Video Google: A Text Retrieval Approach to Object Matching in Videos,
J. Sivic and A. Zisserman
(ICCV 2003)

Object Level Grouping for Video Shots,
J. Sivic, F. Schaffalitzky, and A. Zisserman
(ECCV 2004)

Robin Hewitt

Video Google

Goal: fast, accurate googling on video files (movies).

Uses document-retrieval methods.

- A video frame is analogous to a document.
- A visual interest point is analogous to a word.
- Each movie is treated as a separate database.

Video data is analyzed and indexed once.

Arbitrary visual-content queries execute very quickly.

Video Google – indexing

Document Indexing	Video Indexing
Document → Words	Frame → Salient-point feature vectors (~1000/frame)
Word → Word Root (walking → walk)	Individual feature vectors → Cluster-centroid vectors
Update inverted file	Create inverted file
Optional: stop list identifies very common words (“a”, “the”, “and”).	Optional: stop list identifies very common and very rare feature vectors.

Inverted File Format – text

Word Lookup Table

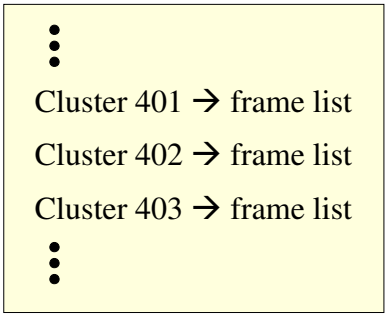
⋮
great → document list
grebe → document list
greed → document list
greek → document list
green → document list
⋮

document list – each entry contains
- document ID
- word-position list

(Yes, this is a lot of data.)

Inverted File Format – video

Feature Lookup Table



frame list – each entry contains
- frame number
- feature-location list

Feature Types

1. Corners: affine invariant interest points.

2. Blobs: Maximally Stable Extremal Regions (MSER).

Affine Invariant Interest Point Detector

1. Detect "corners" with Harris/Förstner method:

$$\mu = \begin{bmatrix} I_x^{\,2} & I_x I_y \\ I_x I_y & I_y^{\,2} \end{bmatrix} \qquad \text{Resp} = \text{Det}(\mu) - \alpha \text{Tr}^2(\mu)$$

2. Select scale – maximum of (scale-normalized) Laplacian:

$$\mathbf{L} = \left| \sigma^2 \left(I_{xx}(x, y, \sigma) + I_{yy}(x, y, \sigma) \right) \right|$$

(In practice, use difference of Gaussians at σ and kσ.)

Affine Invariant Interest Point Detector

Isotropic corners have no privileged axis.

Circle is a good bounding shape.

Harris corner detector handles these well.

Isotropic

Ellipse is a better bounding shape for anisotropic corners.

Scale has two dimensions.

Orientation axis is the principle eigenvector.

Anisotropic

Affine Invariant Interest Point Detector

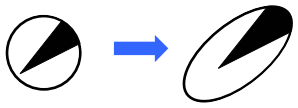
3. Iterative procedure estimates anisotropy, orientation, and location.

- Initialize bounding ellipse as a circle. U is the local window, μ is the second-moments matrix.

- Iterate until convergence (no further change):

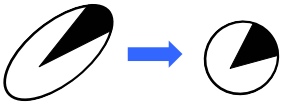
$$U^k = \left(\mu^{(k-1)}\right)^{\frac{-1}{2}} \ldots \left(\mu^{(1)}\right)^{\frac{-1}{2}} U^{(0)}$$

- Each iteration rescales U by the eigenvalues of μ and orients it to the eigenvectors. μ is recomputed each iteration, using the new window, U.



Affine Invariant Interest Point Detector

4. The interest region, U, is warped into a circle to create the affine-invariant "pre-image".

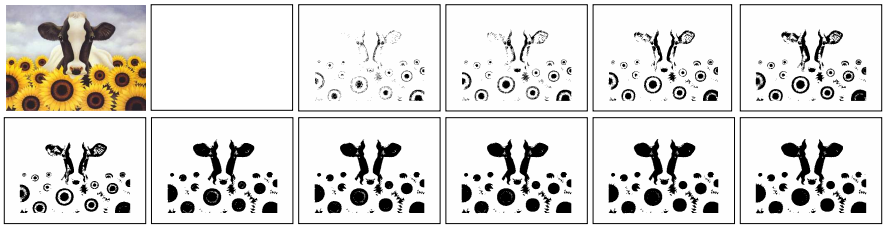


5. The normalized interest point is represented by gradient histograms from 16 subwindows (SIFT).

Maximally Stable Extremal Regions - MSER

Watershed method for blob detection:

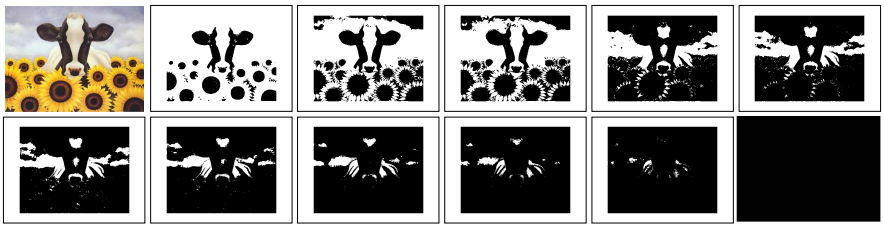
- Apply a series of thresholds – one for each grayscale level.
- Threshold the image at each level to create a series of black and white images.
- One extreme will be all white, the other all black. In between, blobs grow and merge.



Maximally Stable Extremal Regions - MSER

Blob that remains stable over a large threshold range are MSERs.

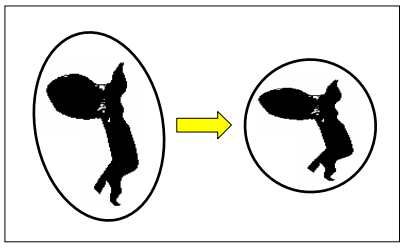
Criterion: dA/dt, where A = area, and t = threshold.

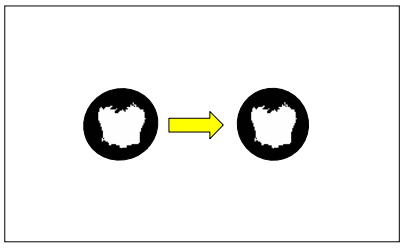


Maximally Stable Extremal Regions - MSER

MSERs are located with an ellipse.

These are also warped into circles and represented with SIFT descriptors.

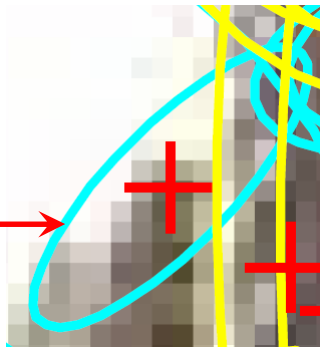




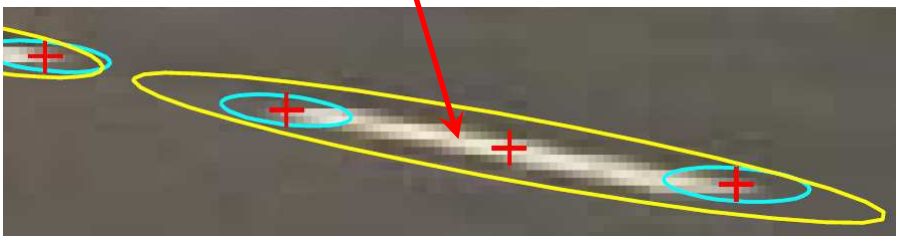
Features

Corners – cyan
Blobs - yellow

Sawtooth skyline



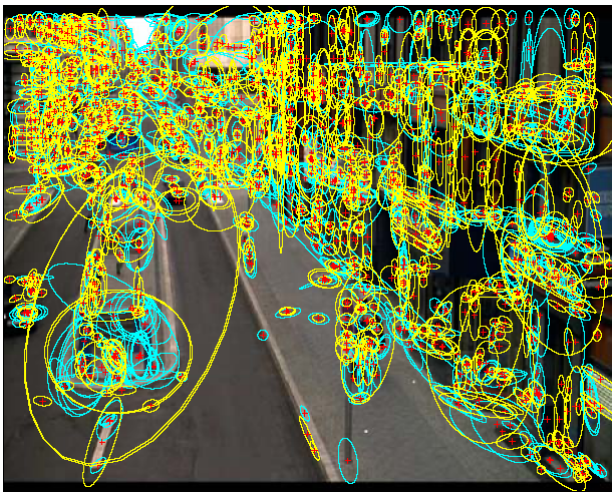
Dashed line on street



Features

Example video frame

Features



~1600 features per frame

Cyan - corners
Yellow - blobs

From Features to "Words"

Track each feature over several frames

- Skip features that don't persist for 3 or more frames
- Average features that do

Remove the 10% with largest covariance

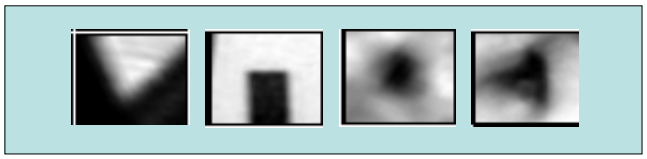
Feature count is now ~1000/frame

From Features to "Words"

