Recognition by alignment

370: Intro to Computer Vision

Subhransu Maji

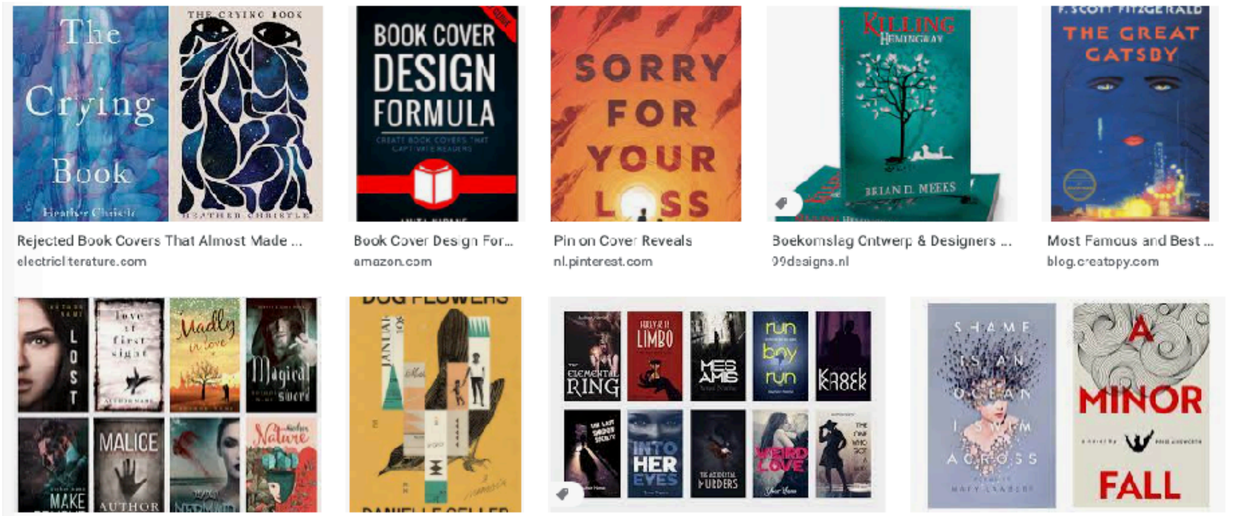
April 8, 2025

College of
INFORMATION AND
COMPUTER SCIENCES

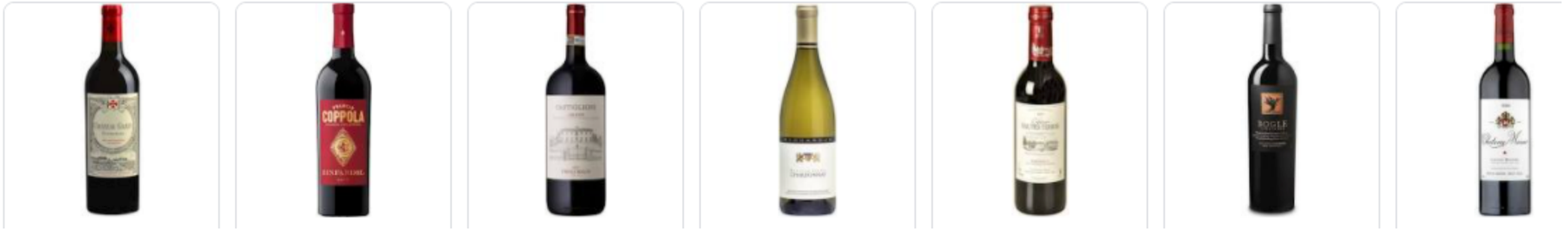
UMASS
AMHERST

Instance recognition

Landmarks



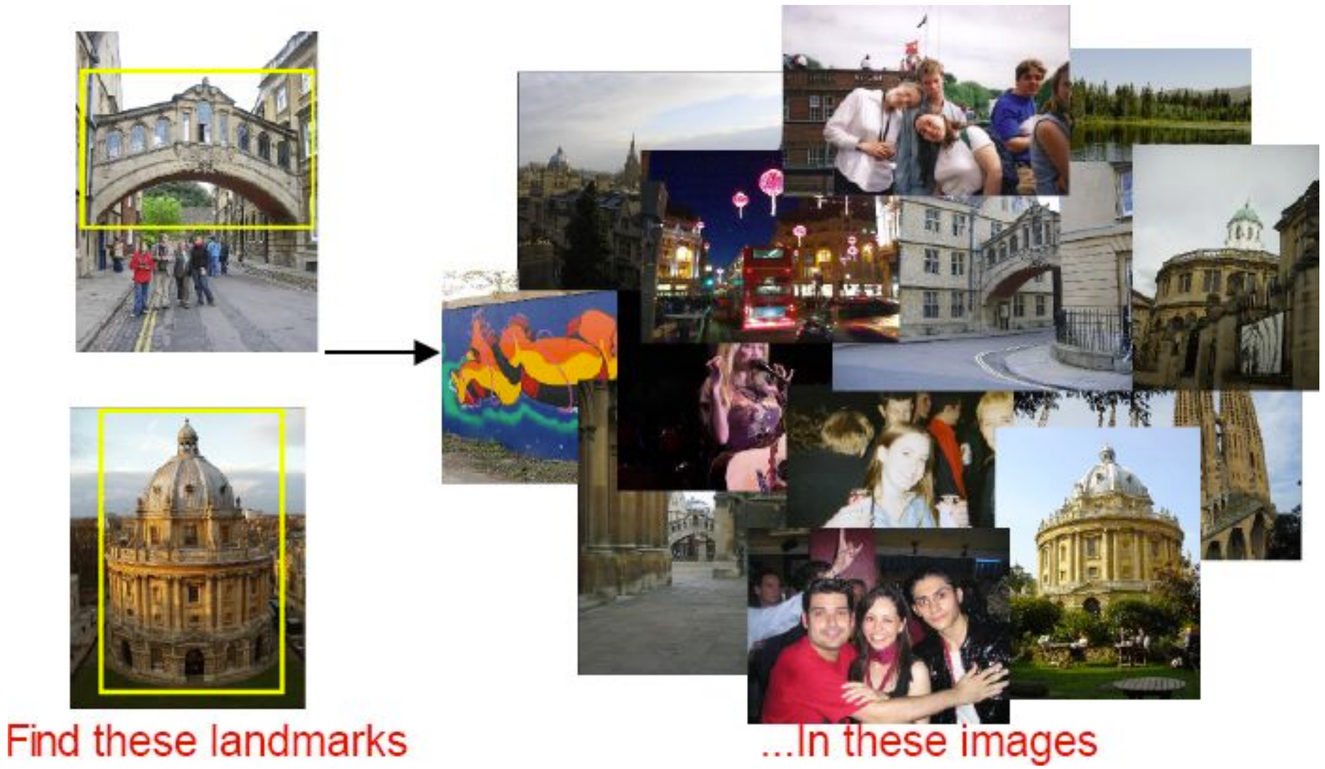
Books



Wines

Oxford building search demo

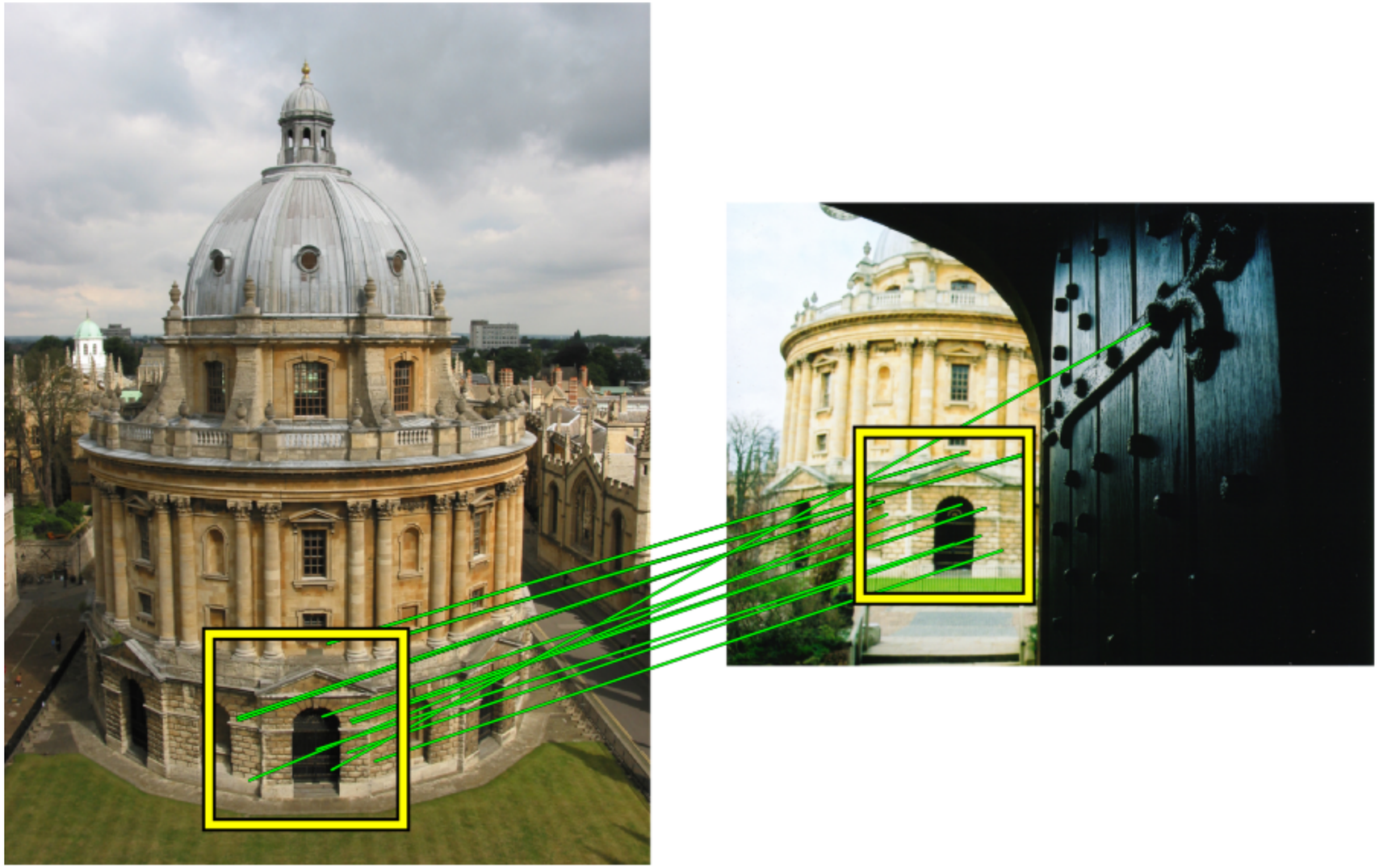
http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html



