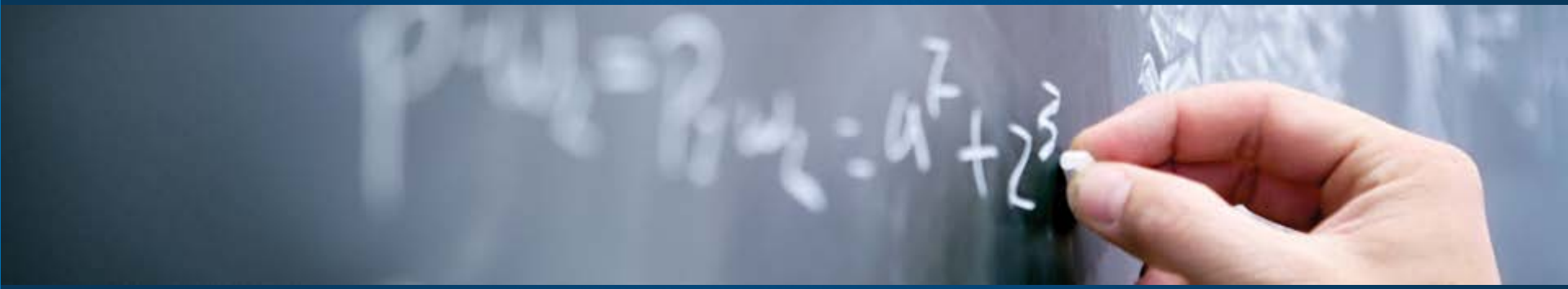
Анализ изображений и видео

Лекция 6: Поиск по подобию. Поиск нечетких дубликатов.

Наталья Васильева

nvassilieva@hp.com
HP Labs Russia



26 октября 2012, Computer Science Center

Поиск по визуальному подобию: описание

Задача: найти картинки, похожие на запрос визуально и семантически

Похожи	Отдаленно	Не похожи
		
		

Поиск нечетких дубликатов: описание

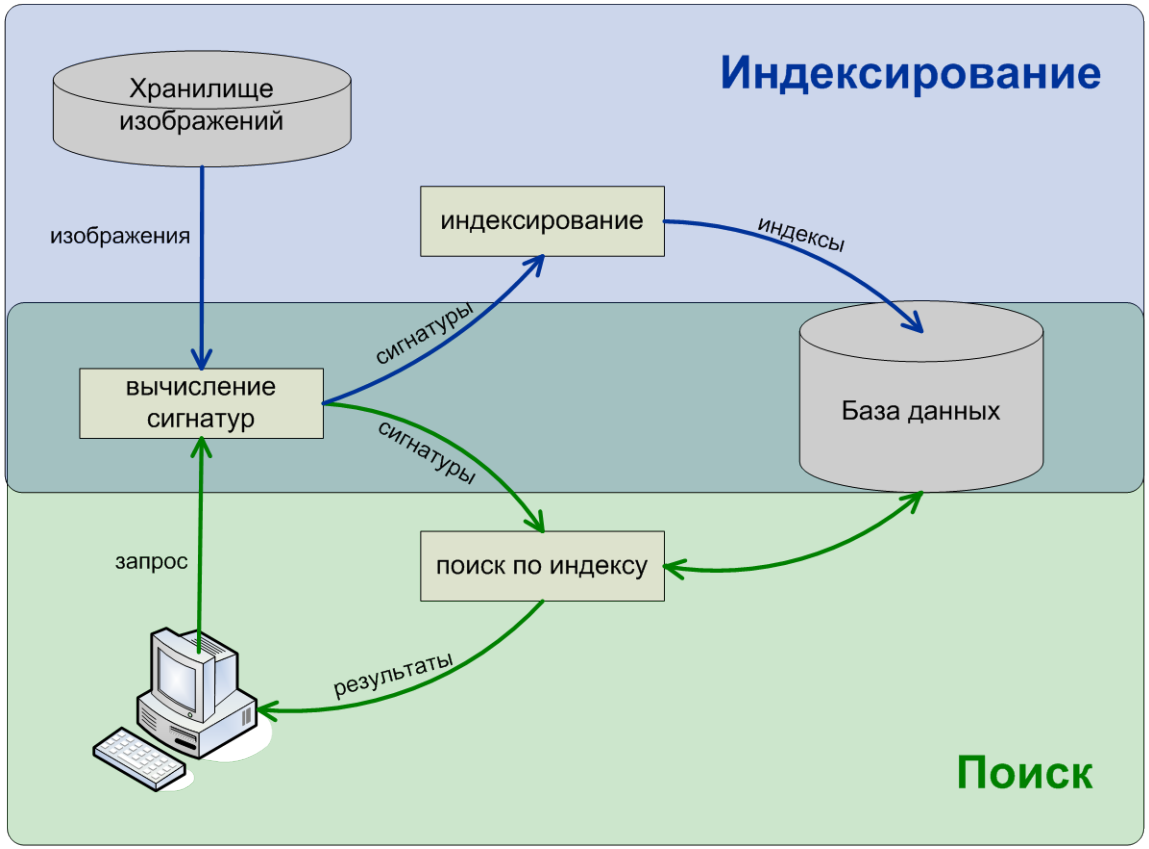
Задача: найти группы нечетких дублей в коллекции

Дубли

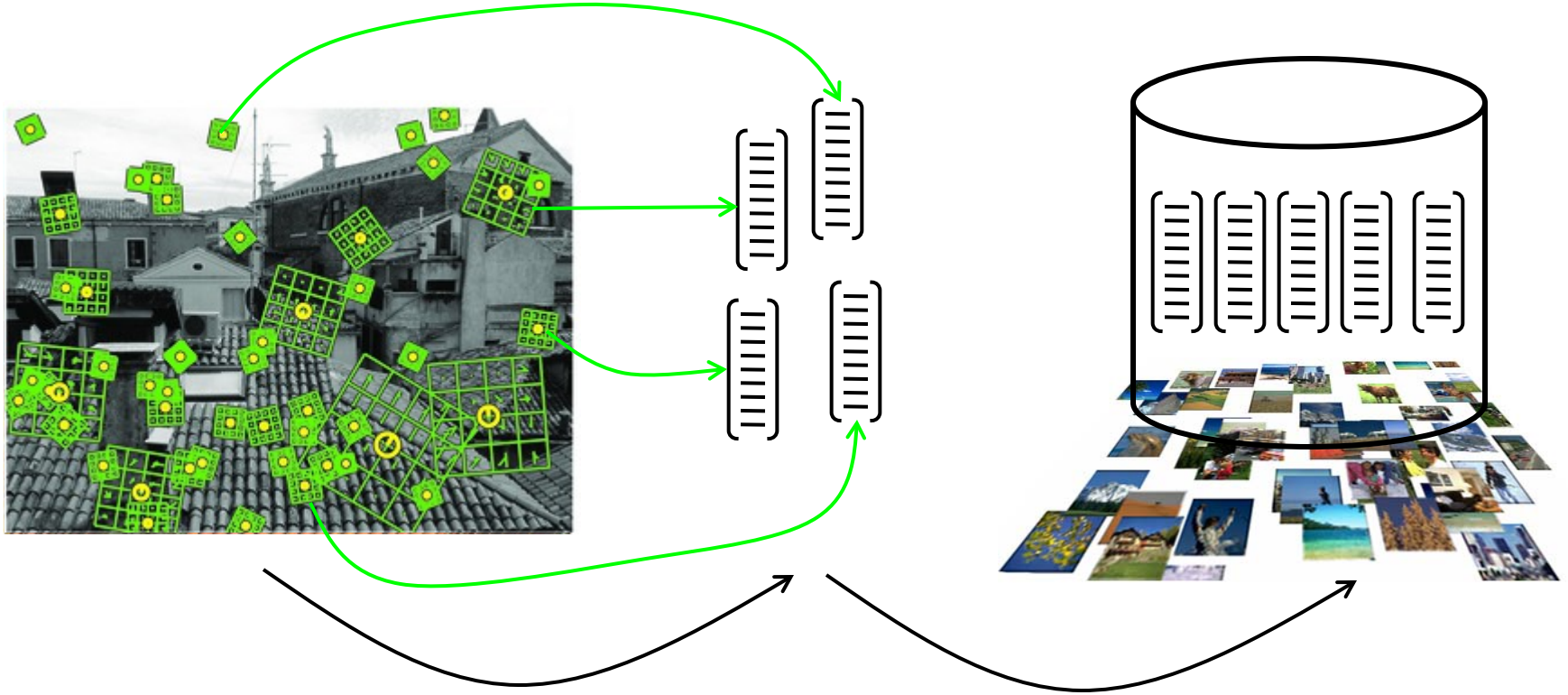
Не дубли

Традиционная архитектура систем CBIR

Поиск по содержанию, по визуальному подобию
Content Based Image Retrieval (CBIR)



Индексирование изображений

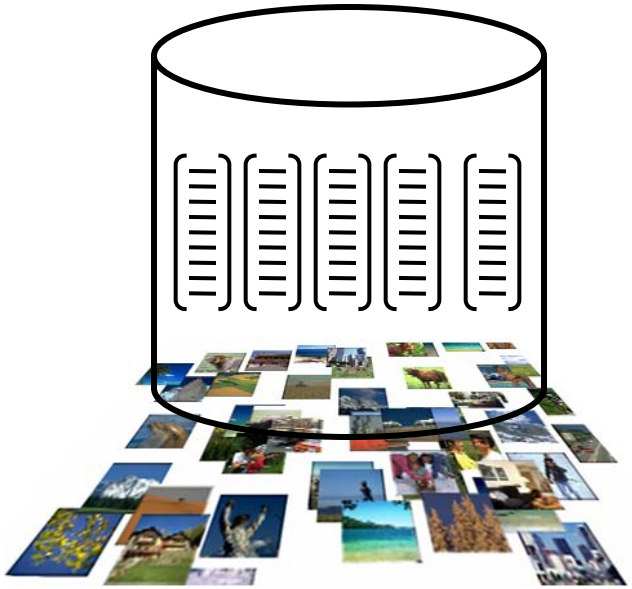


Построение векторов признаков

Помещение векторов в хранилище, построение индекса

Fig. credit: **http://www.vlfeat.org/**

Поиск изображений

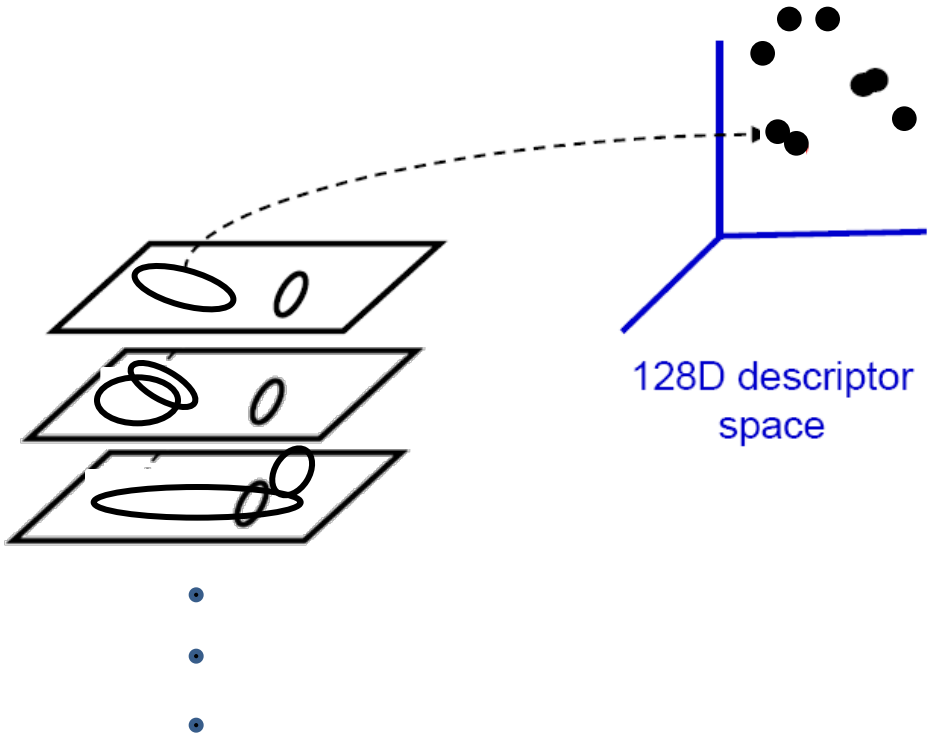


- Данные высокой размерности
  - Среднее значение для каждого цветового канала: dim(x) = 3
  - Гистограмма по одному каналу: dim(x) = 256
  - ICA-based texture: dim(x) = 21*30
  - SIFT: dim(x) = 128 * N
- Поиск по подобию, поиск ближайших соседей

Необходим способ так организовать хранение, чтобы можно было по запросу быстро и эффективно находить в хранилище наиболее похожие объекты

Индексирование

Каждый вектор – точка в пространстве высокой размерности (например, dim=128 для дескрипторов SIFT)



Slide credit: K. Grauman, B. Leibe

Индексирование

Соседние точки в пространстве признаков соответствуют близким векторам, которые соответствуют подобию изображений

