Large Vocabulary Quantization for Instance Search at TRECVID 2011

Cai-Zhi Zhu, Duy-Dinh Le, Sebastien Poullot,Shin'ichi Satoh

National Institute of Informatics, Japan

December 6, 2011

Outline

- Motivation
- Related works
- Algorithm overview
- Results
- Demos
- Discussion and conclusion

- Motivation

Observations from INS 2010

- Almost all teams submitted ad-hoc systems.
  - Combined multiple features.
  - Separately treated different topics, especially face.
  - Elaborately fused multiple pipelines.
  - Even resorted to concept detectors.

✓ A simple while efficient algorithm could be very appealing.

- Instance search task is very difficult.
  - The best MAP is only 0.033@NII.

✓ A high return low risk research direction.

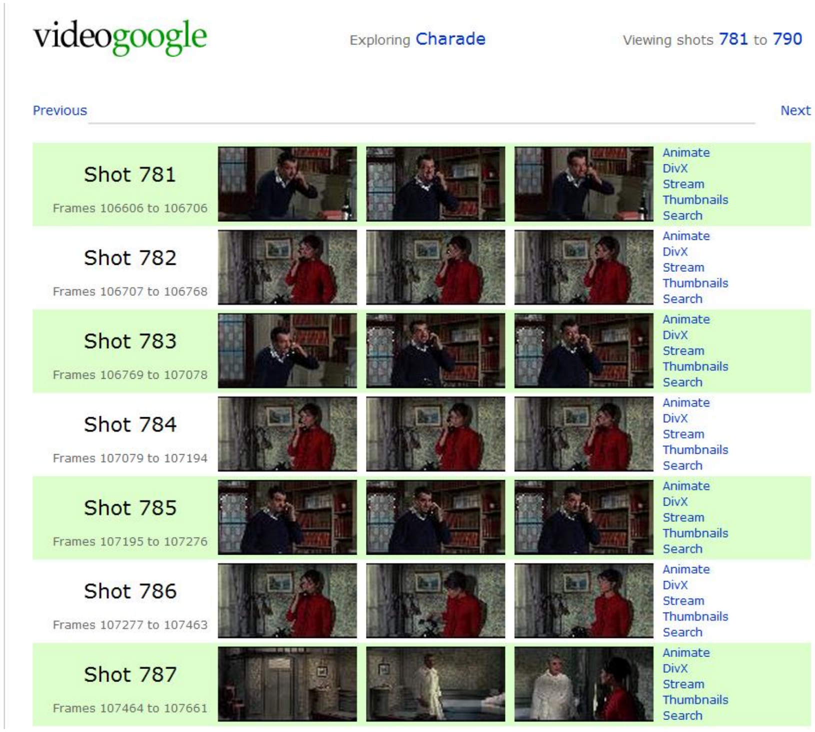
My Proposal in INS 2011

- A simple and unified framework for all topics
  - Only SIFT feature is used.
  - Single BOW model based pipeline for all topics (no any face detector and concept classifiers).
  - For one query topic, only *N* (*N*=20982) times of matching (between extreme sparse histograms) are needed to get the ranking list.

- Related Works

Related Works (1)

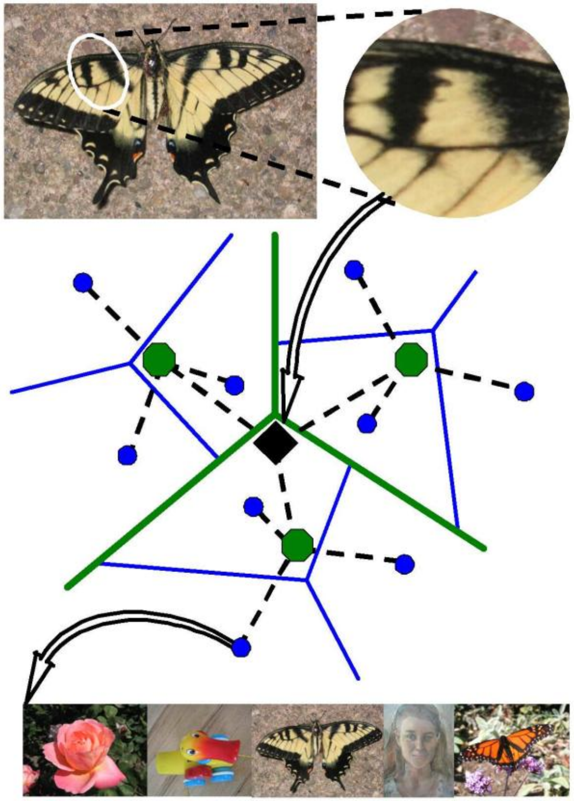
- **Video Google** [J.Sivic,ICCV'03]



- The visual BOW analogy of text retrieval is very efficient for image retrieval.