Challenges: scale, viewpoint, lighting and occlusions

Feature matching + geometric alignment

http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html



Today's lecture

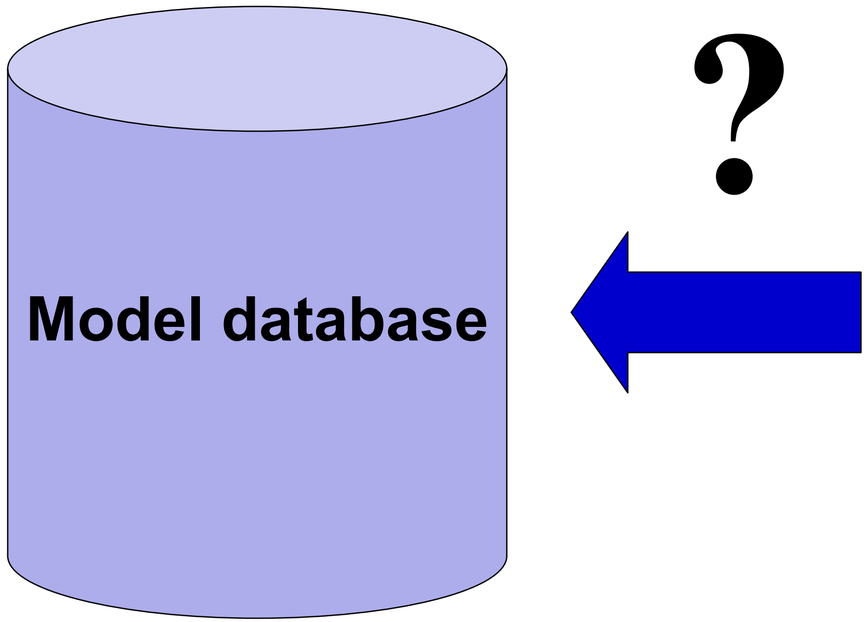
Scaling instance recognition

Beyond instances

Alignment to large databases

What if we need to align a test image with thousands or millions of images in a model database?

- Efficient putative match generation using **approximate search**, **inverted indices**



Test image

Large-scale document search

Inverted index

- Hash table listing all the documents containing a given word.

Query (“blue sky”)

- For each word (i.e, “blue” and “sky”) in the query retrieve all the documents that contain it
- Intersect the lists
- Rerank using page rank, popularity, etc.

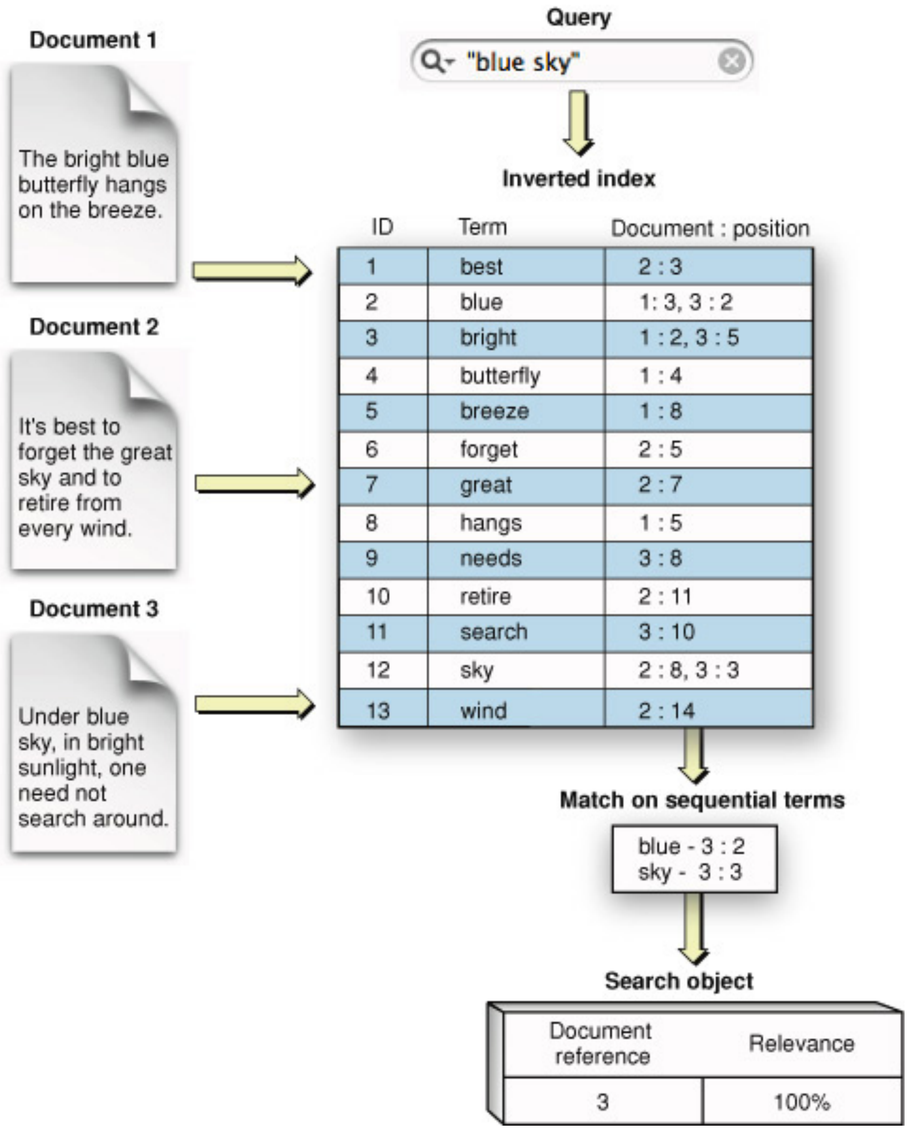


Figure from: https://developer.apple.com/

Large-scale visual search

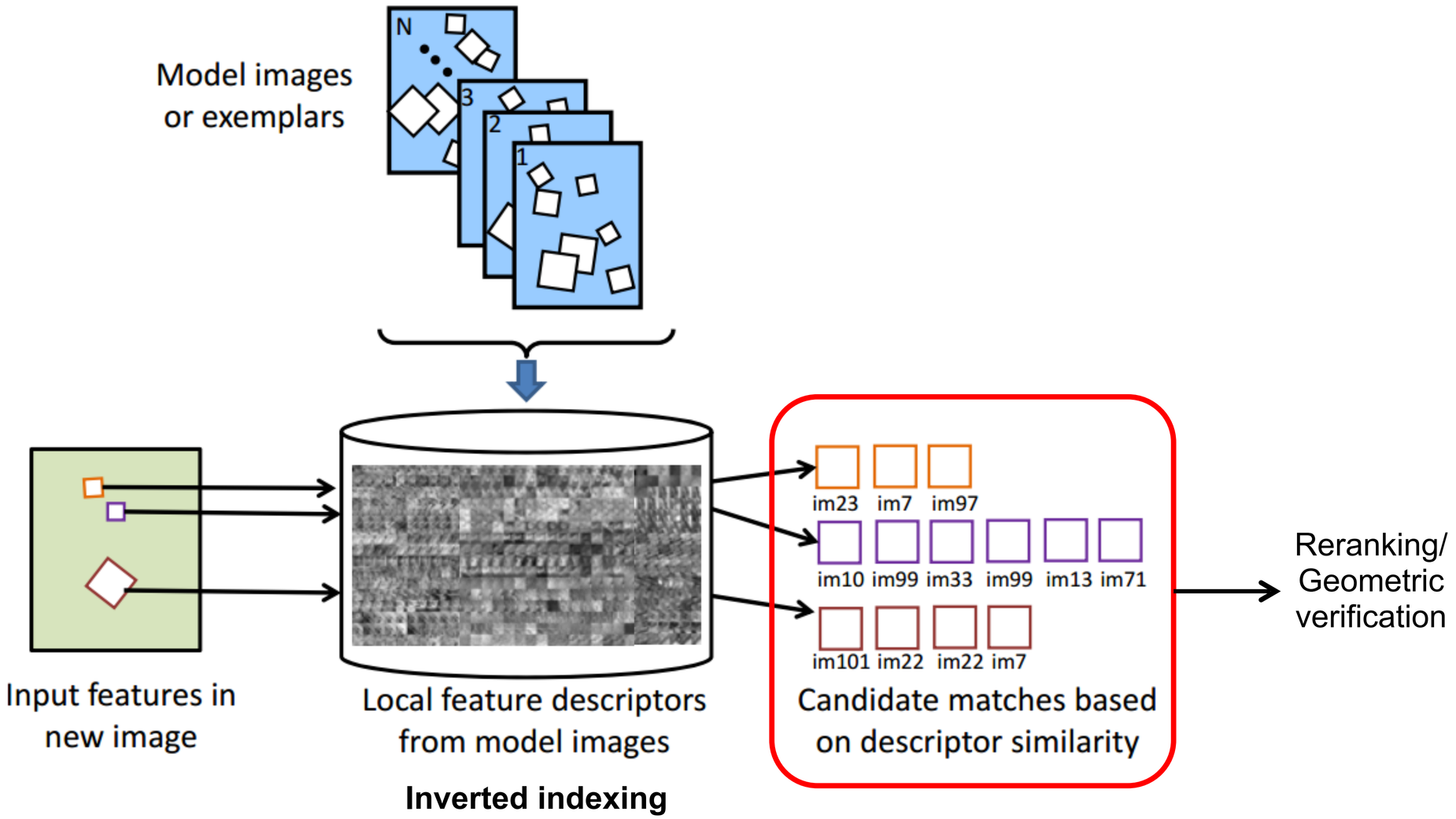
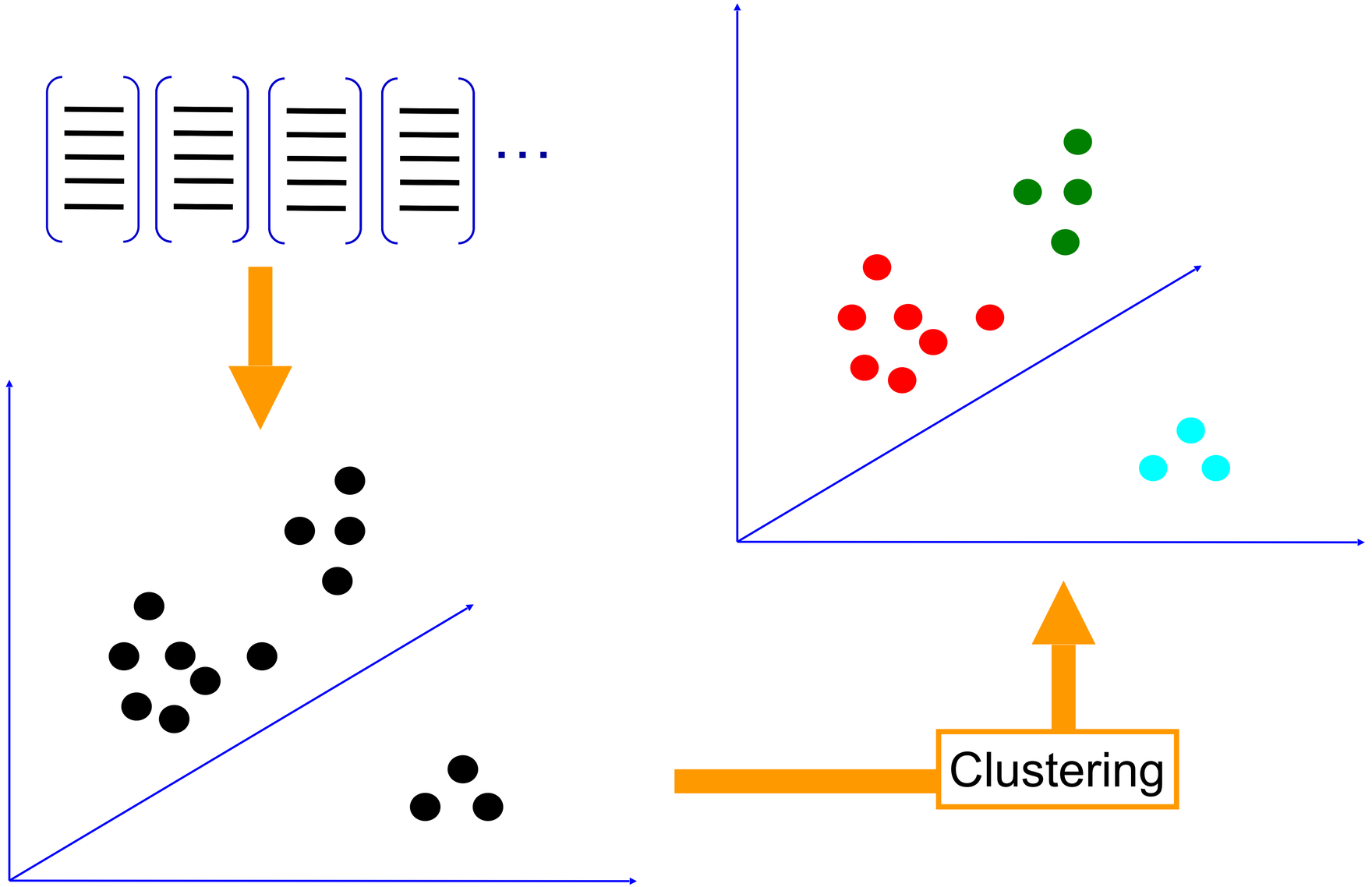


