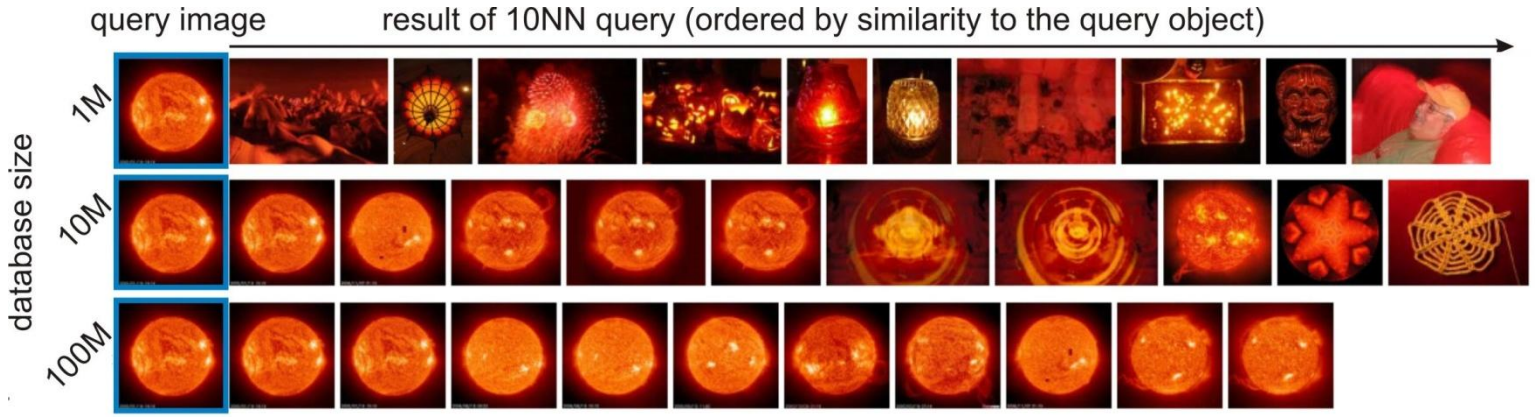
DISTINCT NEAREST NEIGHBORS QUERIES FOR SIMILARITY SEARCH IN VERY LARGE MULTIMEDIA DATABASES

ACM WIDM 2009,
Hong Kong, China

T. Skopal, V. Dohnal, M. Batko, P. Zezula

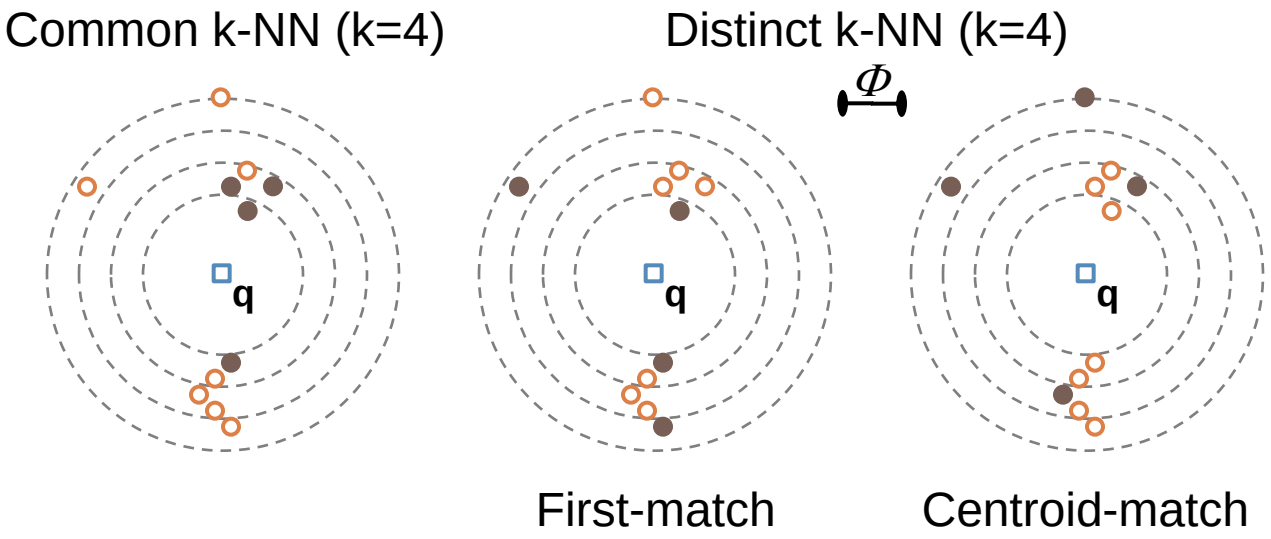
Similarity Searching

- Exact searching in images is not sufficient.
- Content-based searching
  - Users retrieve visually similar images.
  - Even not annotated images are retrieved.
- Nearest neighbors query
  - Loosing its discriminative power