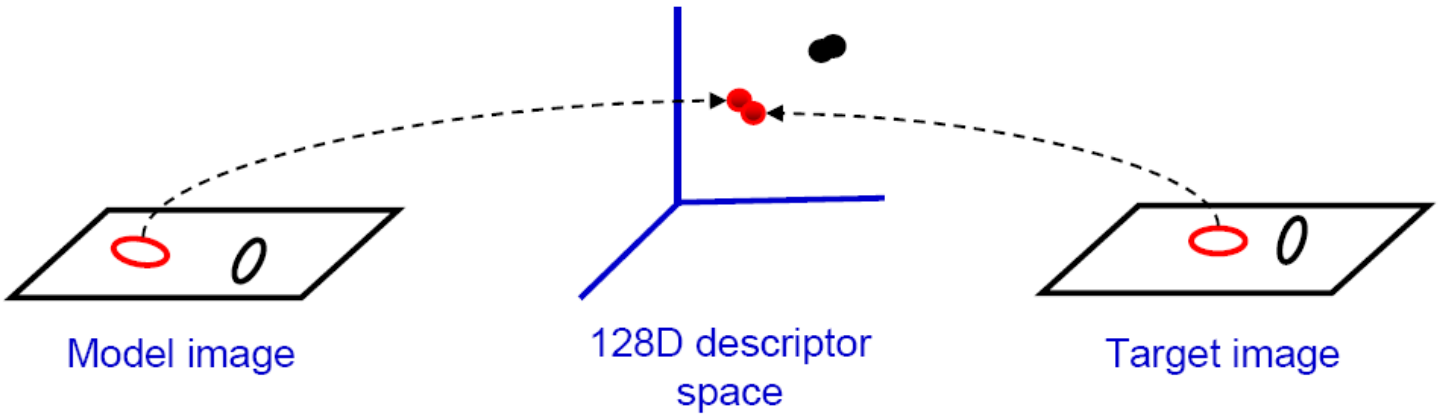
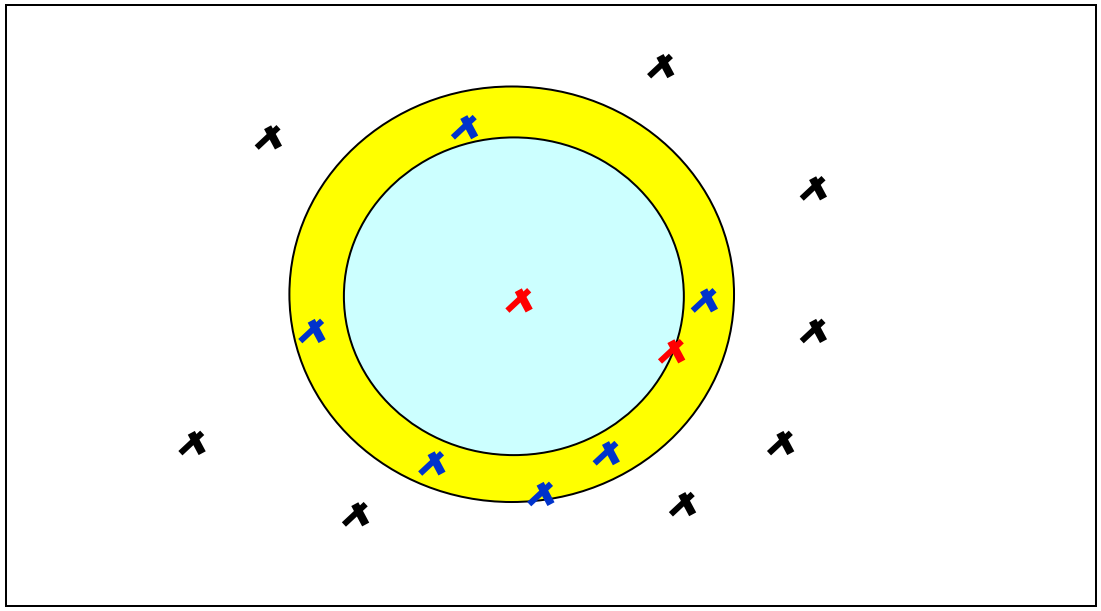


Fig. credit: A. Zisserman

Поиск ближайших соседей

ε-Nearest Neighbor Search (ε - NNS)



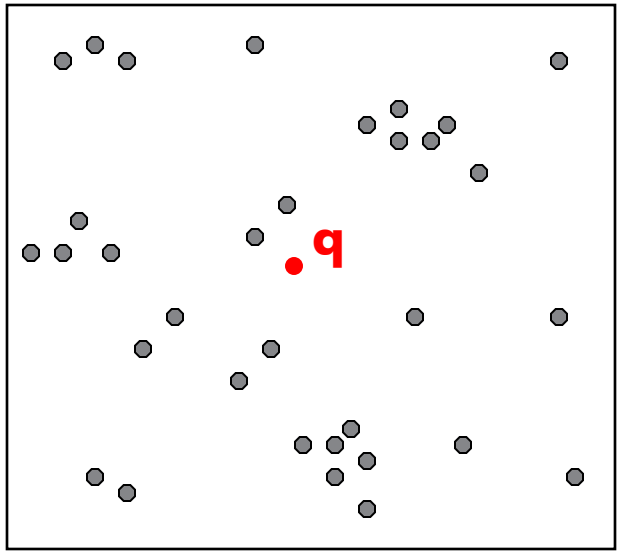
Для запроса q найти подмножество всех точек p ∈ P (P – метрическое пространство), таких что:

$$dist(q, p) \leq (1+\varepsilon) \min_{r \in P} dist(q, r)$$

Можно обобщить до k–nearest neighbor search (K >1) (Top-K)

Поиск ближайших соседей

k–Nearest Neighbor Search (Top-k search)



Для запроса q найти подмножество $A \subseteq P$ (P – метрическое пространство):

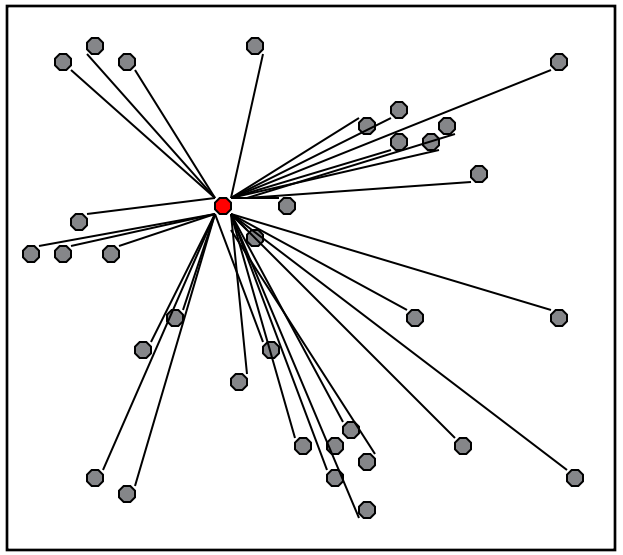
$$|A| = k, \quad \forall p_1 \in A \quad \forall p_2 \in P \setminus A: \quad dist(q, p_1) \leq dist(q, p_2)$$

Наивный подход

Для запроса q

Вычислить dist(q,p) для всех точек $p \in P$

O(N) для каждого запроса!
(|P|=N)



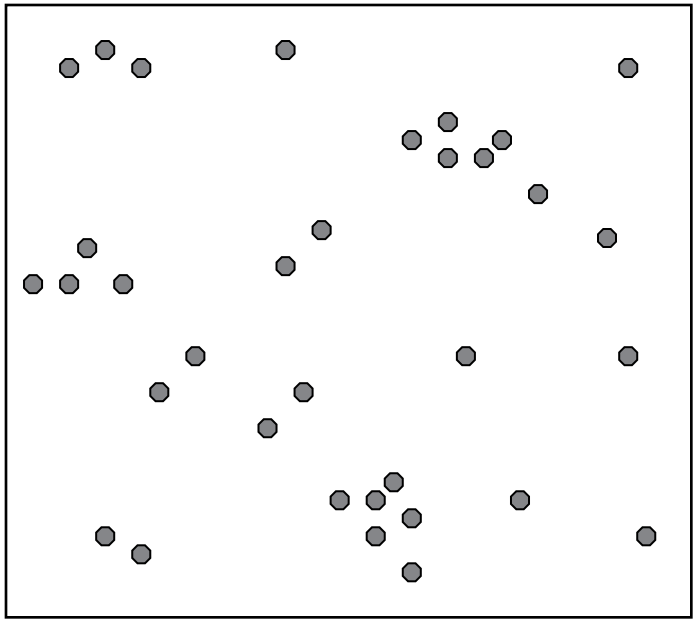
Некоторые индексные структуры

Деревья
- R-tree – подходит только для небольших размерностей (2D), пересечение узлов
- k-D tree – неэффективны в случае ассиметричных данных высокой размерности
- Vantage-Point tree (VP tree) (один из вариантов метрических деревьев)

Обратный индекс
- Vocabulary tree

Хэширование
- LSH

KD-Tree Construction

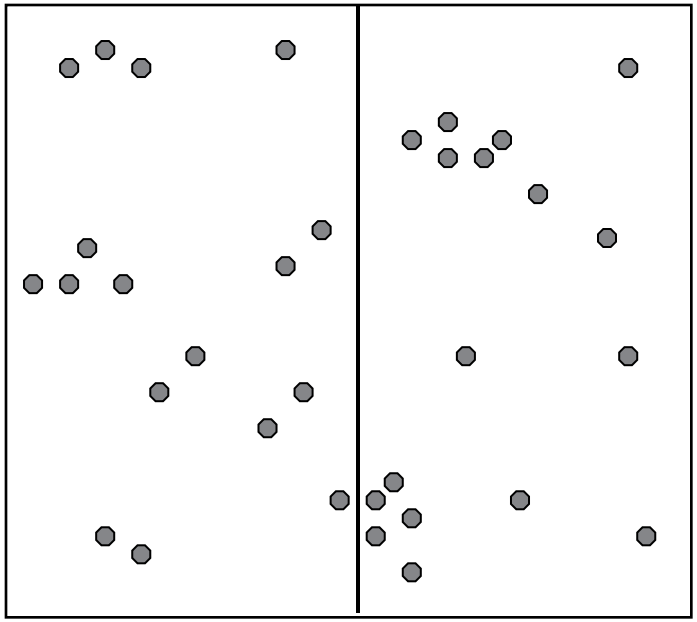


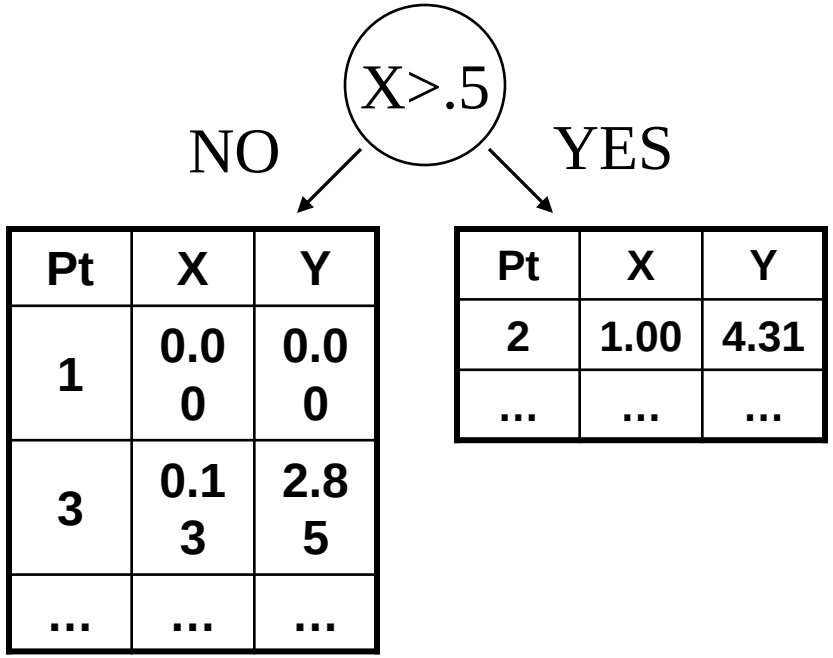
Pt	X	Y
1	0.00	0.00
2	1.00	4.31
3	0.13	2.85
…	…	…

We start with a list of n-dimensional points.

Slide credit: Brigham Anderson

KD-Tree Construction





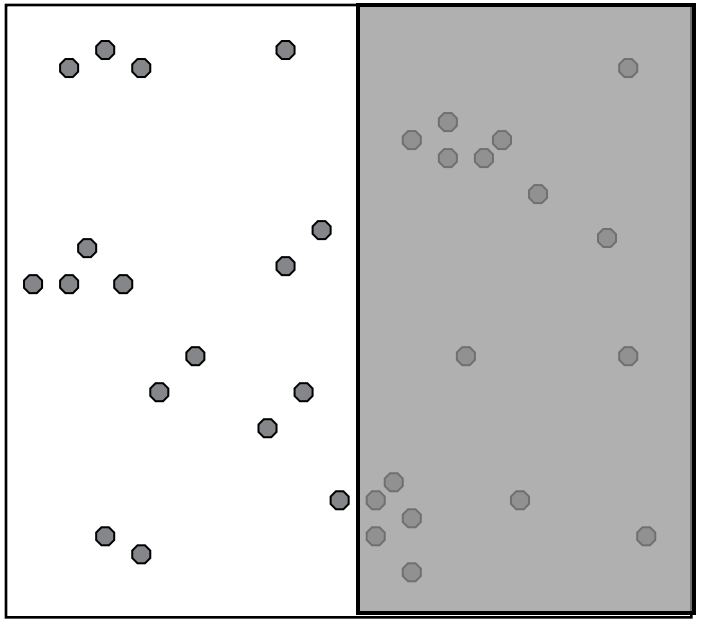
Pt	X	Y
1	0.00	0.00
3	0.13	2.85
…	…	…

Pt	X	Y
2	1.00	4.31
…	…	…

We can split the points into 2 groups by choosing a dimension X and value V and separating the points into X > V and X <= V.

Slide credit: Brigham Anderson

KD-Tree Construction



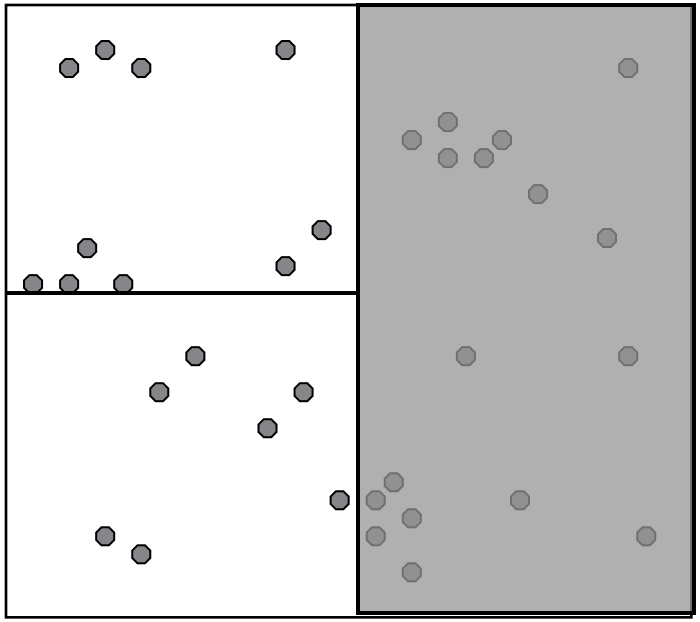
Pt	X	Y
1	0.00	0.00
3	0.13	2.85
…	…	…

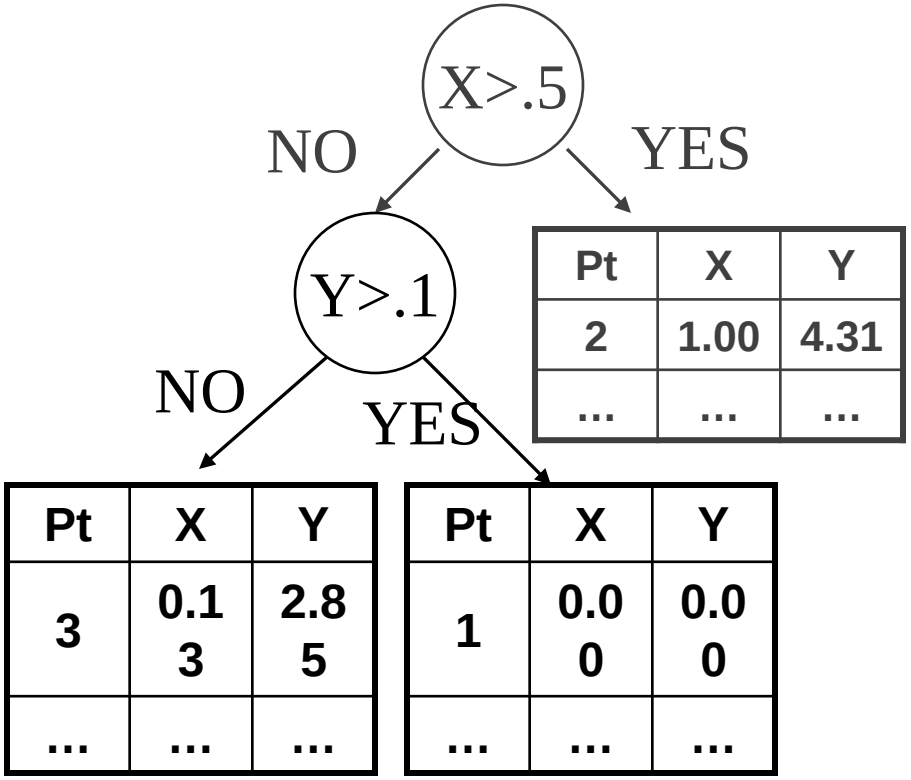
Pt	X	Y
2	1.00	4.31
…	…	…

We can then consider each group separately and possibly split again (along same/different dimension).

