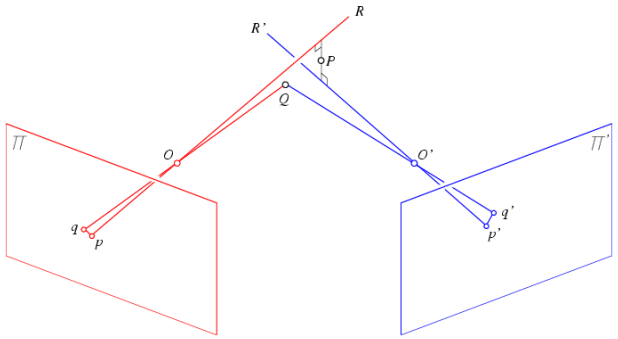
Large-scale Instance Retrieval

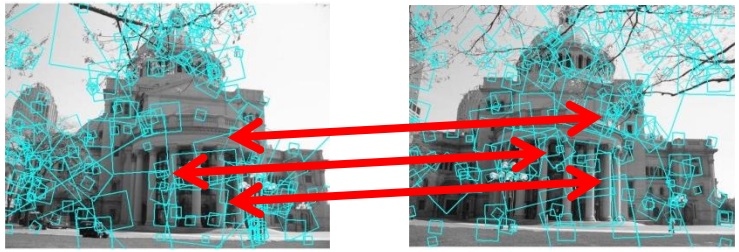
Computer Vision

James Hays

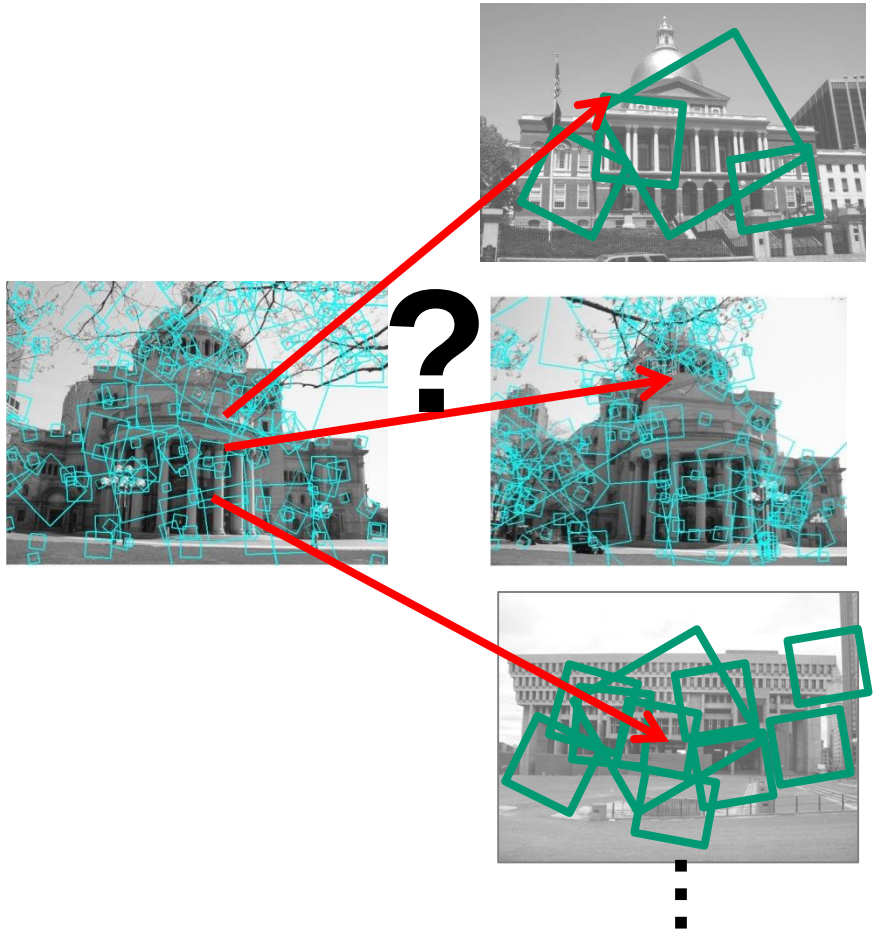
Many slides from Derek Hoiem and Kristen Grauman

Multi-view matching





vs



Matching two given views for depth

Search for a matching view for recognition

Inverted file index

Image #1

Image #2

Image #3

Word #	Image #
1	3
2 …	
7	1, 2
8	3
9	
10 …	
91	2

…

- Database images are loaded into the index mapping words to image numbers

Kristen Grauman

Inverted file index

New query image

Word #	Image #
1	3
2	
…	
7	1, 2
8	3
9	
10	
…	
91	2

- New query image is mapped to indices of database images that share a word.

Kristen Grauman

Inverted file index

- Key requirement for inverted file index to be efficient: sparsity
- If most pages/images contain most words then you're no better off than exhaustive search.
  - Exhaustive search would mean comparing the word distribution of a query versus every page.

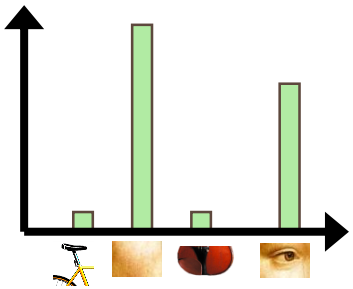
Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Comparing bags of words

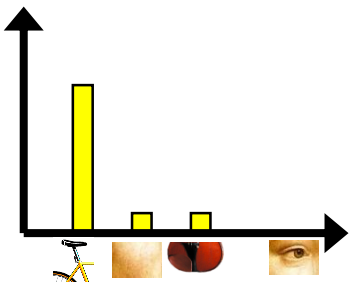
- Rank frames by normalized scalar product between their (possibly weighted) occurrence counts---*nearest neighbor* search for similar images.

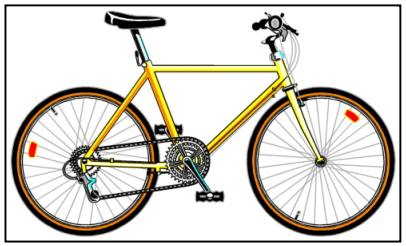
[1 8 1 4]



$\vec{d}_j$

[5 1 1 0]





$\vec{q}$

$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i}}$$

for vocabulary of *V* words

Comparing bags of words

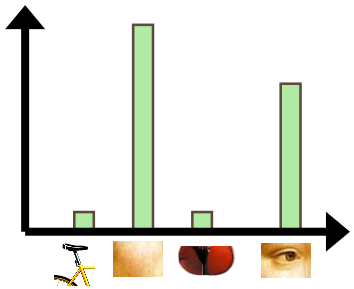
- Other common histogram comparisons:
  - Histogram intersection

$$\frac{\sum_{j=1}^{n} min(I_j, M_j)}{\sum_{j=1}^{n} M_j}$$

  - Chi squared

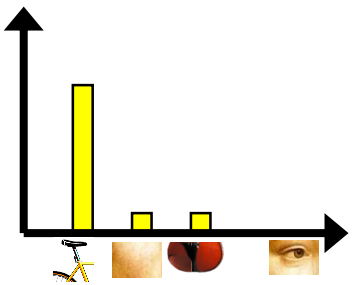
$$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

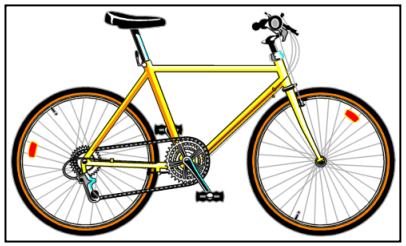
[1 8 1 4]



$\vec{d}_j$

[5 1 1 0]



