CV Project Presentation -2

Team Members:
Avani Agarwal(201530194)
Naila Fatima(201530154)
Soumya Vadlamannati(201501044)

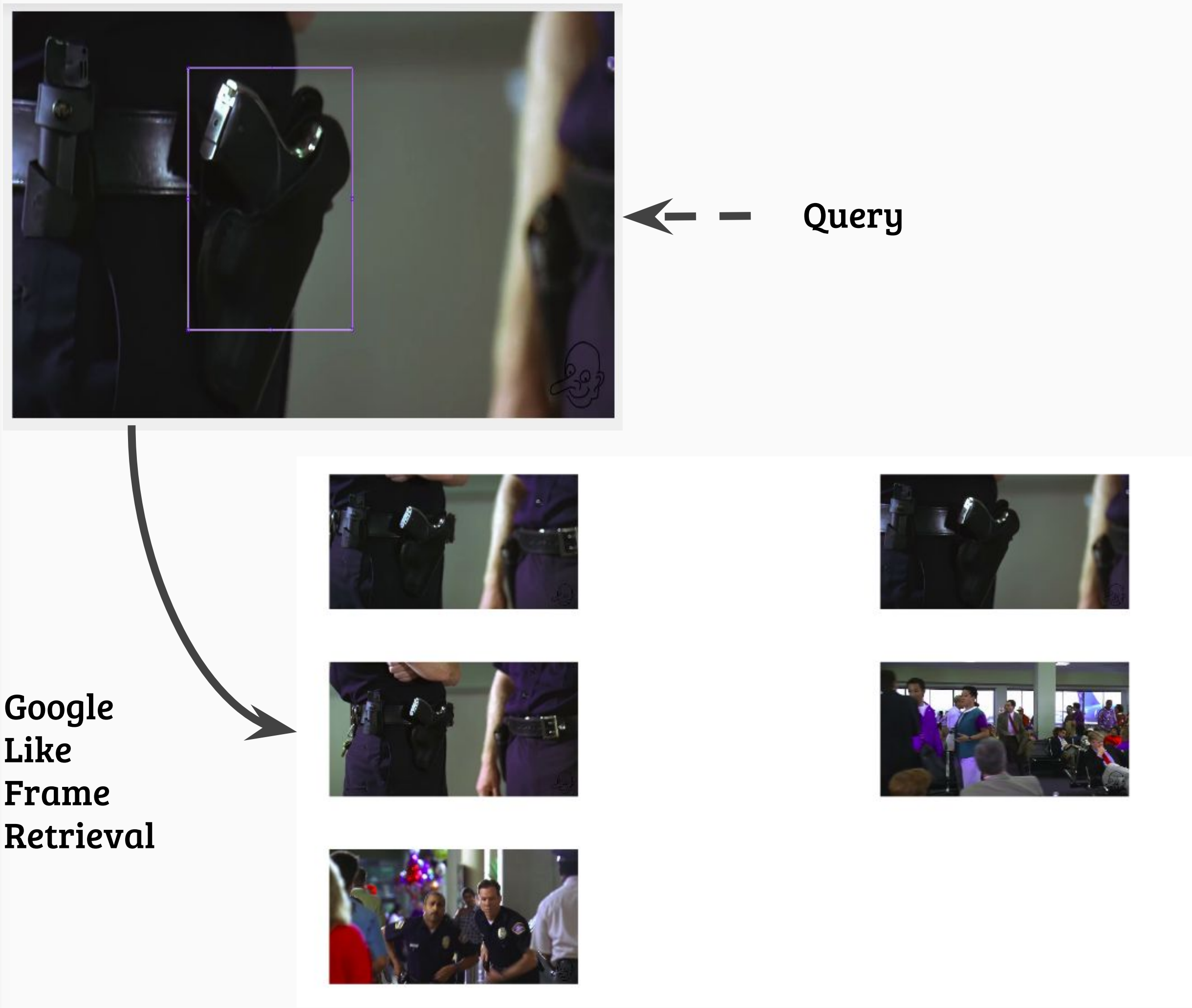
Team name: Still_Thinking
Project no. 38

Project Mentor:
Vishal Batchu

Video Google: A Text Retrieval Approach to Object Matching in Videos

Aim

We aim to achieve object and scene retrieval accomplished by searching and localizing all occurrences of a user outlined object in a video.