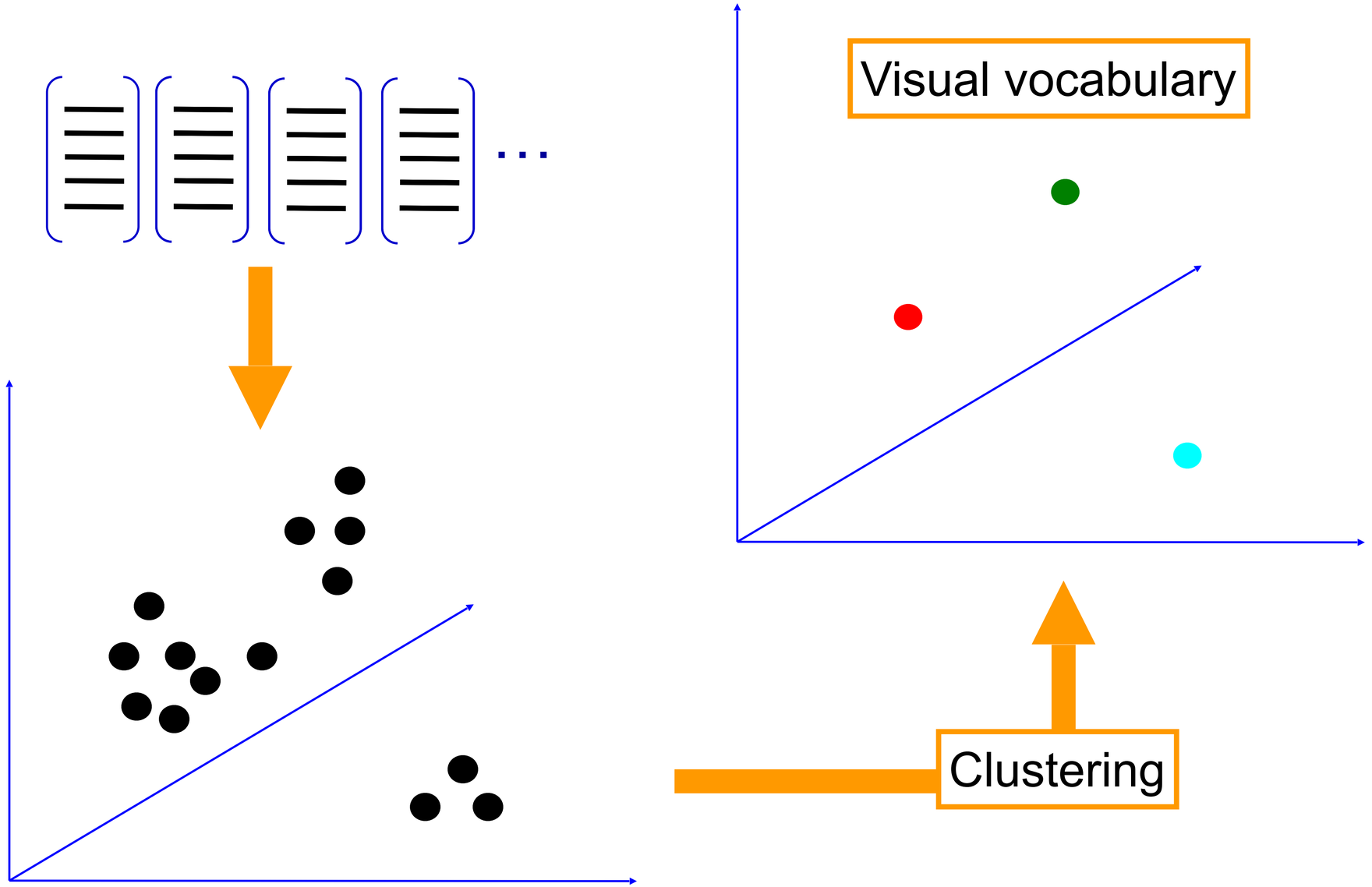
Figure from: Kristen Grauman and Bastian Leibe, Visual Object Recognition, Synthesis Lectures on Artificial Intelligence and Machine Learning, April 2011, Vol. 5, No. 2, Pages 1-181

Learning a dictionary



Slide credit: Josef Sivic

Learning a dictionary

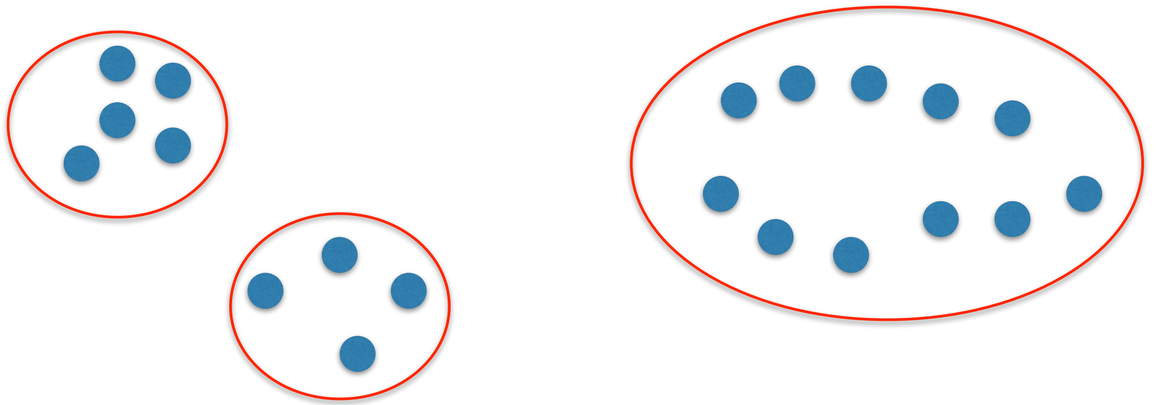


Slide credit: Josef Sivic

Clustering

Basic idea: group together similar instances

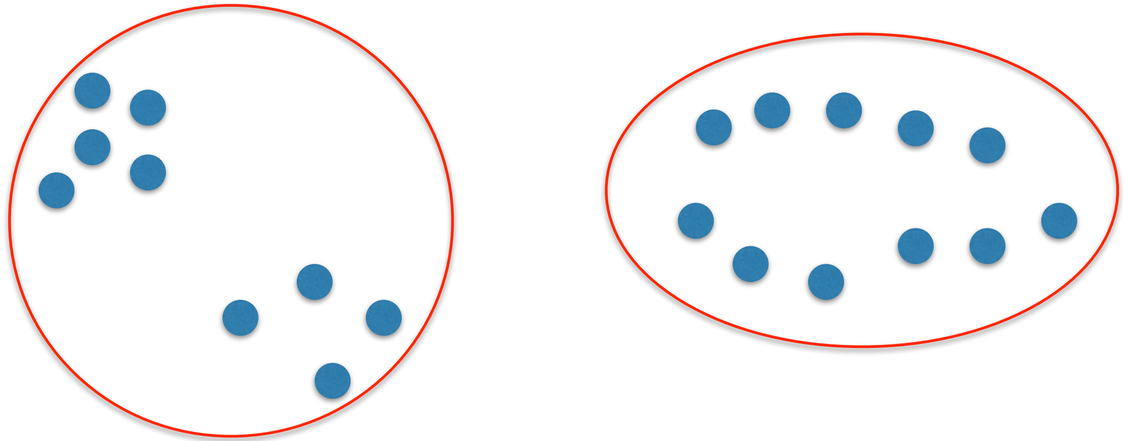
Example: 2D points



Clustering

Basic idea: group together similar instances

Example: 2D points



What could similar mean?