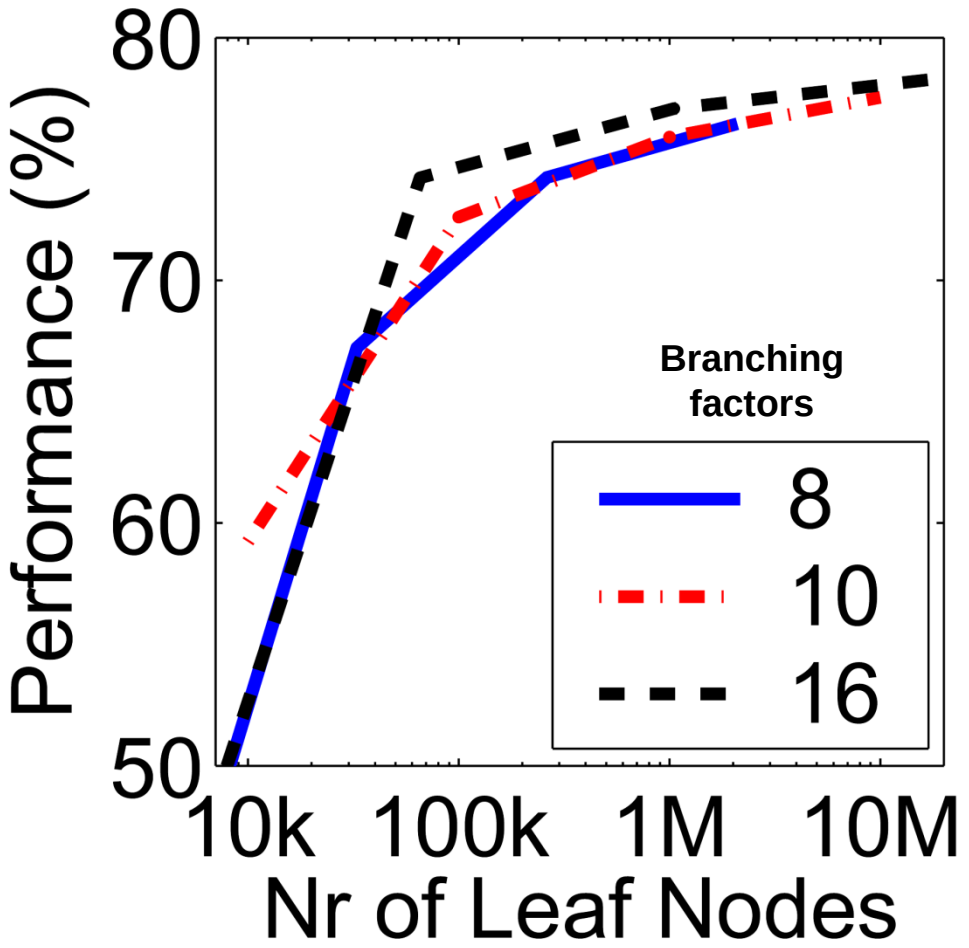


$\vec{q}$

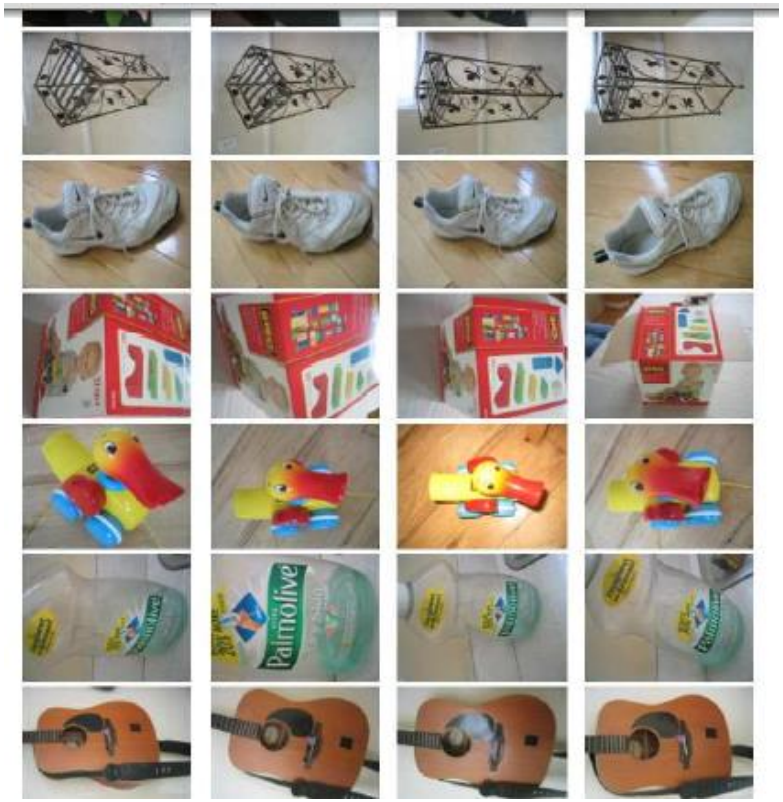
Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Vocabulary size



