].

Given an image pair $(I_j, I_k)$, with corresponding feature points $p$ and $q$, the images are aligned to have the corresponding points along corresponding scanlines and synthesize the features using Equation 5 for varying values of $\alpha$:

$$n_\alpha = p\alpha + (1-\alpha)q$$

The features generated in this manner are compared with the original features extracted from $I_i$. For this, $n_\alpha$ is iteratively generated and compared for varying values of $\alpha$. If there exists an $\alpha$ for which the Sum-of-Squared distance between $n_\alpha$ and $I_i$ is sufficiently small, it means $I_i$ could be generated using $I_j$ and $I_k$ and hence could be removed from the dataset.

After removing an image from the dataset, the neighborhood graph has to be updated, implying fixing the broken links that arise because of deletion of Image $I_i$. This involves creating new connections between original neighbors of $I_i$ and estimating the distances between them. This is accomplished by forcing the neighbors of $I_i$ to be neighbors. It is already known that: Neighbor($I_i$, $I_j$) = True and Neighbor($I_i$, $I_k$) = True. After removal of $I_i$: Neighbor($I_j$, $I_k$) = True. The Euclidean distance is updated as in Equation 6.

$$d(I_j, I_k) = d(I_j, I_i) + d(I_i, I_k)$$

This procedure is repeated till all the images in the data-base are exhausted. It should again be pointed out here that instead of working on complete images, the features are extracted and used for further computation.
2.3 Graph Merging for Cumulative model

Once neighborhood graphs have been generated, one for each object instance; they need to be combined for developing a comprehensive representation for the object category. This requires matching the corresponding viewpoints across graphs. This is because, a variety of single-view classification approaches exist and once it is known that which images correspond to the same viewpoint, they could be used to develop a single-view classifier. Thus, one such classifier could be developed for each viewpoint. One approach could be to identify a one-to-one node correspondence. This is both inefficient and computationally expensive because even after pruning, the two neighboring nodes of the graph may represent nearby viewing angles and should be regarded as images of same viewpoint. In the approach presented in this work, first nodes of one graph are clustered together on the basis of their visual similarity covering a viewing angle of approximately 30° and then the cluster-to-cluster correspondence is found across graphs. This approach of clustering rather than one-to-one correspondence not only bypasses the NP-complete problem of finding one-to-one node matching [3], but also provides means to perform a more robust single-view object categorization. This is because, instead of having a lot of clusters with just a few nodes in each, now there are a few clusters with a lot of nodes in each of them.

2.4 Image-based Clustering

The basic purpose of image clustering is to divide the image dataset into distinct groups, each representing a particular viewpoint. Knowing that similarity in appearance could be used to determine the closeness in viewpoint, the images that are apparently similar are identified as being proximally close and thus grouped together as a cluster. The image-based clustering step proceeds by using the distinctive features (SIFT) of the images, which have already been extracted in the prior step, to identify the centroid of each image. The Euclidean distance has been used as a distance metric as it directly corresponds to the closeness of viewpoints covering changes in both x- and y- directions simultaneously. The clustering has been done with K-means as it has shown to perform more predictably and reliably. Consider Figure 3 for the clustered graph of an airplane.

![Clustered Graph for an Airplane. Here each cluster forms a node corresponding to a particular viewpoint.](image)

2.5 Cluster correspondence across graphs

Once the clusters have been identified for each of the neighborhood graphs, it is required to combine them to have a cumulative representation of the whole object category. In this resulting graph, each node is a collection of the images corresponding to a common point of reference. Since, shape of objects typically remains the same across different instances of the category; hence to find the similarity between images, the gradient orientation histogram [12] has been used. Given a smooth intensity image I(x,y), the vector gradient I(x,y) is defined by:

\[
(u,v) = \nabla I(x,y) = \left( \frac{\partial}{\partial x} I(x,y), \frac{\partial}{\partial y} I(x,y) \right).
\]

Thus, the gradient of an image maps the image plane, parameterized by (x,y) onto the gradient space, parameterized by (u,v). Next the gradient magnitude and orientation can be computed as follows:

\[
GradMag = \sqrt{u^2 + v^2} \quad \text{and} \quad GradOrientation = \tan^{-1} v / u
\]

A grid based gradient orientation histogram of 50 x 50 pixels is extracted for each image, where orientation is divided into 8 bins of 45° each. The gradient histogram of one grid of an airplane from Caltech dataset [22] is shown in Figure 4.
Fig. 4. The image of airplane from Caltech dataset is divided into grids of size 50x50. Gradient Histogram for one grid is shown, where 8 bins of 45° each are formed.