Related Works (2)

- **Scalable Recognition with a Vocabulary Tree** [D. Nister, CVPR’06]



■ Large vocabulary size improves retrieval quality.

Related Works (3)

- **In Defense of Nearest-Neighbor Based Image Classification** [O.Boiman, CVPR'08]

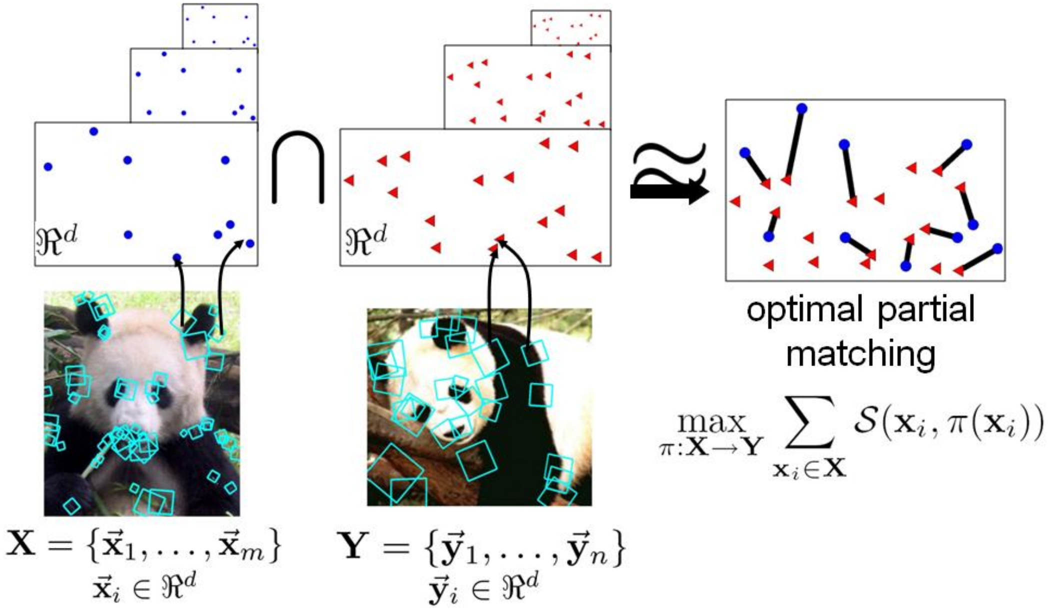
The NBNN Algorithm:

1. Compute descriptors $d_1, \ldots, d_n$ of the query image Q.
2. $\forall d_i \; \forall C$ compute the NN of d_i in C: $\mathrm{NN}_C(d_i)$.
3. $\hat{C} = arg\min_C \sum_{i=1}^{n} \| d_i - \mathrm{NN}_C(d_i) \|^2$.

- Query-to-Class (no Image-to-Image) distance is optimal under the Naive-Bayes assumption;
- Quantization degrades discriminability.

Related Works (4)

- **Pyramid Match Kernel** [K.Grauman, ICCV'05, NIPS'06]