Results for recognition task with 6347 images

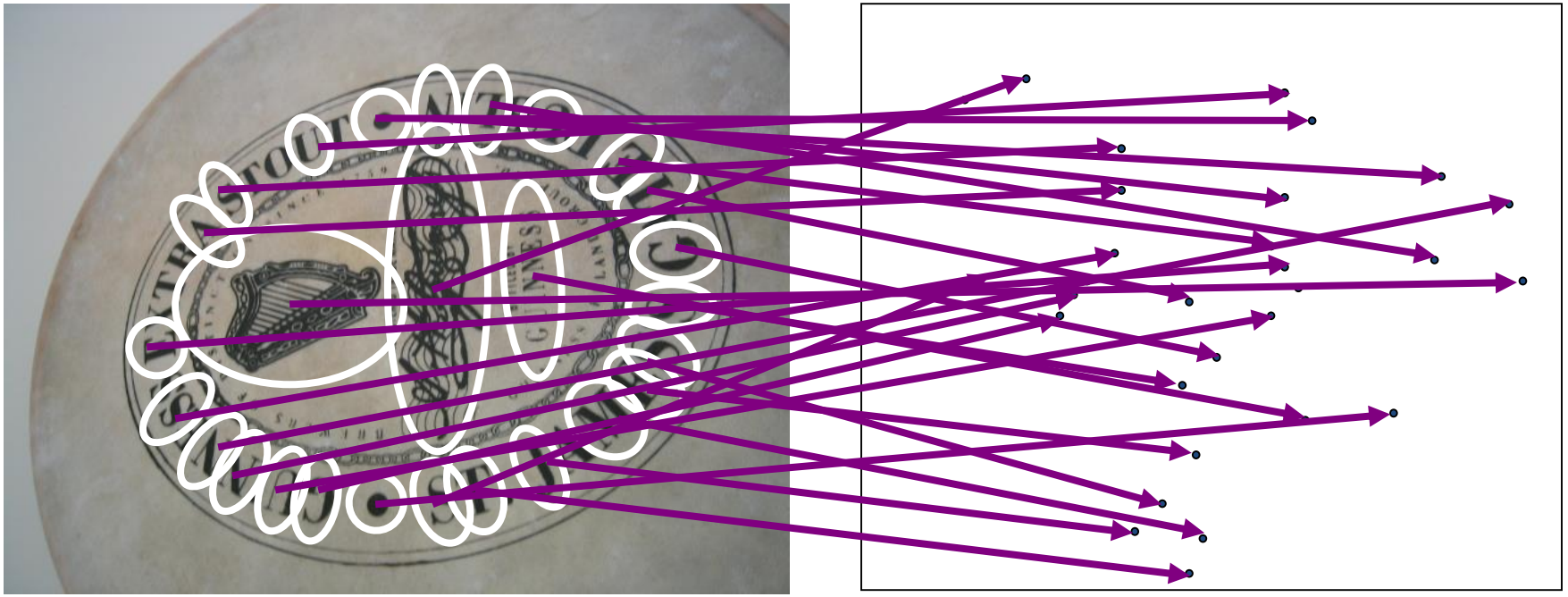


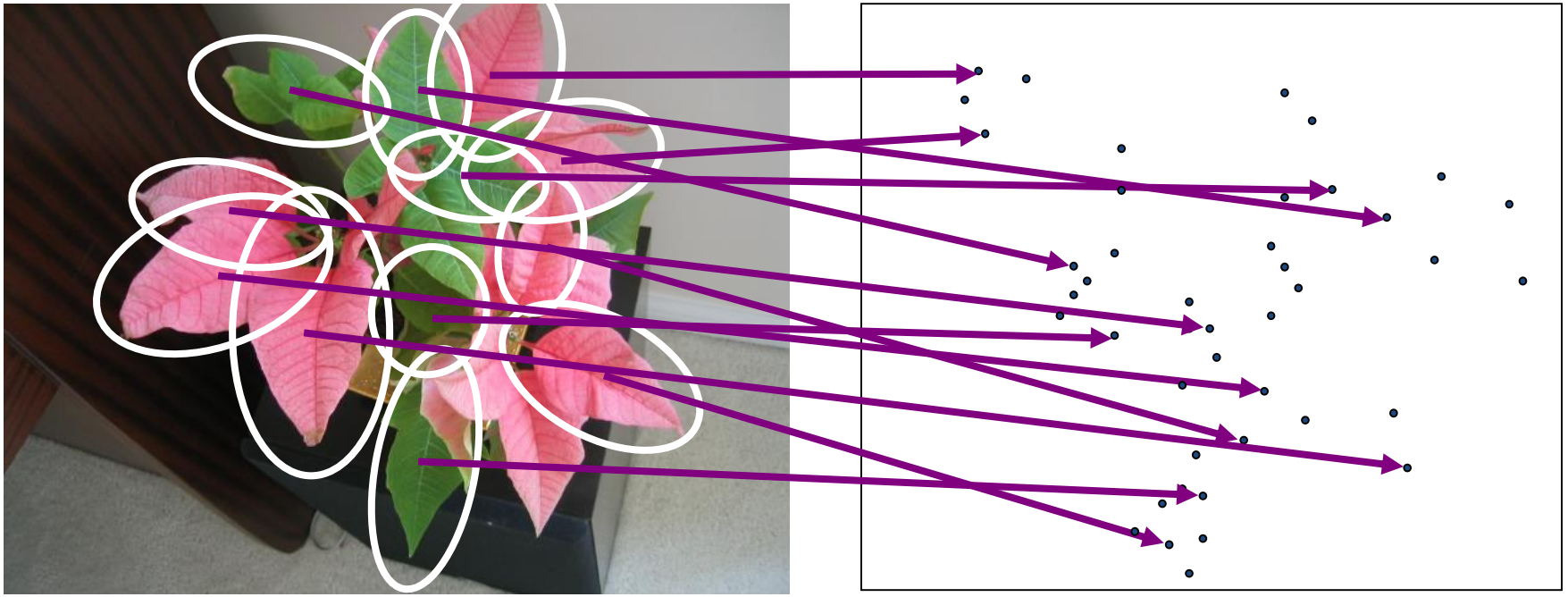
Influence on performance, sparsity

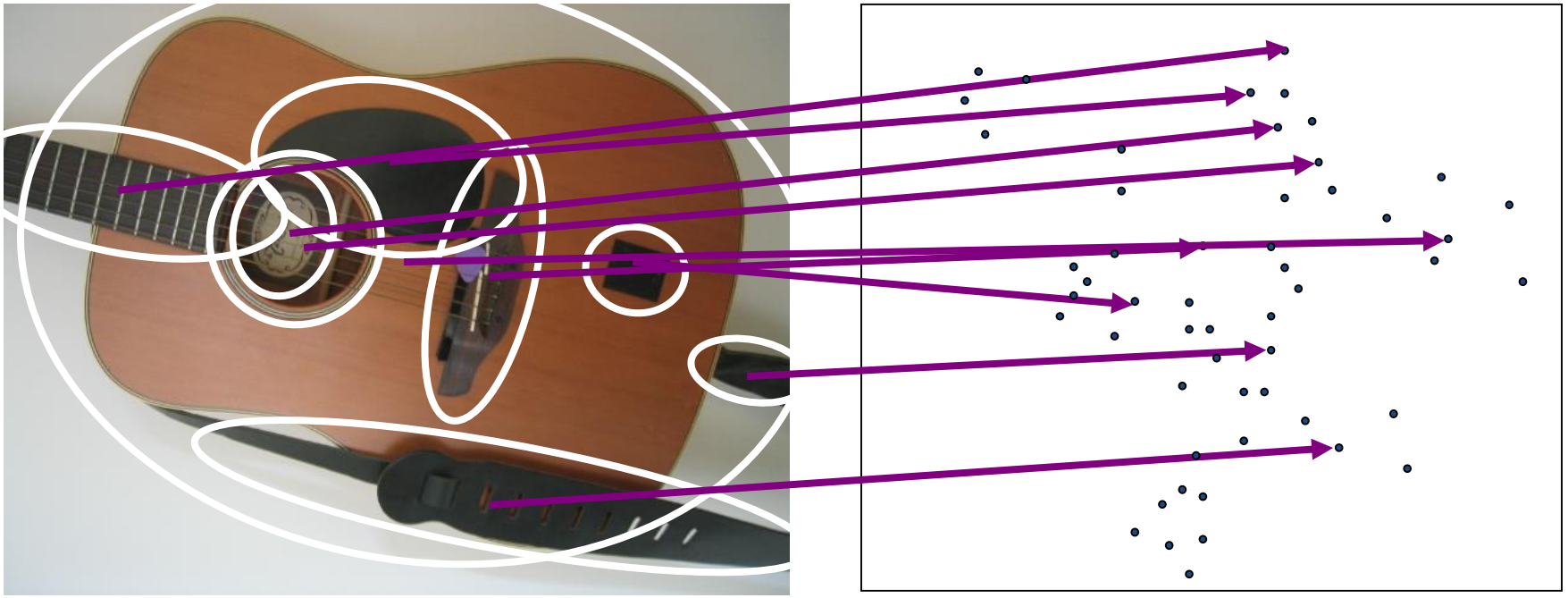
Nister & Stewenius, CVPR 2006

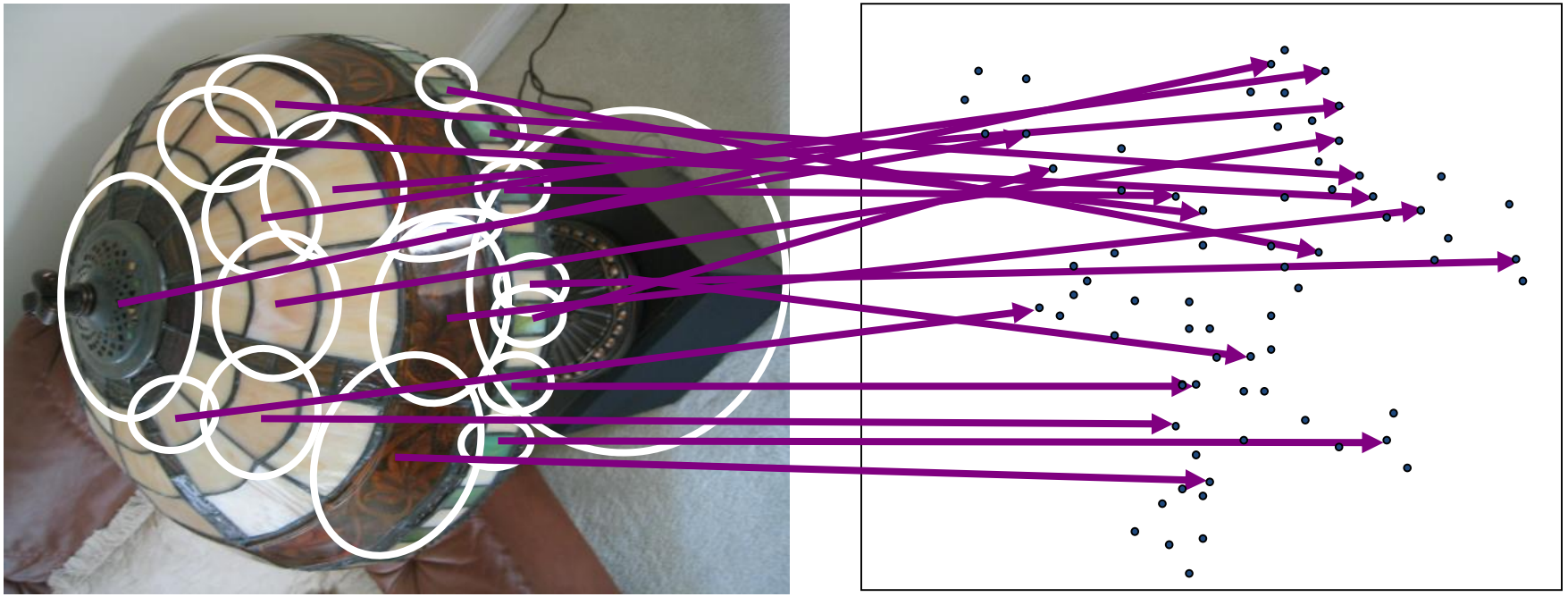
Recognition with K-tree

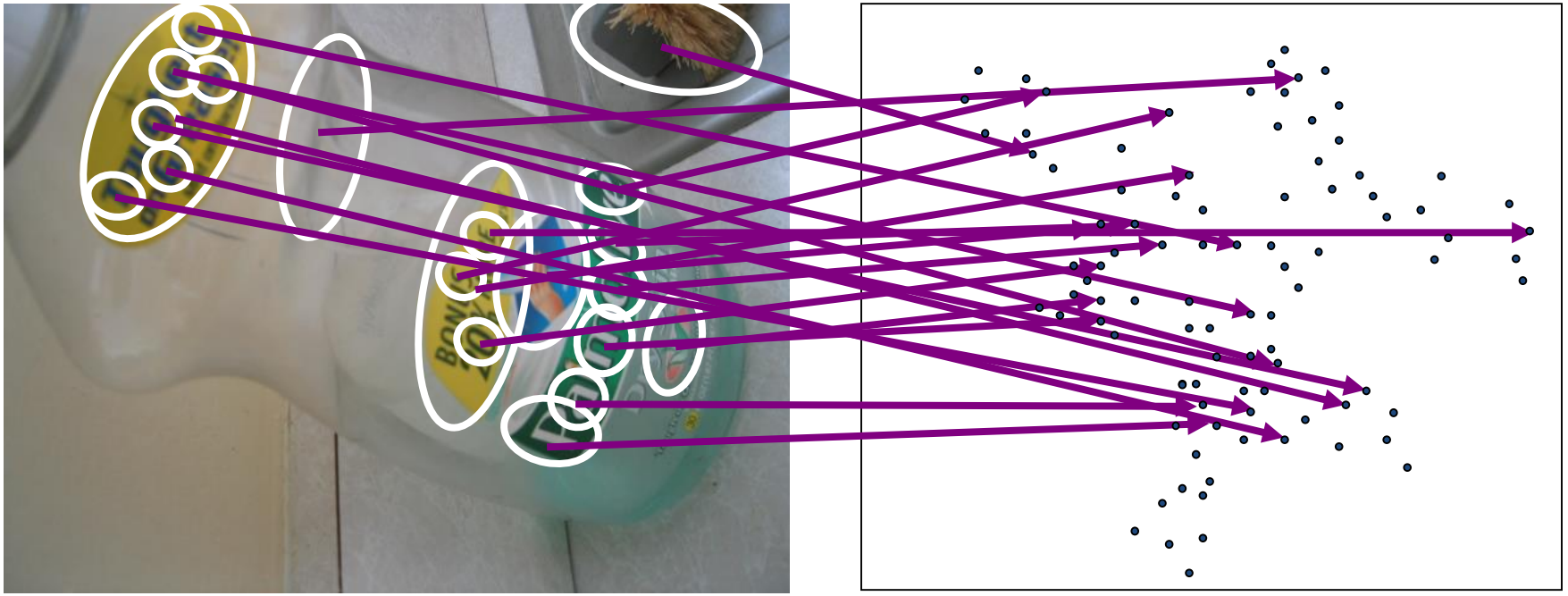
Following slides by David Nister (CVPR 2006)

