



SAIR

Spatial AI & Robotics Lab

CSE 473/573-A

L21: RETRIEVAL

Chen Wang

Spatial AI & Robotics Lab

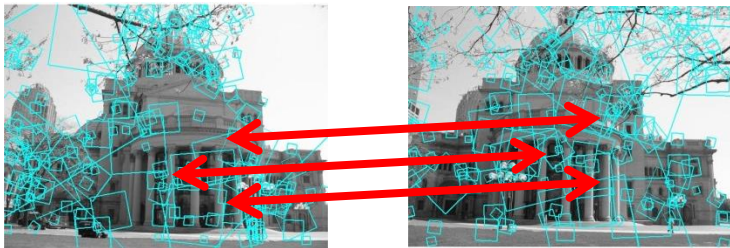
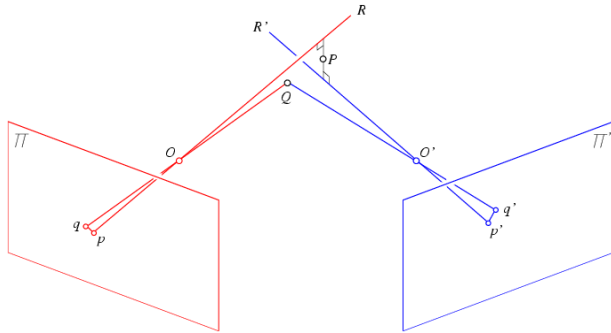
Department of Computer Science and Engineering



University at Buffalo The State University of New York

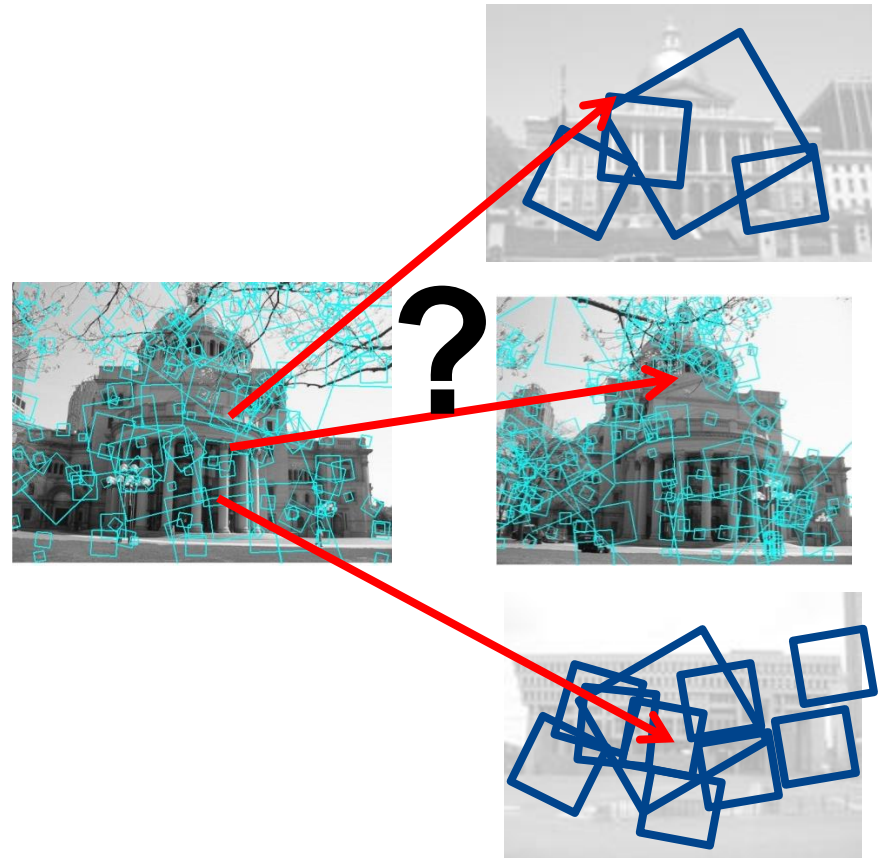
Multi-view matching (Recap)

Matching two given views for depth



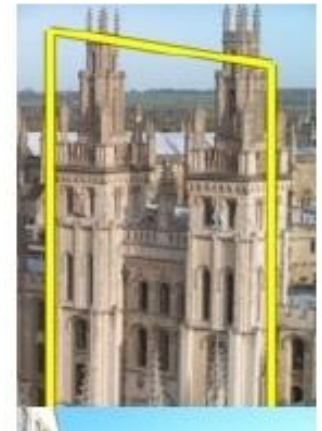
Search for a matching view for recognition

vs



Efficient Retrieval

How to quickly find images in a large database that match a given image region?



Local Retrieval

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification



Query
region

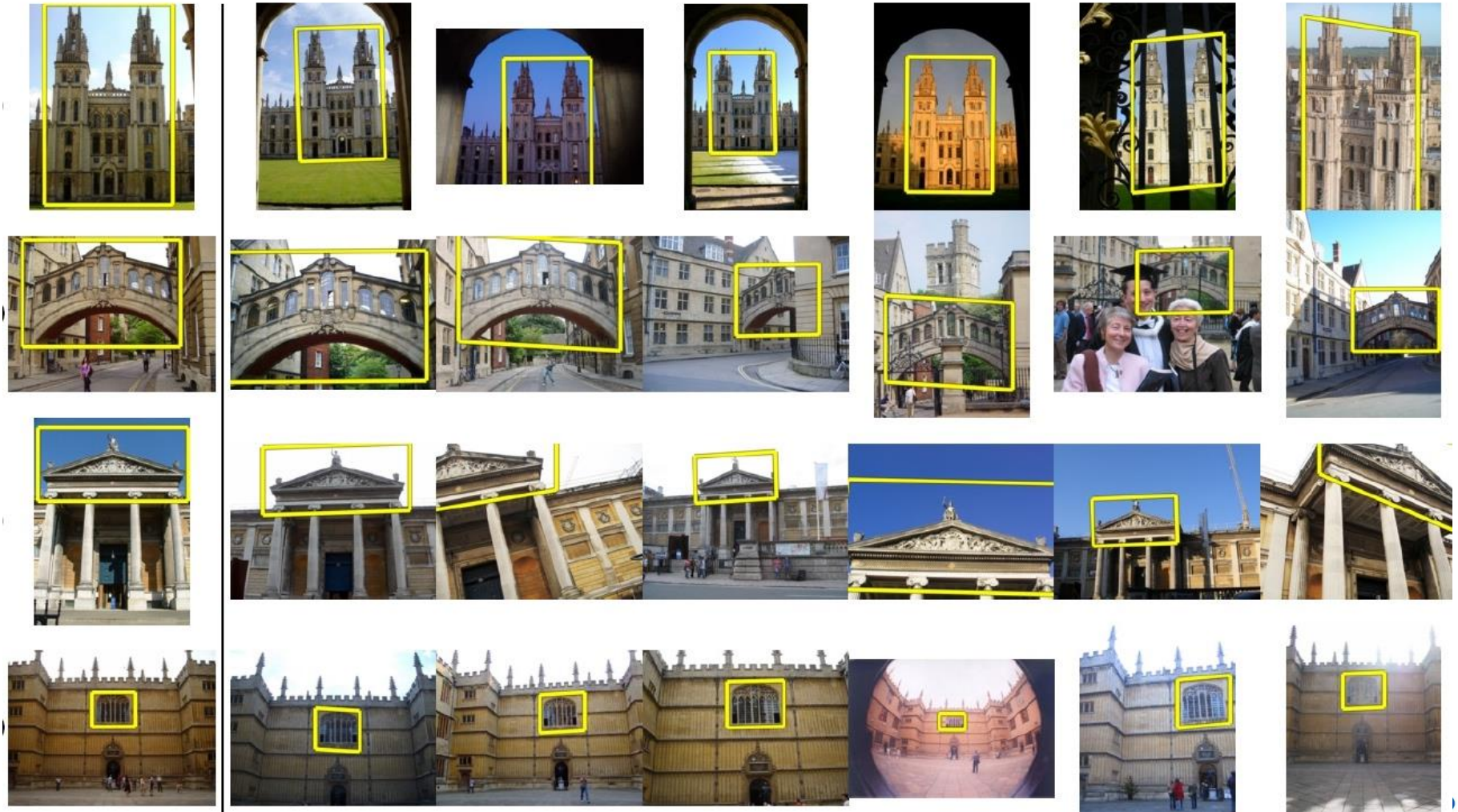


Retrieved frames

Application: Image Retrieval

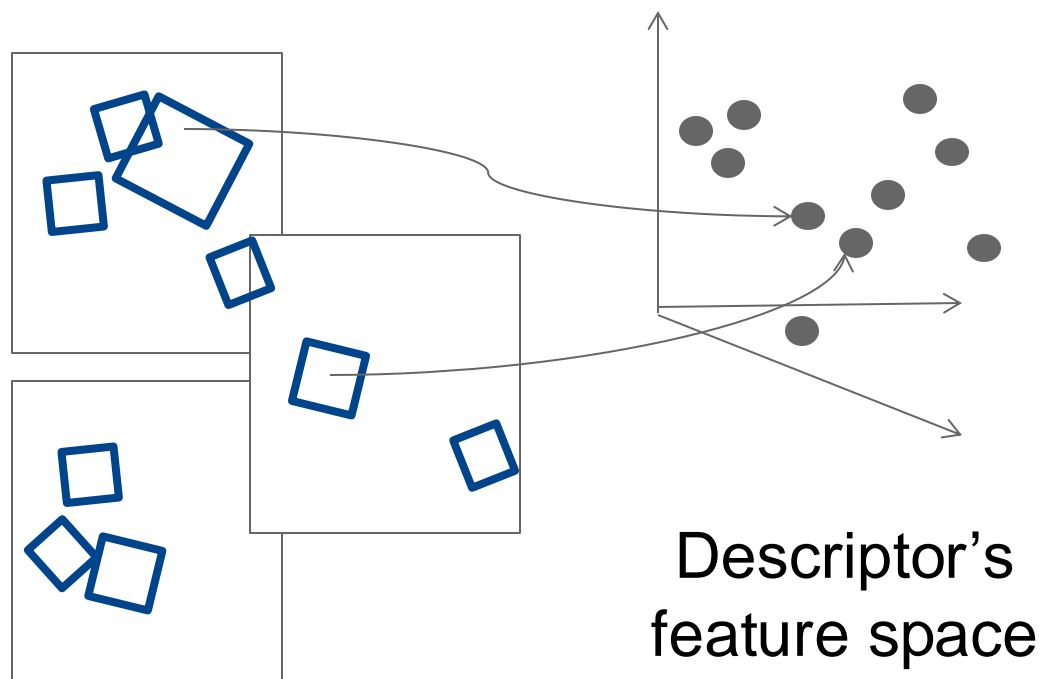
Query

Results from 5k Flickr images (demo available for 100k set)



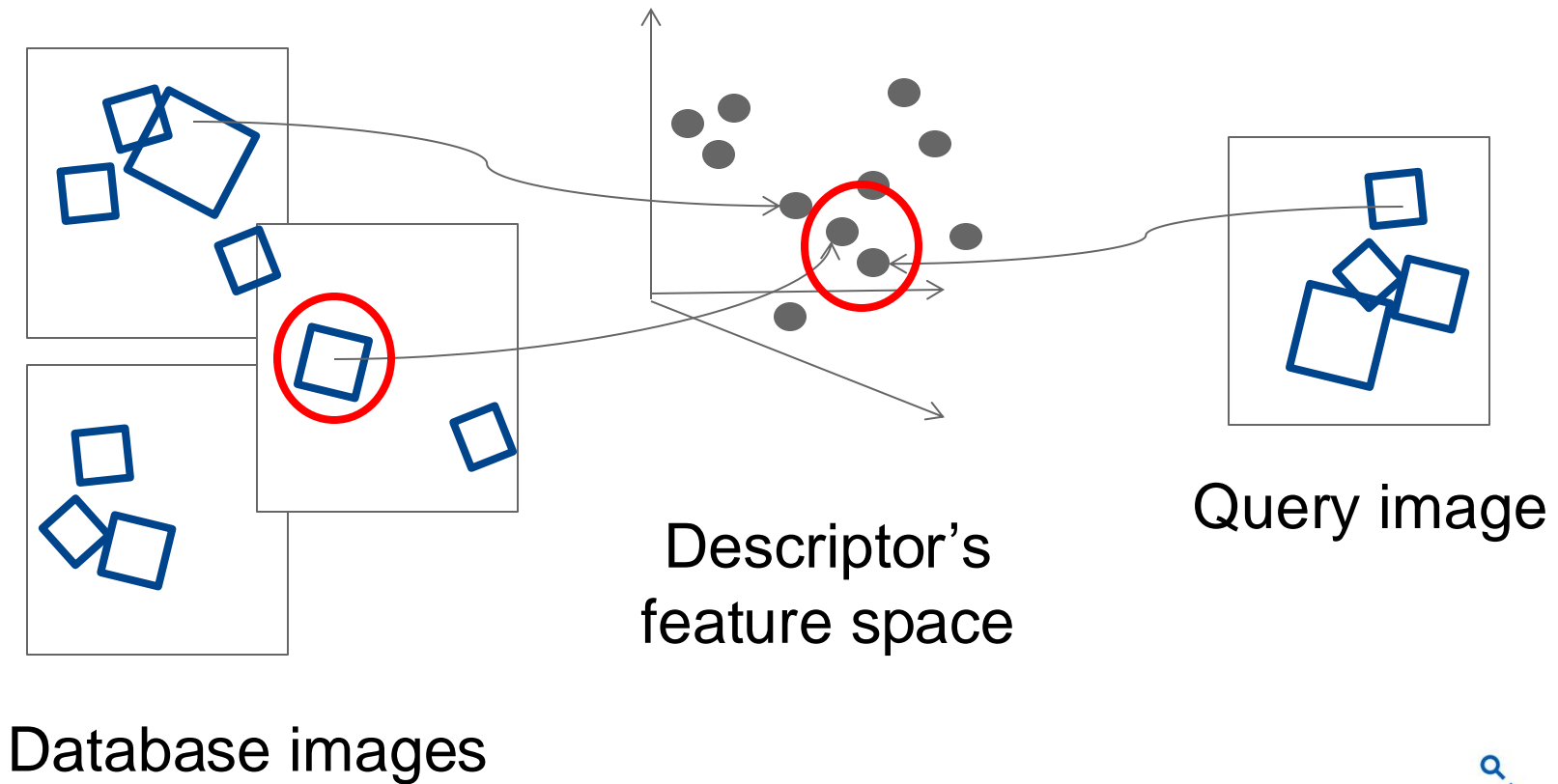
Indexing local features

- Each patch / region has a **descriptor**, which is a **point** in some high-dimensional feature space, e.g., SIFT.



Indexing local features

- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



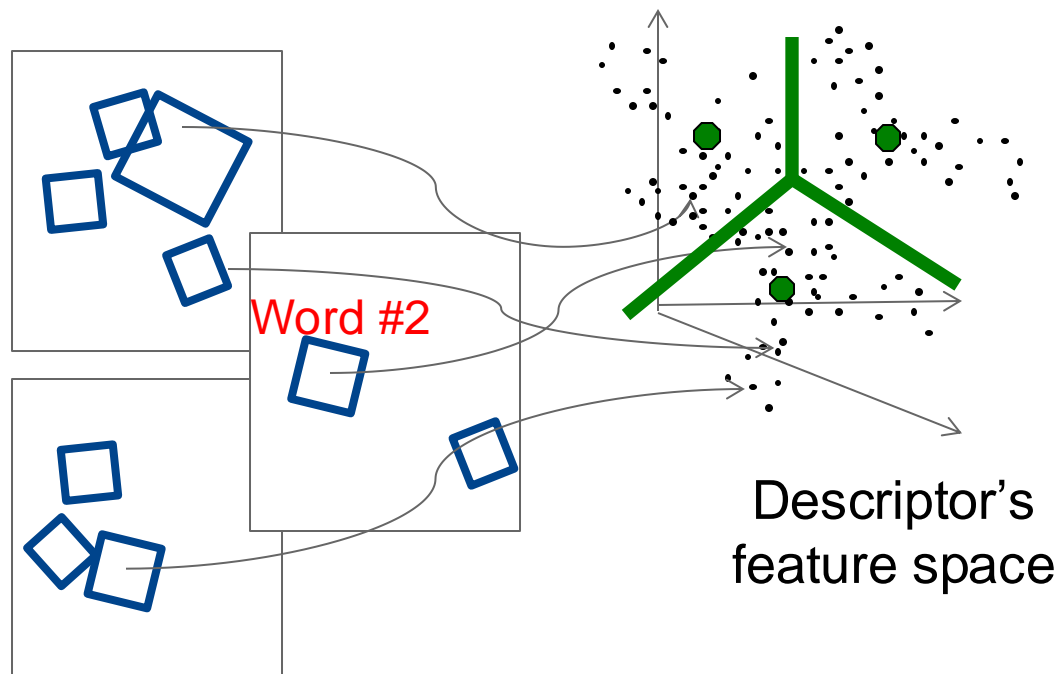
Indexing local features: inverted file index

- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index...
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we map features to “visual words”.

