

A contextual dissimilarity measure for accurate and efficient image search

Hervé Jégou, Hedi Harzallah, Cordelia Schmid

INRIA, LEAR project team

CVPR 2007

Problem setup: image search system

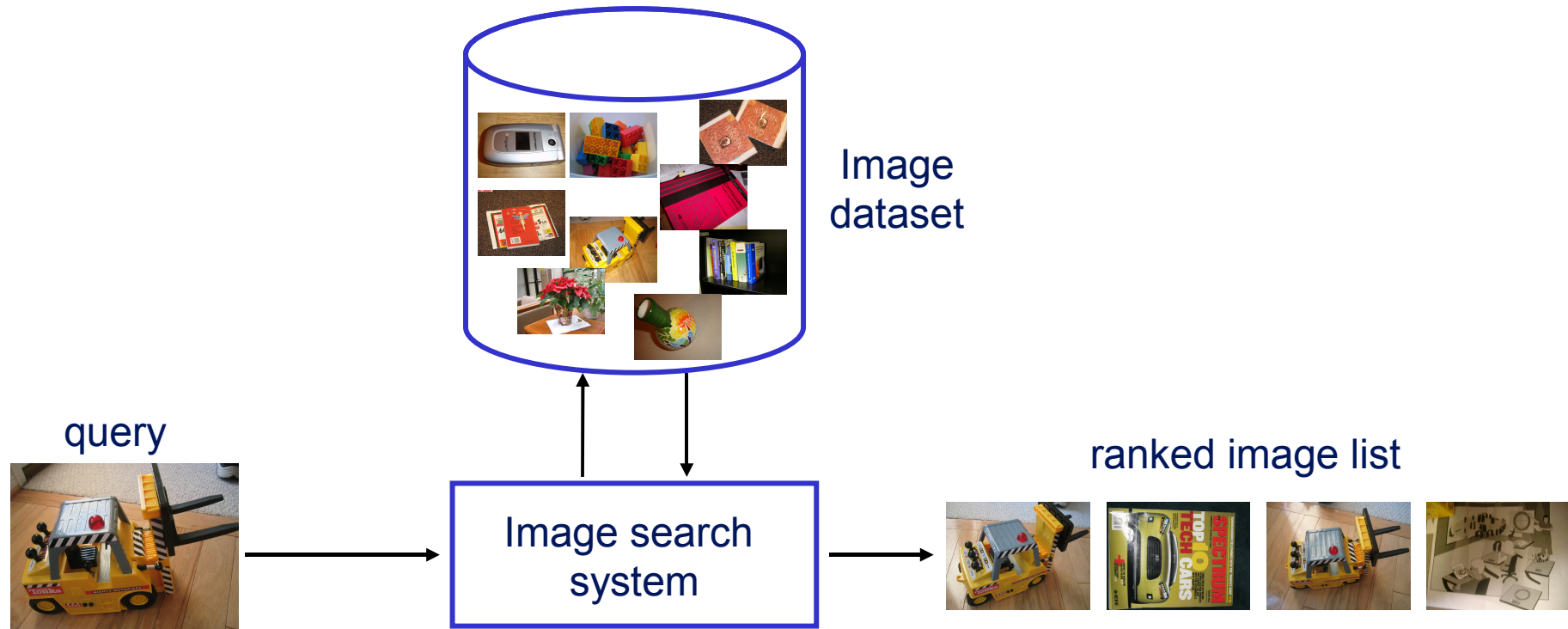
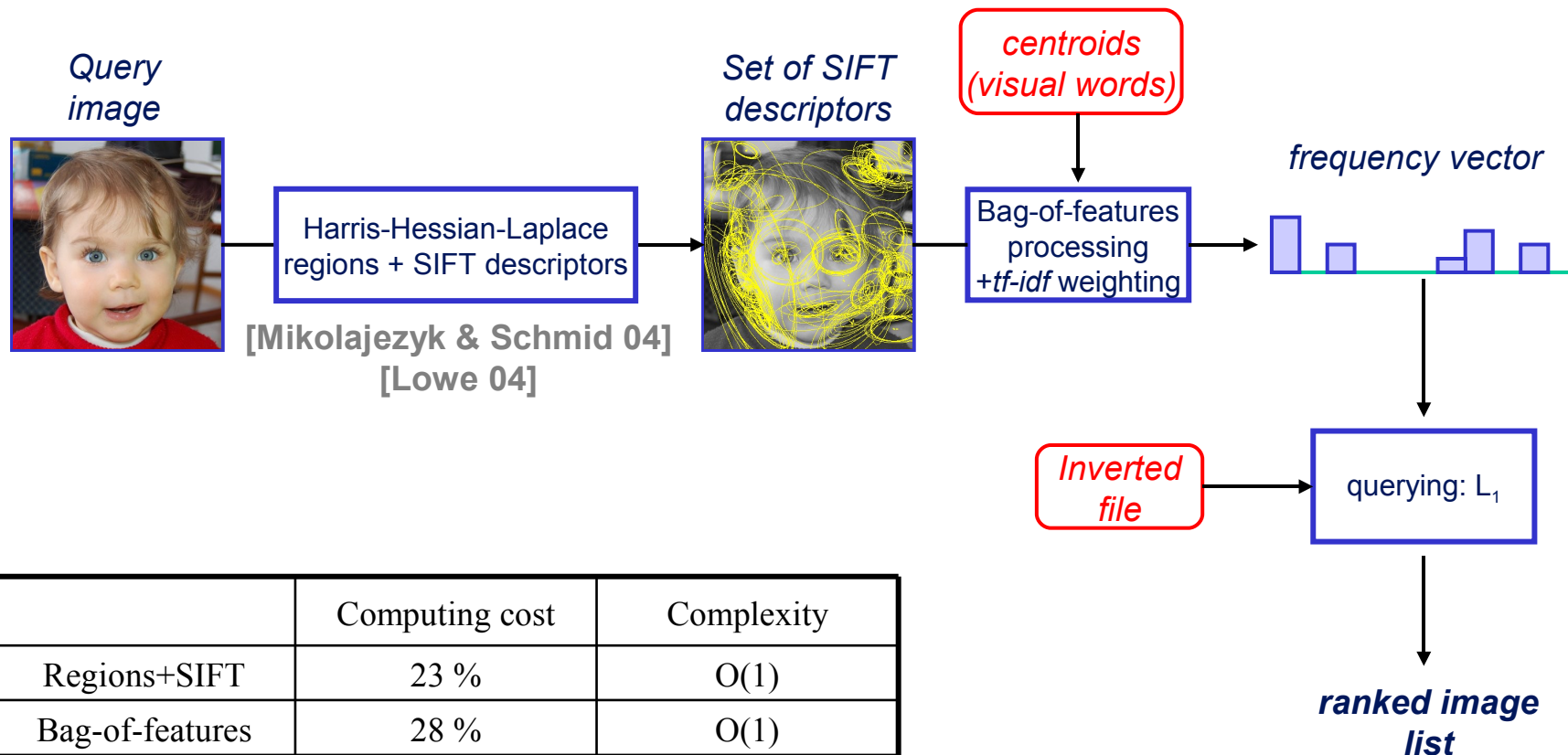


