COMPUTER VISION

SPRING 2018

Final Project Report

VIDEO-GOOGLE: A TEXT RETRIEVAL APPROACH TO OBJECT MATCHING IN VIDEOS

GitHub: https://github.com/Marauderer97/CV-Project

Team Members:

Avani Agarwal(201530194)
Naila Fatima (201530154)
Soumya Vadlamannati(201501044)

Project Mentor:

Vishal Batchu

Table of Contents

ABSTRACT:

In this project, we describe an approach to object and scene retrieval which searches for and localizes all the occurrences of a user outlined object in a video. The object is represented by a set of viewpoint invariant region descriptors so that recognition can proceed successfully despite changes in view-point, illumination and partial occlusion. The temporal continuity of the video within a shot is used to track the regions in order to reject unstable regions and reduce the effects of noise in the descriptors.

The analogy with text retrieval is in the implementation where matches on descriptors are pre-computed (using vector quantization), and inverted file systems and document rankings are used. The result is that retrieval is immediate, returning a ranked list of key frames/shots in the manner of Google.

INTRODUCTION:

This project aims to use a text retrieval approach to retrieve keyframes or shots of a video which contain a particular object. The retrieval should be done at the speed and accuracy with which Google retrieves web pages on a word search. It should be taken into account that an object may appear different across frames because of illumination conditions and viewpoint variation but the use of descriptors allows us to overcome this problem. For each frame, a set of descriptors (which are somewhat invariant to illumination and viewpoint conditions) are computed for various points and regions in the frame. Recognition of a particular object is done by using the descriptors of the input image (which contains the object to be searched in the database) and finding its nearest neighbour matches in the entire database. In order to improve the results, various constraints can be incorporated in order to achieve spatial coherence or global relationships.

We have used an approach which is analogous to that used by Google to find web pages containing certain words. A word is analogous to a cluster of feature descriptors whereas a web page/document is analogous to a frame. A benefit of this approach is that matches between descriptors have been precomputed- this allows us to retrieve frames and shots with minimum delay.

DATASET AND PARAMETERS

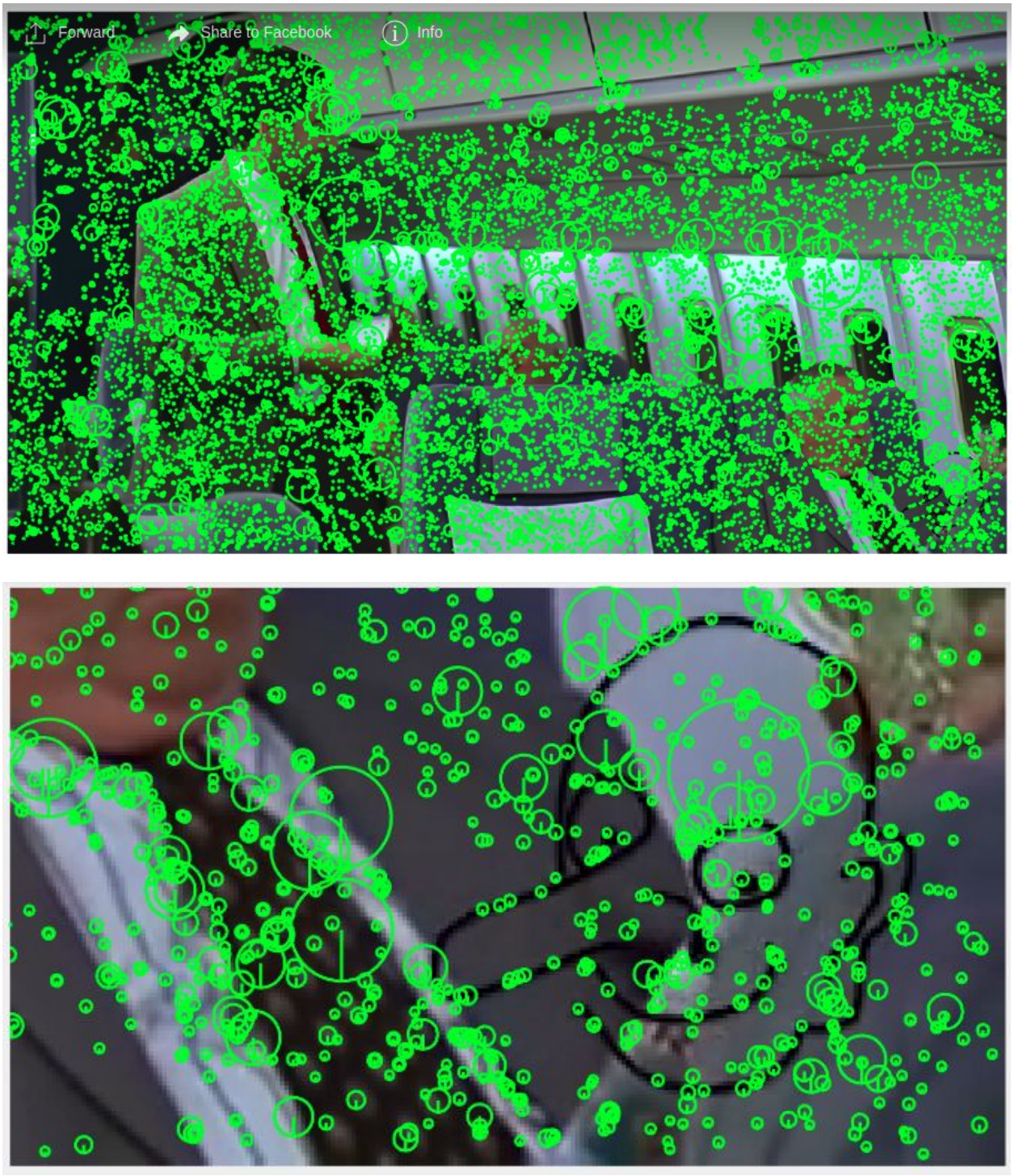
- The video used was a Mr. Bean clip titled ***"First Class Flight"*** which can be found at the URL: **https://www.youtube.com/watch?v=QE6PvNohffc**
- The video was sampled at ***2 fps***
- It should be noted that every frame considered is first checked for uniqueness. Only if it is different from its parent frame by a factor of about 5% of the maximum difference.