Index		
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134	Driving Lanes; 85
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)	Duval County; 163
1929 Spanish Trail Roadway; 101-102, 104	CCC, The; 111, 113, 115, 135, 142	Eau Gallie; 175
511 Traffic Information; 83	Ca d'Zan; 147	Edison, Thomas; 152
A1A (Barrier Is) - I-95 Access; 86	Caloosahatchee River; 152	Eglin AFB; 116-118
AAA (and CAA); 83	Name; 150	Eight Reale; 176
AAA National Office; 88	Canaveral Natl Seashore; 173	Ellenton; 144-145
Abbreviations,	Cannon Creek Airpark; 130	Emanuel Point Wreck; 120
Colored 25 mile Maps; cover	Canopy Road; 106, 169	Emergency Calibboxes; 83
Exit Services; 196	Cape Canaveral; 174	Epiphytes; 142, 148, 157, 159
Travelogue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	Bridge (I-10); 119
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150	County; 120
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93	Estero; 153
Air Conditioning, First; 112	Charlotte County; 149	Everglades; 90, 95, 139-140, 154-160
Alabama; 124	Charlotte Harbor; 150	Draining of; 156, 181
Alachua; 132	Chautauqua; 116	Wildlife MA; 160
County; 131	Chipley; 114	Wonder Gardens; 154
Alafia River; 143	Name; 115	Falling Waters SP; 115
Alapaha, Name; 126	Choctawhatchee, Name; 115	Fantasy of Flight; 95
Alfred B Macley Gardens; 106	Circus Museum, Ringling; 147	Fayer Dykes SP; 171
Alligator Alley; 154-155	Citrus; 88, 97, 130, 136, 140, 180	Fires, Forest; 166
Alligator Farm, St Augustine; 169	ChiyPlace, W Palm Beach; 180	Fires, Prescribed; 148
Alligator Hole (definition); 157	City Maps,	Fisherman's Village; 151
Alligator, Buddy; 155	Fl Lauderdale Expwy; 194-195	Flagler County; 171
Alligators; 100, 135, 138, 147, 156	Jacksonville; 163	Flagler, Henry; 97, 165, 167, 171
Anastasia Island; 170	Kissimmee Expwy; 192-193	Florida Aquarium; 186
Anhaica; 108-109, 146	Miami Expressways; 194-195	Florida,
Apalachicola River; 112	Orlando Expressways; 192-193	12,000 years ago; 187
Appleton Mus of Art; 136	Pensacola; 26	Cavern SP; 114
Aquifer; 102	Tallahassee; 191	Map of all Expressways; 2-3
Arabian Nights; 94	Tampa-St. Petersburg; 63	Mus of Natural History; 134
Art Museum, Ringling; 147	St. Augustine; 191	National Cemetery; 141
Aruba Beach Cafe; 183	Civil War; 100, 108, 127, 138, 141	Part of Africa; 177
Aucilla River Project; 106	Cleaver Marine Aquarium; 187	Platform; 187
Babcock-Web WMA; 151	Collier County; 154	Sheriff's Boys Camp; 126
Bahia Mar Marina; 184	Collier, Barron; 152	Sports Hall of Fame; 130
Baker County; 99	Colonial Spanish Quarters; 168	Sun 'n Fun Museum; 97
Barefoot Mailmen; 182	Columbia County; 101, 128	Supreme Court; 107
Barge Canal; 137	Coquina Building Material; 165	Florida's Turnpike (FTP); 178, 189
Bee Line Expy; 80	Corkscrew Swamp, Name; 154	25 mile Strip Maps; 66
Beiz Outlet Mall; 89	Cowboys; 95	Administration; 189
Bernard Castro; 136	Crab Trap II; 144	Coin System; 190
Big "I"; 165	Cracker, Florida; 88, 95, 132	Exit Services; 189
Big Cypress; 155, 158	Croston Expy; 11, 35, 98, 143	HEFT; 76, 161, 190
Big Foot Monster; 105	Cuban Bread; 184	History; 189
Billie Swamp Safari; 160	Dade Battlefield; 140	Names; 189
Blackwater River SP; 117	Dade, Maj. Francis; 139-140, 161	Service Plazas; 190
Blue Angels	Dania Beach Hurricane; 184	Spur SR91; 76
A4-C Skyhawk; 117	Daniel Boone, Florida Walk; 117	Ticket System; 190
Atrium; 121	Daytona Beach; 172-173	Toll Plazas; 190
Blue Springs SP; 87	De Land; 87	Ford, Henry; 152
Blue Star Memorial Highway; 125	De Soto, Hernando,	Fort Barrancas; 122
Boca Ciega; 169	Anhaica; 108-109, 146	Buried Alive; 123
Boca Grande; 150	County; 149	Fort Caroline; 164
Boca Raton; 182	Explorer; 146	Fort Clinch SP; 161
Bonnie Blue Flag; 124	Landing; 146	Fort De Soto & Egmont Key; 188
Boyd Hill Nature Trail; 188	Napiliac; 103	Fort Lauderdale; 161, 182-184
Bradenton; 145-147	National Park; 147	Fort Myers; 152-153
Breakers, The, Palm Beach; 181	Tallahassee; 108	Fort Pierce; 177-178
Brickell Point, Miami; 185	DeFuniak Springs; 116	Farmers Market; 178
Britton Hill; 116	Name; 115	Fountain of Youth; 170
Brogan Museum; 107	DeInor-Wiggins Pass SP; 155	Frank Lloyd Wright Center; 97
Bromeliads (see Epiphytes)	Denoll Cafe, St Augustine; 169	Gadsden County; 110
Broward County; 159, 181	Devil's Millhopper; 132	Gainesville; 99, 104, 131-135, 146
Broward, Gov. Napoleon; 156	Dickson Azalea Park; 69	Gamble Plantation; 145
Bulow Plantation Ruins; 171	Dinosaur World; 98	Garden of Eden; 112
Bush, Gov. Jeb; 100	Discovery Cove; 90	Gasparilla, Pirate; 150
	Doie Highway; 186	Gatorade; 134
	Don Garlits Drag Racing Mus; 138	Gaylord Palms; 90
	Douglas, Marjory Stoneham; 159	Geology; 102-103, 110, 131-132

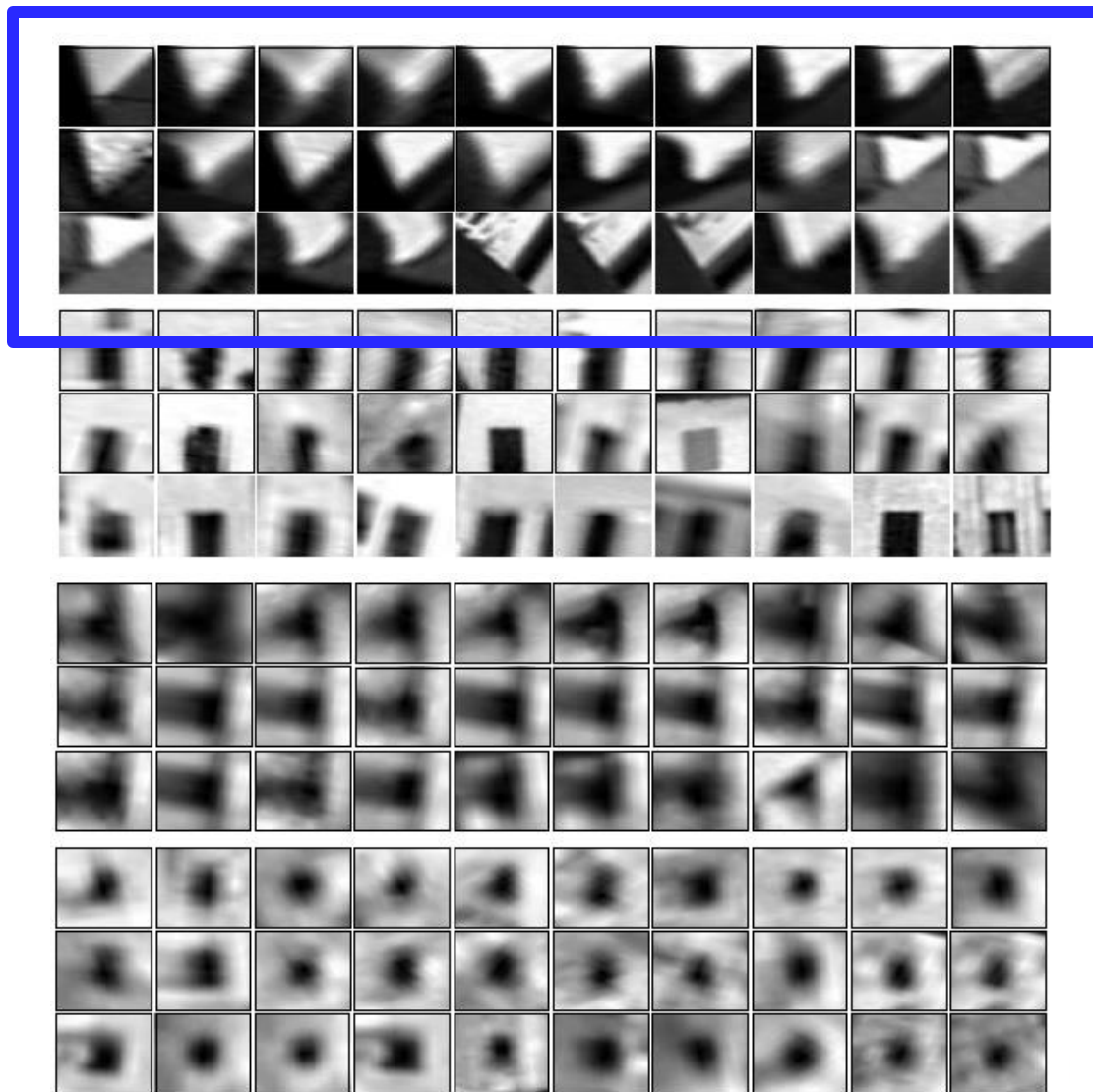
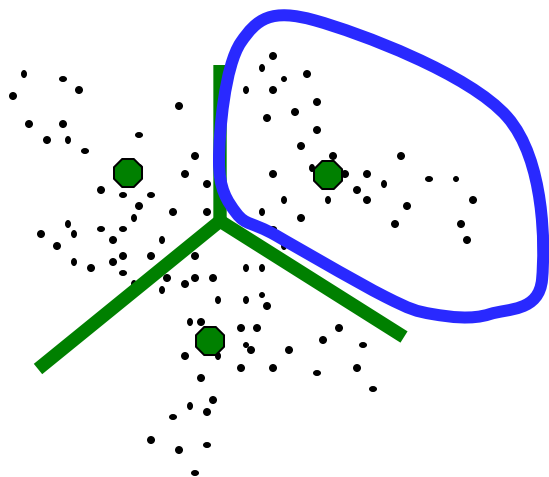
Visual Words

- Map descriptors to “words” by quantizing feature space
 - Quantize via clustering
 - Cluster centers are the prototype “words”
- Determine which word to assign to each new image region by finding the closest cluster center.



Visual words

- Example: each group of patches belongs to the same visual word

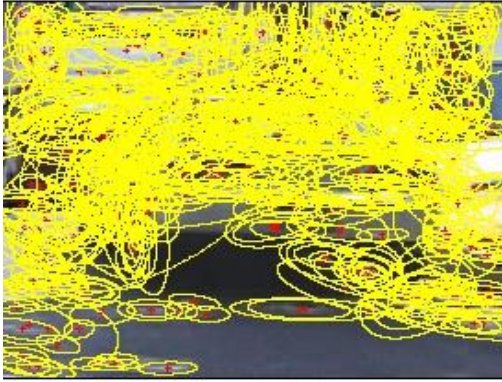


Visual vocabulary formation

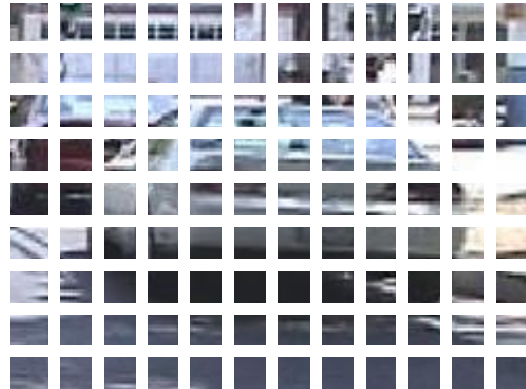
Things need to consider:

- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- Features, vocabulary size, number of words?

Sampling strategies



Sparse, at interest points



Dense, uniformly



Randomly



Multiple interest operators

- To find **specific**, textured **objects**, **sparse sampling** from interest points often more reliable.
- **Multiple** complementary interest **operators** offer more image coverage.
- For object **categorization**, **dense sampling** offers better coverage.