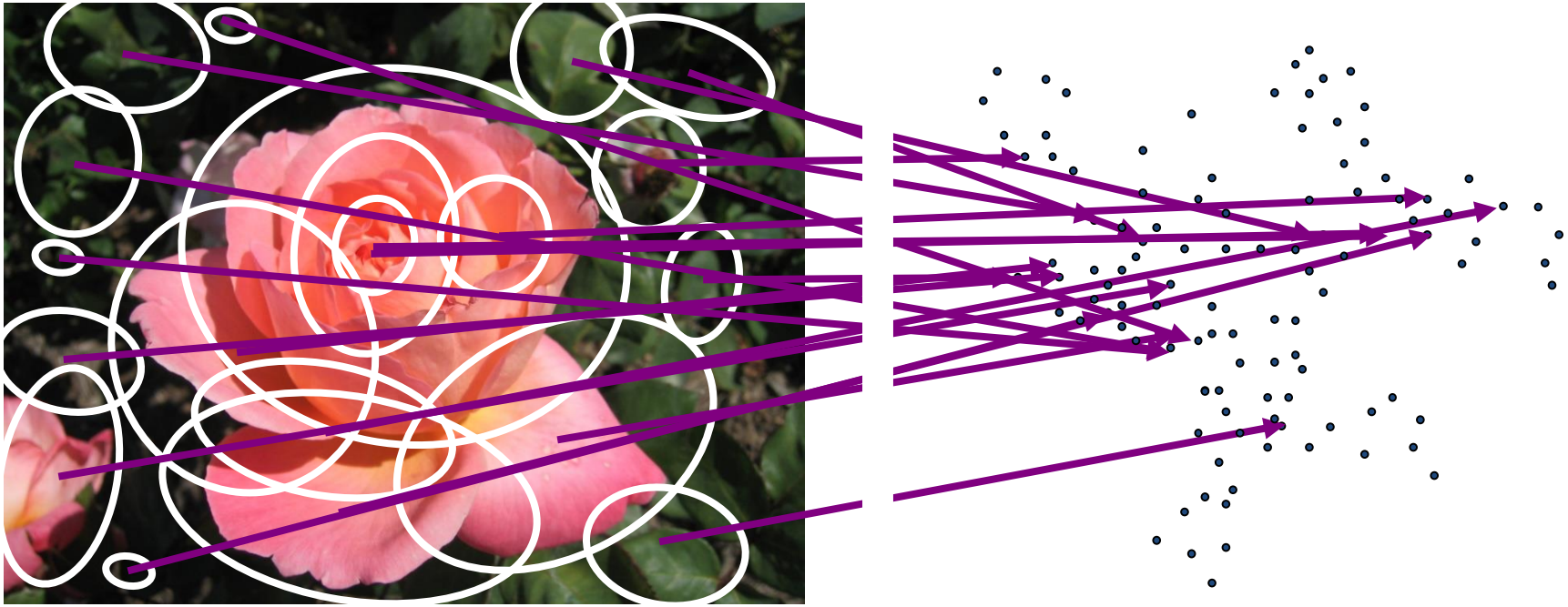


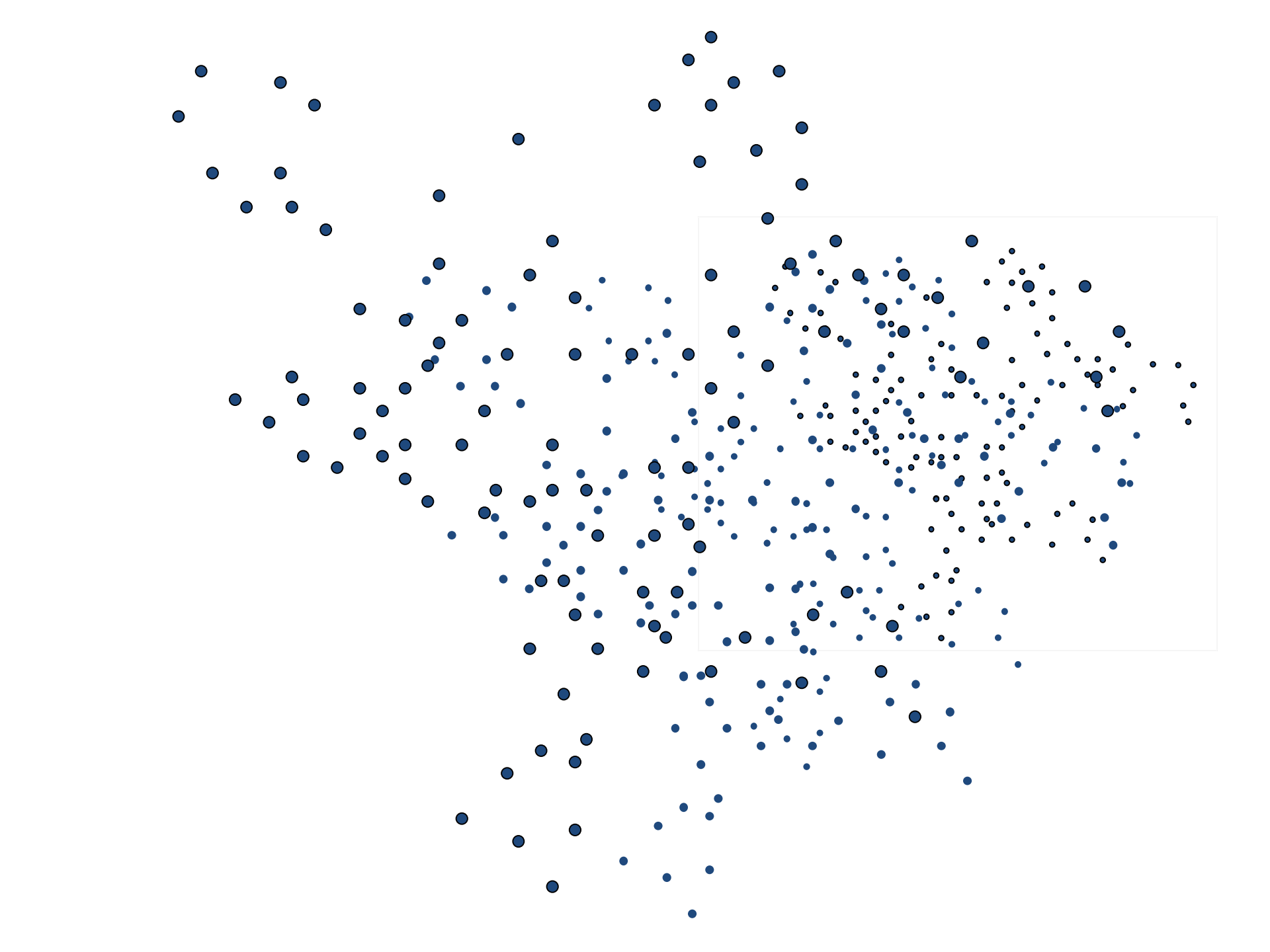


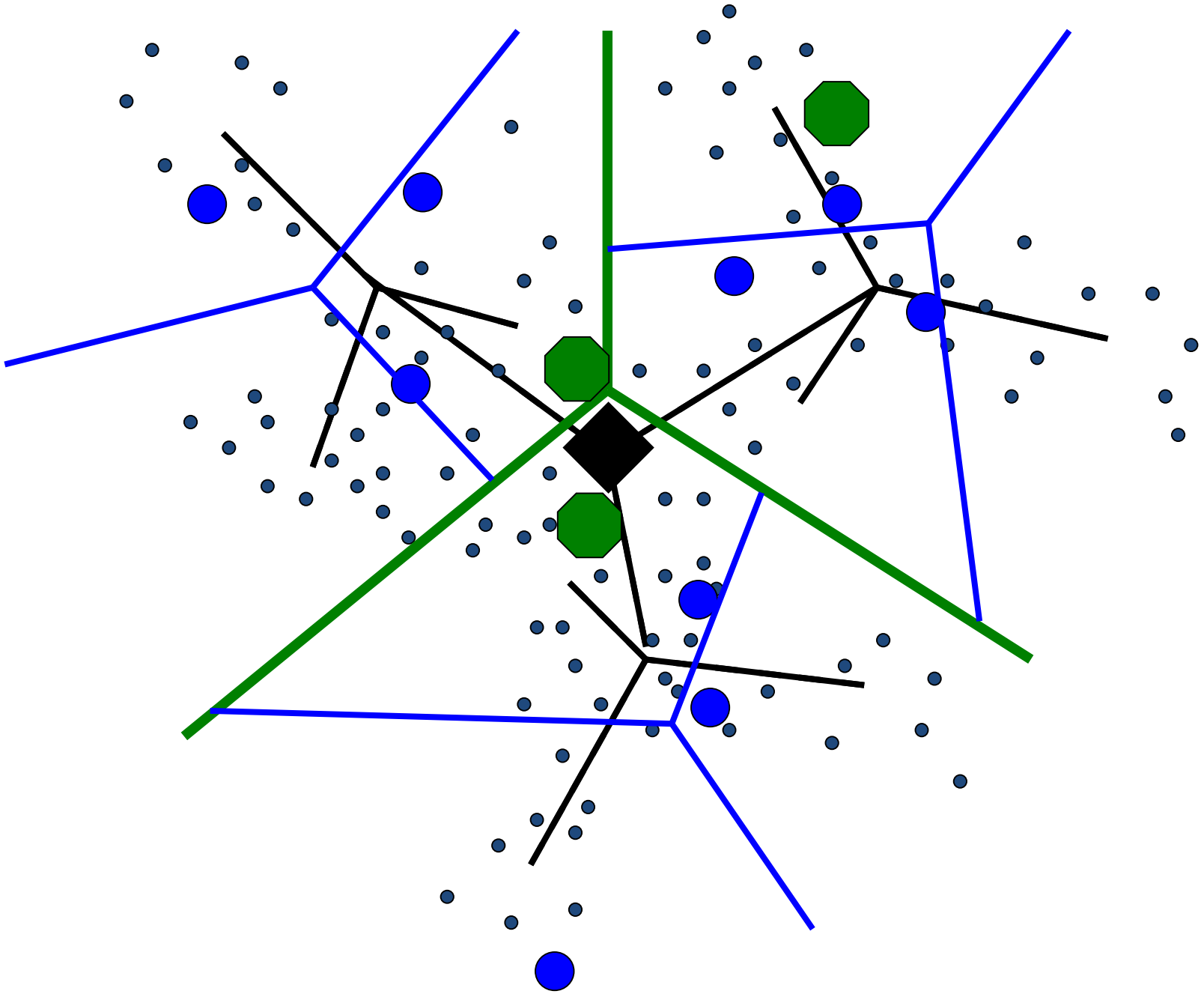


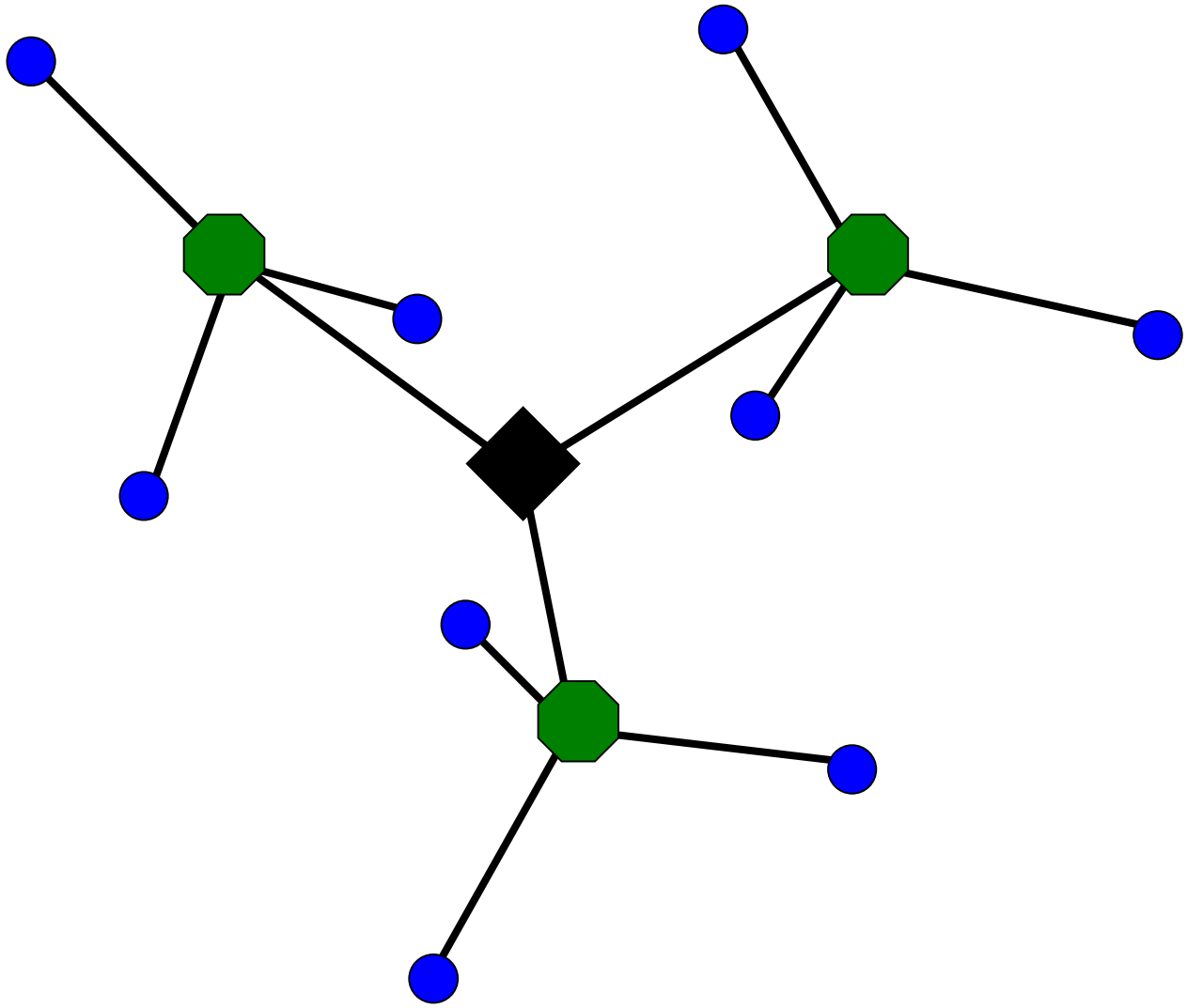


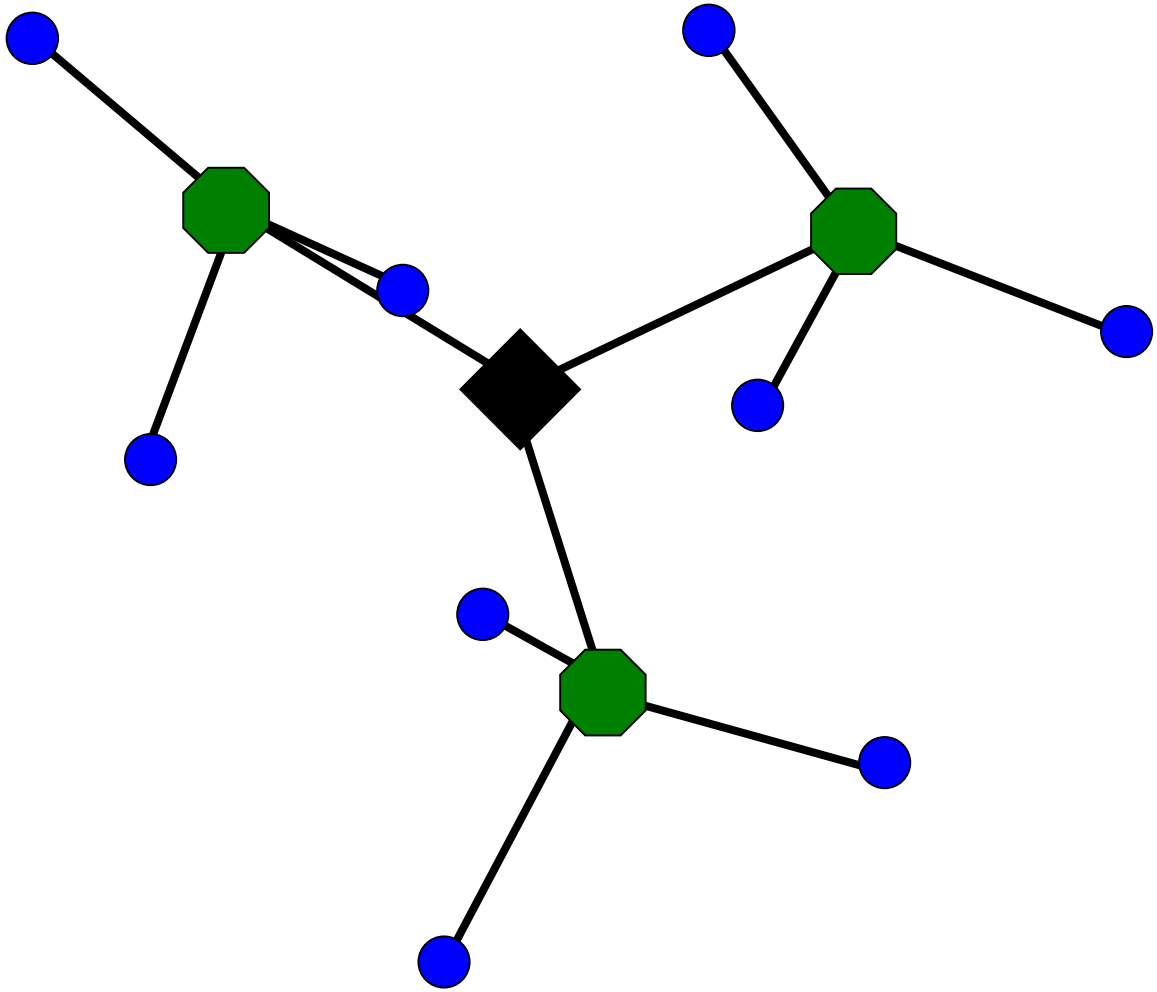


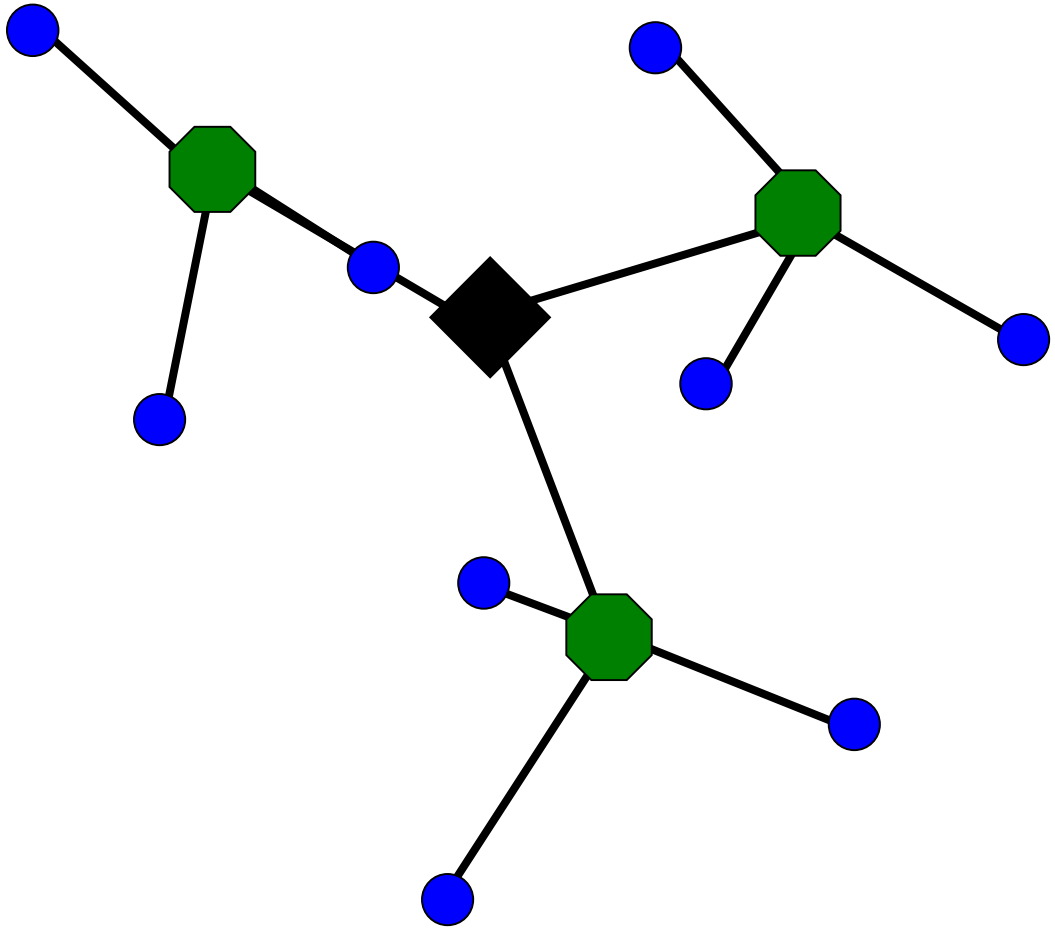


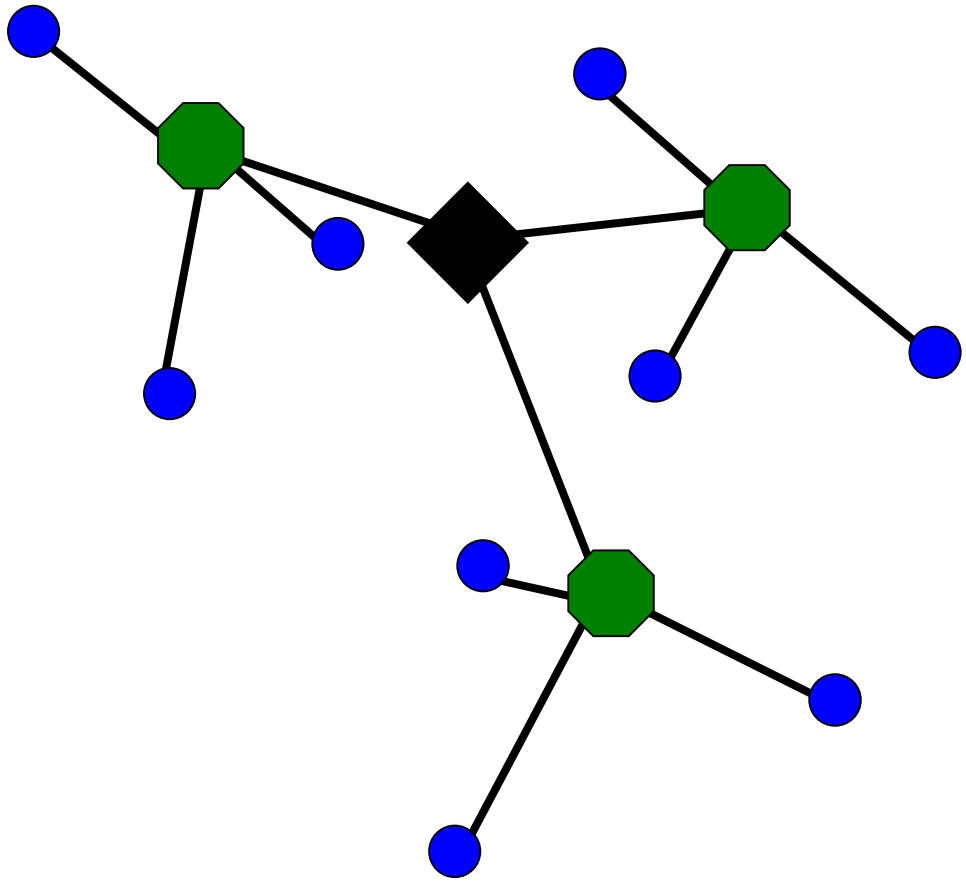


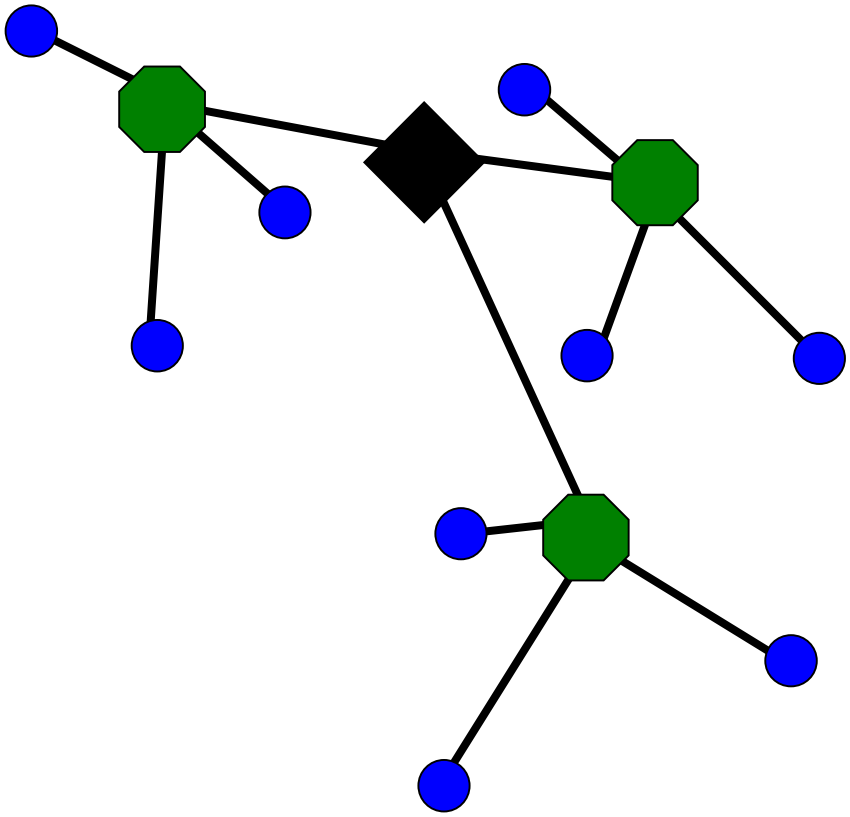


