CONTENT-BASED RETRIEVAL ON VERY LARGE VISUAL DOCUMENT ARCHIVES

Giuseppe Amato
Giuseppe.amato@isti.cnr.it

Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
- Image classifications

Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
- Image classifications

Digital Data Explosion

- Everything we see, read, hear, write, and measure can now be in a digital form!
- In the next few years, we will create more data than has been produced in all of human history.
- Estimations:
  - More than 93% of produced data is digital
  - digital text is important – current technology is functional
  - multimedia, scientific, sensor, etc. is becoming prevalent

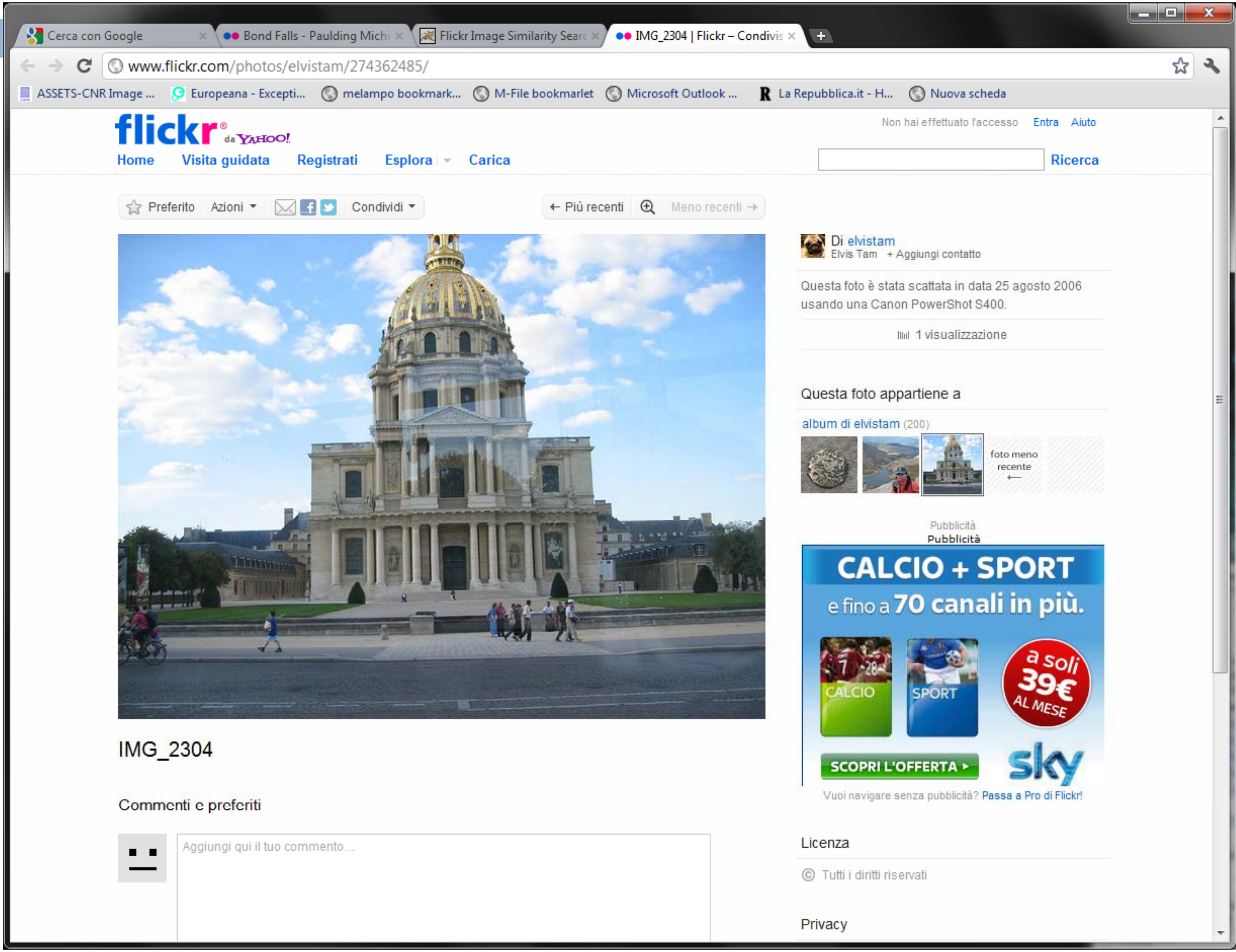
Digital Data Explosion

- The big issue is
  - “Searching” for visual documents
  - “Searching” using visual documents
- In many cases visual data is not associated with text or metadata so traditional search techniques cannot be used
- Twofold problem
  - Poorly annotated archives
  - Query about an unknown image content

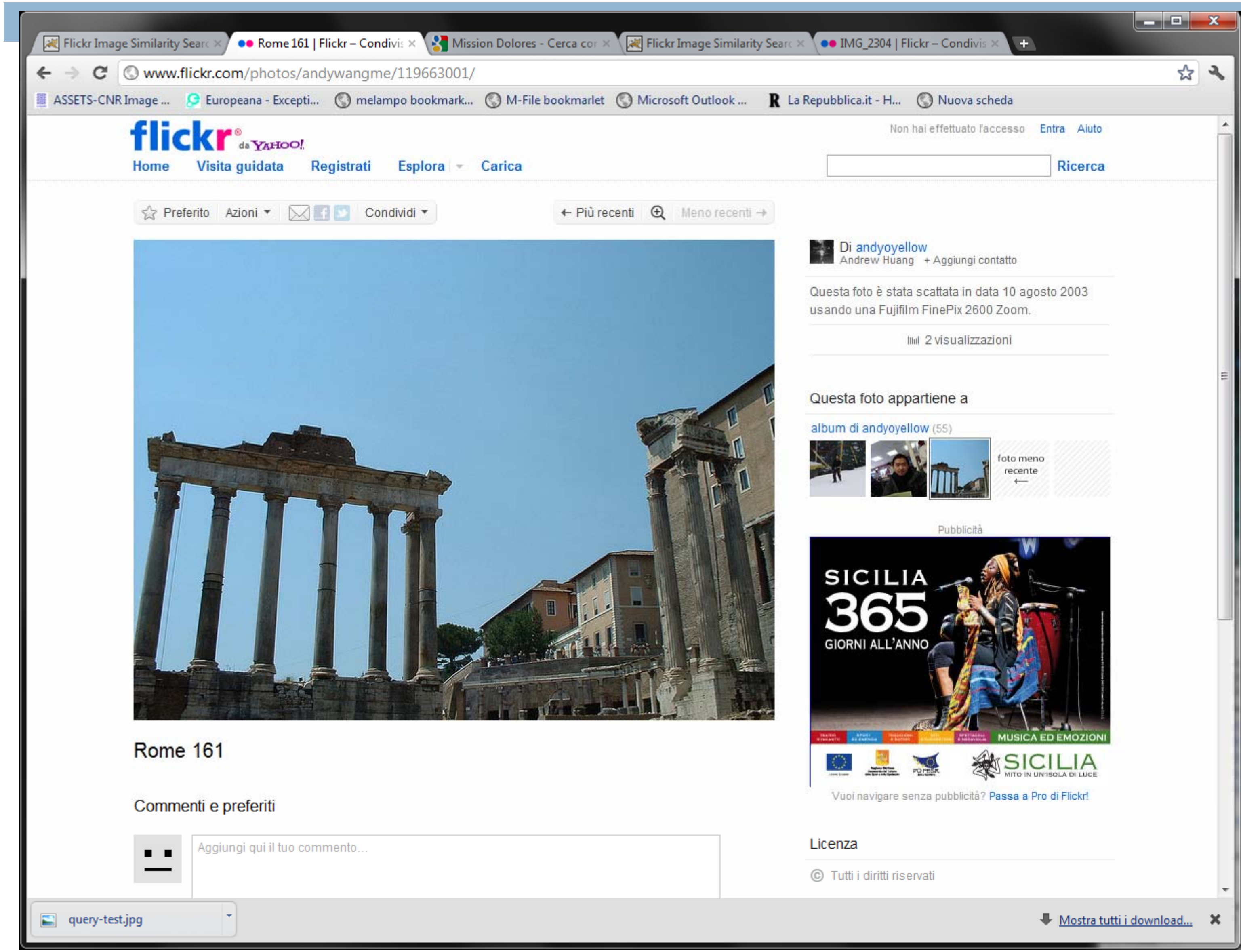
Poorly annotated archives

- A person produce about 1000-2000 untagged pictures per year
- It is becoming difficult to retrieve both images published on the web and stored on personal computers
- Flickr:
  - 30% images are not tagged and not commented
  - 90% images do not have comments