Inverted file index

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

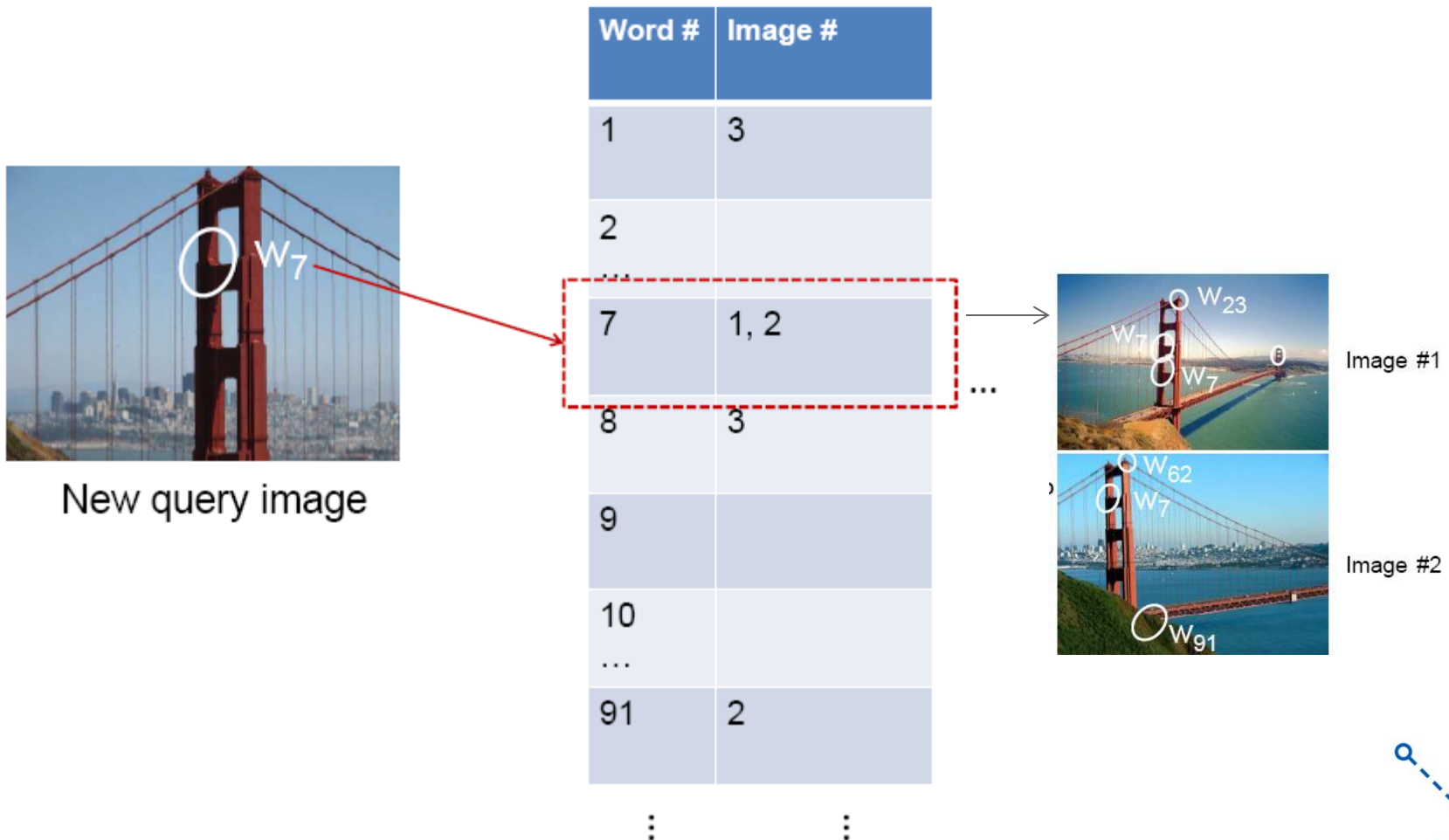
Database images

	Image #1
	Image #2
	Image #3
⋮	⋮

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

Inverted file index

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



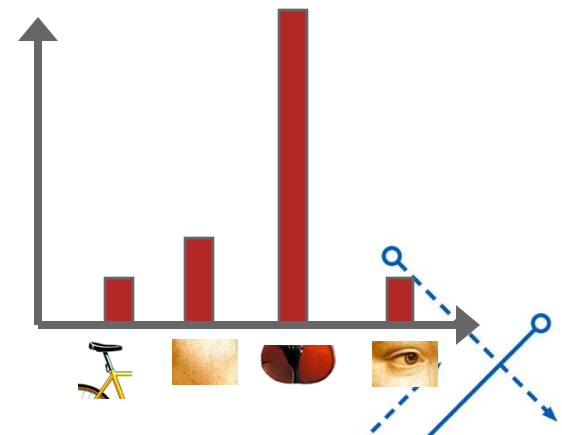
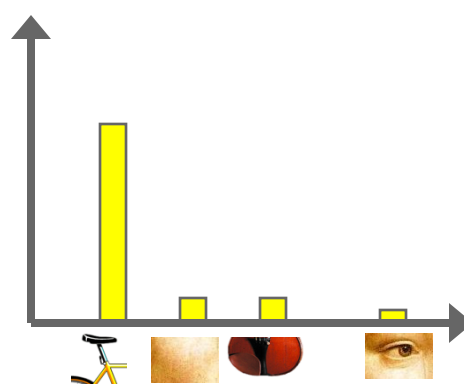
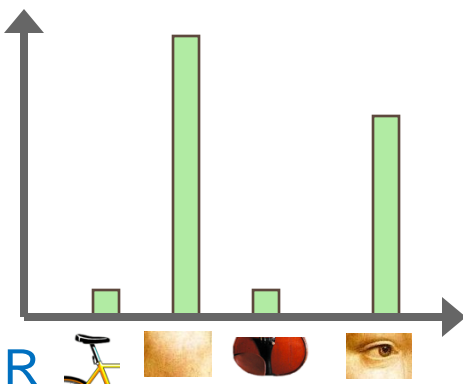
Inverted file index

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

Instance recognition: remaining issues

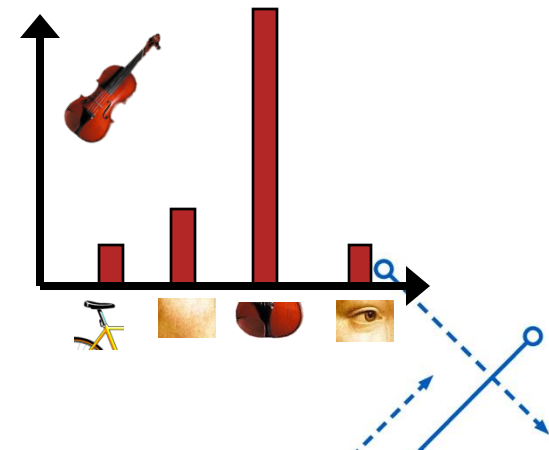
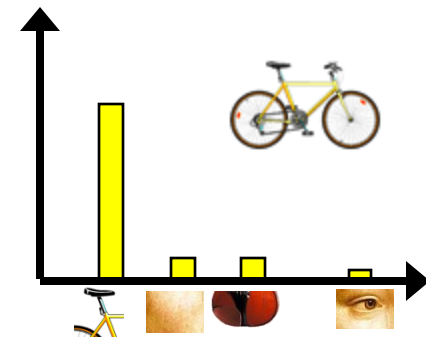
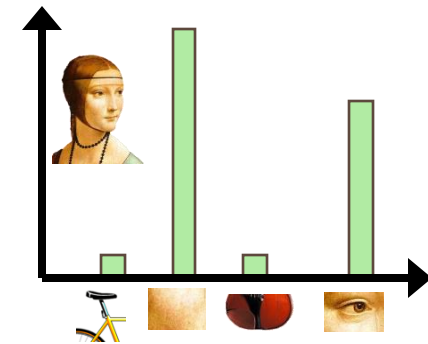
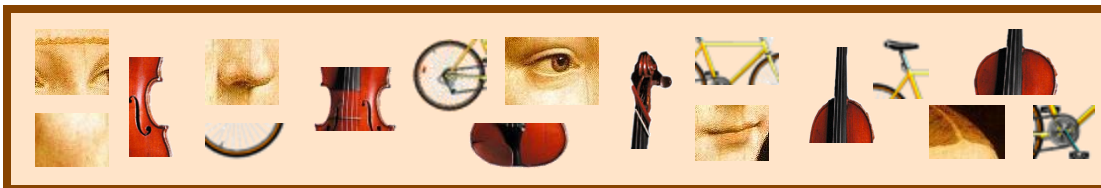
- How to summarize the content of an entire image?
And estimate overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval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

Bags of visual words (Recap)



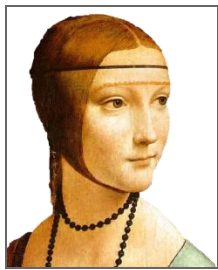
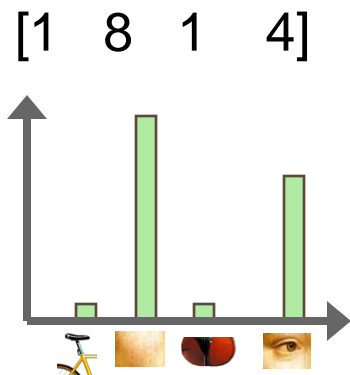
Bags of visual words (Recap)

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Like bag of words representation commonly used for documents.

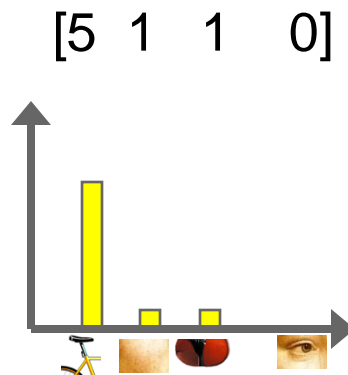


Comparing bags of words (Recap)

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



\vec{d}_j



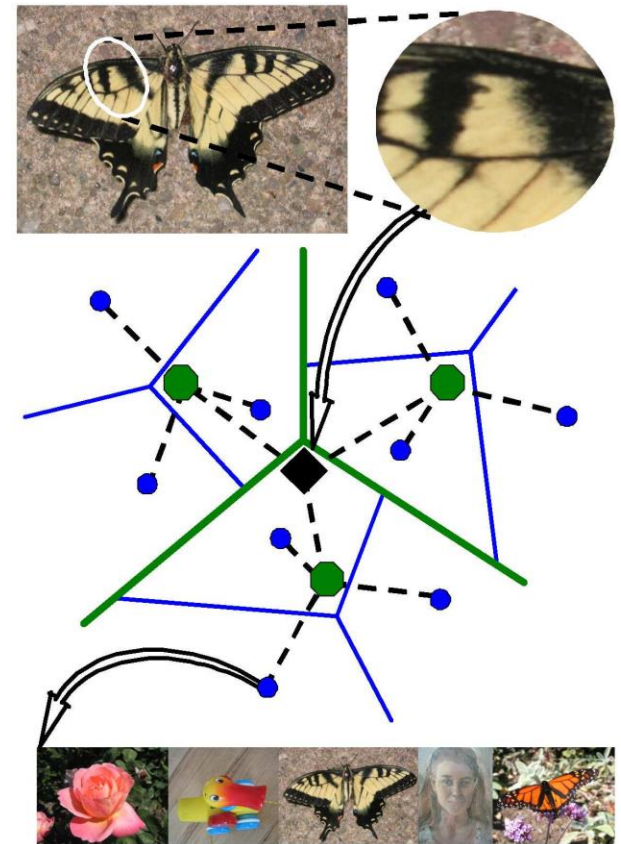
\vec{q}

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

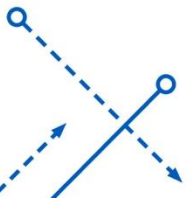
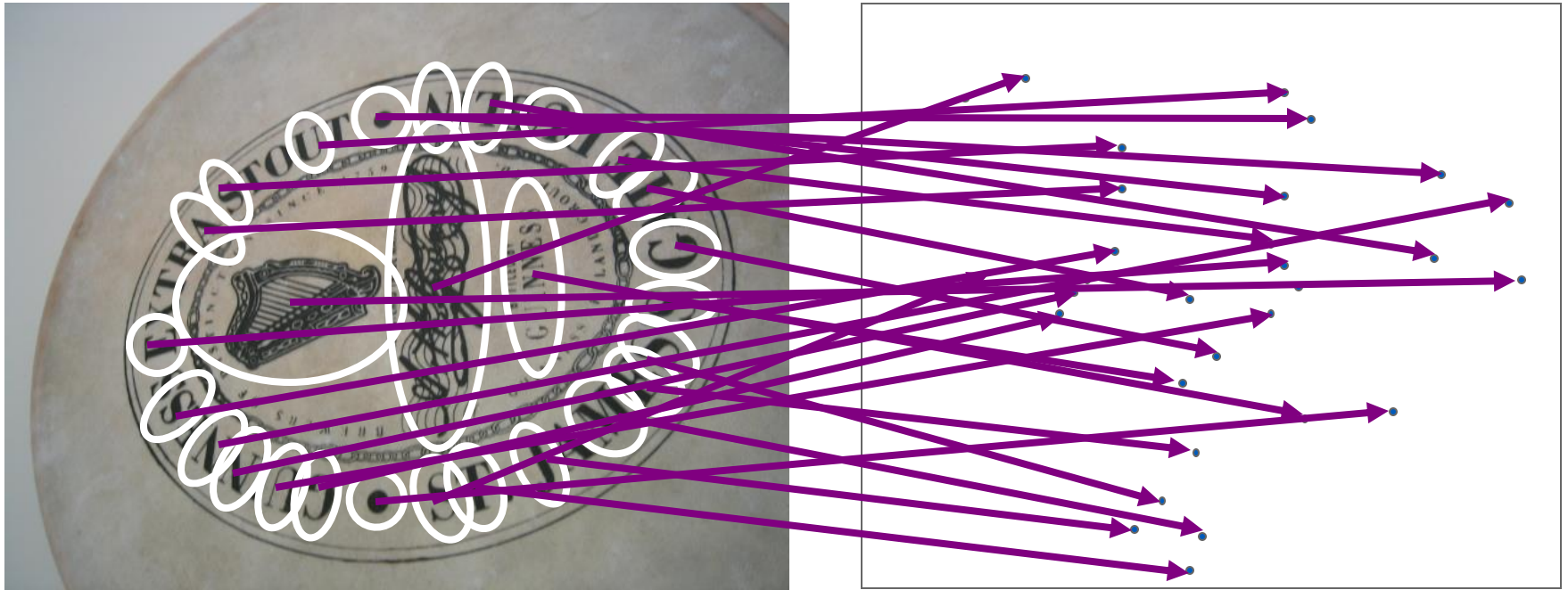
for vocabulary of V words

Visual vocabularies: Issues

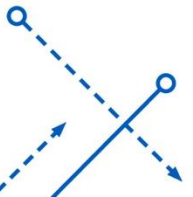
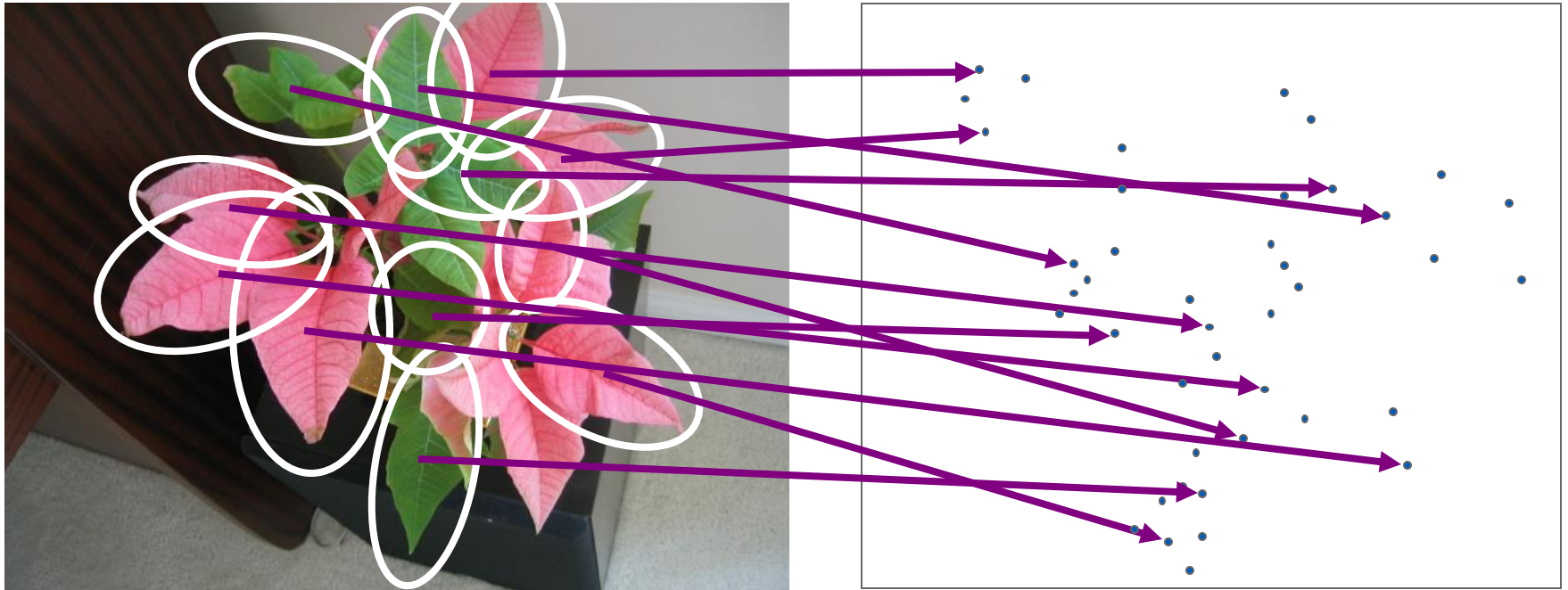
- How to choose vocabulary size?
 - Too small: visual words not representative of all patches
 - Too large: quantization artifacts, overfitting
- Computational efficiency
 - **Vocabulary trees**