Image search system: “Video-Google”-like setup

- Video-Google [Sivic & Zisserman 03]

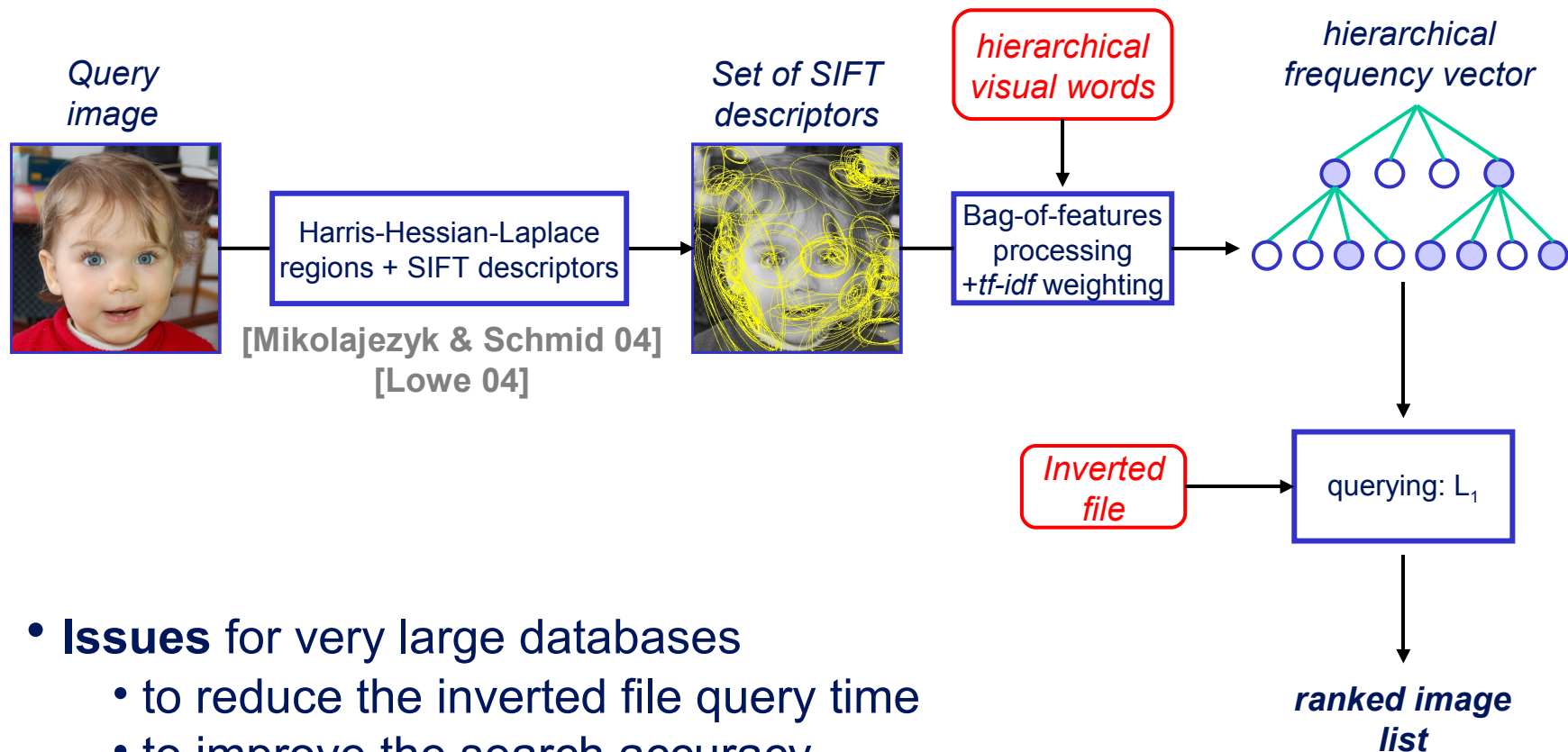


	Computing cost	Complexity
Regions+SIFT	23 %	$O(1)$
Bag-of-features	28 %	$O(1)$
Inverted file query	49 %	$O(n)$

10 000 visual words, $n=100\,000$ images

Recent improvement and remaining issues

- Fast assignment to visual words [Nistèr & Stewénius 06]



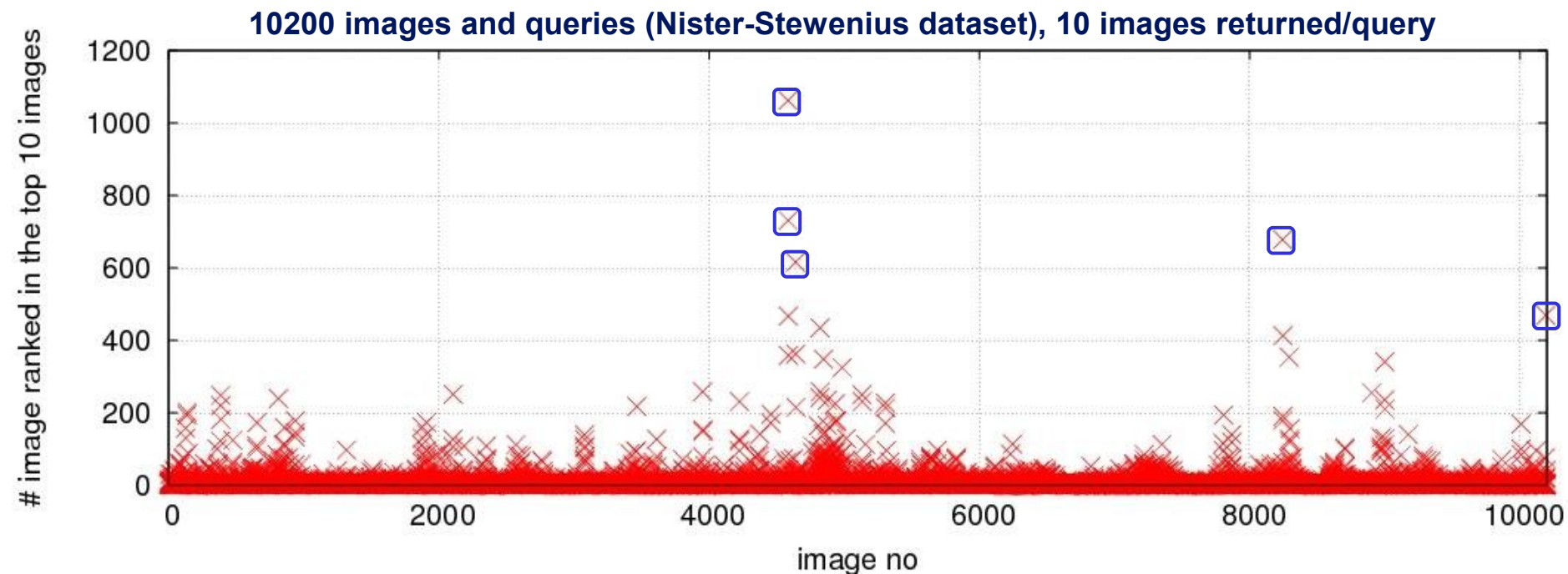
- Issues** for very large databases
 - to reduce the inverted file query time
 - to improve the search accuracy