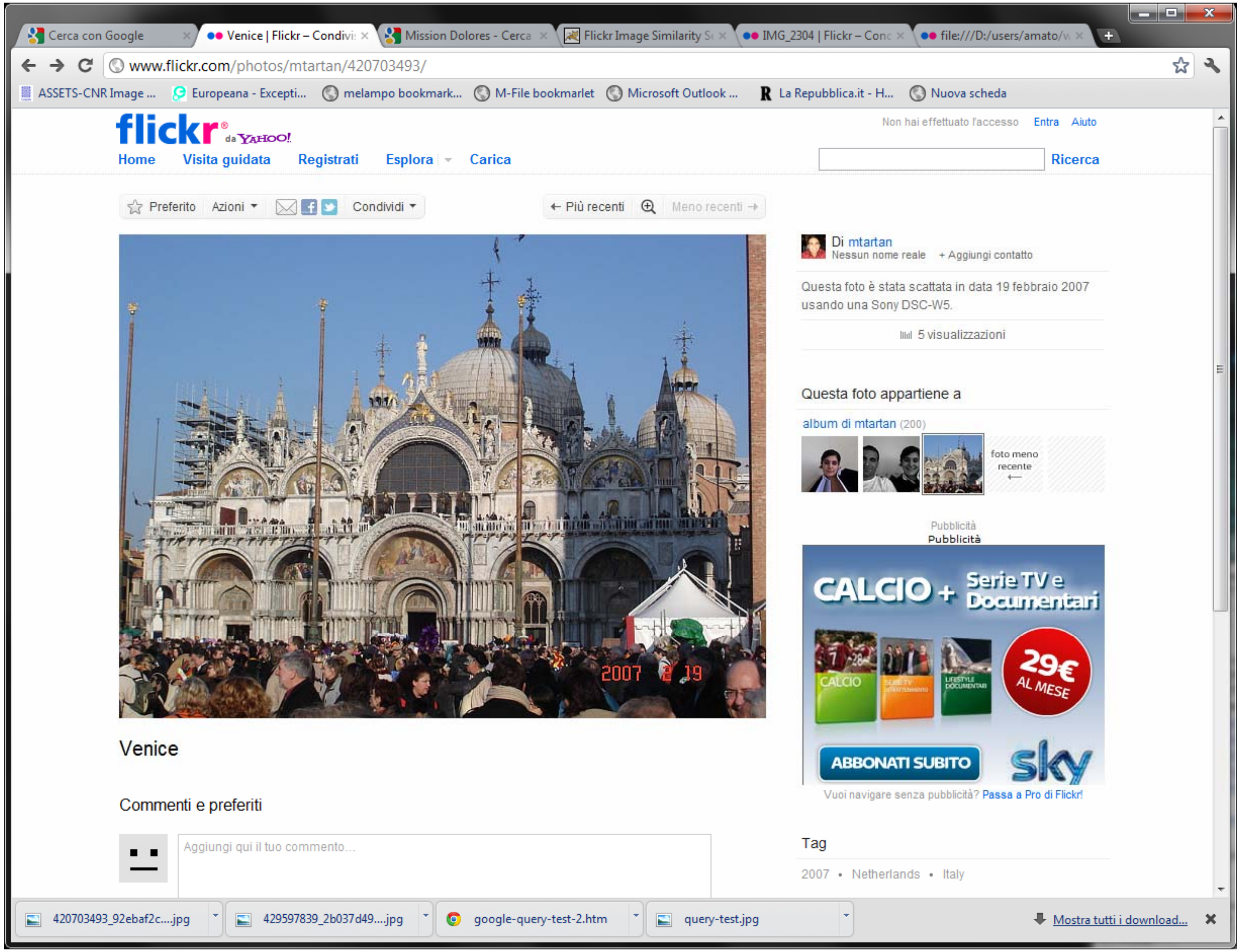
Poorly annotated archives



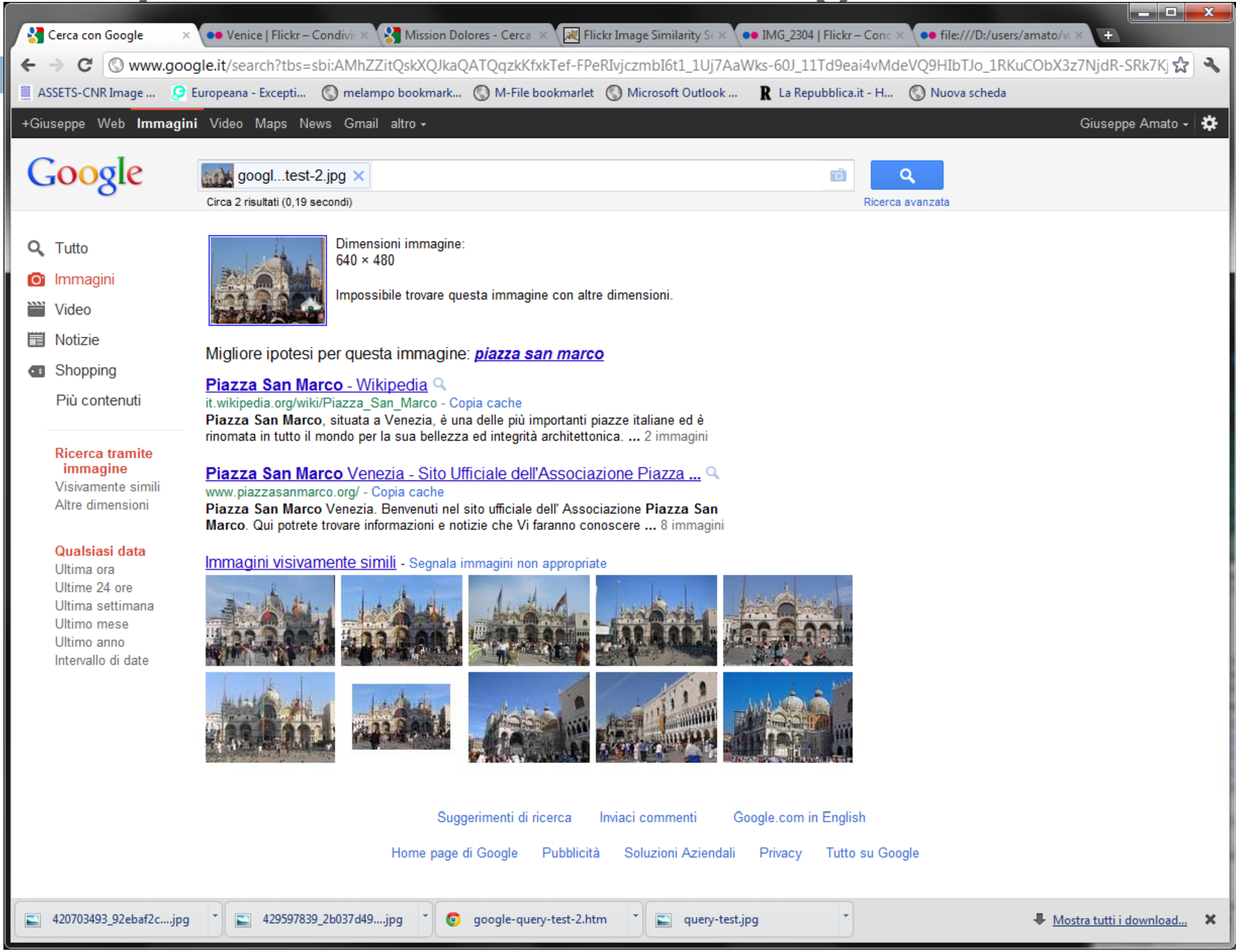
Poorly annotated archives



Poorly annotated archives



Query about unknown image content

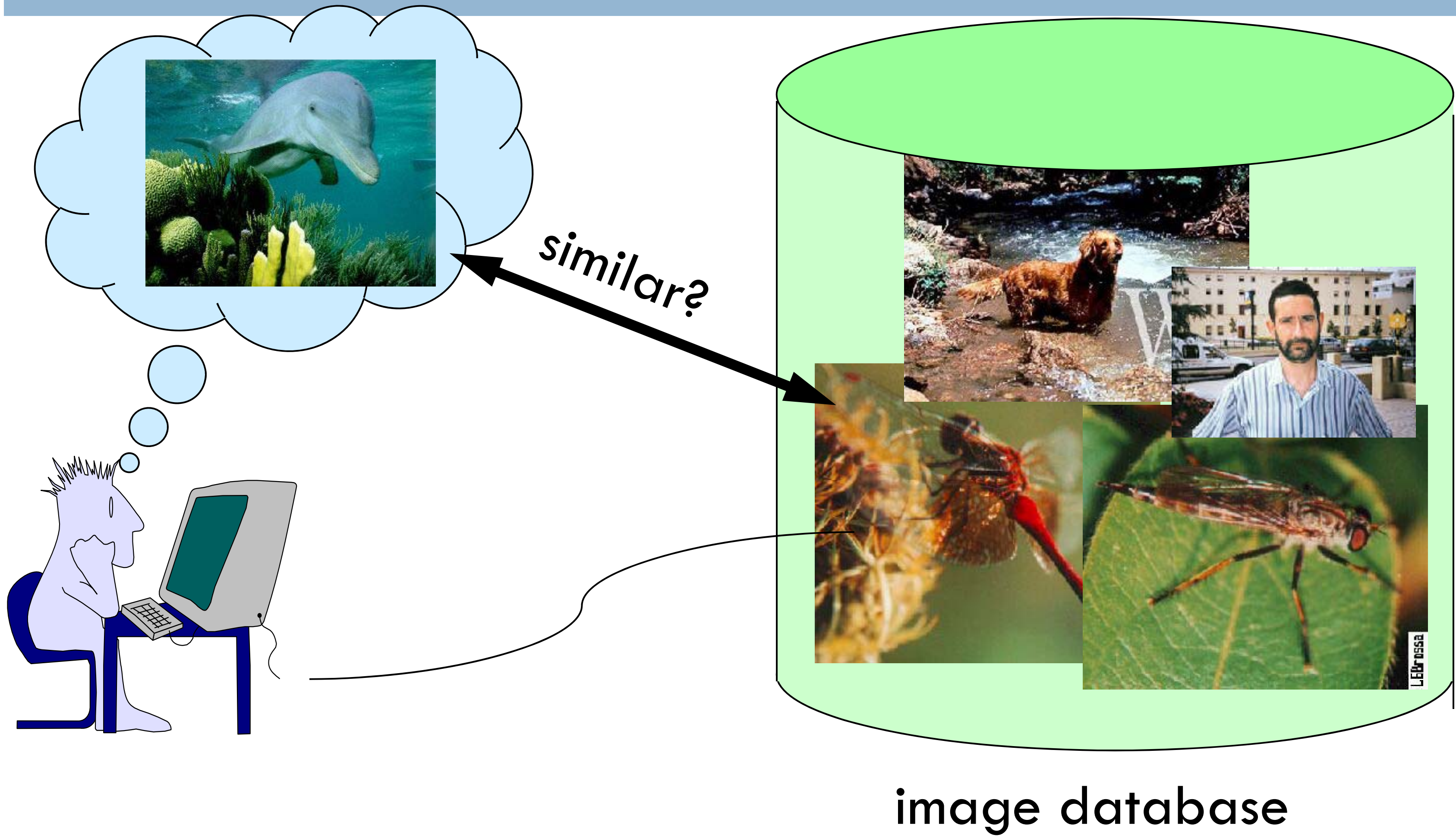


Change of the Search Paradigm

- Traditional YES-NO keyword search will not suffice - sortable domains of data (numbers, strings) are assumed

- New types of data need gradual comparison and/or ranking based on:
  - similarity,
  - dissimilarity,
  - proximity,
  - distance, closeness, etc.

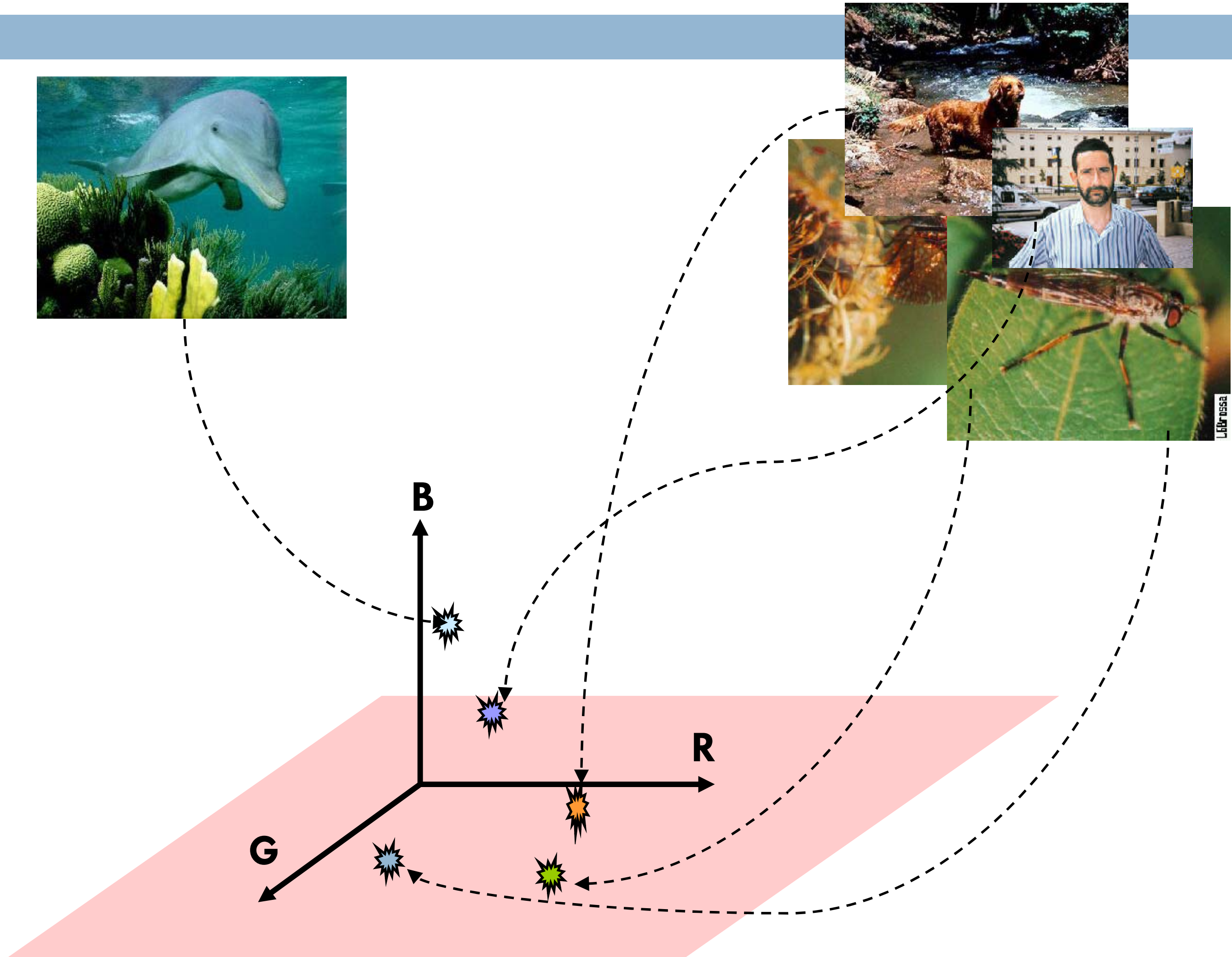
Image Similarity Search Problem



Feature-based Approach

image layer

feature layer



Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
- Image classifications

Outline

- The problem: examples and scenario
- Visual descriptors
  - Global features
  - Local features
- Similarity and approximate similarity searching on a large scale
- Image classifications

Global Features

- Global features
  - Matematical representation of the overall visual appearence of images
    - Colors
    - Textures
    - Shapes
    - …

Colors

- Color spaces: specifies how colors are represented
- Commonly used color spaces
  - RGB
  - HSV
  - YCrCb
  - HMMD
  - Arbitrary 3x3 color transformation matrix

Colors

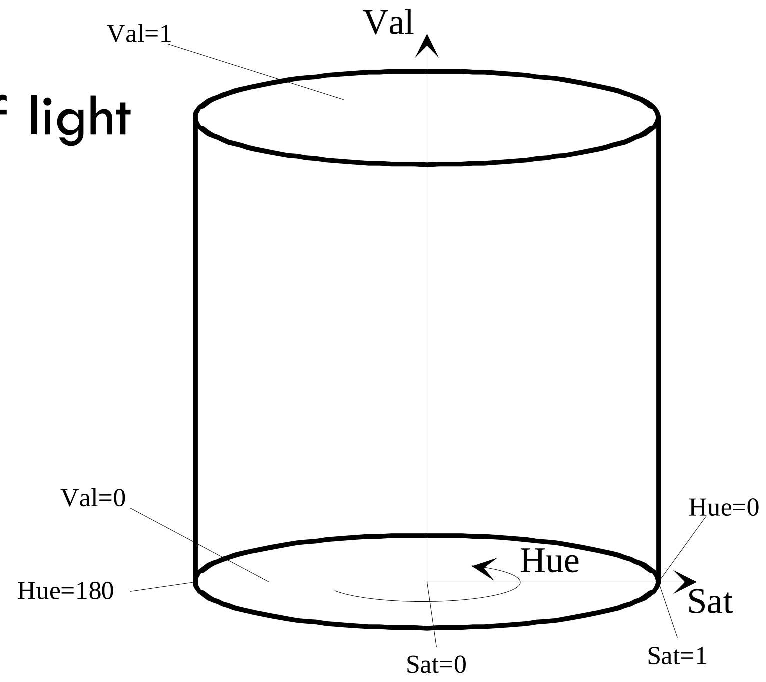
- Wanted properties of colour spaces:
  - Uniformity
    - Close colours are also perceived as similar
  - Completeness
    - All perceivable colours are represent able
  - Compactness
    - No redundancy

Colors

- The most common and intuitive colour space is the RGB (Red Green Blue) colour space
  - Every perceivable colour can be obtained as the sum of three degree of RGB
- However
  - Colours that are close in the RGB colour space might not be similar for the human perception

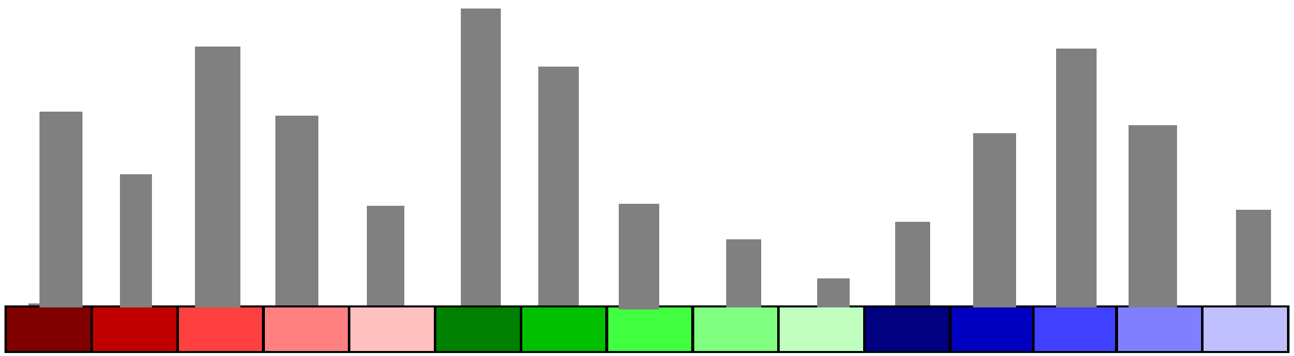
Colors

- HSV
  - HSV
    - Hue: Tint of the colour
    - Saturation: Quantity of colour
    - Value (Brightness): Quantity of light

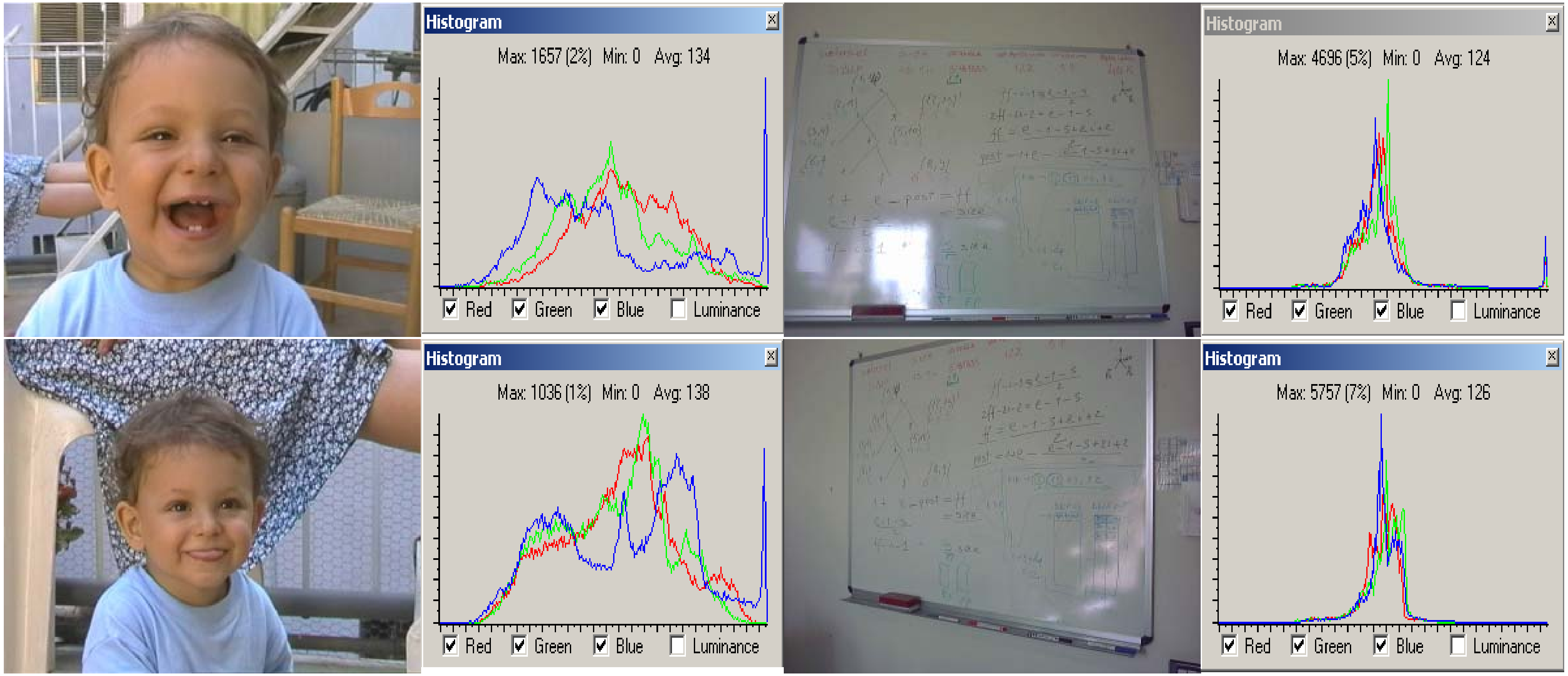


Hystograms of colors

- Colour histograms
  - The colour spectrum is divided into *n* bins
  - The value contained in each bean is proportional to the amount of pixel having colour of that bean