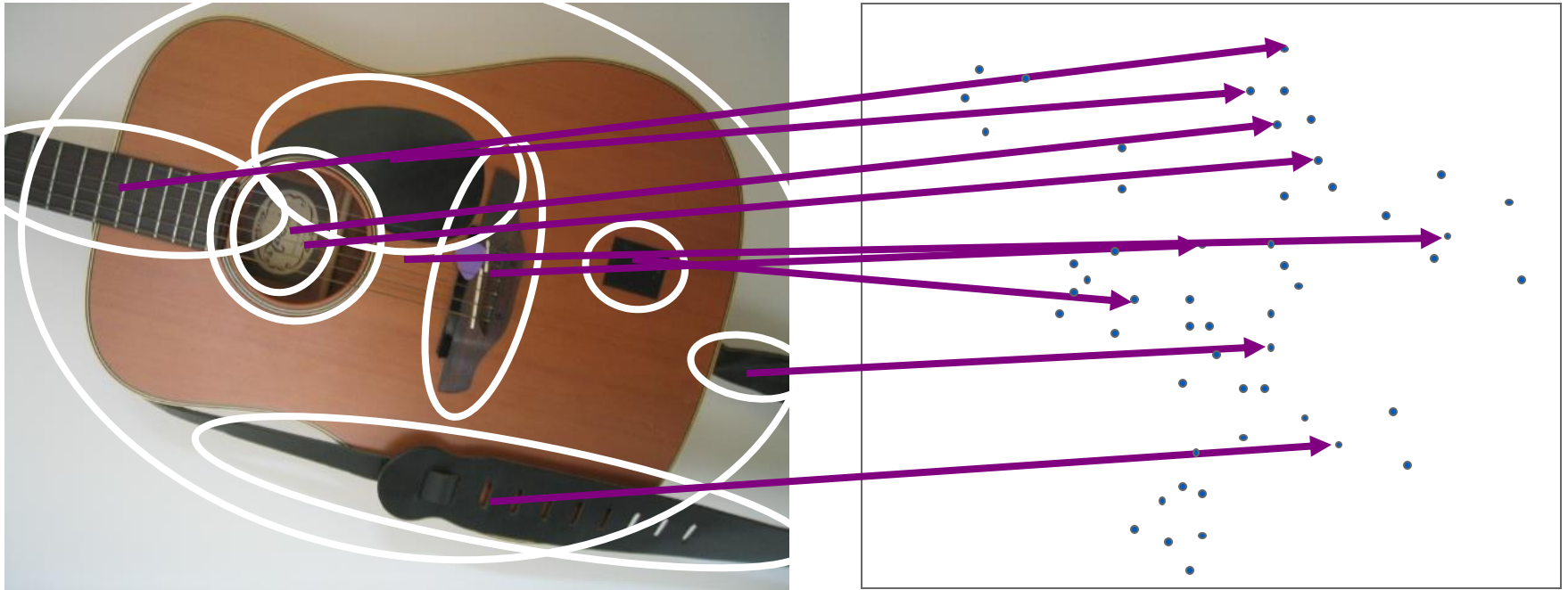
Recognition with K-d-tree



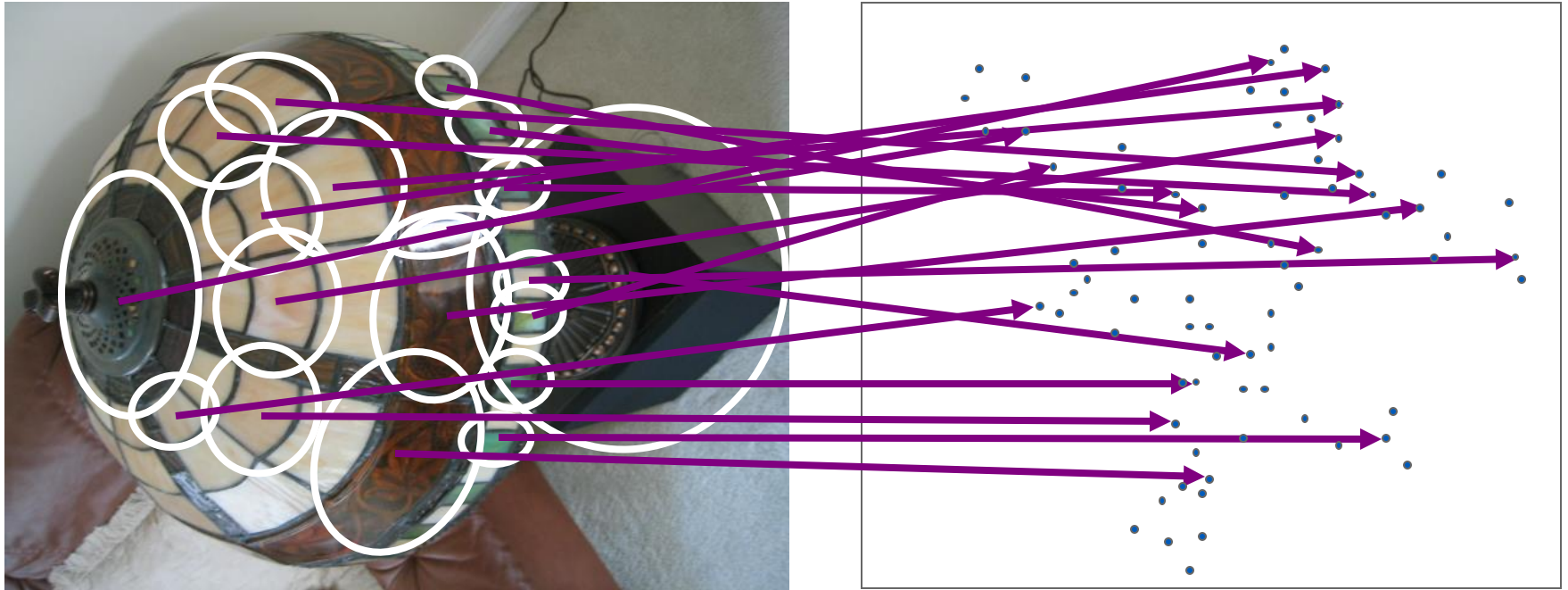
Recognition with K-d-tree



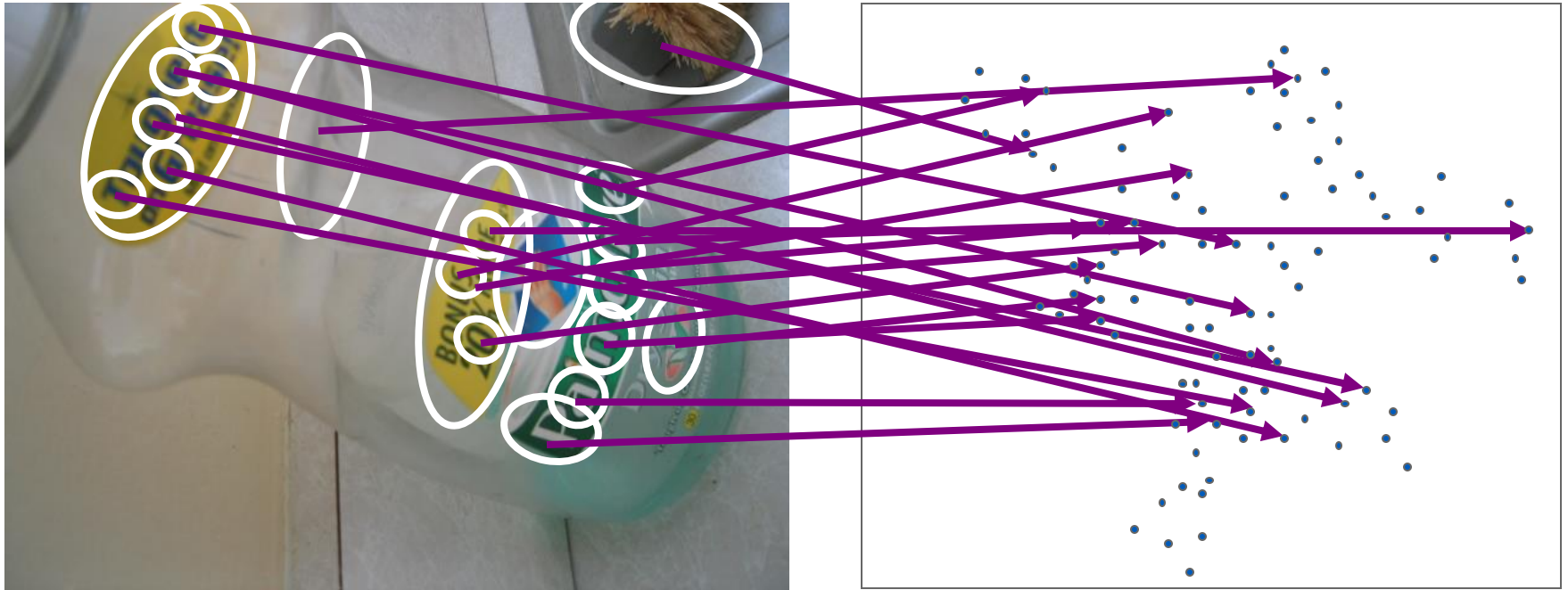
Recognition with K-d-tree



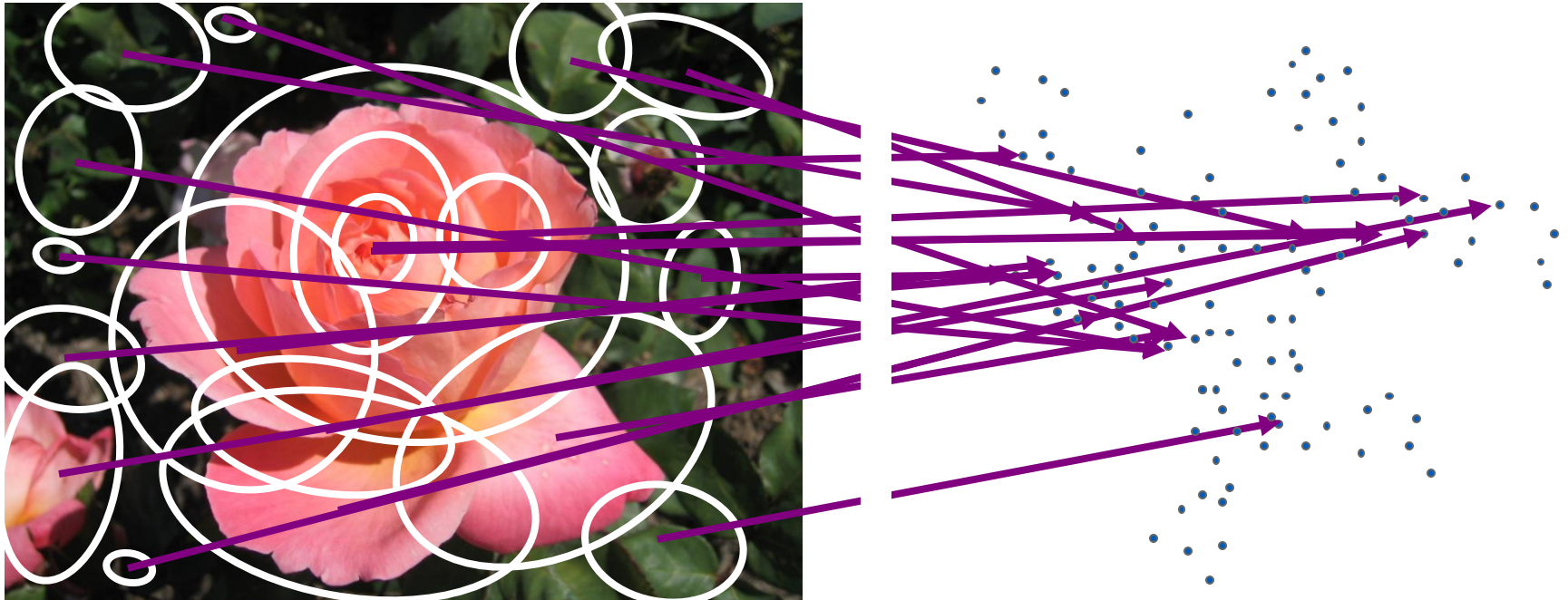
Recognition with K-d-tree



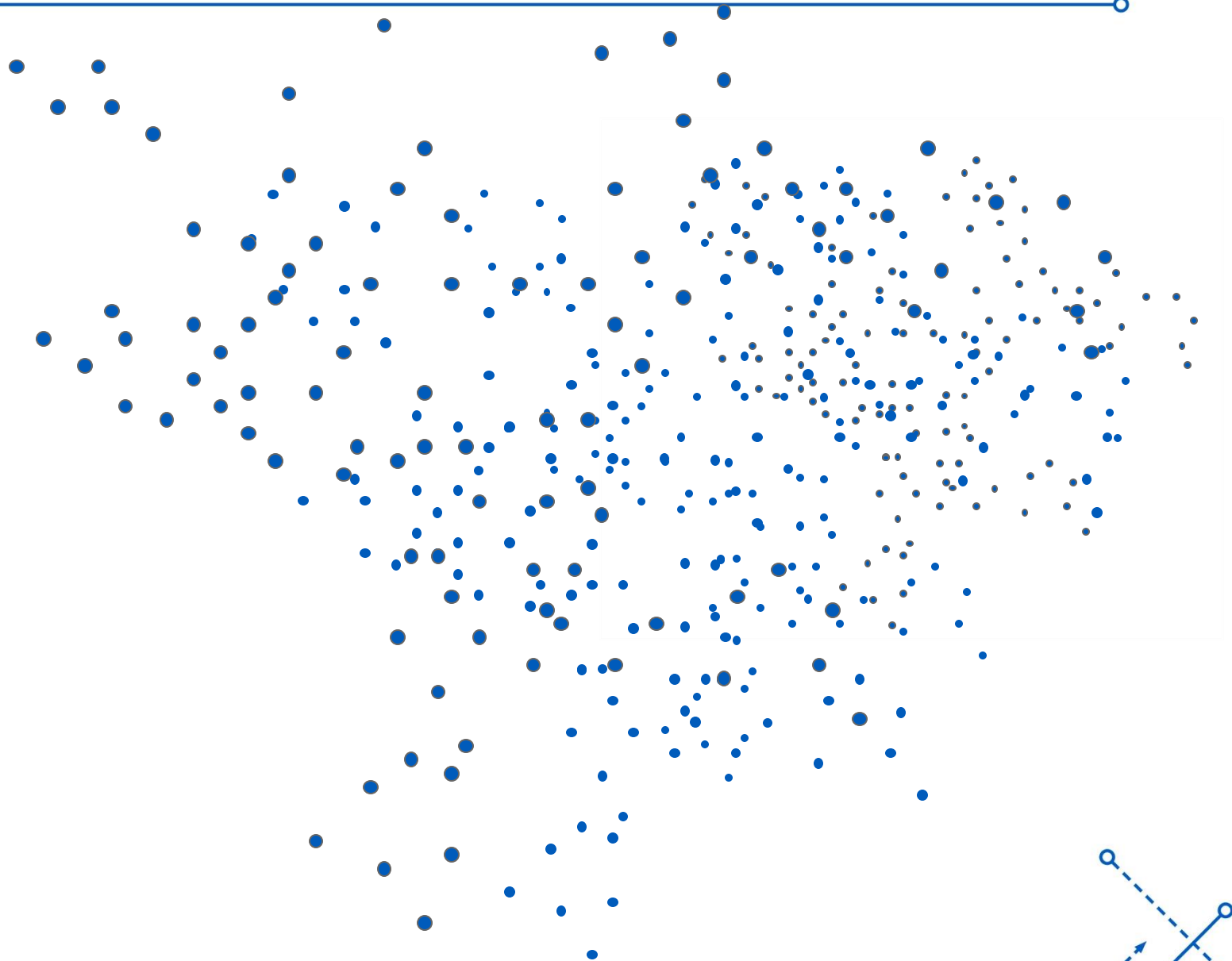
Recognition with K-d-tree



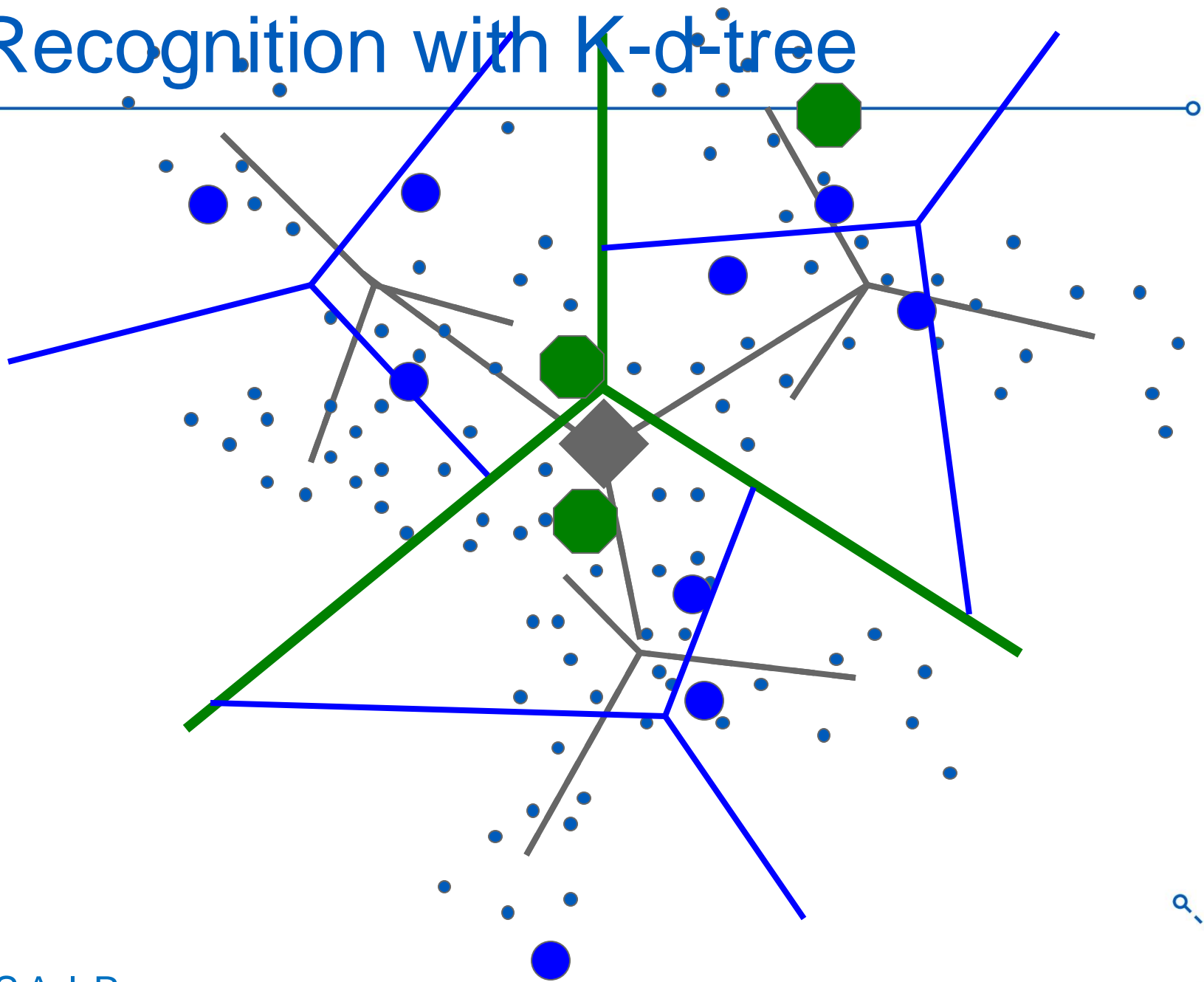
Recognition with K-d-tree



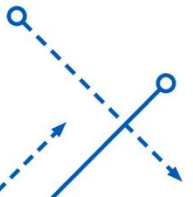