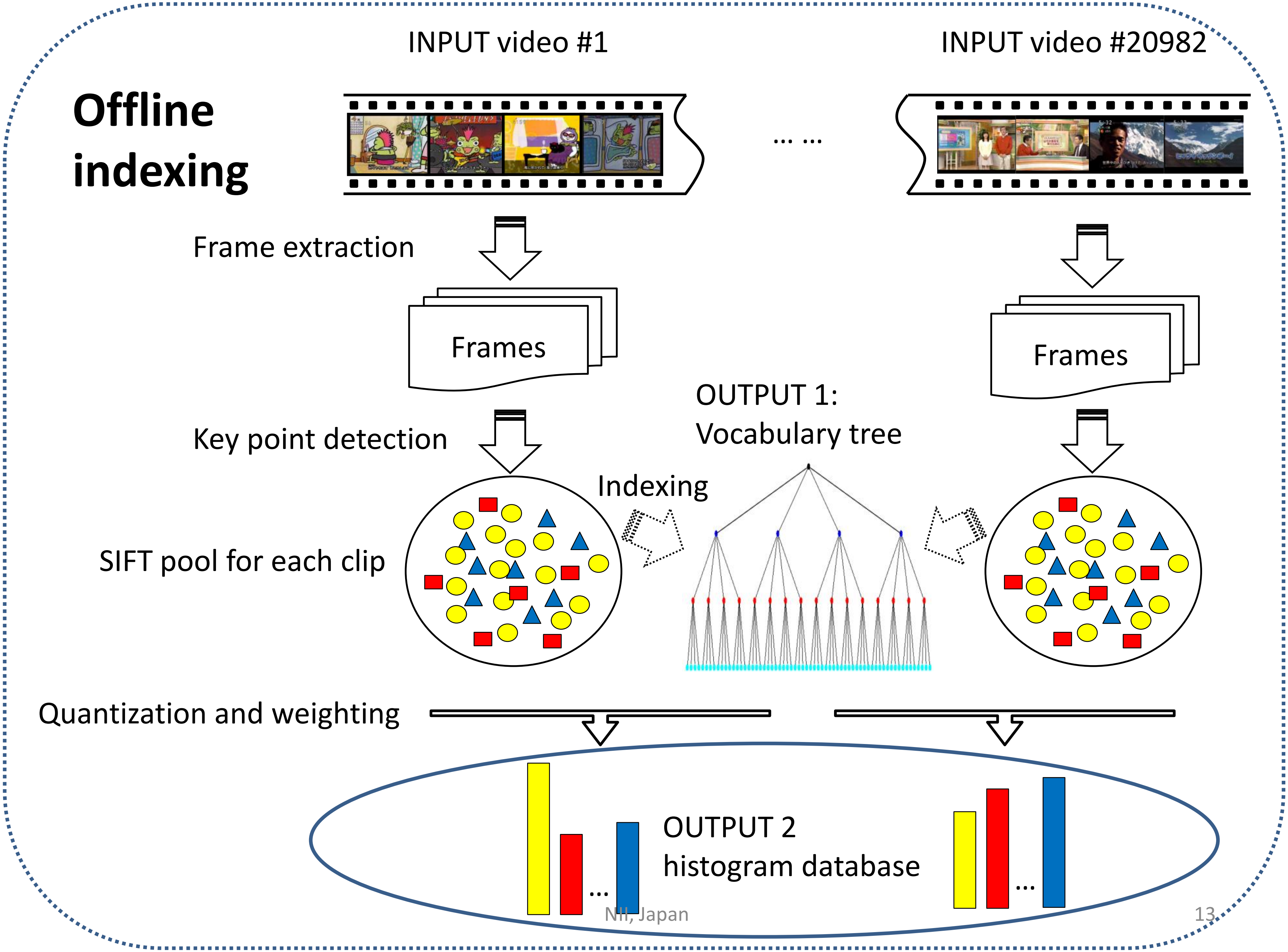
■ Hierarchical tree based pyramid intersection computes partial matching between feature sets without penalizing unmatched outliers.