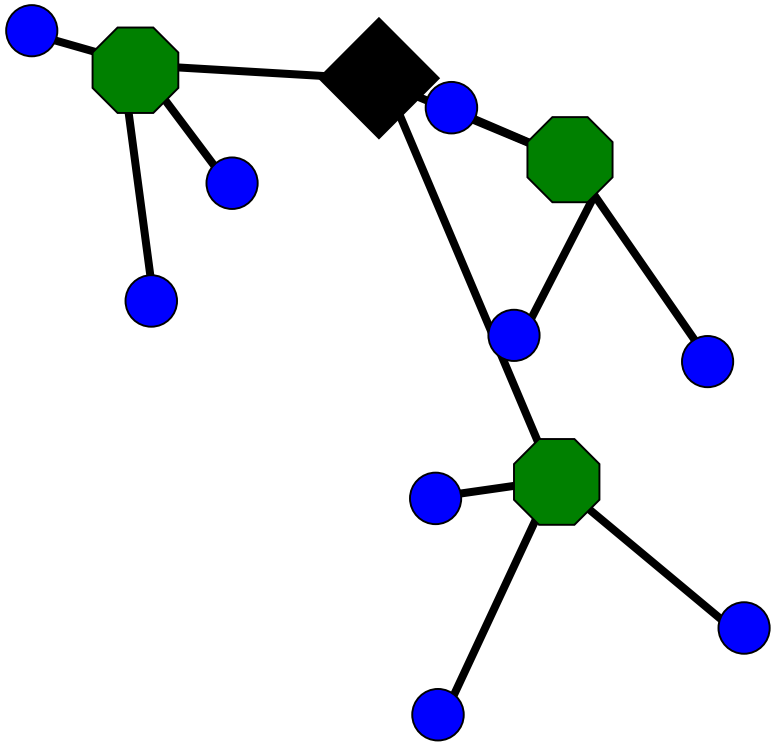


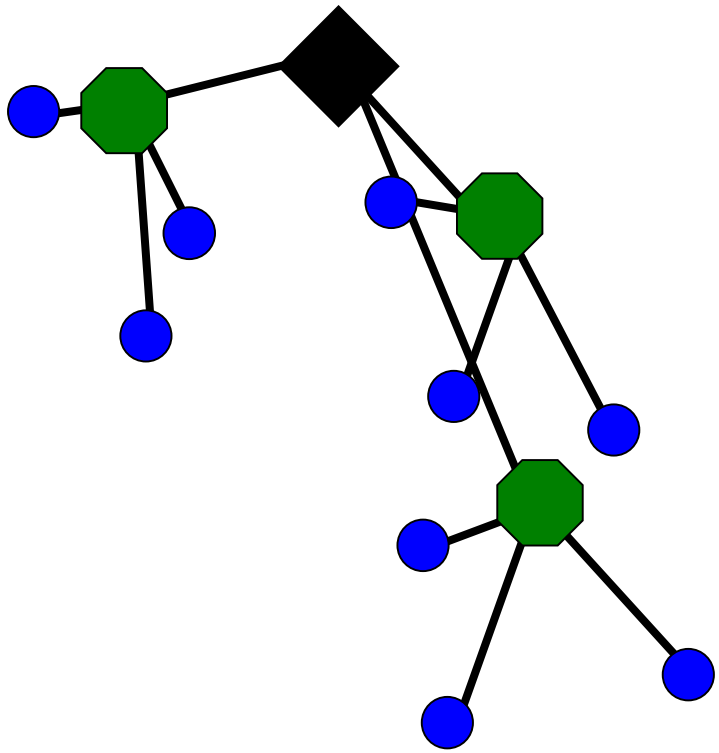


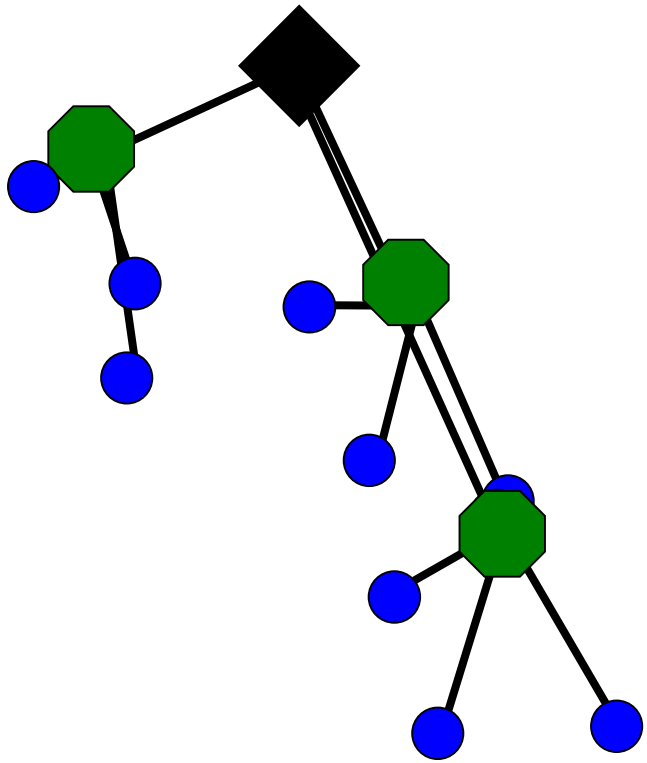


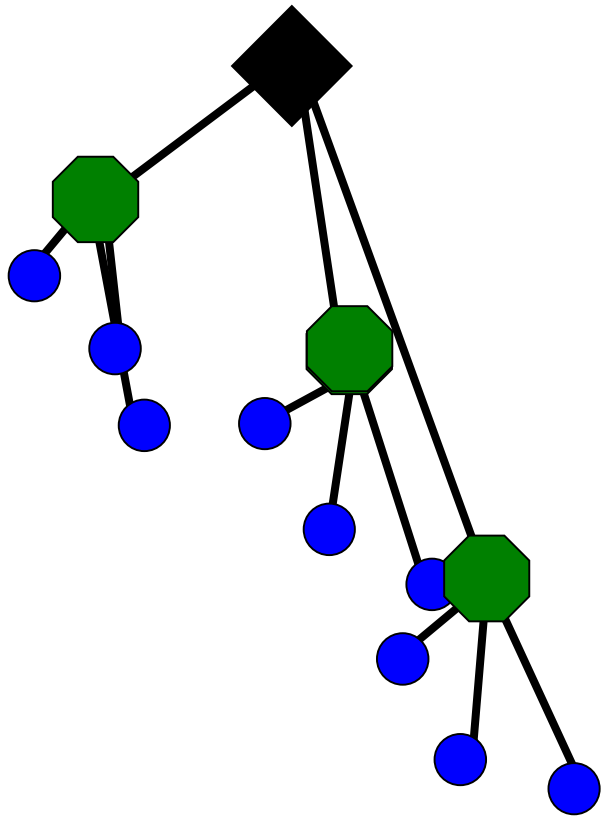


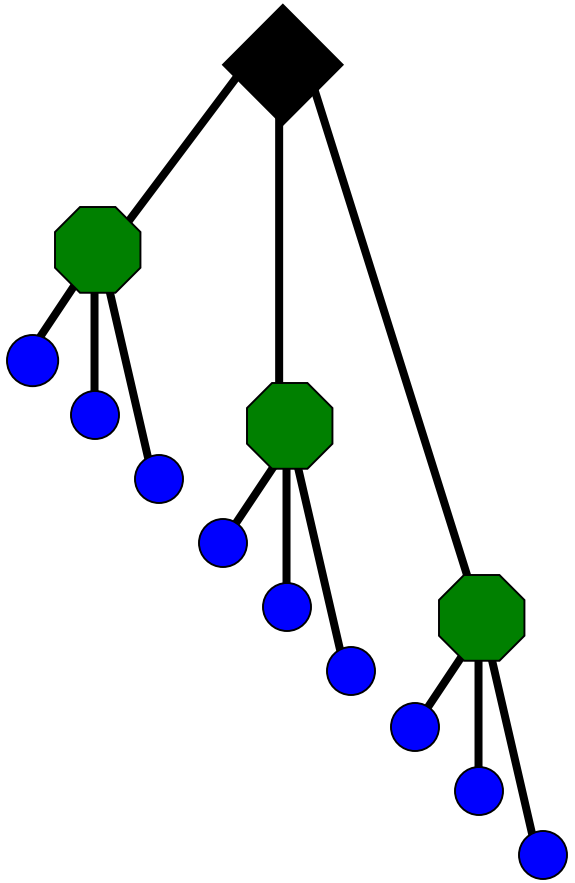


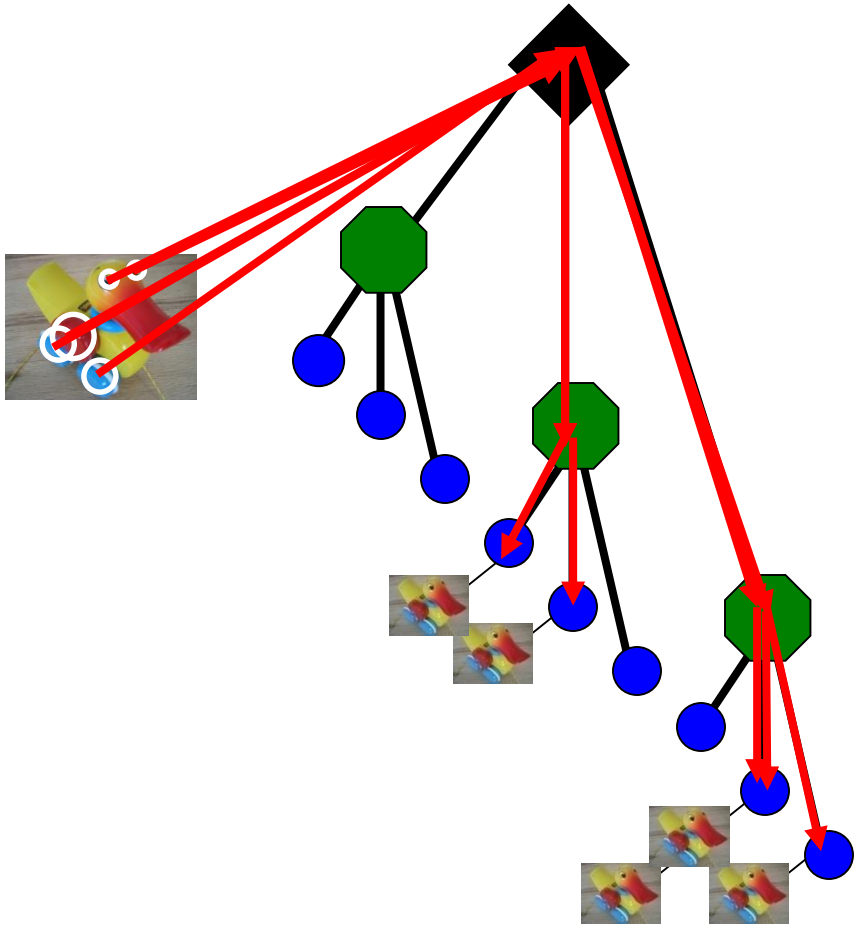


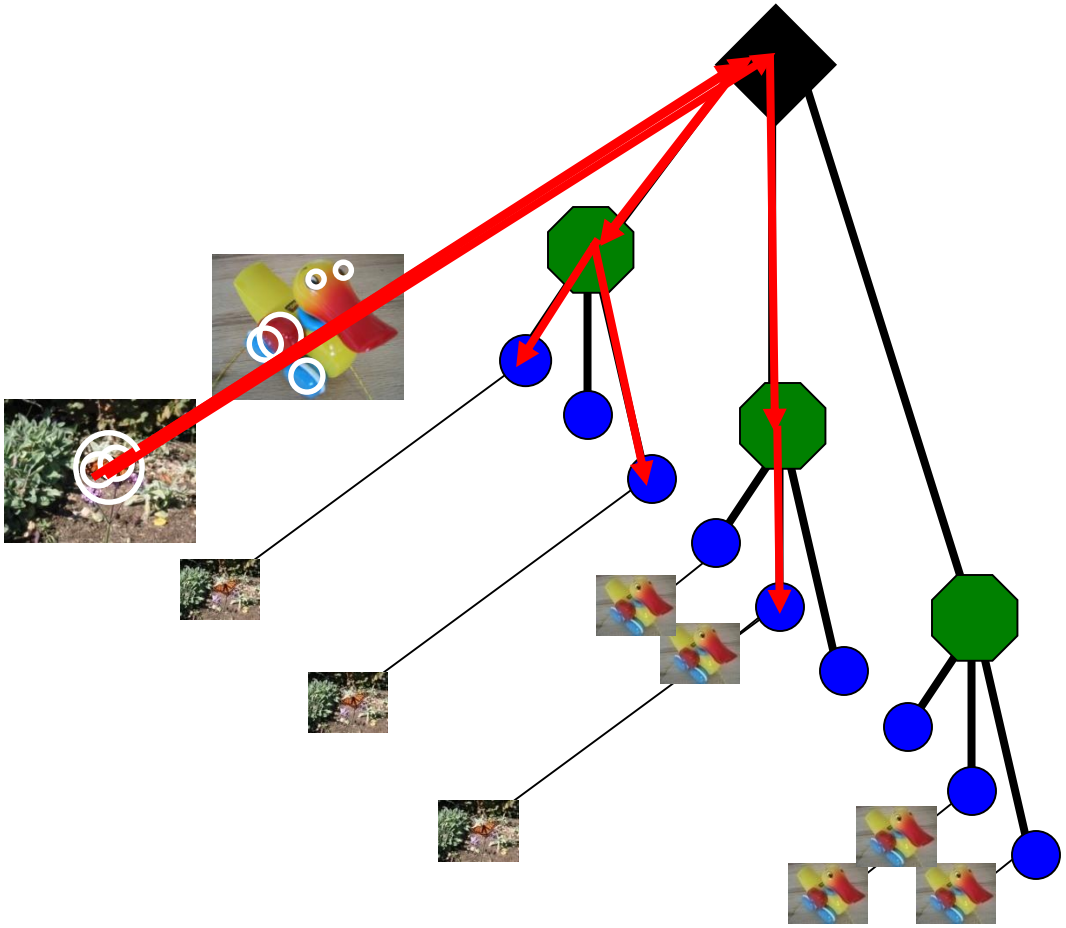


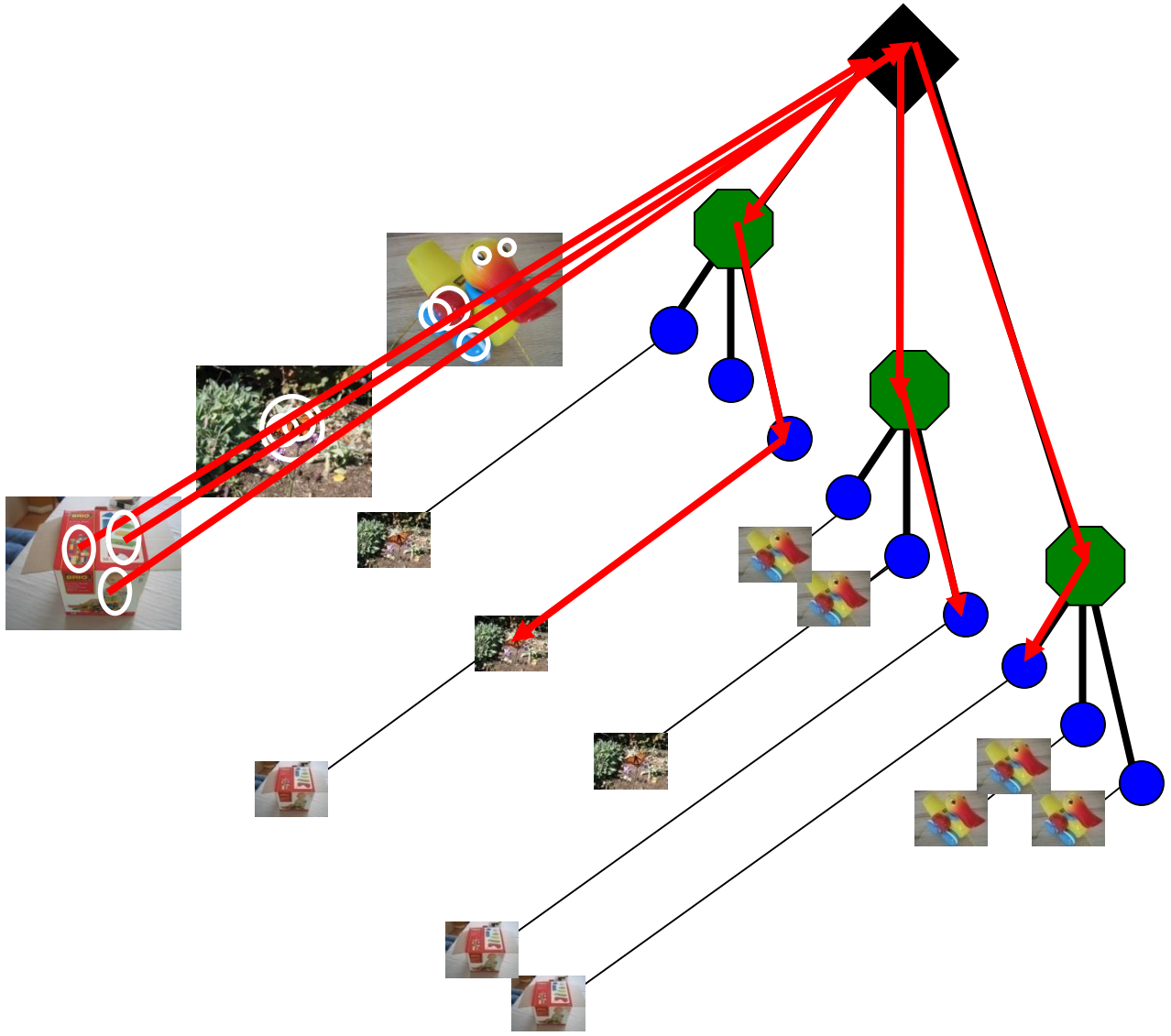


