Video Google: Text Retrieval Approach to Object Matching in Videos

Authors: Josef Sivic and Andrew Zisserman

ICCV 2003

Presented by: Indriyati Atmosukarto

Motivation

- Retrieve key frames and shots of video containing particular object with ease, speed and accuracy with which Google retrieves web pages containing particular words
- Investigate whether text retrieval approach is applicable to object recognition
- Visual analogy of word: vector quantizing descriptor vectors

Benefits

- Matches are pre-computed so at run time frames and shots containing particular object can be retrieved with no delay
- Any object (or conjunction of objects) occurring in video can be retrieved even though there was no explicit interest in object when descriptors were built

Text Retrieval Approach

- Documents are parsed into words
- Words represented by stems
- Stop list to reject common words
- Remaining words assigned unique identifier
- Document represented by vector of weighted frequency of words
- Vectors organized in inverted files
- Retrieval returns documents with closest (angle) vector to query

Viewpoint invariant description

- Two types of viewpoint covariant regions computed for each frame
  - Shape Adapted (SA)
  - Maximally Stable (MS)
- Detect different image areas
- Provide complimentary representations of frame
- Computed at twice originally detected region size to be more discriminating

Shape Adapted region

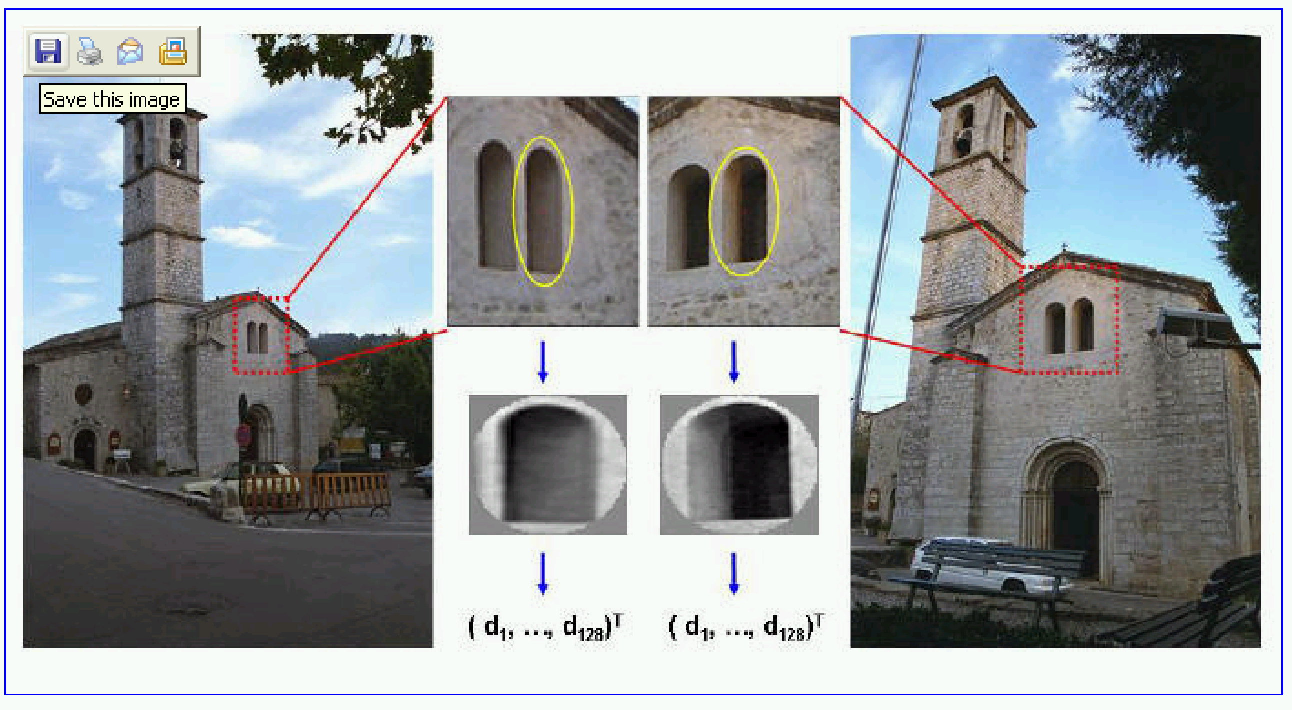
- Elliptical shape adaptation about interest point
- Iteratively determine ellipse center, scale and shape
- Scale determined by local extremum (across scale) of Laplacian
- Shape determined by maximizing intensity gradient isotropy over elliptical region
- Centered on corner like features