Outline

- **Improved accuracy:** contextual dissimilarity measure (CDM)
- **Faster search:** efficient search structure for ANN of frequency vectors
- Combining both approaches
- Large scale evaluation

Motivation for a new dissimilarity measure

- The space of frequency vectors is anisotropic
- Some images are well ranked with very high/very low probability:
 - “*frequently returned images*”
 - “*never seen images*”

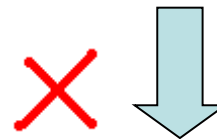
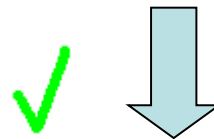
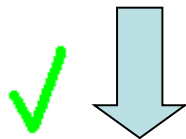
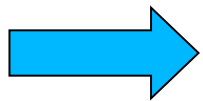


Neighborhood symmetry

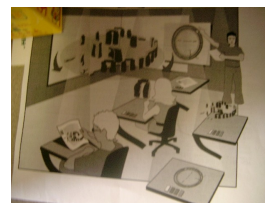
- The neighborhood, defined as a set of k nearest neighbors, is not **symmetric** in a k -nearest-neighbors sense
- What we want:

if an image R is well ranked for a query image Q ,
then Q should be well ranked for query image R

→ not the case for *frequently returned images* or *never seen images*

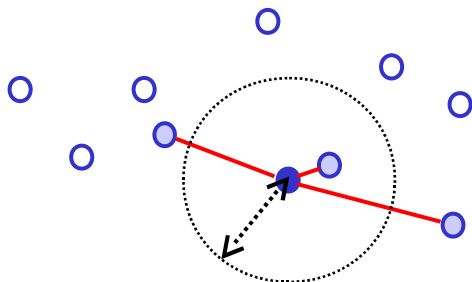


asymmetry of k-nearest
neighbor search



Contextual dissimilarity measure (CDM)

Neighborhood of a frequency vector: its set of k nearest neighbors



Neighborhood “distance” r_i of a frequency vector w_i : the average dissimilarity measure to its neighborhood

Idea:

- to modify the dissimilarity measure between frequency vectors
- so that the average dissimilarity of a vector to its neighborhood is almost a constant

How ?

→ iterative regularization of the frequency vectors' neighborhood

CDM: iterative approach (off-line)

For each iteration

- Compute the neighborhood distance $r^k(w_i)$ for each vector
- Update the distance between frequency vectors:

$$d^{k+1}(w_i, w_j) = d^k(w_i, w_j) \left(\frac{\bar{r}^k}{r^k(w_i)} \frac{\bar{r}^k}{r^k(w_j)} \right)^\alpha$$

$0 < \alpha < 1$: smoothing factor

\bar{r}^k : geometric mean of $\{r^k(w_i)\}_i$

Iterate until it converges (for α small enough) to the fixed point

$$\forall w_i, r(w_i) = \bar{r}$$

CDM: iterative approach

Defining *distance update terms* as

$$\delta_i = \prod_k \left(\frac{\bar{r}^k}{r^k(w_i)} \right)^\alpha$$

The CDM between frequency vectors can be simply written as

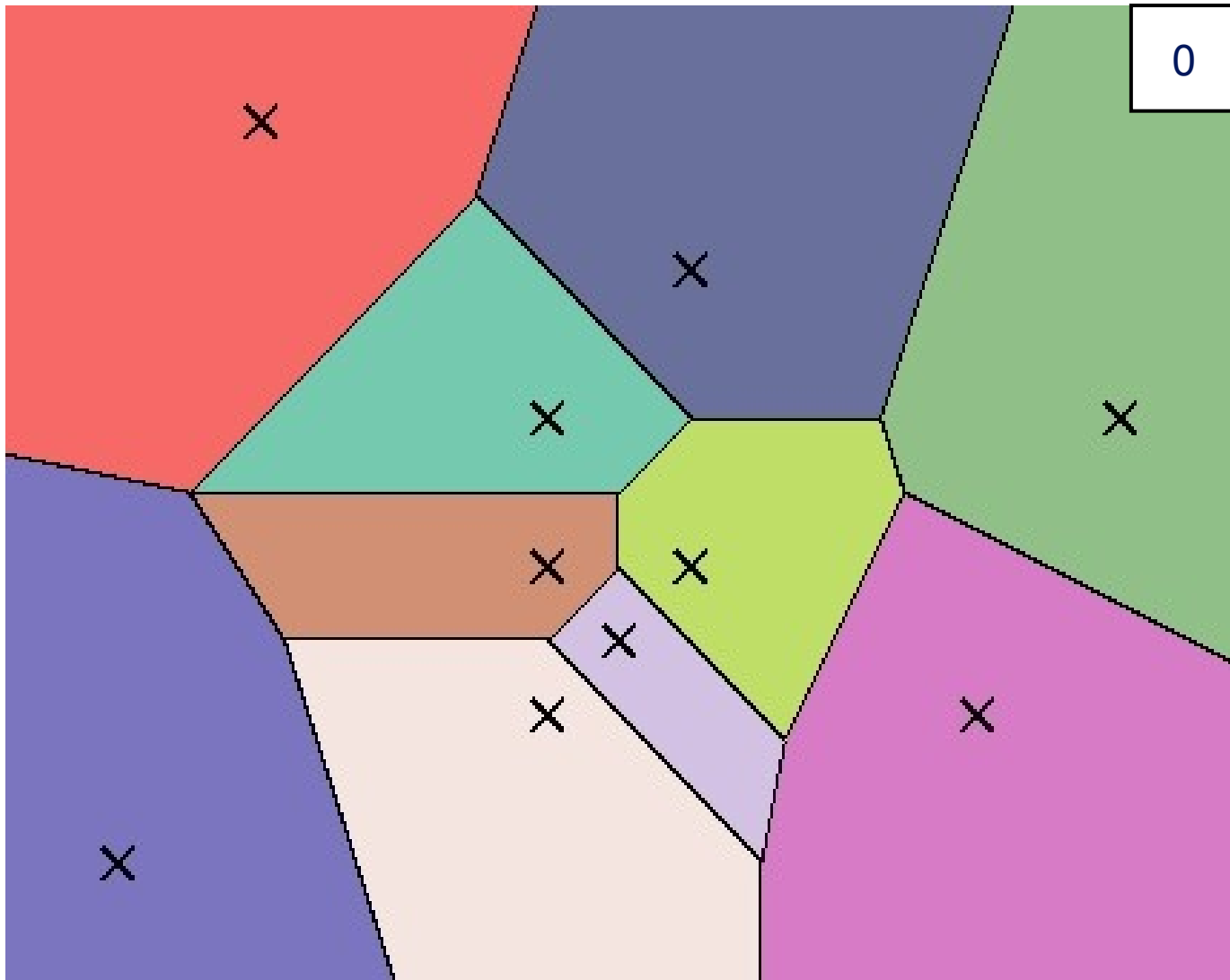
$$\text{CDM}(w_i, w_j) = d(w_i, w_j) \delta_i \delta_j$$

