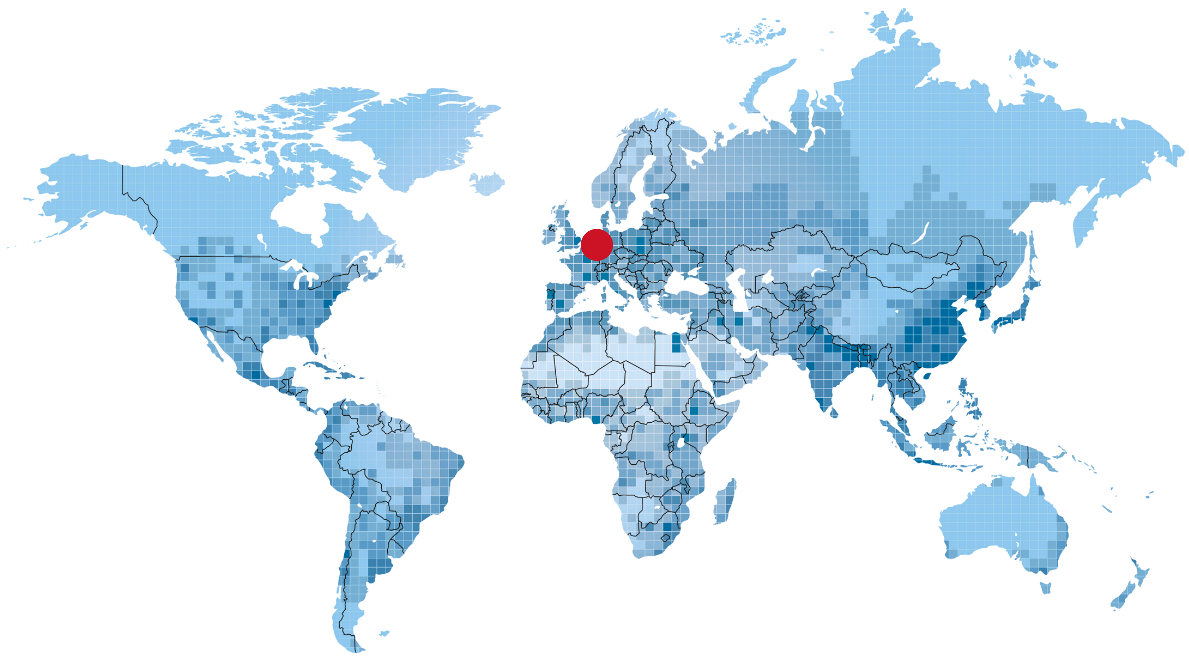
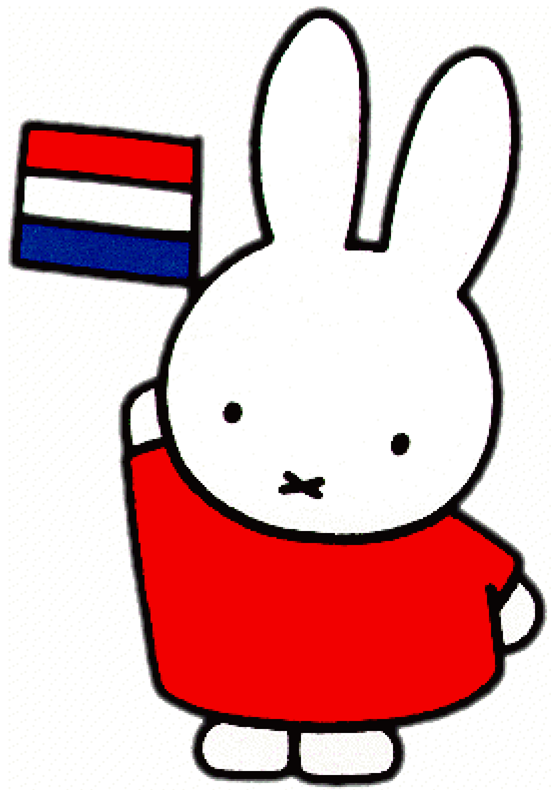
Exhaustive SURF, small bag-of-words and Eigenfaces