Slide credit: Brigham Anderson

KD-Tree Construction

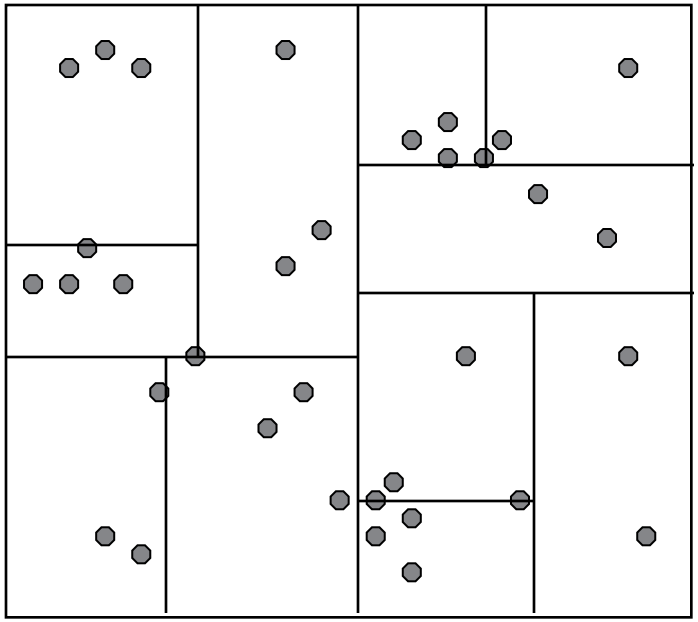


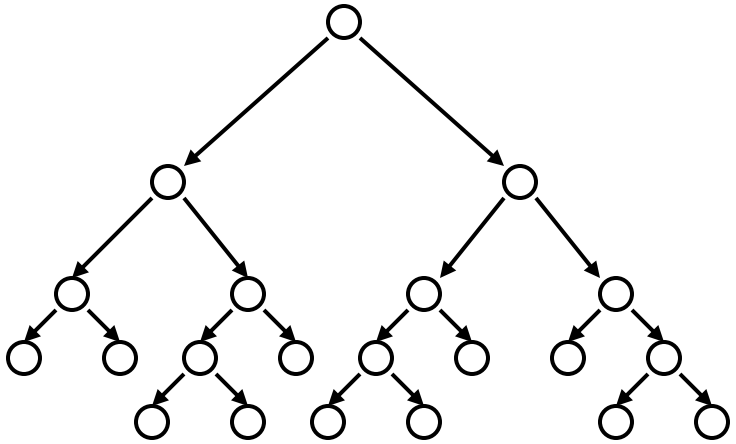


We can then consider each group separately and possibly split again (along same/different dimension).

Slide credit: Brigham Anderson

KD-Tree Construction

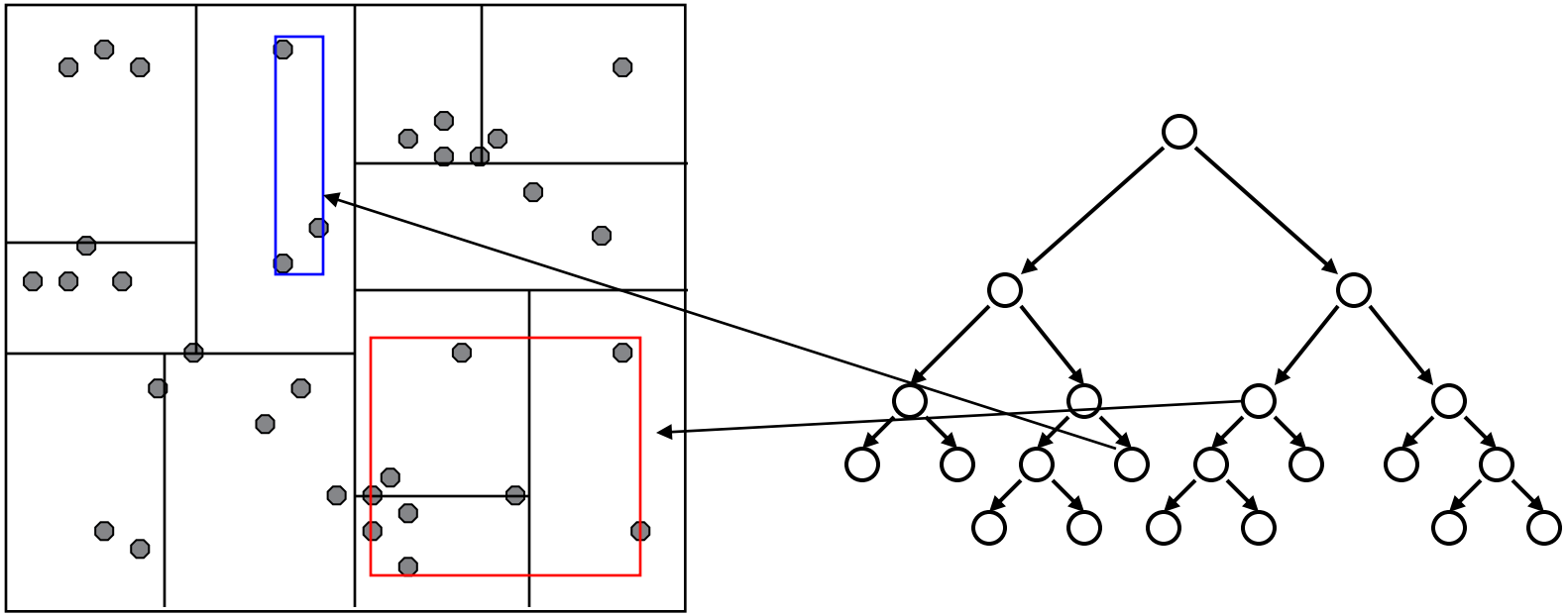




We can keep splitting the points in each set to create a tree structure. Each node with no children (leaf node) contains a list of points.

Slide credit: Brigham Anderson

KD-Tree Construction



We will keep around one additional piece of information at each node. The (tight) bounds of the points at or below this node.

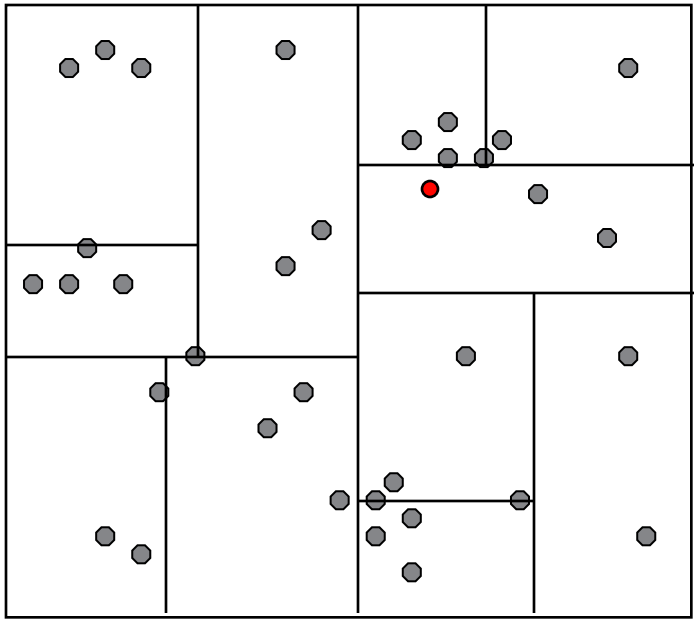
Slide credit: Brigham Anderson

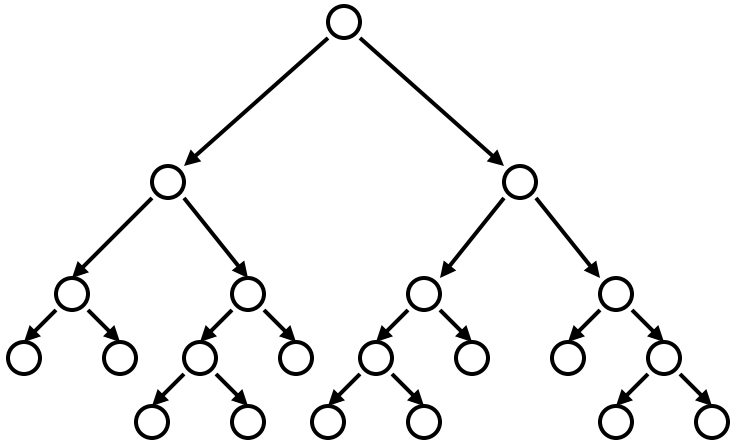
KD-Tree Construction

Use heuristics to make splitting decisions:
Which dimension do we split along? Widest
Which value do we split at? Median of value of that split dimension for the points.
When do we stop? When there are fewer then m points left OR the box has hit some minimum width.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

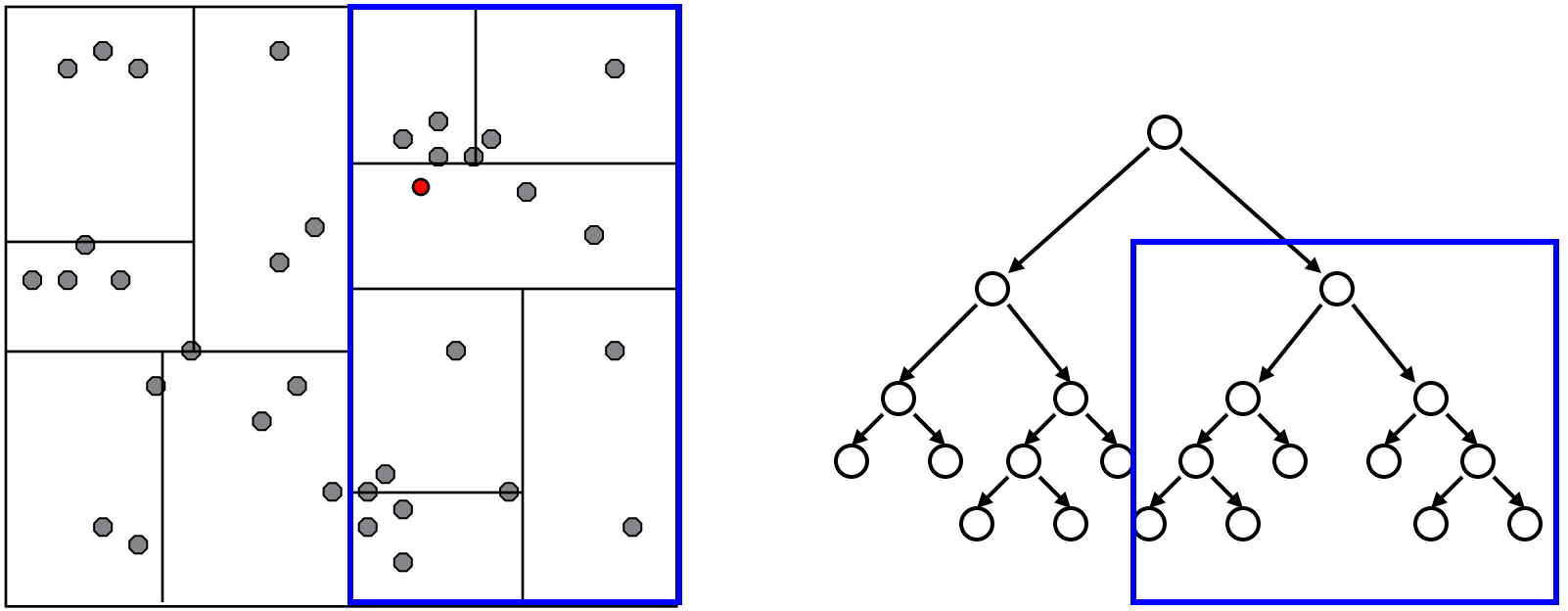




We traverse the tree looking for the nearest neighbor of the query point.