For a query q , we return

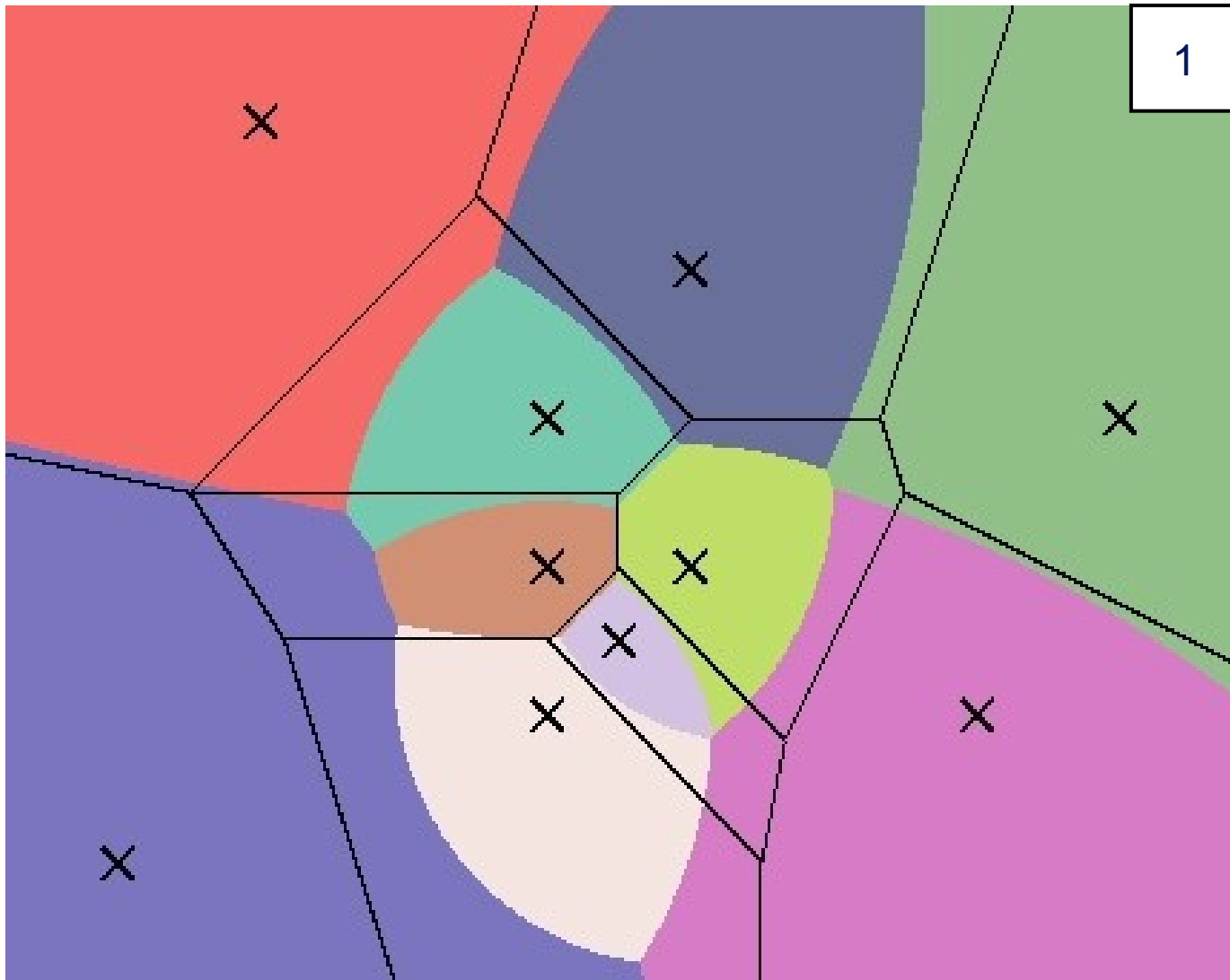
$$\arg \min_{w_i} d(w_i, q) \delta_i$$

The δ_i are computed off-line

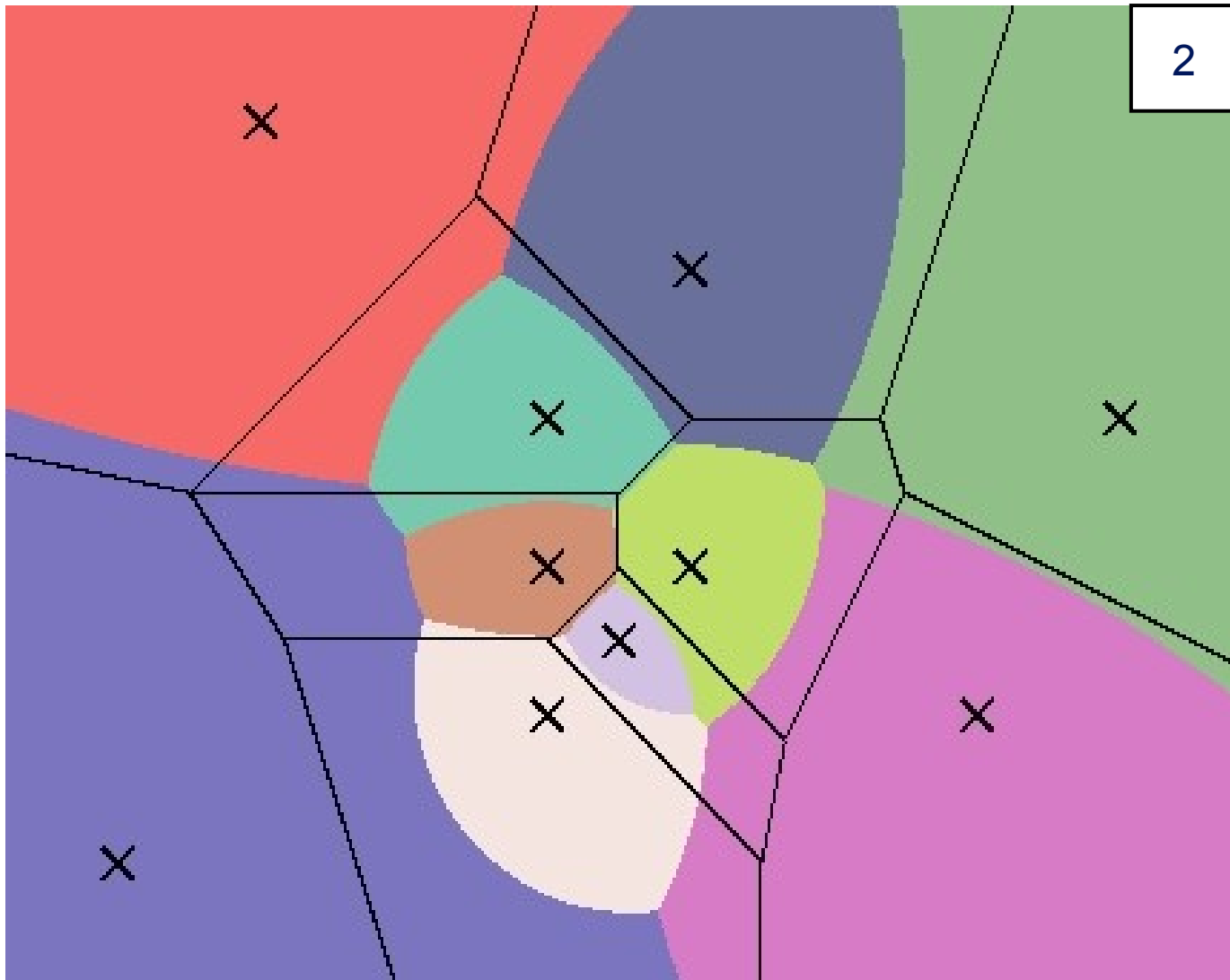
CDM toy example: impact of the iterations