Next, the histogram intersection [21] is determined for respective grids of each image pair. The problem that occurs here is that a change in scale or location of object in the image leads to the correspondences being missed. As a solution, the object has to be isolated from the background, which could be done through segmentation and normalized to a particular size. Two histograms Q and V can be compared using the intersection measurement as given in Equation 9.

\[ H(Q) \cap H(V) = \sum_{j=1}^{n} \min(H_j(Q), H_j(V)). \]  

A larger value of intersection denotes a closer match and vice versa. This gives the distance \(d(I_i, I_j)\) between images \(I_i\) and \(I_j\), where \(I_i \in \text{Object}_M\) and \(I_j \in \text{Object}_N\), both of the same category. Thus, distance \(D(C_i, C_j)\) between clusters \(C_i\) and \(C_j\) is computed using Equation 10 where \(C_i \in \text{Object}_M\) and \(C_j \in \text{Object}_N\).

\[ D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{i \in C_i, j \in C_j} d(I_i, I_j). \]  

Fig. 5. Part of the Cumulative model for cars. Each node represents the cluster of images taken from a common viewpoint.

Using Un-Weighted Pair Group Averaging (UWPGA) [16], the clusters having minimum average distance are identified as being of corresponding viewpoints and merged together to give the cumulative model. It may be noted that at this step, it is required to iterate only once under the hierarchical clustering scheme, as only the closest correspondences are desired to be identified. After the correspondences are determined for all the instances of the object category, the graphs are merged to generate the cumulative model. A link between cumulative nodes exists if the original images are connected in the neighborhood graphs. Figure 5 shows part of the final model generated for the category car.

### 3. TRAINING THE SINGLE-VIEW CLASSIFIERS

Each node of the cumulative graph now comprises of several instances of the objects from a common viewpoint, which can be used to train a single-view object classifier. A variety of approaches exist for classification from a single-view. Appearance-based methods have gained much popularity and in addition to shape-based methods, they are also being used for categorization. The algorithms following appearance-based approach either work directly on the descriptors of all image regions [17] or develop a “codebook” of the appearance patches. In this work, the codebook has been built following the strategy described in [1], which represents each object as a composition of its constituent parts. This approach is both simple and efficient. Initial attempts to part-based approach used the low level primitives like edge
fragments [10] to form a representation for the objects. Later on, attention was directed to utilize the more-expressive and content-rich higher level features [6]. Specially, the codebook approach of [1] gained much popularity and has been adopted by a number of researchers [2, 9, 11] and also in this work. It is computationally efficient as it provides a sparse representation of images in terms of high level parts instead of the low level pixel-based features, which allows performing subsequent operations quickly. It does not assume any probabilistic model and does not require any manual specification of parts, which makes it easily extensible. Moreover, in contrast to previous approaches this approach learns over a larger feature space resulting in a classifier that is robust to variations across images.

For single-view classification, first an interest point detector is applied on each image of a given cumulative-node, and patches are extracted around the features. In the experiments, the Harris corner detector [7] has been used with a patch size of 15x15. Next, patches are clustered together on the basis of their visual similarity with each other. Initially, each part is placed in a separate cluster, followed by merging of similar clusters. Similarity between parts is measured by intersection of gradient histograms. Since each patch is a small grid of 15x15 pixels; it is not possible to further extract features, thus making the histogram intersection approach a practical option. Each cluster then represents the conceptually and visually similar parts of the object and is assigned a unique ID. Thus, a vocabulary of parts is formed and the image is then represented using this vocabulary. To achieve this end, the patches are once again extracted around the corners and their respective clusters identified. These clusters are referred to as active clusters in the image. Moreover, the distance-direction relation between the pairs of parts is identified. Hence, each image is coded in terms of its active parts. The feature vector thus formed is a binary vector with 1 representing presence and 0 denoting absence of a particular part and a relation. Further details of the codebook algorithm are given in [1]. Once the feature vector is set up, it is used for training a classifier. The Support Vector Machine (SVM) has been trained for single-view classification, one for each node of the cumulative graph. All the SVMs are used together to perform multi-view classification. Support Vector Machines are a powerful tool for supervised learning and state of the art results have been reported for various tasks where SVMs have been applied. The training of the SVMs completes the model generation phase and this model is next used for testing.