METHOD:

Part-1 Viewpoint Invariant Descriptors

a. Shape Adapted

Shape adapted regions are constructed by using elliptical shape adaptation about an interest point. They are centered around center like regions and are constructed by iteratively detecting the center, shape and scale for each ellipse. The scale of the ellipse is calculated by the local extremum over a Laplacian whereas the shape is computed by maximizing the intensity gradient isotropy over an elliptical region. The implementation details have been described in detail in [3, 4].

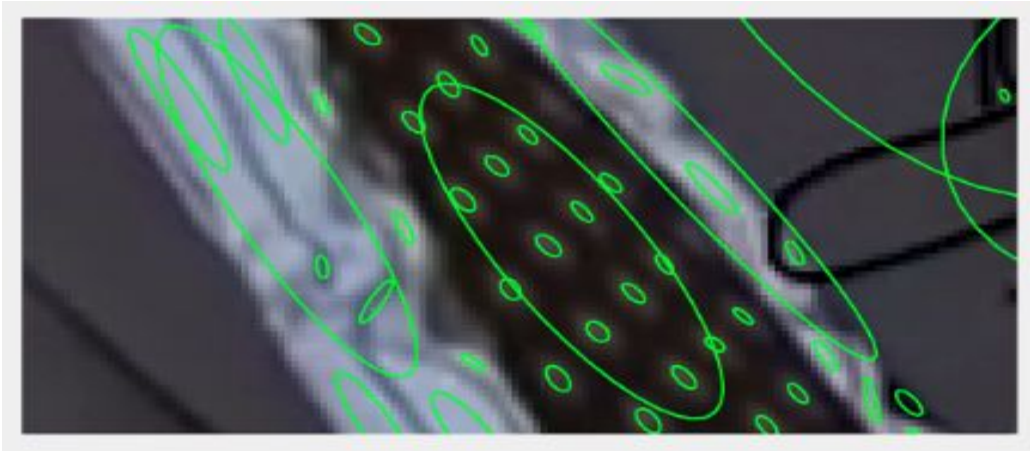


NOTE: We can observe that in the above images, the ellipses are centered around corners and edge points, because of the Harris-Laplace detector.