OUR DATASET

We use the video -”**First Class Flight - Mr. Bean**” to generate the frames for our dataset.

The video was sampled at **2 fps**.

It should be noted that every frame considered is first checked for uniqueness. We used it only if it is different from its parent frame by a factor of 5% of the maximum difference.

Workflow

Split video into frames

↓

Compute viewpoint invariant descriptors (MS and SA) and represent them using 128 dim SIFT vector

↓

Perform K-means clustering on the SIFT vectors. Discard weak clusters.

↓

Compute vector representation V_d for each frame using tf-idf

↓

Compute cosine distance of query vector, apply spatial consistency and use a stop list to obtain final ranking.

VIEWPOINT INVARIANT DESCRIPTION

The first key step in our procedure, is computing viewpoint covariant feature regions for each frame. These are of two kinds:

1. Shape Adapted Region
2. Maximally Stable Region

Viewpoint Invariant Descriptors - Shape Adapted

Centered on **corner** like regions.

It is constructed by elliptical shape adaptation about an interest point.

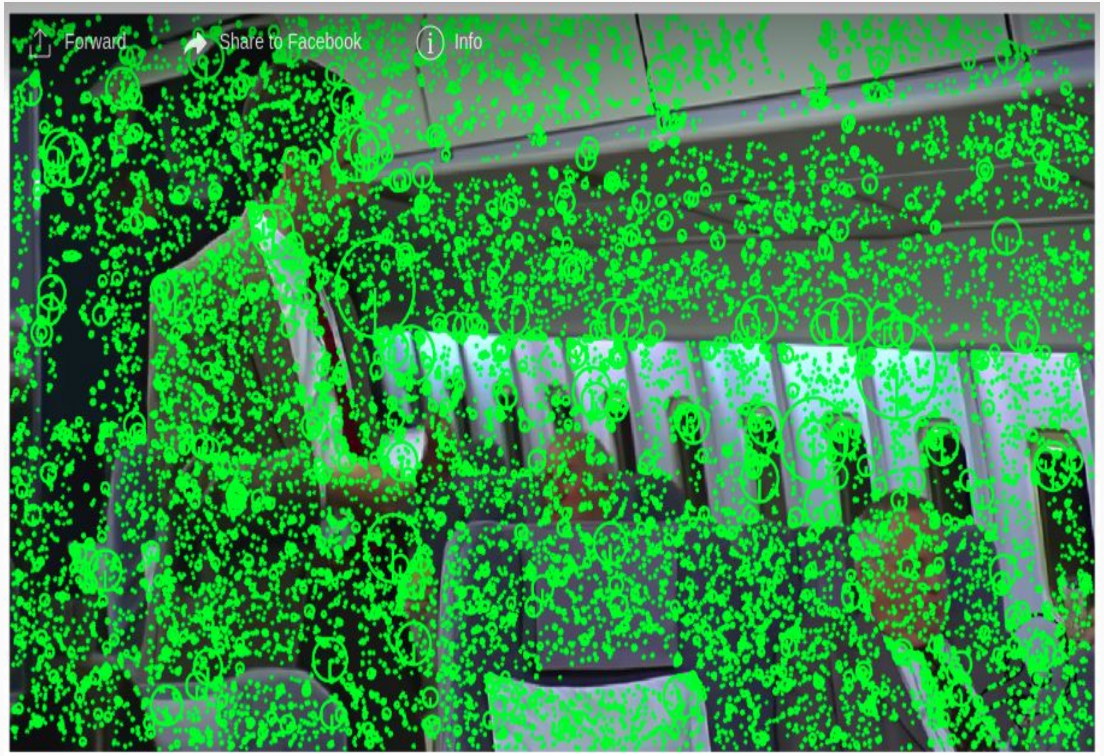
In this method we iteratively detect ellipse center, shape and scale.