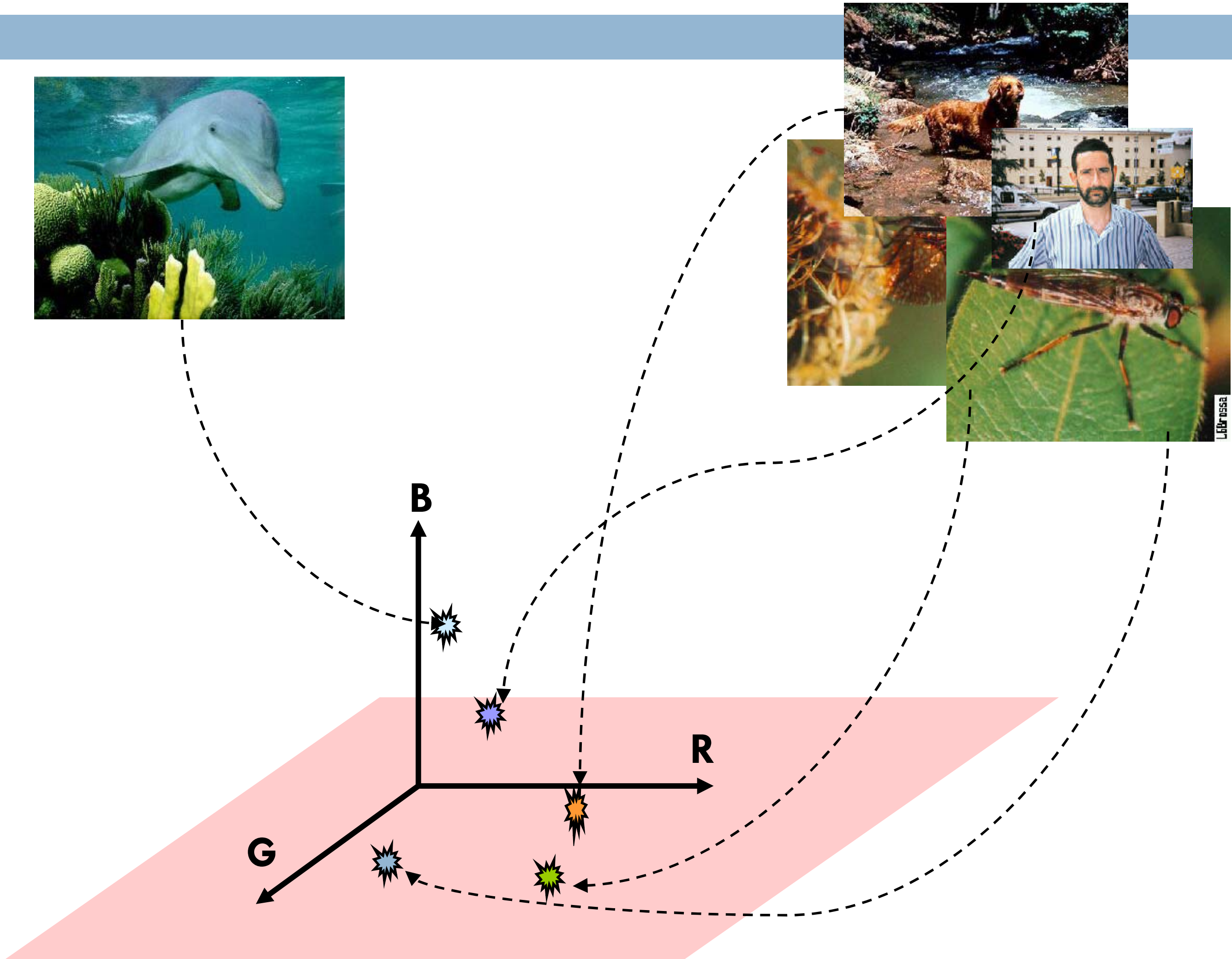
Histograms of colors



Feature-based Approach

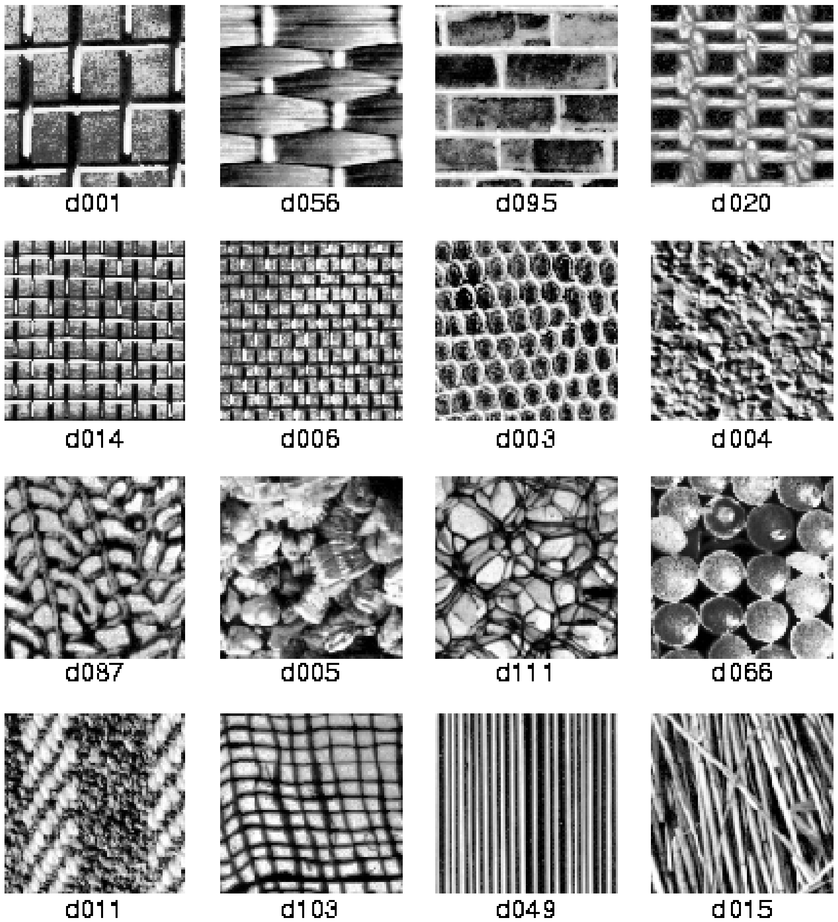
image layer

feature layer



Textures

- Homogeneous patterns
- Spatial arrangement of pixels
  - Colour is not enough to describe



Textures

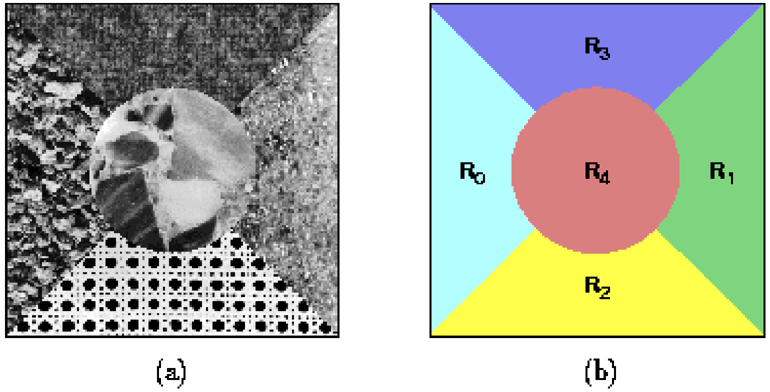
- Textures descriptions are obtained by using statistical methods
  - Spatial distribution of image intensity
  - Several methods exists
  - Texture descriptions can also be represented as histograms (vectors)

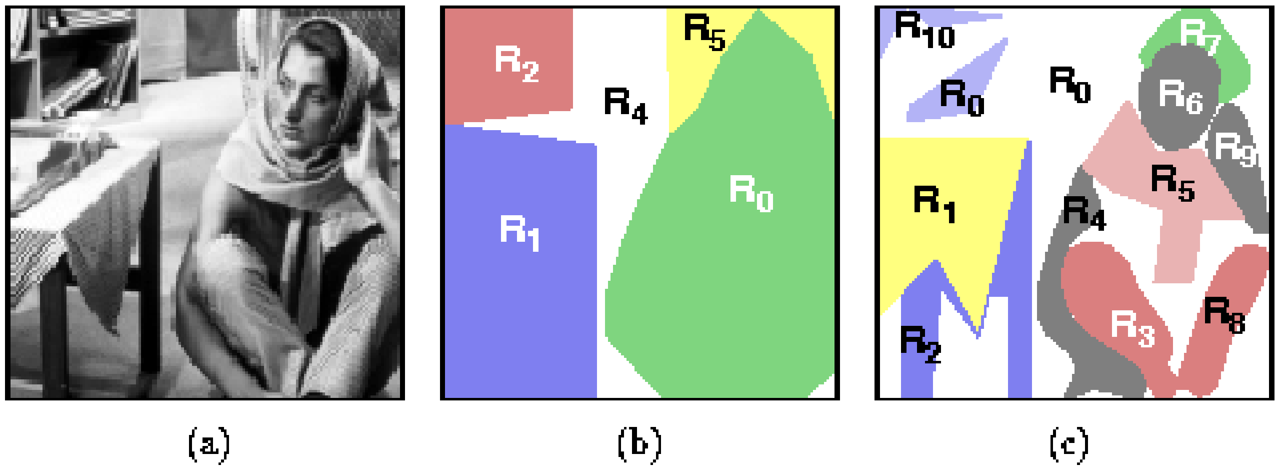
Textures

- Commonly used features for textures are the Tamura features:
  - Contrast
    - Distribution of pixel intensity
  - Coarseness
    - Granularity of a texture
  - Directionality
    - Dominant direction of the texture

Shapes and regions

- Shapes:
  - Segmentation
  - Feature extraction





MPEG-7 Visual descriptors

- Includes a standardization of some visual features
  - Scalable Colour
    - Color hystograms in HSV space, with applied Haar transform
  - Dominant Colour
    - Compact representation of colors in regions or images
  - Color Layout
    - Spatial distribution of colors
  - Color Structure
    - Hystograms with localized color distribution
  - Homogeneous Texture
    - Textures
  - Edge Histogram
    - Spatial distribution of edges in images
  - ContourShape and RegionShape
    - Description of contour of shapes and pixel distribution on a 2D shape

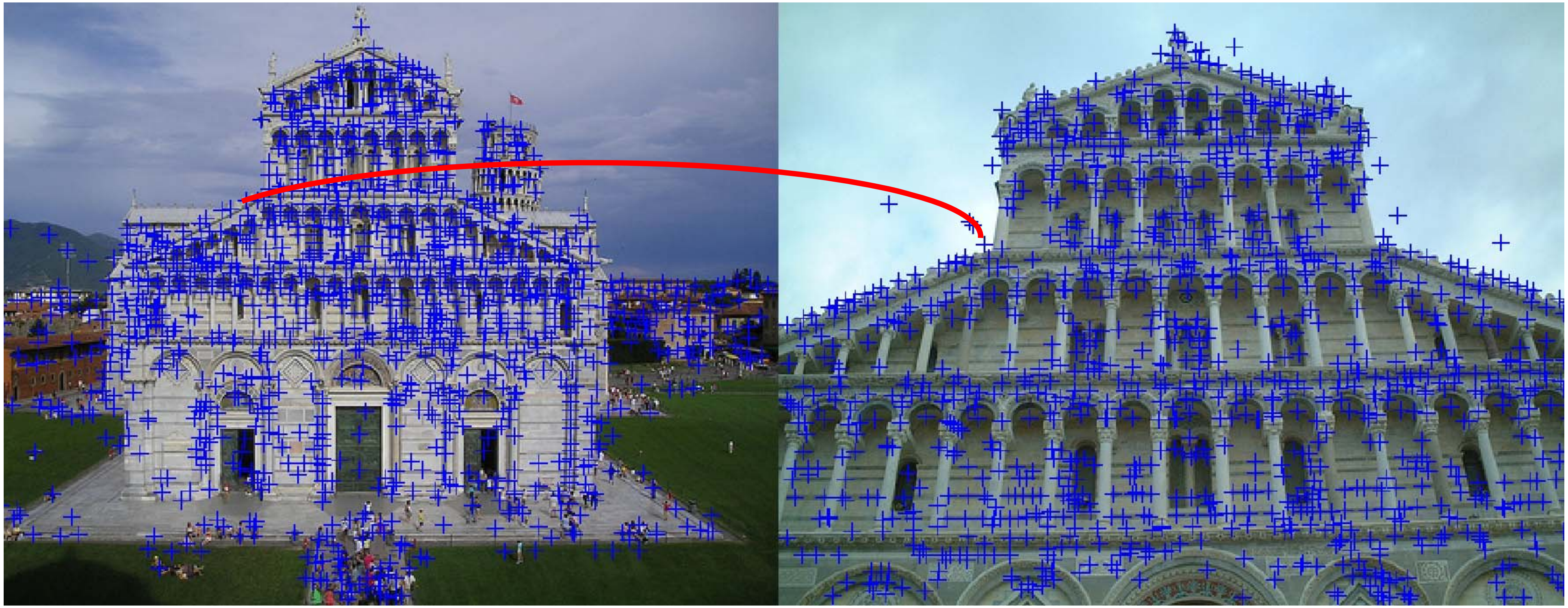
Outline

- The problem: examples and scenario
- Visual descriptors
  - Global features
  - Local features
- Similarity and approximate similarity searching on a large scale
- Image classifications

Local features

- Matematical representation of visual appearence of specific areas (interest point) of images
- Distance or (dis)similarity can be computed between interst points in images
- Images can be compared by analizing matching points

Local features



Local features

- Two phases
  - Keypoint detection
    - Uses strategies to decide which point in the image are interesting to be represented
  - Descriptor building
    - Build a descriptor (feature) for each interest point in the image
- Local features are typically much larger than global features
  - Each interest point has its descriptor

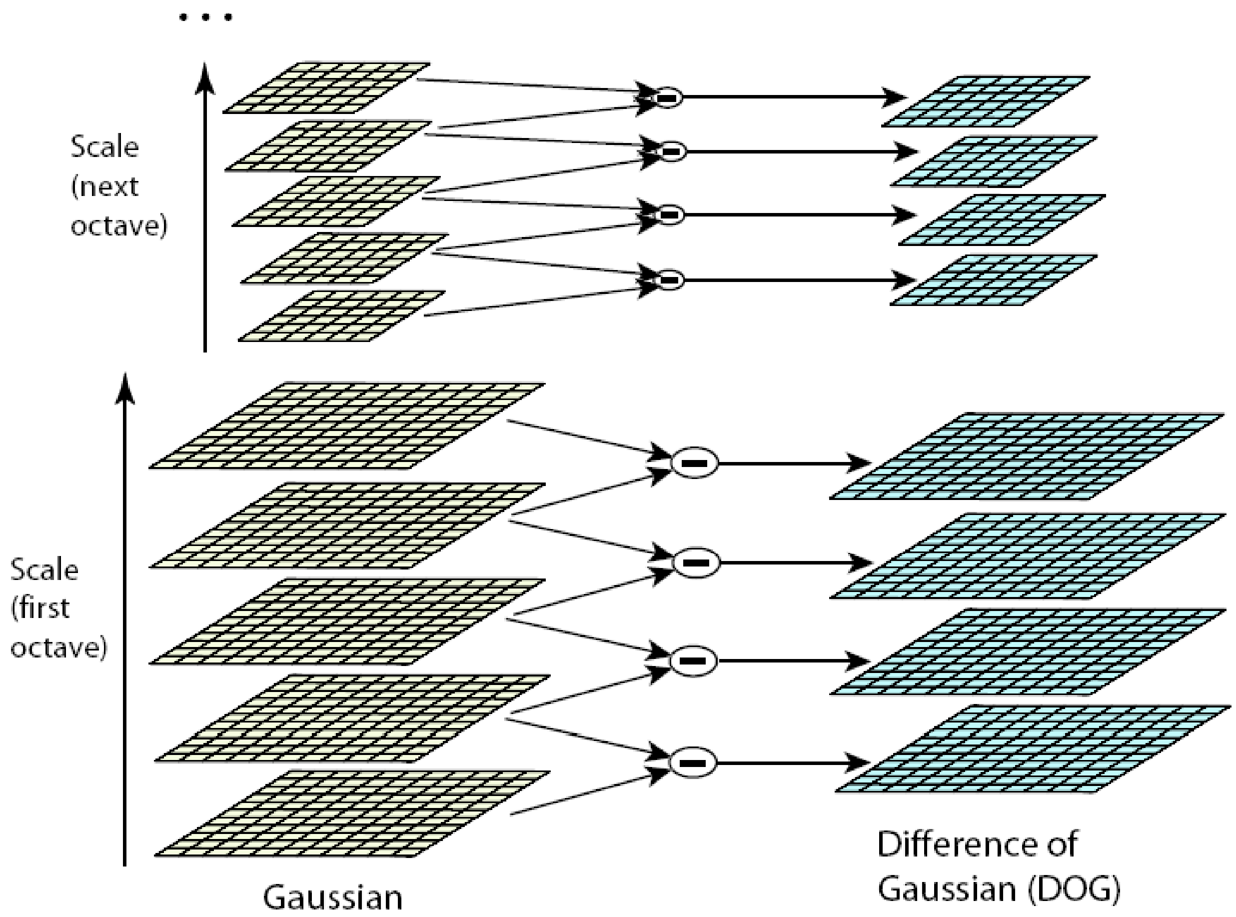
Local features

- A very popular local feature is SIFT and its variations
  - SIFT= Scale Invariant Feature Transform
  - Invariant to
    - scale
    - rotation
  - robust to
    - space changes in viewpoint
    - luminance variations

SIFT: Scale Invariant Feature Transf.