- One option: small Euclidean distance (squared)

$$\text{dist}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2^2$$

- Clustering results are crucially dependent on the measure of similarity (or distance) between points to be clustered

Clustering using k-means

Given **(x_1, x_2, …, x_n)** partition the **n** observations into **k (≤ n)** sets **S =** {S_1, S_2, …, S_k} so as to minimize the within-cluster sum of squared distances

The objective is to minimize:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2$$

cluster center

Lloyd's algorithm for k-means

Initialize k centers by picking k points randomly among all the points

Repeat till convergence (or max iterations)

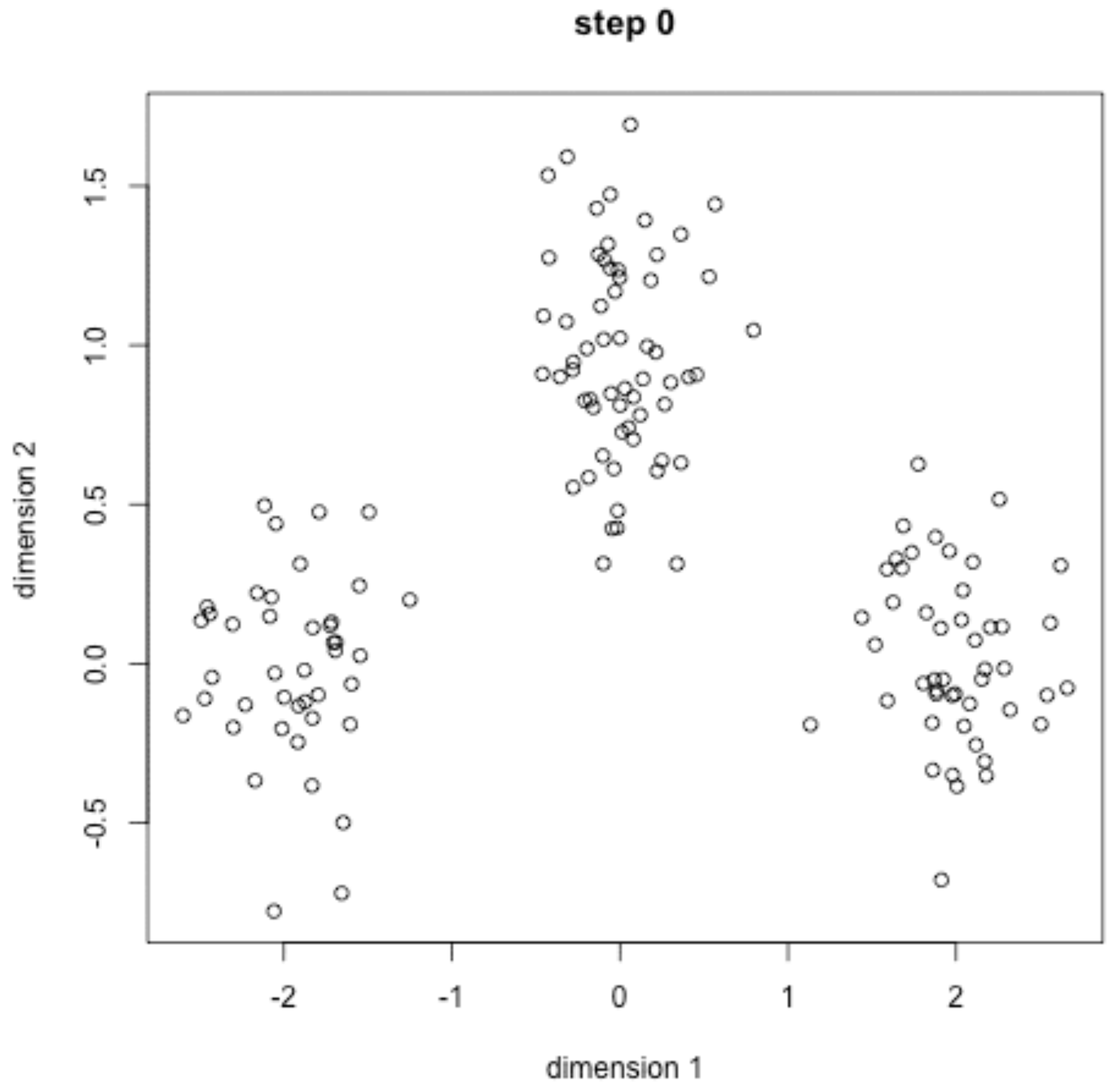
- Assign each point to the nearest center (assignment step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2$$

- Estimate the mean of each group (update step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2$$

k-means in action



http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/

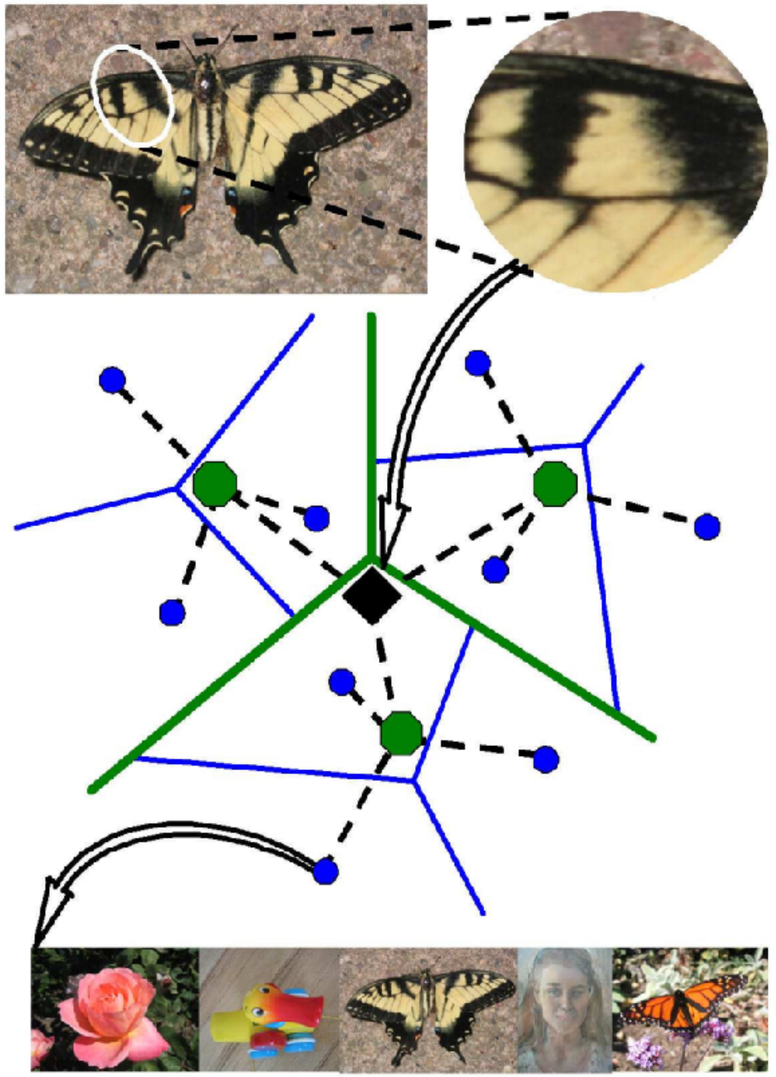
In the standard k-means algorithm, what is the computational cost of assigning a data point to its nearest cluster center when the vocabulary size is k?