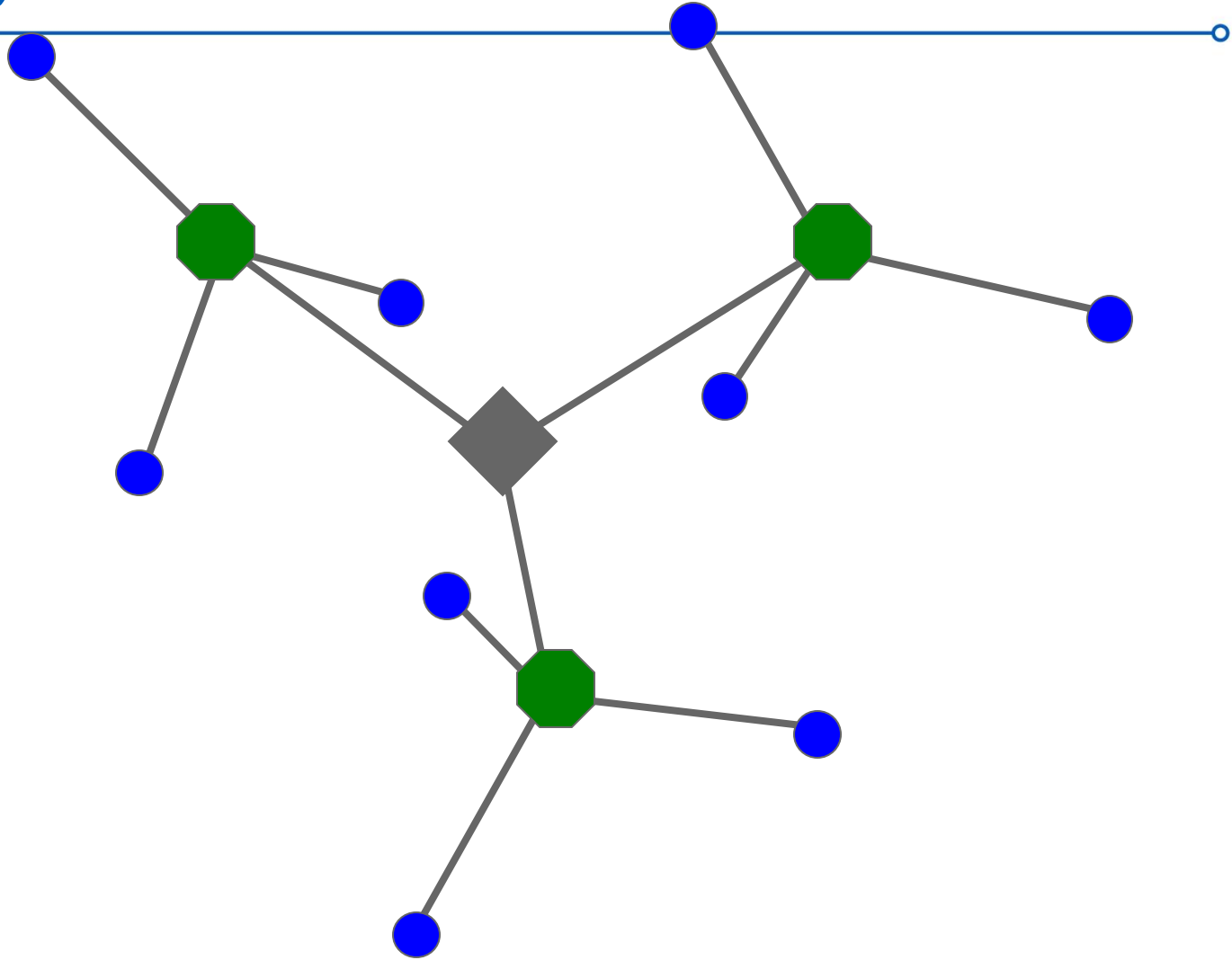
Recognition with K-d-tree



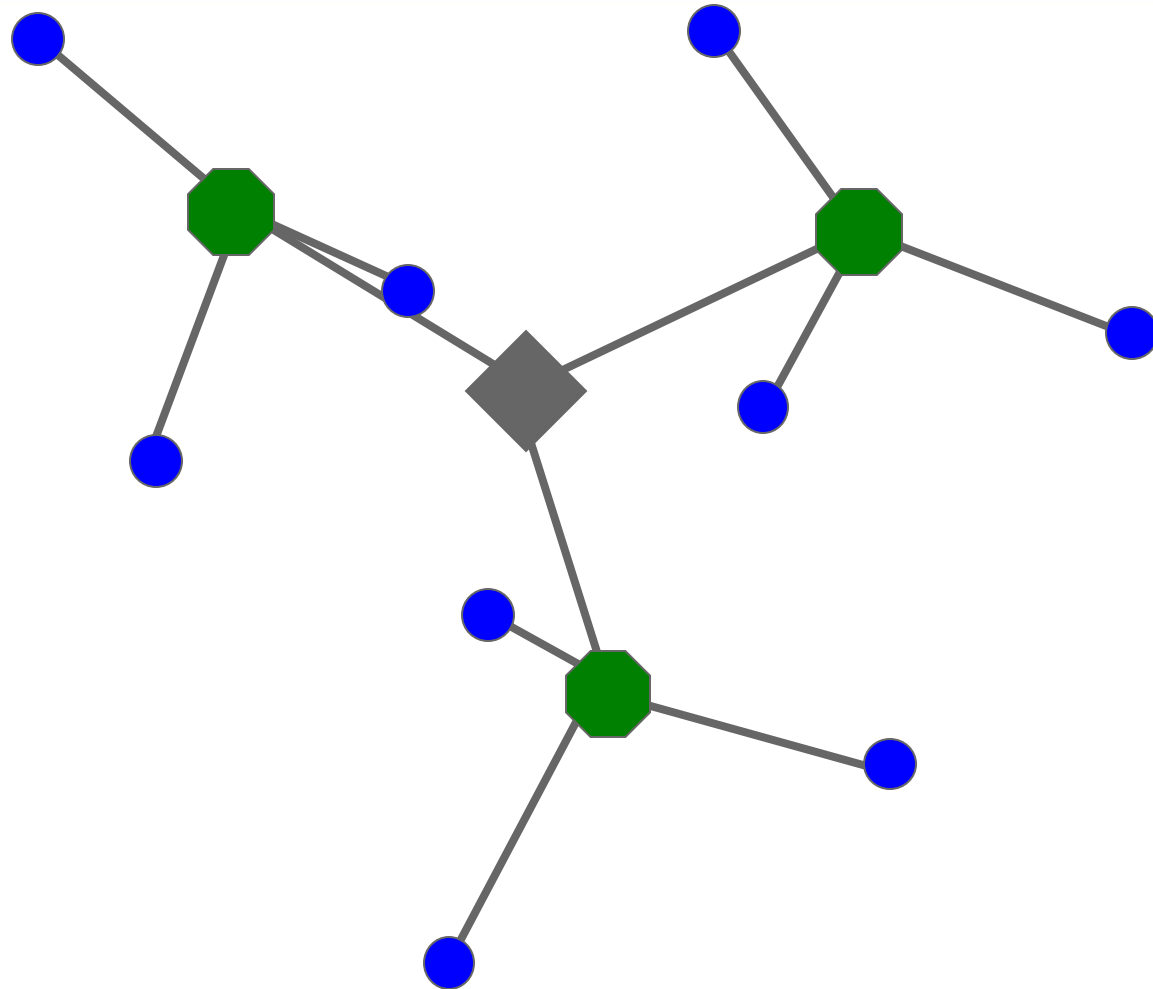
Recognition with K-d-tree



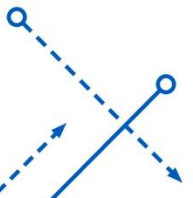
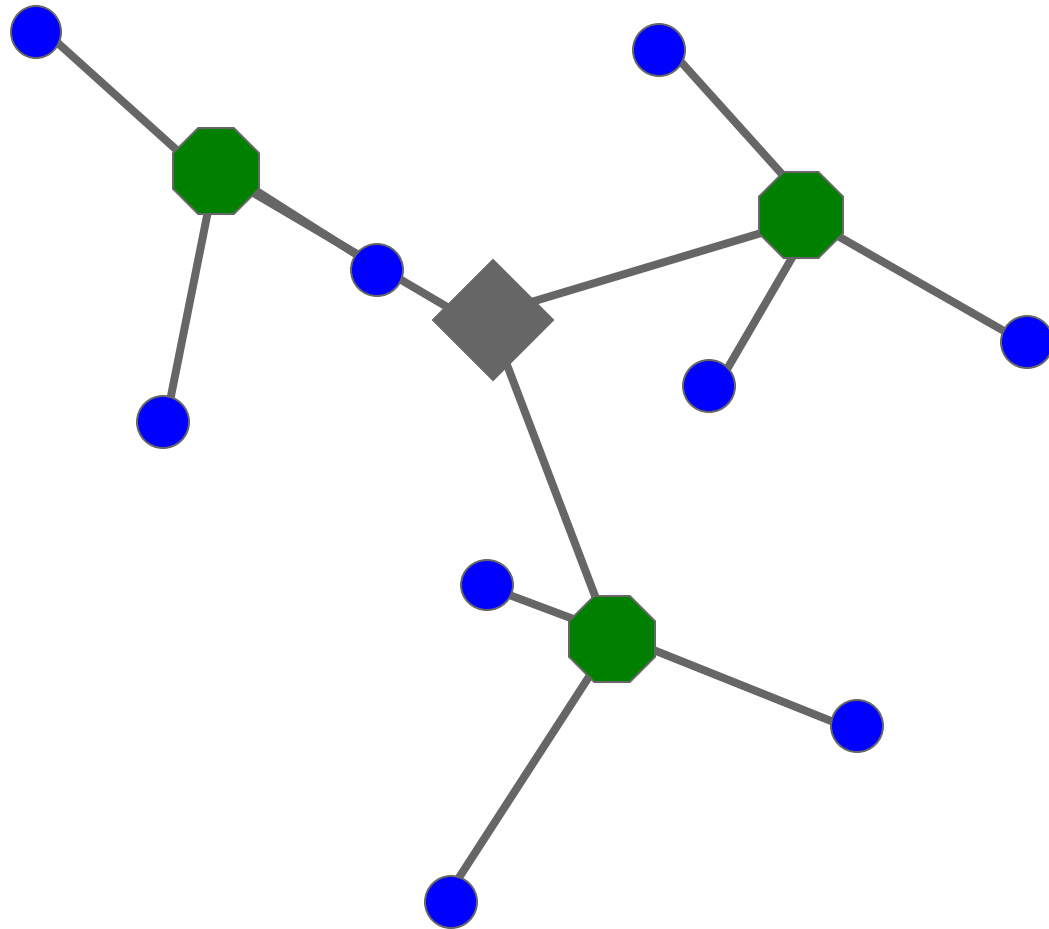
Recognition with K-d-tree



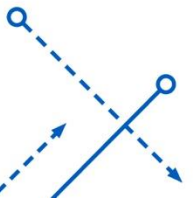
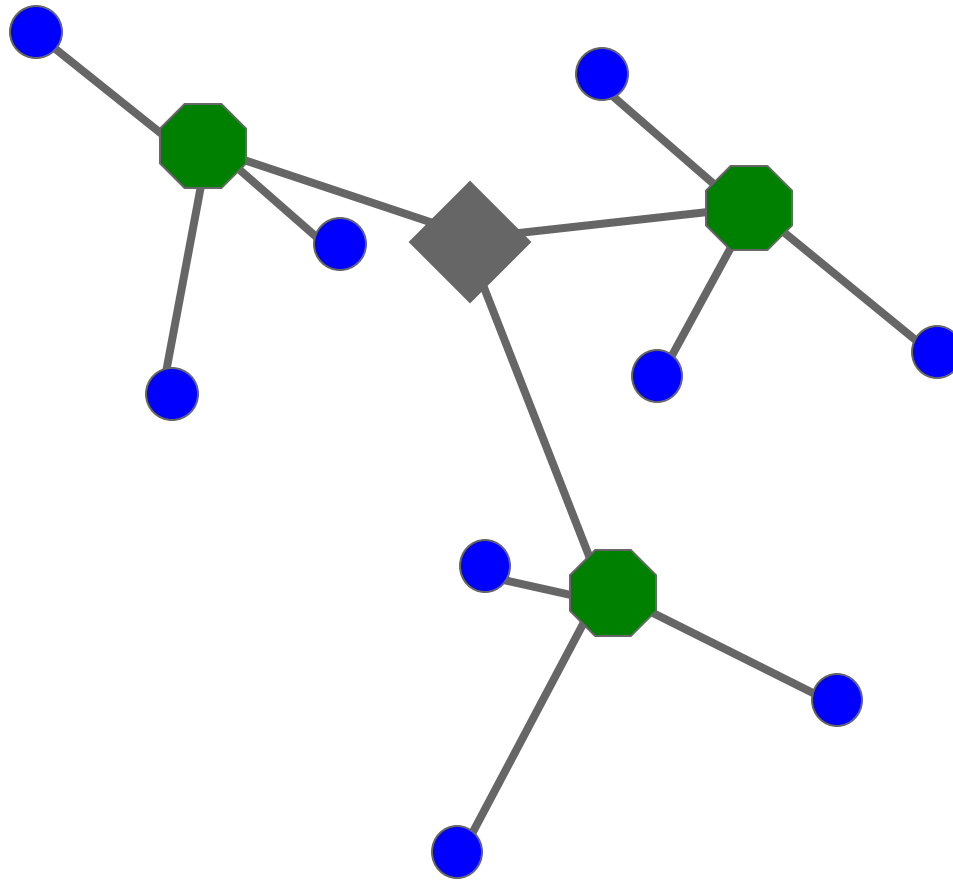
Recognition with K-d-tree



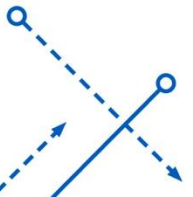
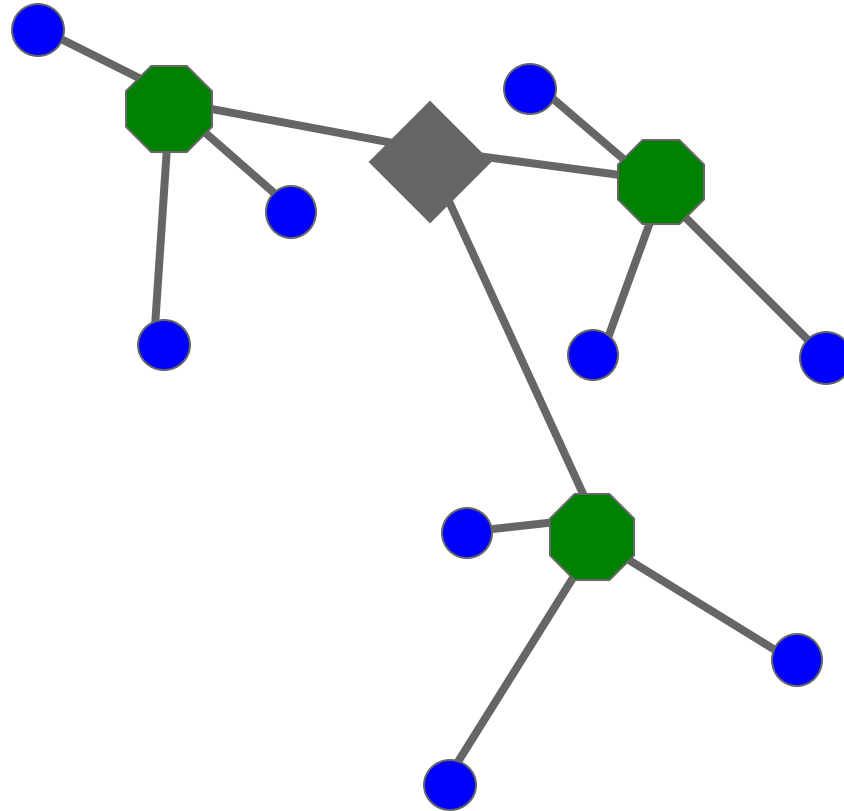
Recognition with K-d-tree