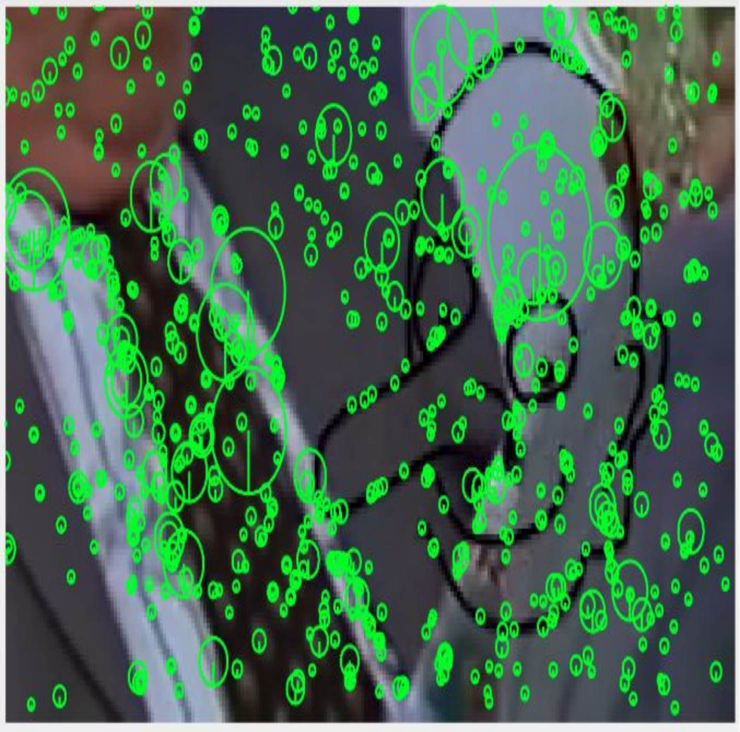
Shape -> Maximize intensity gradient isotropy

over an elliptical region

Scale -> Local extremum over Laplacian

Shape Adapted Regions





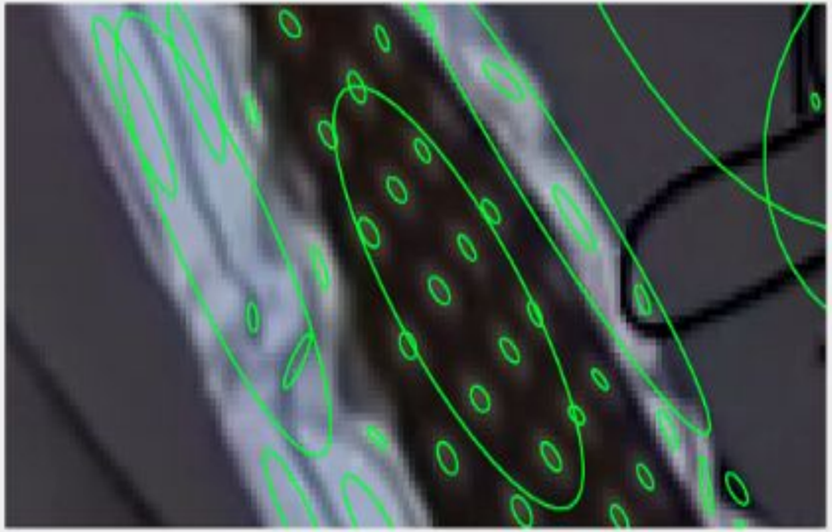
Viewpoint Invariant Descriptors - Maximally Stable

Centered on **blobs and high contrast** features.

It is 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.

Maximally Stable Regions



Viewpoint Invariant Descriptors

- Both elliptical features are computed at **twice** the frame size.
- Each affine invariant elliptical region is represented by a 128 dimensional SIFT feature vector.