Distinct Nearest Neighbors Query

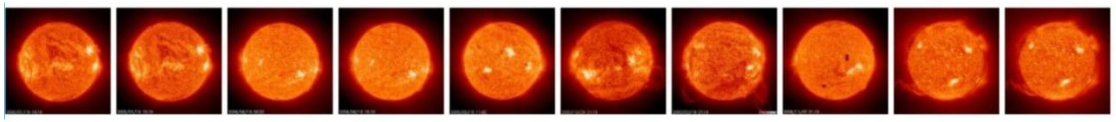
- Cope with density of searching space
- Idea: diminish “duplicates” of objects in the result to increase response quality
  - User defines a separation constant Φ



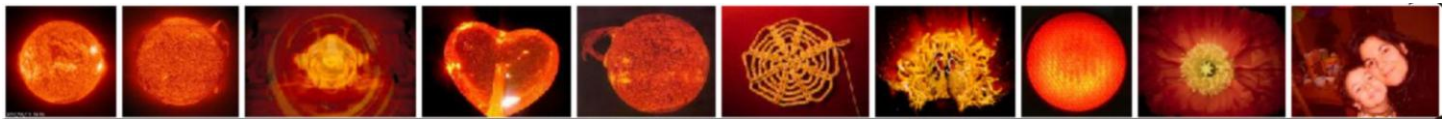
Example of Distinct kNN

- Database: 100 million images
- Query object:

- Result of 10-NN:



- Result of 10-DNN (Distinct Nearest Neighbors):



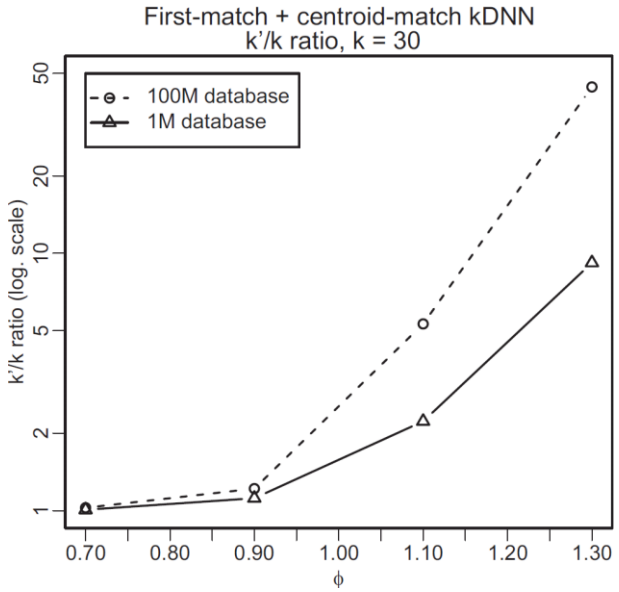
Experimental Evaluation

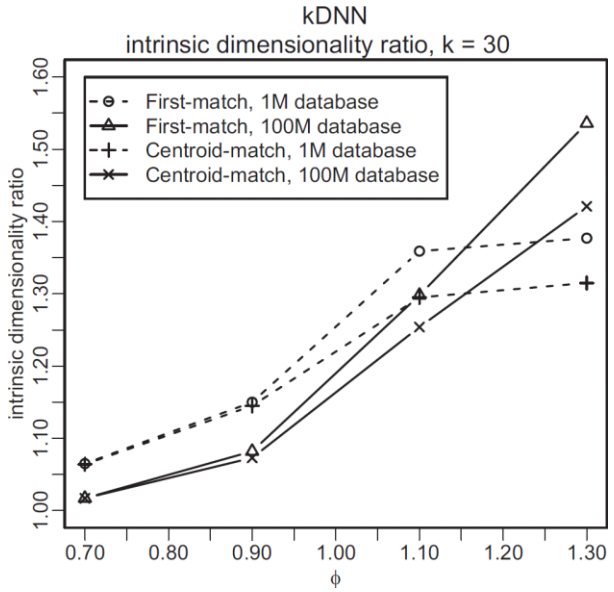
- CoPhIR dataset:
  - 100 mil. photos, MPEG-7 features
- Algorithms for distinct k-NN
  - implemented in MUFIN (http://mufin.fi.muni.cz/)
- User satisfaction with results:
  - 30 users (student of IT)
  - 45 queries
  - User did not know whether the displayed query was k-NN or k-DNN.

Query	Φ	Percentage	
Cannot decide		8%	
Classic k-NN		26%	
10-DNN	0.8	30%	} 66%
10-DNN	1.0	14%	
10-DNN	1.2	22%	

Experimental Evaluation (cont.)

- Statistical comparison of 30-NN and 30-DKNN
  - 100 mil. and 1 mil. subset
  - Ratio k' / k, where k' = # of NN checked by 30-DKNN
  - Ratio of intrinsic dimensionalities: $\rho = \frac{\mu^2}{2\sigma^2}$





Conclusions

- Properties of distinct nearest neighbors:
  - Returns distinct results
  - More robust than k-NN when used on large databases
  - Evaluation by real users confirmed better results
- Performance summary
  - Implemented under the same framework in Java
  - Time overhead is 2-7% of original k-NN costs
    - Including increased number of NN used
    - Including k-DNN algorithm's computation
  - Can be used in real-time