John Schavemaker, Wessel Kraaij, Pieter Eendebak, Mark van Staalduinen

TNO

- Netherlands research organization for Applied Scientific Research
- more than 4000 employees
- 1% in our Media & Network Services group
- our focus : **application** of science for (Dutch) industry





typical Dutch

rationale: fighting child abuse

child-abuse domain: finding objects, locations and persons (child victims) in confiscated video

rationale: using open-source components

- open computer-vision library (OpenCV)
- Lemur text retrieval
- FFmpeg
- openMP
- Intel Threading Building Blocks
- Google Tesseract OCR
- GNU tools

INSTANCE SEARCH

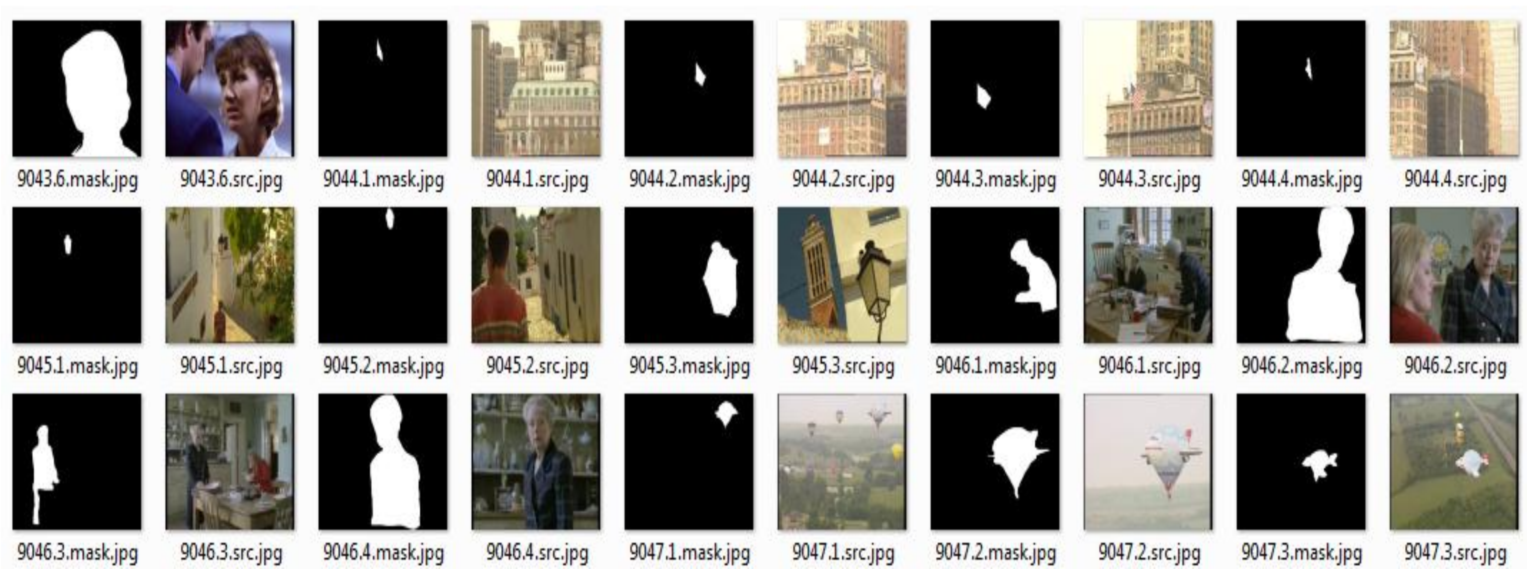
 

TNO resources

- late start at end of June due to budget allocation
- 10 weeks (excluding holidays) to deadline
- 1.5 person
- 2 PCs (i7 8 cores)
- TNO instance search system 2010