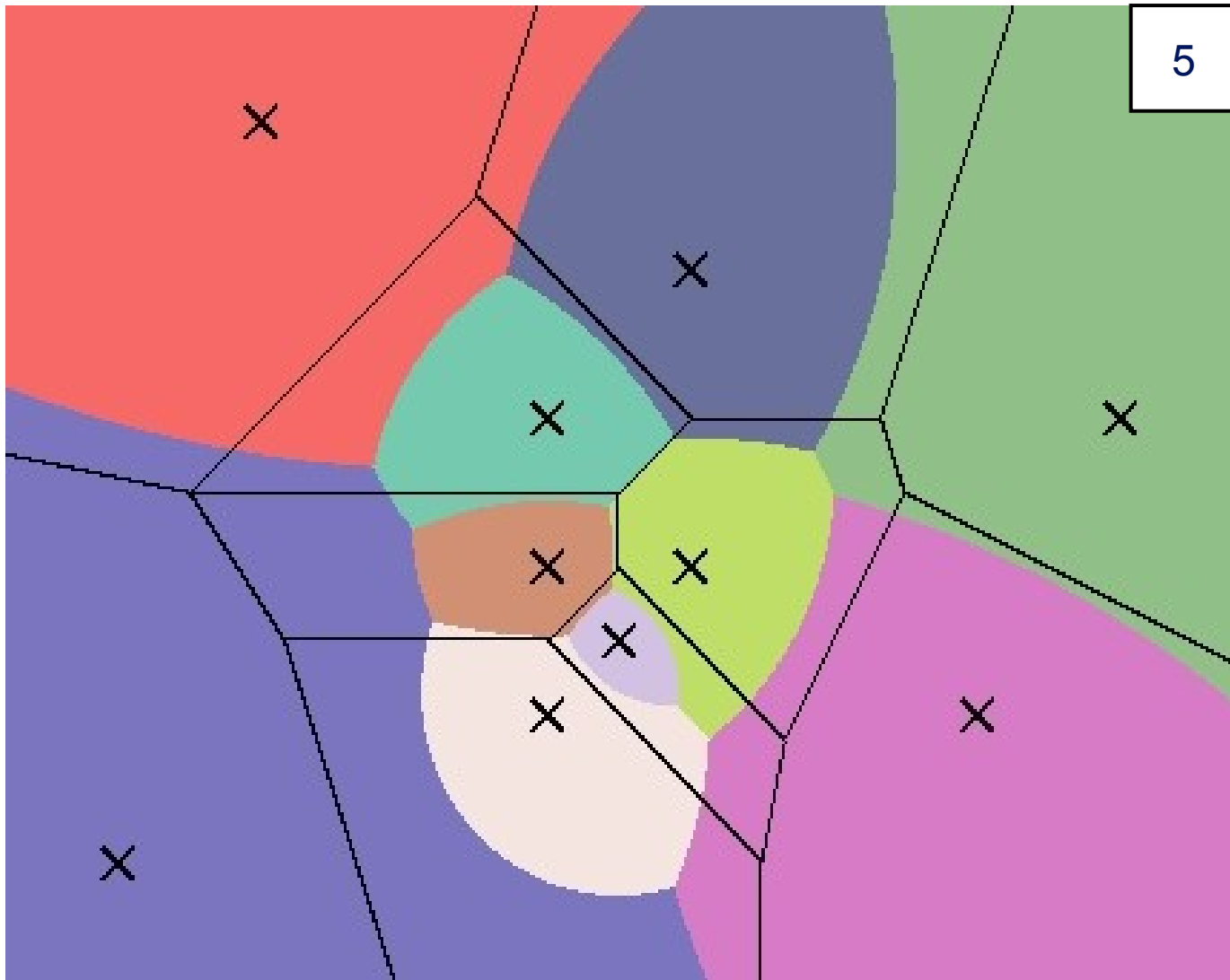
CDM toy example: impact of the iterations