Maximally Stable region

- Use intensity watershed image segmentation
- Select areas that are approximately stationary as intensity threshold is varied
- Correspond to blobs of high contrast with respect to surroundings

Feature Descriptor

- Each elliptical affine invariant region represented by 128 dimensional vector using SIFT descriptor



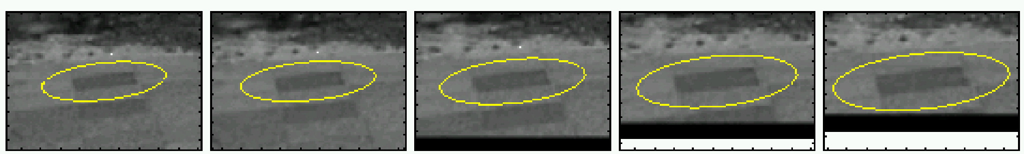
Noise Removal

- Information aggregated over sequence of frames
- Regions detected in each frame tracked using simple constant velocity dynamical model and correlation
- Region not surviving more than 3 frames are rejected
- Estimate descriptor for region computed by averaging descriptors throughout track

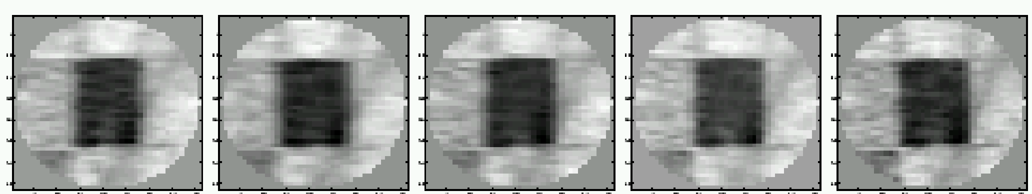
Noise Removal

- **Tracking region over 70 frames**

First (left) and last (right) frame of the track.



Close-up of the 1st, 20th, 40th, 55th, 70th frame.



Visual Vocabulary

- Goal: vector quantize descriptors into clusters (visual words)
- When new frame observed, descriptor of new frame assigned nearest cluster, generating matches for all frames

Visual Vocabulary

- Implementation: K-Means clustering
- Regions tracked through contiguous frames
- Mean vector descriptor x_i computed for each i regions
- Subset of 48 shots selected
- Distance function: Mahalanobis
- 6000 SA clusters and 10000 MS clusters

Visual Vocabulary

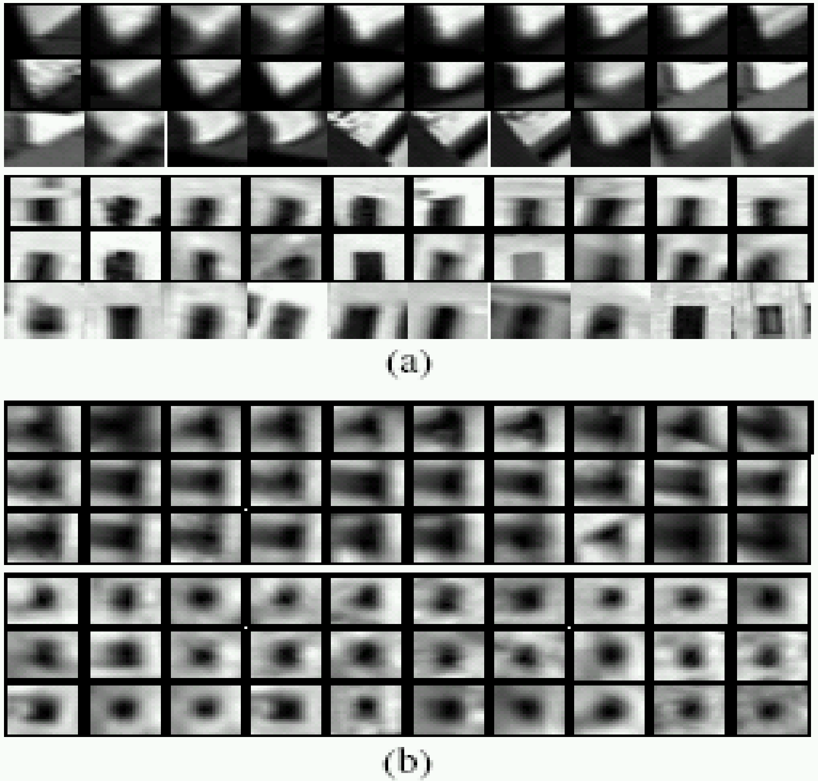


Figure 2: Samples from the clusters corresponding to a single visual word. (a) Two examples of clusters of Shape Adapted regions. (b) Two examples of clusters of Maximally Stable regions.