- Keypoint detection
  - Identify locations that are invariant to scale changes and that can be assigned under different view of an object
  - Difference of Gaussian (DoG) method
    - Repeatedly convolve images with Gaussian
    - Compute differences between adjacent Gaussian Images
    - After each octave (when σ is doubled) down sample image and repeat
    - Detection of maxima and minima between neighbours point across adjacent DoG images

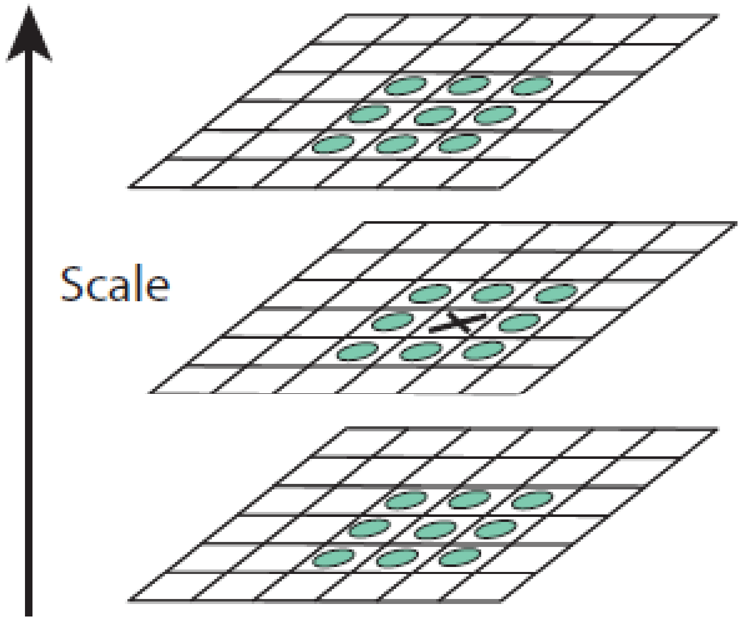
Convolve images and compute differences



Convolve images and compute differences

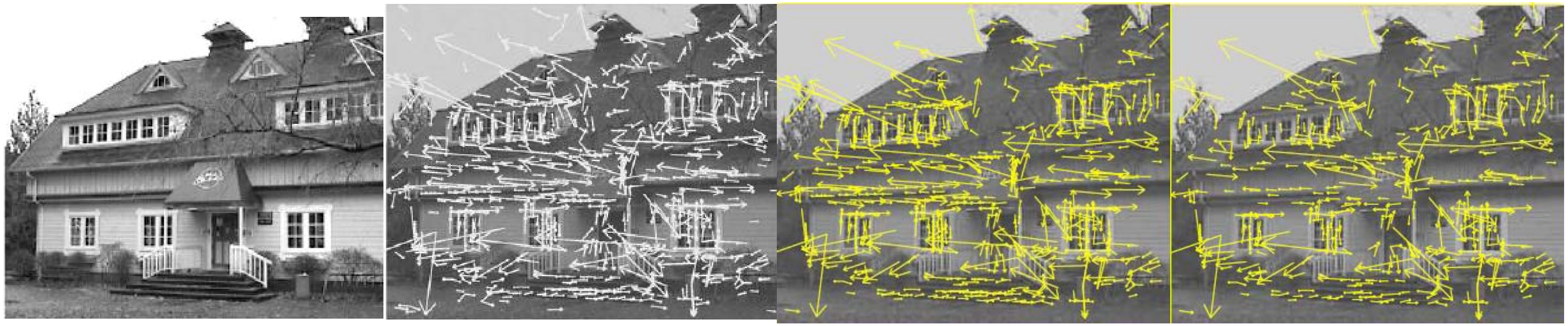
Detection of maxima and minima across adjacent DoG images

- Compare pixels of DoG with neighbors of
  - The same DoG
  - And adjacent DoG
- Select maxima and minima
- Take note of (x,y) positiond
- Take note of the scale



Filter keypoints

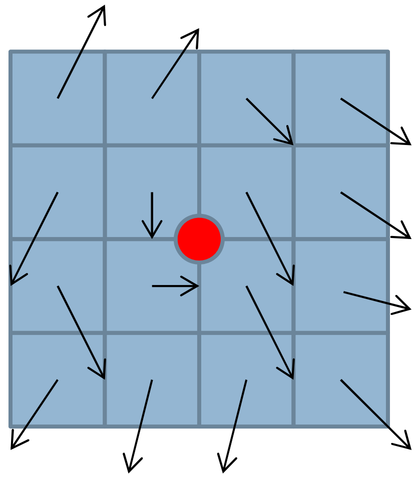
- Detected keypoint can be filtered to eliminate
  - Keypoint with low contrast
  - Edge response elimination

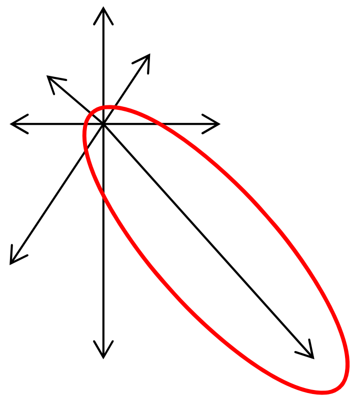


Original Image | Detected keypoints | Filter on contrast | Eliminate edge response

Orientation assignment

- To obtain rotation invariant features all key points are associated with an orientation:
  - Build an histogram of luminance gradients around the key point and take the main direction:





Local image descriptor

- Create an array of orientation histograms:

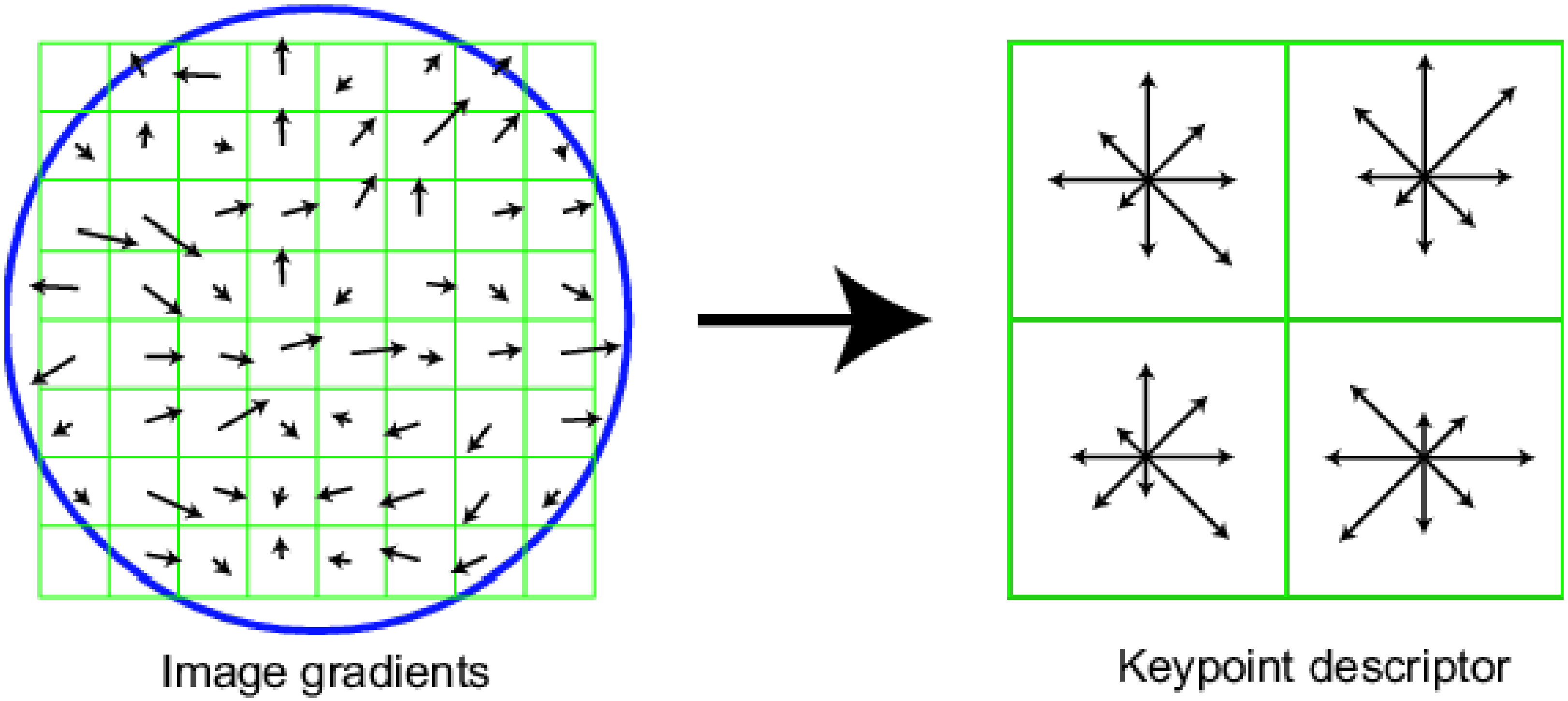
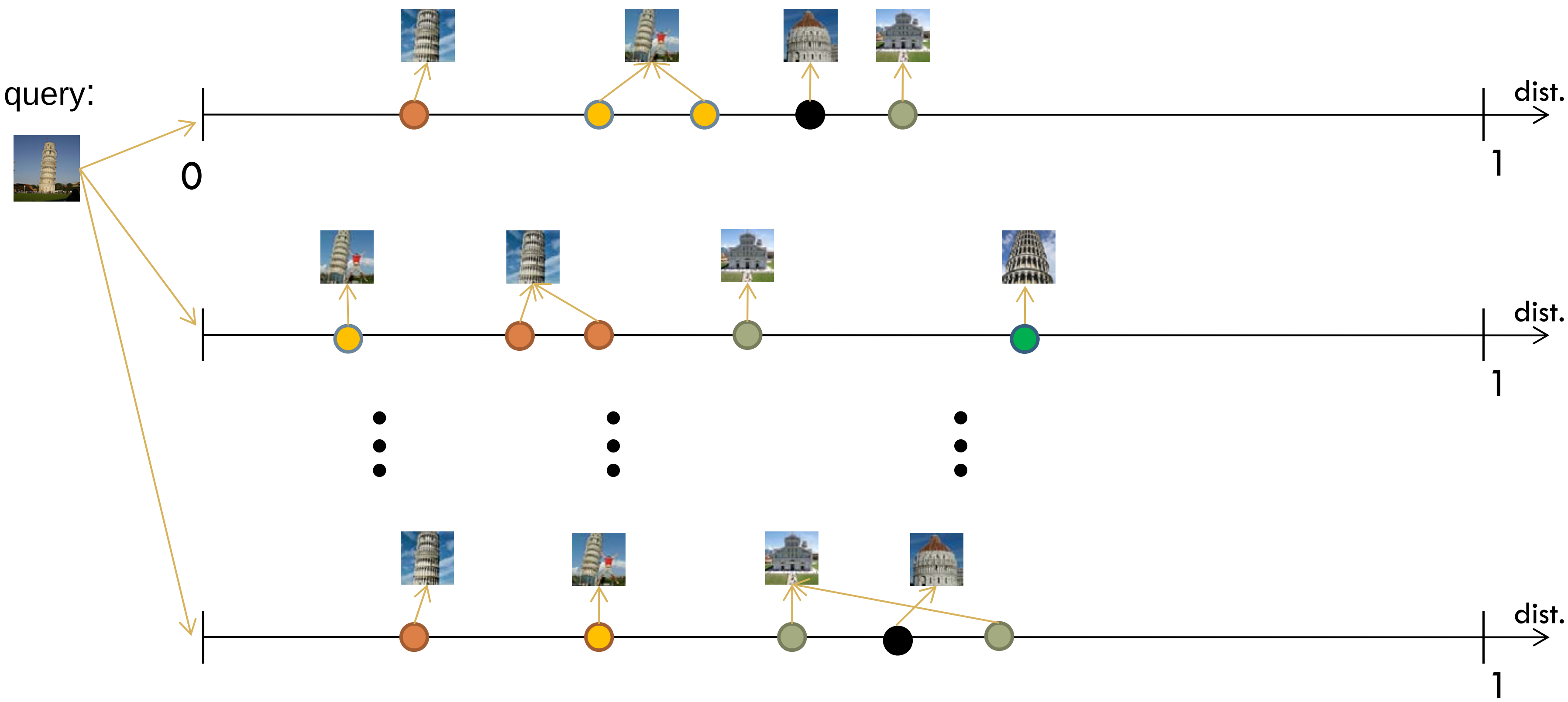


Image match

- Local feature were initially proposed for object recognition in computer vision
- They can also be used for image matching
  - Compare local image descriptors by using Euclidian distance to find matching pairs
    - E.g. serching for the kNN local image desctiptors
  - Check that there is consistence among matching pairs for what concerns
    - Scale
    - Orientation
  - Check that matching point are geometrically consistent

LF Similarity Searching Approach

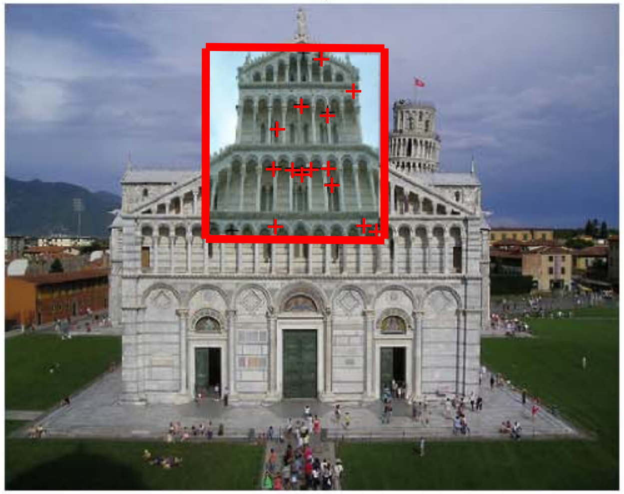


Geometric consistency check

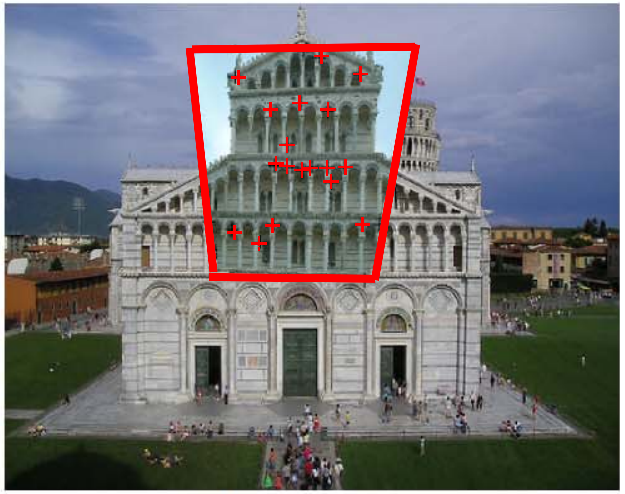
Image 1

Image 2

Rotation, Scale and Translation



Affine

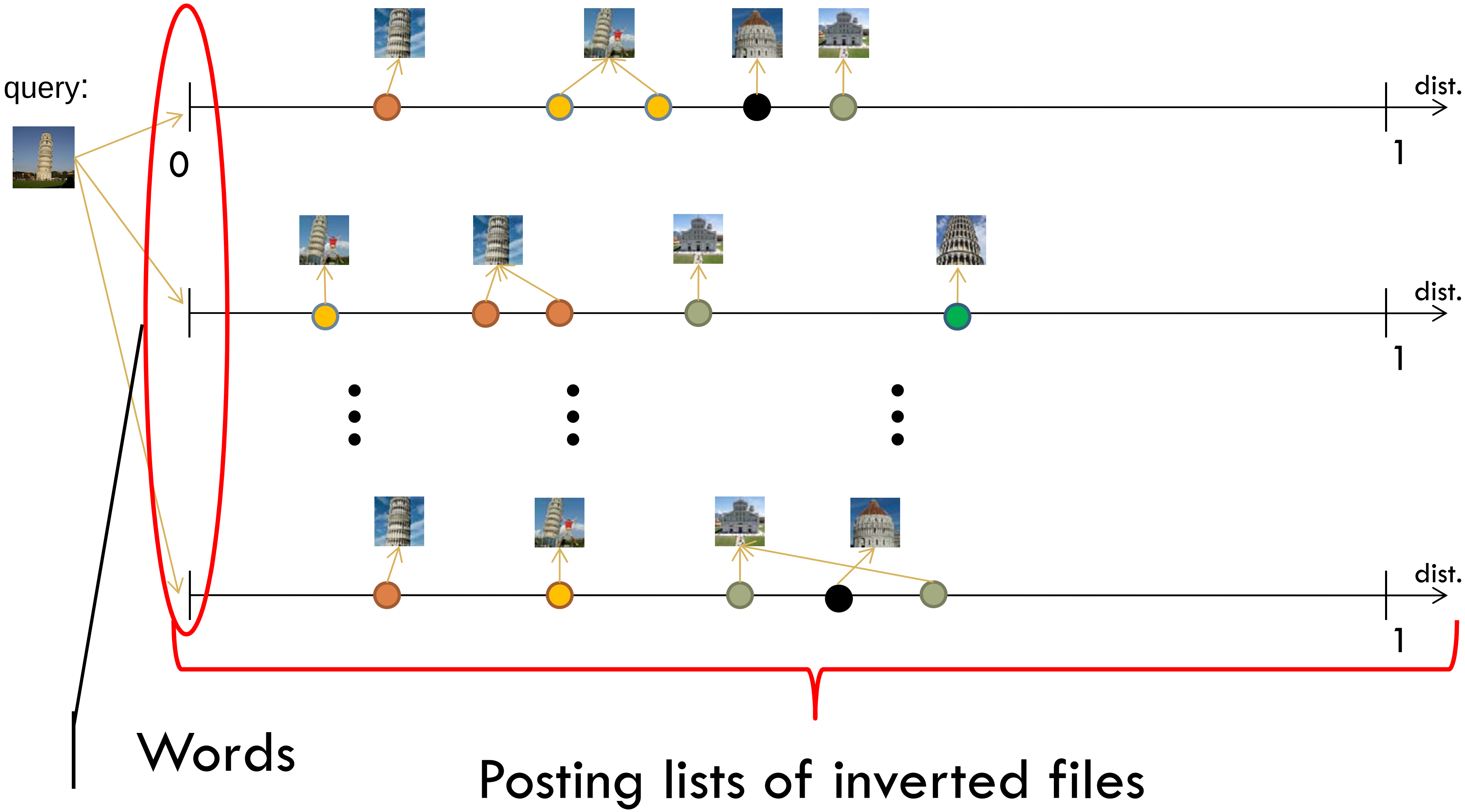


Homography

Efficient approaches to local features

- Using Bag of features (BoF) approach
  - Local features are quantized and images are represented as bag of visual words
  - Inverted files tecniques are used
- Fast similarity search techniques are applied to match local features
- Global similarity functions are defined on top of local features
- More investigation is needed in this direction