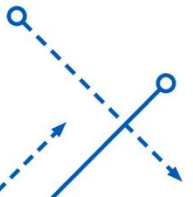
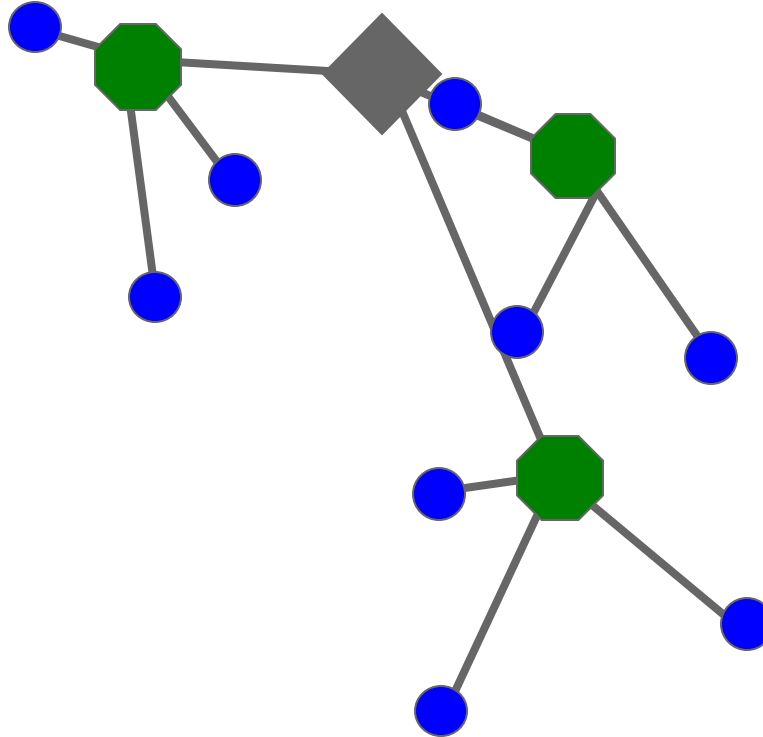
Recognition with K-d-tree



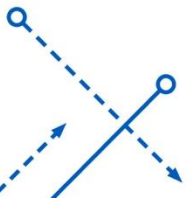
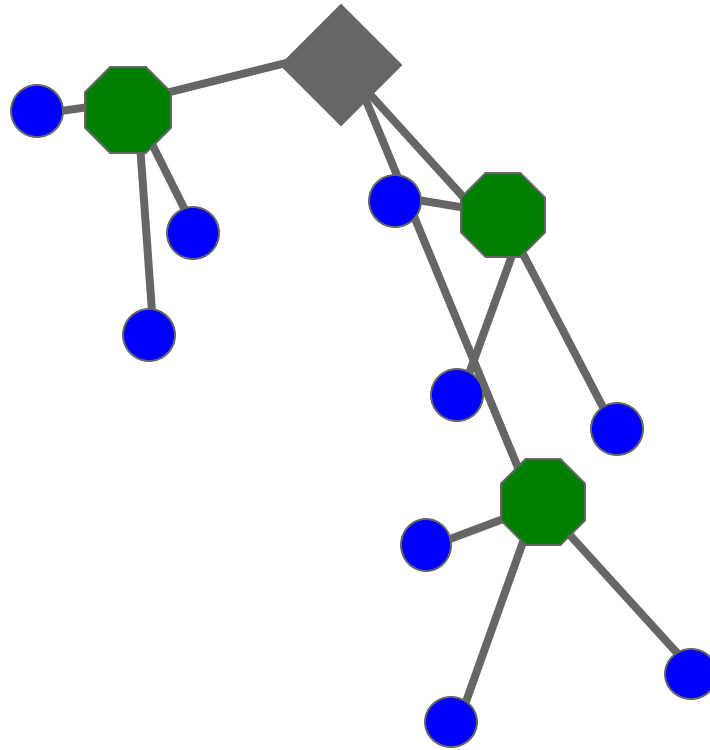
Recognition with K-d-tree



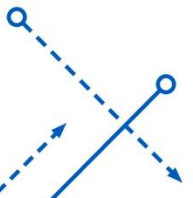
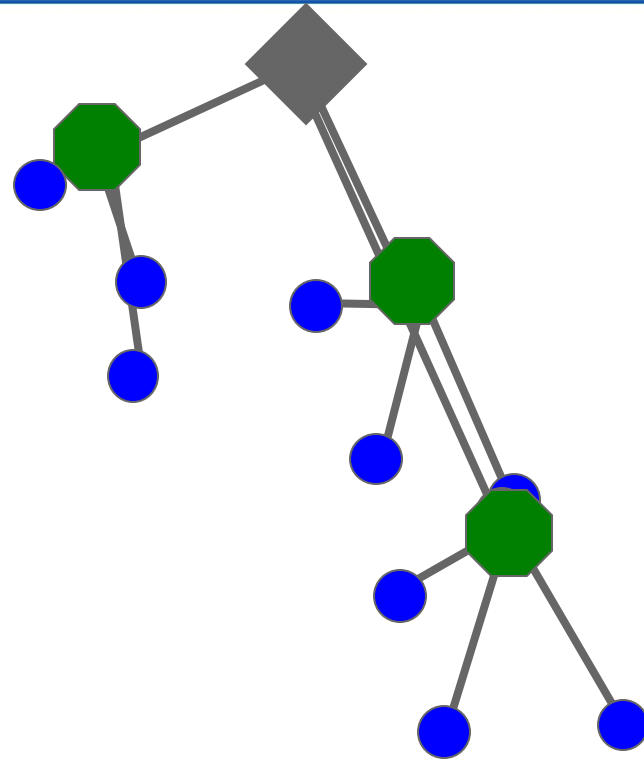
Recognition with K-d-tree