Slide credit: Brigham Anderson

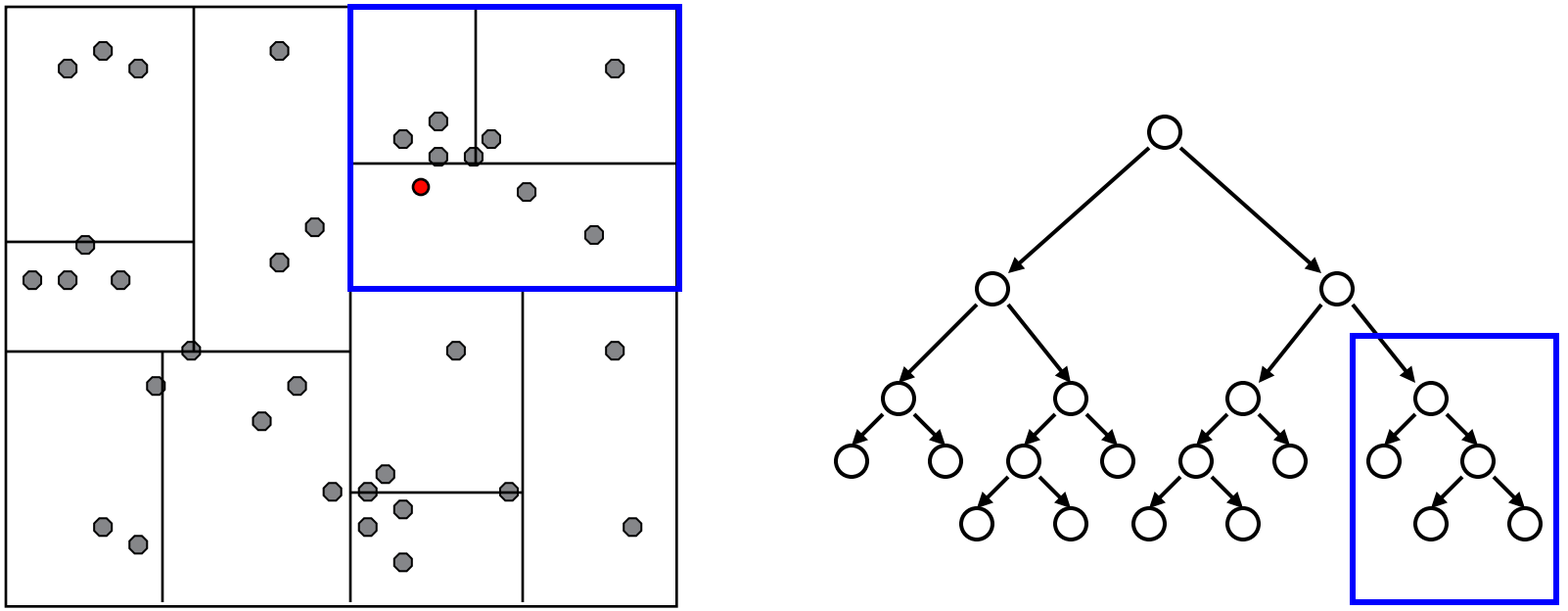
Nearest Neighbor with KD Trees



Examine nearby points first: Explore the branch of the tree that is closest to the query point first.

Slide credit: Brigham Anderson

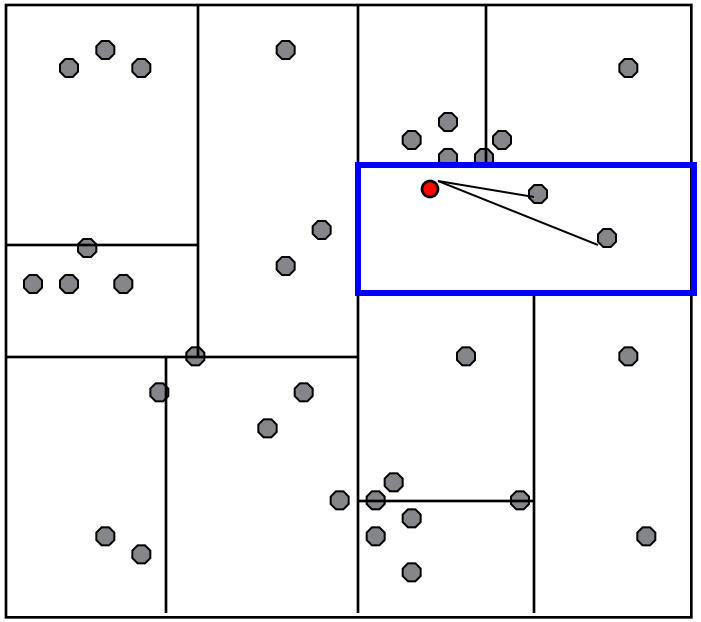
Nearest Neighbor with KD Trees

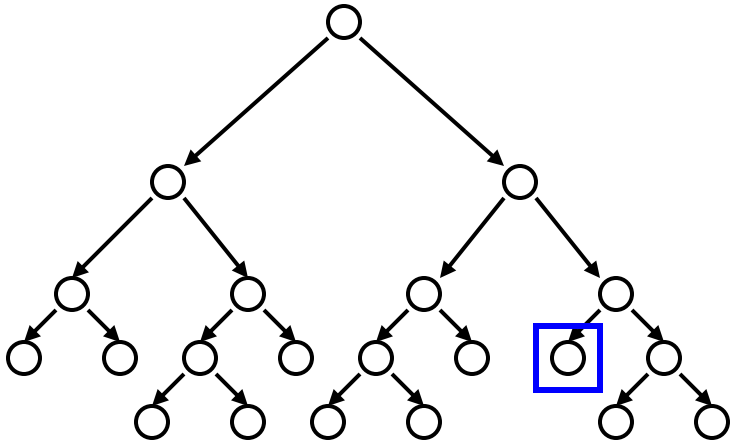


Examine nearby points first: Explore the branch of the tree that is closest to the query point first.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

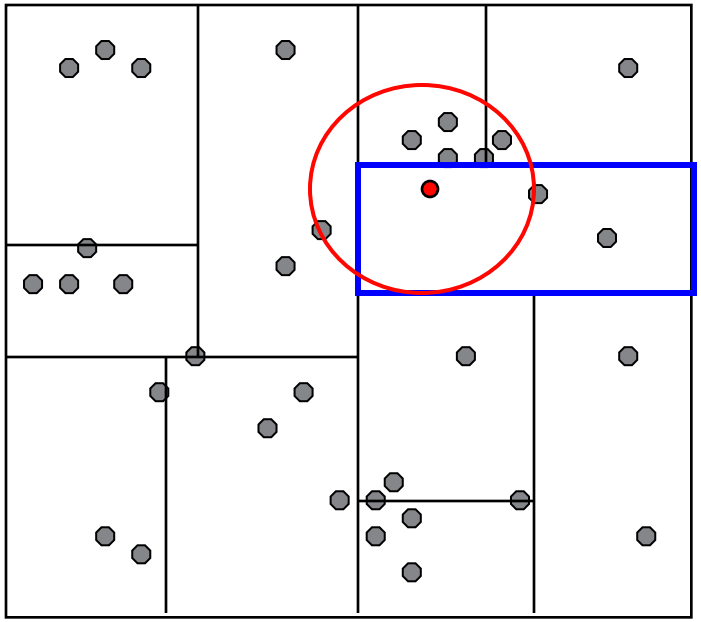


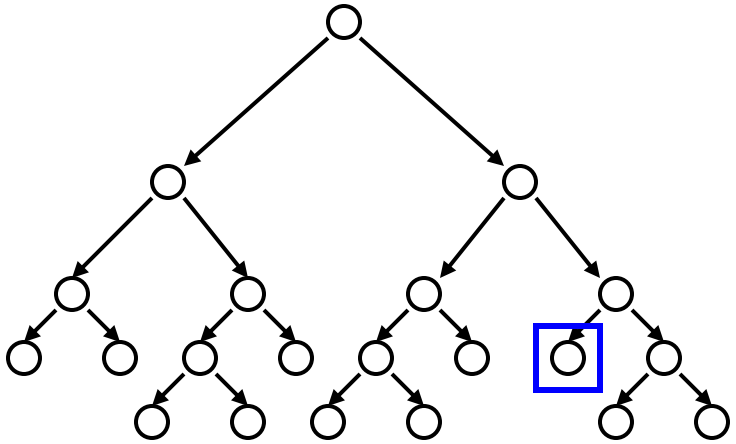


When we reach a leaf node: compute the distance to each point in the node.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

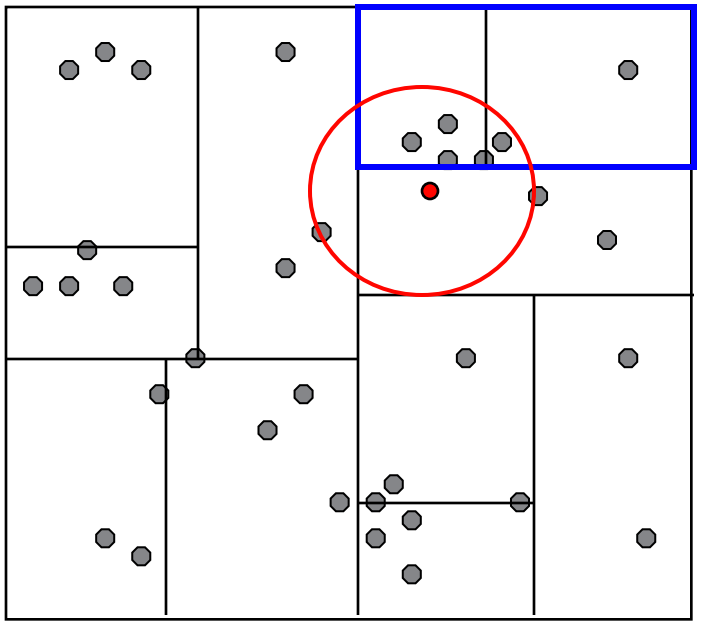


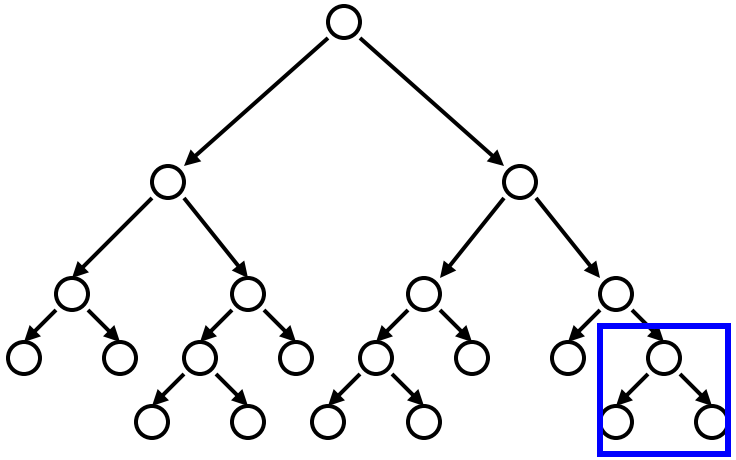


When we reach a leaf node: compute the distance to each point in the node.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

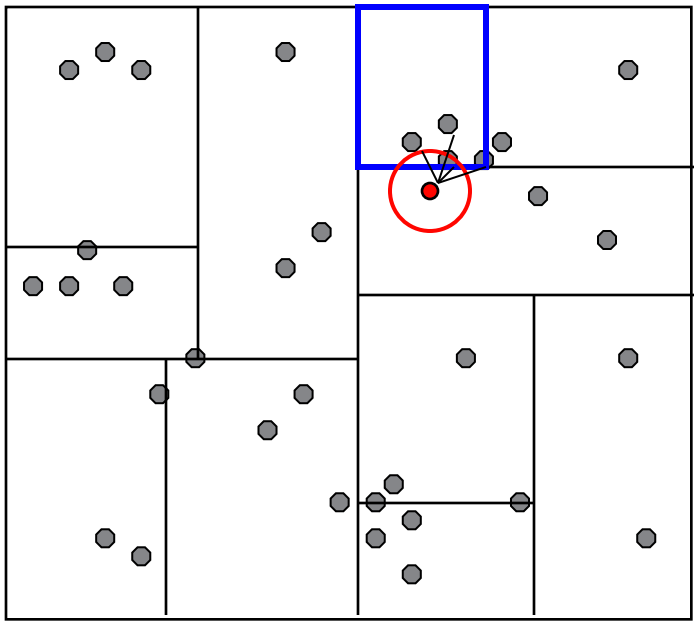


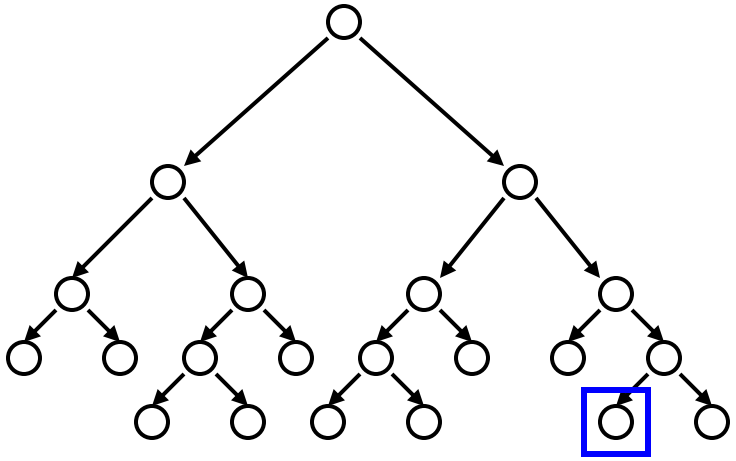


Then we can backtrack and try the other branch at each node visited.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

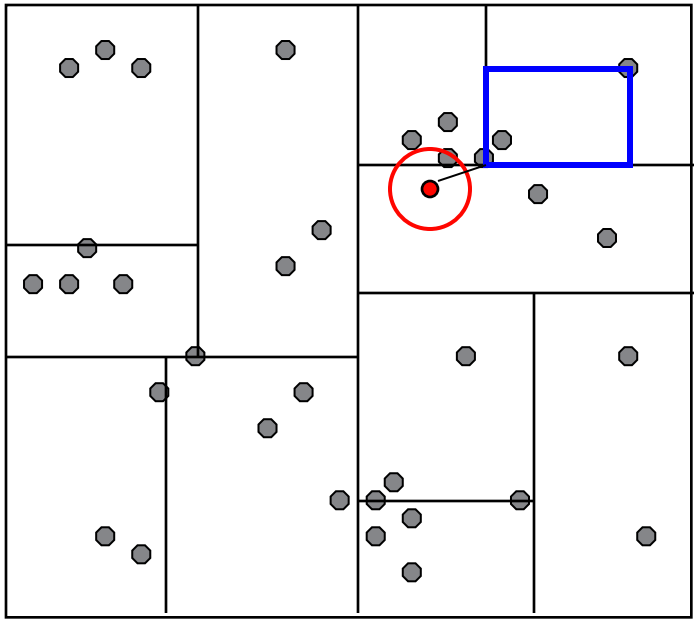


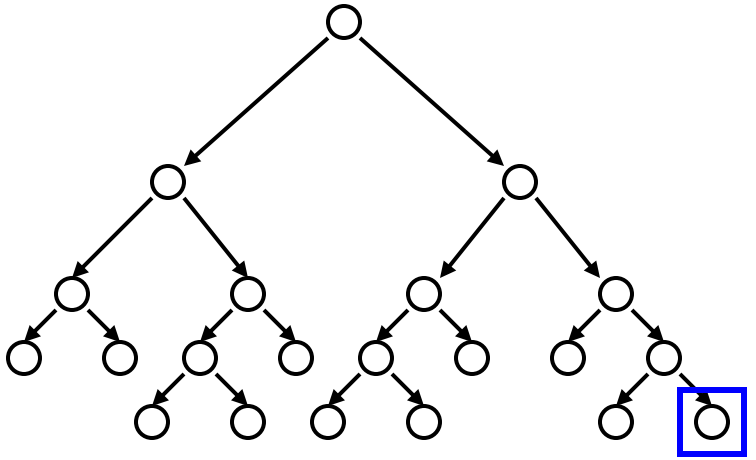


Each time a new closest node is found, we can update the distance bounds.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