b. Maximally Stable

Maximally stable regions, on the other hand, are centered around blobs and high contrast features. They are constructed by selecting areas from an intensity watershed image segmentation. The regions are those for which the area remains approximately stationary as the intensity threshold in the threshold varies. The implementation details have been described in detail in [2]

Result:



NOTE: We can observe that in the above images, the ellipses are centered around regions which are similar to one another- they are centered around blobs.

c. Combination with SIFT

As these two regions detect different areas in the image - the former detects center like regions whereas the latter detects blobs and high contrast features- they are used together in order to represent different areas of the frame in different manners. It should be noted that both types of regions are computed at twice the frame size as this allows us to obtain more features from each image. Each of these regions are elliptical in shape and affine invariant. Each region is represented by a 128- dimensional SIFT feature vector. The SIFT feature vector has been used as it is invariant to a shift of a few pixels in the region position. This property is necessary as it allows us to accurately match features which have been shifted due to camera noise. SIFT is also robust to occlusion and clutter which allows us to recognize objects even if they are not completely in the view of the camera. The SIFT descriptor along with the affine covariant regions (maximally stable and shape adapted) give region descriptor vectors which are invariant to affine transformations in the image.

Part-2 BUILDING A VISUAL VOCABULARY

The second part involves constructing a visual vocabulary in order to quantize the descriptors into clusters which will be the visual 'words' for text retrieval. We have earlier mentioned that clusters and frames are analogous to words and documents in text retrieval, respectively. Regions are tracked through contiguous frames, and a mean vector descriptor computed for each of the regions. 10% of the regions which have the largest diagonal covariance matrix are rejected in order to reduce any kind of instability.

Each of the 128-dimensional descriptor vectors is clustered by using k-means clustering. The distance metric which has been used is the Mahalanobis distance as it allows the noisy components of feature descriptors to be weighted down while simultaneously decorrelating the components. It has been assumed that covariance is the same for all frames.

The Mahalanobis distance metric is given by the following formula:

$$d(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2) = \sqrt{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \Sigma^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)}.$$

As the clustering step can be highly time consuming, one can take a subset of the dataset which will cause the number of feature descriptors to be less.

The shape adapted (SA) and the maximally stable (MS) regions are clustered separately as they cover different regions of an image which are independent of one another. It should however be noted that the ratio of the number of SA clusters to the number of MS clusters should be equal to the ratio of the number of SA regions to the number of MS regions. For example, if we had 3000 SA regions and 2000 MS regions, we would want the number of SA and MS clusters to be in the ratio 3:2.

- **Noise Handling**

Noise handling is done by removing the top 10% clusters as well as the 10% clusters which have the highest variance. This is done as the top 10% clusters are the most common clusters. In text retrieval, these clusters would be analogous to commonly occuring words such as *'an'*, *'a'* and *'the'*. The 10% clusters which have the highest variance are not proper clusters which if kept lead to a higher number of errors.

Part-3 VISUAL INDEXING USING TEXT RETRIEVAL METHODS

The third component of the algorithm involves each document being represented by a vector of word frequencies. We use the standard weighting known as 'term frequency - inverse document frequency' which is computed as follows:

If there are k visual words, then each frame is represented by a k-d vector
$V_d = \{t_1, t_2 \ldots t_k\}$ where each term is

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

Cosine distance is used to measure the similarity of the query vector V_q to the rest of the document (frame) vectors

Where

n_{id} = no. of occurrences of word i in frame d n_d = no. of words in frame d

n_i = no. of frames with term i

N = total no.of frames

n_d = no. of words in frame d

The intuition is that word frequency weights words occurring often in a particular document, and thus describe it well, whilst the inverse document frequency down-weights words that appear often in the database.

Part-4 OBJECT RETRIEVAL

a. Stop lists

A stop list has been used in order to reduce the number of mismatches as well as the number of descriptors present. The top 5% and bottom 10% clusters have been suppressed as they are analogous to frequently occurring words (such as *'a'*, *'an'* and *'the'*) in text retrieval.