WHY SIFT?

- Invariant to a shift of a few pixels in the region position
- SIFT descriptor with affine covariant regions gives region description vectors which are invariant to affine transformations of the image.

BUILDING A VISUAL VOCABULARY

The next step is to build a visual vocabulary by clustering the identifiers from each frame into groups.

Building a Visual Vocabulary

- Regions are tracked through contiguous frames, and a mean vector descriptor $\bar{\mathbf{x}}_i$ computed for each of the i regions
- K-means clustering was used in order to cluster the feature descriptors into regions.
- The k value we used was 100
- The distance Metric for clustering

Mahalanobis

MAHALANOBIS

- Common covariance Σ for all frames
- The distance between two mean track descriptors $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by:

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

Mahalanobis suppresses noise and decorelates individual components.

Building a Visual Vocabulary

- SA and MS regions are clustered separately because they cover different regions of the scene which are independent of one another.
- The ratio of SA to MS clusters depends on the ratio of the regions found.

Building a Visual Vocabulary - Noise Handling

- After k means clustering we removed the top 10% clusters (analogy with text retrieval).
- And then 10% clusters with highest variance are also deleted (not proper clusters).
- We had **97209** MS regions and **2594670** SA regions across 171 distinct frames (after all the removing similar frames and noise removal).

Building an inverted index list

- We create an inverted index list which allows us to know the frames in which a particular cluster appears.

- In the text retrieval analogy, this is equivalent to having a list of documents which contain a particular word.

VISUAL INDEXING USING TEXT RETRIEVAL METHODS

We now represent each frame as a weighted vector of identifiers.

Visual Indexing

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

OBJECT RETRIEVAL

Here the objective is to detect the user specified object in all the frames of the video and then rank them accordingly.

Object Retrieval Workflow

Selecting region to be searched

↓

Compute viewpoint invariant descriptors for the selected region

↓

Find the clusters to which the features in the selected region belong

↓

Using inverted list we find all the frames in which those features belong

↓

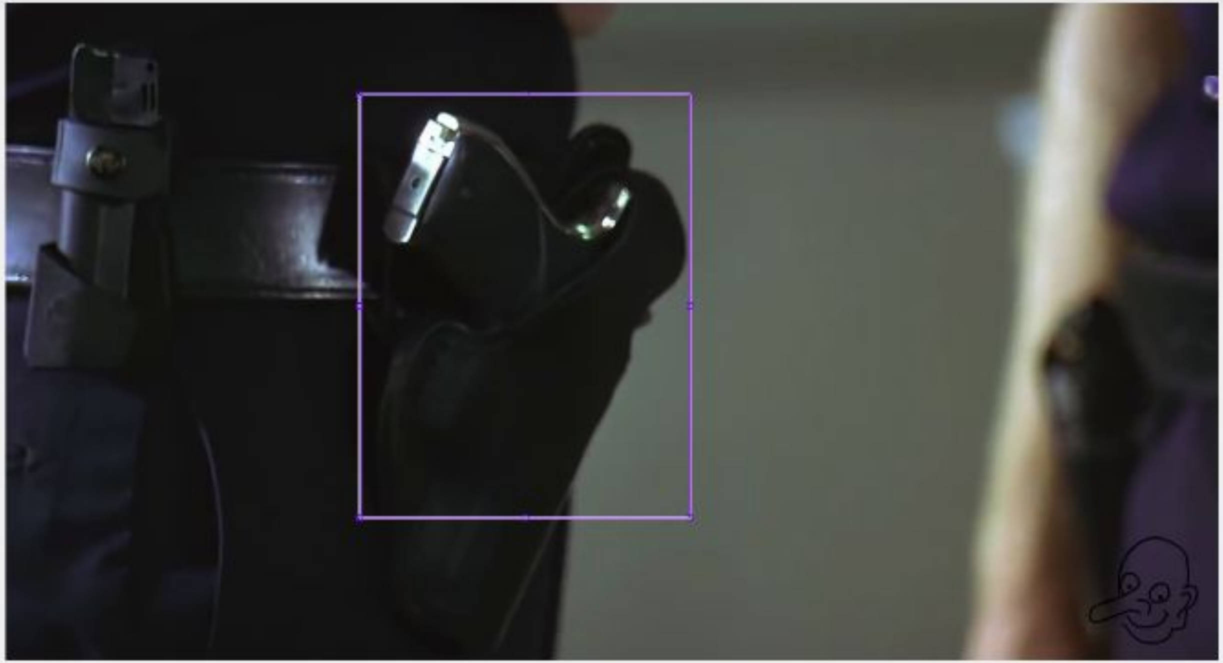
All the frames corresponding to those clusters will be the result for the retrieval step and would be ranked later based on how close they are to the search query