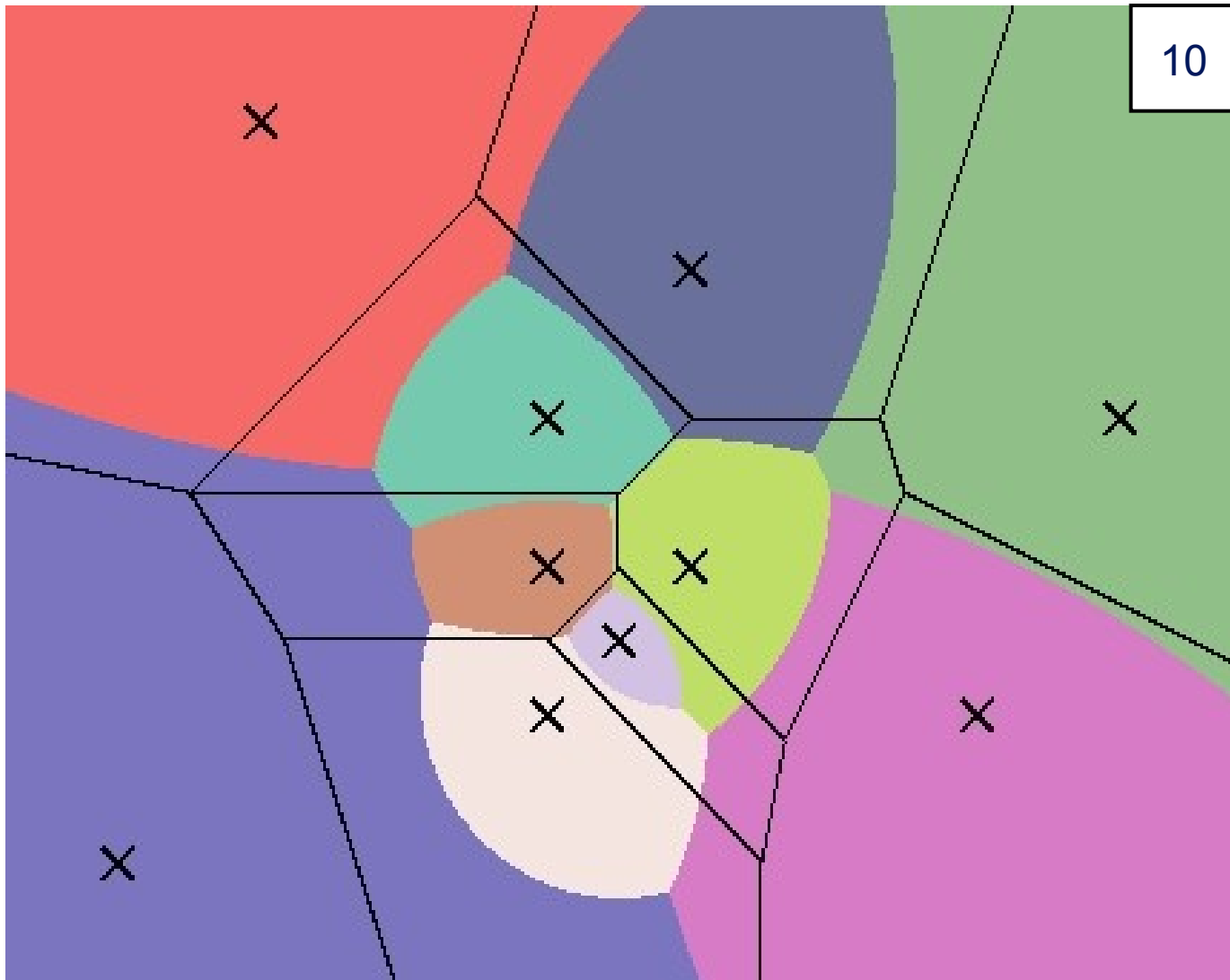
CDM toy example: impact of the iterations



CDM toy example: impact of the iterations

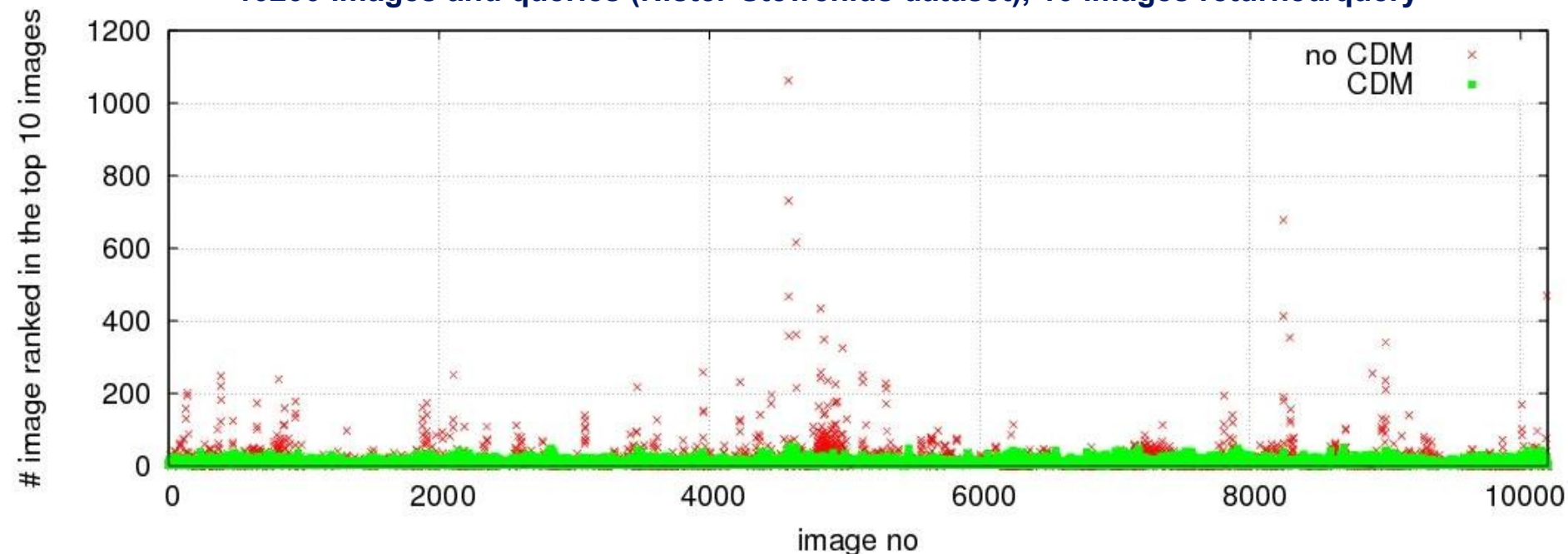


CDM toy example: impact of the iterations



Impact on never/frequently returned images

10200 images and queries (Nister-Stewenius dataset), 10 images returned/query



	Never seen images	The most frequent image appears
No CDM	9.7 %	1 062 times
CDM	0.2 %	54 times