BoF approach: each local feature is associated to a word



Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
- Image classifications

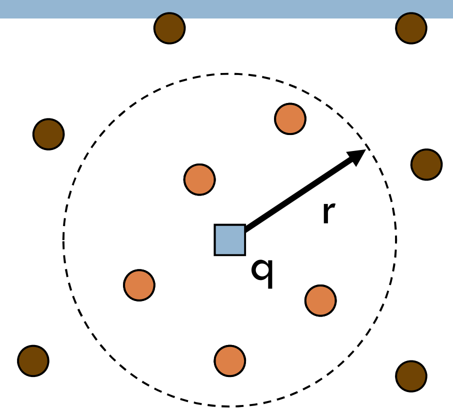
Fast similarity search

- Problem we want to discuss:
  - How to efficiently search for the most similar elements to a query element
    - In databases containing millions of elements
    - Obtaining response in fraction of seconds or a few seconds at most
    - Techniques applicable to any distance function
      - With some specific properties (e.g. any metric function)

Metric space

- M = (D,d)
  - Data domain D
  - Total (distance) function d: D × D → R (metric function or metric)
- The metric space postulates:
  - Non negativity $\forall x, y \in \mathcal{D}, d(x, y) \geq 0$
  - Symmetry $\forall x, y \in \mathcal{D}, d(x, y) = d(y, x)$
  - Identity $\forall x, y \in \mathcal{D}, x = y \Leftrightarrow d(x, y) = 0$
  - Triangle inequality $\forall x, y, z \in \mathcal{D}, d(x, z) \leq d(x, y) + d(y, z)$

Similarity Range Query

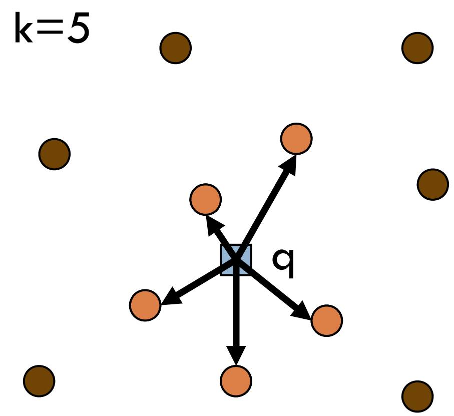


- range query
  - $R(q,r) = \{ x \in X \mid d(q,x) \le r \}$

- … all images having distance from the query smaller than …

Nearest Neighbor Query

- the nearest neighbor query
  - NN(q) = x
  - $x \in X$, "$y \in X$, $d(q,x) \leq d(q,y)$
- k-nearest neighbor query
  - k-NN(q,k) = A
  - $A \subseteq X$, $|A| = k$
  - $\forall x \in A$, $y \in X - A$, $d(q,x) \leq d(q,y)$



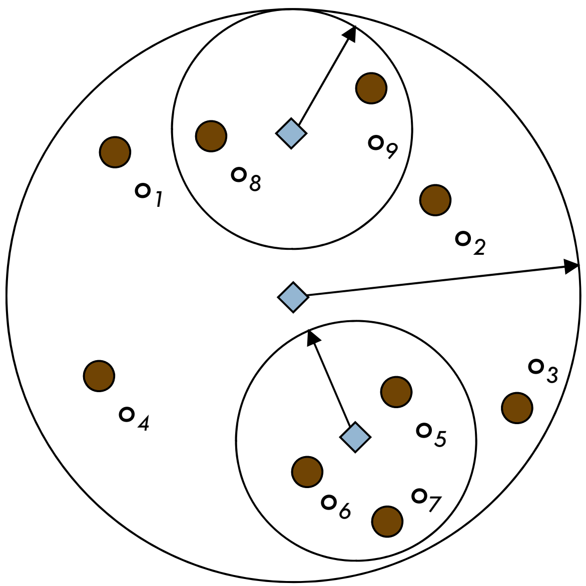
- … five closest images to this one…

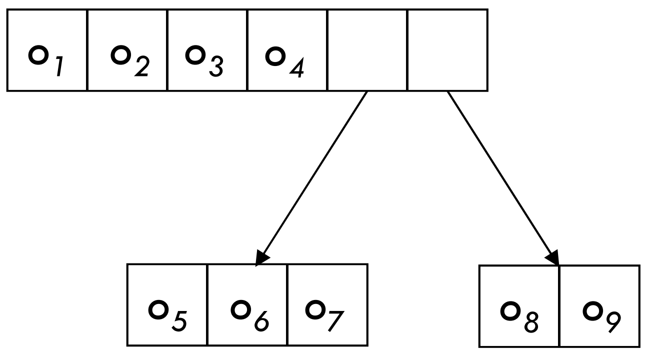
Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
  - Tree based approaches
  - LSH
  - Permutations
  - MI-File
- Image classifications

Tree based index organization

- Hierarchical decomposition of the space
- Using ball regions
  - Root node organizes some objects and some regions.
  - Internal nodes also organize other object and regions

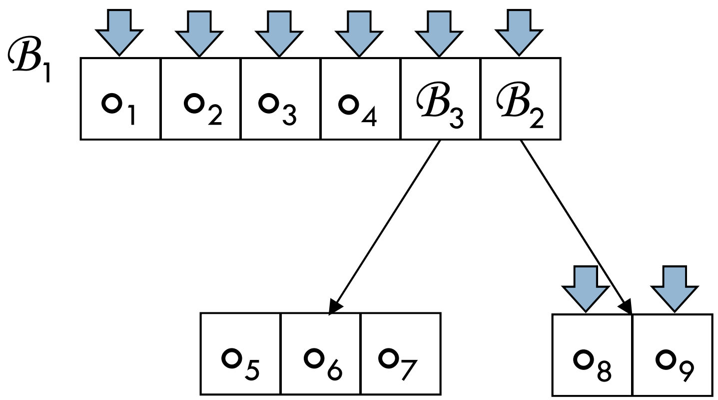


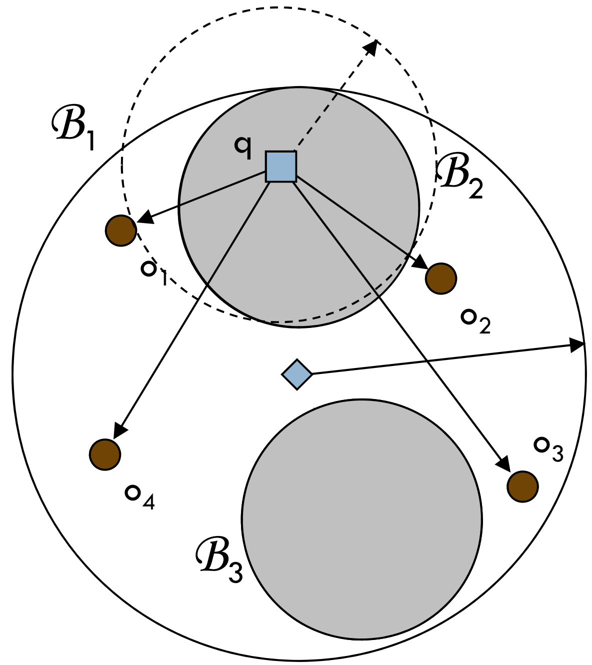


Range Search Algorithm with hierarchical ball partitioning

R(q,r):

- Start inspecting elements in $\mathcal{B}_1$.
- $\mathcal{B}_3$ is not intersected.
- Inspect elements in $\mathcal{B}_2$.
- Search is complete.





Response = o_8, o_9

Principles of Approx. Similarity Search

- Approximate similarity search over-comes problems of exact similarity search when using traditional access methods.
  - Moderate improvement of performance with respect to the sequential scan.
  - Dimensionality curse
- Similarity search returns mathematically precise result sets.
  - Similarity is often a formalisation of an non-objective measure, so in some cases also approximate result sets satisfy the user's needs.

Principles of Approx. Similarity Search

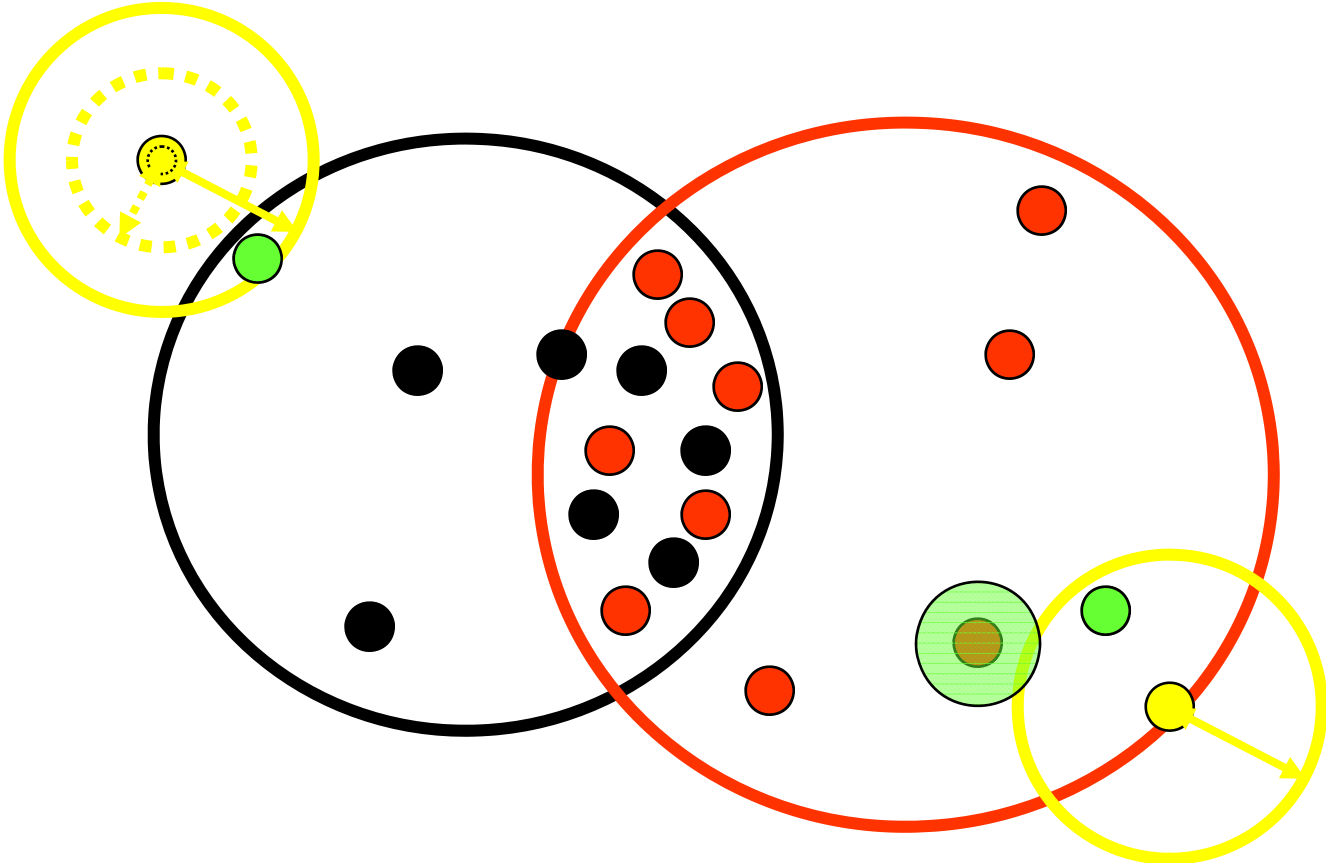
- Approximate similarity search processes a query faster at the price of imprecision in the returned result sets.
  - Useful, for instance, in interactive systems:
    - Similarity search is typically an iterative process
    - Users submit several search queries before being satisfied
      - Fast approximate similarity search in intermediate queries can be useful.
  - Improvements up to two orders of magnitude

Approx. Similarity Search: Basic Strategies

- Space transformation
  - Data are managed in an easier space representation

- Reducing subsets of data to be examined
  - Not promising data is not accessed.
  - False dismissals can occur.