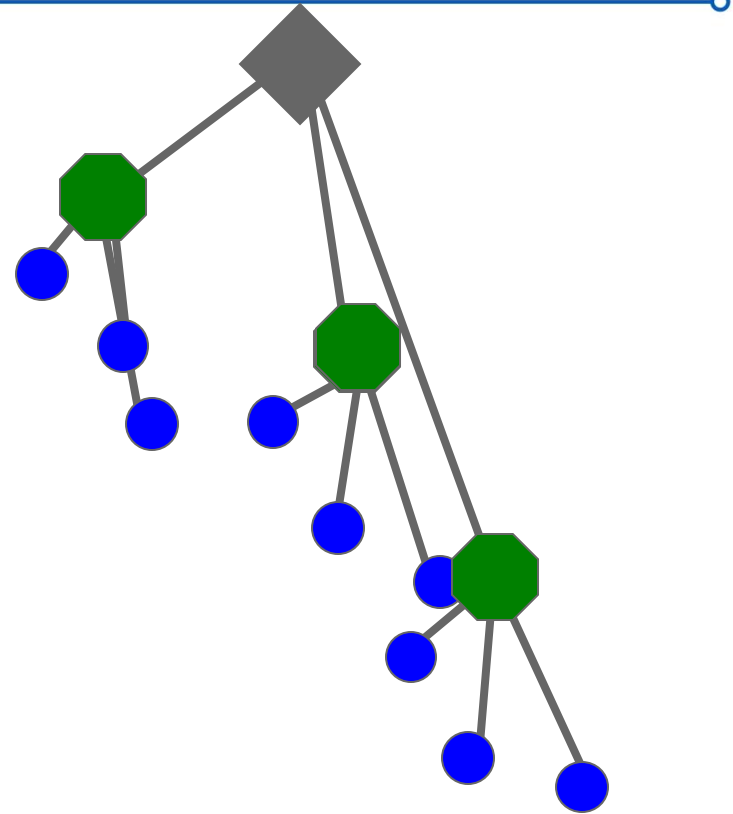
Recognition with K-d-tree



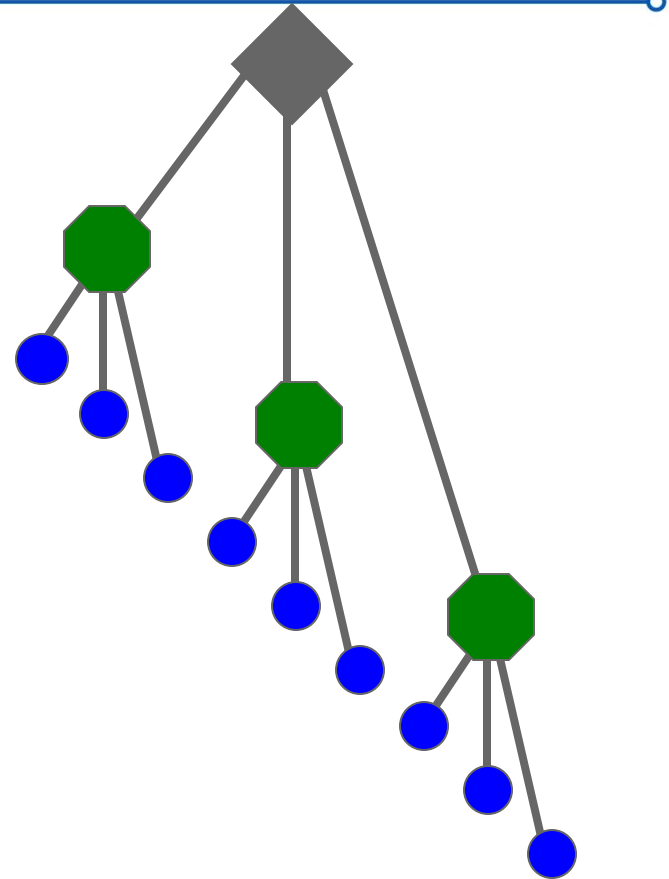
Recognition with K-d-tree



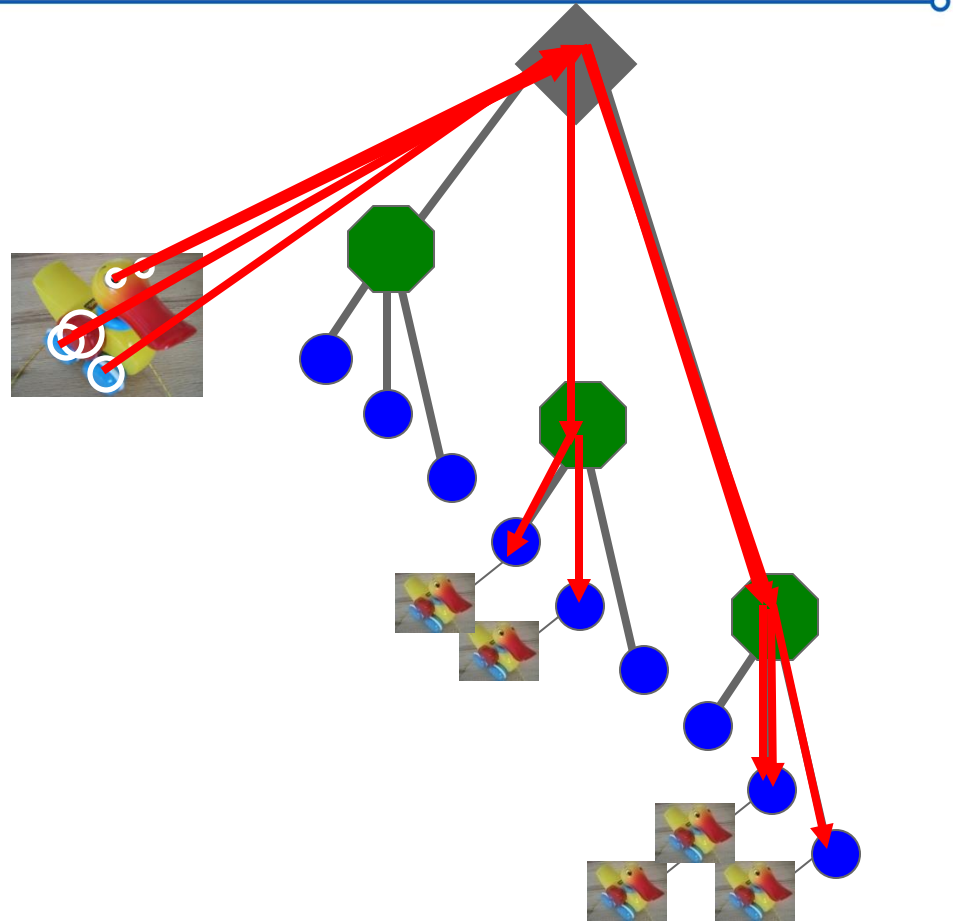
Recognition with K-d-tree



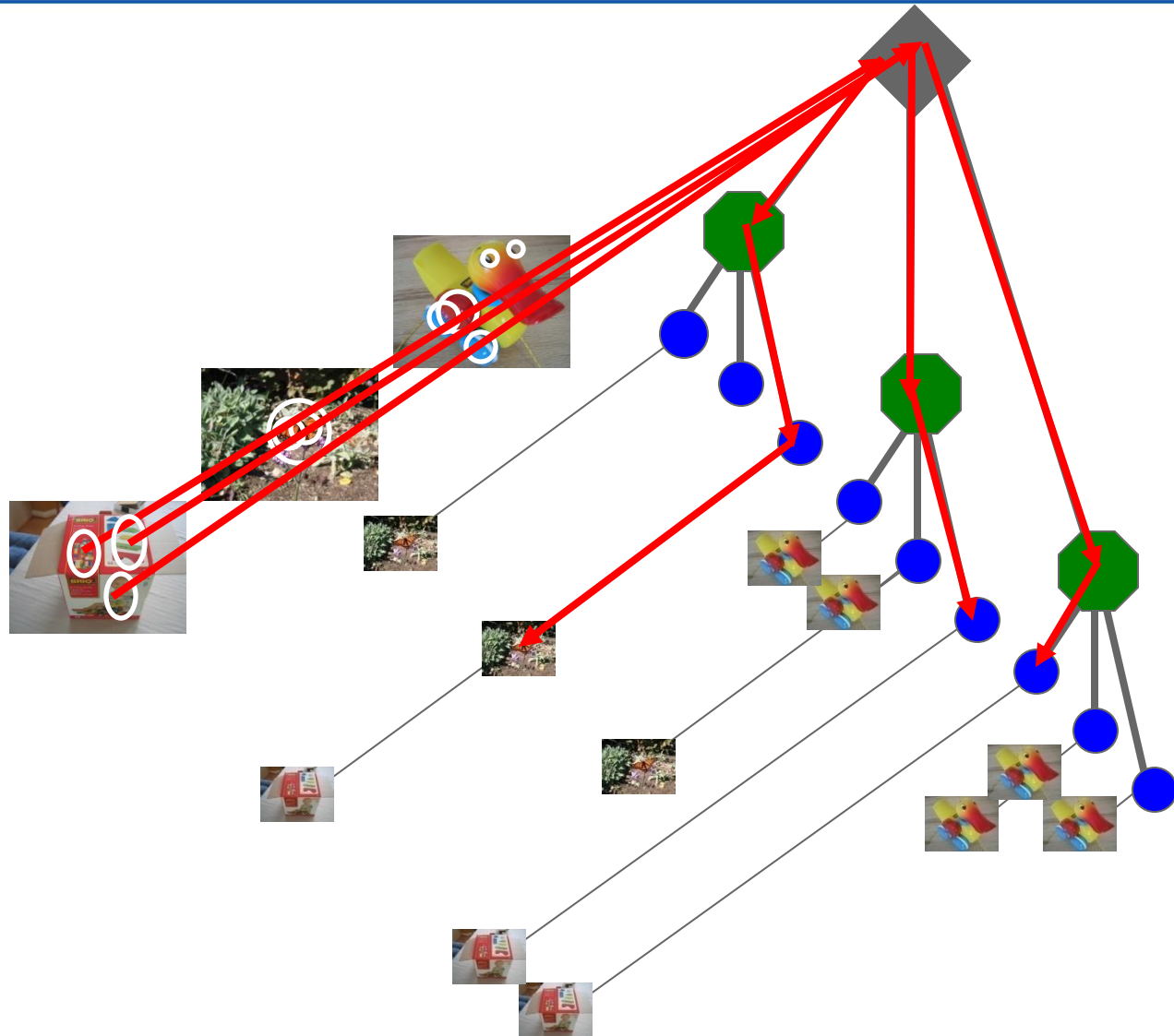
Recognition with K-d-tree



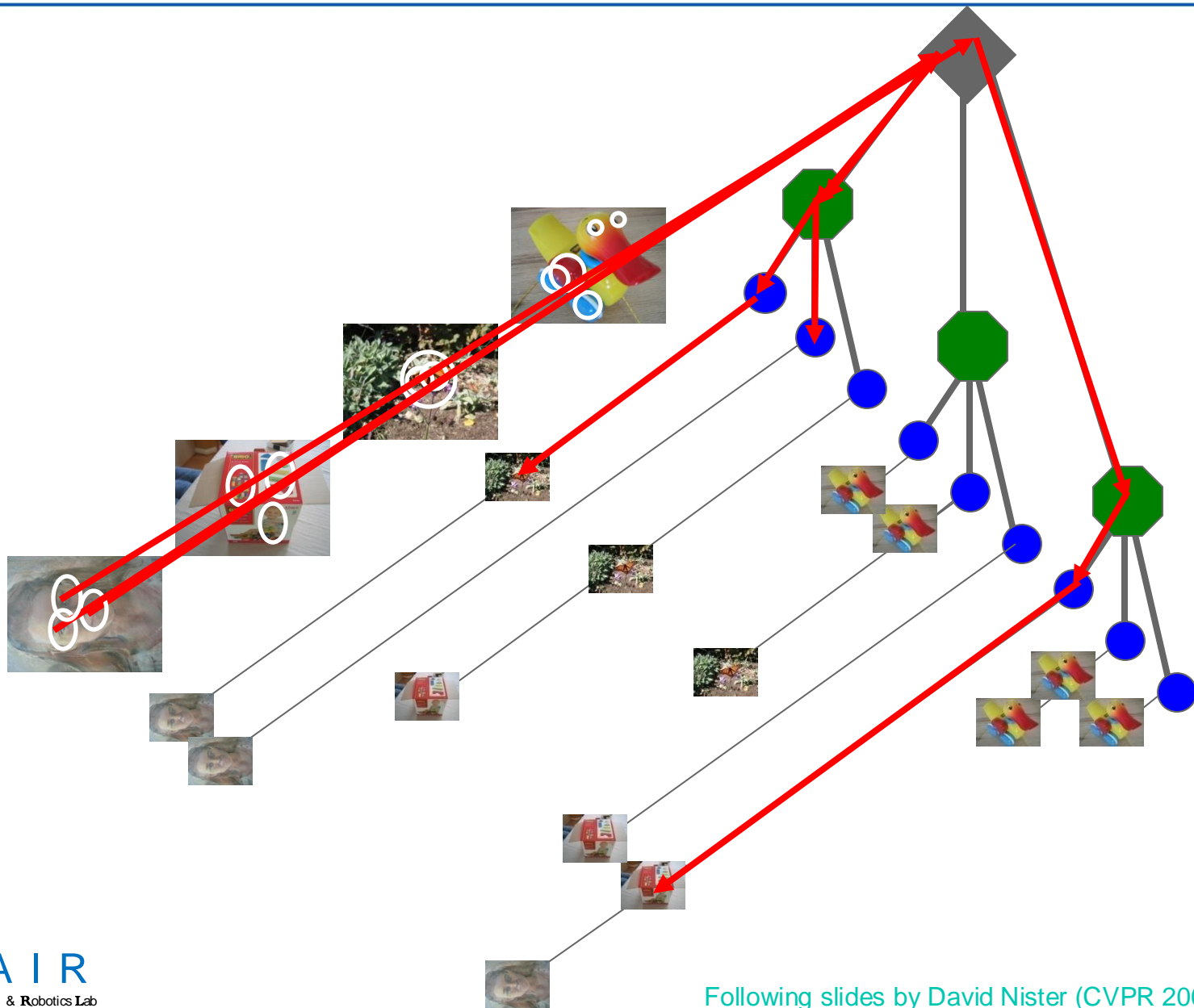
Recognition with K-d-tree



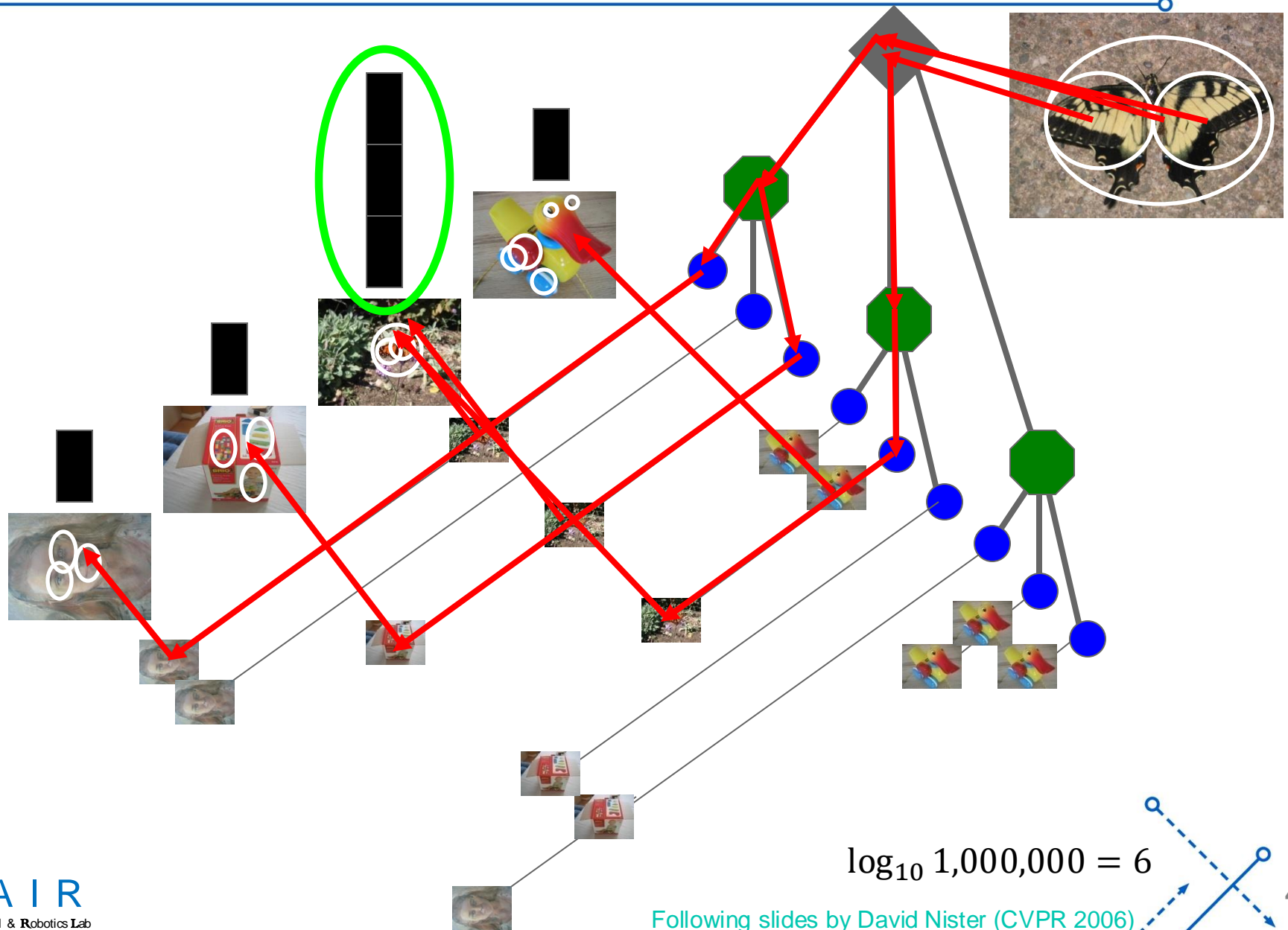
Recognition with K-d-tree



Recognition with K-d-tree



Recognition with K-d-tree



Vocabulary Tree: Performance

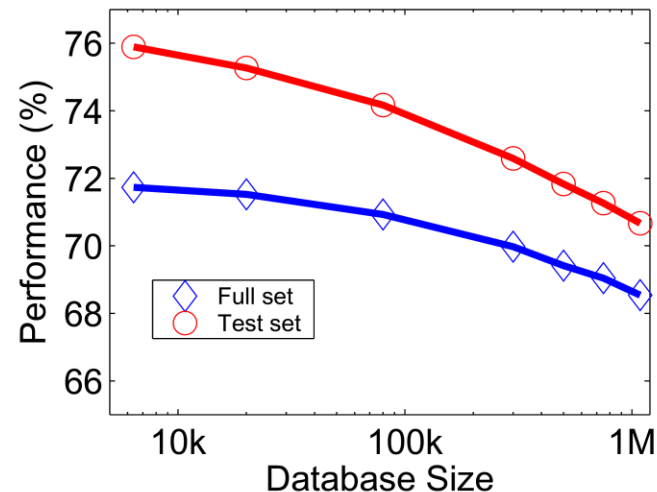
Evaluated on large databases

- Indexing with up to 1M images

Online recognition for database
of 50,000 CD covers

- Retrieval in ~1s

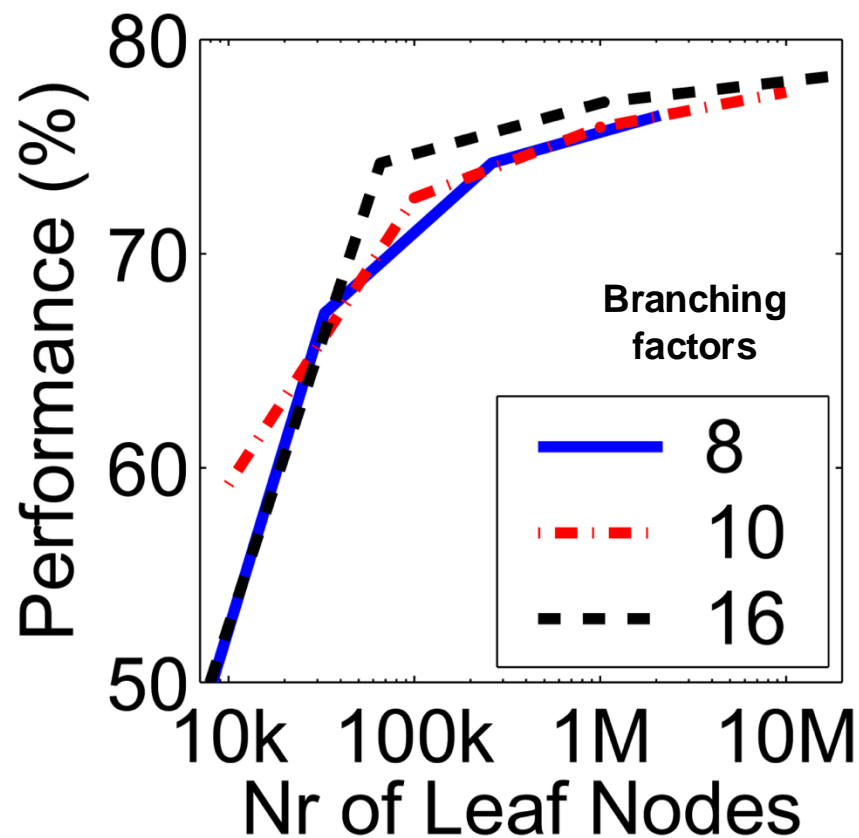
Find experimentally that large
vocabularies can be beneficial for
recognition



Instance recognition: remaining issues

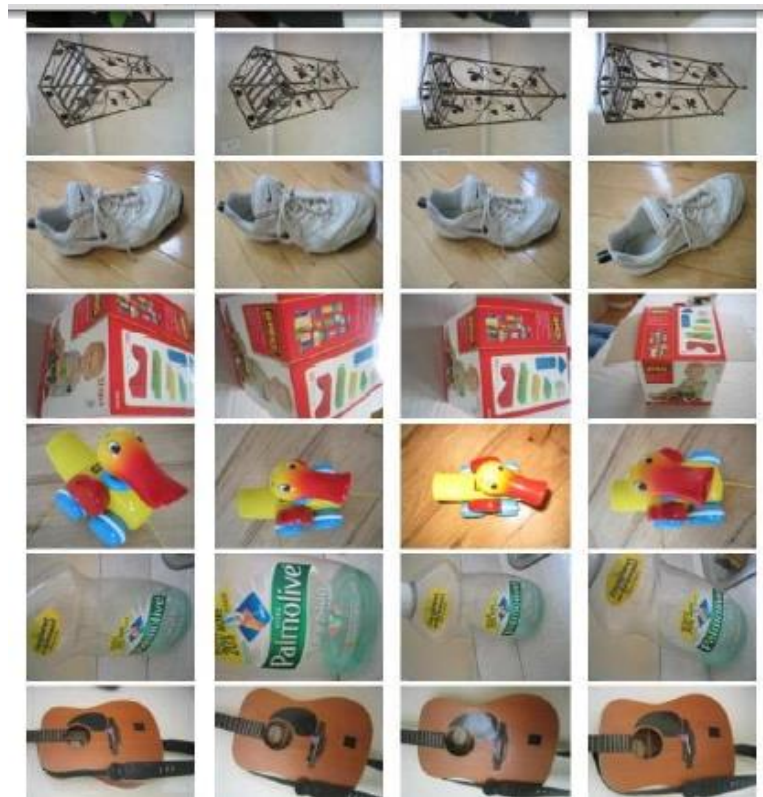
- How to summarize the content of an entire image? And estimate overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval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

Vocabulary size



Influence on performance, sparsity

Results for recognition task with 6347 images



Nister & Stewenius, CVPR 2006

Visual words/bags of words

Pro

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

Cons

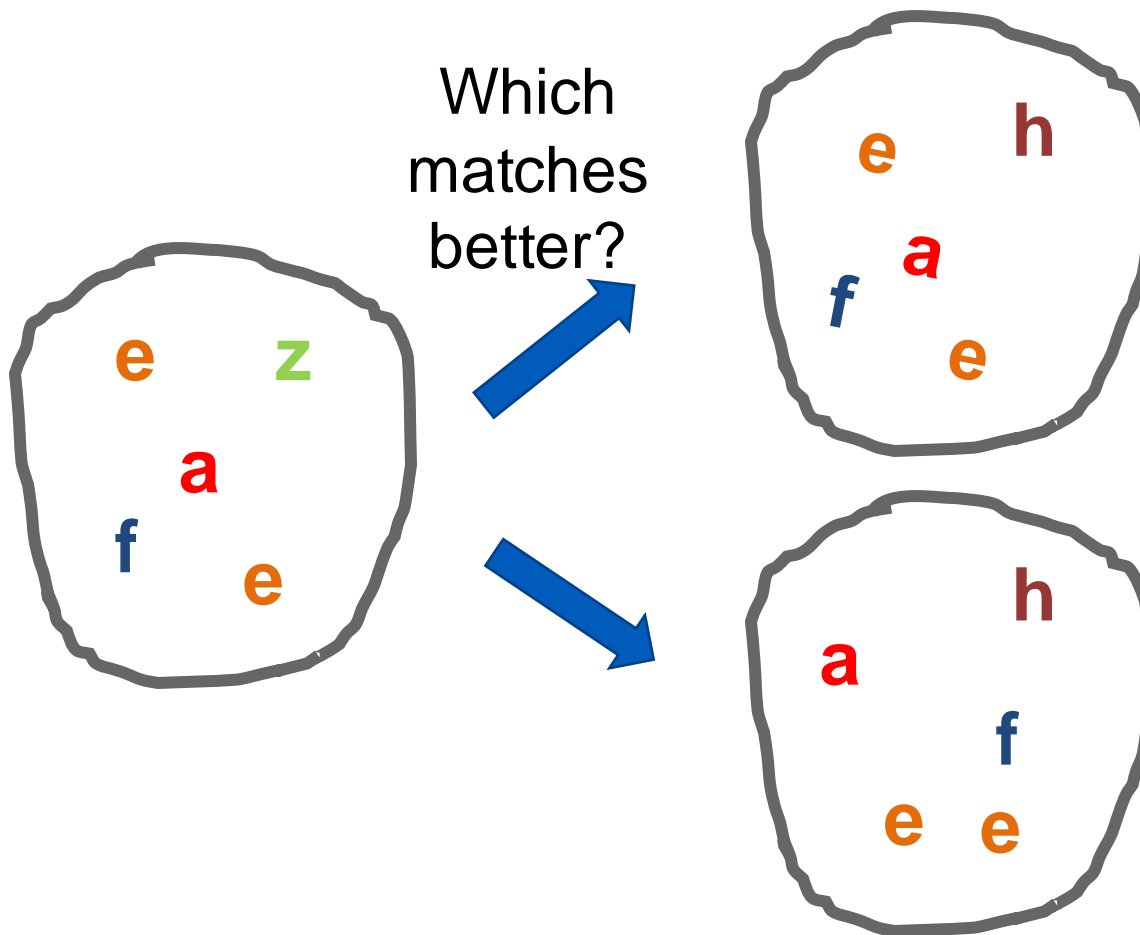
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear
- basic model ignores geometry – must verify afterwards, or encode via features

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



Can we be more accurate?