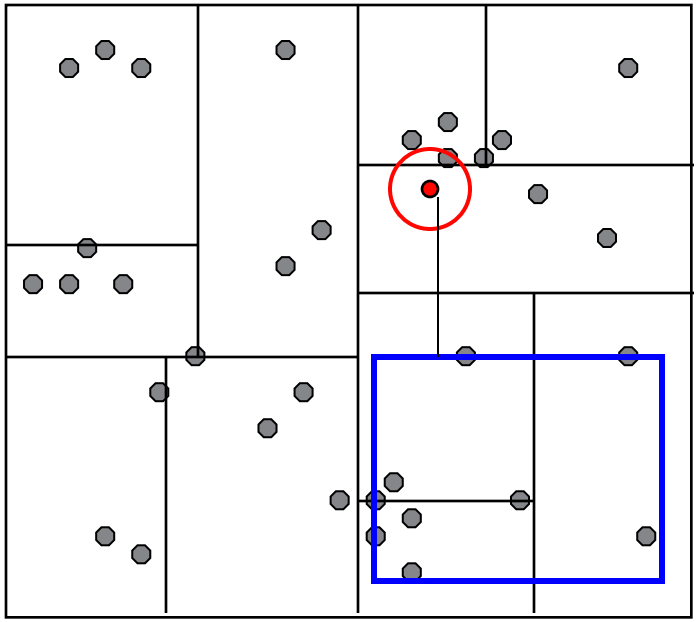


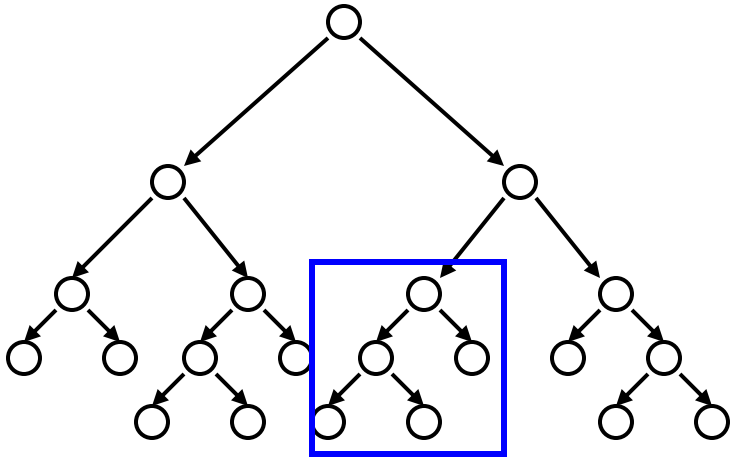


Using the distance bounds and the bounds of the data below each node, we can prune parts of the tree that could NOT include the nearest neighbor.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees

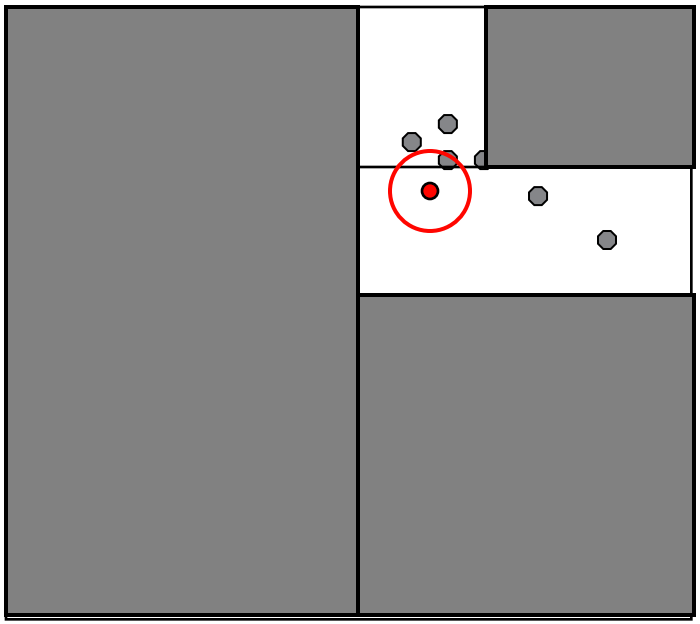


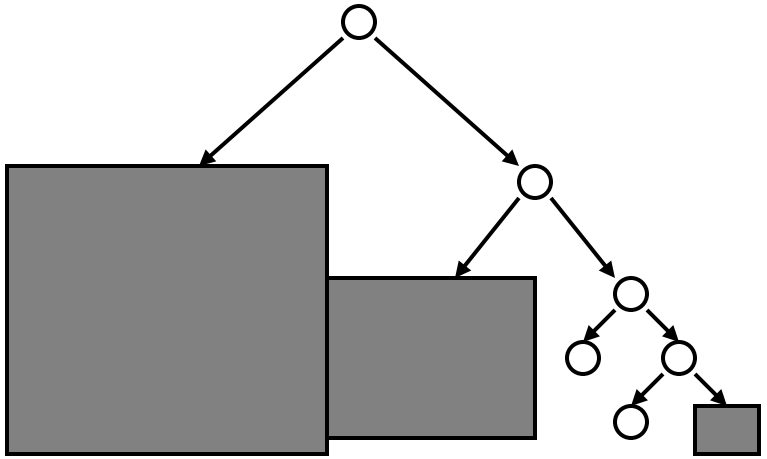


Using the distance bounds and the bounds of the data below each node, we can prune parts of the tree that could NOT include the nearest neighbor.

Slide credit: Brigham Anderson

Nearest Neighbor with KD Trees





Using the distance bounds and the bounds of the data below each node, we can prune parts of the tree that could NOT include the nearest neighbor.

Slide credit: Brigham Anderson

Spheres vs. Rectangles

Figure 1: vp-tree decomposition

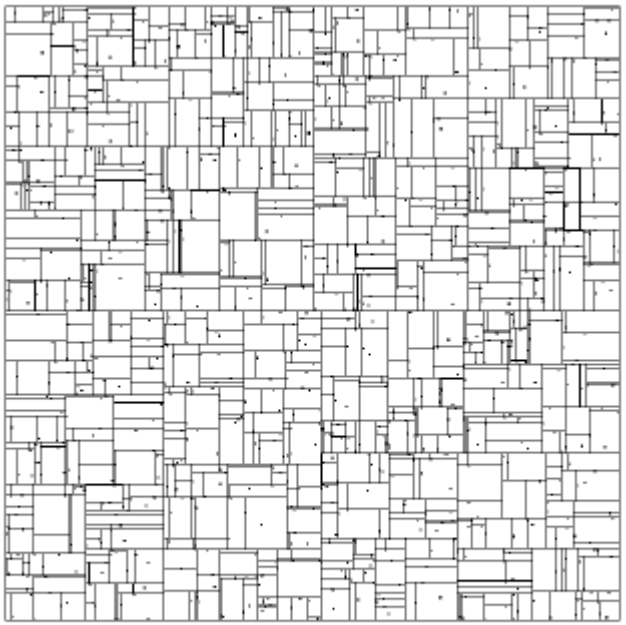
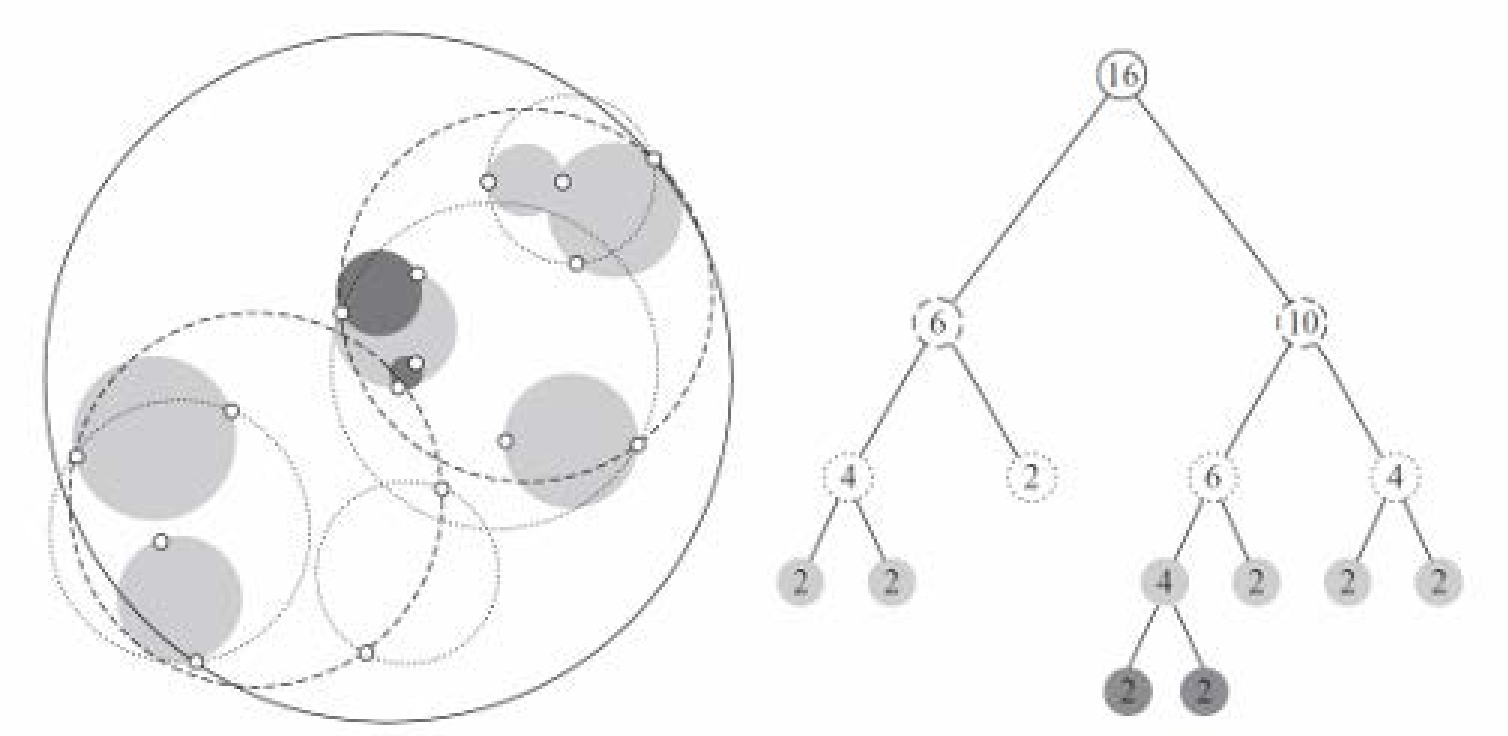


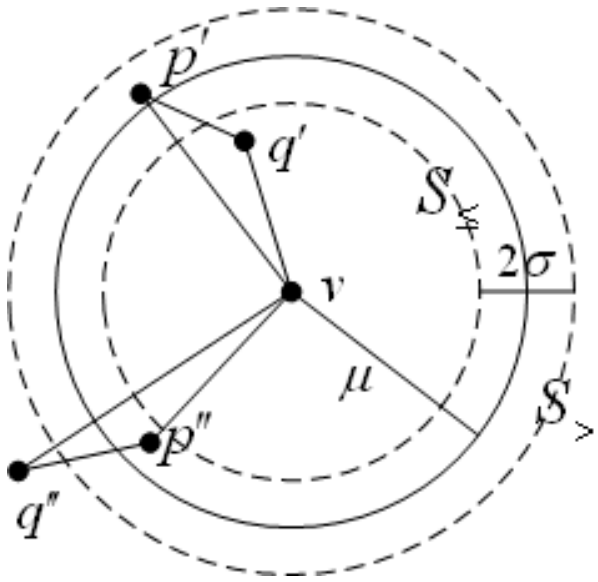
Figure 2: kd-tree decomposition

- $\text{ratio} = \dfrac{\text{Volume(Sphere)}}{\text{Volume(Cube)}} \leq 1$
- $\text{dimensionality} \uparrow \Rightarrow \text{ratio} \uparrow$
- relative distances

Метрические деревья



Vantage point method



$$d(v,q) \le \mu - \sigma \quad p \in S_{>}$$

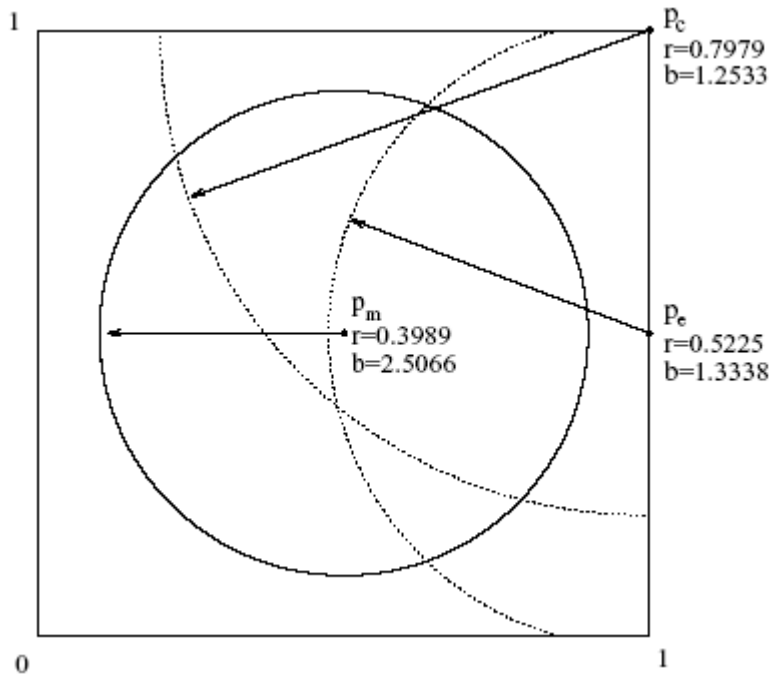
$$d(q,p) \ge |\, d(v,p) - d(v,q) \,| > |\, \mu - (\mu - \sigma) \,| = \sigma$$