Procedure

- We had earlier created an inverted index list which stores all possible frames in which a particular cluster is found.
- 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.

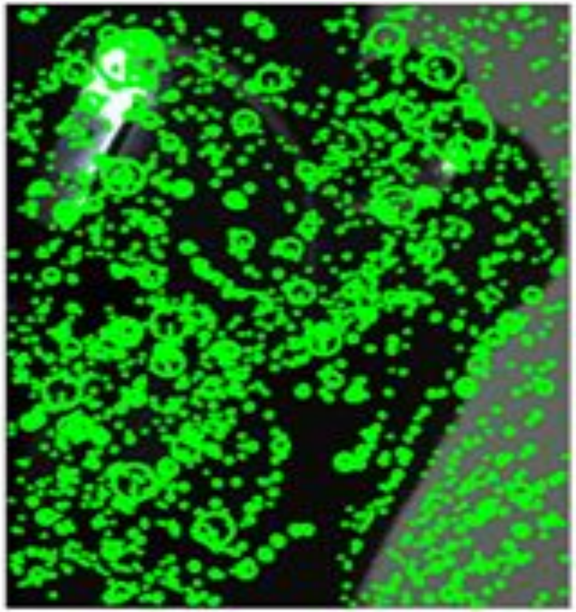
Object Retrieval (1)

Input Image

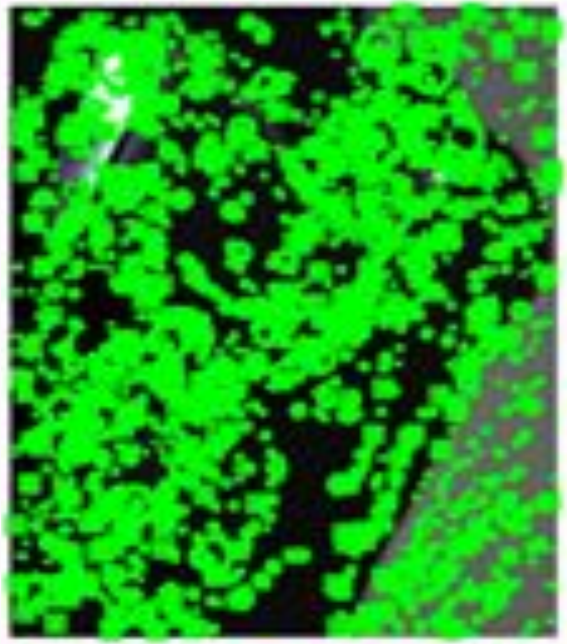


Object Retrieval (2)