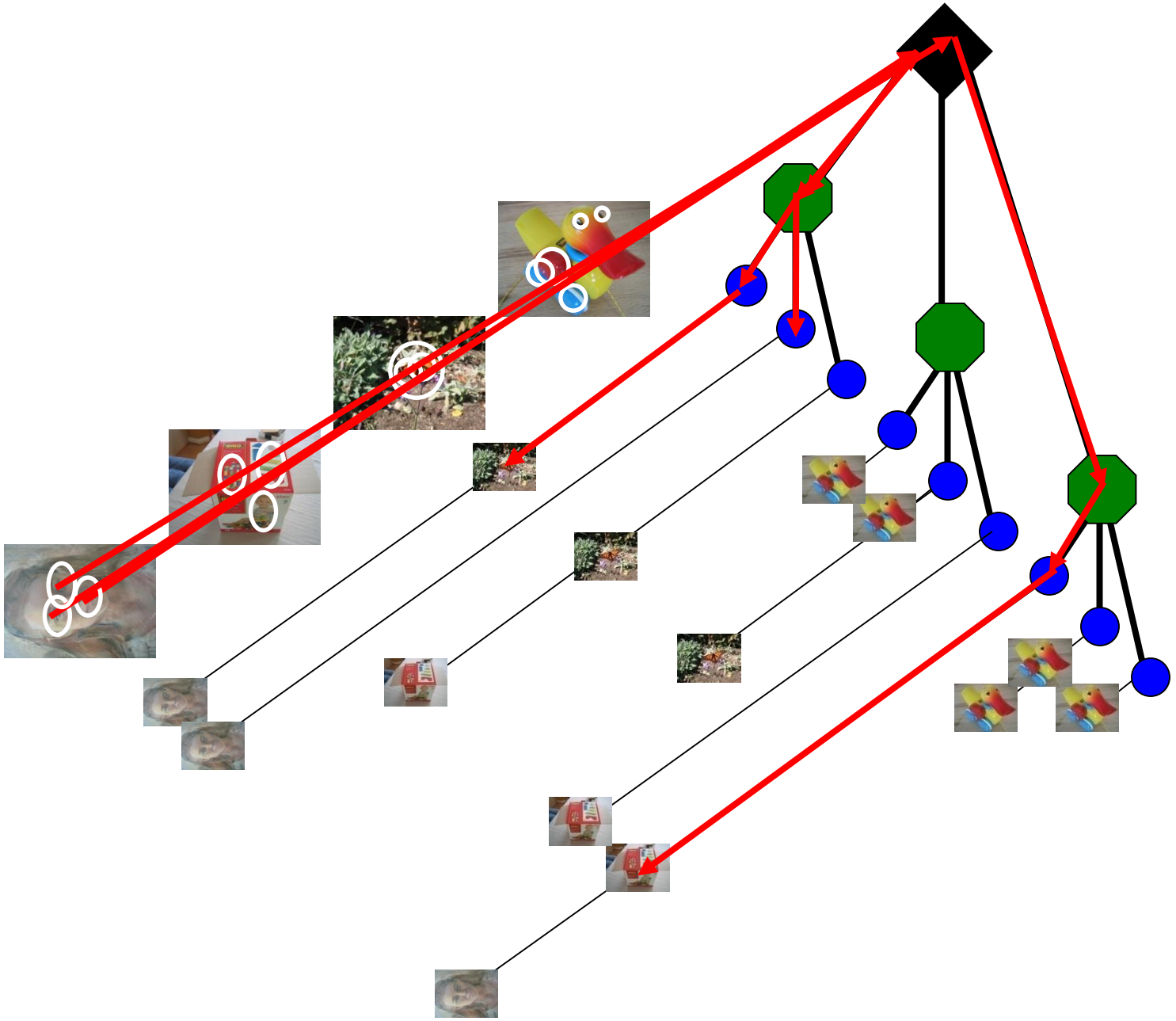


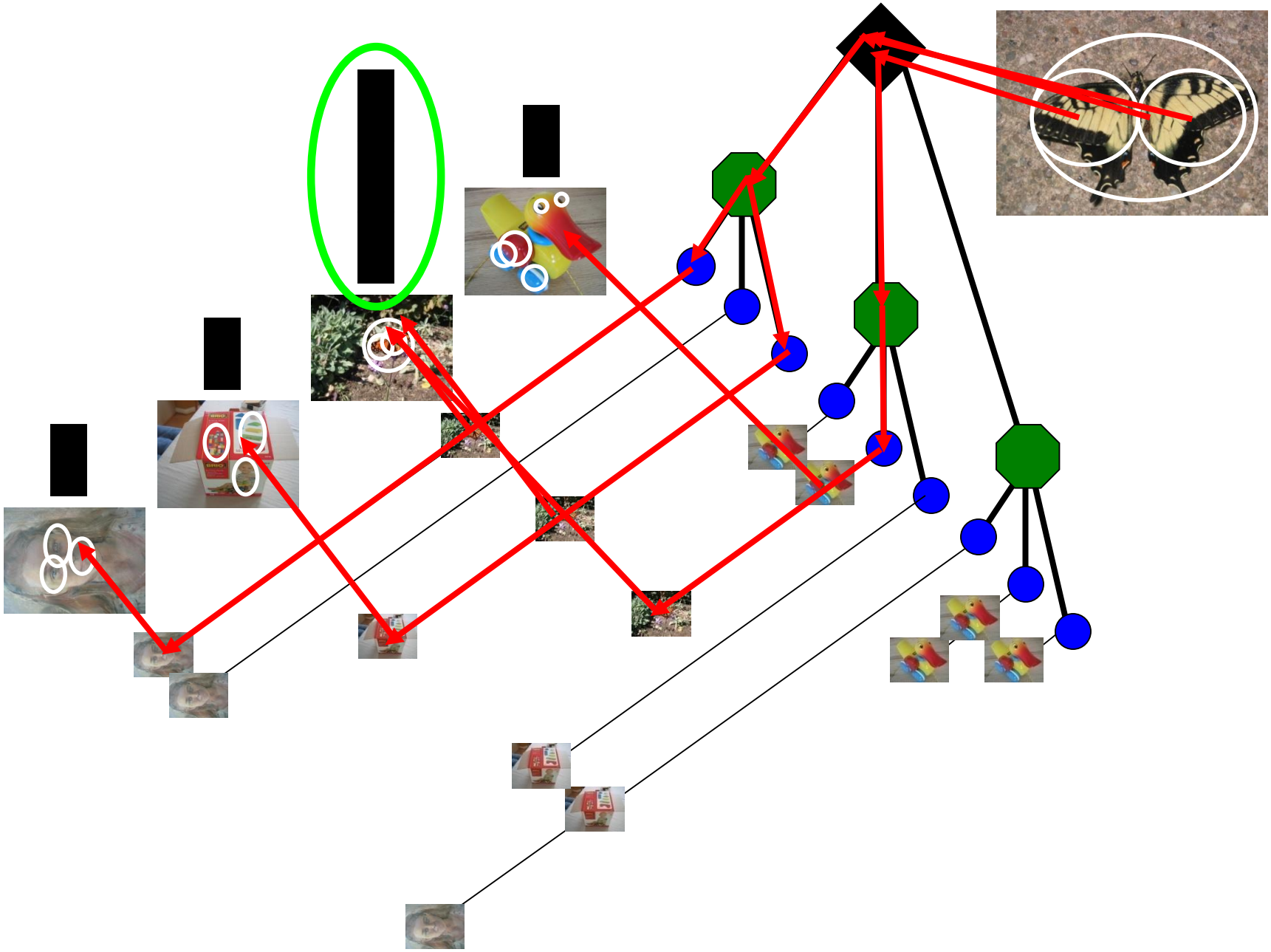












Vocabulary trees: complexity

Number of words given tree parameters: branching factor and number of levels

branching_factor^number_of_levels

Word assignment cost vs. flat vocabulary

O(k) for flat

$O(\log_{branching\_factor}(k) * branching\_factor)$

Is this like a kd-tree?