Approx. Similarity Search: Basic Strategies



Similarity Search
The Metric Space Approach

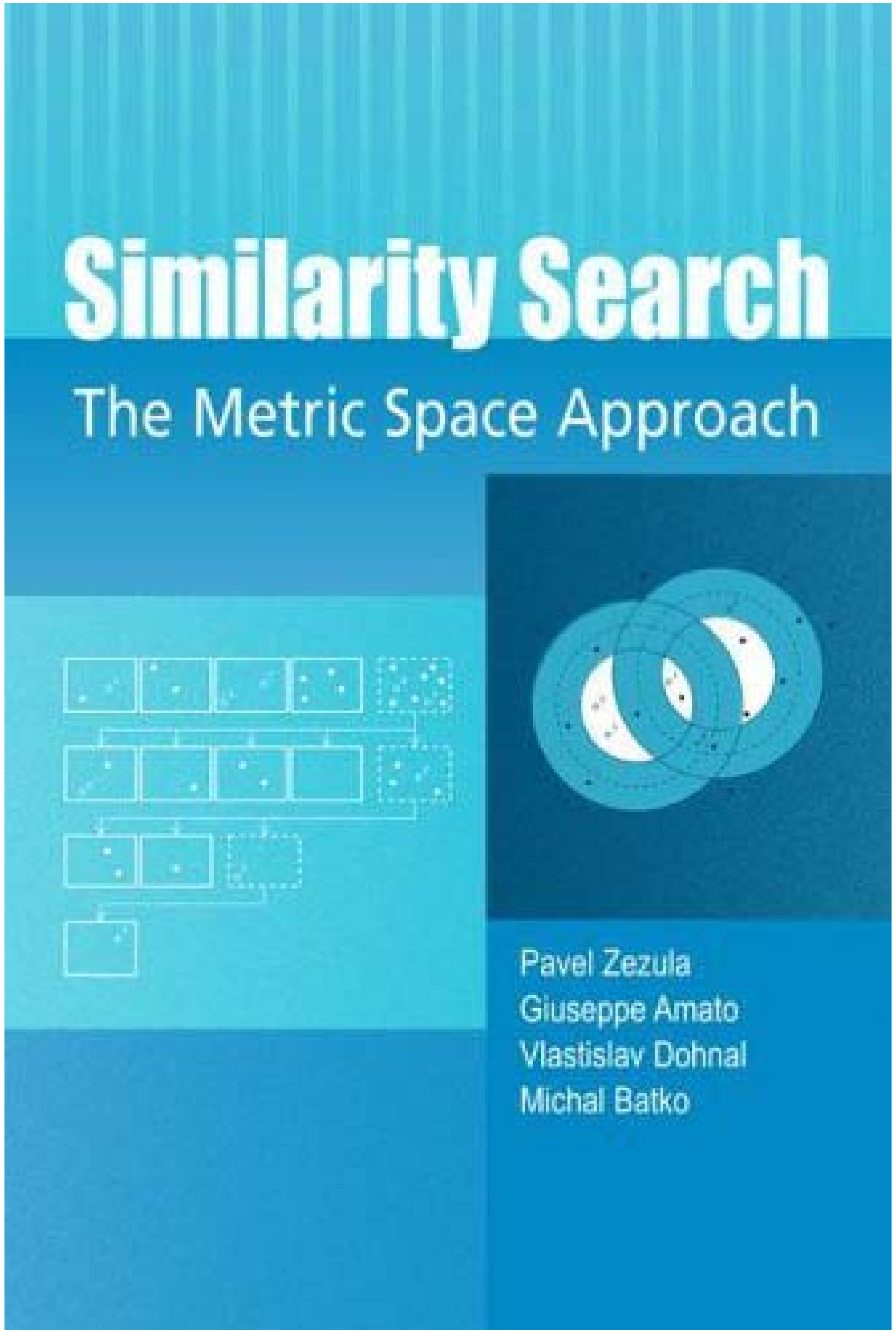
Pavel Zezula

Giuseppe Amato

Vlastislav Dohnal

Michal Batko

Published by Springer



Outline

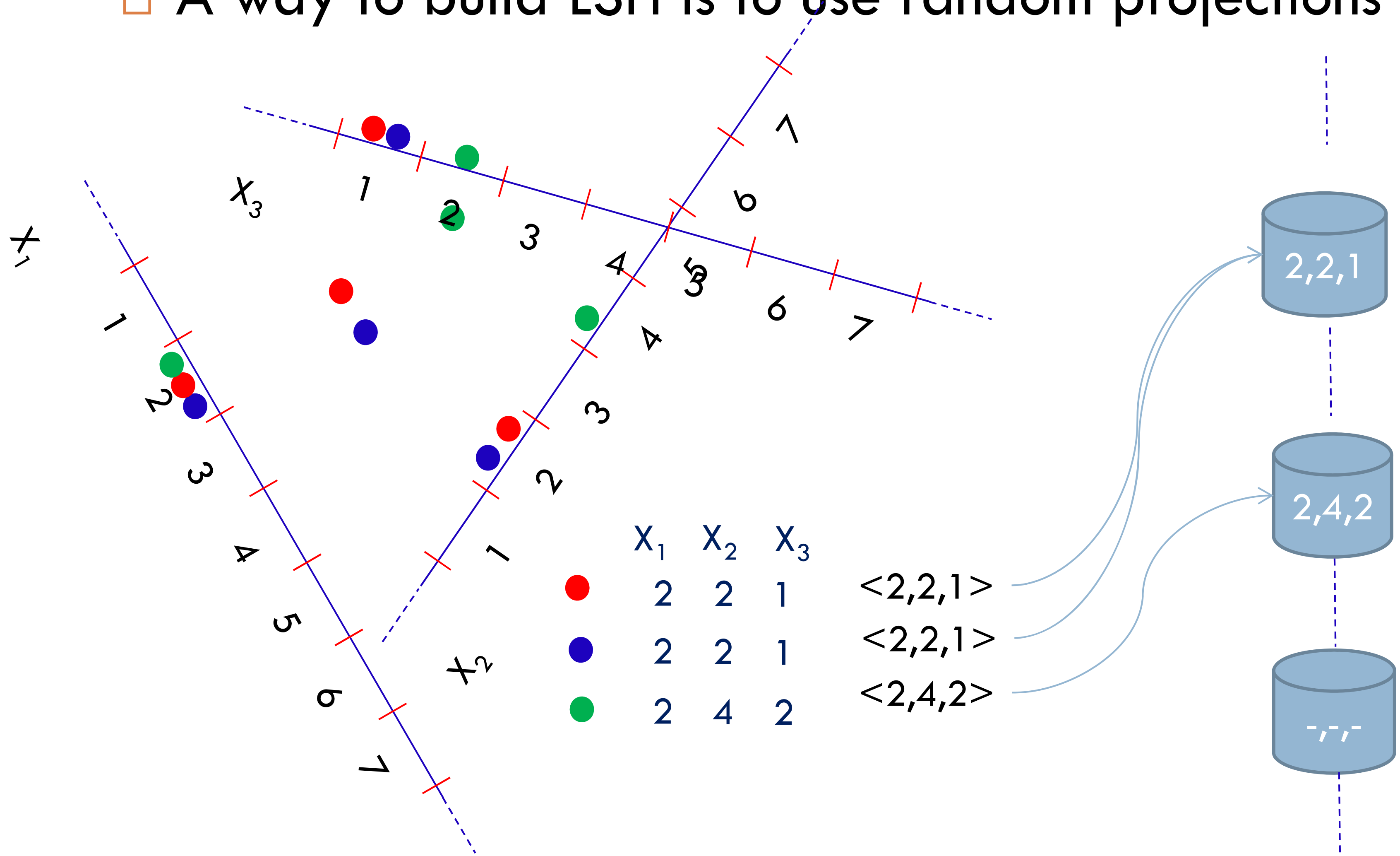
- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
  - Tree based approaches
  - LSH
  - Permutations
  - MI-File
- Image classifications

LSH – Locality Sensitive Hashing

- Basic idea
  - Build various hash functions g() such that given any two objects p,q
    - If d(p,q) > c*r then Pr[g(p)=g(q)] is small
    - If d(p,q) ≤ r then Pr[g(p)=g(q)] is not so small
  - In other words the hash function g()
    - assigns similar objects to the same bucket with good probability
    - Very dissimilar objects goes to the same bucket with small probability

Projection based LSH

- A way to build LSH is to use random projections



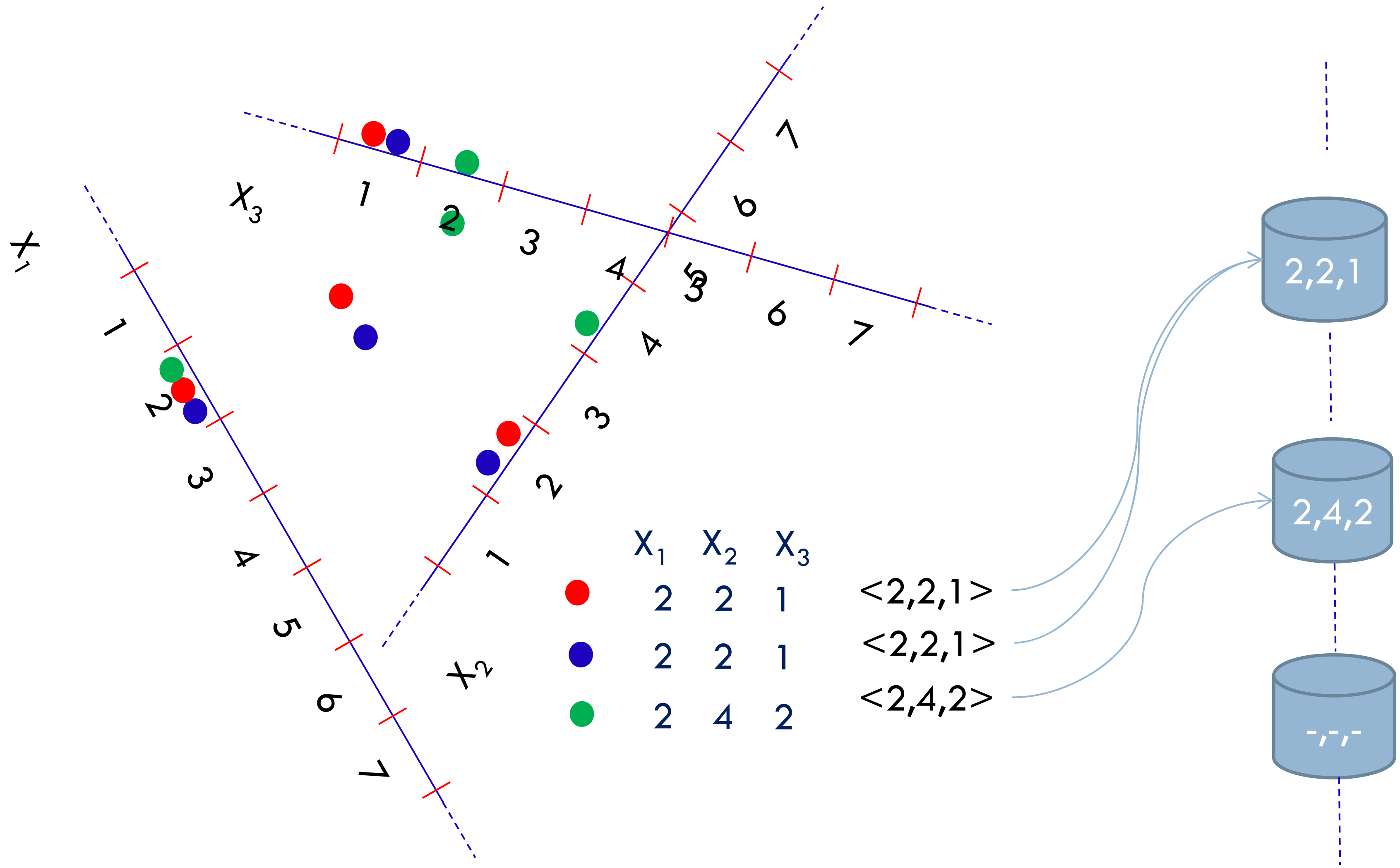
LSH more formally

- Hash functions g() are defined as
  - $g(p)=<h_1(p), h_2(p),\ldots, h_k(p),>$
- Where
  - $h_i(p)=((p*X_i+b_i)/w)$
- and
  - w is the size of the segments in the projection vectors
  - $X_i=(x_{i,1},\ldots,x_{i,d})$ is a vector where $x_{i,j}$ is chosen from
    - a Gaussian distribution (for l_2 norm)
    - a s-stable distribution (for l_s norm)
  - b_i is a random scalar value

LSH pre-processing and query execution

- Preprocessing
  - Select various hash functions $g_1, \ldots, g_L$
- Insertion
  - Any point p of the dataset is hashed and inserted into L buckets $g_1(p), \ldots, g_L(p)$
- Query execution
  - Given a query q retrieve all point from the L buckets $g_1(q), \ldots, g_L(q)$ and reorder according to the original distance function

Query execution: access all objects in the buckets of the query



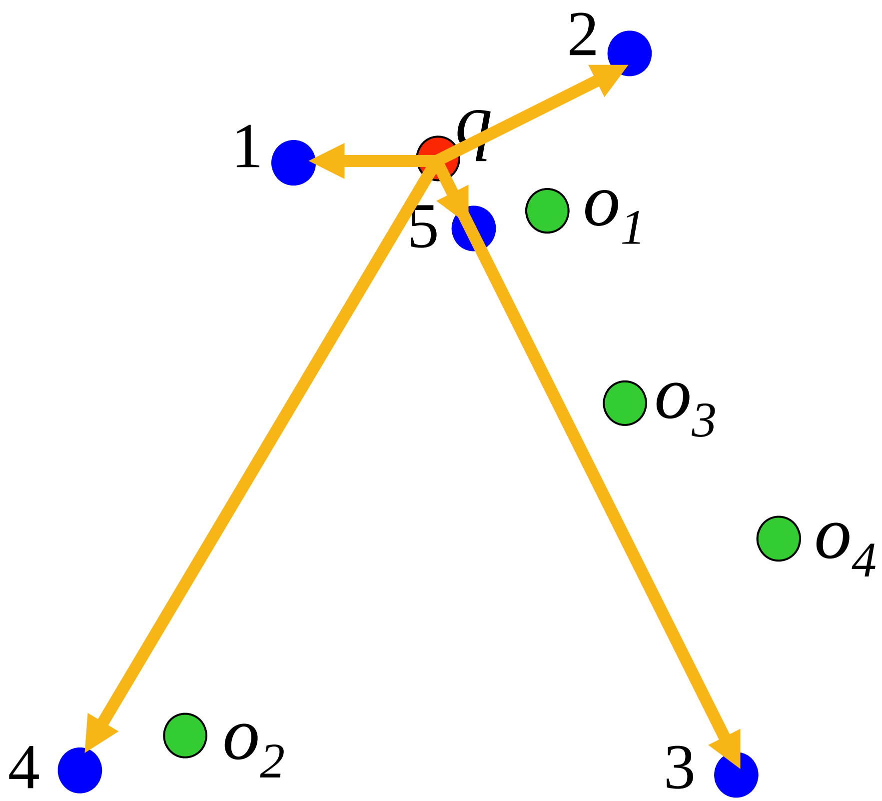
Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
  - Tree based approaches
  - LSH
  - Permutations
  - MI-File
- Image classifications

Permutation based indexes

- Objects that are close one to each other see the space in a "similar" way
- Suppose we chose a set of reference objects RO
- Permutations of RO ordered according to the distances from two similar data objects are similar as well
  - We can represent every data object o as an ordering of RO according to the distance from o
  - We can measure the similarity between two data objects by measuring the similarity between the corresponding orderings

Permutation based transformation



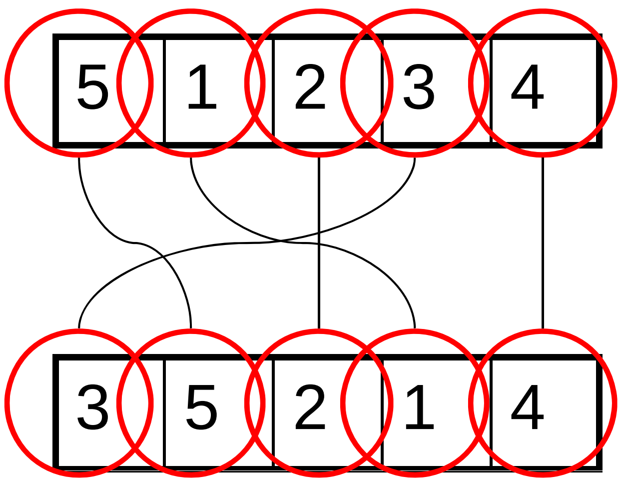
- Mapped objects:

- q = (5,1,2,3,4)
- O1 = (5,2,1,3,4)
- O2 = (4,3,5,1,2)
- O3 = (5,2,3,1,4)
- O4 = (3,5,2,1,4)

Distance in the permuation space

- The Spearman Footrule distance:

$$SFD(S_1, S_2) = \sum_{ro \in RO} \left| S_1(ro) - S_2(ro) \right|$$



1 +
2 +
0 +
3 +
0 =

6

Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
  - Tree based approaches
  - LSH
  - Permutations
  - MI-File
- Image classifications

Fast similarity search on permutations: MI-File: Metric Inverted File

- We first perform a space transformation into a suitable metric space (perspective based transformation, or permutation space transformation)
- We show that inverted files can be used in the transformed space to rank objects by similarity
- We propose and analyse some optimisation/relaxation that allow similarity search to be executed very efficiently and with very little approximation
- Result is approximate:
  - Approximate similarity ranking can be different from exact similarity ranking

Fast similarity search on permutations: MI-File: Metric Inverted File

- O1 = (5,2,1,3,4)
- O2 = (4,3,5,1,2)
- O3 = (5,2,3,1,4)
- O4 = (3,5,2,1,4)

Entries

Posting lists

1	3,O1	4,O2	4,O3	4,O4
2	2,O1	2,O3	3,O4	5,O2
3	1,O4	2,O2	3,O3	4,O1
4	1,O2	5,O1	5,O3	5,O4
5	1,O1	2,O3	2,O4	3,O2

Using inverted files: query execution

Spearman footrule distance can be computed incrementally!