Shape
Adapted
Regions

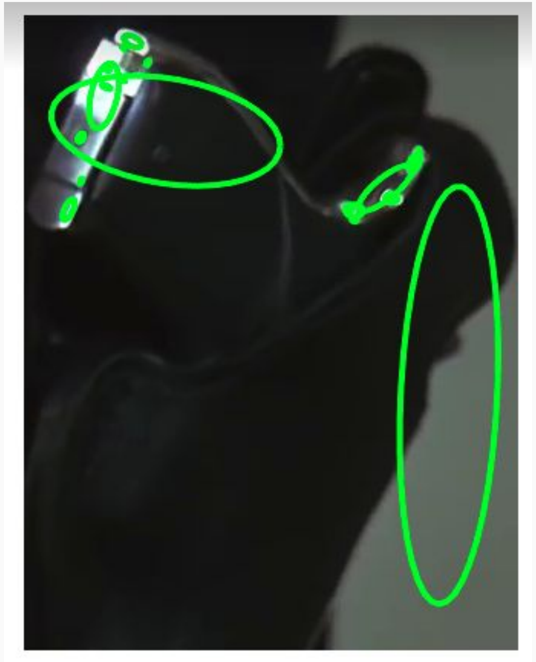


Shape
Adapted
Regions
with SIFT

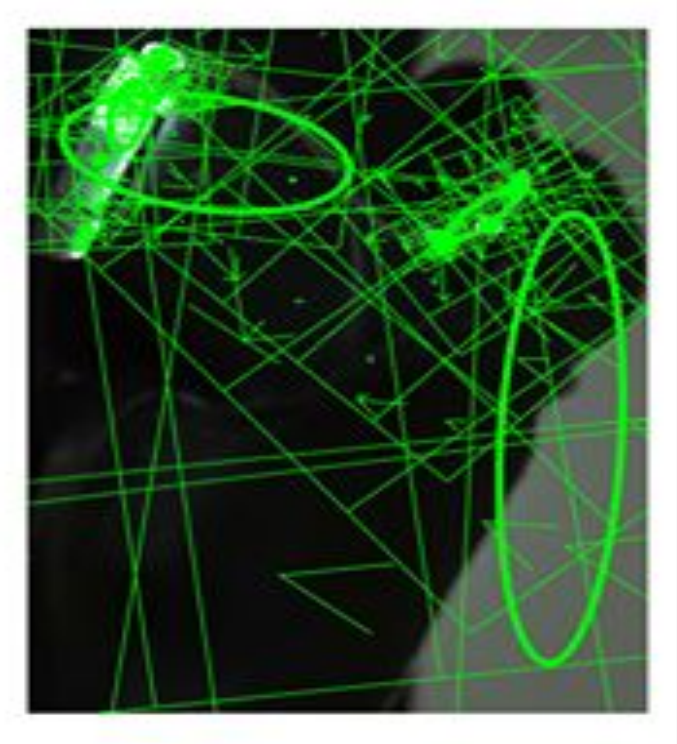


Object Retrieval (3)