TRECVID instance search task 2011

- 250 hours of video of rushes from BBC archives
- 25 queries: objects, locations, and persons



some example queries

query images

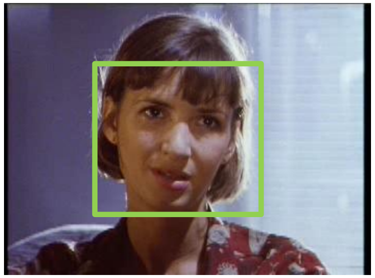
mask to indicate object of interest

multiple images from one query

TNO runs

1. SURFAC: exhaustive search in SURF points
2. BOWCOL: approach with bag-of-words, global color, face, and skin color detection
3. SURFEIG: approach with face recognition (Eigenface)

open-source components: Lemur, OpenCV, Eigenface, FFmpeg

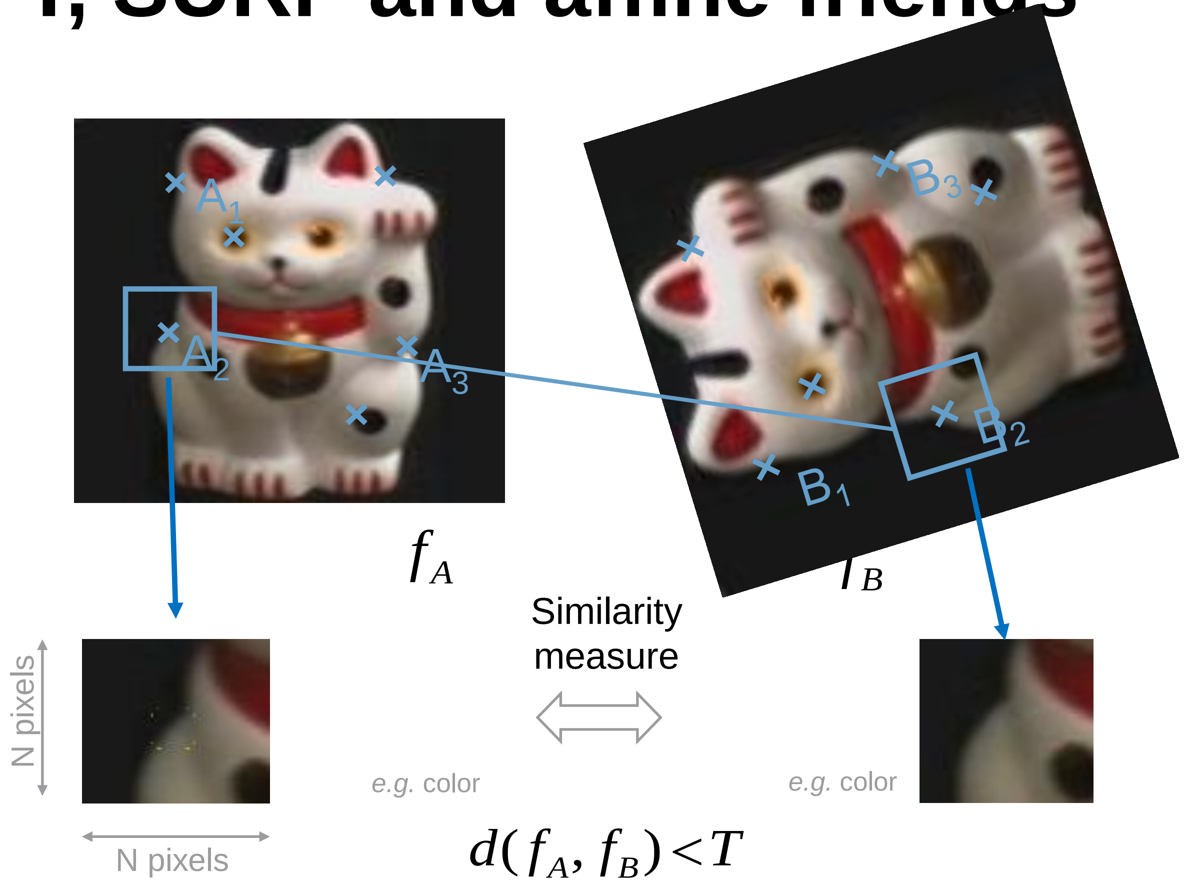


TNO runs in more detail

- all runs: video decoding using FFmpeg library, sampling every 25th frame (every second).
- SURFAC: standard SURF keypoint detection, exhaustive search.
- BOWCOL: standard SURF keypoint detection, bag-of-words using 256 prototypes from queries, indexing and querying using Lemur.
- SUREIG: Viola & Jones face detection & face retrieval with Eigenface descriptors.