Cluster the averaged feature vectors (K-Means)

- Used 164 frames
- Hand selected to include 19 locations, 4-9 frames each
- Wide variation in viewpoint
- Clustered each feature type separately

Cluster centroids become "words" in the inverted file



Representing Frames as Documents

Create frame vectors

k-dimensional vectors, **V**, where k = total number of "words"

Each component of **V** is a weighted word-occurance count:

$$\mathbf{V}_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

where

n_{id} = occurances of word in frame

n_d = number of words in frame

N = number of frames in movie

n_i = occurances of word in movie

Video Google – search

Specify query with a bounding box

Features in query → words (cluster centroids)

Each word → frame list

Frame → weighted word vector

Angle between frame vector and query vector gives relevance

Finally, filter and re-rank frames based on spatial consistency

Object Level Grouping

Goal: detect and recognize objects in all frames of a video.

Extends Video Google work.

Uses motion and continuity between frames.

Infers object's significance from its occurance frequency.

Leverages artistic effects – tracking, selective focus, etc.

Uses same features as in Video Google.

Object Level Grouping

Algorithm Overview

1. Detect features in each frame.
2. Link features between consecutive frame pairs.
3. Short-range track repair – interpolate tracks for missing features through 2-5 frames.

Object Level Grouping

Algorithm Overview, cont.

4. Cluster feature tracks into oversegmented, but "safe" groupings.

5. Merge the track clusters with consistent 3D motion to extract objects.

6. Long-range track repair using wide-baseline stereo.

Object Level Grouping

2. Linking features between frame pairs:

Same features as in Video Google.

Match between each frame pair, within 50 pixels.

Validate by cross-correlation.

Delete ambiguities – anything that matches more than once between frames.

Eliminate additional outliers by loosely enforcing epipolar geometry (RANSAC with a 3 pixel inlier threshold).

Object Level Grouping

3. Short-range track repair:

Estimate motion from momentum of previous n frames (n ≈ 5).

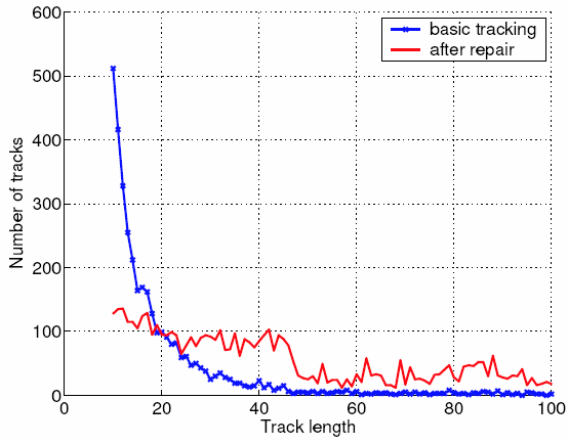
In frame with missing feature, look for a match in the predicted area.

If not found, keep looking for a match in few (2-5) more frames.

If a match is found, extend the track.

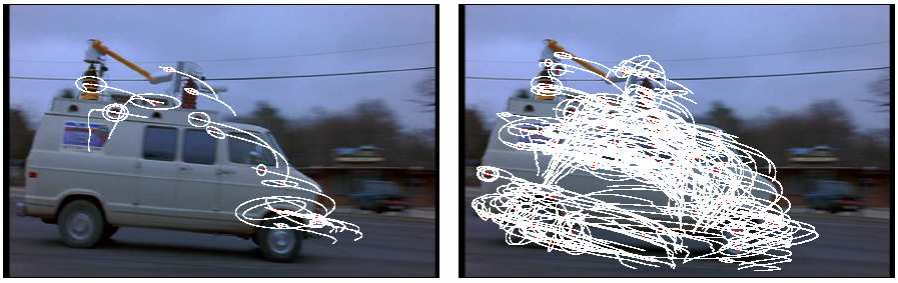
Object Level Grouping

Effect of short-range track repair



Object Level Grouping

Effect of short-range track repair



Left: feature matching between frame pairs.
Right: tracks after short-range track repair.

Object Level Grouping

4. Cluster tracks into safe (i.e., conservative) groupings:

Group tracks that match the same projective homography in all three frame pairs of three consecutive frames.

Move ahead one frame and repeat for next (overlapping) triplet.

This over-segments the tracks into small, short motion groups, each centered on one frame.