Maximally Stable Regions



Maximally Stable Regions with SIFT

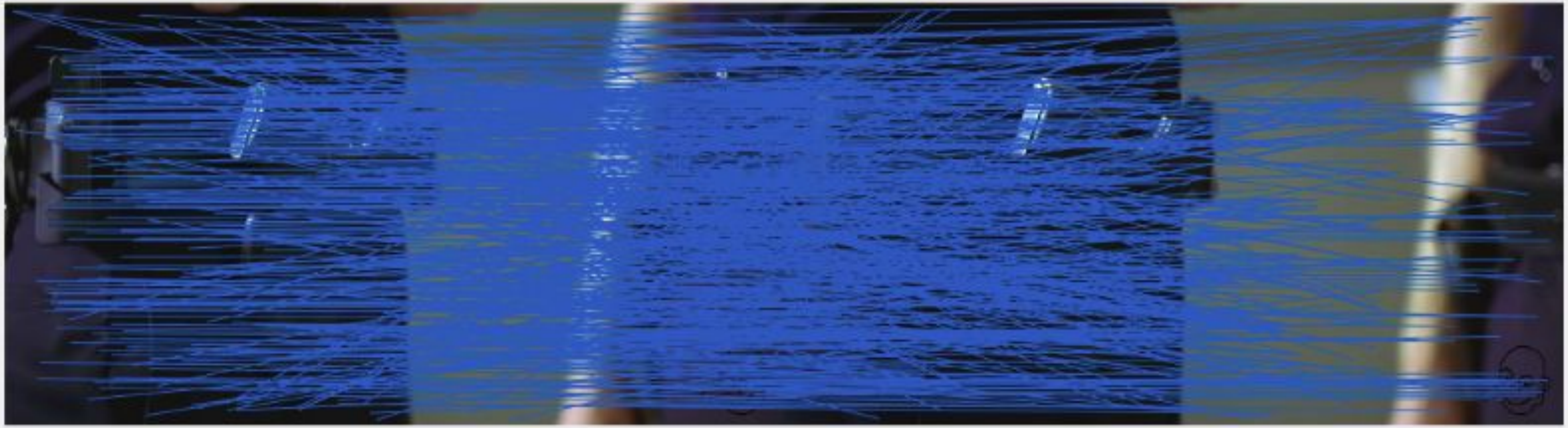


Object Retrieval (4)

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 display the top 5 retrieved frames.

Object Retrieval (5)



Tf-idf along with clustering allows us to find frames with a dense feature correspondence

RESULTS

We test our method with different types of region descriptors and objects in frames.

Results - Using different Region descriptors

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)

Results - Input Image

The object to be searched is the tie.

Results - Using SIFT





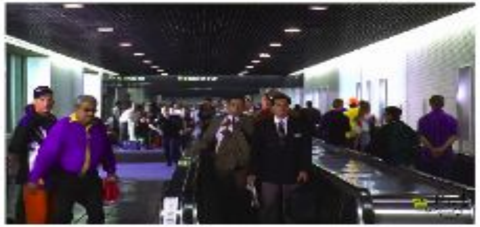

Results - Using Shape Adapted + SIFT descriptors

Shape Adapted regions detect corners and edge points across different viewpoints

Results - Using Maximally Stable + SIFT descriptors

Maximally stable regions detect blobs of the same color.

Results - Using Shape Adapted + Maximally Stable + SIFT descriptors






Blobs and corners are now detected.

Observations

We noticed that **SA+ MS+ SIFT descriptors** gives us the best results and we have used this in further examples.

This occurs because:

- SA descriptors take into account the center points whereas
- MS descriptors are centered around blobs and high contrast features.

Results - Input Image 2

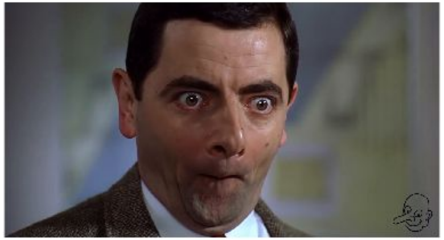
The object to be searched is the no-smoking sign.

Results - Using Shape Adapted + Maximally Stable + SIFT

Results - Input Image 3

The object to be searched is the window in the airplane.

Results - Using Shape Adapted + Maximally Stable + SIFT



EVALUATION OF SCENE MATCHING USING RANK

Here the objective is to match the scene with a closed world of shots.

Evaluation using Rank

For the retrieval tests,

- The entire frame is used as a query region
- It is measured over all frames, using each as a query region each time
- The ground truth is established manually
- The correct retrieval consists of all the other frames with the same location.
- Average normalized rank of relevant images is calculated by:

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

Here,

N -> size of the image set

N_{rel} -> no. of relevant images for a particular query

R_i -> rank of ith relevant image

Rank evaluated for tf-idf weighting scheme

We have computed the following metrics by using the input image of a gun in slide 22.

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

Limitations

- 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.

Blurred Input object case

The input object (ticket) is blurred.

Result obtained for blurred object

We observe that the images returned do not contain the ticket, and in fact most of them contain the no-smoking sign, probably confused for the ticket.

Future work

- Defining the object of interest over more than one frame
- Expanding over more than one video
- Using more text retrieval ideas such as automatic clustering to see what the principal images in the video are.

Thank you