1	2
2	3
3	4
4	5
5	1

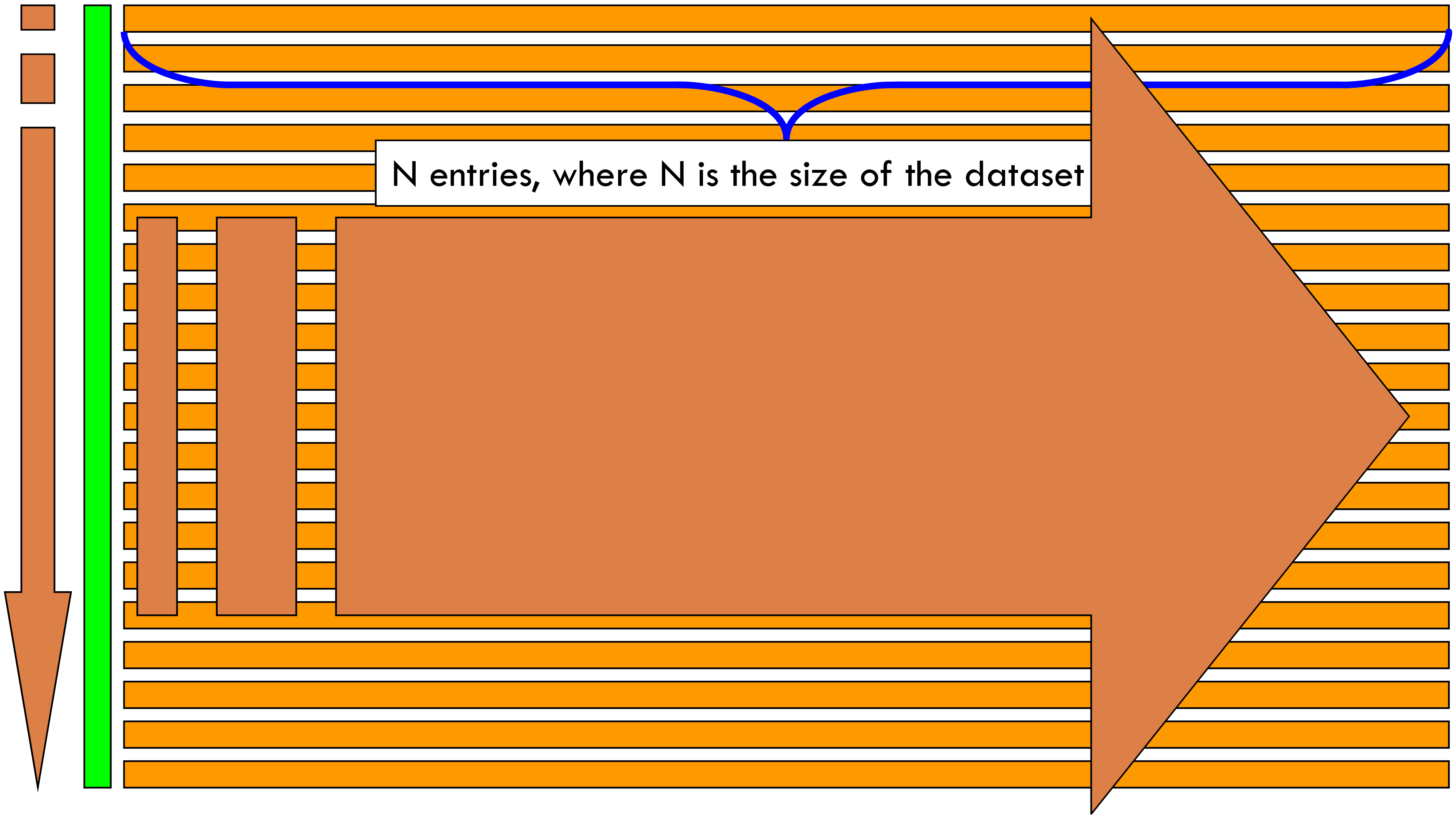
q

1	3,O1	4,O2	4,O3	4,O4
2	2,O1	2,O3	3,O4	5,O2
3	1,O4	2,O2	3,O3	4,O1
4	1,O2	5,O1	5,O3	5,O4
5	1,O1	1,O3	2,O4	3,O2

	O1	**O2**	**O3**	**O4**
	1+	2+	2+	2+
	1+	2+	1+	0+
	0+	2+	1+	3+
	0+	4+	0+	0+
	0=	2=	0=	1=
SFD:	2	12	4	6

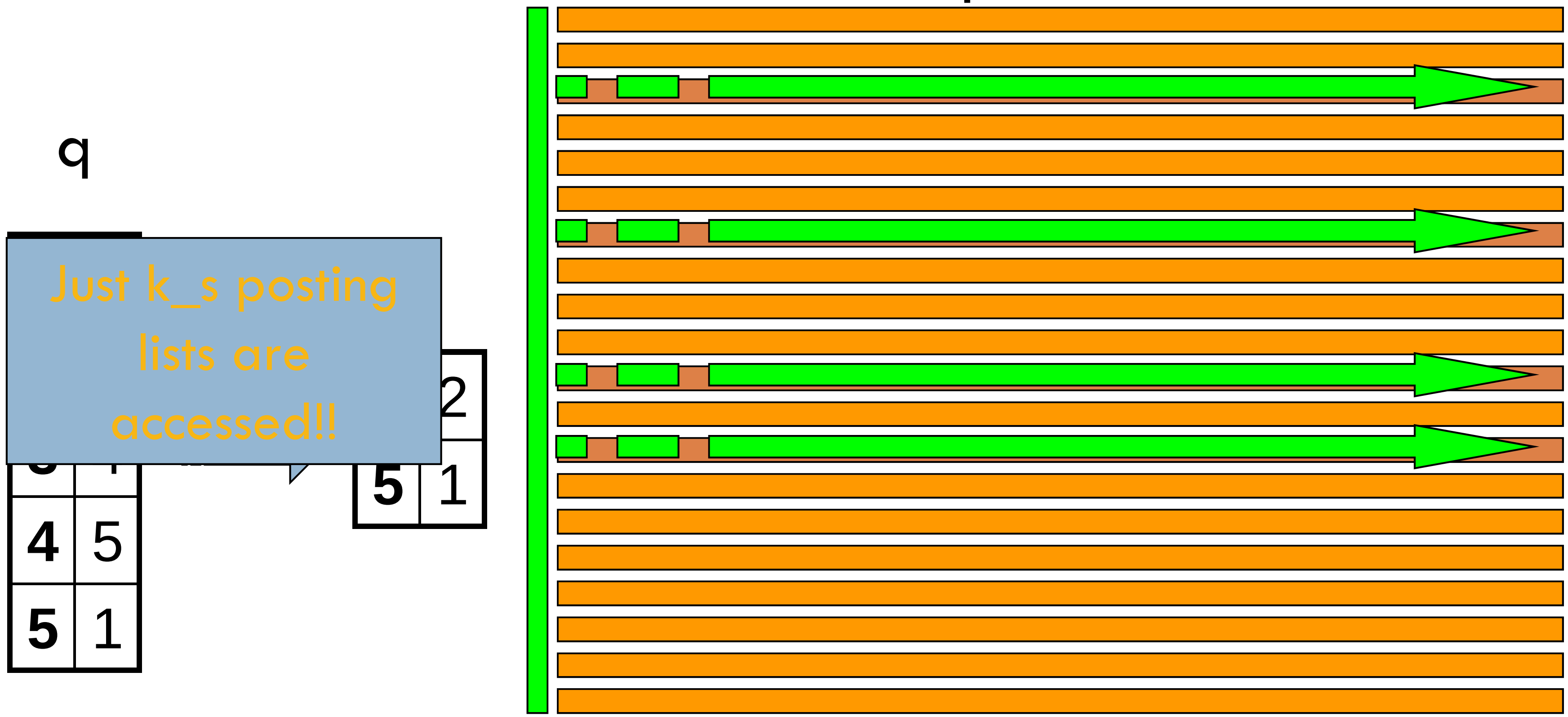
Problem: the matrix is not sparse

3) Posting lists are accessed entirely



Reducing accessed posting lists

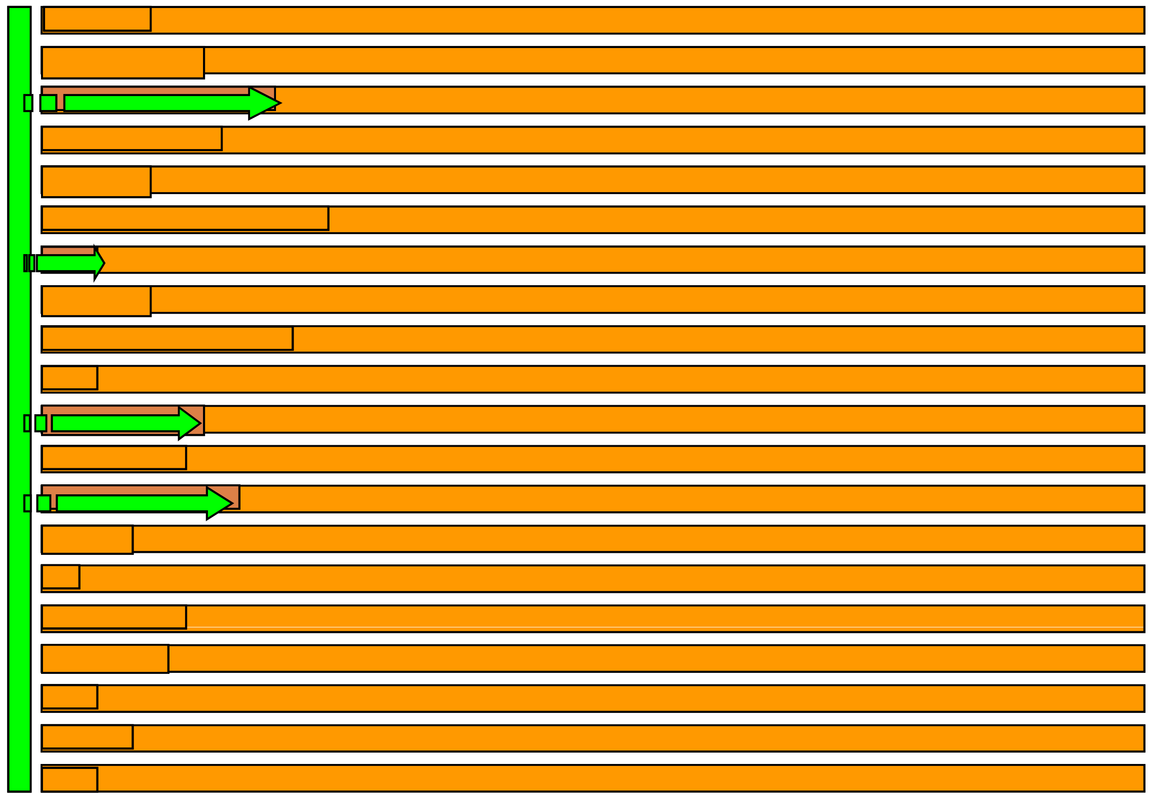
- Queries are represented by the the k_s closest reference objects, where k_s can be much smaller than the number of reference objects



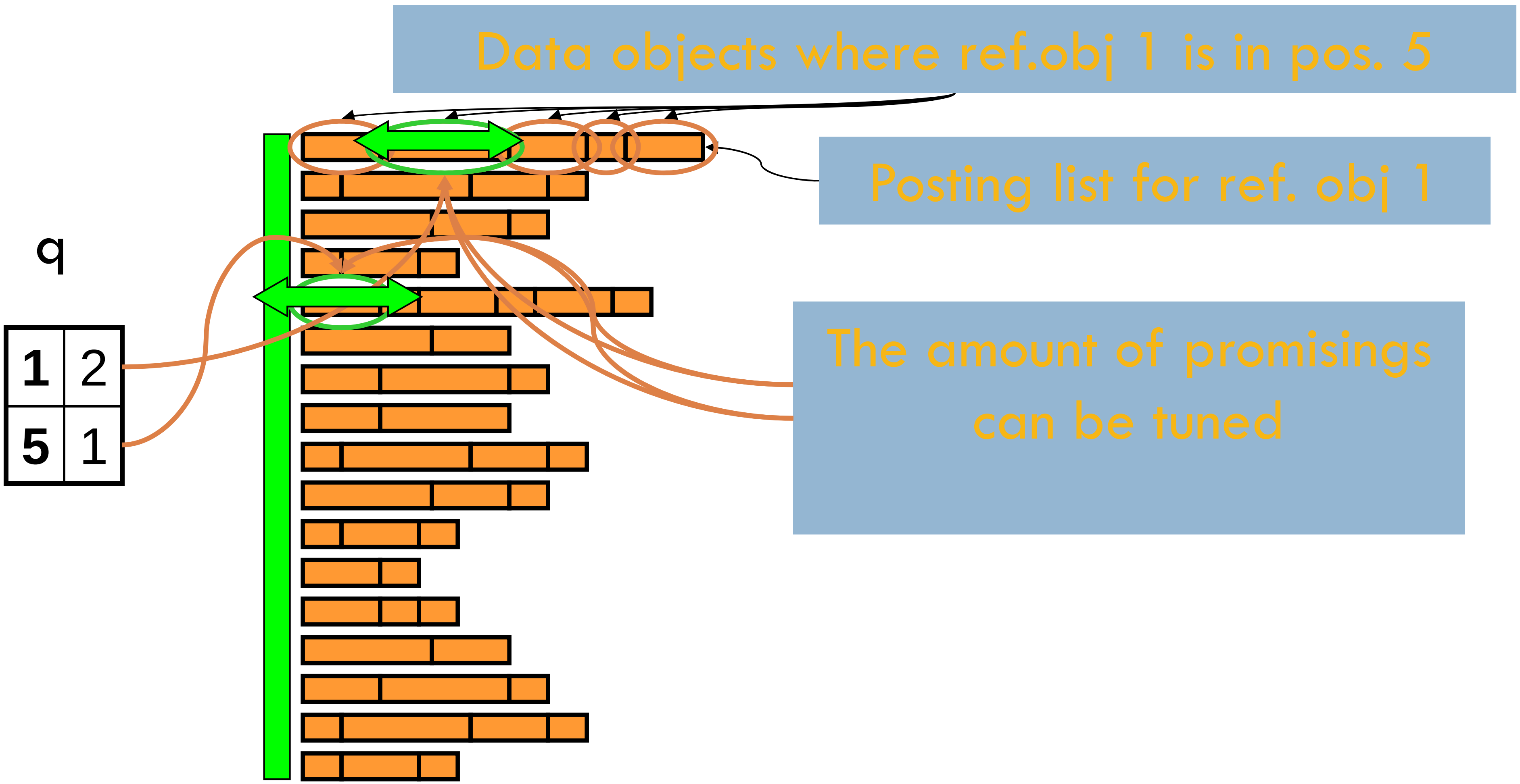
Reducing size of posting lists

- Data objects are represented by the k_i closest reference objects, where k_i can be much smaller than the number of reference objects

- An objects is referred in just k_i posting lists
- Posting list are much shorter

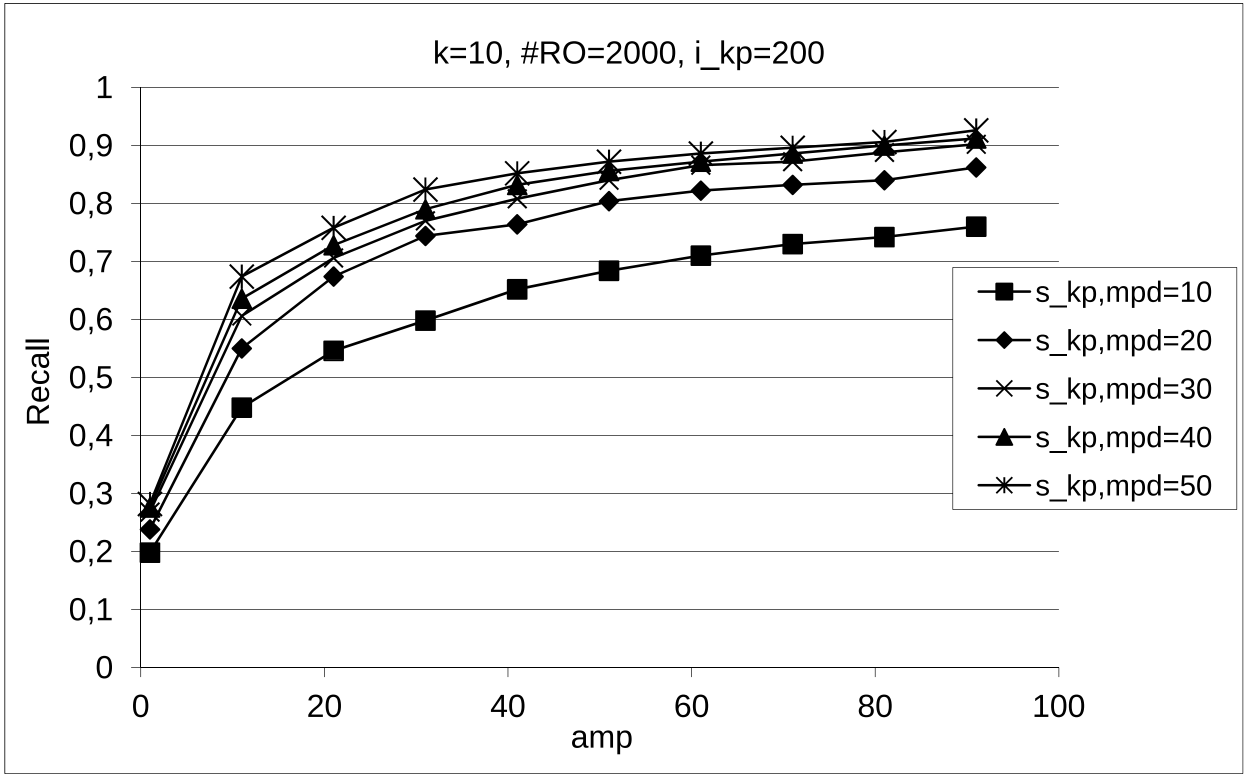


Accessing a fraction of the posting lists



Results

- We can retrieve more objects than required and reorder, with small additional cost



Outline

- The problem: examples and scenario
- Visual descriptors
- Similarity and approximate similarity searching on a large scale
- Image classifications