Object Level Grouping

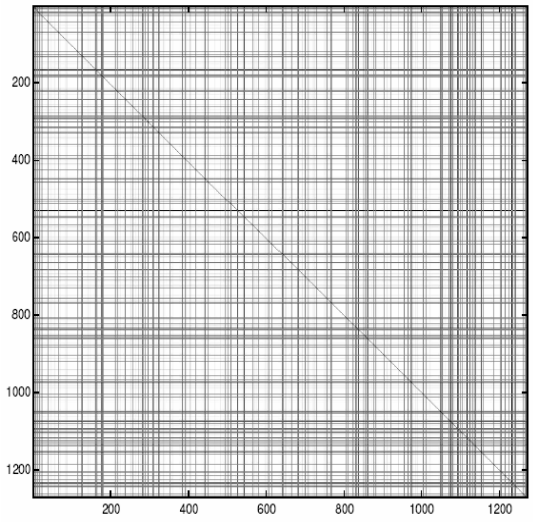
4. Cluster tracks into safe groupings, cont:

Count co-occurances of each track pair over n (~10) consecutive frames. Co-occurance matrix, W, accumulates votes.

W is similar to a correlation matrix. w_{ij} accumulates a vote each time tracks i and j are in the same short-term motion group.

Object Level Grouping

Short-motion co-occurance matrix



Object Level Grouping

4. Cluster tracks into safe groupings, cont:

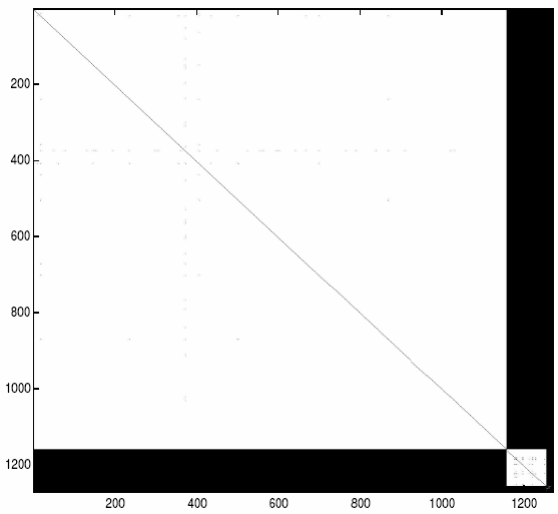
Threshold W, s.t. threshold > n/2 to ensure that no track is assigned to more than one cluster.

Find connected components of the graph corresponding to thresholded W.

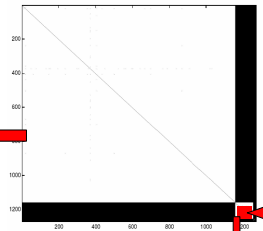
This is equivalent re-ordering the rows of W s.t. the white regions form bands that continue to the diagonal.

Object Level Grouping

Matrix W after re-ordering rows



Object Level Grouping



outliers

Two clusters of “correlated” motion

Object Level Grouping

5. Merge clusters to extract objects:

For each pair of clusters

Fit a full, 3D affine transformation over >20 frames using RANSAC on 4 tracks

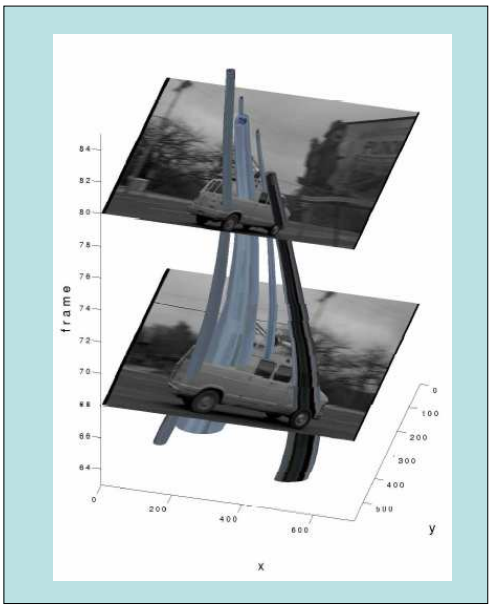
Validate by projecting the remaining tracks

If $\geq 90\%$ of tracks project consistently, merge clusters

All agglomerated groupings are presumed to be objects

Object Level Grouping

Trajectories of 5 feature ellipses, all in the same object



Object Level Grouping

Tracking the B&B owner

Each 3D projection is over any set of 20 or more frames. But the track may continue further. This allows flexbility to accommodate slowly deforming objects.

Object Level Grouping

6. Long-range track repair:

If an object's grouped tracks all disappear, that may be due to occlusion.

The "same" tracks should then reappear later on.

Track sets are matched with wide-baseline stereo. Only features within grouped tracks are matched.

Object Level Grouping

Long-range track repair