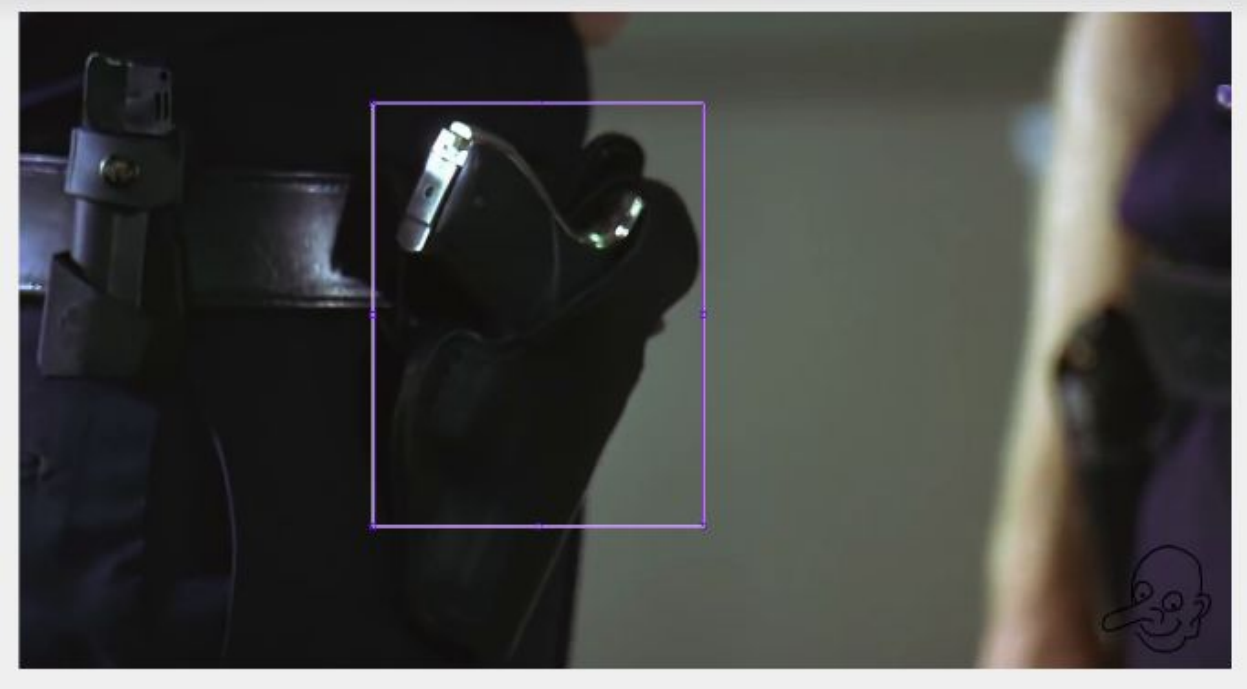
b. Object Retrieval using a Query vector

Using the query vector Vq, and computing its cosine distance with all the frame vectors Vd, we can output a ranked list of frames.

We create an inverted index list which stores all possible frames in which a particular cluster is found. In the text retrieval analogy, this is equivalent to having a list which tells us which documents a particular word appears in. Using the query vector, we are able to find the clusters to which the feature descriptors of the query image belong to. Using the inverted index list, we know the frames in which the clusters appears. However, we have to weight the retrieved frames in order to give more importance to clusters which rarely occur as compared to the frequently occurring clusters. We use tf-idf to obtain a ranked list of frames which give us the best possible matches.

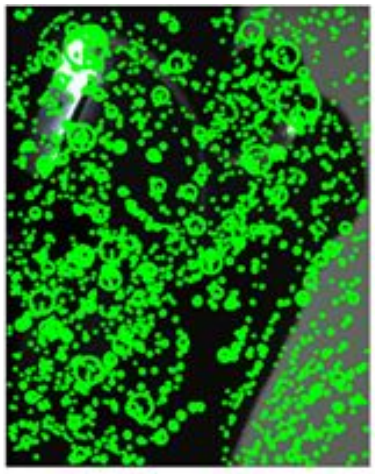
This is the workflow for object retrieval:

1. We select the region to be searched, by drawing a bounding box around the object:

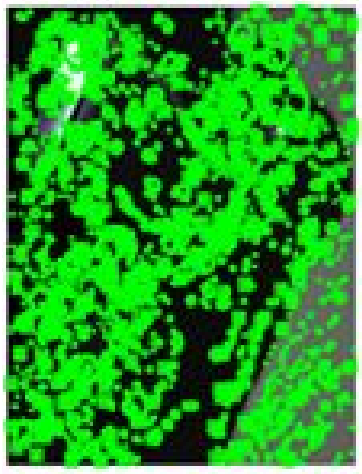


2. We compute viewpoint invariant descriptors for the selected region of the image (SA + MS):

Shape Adapted Regions

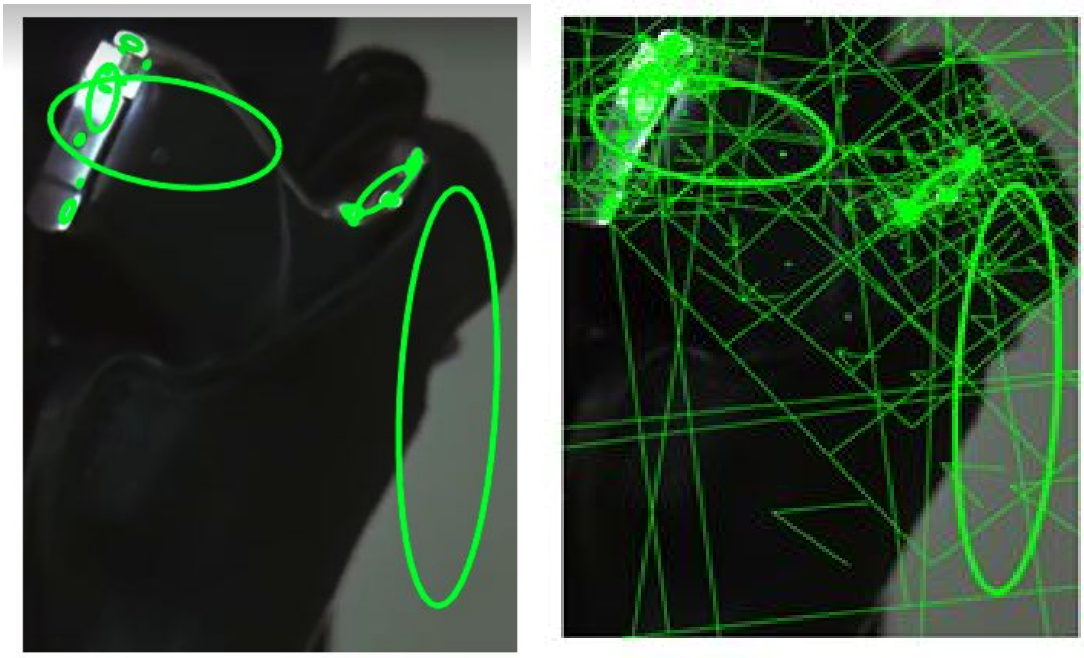


Shape Adapted Regions with SIFT

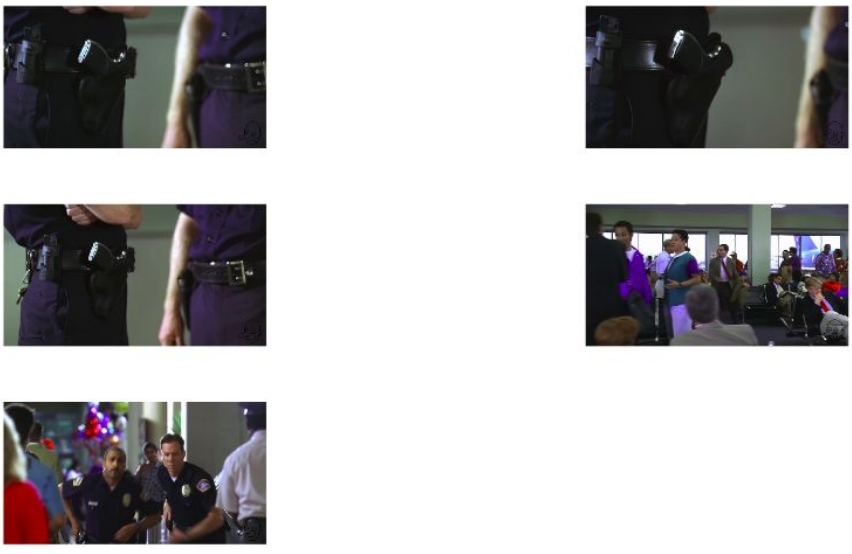


Maximally Stable Regions

Maximally Stable Regions with SIFT



3. We then find the clusters to which the regions in the bounding box belong.
4. Using inverted list we find all the frames in which those features belong.
5. All the frames corresponding to those clusters will be the result for the retrieval step and would be ranked later based on how close they are to the search query. We display the top 5 frames that we obtain.