Vocabulary trees

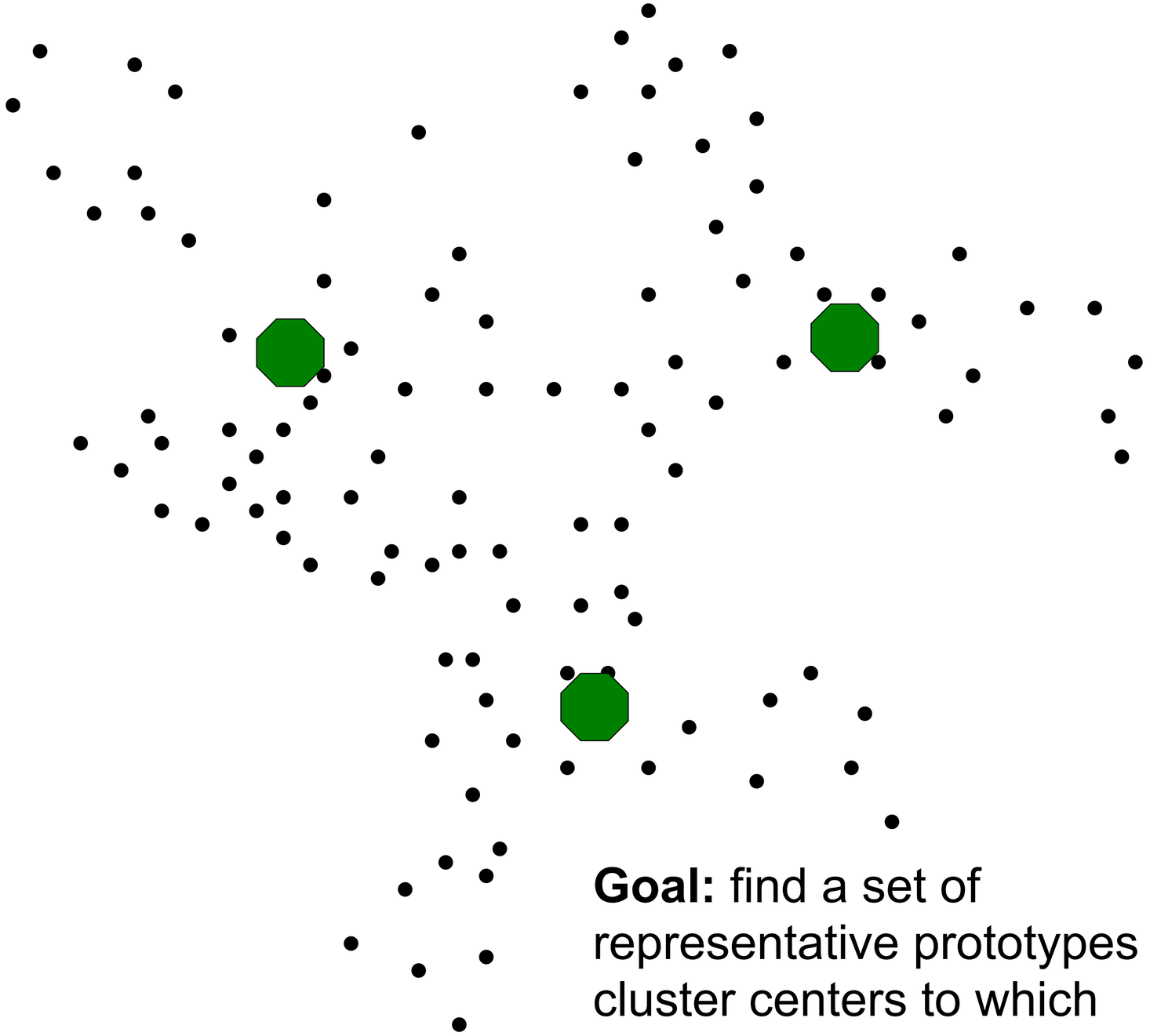
Test image

Vocabulary tree with inverted index

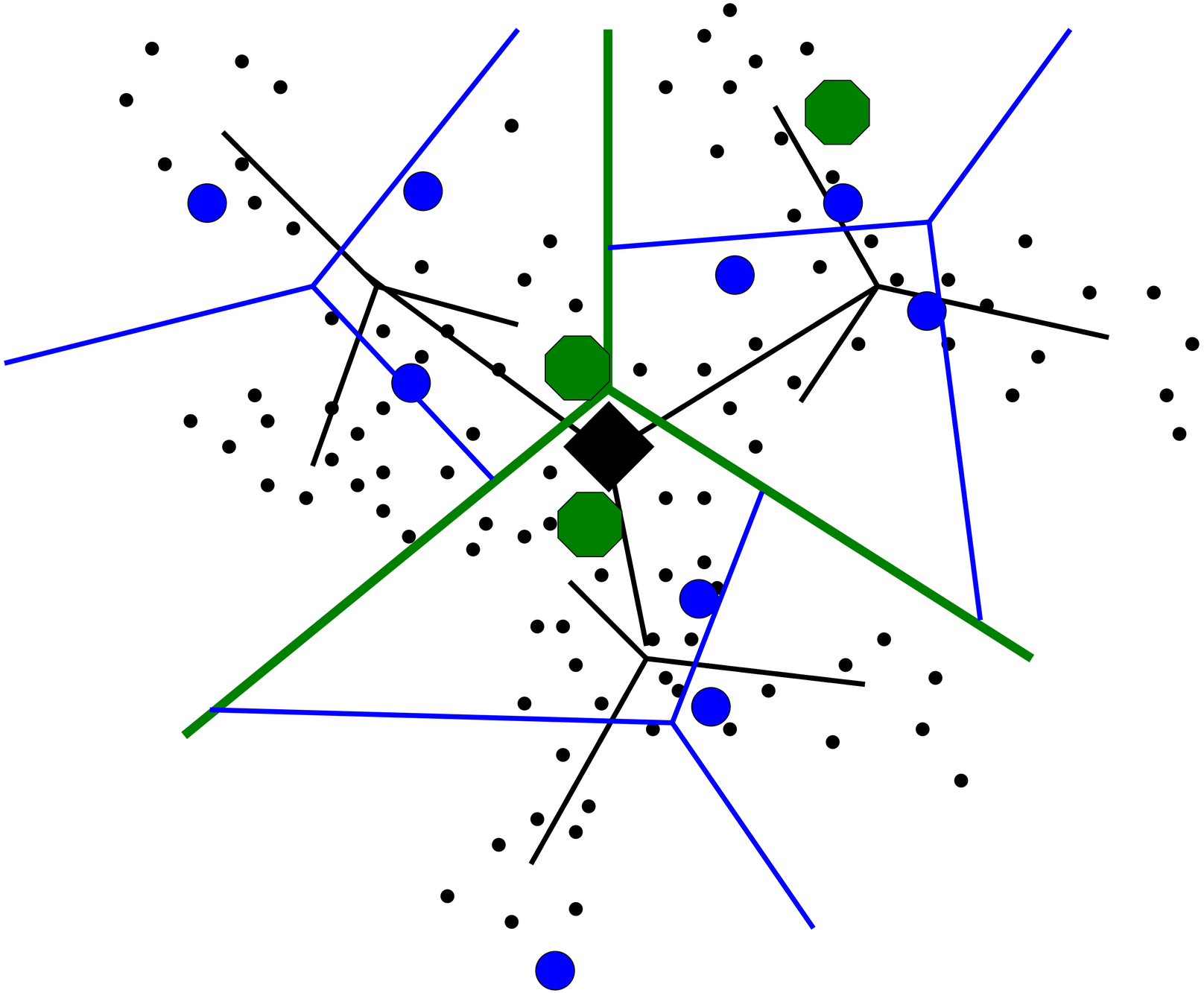
Database



D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

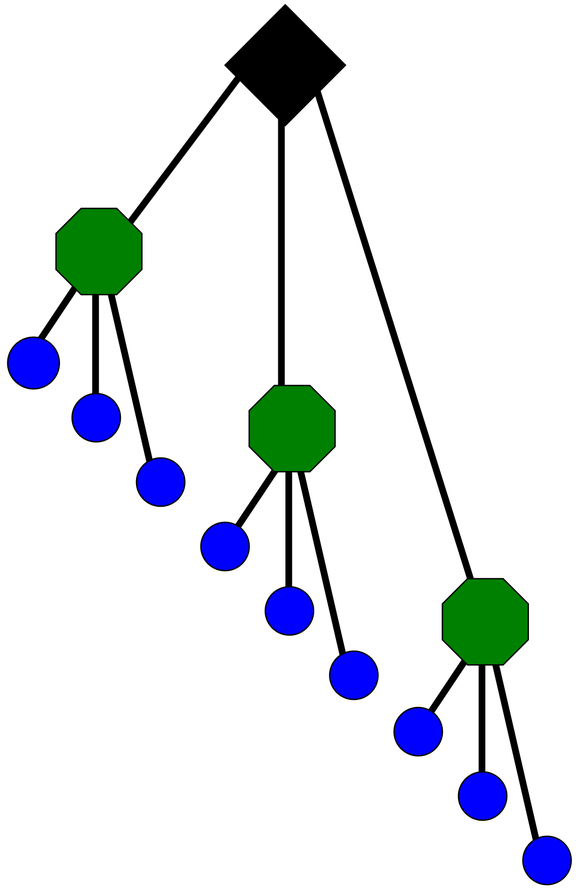


Goal: find a set of representative prototypes or cluster centers to which descriptors can be quantized



Slide credit: D. Nister

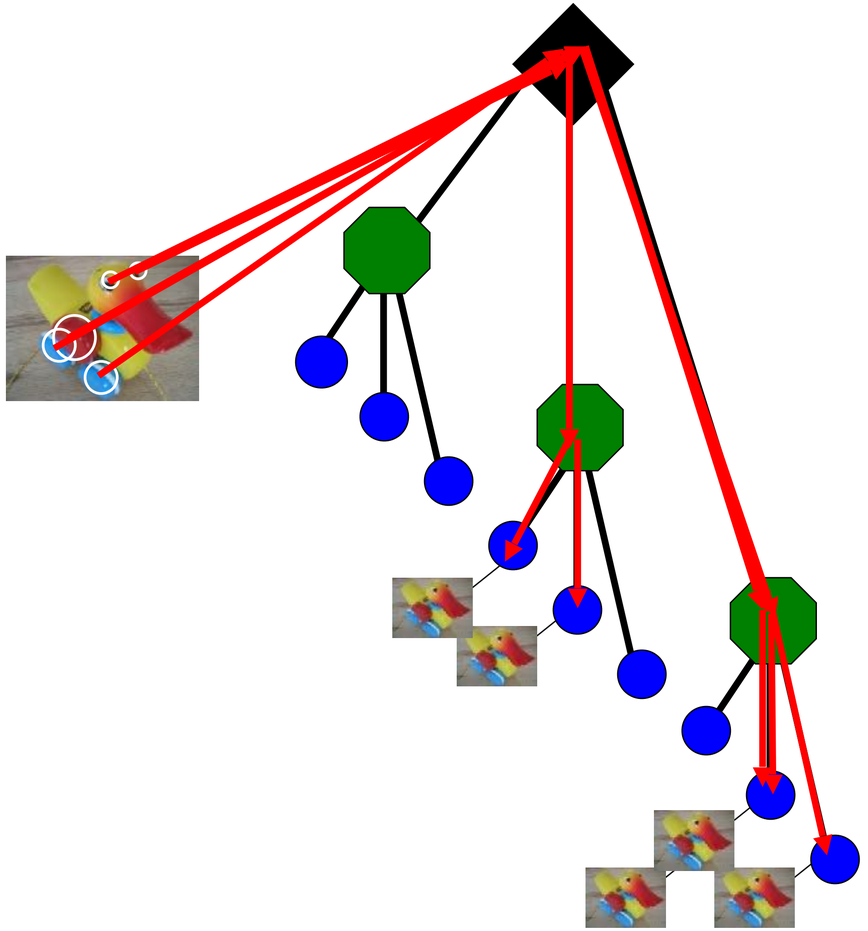
In the hierarchical k-means algorithm, what is the computational cost of assigning a data point to its nearest cluster center when the vocabulary size is k?