EXHAUSTIVE SURF

SIFT, SURF and affine friends



invariants: scale, intensity, translation, rotation

pre-processing

SURF key points are detected in every decoded and stored video frame of the entire video data set.

- chosen implementation: cvExtractSurf from OpenCV [2,6] using default parameters;
- we have chosen SURF over SIFT because of processing time
- key point detection is sparse and based on the Hessian structure tensor, differing here from the classical dense-sampled approach using a pre-defined grid of points with fixed scales;
- SURF descriptors are computed (128 floating point numbers) and stored with key point information in XML.

Query by exhaustive search

- original approach of David Lowe: detecting key points, computing and matching local descriptors
- descriptor matching is done with an approximate nearest-neighbor implementation where only a part of all descriptors can be in memory.

Query by exhaustive search

Initialization:

- convert all database descriptors in XML to (large-sized) floating-point binary files

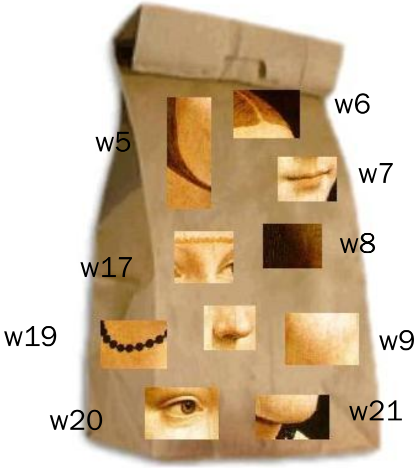
Query:

- detect keypoints and compute descriptors in all query images
- match query descriptors with database descriptors by loading binary descriptor files one-by-one and doing an approximate NN search
- aggregate results per video (sum matches)

SMALL BAG-OF-WORDS

bag-of-words

from picture to text



<DOC>
<DOCNO>1000_mpg_item3_sample2</DOCNO>
<TEXT>
w5 w6 w7 w8 w9 w17 w19 w20 w21 w22 w24 w25 w26 w27 w28 w29 w30 w34 w42 w51 w52 w59 w60 w62 w64 w66 w67 w68 w69 w71 w73 w76 w77 w78 w81 w84 w86 w89 w90 w96 w97 w113 w114 w117 w120 w130 w171 w177 w180 w181 w182 w189 w192 w196 w197 w203 w206 w207 w222 w224 w241 w249
face LowOrange MediumOrange HighOrange LowYellow MediumYellow LowBlack
</TEXT>
</DOC>

BOWCOL

- Clustering
- Quantization
- Creation of Lemur/Indri index
- Query


BOWCOL : CLUSTERING

- All SURF key point descriptors from the query image set are clustered to a pre-defined number (here 256 cluster prototypes)
- chosen implementation: flann::hierachicalClustering from FLANN[3] in OpenCV [2] using default parameters;
- the visual vocabulary cluster prototypes are written to XML.

BOWCOL : Quantization

Quantization of SURF descriptors

Nearest prototype, if more than 1 keypoint maps to prototype 1 add "w1" to text

Quantization of global color

- find black (low value) and white (high value, no saturation) pixels
- make global color HSV histogram (9 bins) of remaining pixels
- quantize bin (color) with scheme:
  - > 1% "Low", >10% "Medium", >25% "High"
  - for example "LowYellow"

Quantization of face detection: face detected add "face" to text

Quantization of skin detection: similar to color using global skin probality: add "LowSkin", "MediumSkin", "HighSkin" to text