Yes, but with better partitioning and defeatist search.

This hierarchical data structure is lossy – you might not find your true nearest cluster.

110,000,000 Images in 5.8 Seconds

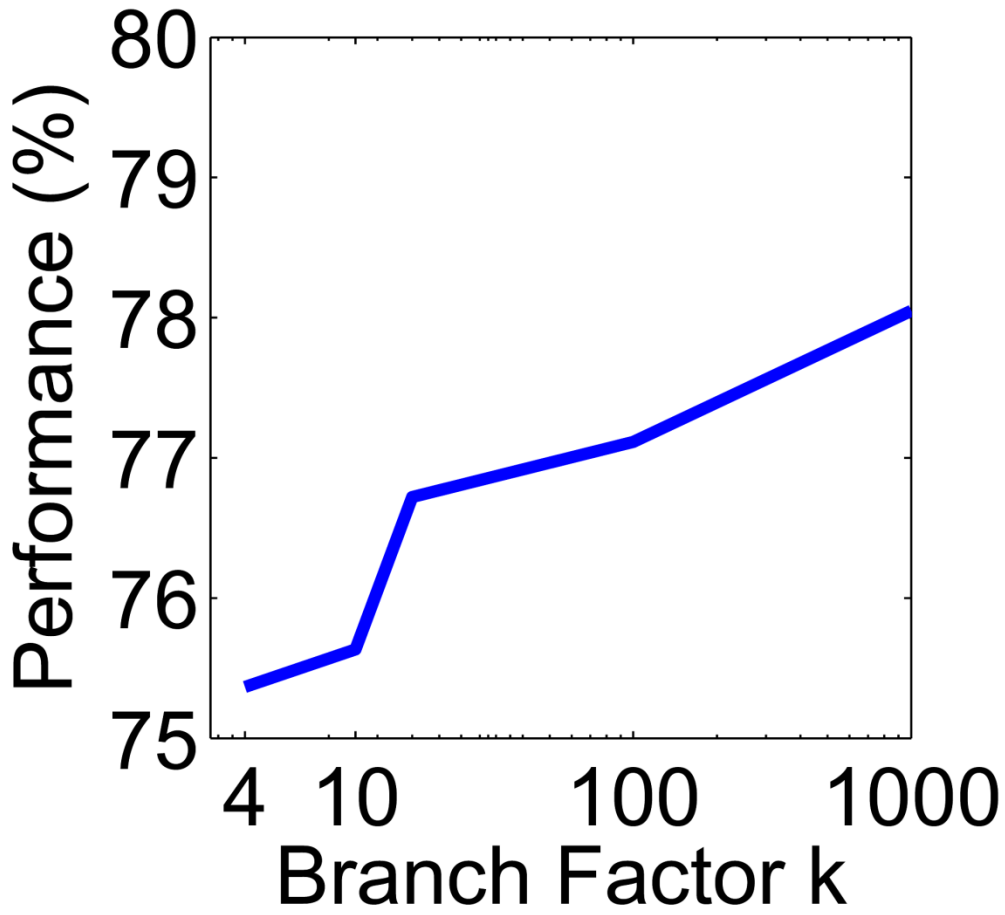
Slide Credit: Nister

Slide Credit: Nister

Slide Credit: Nister

Slide Credit: Nister

Higher branch factor works better (but slower)



Visual words/bags of words

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides fixed dimensional vector representation for sets
+ very good results in practice

- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Can we be more accurate?

So far, we treat each image as containing a "bag of words", with no spatial information