Visual Indexing

- Apply weighting to vector components
- Weighting: term frequency-inverse document frequency (tf-idf)
- Vocabulary k words, each doc represented by k-vector $V_d = (t_1,\ldots,t_i,\ldots,t_k)^T$ where

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

n_{id} = # of occurences of word i in doc d

n_d = total # of words in doc d

n_i=# of occurences of word I in db

N = # of doc in db

Experiments - Setup

- Goal: match scene locations within closed world of shots
- Data:164 frames from 48 shots taken at 19 different 3D locations; 4-9 frames from each location

Experiments - Retrieval

- Entire frame is query
- Each of 164 frames as query region in turn
- Correct retrieval: other frames which show same location
- Retrieval performance: average normalized rank of relevant images

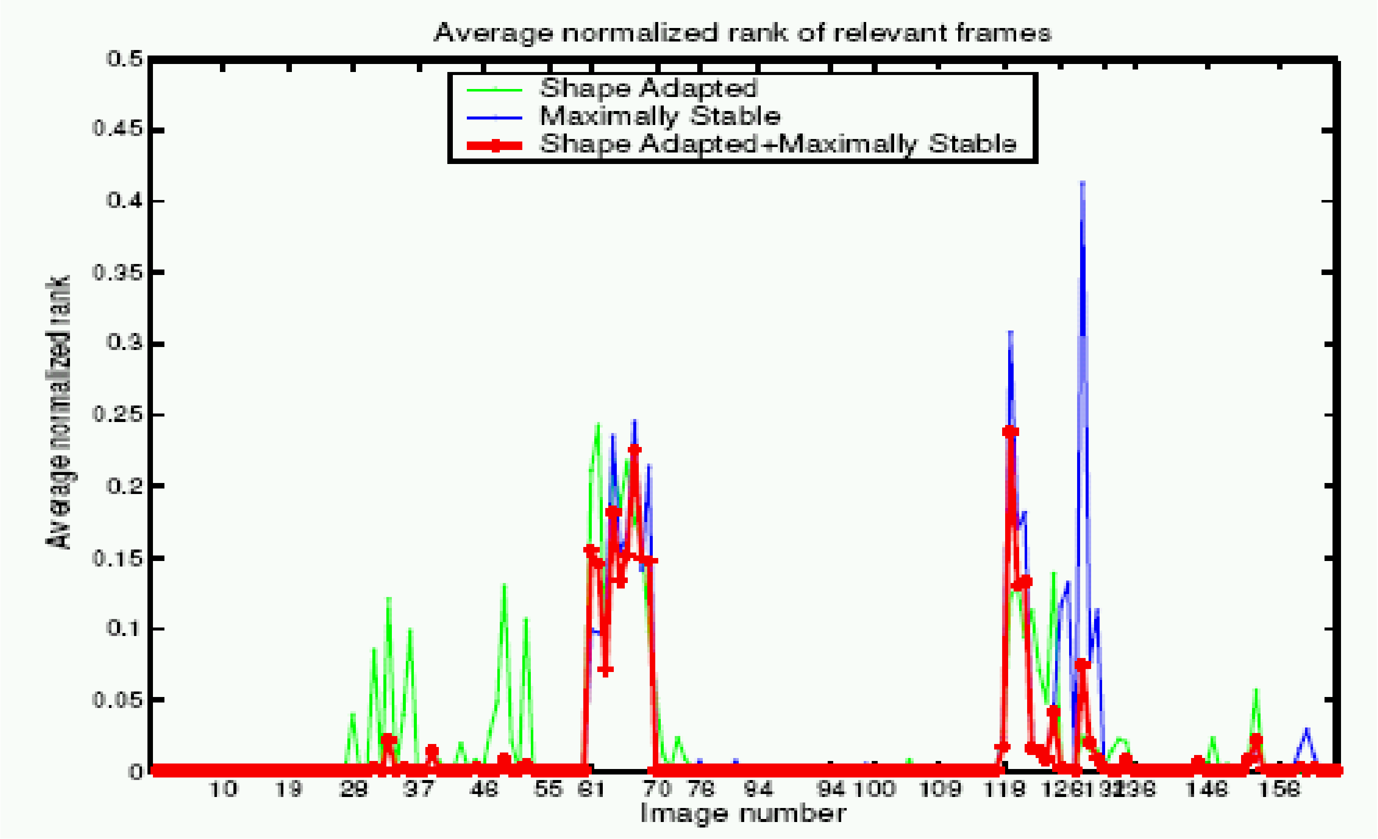
$$\widetilde{Rank} = \frac{1}{NN_{rel}}\left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2}\right)$$