BOWCOL : Quantization result

<DOC>

<DOCNO>1000_mpg_item3_sample2</DOCNO>

<TEXT>

w5 w6 w7 w8 w9 w17 w19 w20 w21 w22 w24 w25 w26 w27 w28 w29 w30 w34 w42 w51 w52 w59 w60 w62 w64 w66 w67 w68 w69 w71 w73 w76 w77 w78 w81 w84 w86 w89 w90 w96 w97 w113 w114 w117 w120 w130 w171 w177 w180 w181 w182 w189 w192 w196 w197 w203 w206 w207 w222 w224 w241 w249

face LowOrange MediumOrange HighOrange LowYellow MediumYellow LowBlack

</TEXT>

</DOC>

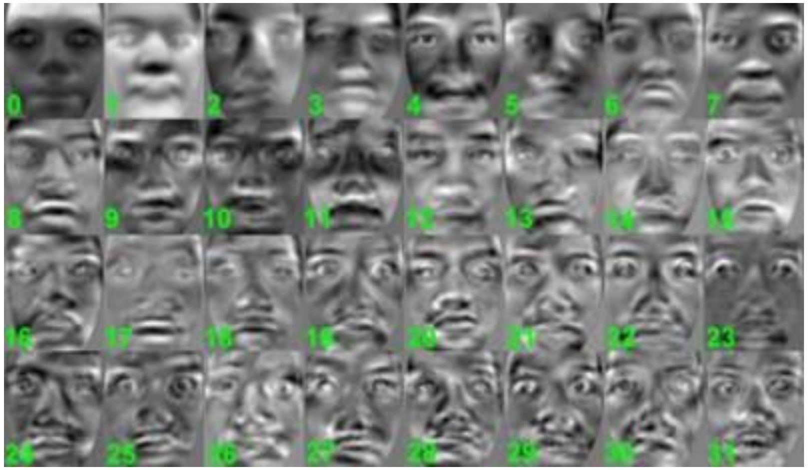
BOWCOL : Query

- detect and compute features like ingestion
- make the TRECTEXT query document
  - differentiate between type of query
  - person: face and skin features
  - object/location no face, less skin
- query Lemur repository with document
- aggregate results per video file (summation)

EIGENFACES

Eigenfaces

describe one face by using 100 other faces



From OpenCV web tutorial

Cognotics

Resources for Cognitive Robotics

Search Cognotics:

Google™ Custom Search

Search

Home > OpenCV Resources > *Seeing With OpenCV* > **Part 4**

Pages: 1 2 3 Next

Seeing With OpenCV, Part 4: Face Recognition With Eigenface

by Robin Hewitt

This article originally appeared in SERVO Magazine, April 2007. Reprinted by permission of T & L Publications, Inc.

Last month's article introduced Camshift, OpenCV's built-in face tracker. This month and next, this series concludes by showing you how to use OpenCV's implementation of eigenface for face recognition.

Face recognition is the process of putting a name to a face. Once you've detected a face, face recognition means figuring out whose face it is. You won't see security level recognition from eigenface. It works well enough, however, to make a fun enhancement to a hobbyist robotics project.

This month's article gives a detailed explanation of how eigenface works and the theory behind it. Next month's article will conclude this topic by taking you through the programming steps to implement eigenface.

What is Eigenface?

Eigenface is a simple face recognition algorithm that's easy to implement. It's the first face-recognition method that computer vision students learn, and it's a standard, workhorse method in the computer vision field. Turk and Pentland published the paper that describes their Eigenface method in 1991 (Reference 3, below). Citeseer lists 223 citations for this paper - an average of 16 citations per year since publication!

http://www.cognotics.com/opencv/servo_2007_series/part_4/index.html

Eigenfaces

Ingestion:

- detect faces in all video frames
- compute Eigenface descriptor and store it
- construct one binary file holding all face descriptors

Query:

- load face descriptor matrix
- detect face in query image, and compute descriptor
- do approximate nearest-neighbor search
- aggregate results: minimum distance per video