4. VIDEO-BASED MULTI-VIEW OBJECT CATEGORIZATION

Traditional solutions to object categorization take a single image in the test phase for classification. In order to strengthen the confidence measure of the results, a video clip has been used for categorization. The major advantage of this technique is that the video provides information of multiple views. Many objects in real world look alike, if observed from a particular viewpoint and completely different when observed from some other point of reference. Considering only a single image leads to an increase in the number of false positives and negatives generated by the system. Using a video for classification, the fact can be exploited that the two adjacent images in the video sequence represent proximally closer views of the object. Since the model has been generated such that the closer views are linked together, hence the adjacent frames of the video sequence should point to the same (or adjacent) nodes of the graph. Thus, a correct classification results in a smooth transition across the multiple nodes, following an unbroken trajectory in the model. On the other hand, an incorrect match results in jitters across the multiple frames, which helps in identifying the incorrect matches. See Figure 6 that shows a smooth trajectory for a correct identification and jitters across the model for a mismatch. The approach for developing the topological structure of the images in database provides ease of traversing while using video sequence. Given the stored networks of objects and a test video sequence, only the correct object follows a smooth trajectory along the graph and others suffer from discontinuities.

![Fig. 6. The dark bounding boxes represent the images identified by the algorithm. (a) Smooth trajectory through the graph for a correct match for a car. (b) Jitters across graph representing an incorrect match for a non-car object.](image-url)
5. EXPERIMENTS AND RESULTS

To evaluate the approach for multi-view object classification, the motorbikes of the publicly available benchmark set from the PASCAL Visual Object Classes (VOC) Challenge were considered. The 2005’s set having 202 images of 30 motorcycles taken from 11 viewpoints, and the 2006's dataset having 239 images of 46 bikes from average 6 viewing angles were used. Moreover the Airplane and TV/Monitor categories of PASCAL 2007 were considered. Since, PASCAL datasets do not have video sequences available, the experiments were conducted with series of images taken from distinct viewpoints. Furthermore, to test the effectiveness of using video sequence for classification, the Video Verification of Identity (VIVID) dataset [15] from DARPA and NASA's airplane sequence maintained at Open-Video Project [18] were used. Out of the average 2000-frame long video sequences, 150 frames were extracted for training. Moreover, in order to portray the randomness of the real-world image capture, on average, 100 frames were extracted from video sequences shot around motor bikes and cars following arbitrary trajectories. Figure 7 shows a sample of images used.

![Fig. 7. A few test images: PASCAL 2005 and 2006 bikes, 2007's cars and airplanes, PASCAL 2007's TV/Monitor, cars' sequence from VIVID, Airplane sequence by NASA, and on-site motorbike and car sequences.](image)

The algorithm begins with developing the neighborhood graph for each object instance separately by determining the SIFT feature points and computing the Euclidean distance between them. See Figure 8 that shows neighborhood graphs for cars' sequence of VIVID dataset, airplane sequence of NASA (at PRIP), on-site motor bikes and cars.

![Fig. 8. Neighborhood Graph generated for (a) Cars of VIVID dataset (b) On-site car (c) Airplane of NASA at Open Video project (d) On-site Motorbike.](image)

Experiments show that the algorithm can generate the neighborhood graph for an object instance with a precision of 97.86%. The graphs thus generated are quite dense with a lot of redundant images. As shown in Figure 9, the time to generate and train the model and the space occupied increases drastically with the increase in number of images in the dataset. This is because a lot of computations are to be performed on all the images of the dataset. Hence, the surplus nodes are identified and removed. For this the virtual view synthesis algorithm is applied using which those nodes are identified which could be represented using their neighbors.