N_{rel} = # of relevant images for query image

N = size of image set

R_i = rank of ith relevant image

Experiment - Results

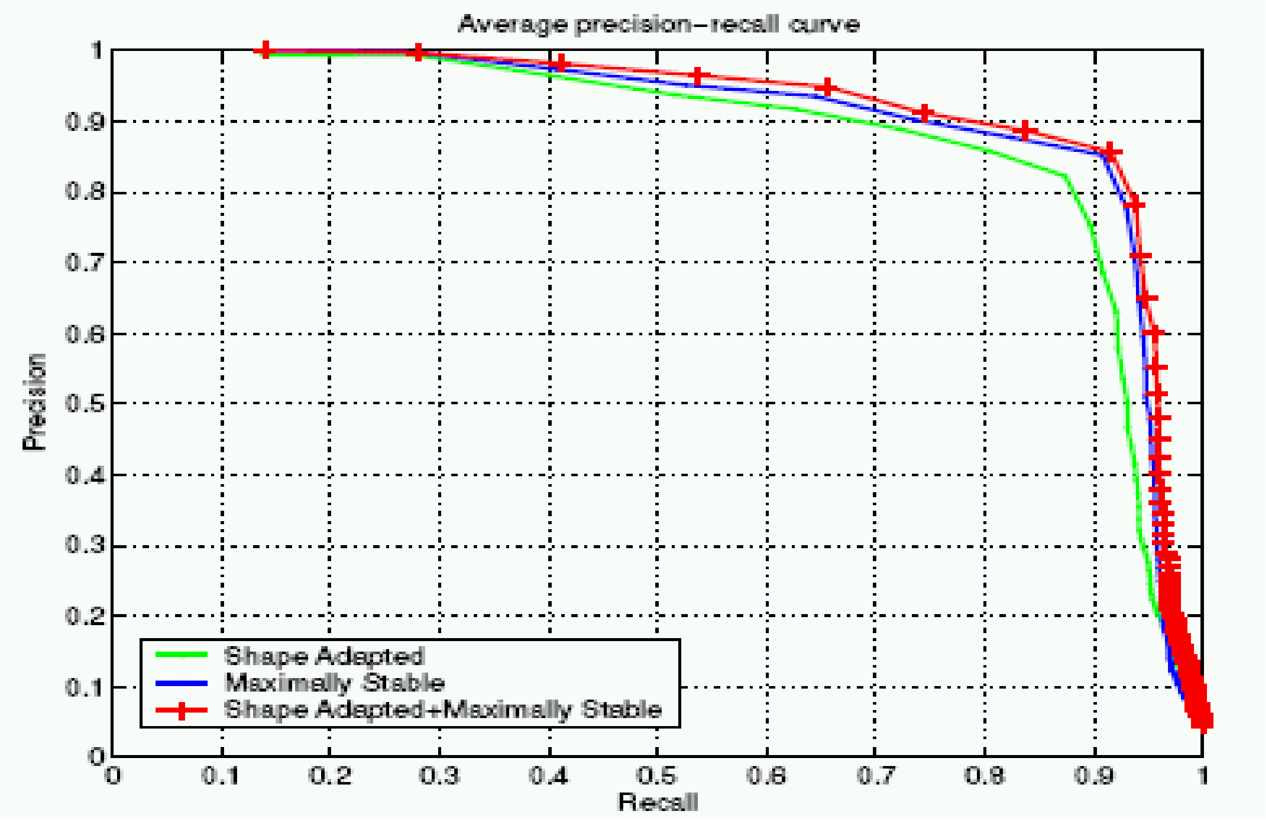


Experiments - Results

	binary	*tf*	*tf-idf*
SA	0.0265	0.0275	0.0209
MS	0.0237	0.0208	0.0196
SA+MS	0.0165	0.0153	0.0132

Table 1: The mean of the $\widetilde{Rank}$ measure computed from all 164 images of the ground truth set for different term weighting methods.