RESULTS

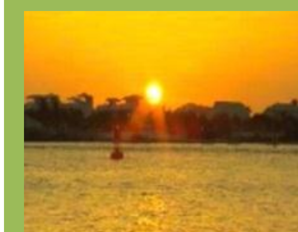
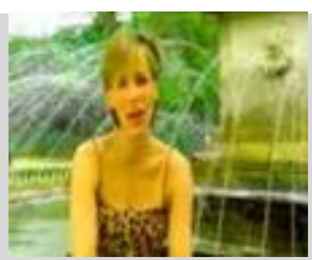
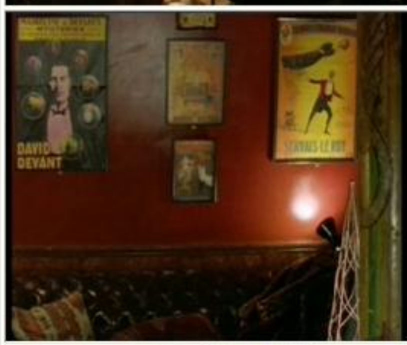
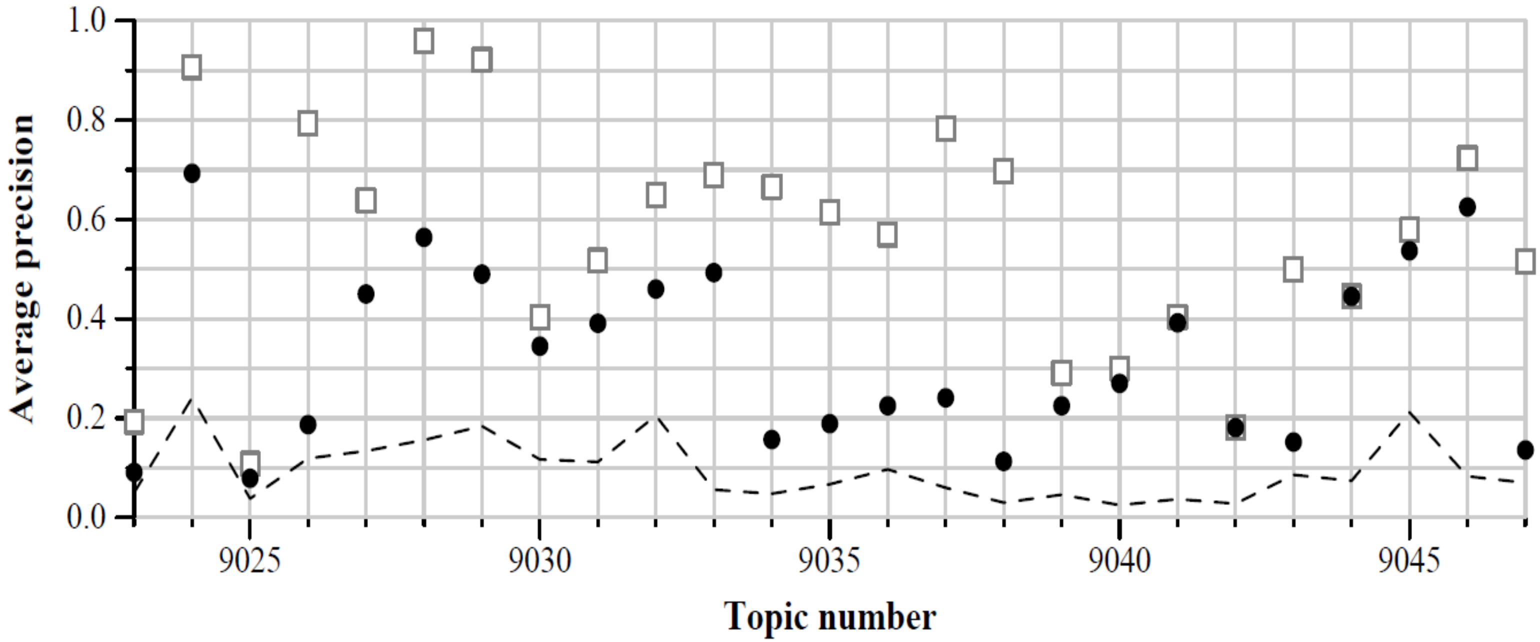
QUERY	RESULT	RESULT	RESULT
 LOCATION	 SURF punten	 SURF punten	 SURF punten
 OBJECT	 bag-of-words	 bag-of-words	 bag-of-words
 PERSON	 Eigenface	 Eigenface	 Eigenface

image match	match score	media link	sequence number	shot id
	53	E:/trecvid2011/m3/server/database/trecvid2011/5511_mpg/img_0019.jpg	1	5511
	50	E:/trecvid2011/m3/server/database/trecvid2011/10041_mpg/img_0000.jpg	2	10041
	38	E:/trecvid2011/m3/server/database/trecvid2011/10813_mpg/img_0017.jpg	3	10813
	23	E:/trecvid2011/m3/server/database/trecvid2011/3678_mpg/img_0017.jpg	4	3678