$$d(v,q) > \mu + \sigma \quad p \in S_{\le}$$

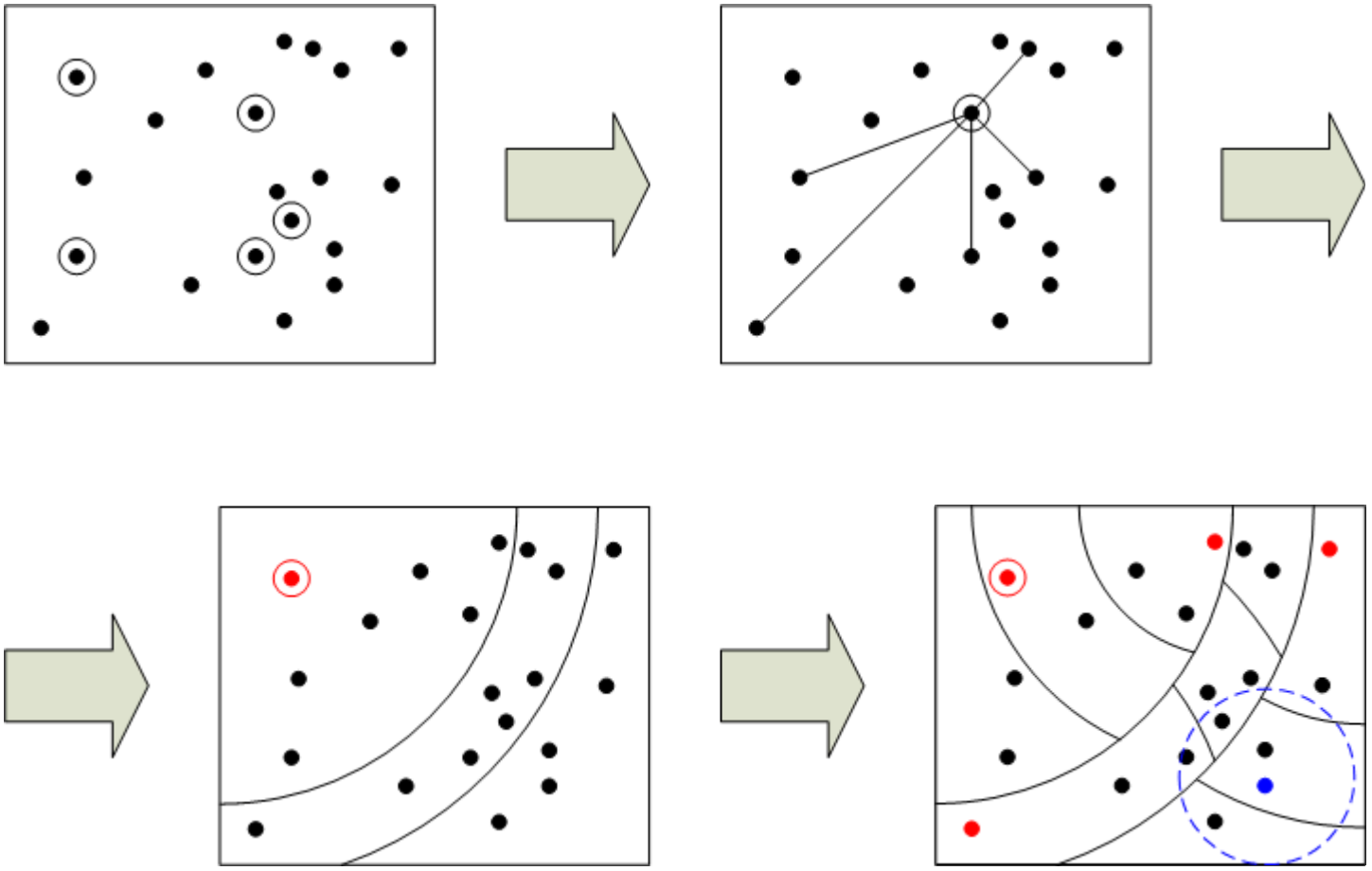
$$d(q,p) \ge |\, d(v,q) - d(v,p) \,| > |\, (\mu + \sigma) - \mu) \,| = \sigma$$

Как выбрать опорные точки?

- Минимальная длина разделяющей дуги
- “Угловые” точки пространства
- Чтобы дерево было сбалансированным
- Максимум среднеквадратичного отклонения



Построение VP-tree



Обратный индекс

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

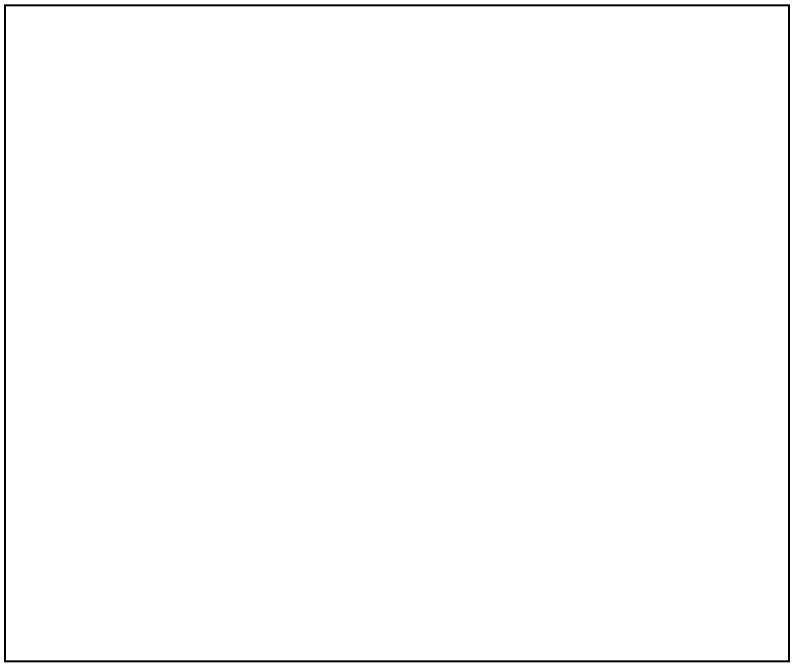
Для текстов: структура для эффективного поиска страниц, на которых упомянуто слово…

Наша цель: найти изображения, которым соответствуют заданные вектора признаков.

Чтобы использовать эту идею, нам надо построить из признаков словарь “визуальных слов”.

«Визуальные слова»: основная идея

Извлечем локальные признаки из нескольких изображений…

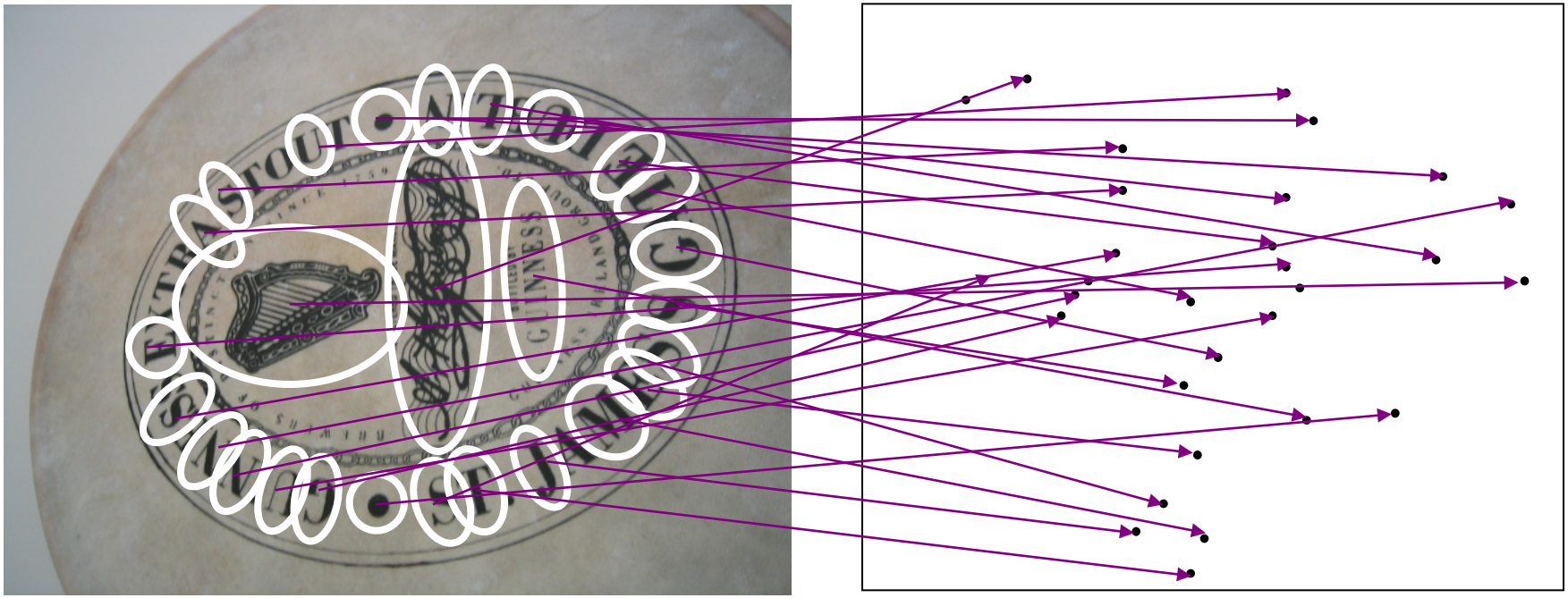


Пространство локальных признаков
(например, SIFT-дескрипторов)

Slide credit: D. Nister

«Визуальные слова»: основная идея

Извлечем локальные признаки из нескольких изображений…

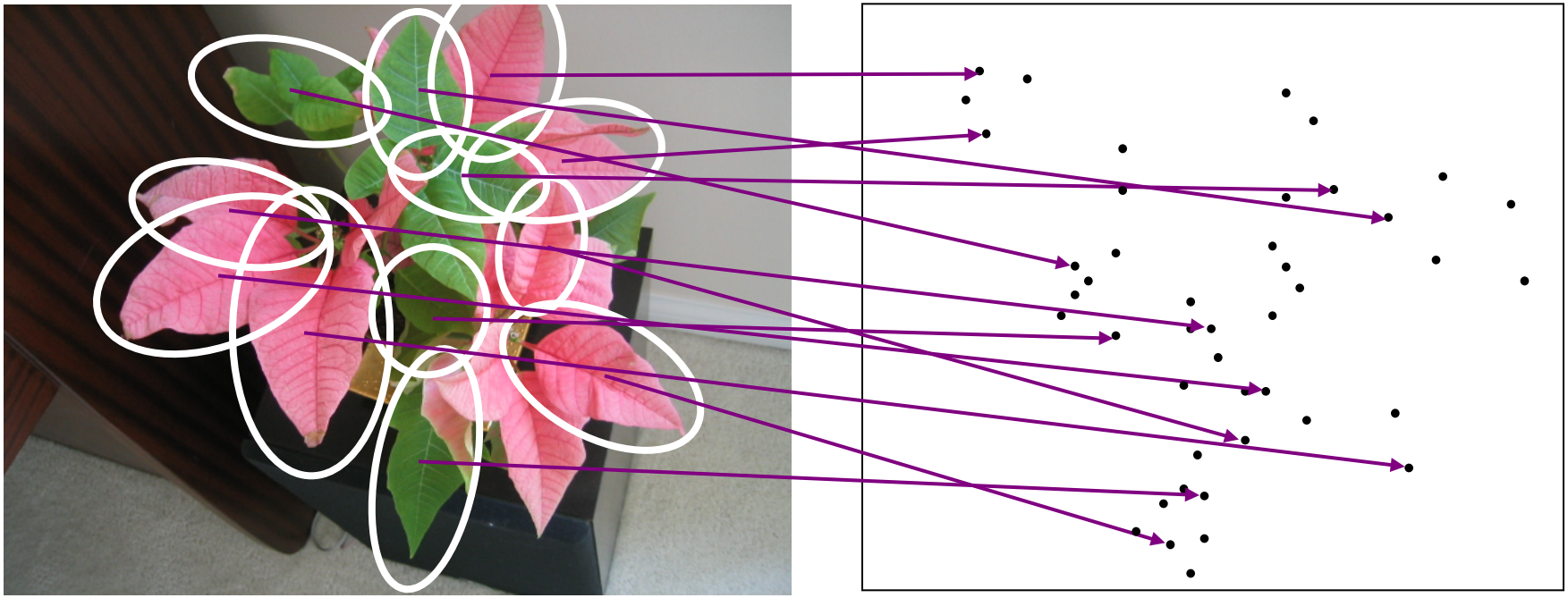


Пространство локальных признаков
(например, SIFT-дескрипторов)

Slide credit: D. Nister

«Визуальные слова»: основная идея

Извлечем локальные признаки из нескольких изображений…

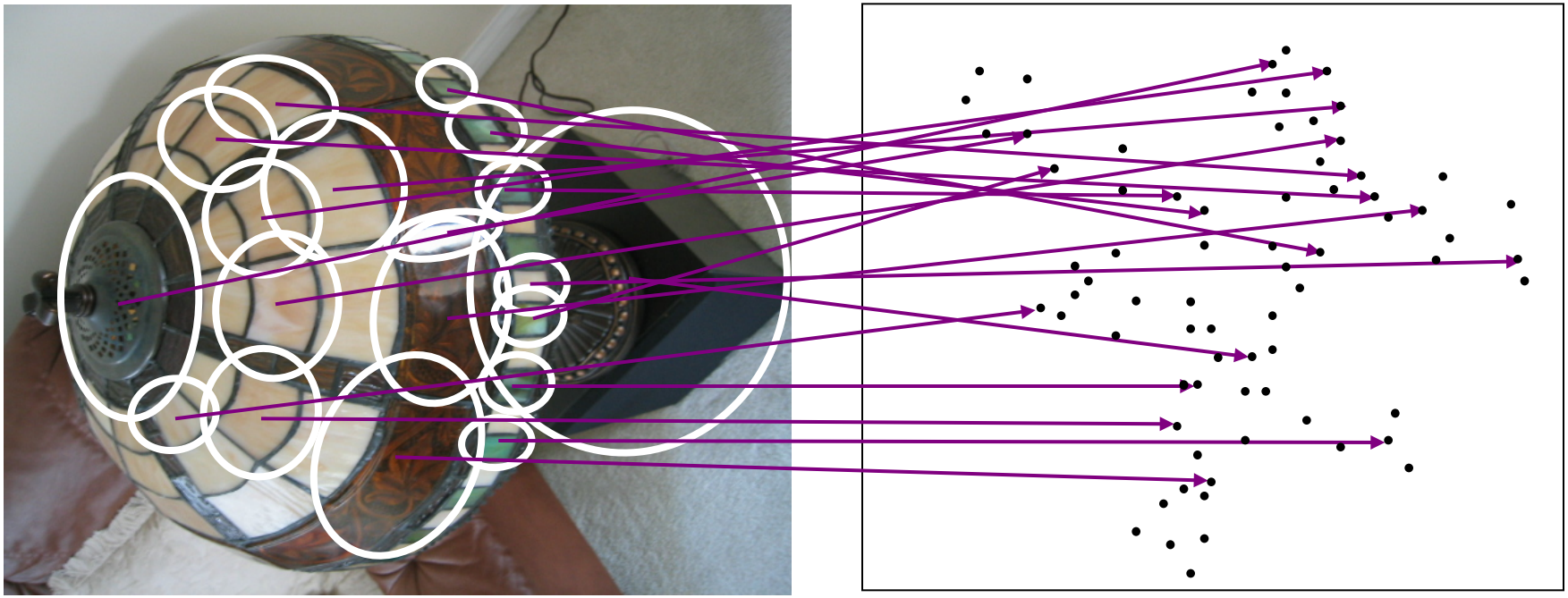


Пространство локальных признаков
(например, SIFT-дескрипторов)