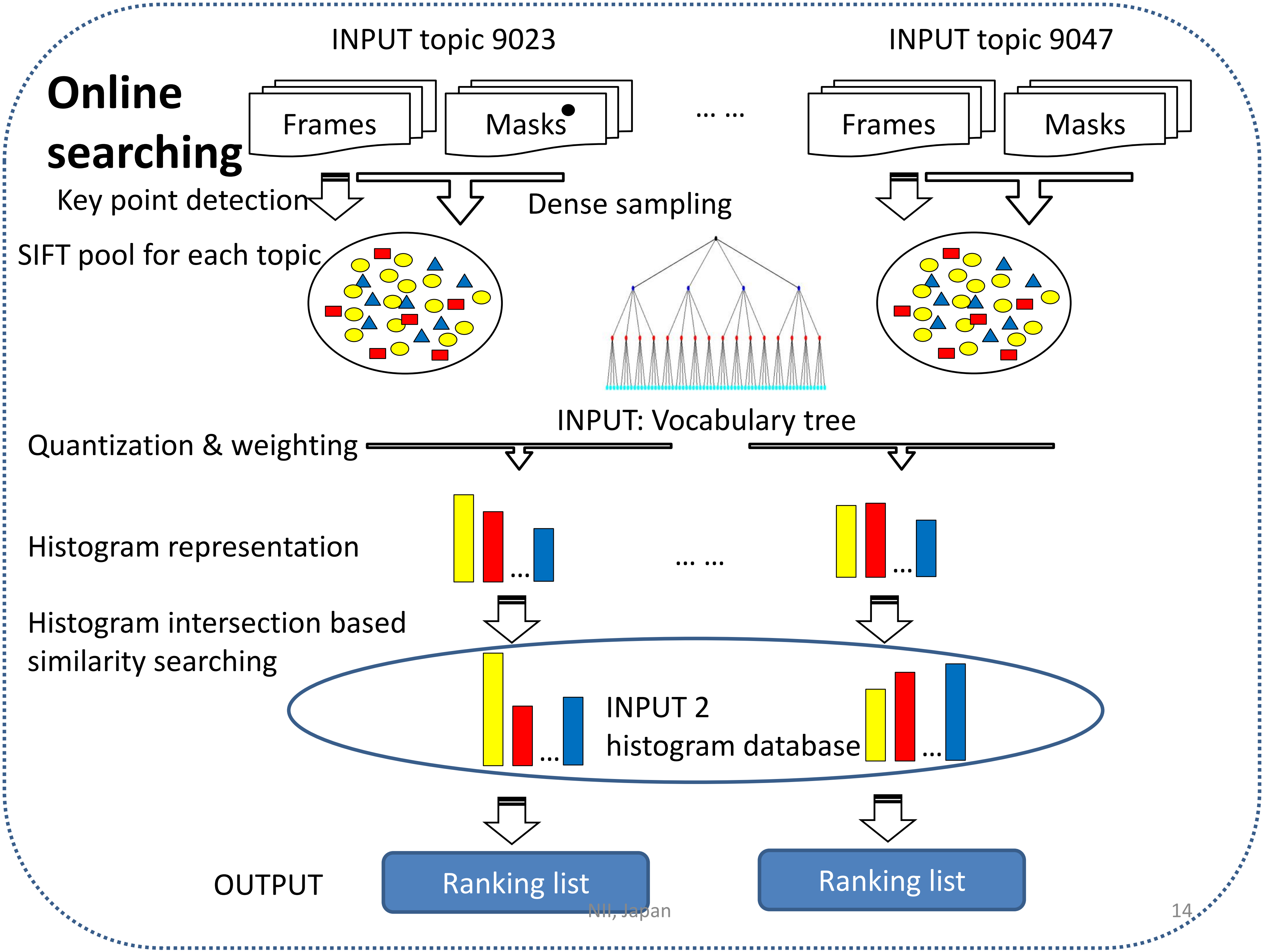
- Algorithm Overview

Large Vocabulary Tree Based BOW Framework

1. Offline indexing

2. Online searching


Offline indexing
INPUT video #1
INPUT video #20982
… …
Frame extraction
Frames
Frames
Key point detection
OUTPUT 1:
Vocabulary tree
Indexing
SIFT pool for each clip
Quantization and weighting
OUTPUT 2
histogram database
…
…

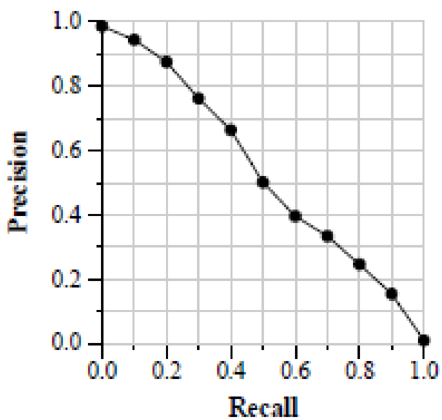

INPUT topic 9023
INPUT topic 9047
Online searching
Frames
Masks
Frames
Masks
… …
Key point detection
Dense sampling
SIFT pool for each topic
INPUT: Vocabulary tree
Quantization & weighting
Histogram representation
… …
…
…
Histogram intersection based similarity searching
INPUT 2
histogram database
…
…
OUTPUT
Ranking list
Ranking list

- Results

Run 'NII.Caizhi.HISimZ'

- Feature: 192-D color sift (cf. featurespace lib)
- Vocabulary tree: branch factor 100, number of layers 3.
- Similarity measure for ranking: histogram intersection upon *idf* weighted full histogram of codewords.
- Speed: ~15 mins for searching one topic with matlab implementation (includes all steps: feature extraction, quantization,file I/O …)