results SURF

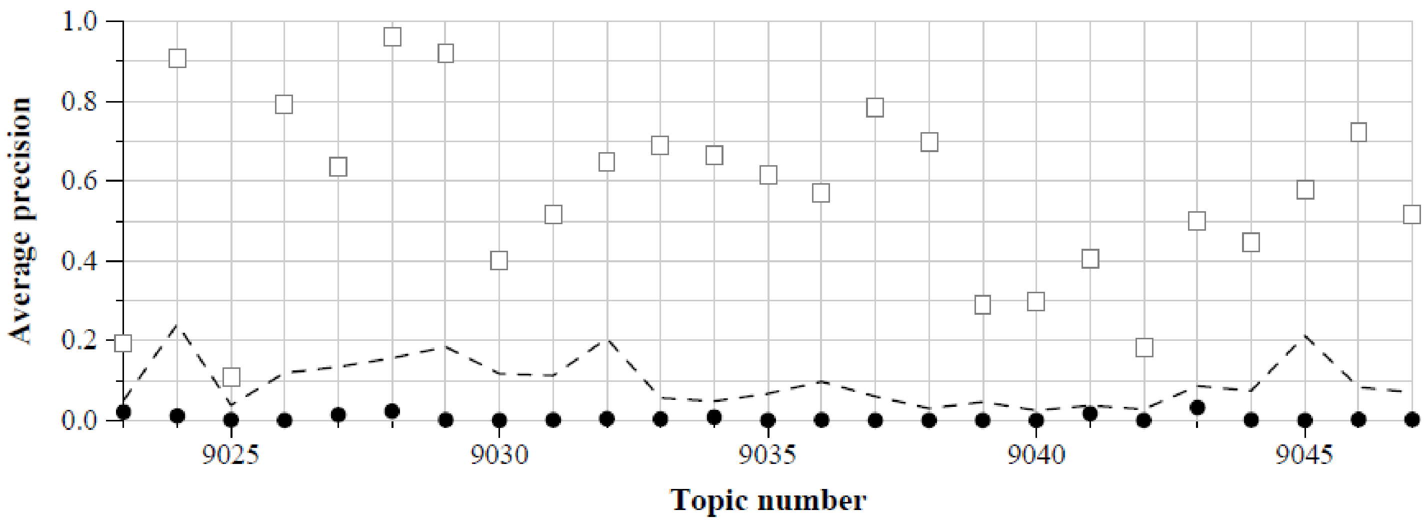


Run score (dot) versus median (---) versus best (box) by topic

John Schavemaker
TNO instance search submission

	0.00970379	E:/trecvid2011/in3/server/database/trecvid2011/18722_mpg/img_0029.jpg	15	18722
	0.00969963	E:/trecvid2011/in3/server/database/trecvid2011/16293_mpg/img_0021.jpg	16	16293
	0.00969786	E:/trecvid2011/in3/server/database/trecvid2011/6880_mpg/img_0029.jpg	17	6880
	0.00969786	E:/trecvid2011/in3/server/database/trecvid2011/866_mpg/img_0029.jpg	18	866
	0.00942212	E:/trecvid2011/in3/server/database/trecvid2011/510_mpg/img_0026.jpg	19	510