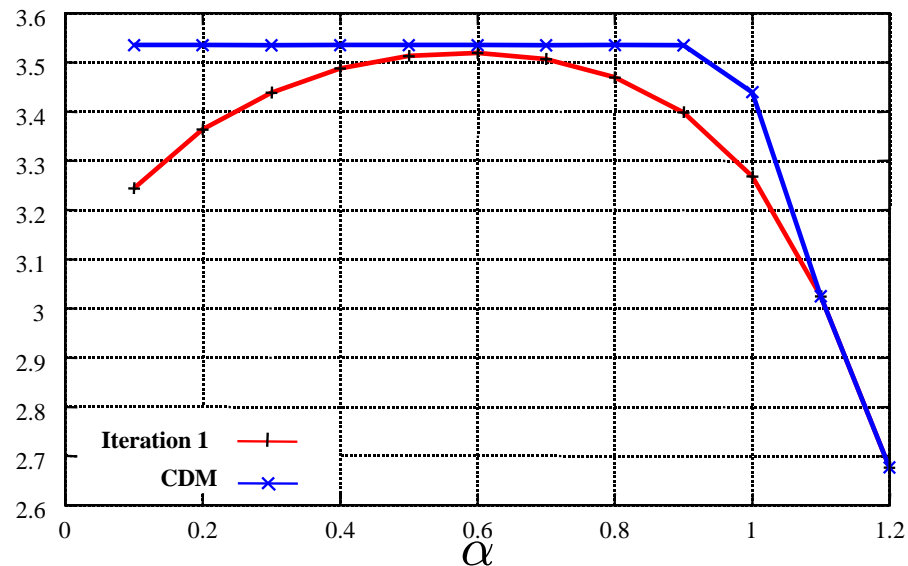
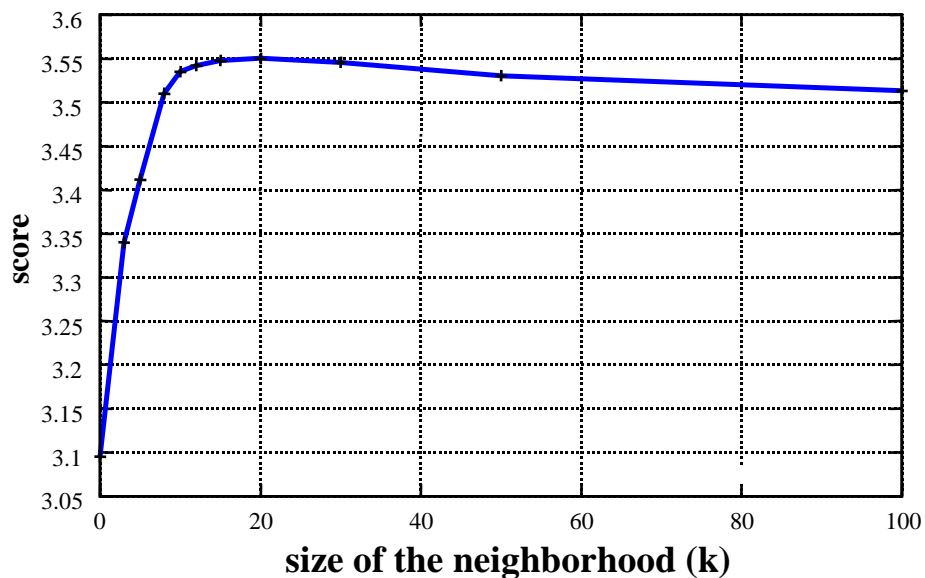
CDM: results

- Nister-Stewenius dataset: 10200 images, 4 per object
- Relevance measure: number of relevant images among the first 4 images → best possible score = 4
- SIFT centroids obtained on an independent image dataset

frequency vector dimension	distance	No CDM	CDM
30000	L_1	3.136	3.548
30000	L_2	2.676	3.368

- The best score obtained by Nister and Stewenius is **3.29** (codebook learnt on the training set, hierarchical vocabulary $\sim 10^6$ visual words)

Impact of the parameters



- The results are stable w.r.t. all parameters
- The parameter α only impacts the convergence speed

CDM: query example

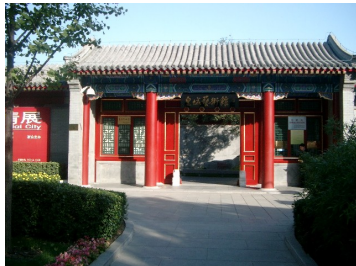


query

Results using L_1



Results using L_1 +CDM



Efficient search structure for frequency vectors

Vast literature on approximate nearest neighbors search

- Locality-Sensitive Hashing [Gionis, Indyk & Motwani 99]
- ANN [Mount & Arya 98]

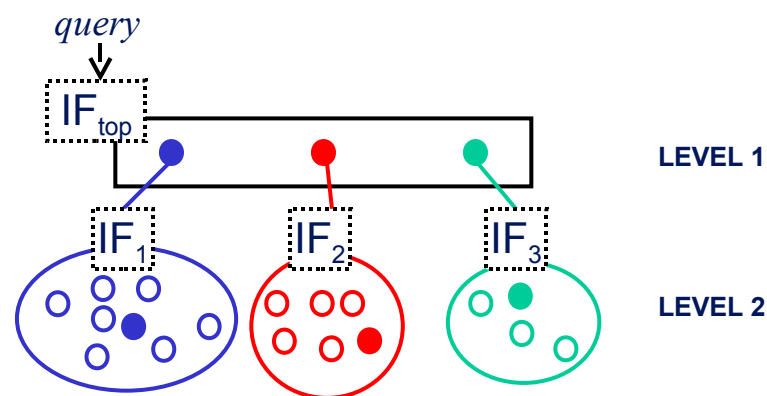
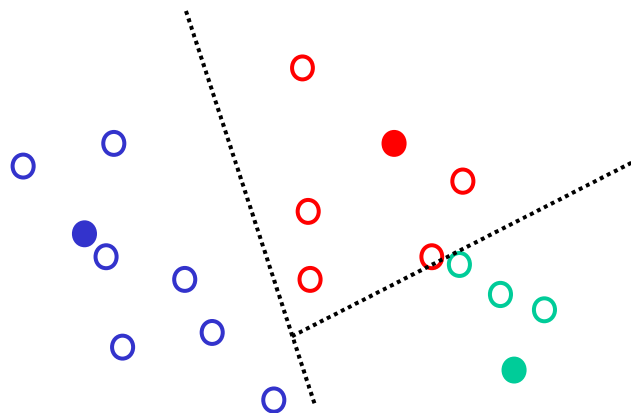
However,

- not relevant for very high dimensions (e.g., 10000)
- not adapted to sparse vectors
⇒ inverted files are efficient while performing exact search