Slide credit: D. Nister

«Визуальные слова»: основная идея

Извлечем локальные признаки из нескольких изображений…

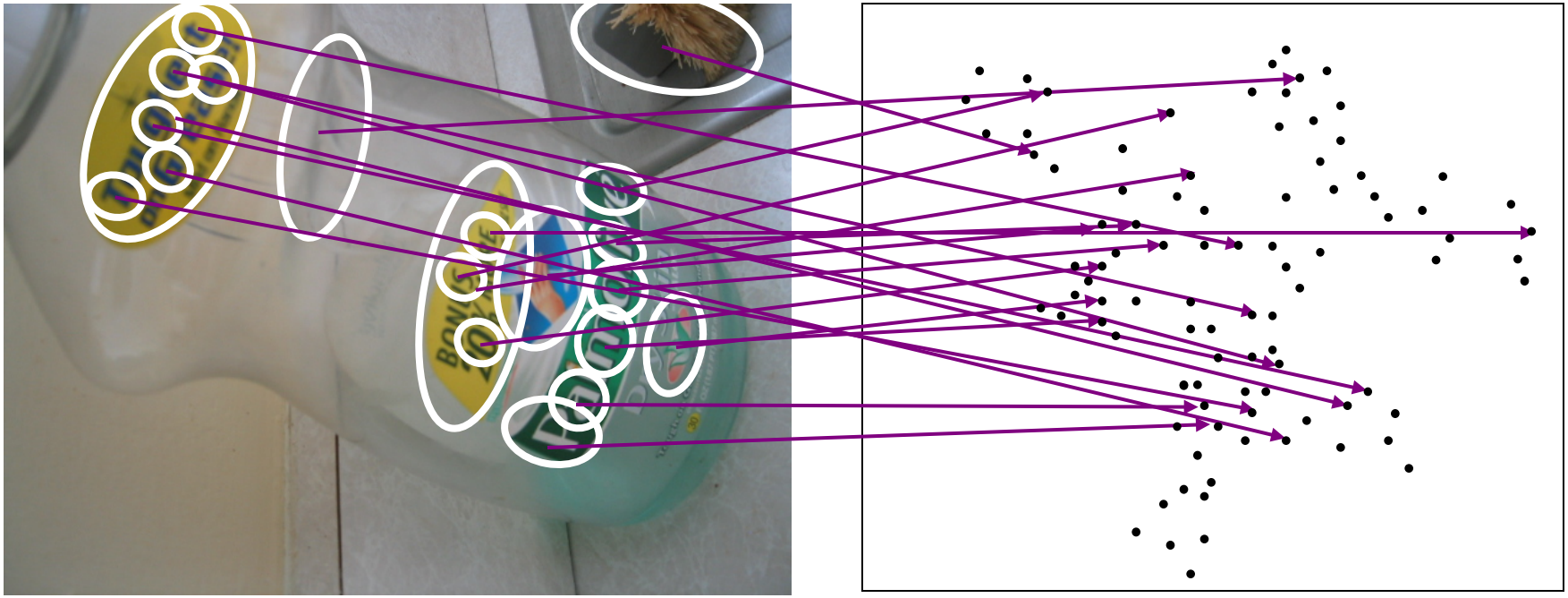


Пространство локальных признаков
(например, SIFT-дескрипторов)

Slide credit: D. Nister

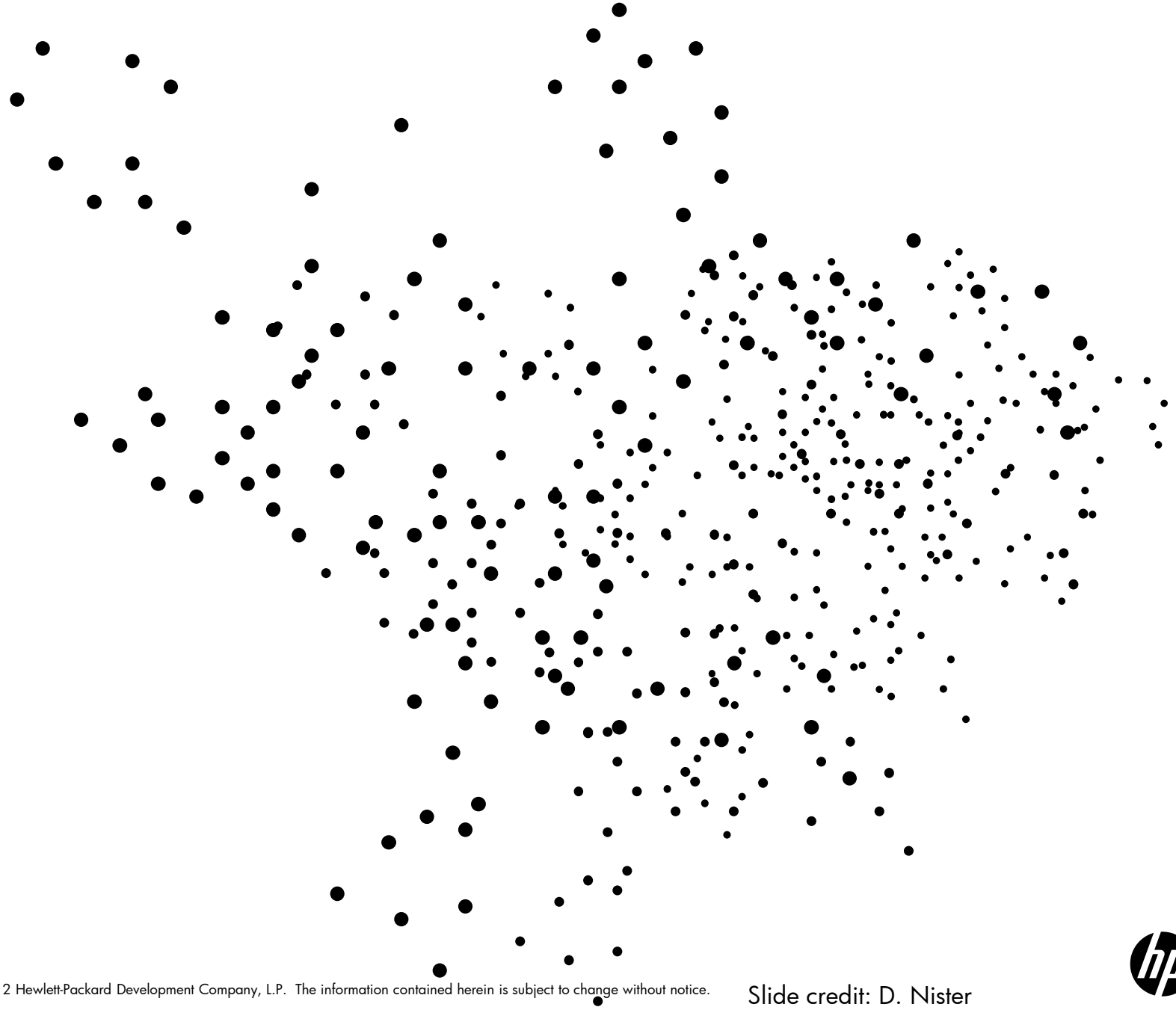
«Визуальные слова»: основная идея

Извлечем локальные признаки из нескольких изображений…

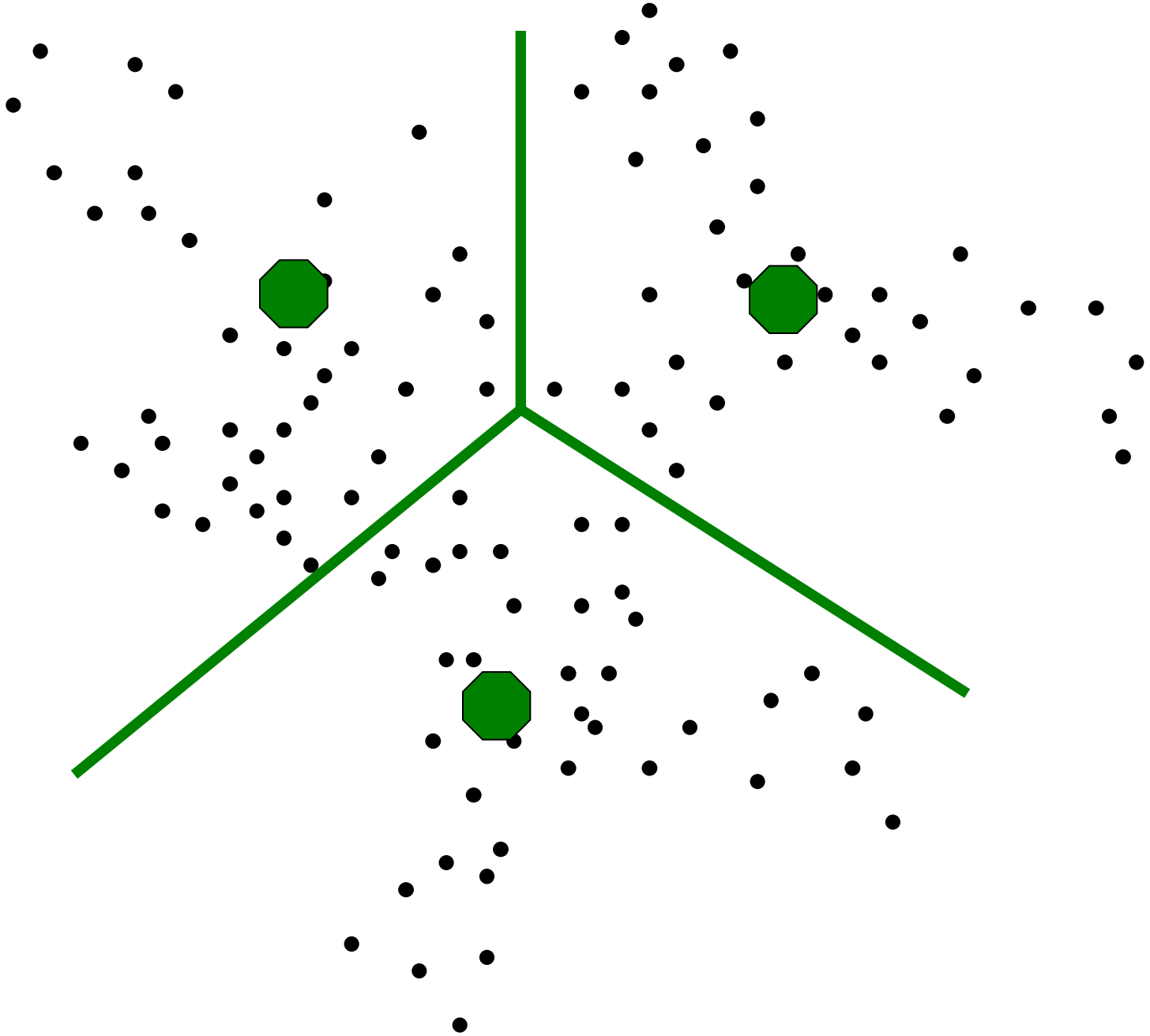


Пространство локальных признаков
(например, SIFT-дескрипторов)