results BOWCOL

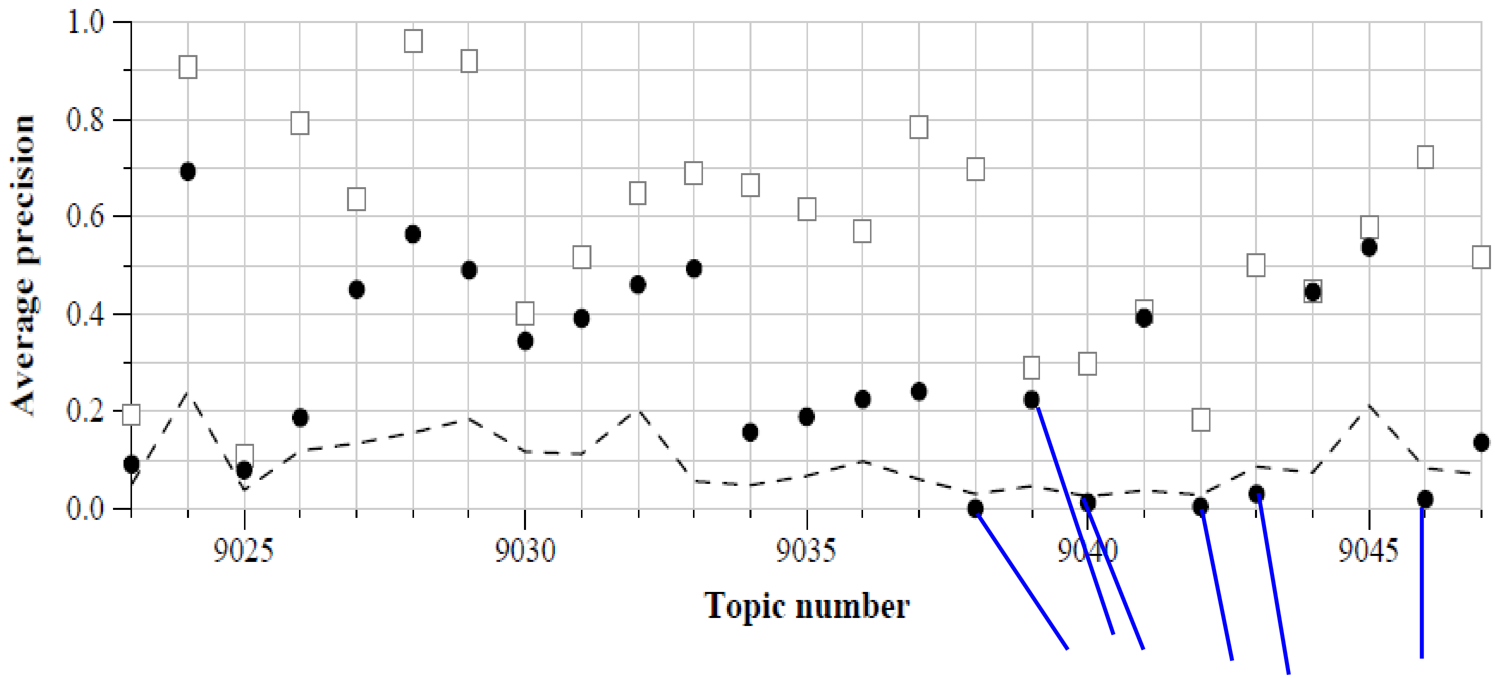


Run score (dot) versus median (---) versus best (box) by topic

John Schavemaker
TNO instance search submission

image match	match score	media link	sequence number	shot id
	0.940833	E:/trecvid2011/m3/server/database/trecvid2011/19565_mpg/img_0003.jpg	1	19565
	0.935111	E:/trecvid2011/m3/server/database/trecvid2011/5139_mpg/img_0001.jpg	2	5139
	0.921997	E:/trecvid2011/m3/server/database/trecvid2011/5666_mpg/img_0009.jpg	3	5666
	0.913447	E:/trecvid2011/m3/server/database/trecvid2011/4811_mpg/img_0002.jpg	4	4811

results Eigenfaces



Run score (dot) versus median (---) versus best (box) by topic


computational costs

METHOD	**Offline** (all video)	**Online** (1 query)
Exhaustive SURF	1 day	30 minutes
Small BOW	1 day	1-5 minutes
EigenFace	1 day	1 minute

conclusions

- (SURFAC2_1) Using a brute-force search over all SURF keypoint descriptors gives real instance search results: hits for every query. However, there is no generalization (for example, sunset) of objects and it takes relatively long (15-30 minutes on a single-core PC).
- (BOWCOL_2) Bag-of-words gives for some queries nice concept detection results (sunset) and is relatively fast.
- (SUREIG_3) Some interesting results for searching persons. With good frontal faces in the query set, interesting results are possible and quick to attain.

Thank you for your attention!

questions welcome

e-mail: john.schavemaker@tno.nl