Experiments - Results



Precision = # relevant images/total # of frames retrieved

Recall = # correctly retrieved frames/ # relevant frames

Stop List

- Top 5% and bottom 10% of frequent words are stopped

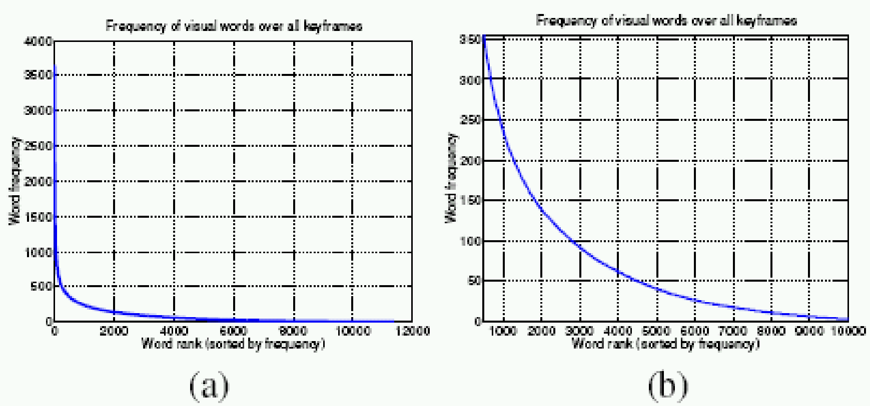


Figure 5: Frequency of MS visual words among all 3768 keyframes of Run Lola Run (a) before, and (b) after, application of a stoplist.