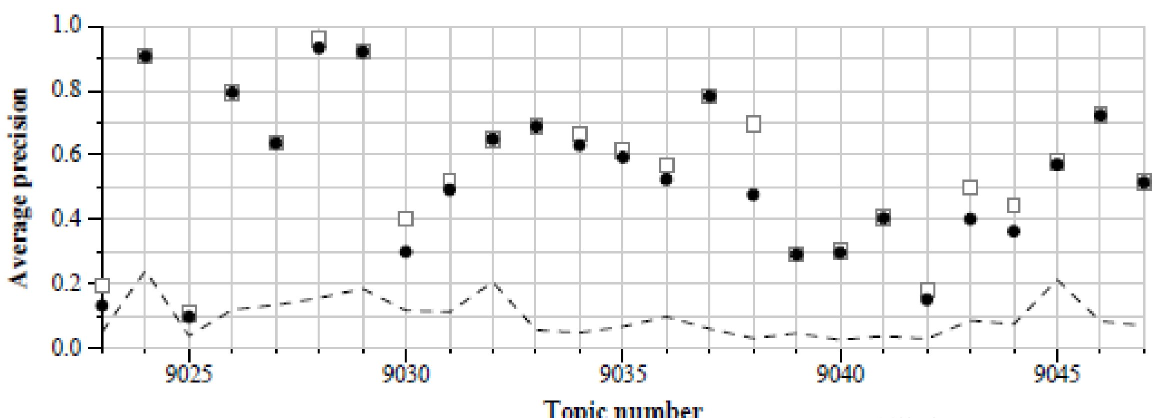
```
TRECVID 2011: instance.search results

Run ID:                          NII.Caizhi.HISimZ
Processing type:                 automatic
System training type:            X (not specified)
Condition:                       N (No IACC.1 *_meta.xml used)
Priority:                        4

          Across 25 test topics (9023-9047)

       Total relevant shots:   1830
Total relevant shots returned:   1224

Mean(prec. @ total relevant shots): 0.513
         Mean(average precision): 0.531
```



Interpolated recall	precision
0.0	0.9867
0.1	0.9446
0.2	0.8760
0.3	0.7623
0.4	0.6646
0.5	0.5019
0.6	0.3967
0.7	0.3344
0.8	0.2467
0.9	0.1547
1.0	0.0104

Precision at n shots	
5	0.9600
10	0.9040
15	0.8347
20	0.7720
30	0.6827
100	0.3840
200	0.2118
500	0.0918
1000	0.0490

Top ranked in 11 out of 25 topics, and nearly top in other 8 topics.



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

Run ‘NII.Caizhi.HISim’

- A run fused multiple combinations
  - Feature: 192-D color sift and 128-D grey sift
  - Vocabulary tree:
    - branch factor 100, and #layer 3.
    - branch factor 10, and #layer 6.
  - Weighting schemes:
    - *idf* weighting
    - hierarchically weighting (times number of nodes in that layer)
    - double weighting
- Fusion strategy: simply sorted the summation of ranking orders appeared in 12 different runs.

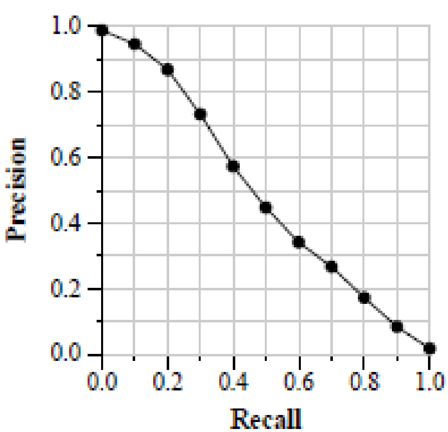
TRECVID 2011: instance.search results

Run ID: NII.Caizhi.HISim
Processing type: automatic
System training type: X (not specified)
Condition: N (No IACC.1 *_meta.xml used)
Priority: 3

Across 25 test topics (9023-9047)

Total relevant shots: 1830
Total relevant shots returned: 1124

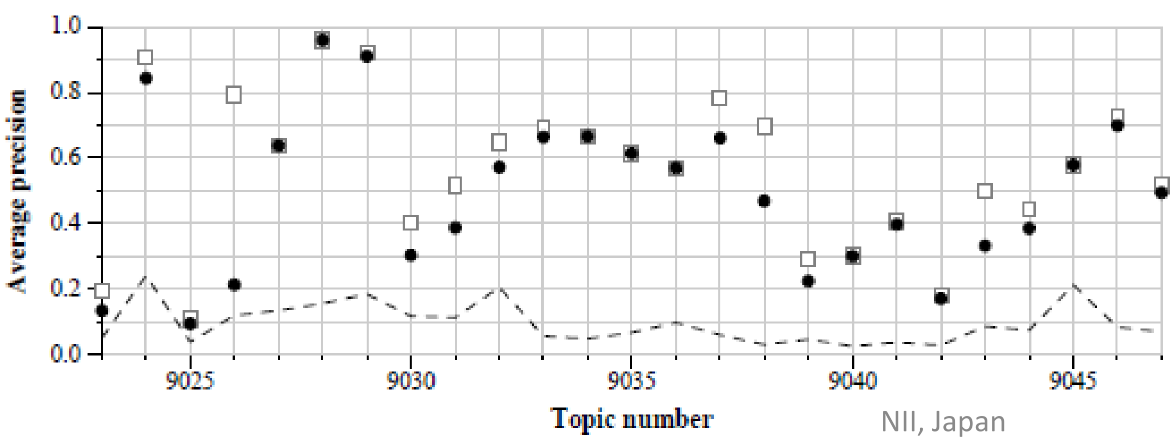
Mean(prec. @ total relevant shots): 0.488
Mean(average precision): 0.491