Part-5 EXPERIMENTAL EVALUATION OF SCENE MATCHING USING VISUAL WORDS

In order to evaluate the retrieval performance, we use an entire frame as a query region and consider the retrieval to be correct if all the retrieved frames show the same object/region. The ground truth is determined by hand for the frame set. The evaluation is measured over all the frames, using each frame at a time as a query region. The average normalized rank is used as the evaluation metric which can be calculated by using the following formula:

$$\widetilde{Rank} = \frac{1}{NN_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2} \right)$$

In this,

N -> size of the image set

N_{rel} -> number of relevant images for a particular query

R_i -> rank of *i*th relevant image

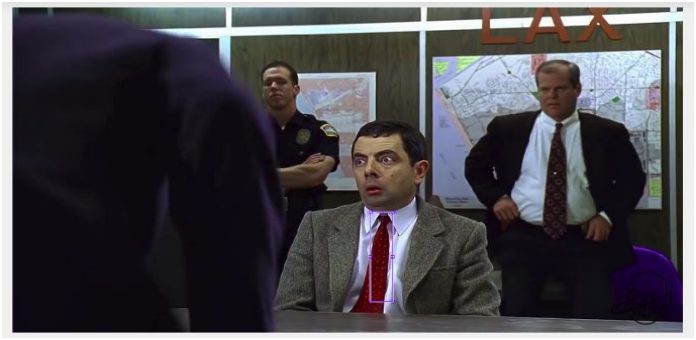
If the average normalized rank is zero, it means that all of the N_{rel} images are returned first- a desired result. The average normalized rank lies in the interval 0 to 1, with 0.5 denoting random retrieval of images.

Results and Observations

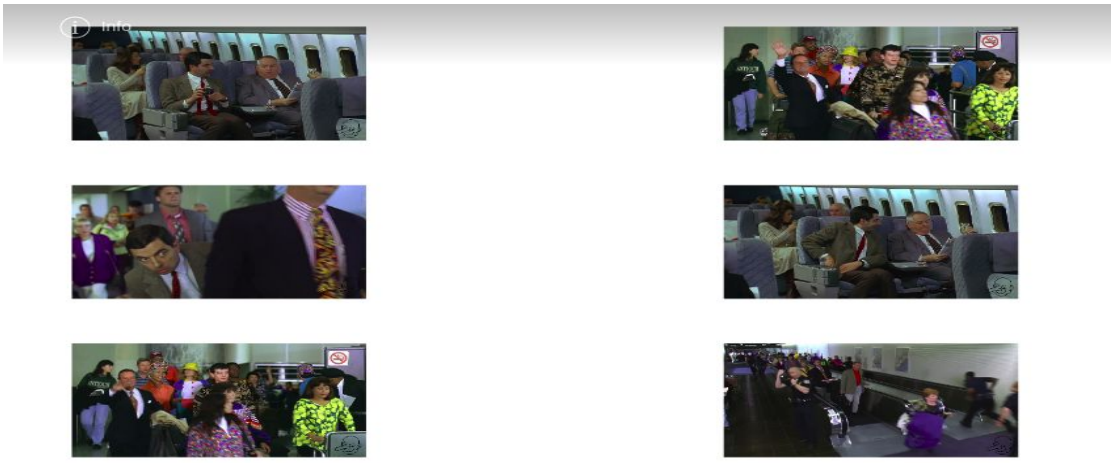
We have implemented Video-Google in a total of four ways

1. Using only SIFT descriptors.
2. Using SIFT descriptors for Shape Adapted Regions (SA + SIFT)
3. Using SIFT descriptors for Maximally Stable Regions (MS + SIFT)
4. Using SIFT descriptors for Shape Adapted Regions and Maximally Stable Regions (SA + MS + SIFT)