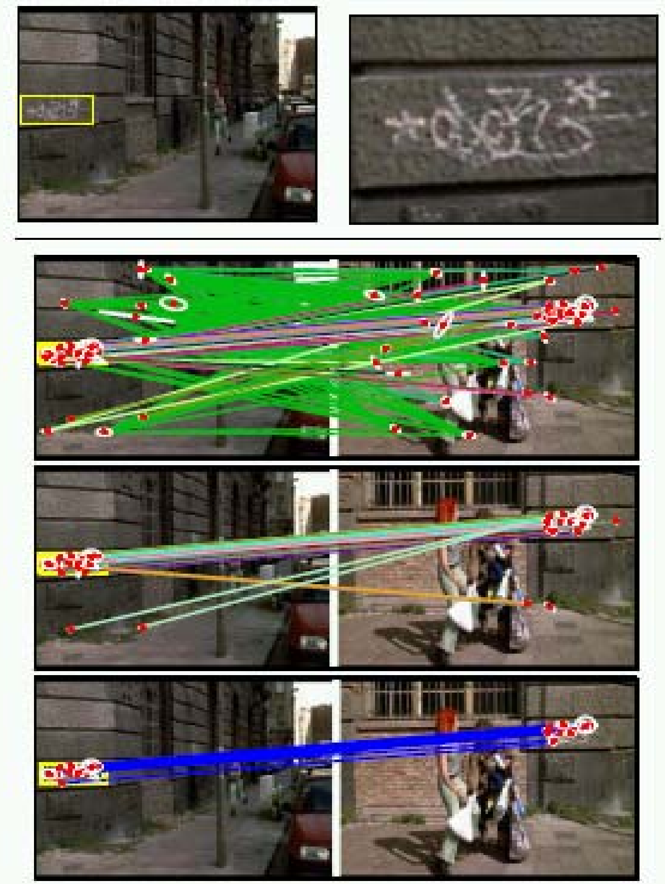


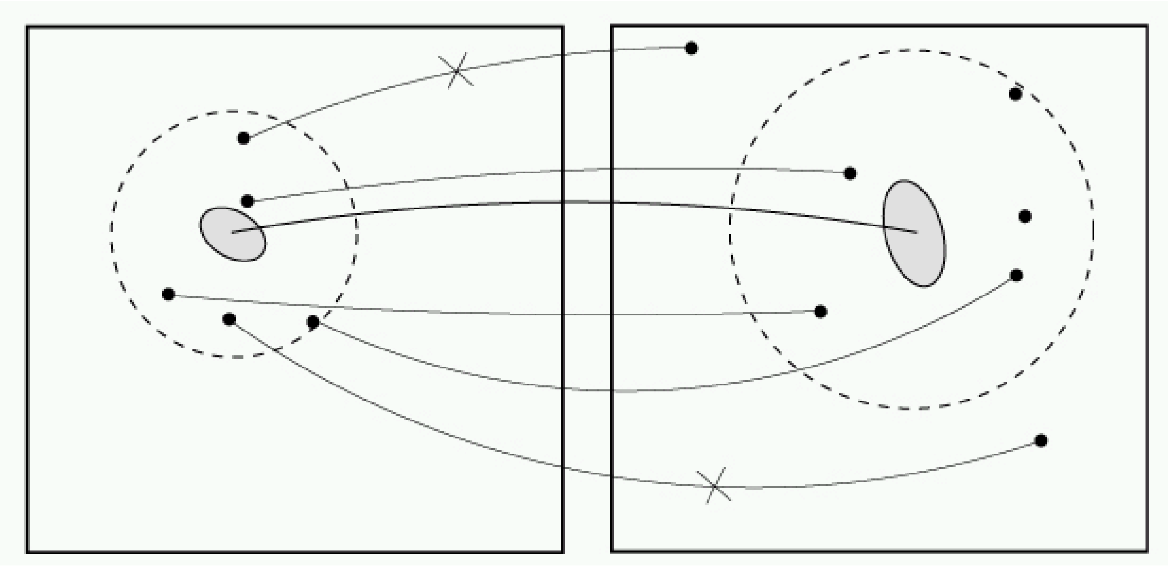
Figure 6: Matching stages. Top row: (left) Query region and (right) its close-up. Second row: Original word matches. Third row: matches after using stop-list, Last row: Final set of matches after filtering on spatial consistency.

Spatial Consistency

- Matched region in retrieved frames have similar spatial arrangement to outlined region in query
- Retrieve frames using weighted frequency vector and re-rank based on spatial consistency

Spatial Consistency

- Search area of 15 nearest neighbors of each match cast a vote for the frame
- Matches with no support are rejected
- Total number of votes determine rank

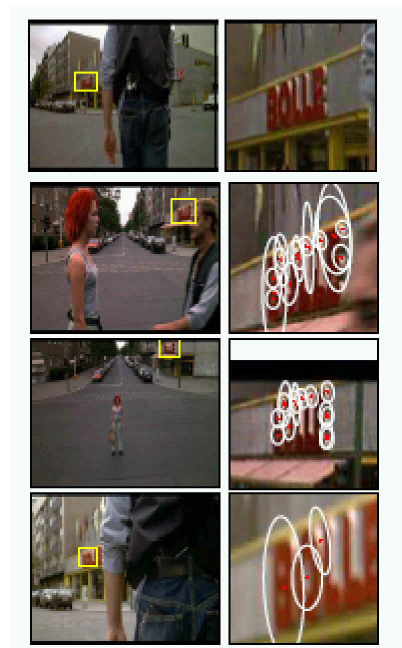


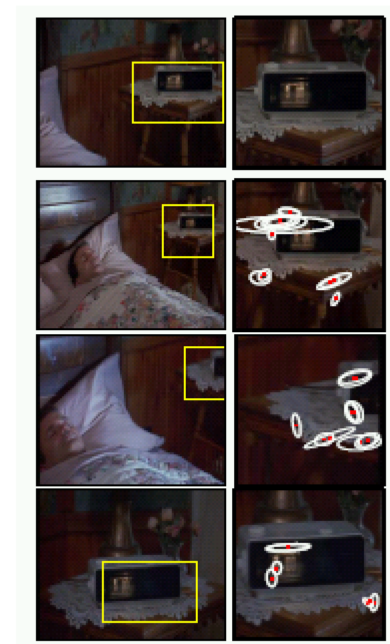
circular areas are defined by the fifth nearest neighbour and the number of votes cast by the match is three.

Inverted File

- Entry for each visual word
- Store all matches : occurences of same word in all frames

More Results





Future Works

- Lack of visual descriptors for some scene types
- Define object of interest over more than single frame
- Learning visual vocabularies for different scene types
- Latent semantic indexing for content
- Automatic clustering to find principal objects throughout movie

Demo

- http://www.robots.ox.ac.uk/~vgg/research/vgoogle/how/method/method_a.html
- http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html