Interpolated recall	precision
0.0	0.9891
0.1	0.9473
0.2	0.8691
0.3	0.7331
0.4	0.5741
0.5	0.4483
0.6	0.3425
0.7	0.2680
0.8	0.1734
0.9	0.0849
1.0	0.0179

Precision at n shots	
5	0.9600
10	0.9160
15	0.8240
20	0.7480
30	0.6640
100	0.3616
200	0.1948
500	0.0840
1000	0.0450

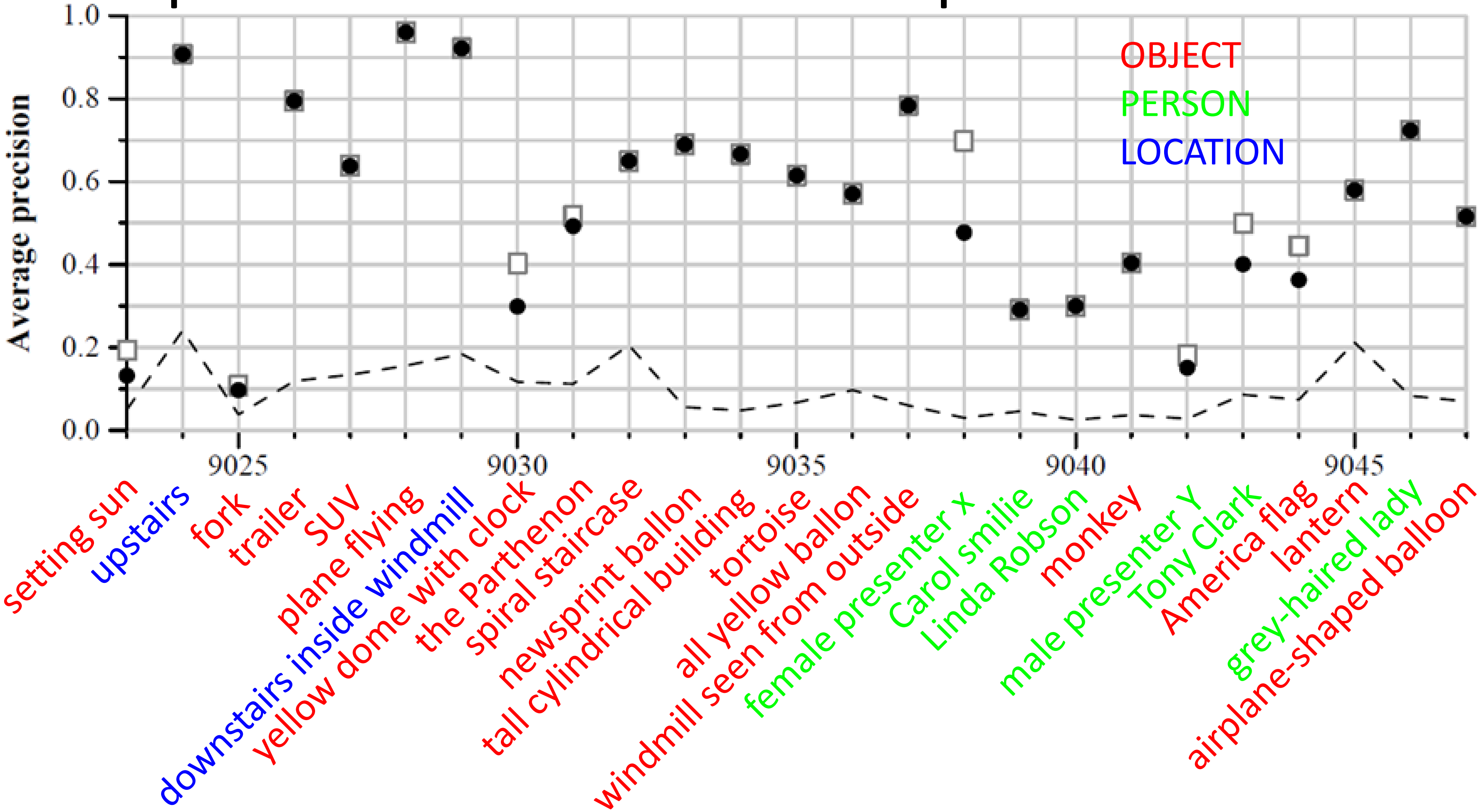
Top ranked in 7 topics



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

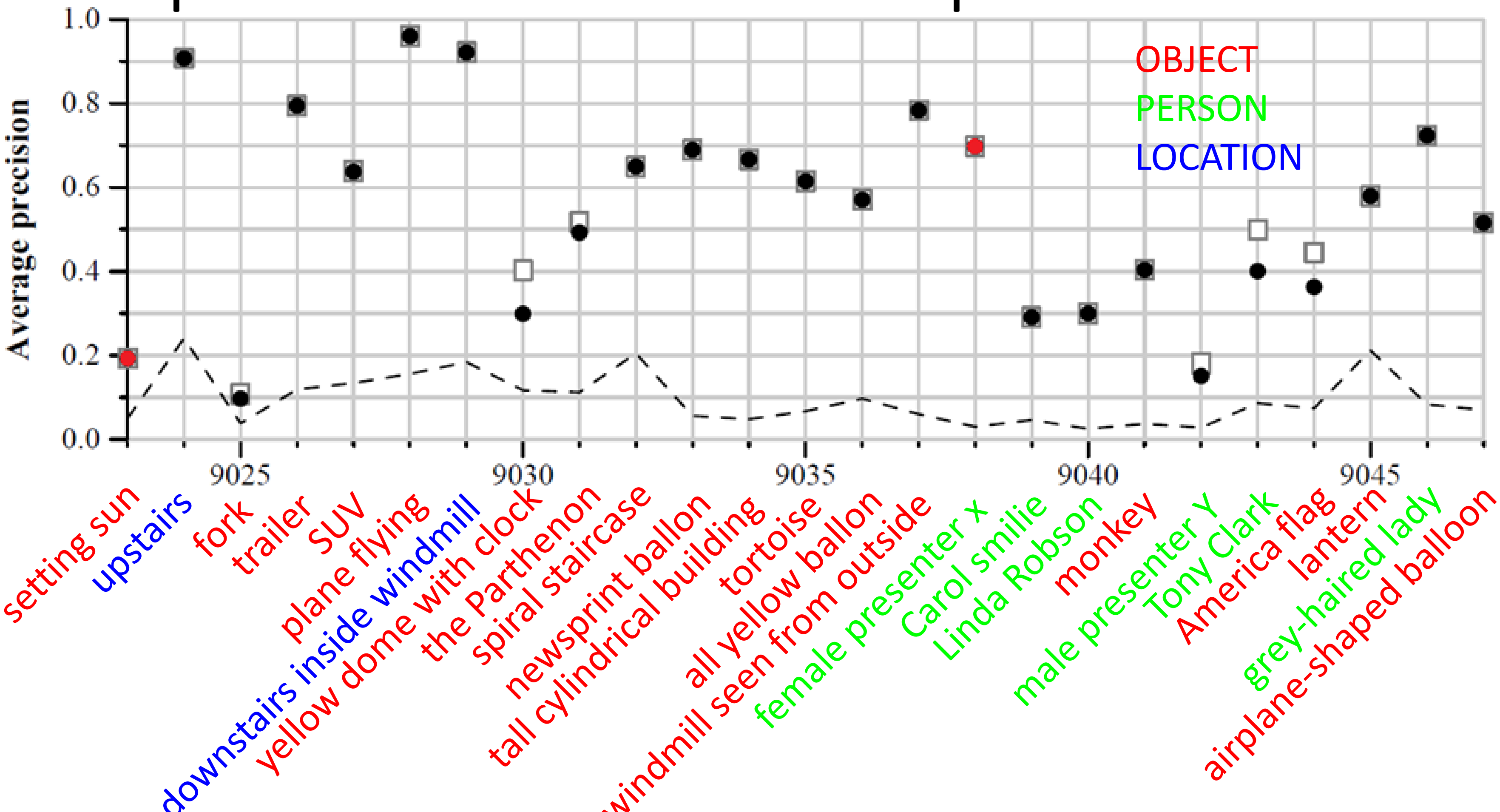
Best cases of two runs with this algorithm