Classification: 1NN

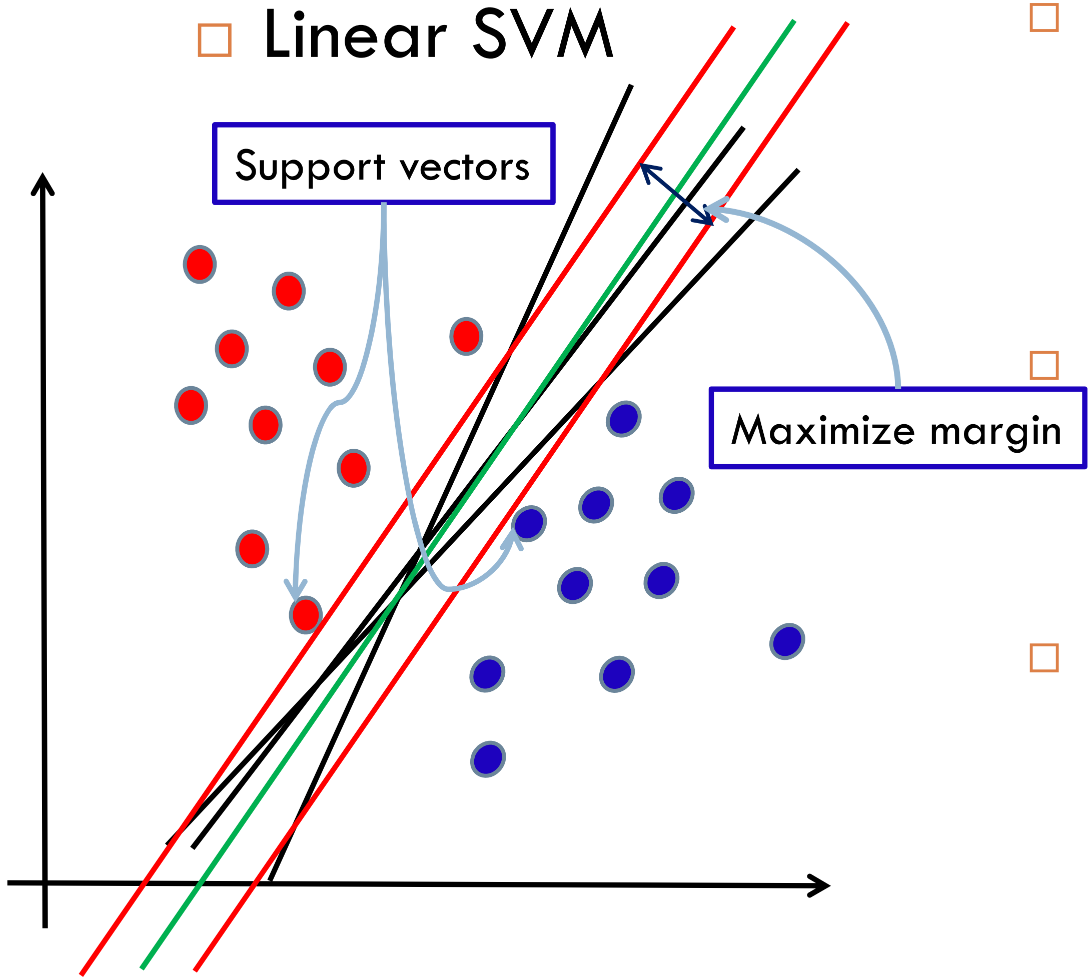
- Training set:
  - $\{(x_1,y_1), \ldots, (x_n,y_n)\}$
  - (for instance x are images and y are classes (e.g. monuments in images))
- Task: given a new image x_q determine its class (content??)
- Algorithm:
  - Search the image x_i closest to x_q
  - The class of x_q is y_i

Classification: kNN

- Estension of the 1NN to k nearest neighbours
- Algorithm:
  - Search the images $[x_{i1} \ldots x_{k1}]$ closest to x_q
  - The class of x_q is decided by majority voting of the lables associated with $[x_{i1} \ldots x_{k1}]$.
- Parameters
  - What k should be used?
  - Which majority voting strategy?
    - Typical choice: Weighted majority voting (votes of closest objects count more than far neighbors)

Classification: Support Vector Machines (SVM)

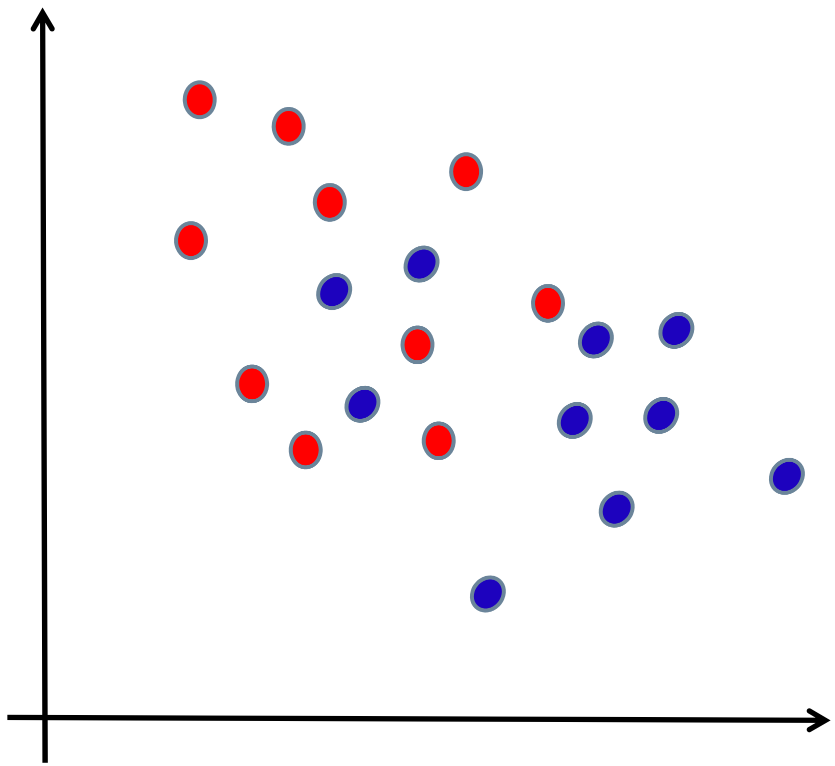
- Linear SVM



- Green hyperlane obtained using
  - Quadratic programming,
  - Gradient descent algorithm (e.g. Adatron)
- Given a point x decision on its class is taken using the decision function
  - f(x)=sign(x*ω-b)
- Where, ω is the vector representing the green hyperplane, obtained during the learning phase

Classification: Support Vector Machines (SVM)

- Non linear SVM



- Sometimes there is not an hyperplane that divedes red points from blue points
- However, such an hyperplane can be found mapping data in an higher dimension
- To solve it easily we can use the kernel trick

Classification: Support Vector Machines (SVM)

- Kernel trick:
  - Given a mapping function Φ that maps x into an higher dimensional space, the decision function could be

$$f(\mathbf{x}) = \boldsymbol{\omega} * \Phi(\mathbf{x}) + b$$

  - However, if we have a kernel function, K, defined as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) * \Phi(\mathbf{x}_2)$$

  - It can be demonstated that the decision function is equivalent to

$$f(\mathbf{x}) = y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) - b$$

  - And the problem can be solved using the kernel function without actually mapping data in higher dimensional space

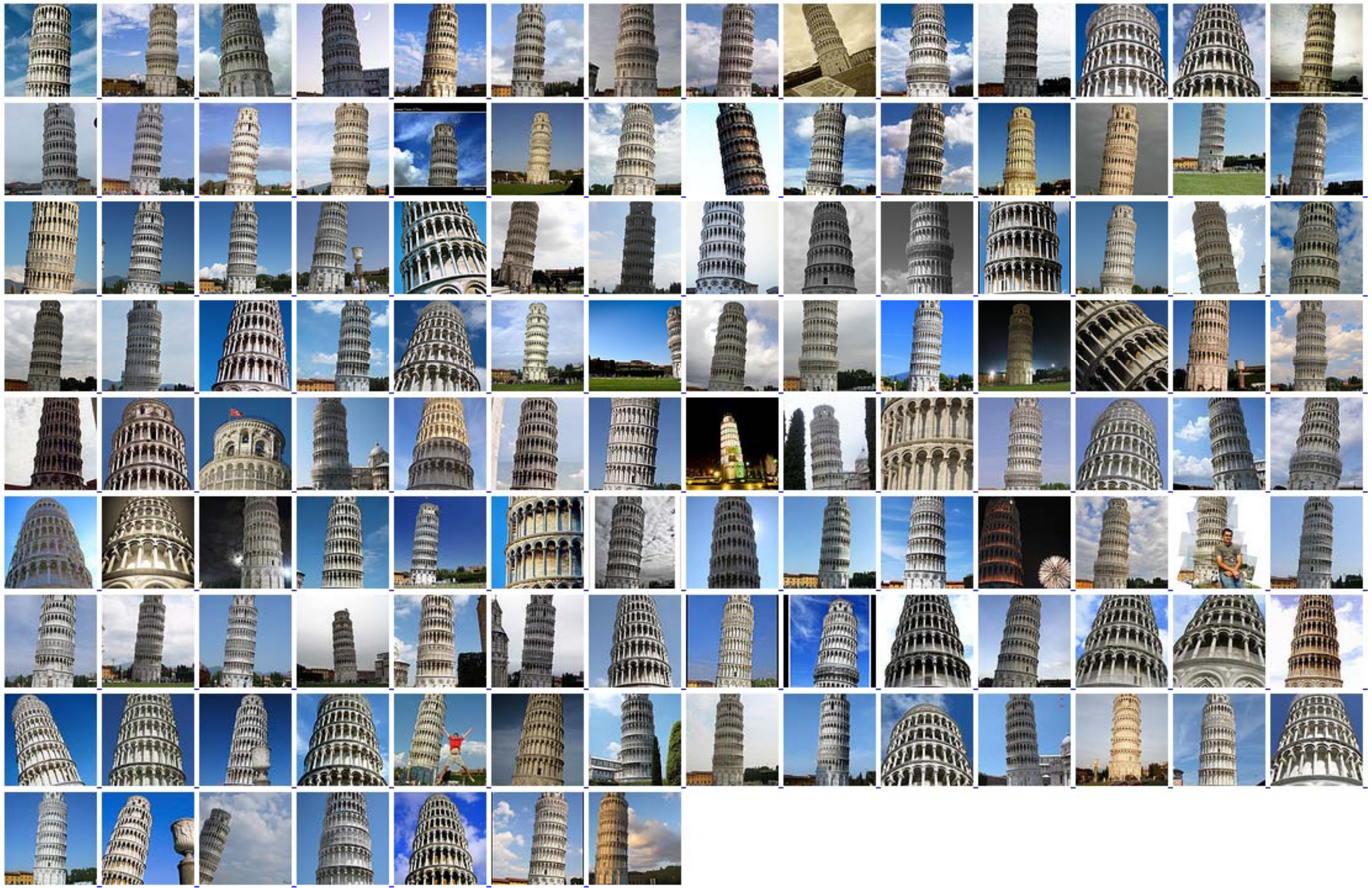
Image classification: what is this image content?

- Create a training set containing annotated images with the content we want to recongize
  - Manually or
  - Automatically

- Using local or global features to represent image content

- Build a classifier using the training set, for example a K-NN classifier

Pisa-Dataset



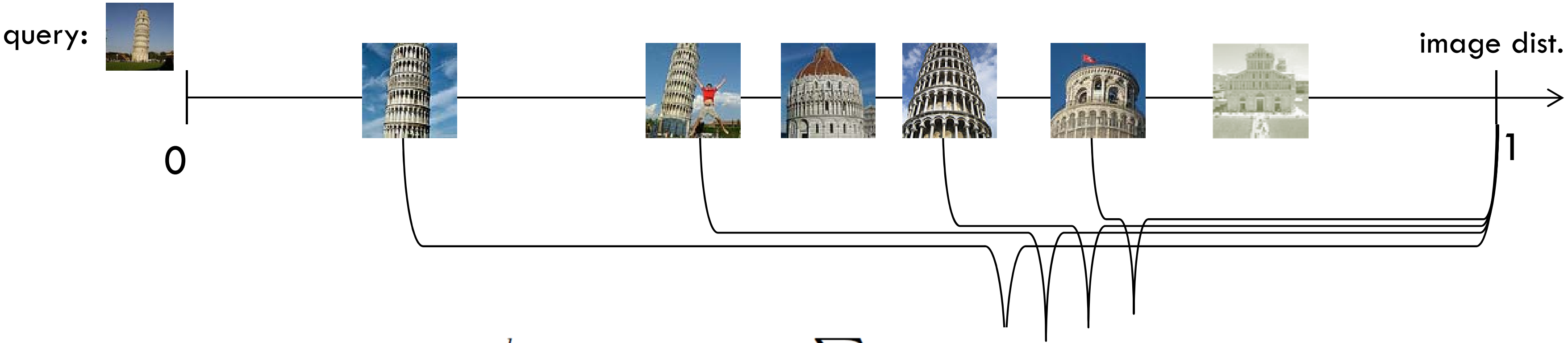
Pisa-Dataset

Pisa-Dataset

Image Classification: what is this image content?

K-NN classification:

- predicted class label: k=5



$$z^k(p_x, c_j) = \sum_{p_i \in \chi^k(p_x)\ :\ \Phi(p_i)=c_j} s(p_x, p_i)$$

$$\hat{\Phi}^s(d_x) = \arg\max_{c_j \in C} z(d_x, c_j)$$

Conclusions

- Searching by using the visual content of documents is a very important issue
- Methods for similarity search on a very large scale are now mature and can be applied to search by similarity on very large archives in a fraction of a second
- Open research directions
  - Dealing with local features
  - Scalable classification techniques
  - Automatic annotation

Demos

- MI-File
  - http://mi-file.isti.cnr.it:/CophirSearch/
- Europeana image search – ASSETS
  - http://virserv.isti.cnr.it:8080/assetsIRService
- Alinari24Ore
  - http://melampo.isti.cnr.it/
- Image Classification
  - http://multimatch01.isti.cnr.it:18080/ImageClassificationUI/
- Image Searching from mobile phones

Suggested references:

Image Features:

•"MPEG-7". *ISO/IEC JTC1/SC29/WG11N6828.* 2004 http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm.
•*Distinctive Image Features from Scale-Invariant Keypoints*. Lowe, David G. 2, s.l. : Kluwer Academic Publishers Hingham, MA, USA , 2004, International Journal of Computer Vision, Vol. 60.
•*"SURF: Speeded Up Robust Features.* Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool,. 3, 2008, Computer Vision and Image Understanding (CVIU), Vol. 110, pp. 346--359.

Image retrieval/recognition systems:

•Google Images. *Google.* [Online] http://images.google.com/
•Google Googles. [Online] http://www.google.com/mobile/goggles/
•*Visito Tuscany.* [Online] http://www.visitotuscany.it.
•**Gennaro, Claudio.** Lucignolo. *Lucignolo Image Search Engine.* [Online] http://lucignolo.isti.cnr.it/cophirUI/.
•**Amato, Giuseppe.** MI-File. *Metric Inverted Image Search Engine.* [Online] http://mi-file.isti.cnr.it/CophirSearch/.

Efficient similarity Search:

•Similarity Search – The Metric Space Approach, Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, Michal Batko, 2005 Springer
•*Region Proximity in Metric Spaces and Its Use for Approximate Similarity Search.* Giuseppe Amato, Fausto Rabitti, Pasquale Savino, Pavel Zezula,. 2, 2003, Transaction on Information Systems (TOIS), Vol. 21.
•*Effective proximity retrieval by ordering permutations*,", E. Chavez Gonzalez, K. Figueroa, and G. Navarro, IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 9, pp. 1647–1658, 2008.
•*Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, A. Andoni and P. Indyk,, Commun. ACM, vol. 51, no. 1, pp. 117–122, Jan. 2008.
•*Use of Permutation Prefixes for Efficient and Scalable Approximate Similarity Search.* **E**suli, Andrea. 5, Information Processing & Management, Special Issue on cScale Distributed Indexing and Retrieval., Vol. 48.
•*Approximate Similarity Search in Metric Spaces using Inverted Files.* Giuseppe Amato, Pasquale Savino. 2008. INFOSCALE 2008, the Third Interational ICST Conference on Scalable Information Systems.
•MI-File: Using Inverted Files for Scalable Approximate Similarity Search, Multimedia Tools and Applications Journal, Springer, to appear,
•MI-File Library Source Code: http://code.google.com/p/mi-file/

Image search recognition classification

•*Mobile Visual Search.* Bernd Girord, et alt. IEEE SIGNAL PROCESSING MAGAZINE.
•*Tour the World: building a web-scale landmark recognition engine.* Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeie, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, Hartmut Neven. 2009. CVPR.
•*The Distance-Weighted k-Nearest-Neighbor Rule.* Dudani, Sahibsingh A. 4, 1976 , IEEE Transactions on Systems, Man and, Vol. 6.
•An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, N. Cristianini and J. Shawe-Taylor, , Cambridge University Press, March 2000.
•*Geometric consistency checks for kNN based image classification relying on local features.* Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro. 2011. SISAP '11 Proceedings of the Fourth International Conference on Similarity Search and Applications.