![Fig. 9.](image)

Experiments show that the algorithm can generate the neighborhood graph for an object instance with a precision of 97.86%. The graphs thus generated are quite dense with a lot of redundant images. As shown in Figure 9, the time to generate and train the model and the space occupied increases drastically with the increase in number of images in the dataset. This is because a lot of computations are to be performed on all the images of the dataset. Hence, the surplus nodes are identified and removed. For this the virtual view synthesis algorithm is applied using which those nodes are identified which could be represented using their neighbors.

Applying virtual view synthesis for detection and removal of redundant nodes led to a reduced graph of up to 64.54% of the original size depending on the density of viewpoints of captured images. This reduction helped in developing the model with reduced storage requirements as well as faster training. This is because several phases of the algorithm
require repetitive access to ‘all’ the images in the dataset, e.g. assigning images to clusters, determining gradient orientation histogram, comparing image histograms, developing the feature vector and training the classifier etc. Table 1 shows the dataset statistics for the memory utilized as well as time consumed for model generation using the original neighborhood graph and the pruned one. The algorithm was executed on a system having processor @ 1.66 GHz and 256 MB RAM. It may be noted that since PASCAL's datasets have most images of different object instances, so a graph has not been generated for the independent objects. Consequently, no pruning has been applied for PASCAL’s images.

Table 1. Summary of space/time requirement for original and pruned networks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Original Nodes</th>
<th>Number of nodes after pruning</th>
<th>Space Requirement (MB)</th>
<th>Time consumed in training (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
<td>Pruned</td>
<td>Original</td>
<td>Pruned</td>
</tr>
<tr>
<td>Cars sequence from VIVID</td>
<td>85</td>
<td>50</td>
<td>71.09</td>
<td>41.78</td>
</tr>
<tr>
<td>Motorbike sequence 1</td>
<td>60</td>
<td>32</td>
<td>355</td>
<td>188.31</td>
</tr>
<tr>
<td>Motorbike sequence 2</td>
<td>85</td>
<td>63</td>
<td>378</td>
<td>279.54</td>
</tr>
<tr>
<td>Mugs sequence 1</td>
<td>46</td>
<td>31</td>
<td>59.2</td>
<td>39.63</td>
</tr>
<tr>
<td>Mugs sequence 2</td>
<td>32</td>
<td>25</td>
<td>6.6</td>
<td>5.29</td>
</tr>
<tr>
<td>Airplane seq. 1 from NASA</td>
<td>35</td>
<td>23</td>
<td>10.5</td>
<td>6.96</td>
</tr>
<tr>
<td>Airplane seq. 2 from NASA</td>
<td>26</td>
<td>12</td>
<td>11.3</td>
<td>5.01</td>
</tr>
<tr>
<td>On-site car sequence</td>
<td>60</td>
<td>40</td>
<td>79.7</td>
<td>52.72</td>
</tr>
</tbody>
</table>

Fig. 9. Space-Time dependency on Number of Images.

As described in Section 2.4, the nodes within a given graph were clustered using the k-means clustering algorithm. The number of clusters were computed using the v-fold cross validation scheme with a value of v = 5. Clustering was performed with a precision of 80.86% and recall of 63.21%. The formulas of precision and recall from [1] have been customized as in Equation 11 for clustering. In this equation misclassifications are considered in two aspects: for any image \( I_i \) that should have been placed in Cluster \( C_i \) but it was placed in any other cluster \( C_j \); it is considered as False Negative with respect to \( C_i \) and False Positive with respect to \( C_j \).

\[
\text{Recall} = \frac{\text{CC}_i}{\text{CC}_i + \text{IOC}_i} \quad \text{and} \quad \text{Precision} = \frac{\text{CC}_i}{\text{CC}_i + \text{IC}_i}
\]  

(11)

Where, the abbreviations denote images:
- \( \text{CC}_i \) = Correctly placed in \( C_i \),
- \( \text{IOC}_i \) = Incorrectly placed in cluster other than \( C_i \),
- \( \text{IC}_i \) = Incorrectly placed in \( C_i \).