- Top ranked in 17 out of 25 topics



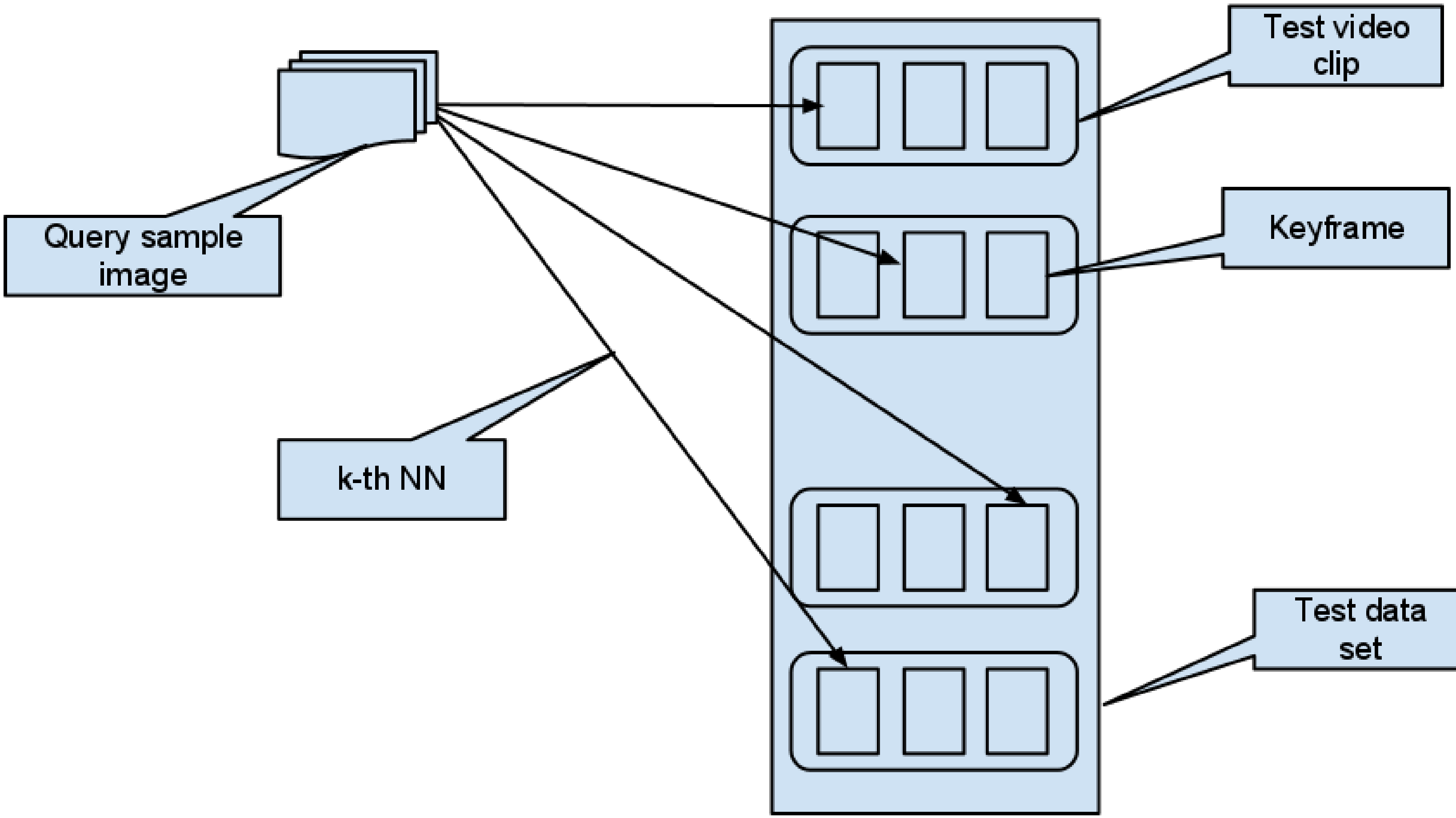
Best cases of all runs submitted by our lab

- Top ranked in 19 out of 25 topics



NOTE: other two red best cases are from the Run ‘NII.SupCatGlobal’ contributed by Dr. Duy-Dinh Le

Framework of Run 'NII.SupCatGlobal'



- Demos


DAVID DEVANT
G-ELZY

- Discussion and conclusion

Discussion

- Is INS2011 much easier than INS2010?
  - Average MAP increased from ~0.01 to ~0.1.
- Is performance influenced by object size?
  - MAP on smallest objects ‘setting sun’ and ‘fork’ are lowest.
- How to make a true instance search algorithm rather than a duplicate detection one?
  - Mostly only (near) duplicates can be retrieved with current algorithm.
- How to improve performance on those ‘hard’ topics?
  - To combine current algorithm with concept detectors.
  - To make a tradeoff between object and context regions, does that make a great difference?
- Current framework acquired top performance in 3 out of 6 ‘person’ topics, how to explain it?

Conclusion of Our Algorithm

- Building BOW framework upon hierarchical k-means based large vocabulary quantization.
- Matching similarity between topics and video clips.
- Balancing both context and object regions while computing similarity distance.
- Computing histogram intersection on hierarchically weighted histogram of codewords for ranking.

Thanks!