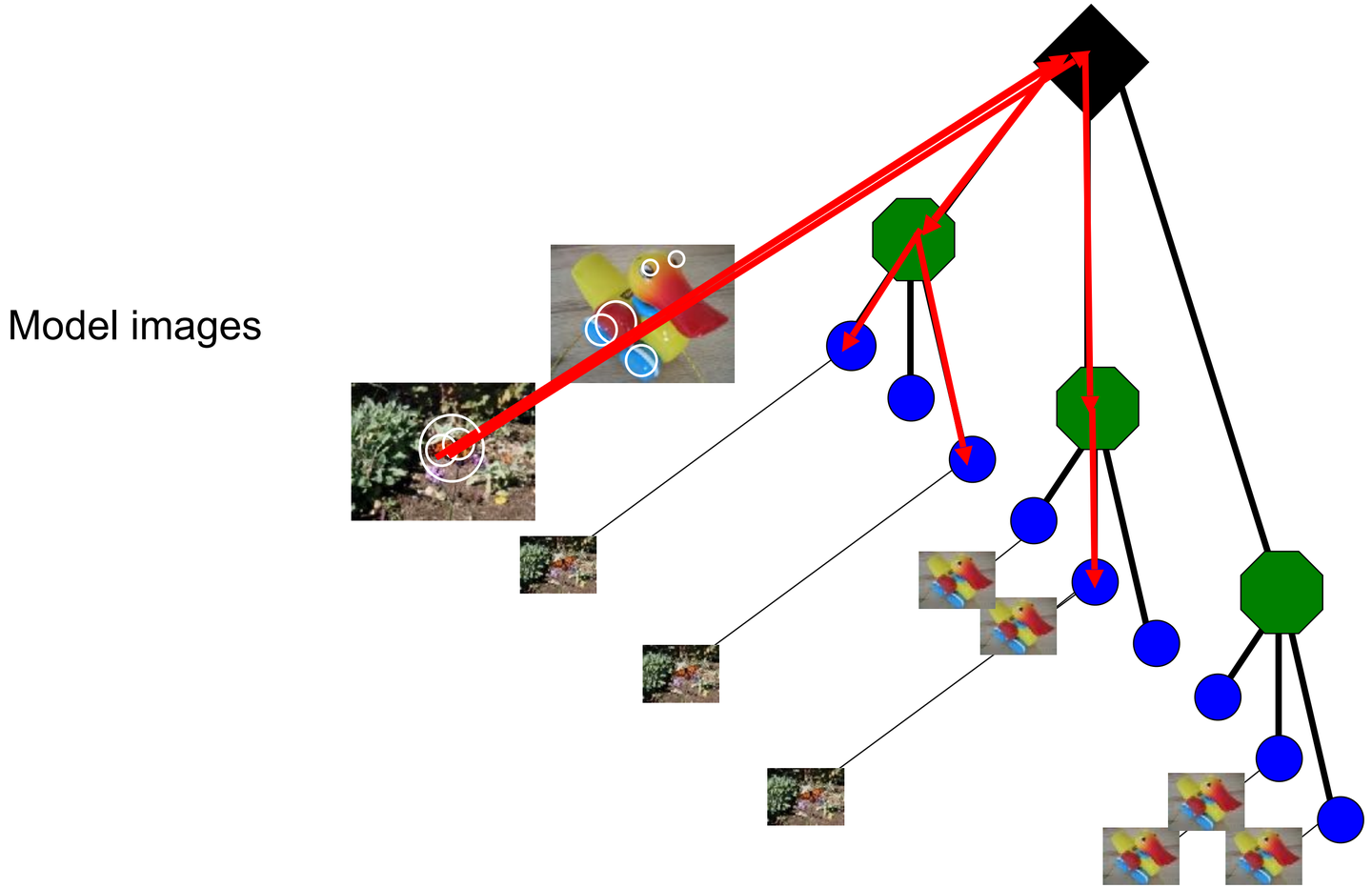
Vocabulary tree/inverted index

Slide credit: D. Nister

Model images

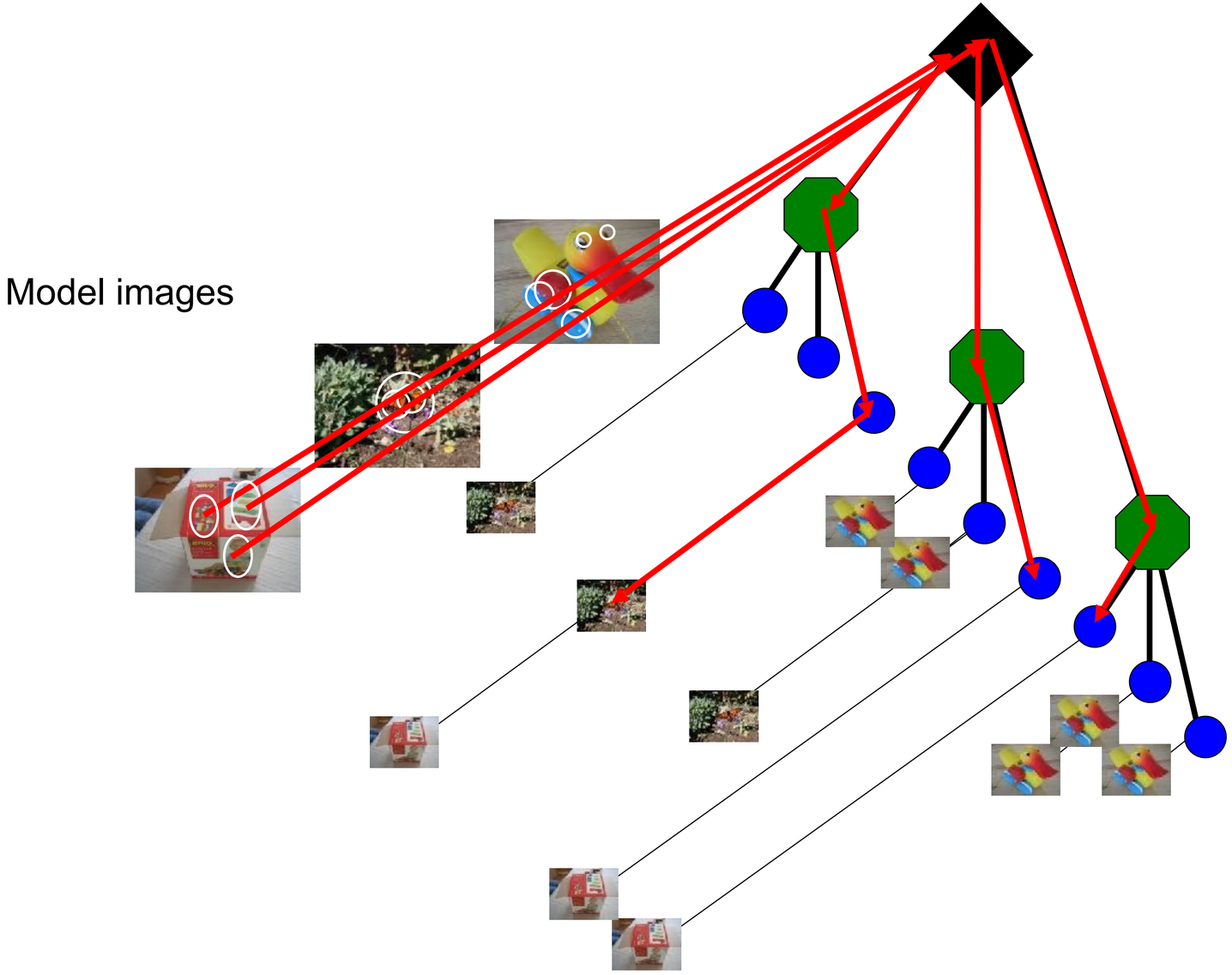


Slide credit: D. Nister



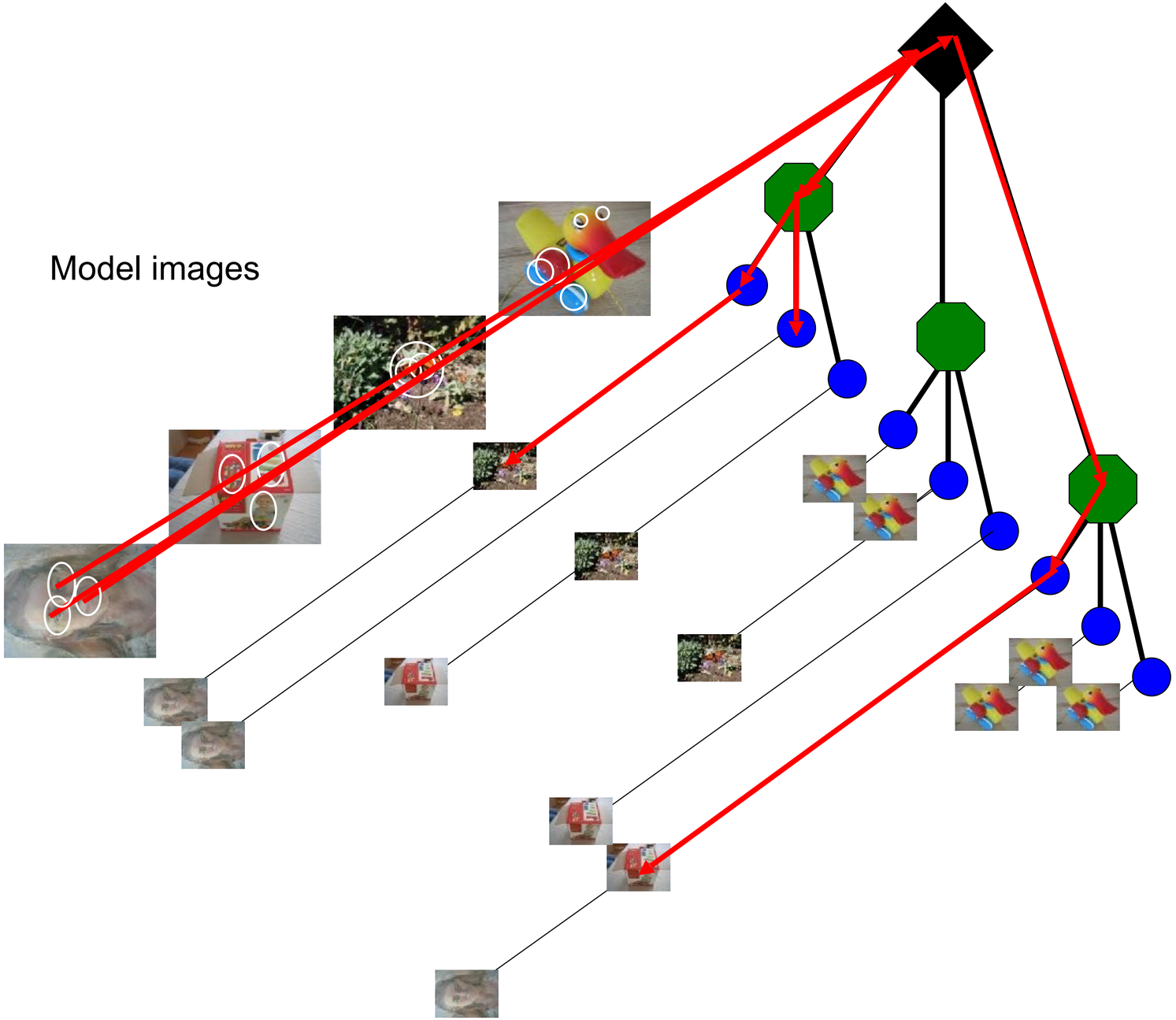
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



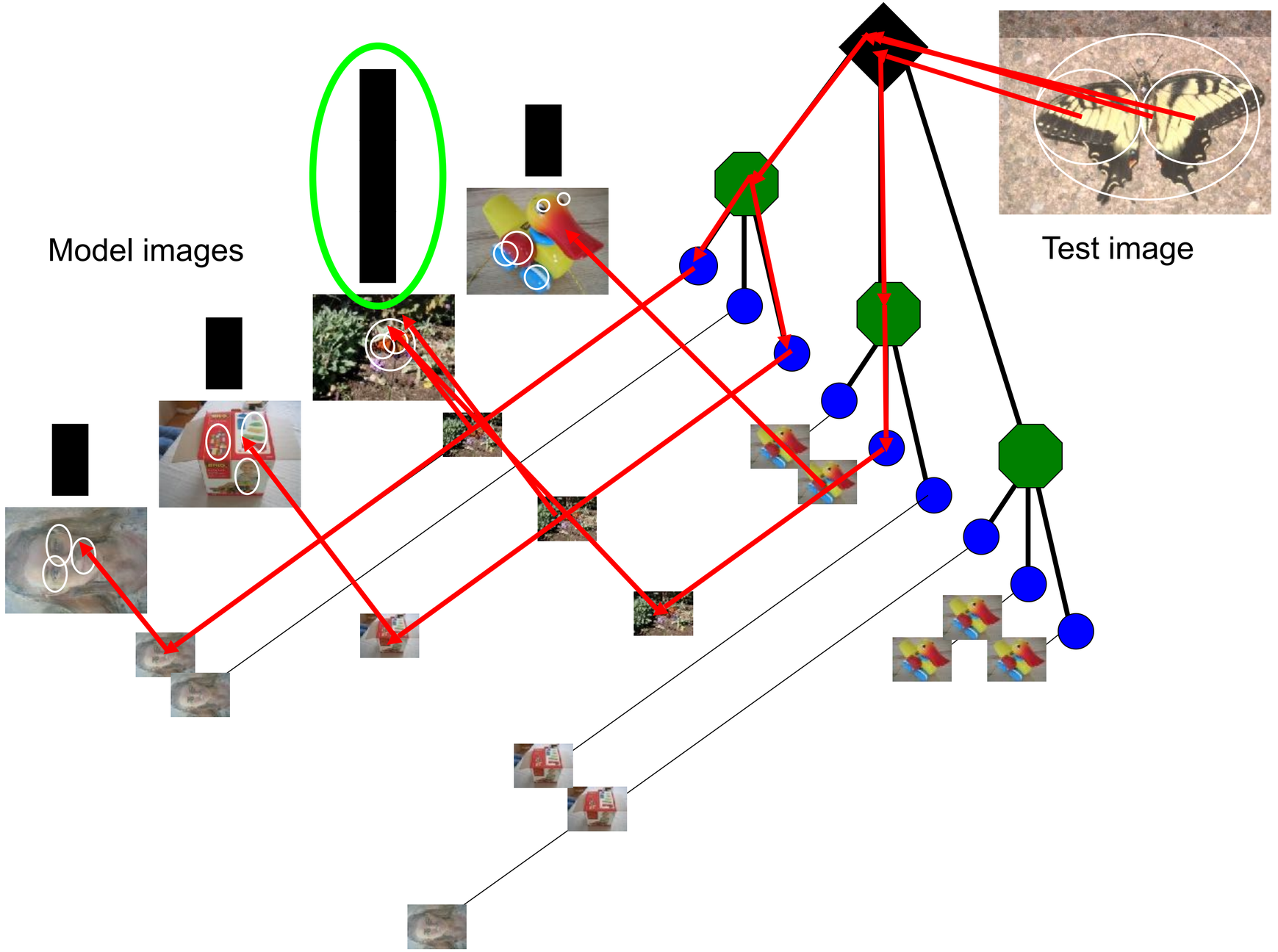
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Looking up a test image