Next, to generate the cumulative model, cluster correspondences are identified across graphs as described in Section 2.5, with 82.38% precision, with symmetry being the major reason for incorrect identification. In this case, precision was computed as follows:
\[ \text{Precision} = \frac{CC}{CC + IC} \] 

Where,
CC ≡ Number of Correct correspondences identified,
IC ≡ Number of incorrect correspondences identified.

Consider Figure 10 that shows some cluster correspondences identified correctly for on-site motorbike sequence, PASCAL's airplanes and motorbikes, along with a few examples of incorrect clustering as for the two symmetric viewpoints of mugs and highly cluttered motor bikes of PASCAL dataset. The corresponding clusters then form the cumulative nodes of the final model, which are connected by a link if the constituent images are connected. It should be noted that only the presence or absence of link is considered. It is possible to assign weightage to the links depending on the inter-cluster strengths, however that option was not explored.

![Cluster Correspondences identified across neighborhood graphs](image)

(a) Correspondences identified correctly for PASCAL's airplanes and motor bikes (b) Misidentification for a specific instance of mug and highly cluttered PASCAL's motor bike.

Next, for each viewpoint (i.e. node of the cumulative graph) a classifier was trained. For this purpose, the codebook approach of [1] was used as described in Section 3. The main motivation behind adopting this approach was its compactness, simplicity and efficiency, in addition to being automated. Initially, SIFT operator was used for feature detection, but it resulted in a lot of features, many of which were redundant; hence, Harris operator was used, which returned lesser interest points per image. Another reason was that Harris operator returned interest points that better corresponded to the physical components of the objects as compared to SIFT.

Next, patches of size 15x15 were extracted around the detected features and clustered together with respect to their visual similarity using gradient histogram intersection. As described in Section 3 and mentioned in [1], five distance values were considered ranging from 1 to 5 times that of patch size. Furthermore, the 4 direction bins were considered as of 45° each. Only unidirectional relationships were considered, that is, from top left corner to the other end. This allowed the number of direction bins to reduce from 8 to 4. Hence, a total of 20 distance-direction pairs were formed. Finally, each image was coded in terms of its active parts and distance-direction relationship, packed in the feature vector and passed on to SVM for training.

Finally, for object classification, using a video sequence gave results with 85.73% precision, as against single image where precision was 76.14%. Thus a significant improvement in performance was observed by use of video for classification. Figure 11 shows the results on PASCAL 2005 and 2006 motorbikes, PASCAL 2007 Airplane and TV/Monitor, VIVID's cars and NASA's airplanes. The results have been compared with those reported by Thomas et al. [25], where they were able to perform multi-view categorization. An overview to their approach is given in Section 1. The approach in this work shows comparable performance with the significant advantage of having eliminated the need of manual alignment of images, as was desired by them.
Consider Figure 12 that shows few examples of correct and incorrect classification of various objects using the algorithm. As can be seen in Figure 12 (a) the algorithm was able to classify the different instances of motorbikes, cars and airplanes from various viewpoints under partial occlusion and background clutter as well. However, the algorithm could not handle significant scale change and large occlusions as shown in Figure 12 (b).

Fig. 12. Classification of Objects (a) Correct Categorization (b) Misclassification.

6. CONCLUSIONS

In this paper, an approach for video-based multi-view object classification is presented. A sparse 2D model is generated, based on geometry and image measurables only. The system does not require camera calibration or prior knowledge of object pose. It does not assume a known 3D CAD model and does not place any constraints on motion of objects while video capture. The use of view synthesis algorithm allowed identifying the redundant images in the model and removing them to give a computationally economical model in terms of space and time. The strength of this approach is that it eliminates the need of manual alignment of common viewpoints across object instances. Another strength is that the video sequences have been used, instead of images, for object classification and it has been shown that this approach can efficiently remove false positives and thus increase precision. The strategy for developing the topological structure of the images in database provides ease of traversing while using video sequence. One of the major strengths of this approach lies in the flexibility of framework. The model is adaptable; hence it could be updated online during testing. Any new image could be linked to the network to provide additional information. This could be a future extension of the system. Moreover, currently only the SIFT points are being utilized as the distinctive feature for image representation. Other information cues like color and shape etc may be utilized and combined together to give an even more strengthened classifier.
REFERENCES