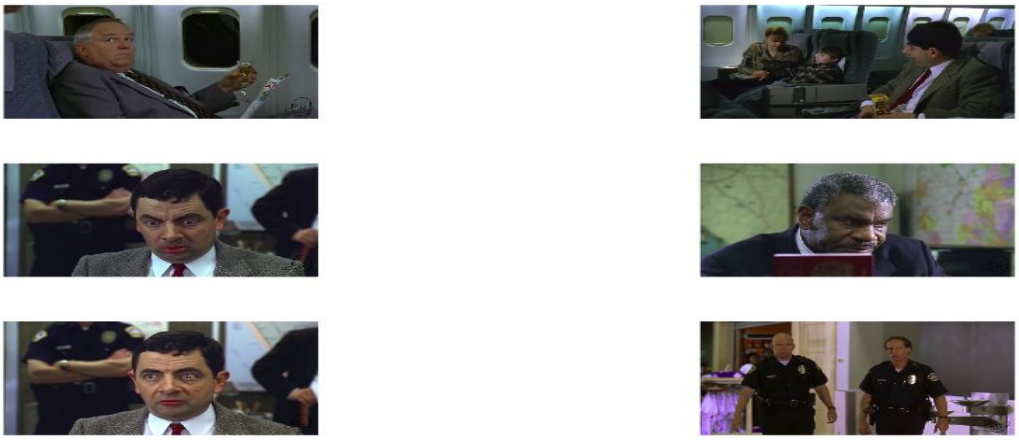
Input Image: The object to be searched is the tie.



Using only SIFT:



Using Shape Adapted and SIFT:



Using Maximally Stable and SIFT:






Using Shape Adapted, Maximally Stable and SIFT:






We observed that we obtained the best results with SA + MS + SIFT.

This also corresponds to the Ranks that we calculated for the four methods.

The rank evaluation table that we have computed is shown below for different types of regions:

METHOD	**RANK**
Shape Adapted	0.326
Maximally Stable	0.272
Shape Adapted + Maximally Stable	0.218
Only SIFT	0.261

Limitations:

Though video-google works well for a wide range of queries but there are some cases where it fails to produce the desired results. Some of them are mentioned here.

- In case of a wide difference in viewpoint and occlusion, we are not able to detect all the relevant frames for the object.
- If the image is even slightly blurred, some spurious features are detected, especially using shape adapted regions.

Related work:

Over time, several papers which describe different techniques to form object retrieval have been published. [6] uses an integrated matching procedure based on adjacency matrix of a bipartite graph between the tiles constructed on an image as well as a two level grid framework is used for color and texture analysis. [7] uses a region proposal network to learn which regions should be pooled to form the final global descriptor. [8] uses a quantization method based on randomized tree which improves the quality of retrieval.

Acknowledgements:

We would like to give our sincere thanks and gratitude to the people who have been a part of this project from its inception.We would like to take this opportunity to add a special note of thanks to our Professor Avinash Sharma and our mentor Vishal Batchu for their invaluable guidance, support and encouragement.

References:

1. .Sivic, Josef, and Andrew Zisserman. "Video Google: A text retrieval approach to object matching in videos." *null*. IEEE, 2003.
2. Matas, Jiri, et al. "Robust wide-baseline stereo from maximally stable extremal regions." *Image and vision computing* 22.10 (2004): 761-767.

3. Mikolajczyk, Krystian, and Cordelia Schmid. "An affine invariant interest point detector." *European conference on computer vision*. Springer, Berlin, Heidelberg, 2002.
4. Schaffalitzky, Frederik, and Andrew Zisserman. "Multi-view matching for unordered image sets, or “How do I organize my holiday snaps?”." *European conference on computer vision*. Springer, Berlin, Heidelberg, 2002.
5. Baumberg, Adam. "Reliable feature matching across widely separated views." *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Vol. 1. IEEE, 2000.
6. Hiremath, P. S., and Jagadeesh Pujari. "Content based image retrieval using color, texture and shape features." *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on*. IEEE, 2007.
7. Gordo, Albert, et al. "Deep image retrieval: Learning global representations for image search." *European Conference on Computer Vision*. Springer, Cham, 2016.
8. Philbin, James, et al. "Object retrieval with large vocabularies and fast spatial matching." *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007.