We propose a two-level file search structure based on

- document clustering
- inverted files at each level

Clustering of frequency vectors

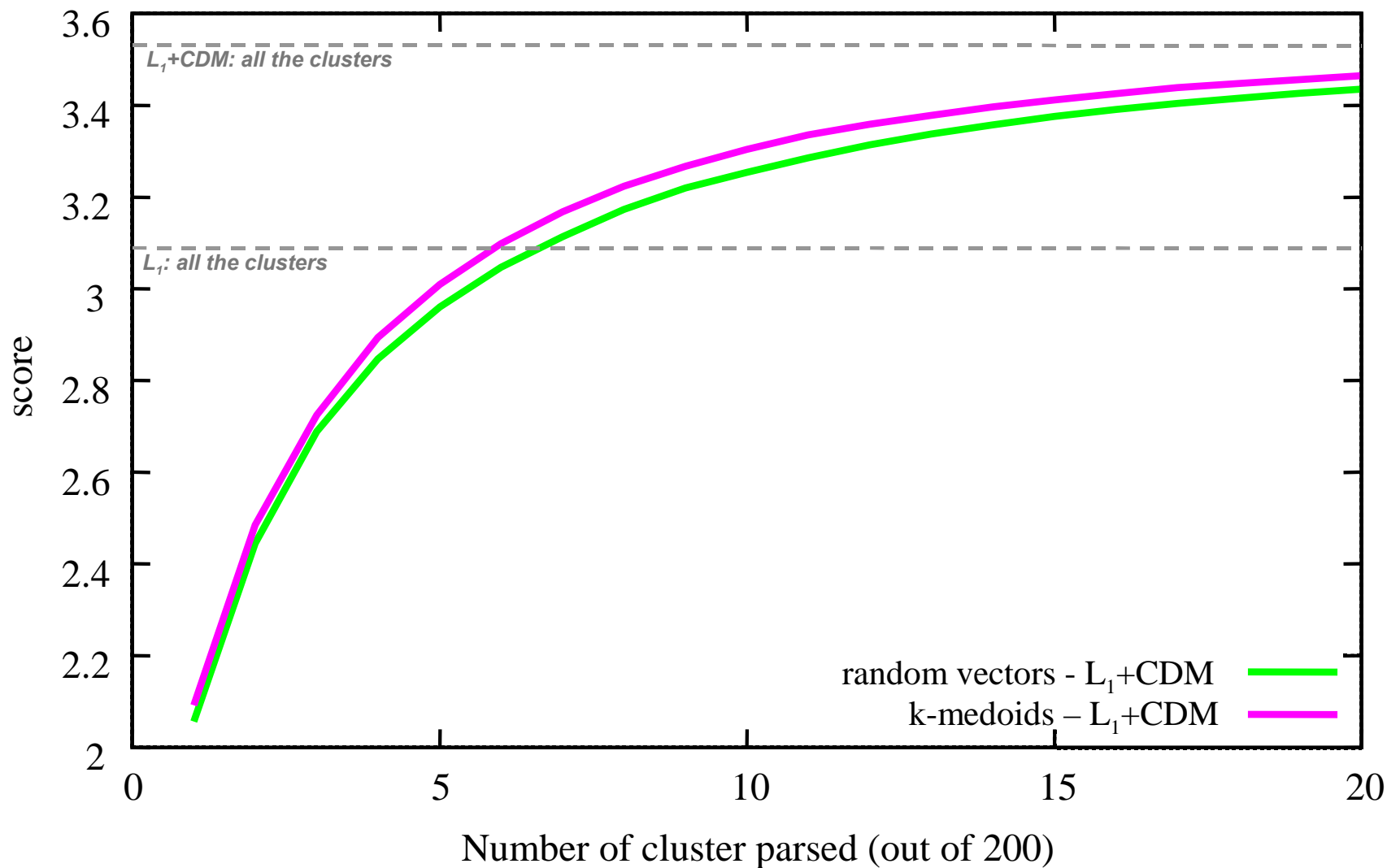


Cluster centers obtained using k-medoids or randomly chosen
⇒ sparse centroids (\neq k-means)

Search:

- query the first-level inverted file and find the k' closest medoids
 - use the associated inverted files of the second level to compute the distances and to rank the images
- close-to-optimal accuracy with 10% of the clusters

Results for the efficient search structure



Combining the approaches: efficient CDM calculation

Issue

→ the CDM computation is of quadratic complexity in the number of database images

Proposed solution

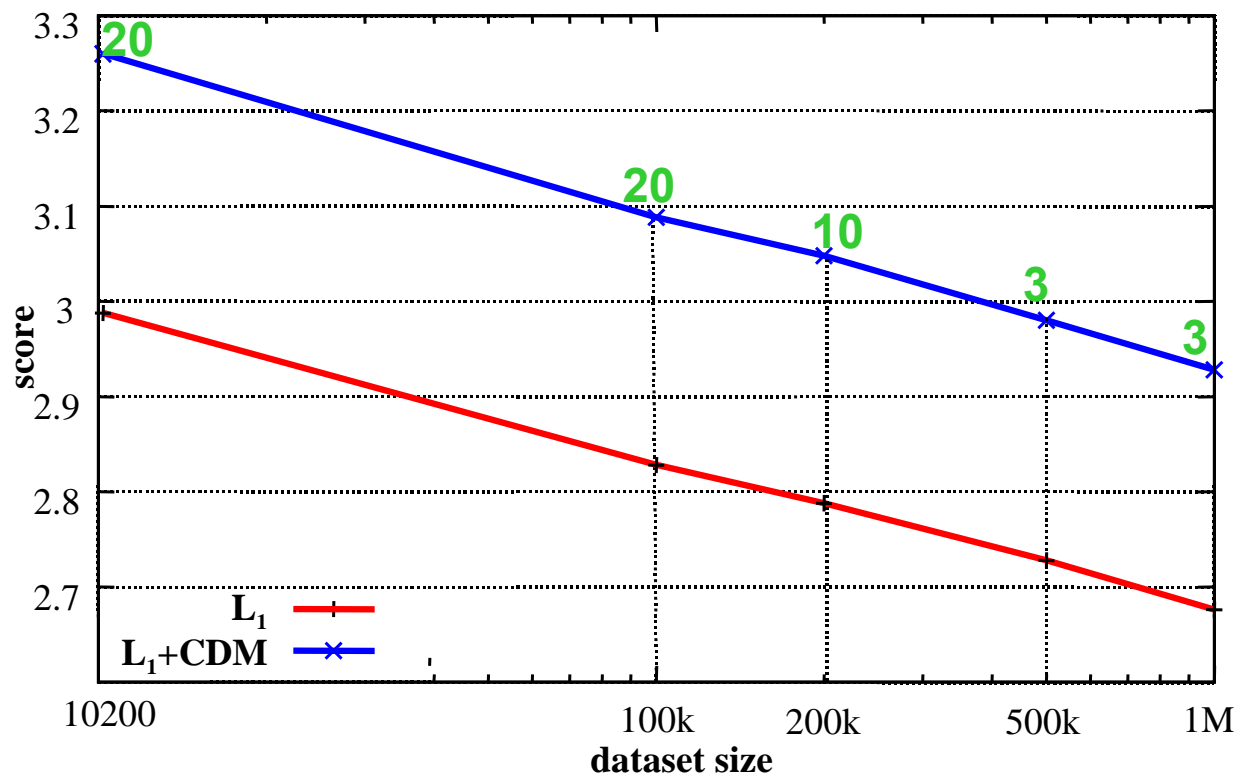
→ to use the efficient search structure to retrieve the neighborhood and to compute the update terms

Observation

- the neighborhood approximation does not noticeably impact the accuracy, even when a small number of clusters is used
→ the CDM can be used for very large datasets

Large scale evaluation of the combined approach

- Dataset: 1M images from the web + Nister-Stewenius dataset
- 1 iteration ($\alpha=0.6$), two-level efficient search structure (400 clusters)



In **green**: the number of clusters used to compute the CDM update terms (out of 400)

40 image clusters for the search

Conclusion and perspectives

Image search accuracy is significantly improved using the CDM

Efficient search structure

- approximate nearest neighbors search for sparse vectors
- based on a two-level inverted file
- this challenging problem requires further research

Combining these two approaches leads to a scalable image search system that improves both efficiency and accuracy