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



Real objects have consistent geometry

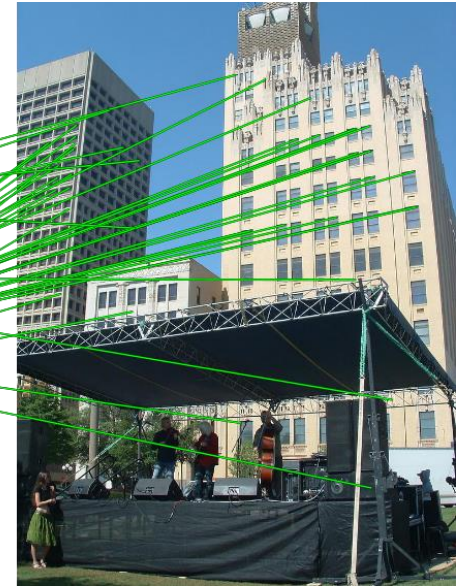
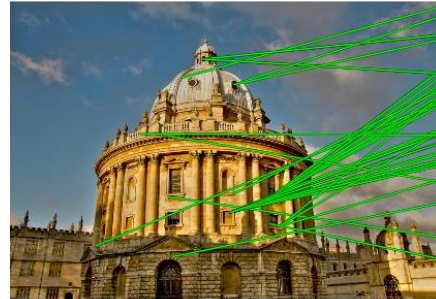
Spatial Verification

Query



DB image with high BoW
similarity

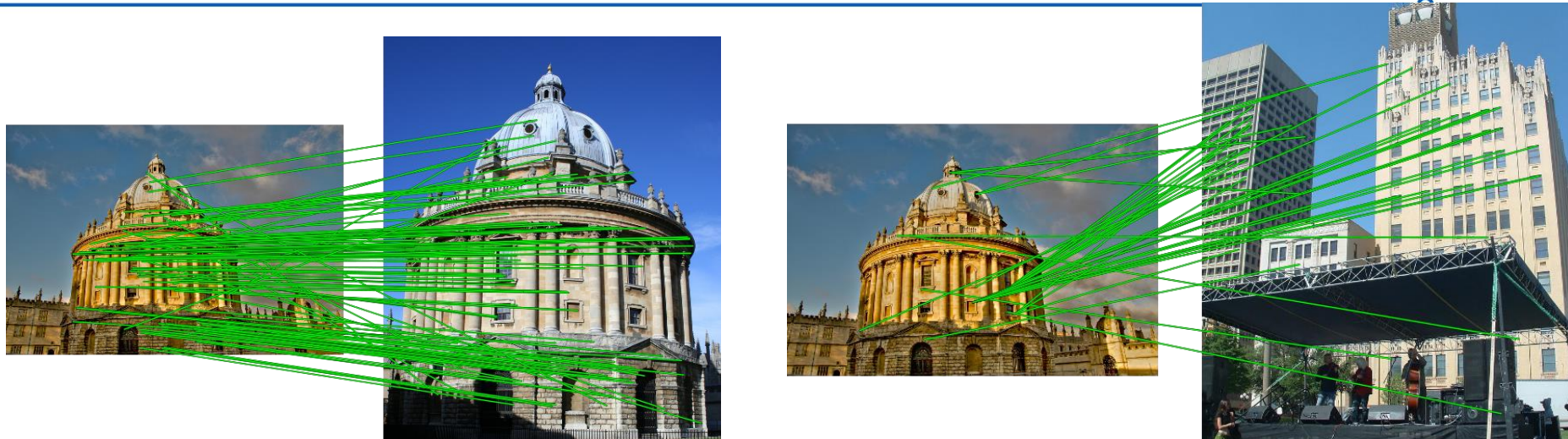
Query



DB image with high BoW
similarity

Both image pairs have many visual words in common.

Spatial verification



Only some of the matches are mutually consistent



Spatial Verification: three basic strategies

- RANSAC

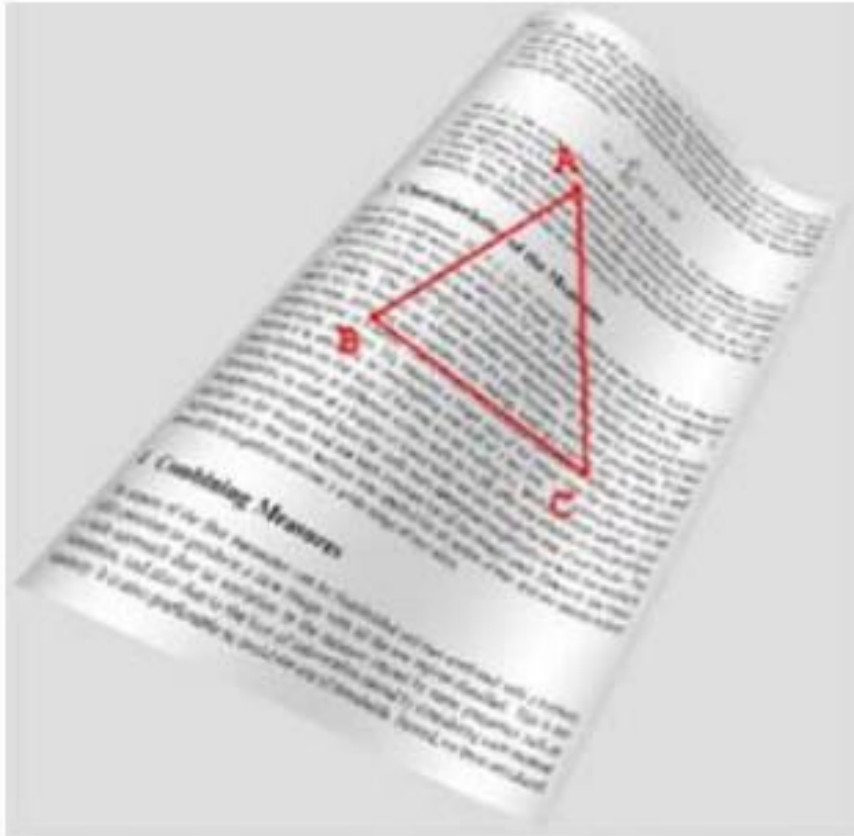
- Typically sort by BoW similarity as initial filter
- Verify by checking support (inliers) for possible transformations
 - e.g., “success” if find a transformation with $> N$ inlier correspondences

- Generalized Hough Transform

- Let each matched feature cast a vote on location, scale, orientation of the model object
- Verify parameters with enough votes

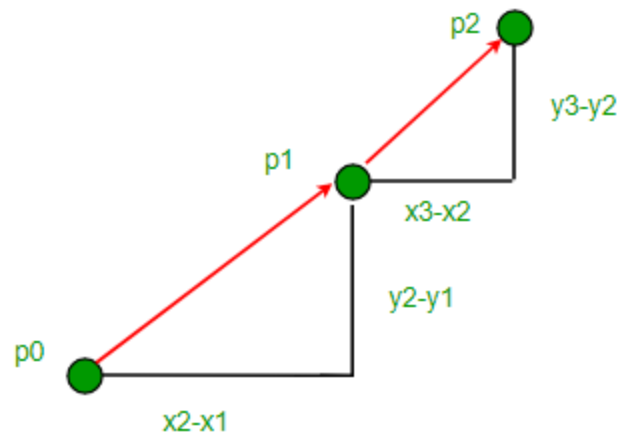
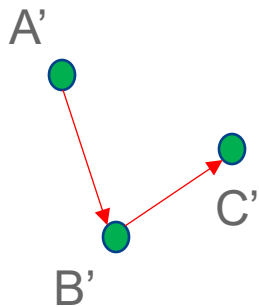
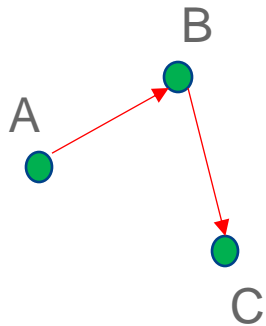
- Triplet Verification

Triplet Verification



Using Slope to Determine Orientation

Consider the slopes....

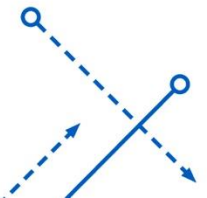


Slope(BC) – Slope(AB)?

< 0

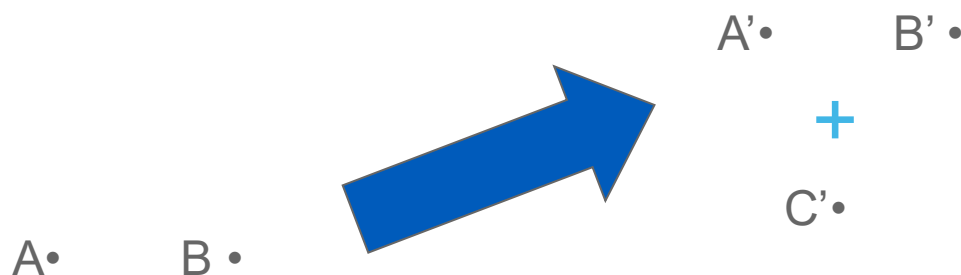
Slope(BC) – Slope(AB)?

> 0

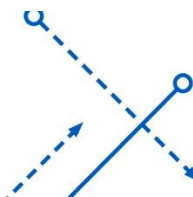
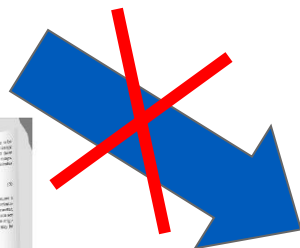
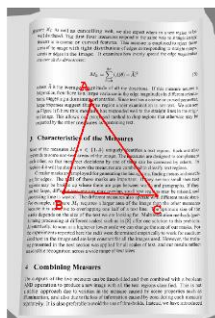
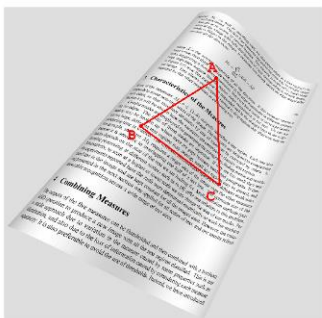


Triplet Verification

- Use the spatial relationship between “visual words”.
- Orientation of triplet is invariant under rotation, translation, scaling, warping and even crinkling



$$\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}$$



Triplet Verification: Scoring

- When viewed from another angle

$$\text{Sign}\left(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}\right) \times \text{Sign}\left(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix}\right) = 1$$

- Score is simply

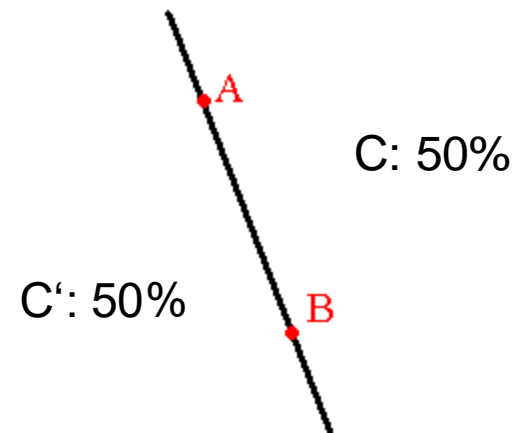
$$\sum_{A,B,C \in S} \left(\text{Sign}\left(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}\right) \times \text{Sign}\left(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix}\right) \right)$$

Triplet Verification: Failure rate

- Assume one triplet:
 - $P(A,B,C \text{ clockwise}) = 0.5$
 - $P(A,B,C \text{ anticlockwise}) = 0.5$
- Triplets are independent.
- Then possibility that M triplets out of N triplets accidentally satisfies orientation verification is

$$Q(N,M) = \left(\frac{1}{2}\right)^N \binom{N}{M}$$

N	10	30	50	60
M	5	20	40	50
Q(N,M)	0.25	0.027	0.0001	6.5×10^{-8}



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?
 - Precision, Recall, and F1 score

Precision-Recall Curve (Recap)

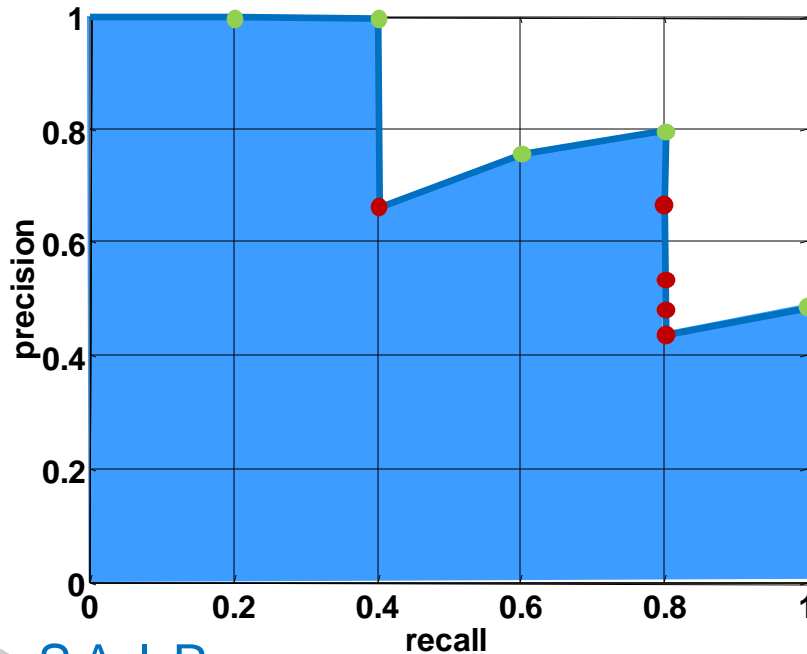
Database size: 10 images
Relevant (total): 5 images

$\text{precision} = \frac{\# \text{relevant}}{\# \text{returned}}$
 $\text{recall} = \frac{\# \text{relevant}}{\# \text{total relevant}}$

Query



Results (ordered):



Summary

- **Matching local invariant features**
 - Useful not only to provide matches for multi-view geometry, but also to find objects and scenes.
- **Bag of words** representation: quantize feature space to make discrete set of visual words
 - Summarize image by distribution of words
 - Index individual words
- **Inverted index**: pre-compute index to enable faster search at query time
- **Recognition of instances via alignment**: matching local features followed by spatial verification
 - Robust fitting: RANSAC

Lessons from a Decade Later

- For ***instance* retrieval** (this lecture)
 - Still widely used **in the field of simultaneously localization and mapping (SLAM)**.
 - Learn better local features (replace SIFT), e.g., MatchNet
 - or learn better image embeddings (replace the histograms of visual features), e.g., Vo and Hays 2016.
 - or learn to do spatial verification, e.g., DeTone, Malisiewicz, and Rabinovich 2016.
 - or learn a monolithic deep network to recognition all locations, e.g., Google's PlaNet 2016.

Lessons from a Decade Later

- For ***Category*** recognition

- Bag of Feature models remained the state of the art until Deep Learning.
- Spatial layout either isn't that important or its too difficult to encode.
- Quantization error is, in fact, the bigger problem. Advanced feature encoding methods address this.
- BoW is inspiring deep models, e.g., [NetVLAD](#) (deep learning model for place recognition).

Things to remember

- Object instance recognition
 - Find keypoints, compute descriptors
 - Match descriptors
 - Vote for / fit affine parameters
 - Return object if $\# \text{ inliers} > T$
- Keys to efficiency
 - Visual words
 - Used for many applications
 - Inverse document file
 - Used for web-scale search