Slide credit: D. Nister

Approximate nearest neighbors

Vocabulary trees are one of many data structures to accelerate nearest neighbor search

Other examples

- **k-d tree** — recursively split each dimension along the median
- **locality sensitive hashing**

Tradeoff between speed and accuracy

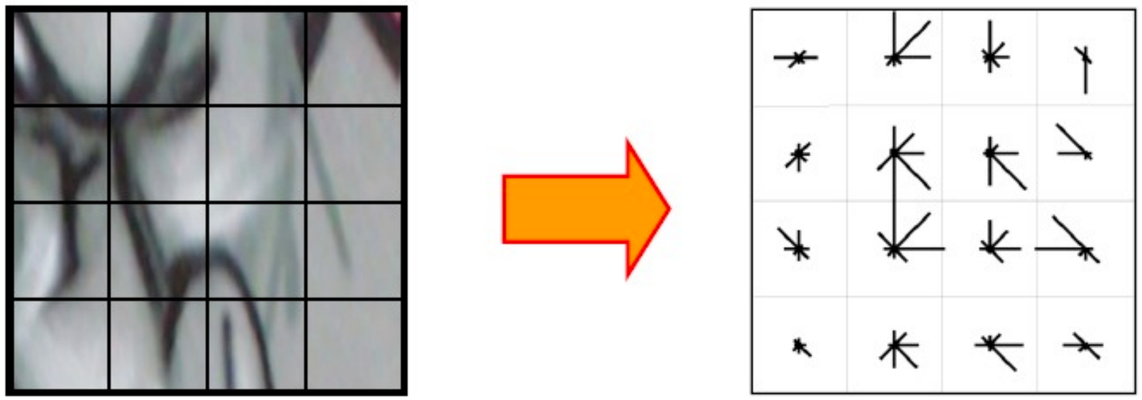
Today's lecture

Scaling instance recognition

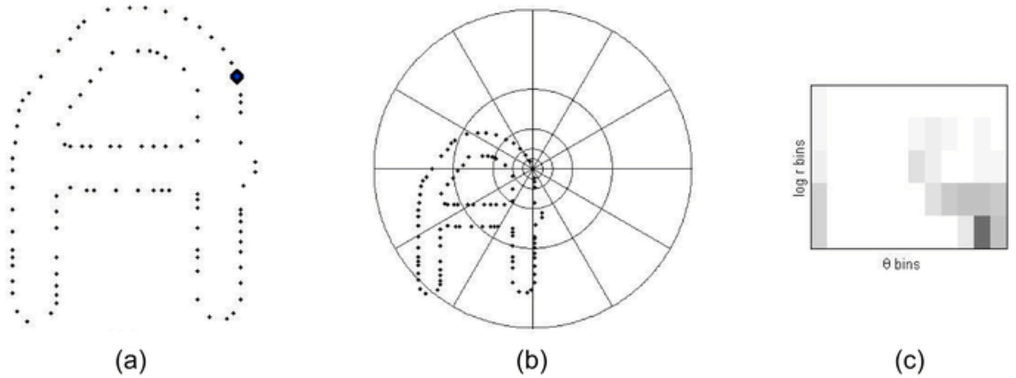
Beyond instances

Descriptors for shape matching

Generalize SIFT to incorporate more distortions



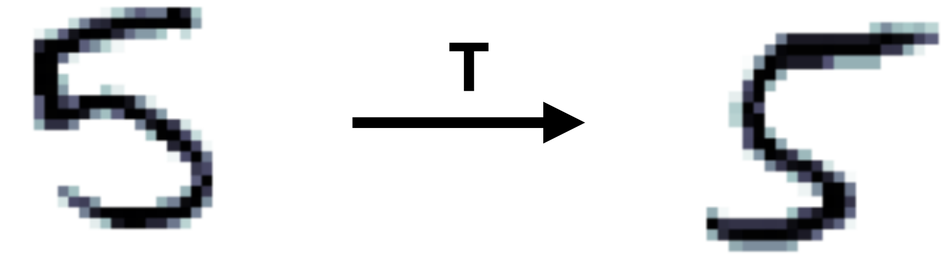
Scale Invariant Feature Transform (SIFT)

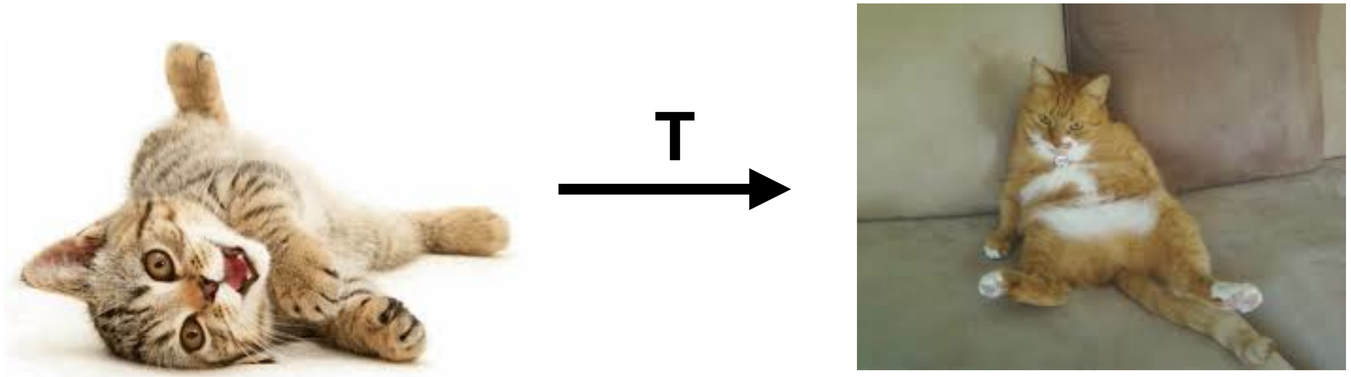


Shape context [Belongie et al., 2000]

Note the use of log-polar bins

Non-rigid transformations





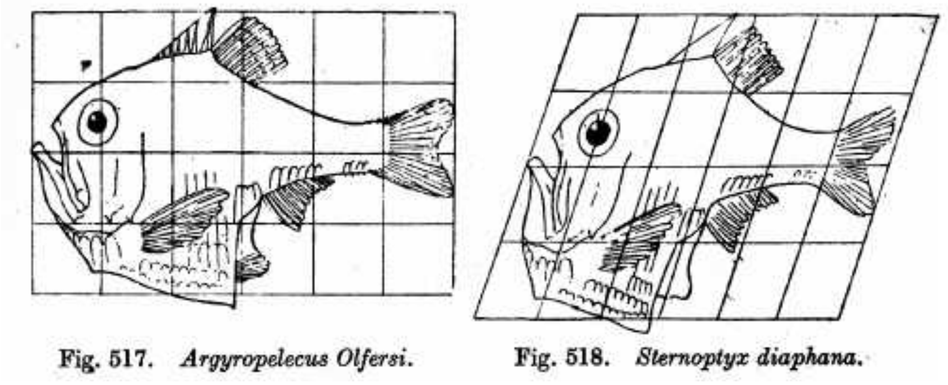
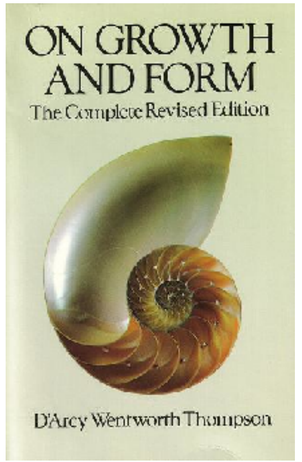


Fig. 517. *Argyropelecus Olfersi.* Fig. 518. *Sternoptyx diaphana.*

On Growth and Form, D'Arcy Thompson 1915



Non-rigid transformations

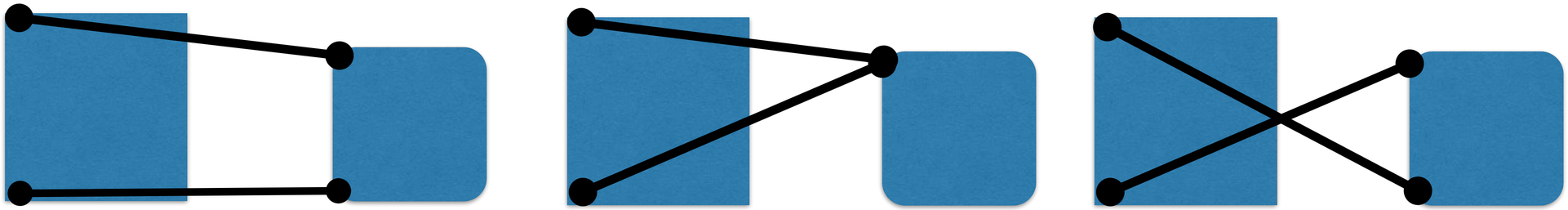
How to characterize a non-rigid transformation?

Global parameterization

- affine, thin-plate spline transformations

Local parameterization

- Low distortion
- want fewer many-to-one and one-to-many matches, criss-crossing etc
- Not that different from estimating a smooth optical flow!



low distortion

high distortion

Shape Matching and Object Recognition Using Shape Contexts

Serge Belongie, *Member*, *IEEE*, Jitendra Malik, *Member*, *IEEE*, and Jan Puzicha

Abstract—We present a novel approach to measuring similarity between shapes and exploit it for object recognition. In our framework, the measurement of similarity is preceded by 1) solving for correspondences between points on the two shapes, 2) using the correspondences to estimate an aligning transform. In order to solve the correspondence problem, we attach a descriptor, the *shape context*, to each point. The shape context at a reference point captures the distribution of the remaining points relative to it, thus offering a globally discriminative characterization. Corresponding points on two similar shapes will have similar shape contexts, enabling us to solve for correspondences as an optimal assignment problem. Given the point correspondences, we estimate the transformation that best aligns the two shapes; regularized thin-plate splines provide a flexible class of transformation maps for this purpose. The dissimilarity between the two shapes is computed as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform. We treat recognition in a nearest-neighbor classification framework as the problem of finding the stored prototype shape that is maximally similar to that in the image. Results are presented for silhouettes, trademarks, handwritten digits, and the COIL data set.

Index Terms—Shape, object recognition, digit recognition, correspondence problem, MPEG7, image registration, deformable templates.

Shape context descriptor

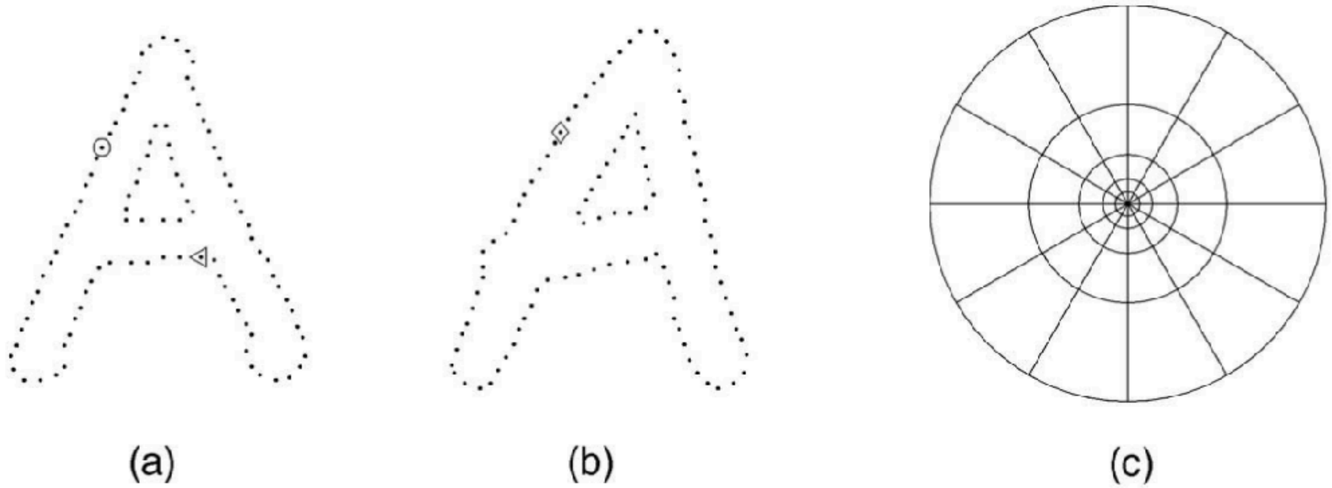
Extract boundaries and sample points

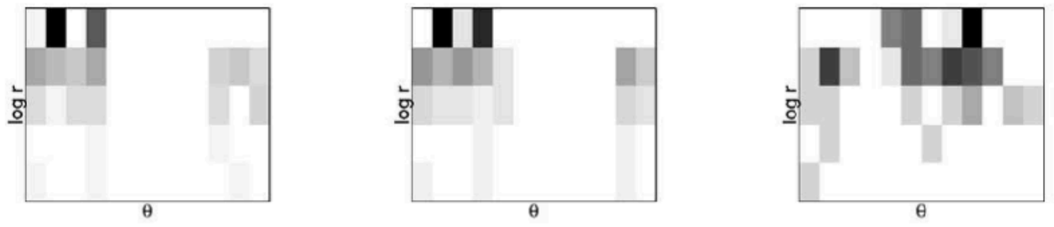
For each point build a log-polar histogram describing how many points belong to each bin.

Match points based on distance between the histograms (L1 or χ^2 distance used)

$$d_{L1}(h_1, h_2) = \sum_i |h_1[i] - h_2[i]|$$

$$d_{\chi^2}(h_1, h_2) = \frac{1}{2} \sum_i \frac{(h_1[i] - h_2[i])^2}{h_1[i] + h_2[i]}$$



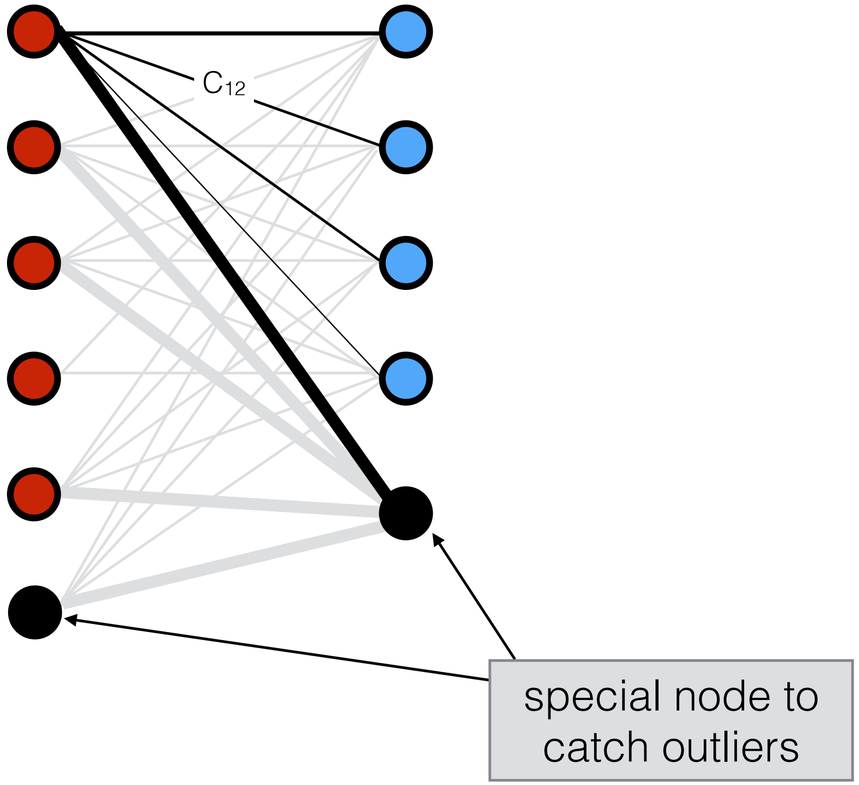


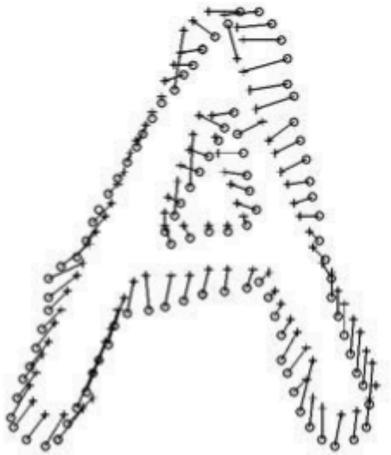
(d) (e) (f)

Shape matching

Solve a bipartite (or Hungarian) matching problem

$O(N^3)$ for matching with N nodes





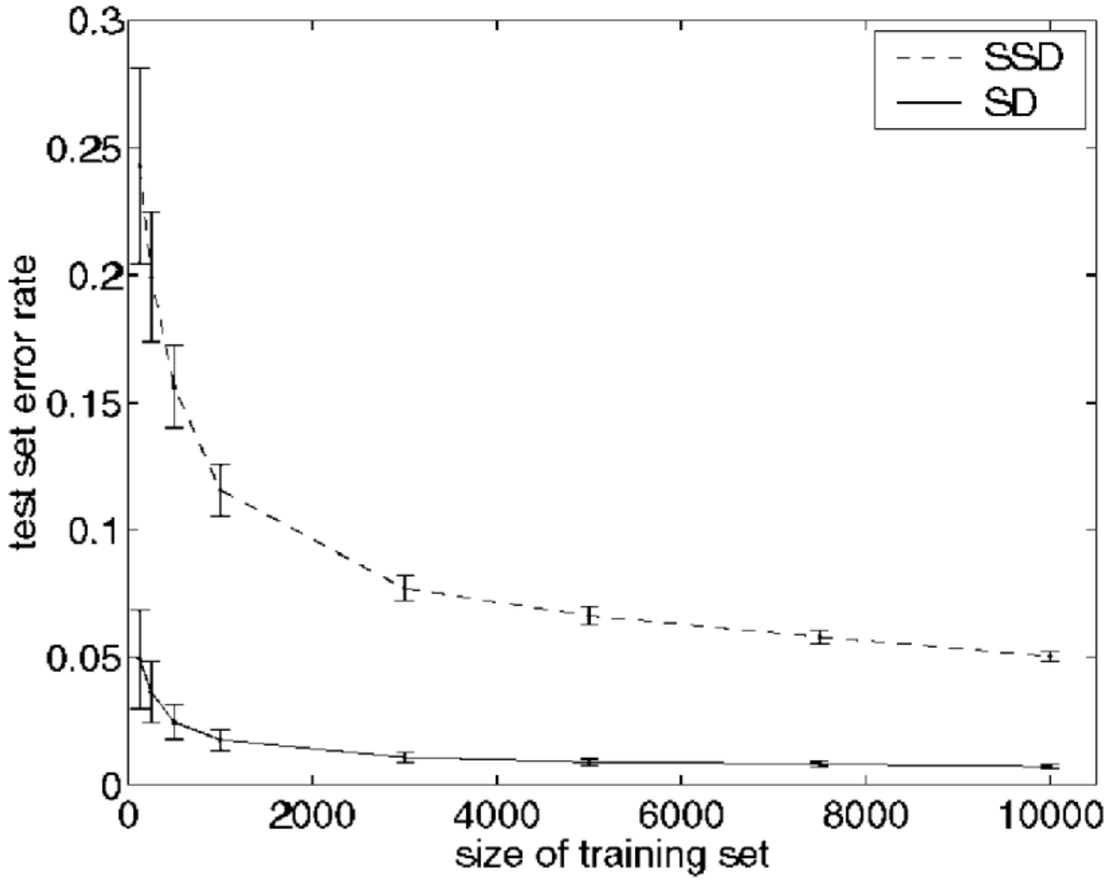
c_{ij} = cost of matching node i to j

MNIST classification

Performance using 1-nearest neighbor.

For each test example, find the training image with the smallest distance and assign its label.

MNIST dataset (0-9 digits)



Further thoughts and readings …

Chapters 6, 7 and 9 from Richard Szeliski's book.

Shape matching references

- Shape matching and object recognition using shape contexts, Belongie, Malik and Puzicha, PAMI 2002 (**paper**)
- Hierarchical matching of deformable shapes, Felzenszwalb and Schwartz, CVPR 2007 (**paper**)
- David Nister's Vocabulary Tree **paper**
- Shape matching and object recognition using low distortion correspondences, A.C. Berg, T.L. Berg, J. Malik, CVPR 2005 (**paper**)

Web demos from Oxford VGG group

- **Video google**, **Oxford building search**, **Sculpture retrieval**