Slide credit: D. Nister



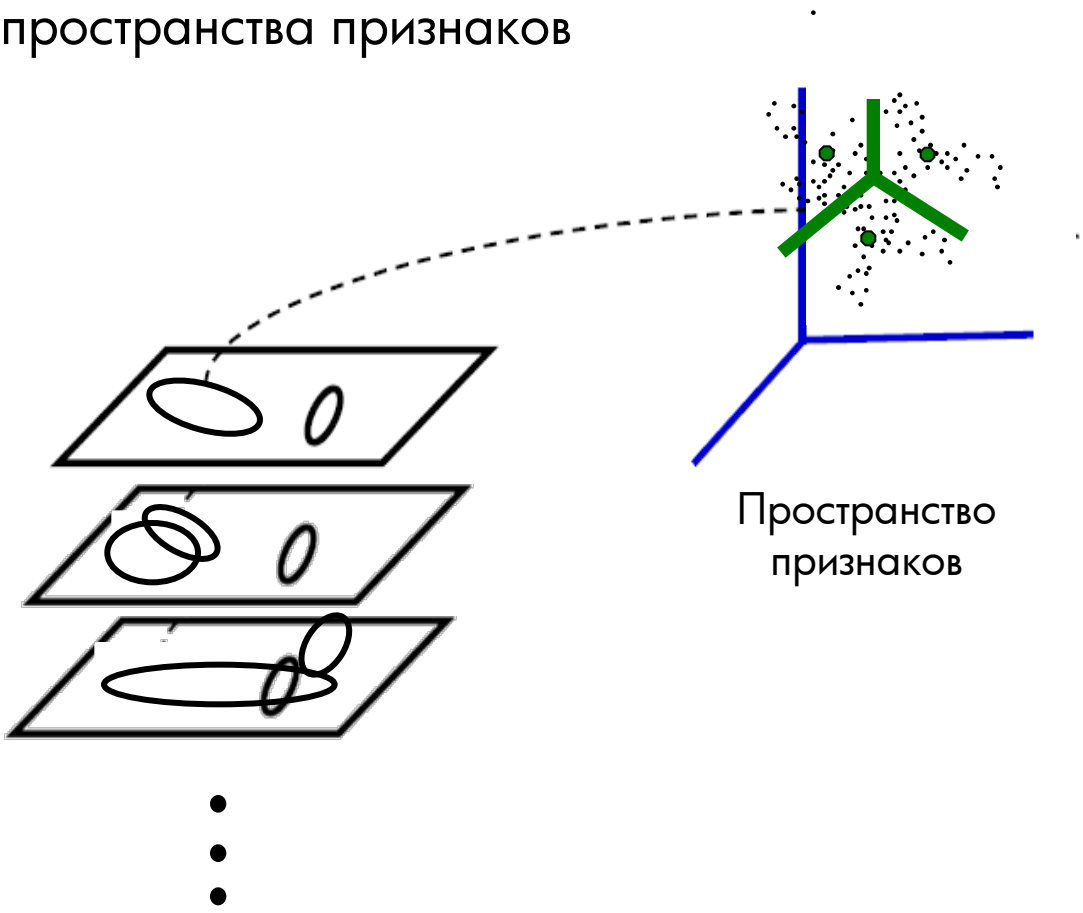
Slide credit: D. Nister



Slide credit: D. Nister

«Визуальные слова»: основная идея

Отобразим дескрипторы в токены/слова путем квантования пространства признаков



Пространство признаков

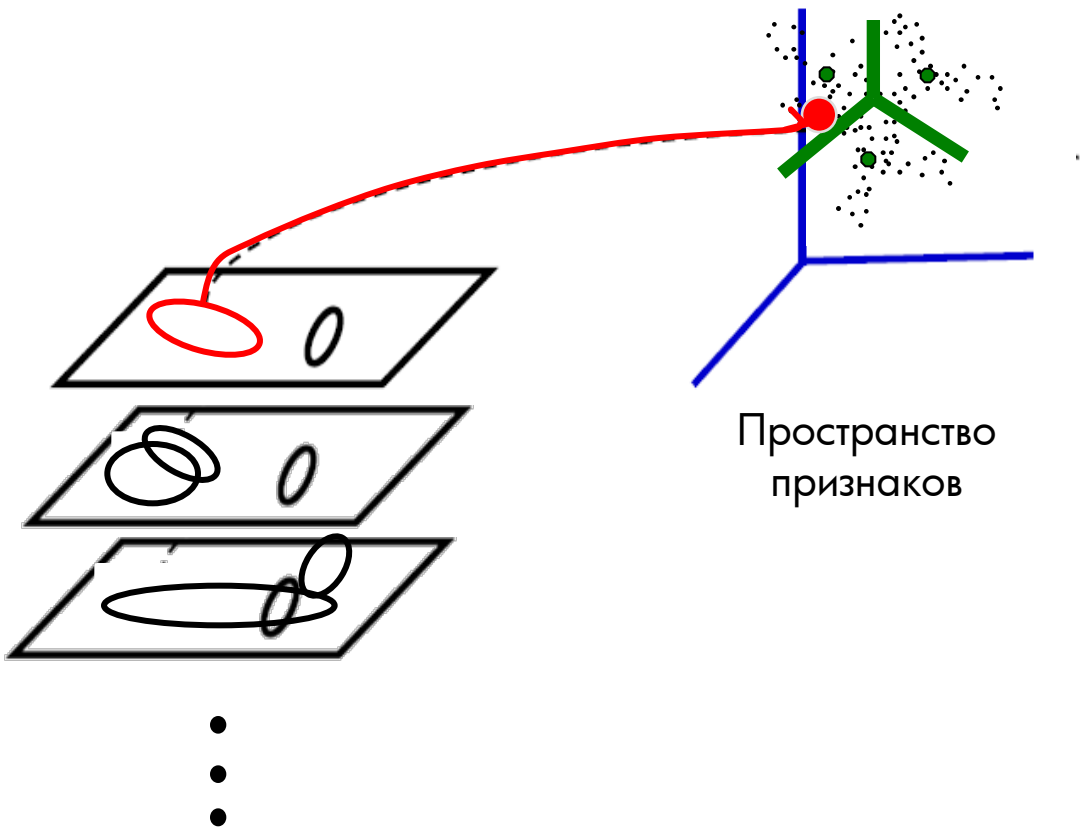
Квантование при помощи кластеризации пространства

Центры кластеров – «леммы» «визуальных слов»

Slide credit: K. Grauman, B. Leibe

«Визуальные слова»: основная идея

Отобразим дескрипторы в токены/слова путем квантования пространства признаков



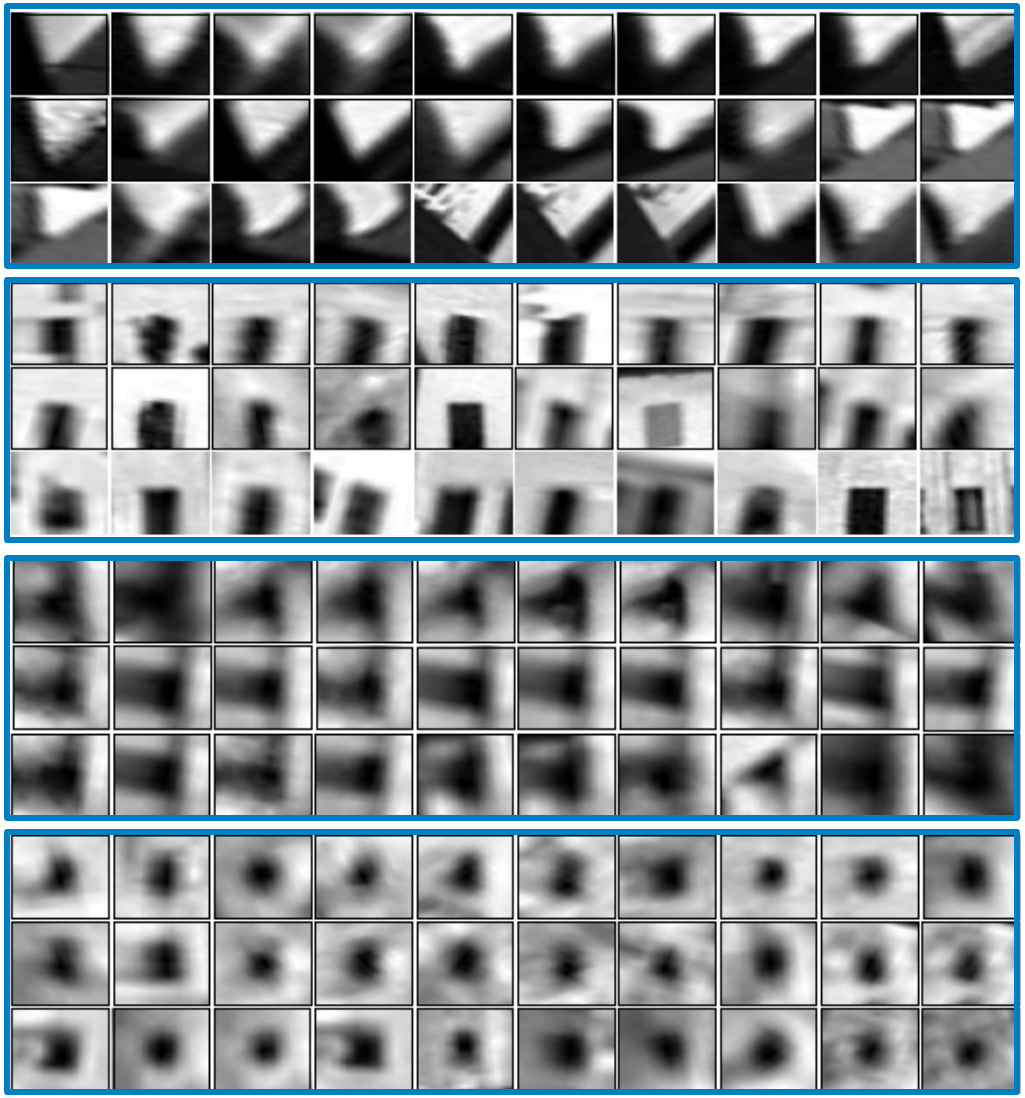
Пространство признаков

Каждый вектор признаков отображаем в «слово» – находим ближайший центр кластера

Описваем изображение через набор «визуальных слов»

Slide credit: K. Grauman, B. Leibe

Визуальные слова



Фрагментам из одной группы соответствует одно и то же визуальное слово

Fig. credit: Sivic & Zisserman

Обратный индекс изображений по визуальным словам

Изображение №5

Изображение №10

Номер слова	Список номеров изображений
1	5,10, ...
2	10,...
...	...

Slide credit: K. Grauman, B. Leibe

«Мешок визуальных слов»

Bag of visual words

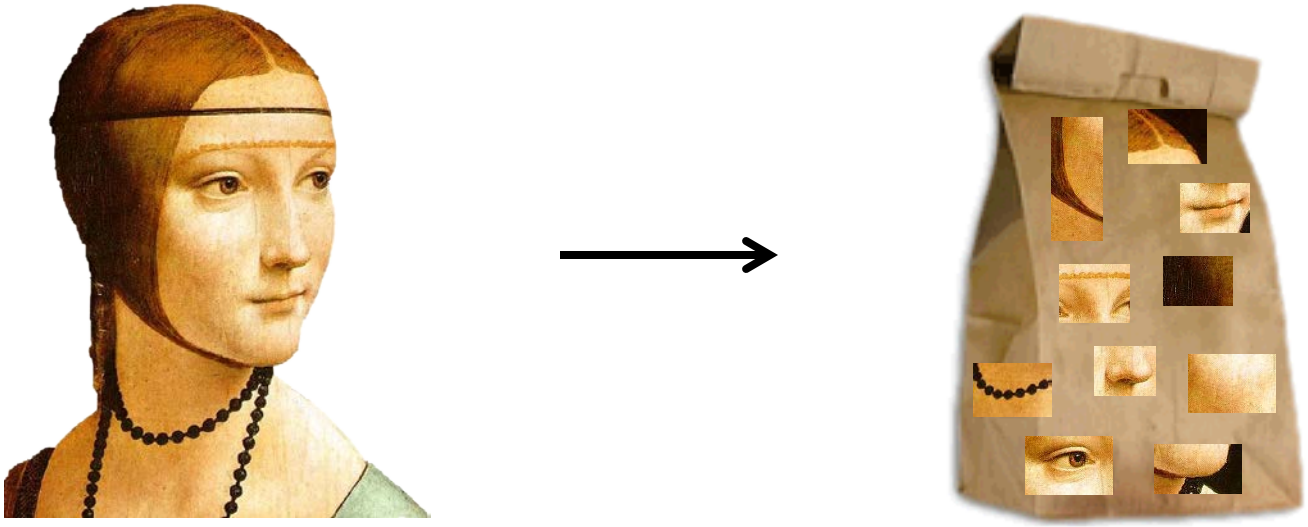
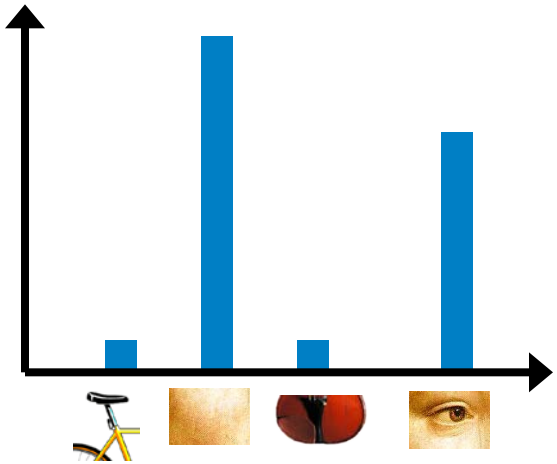


Fig. credit: Fei-Fei Li

«Мешок визуальных слов»

Bag of visual words



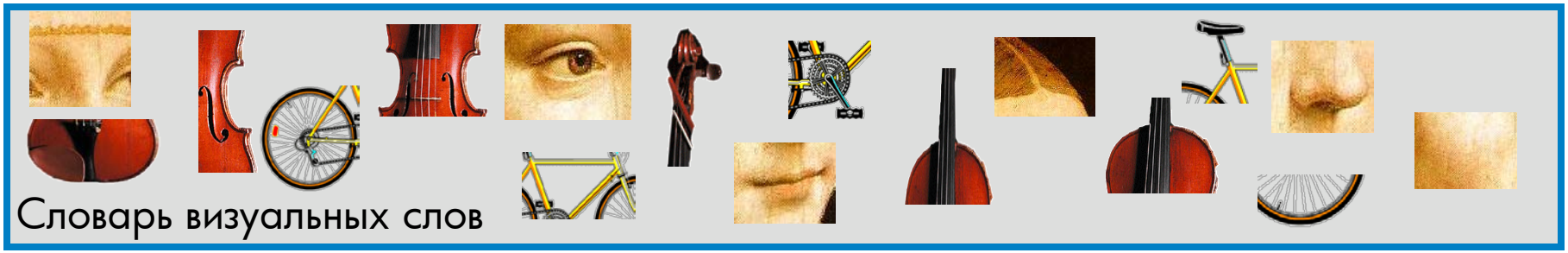


Fig. credit: Fei-Fei Li

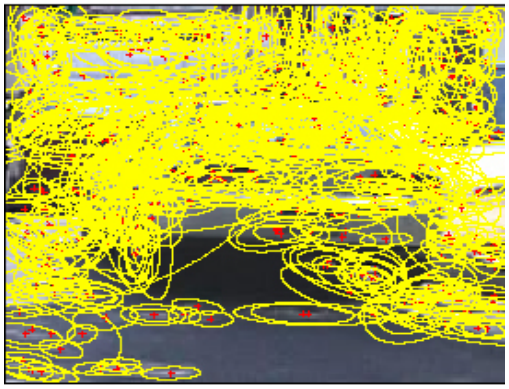
Как построить словарь?

Основные вопросы:

- Как выбрать фрагменты изображений?
- Какой алгоритм квантования/кластеризации использовать?
- С учителем / без учителя
- Как выбрать множество изображений, чтобы получить универсальный словарь?
- Размер словаря, число слов?

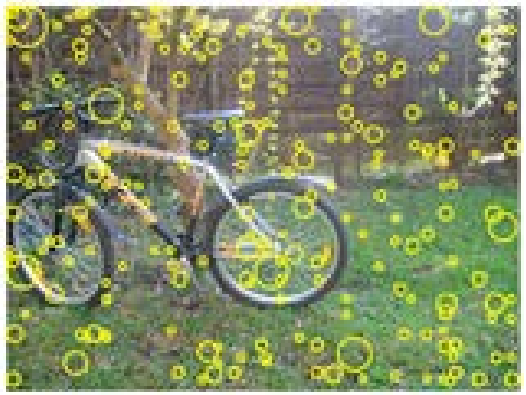
Slide credit: K. Grauman, B. Leibe

Выбор фрагментов

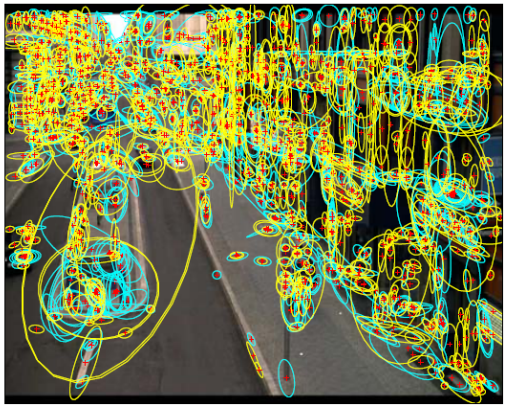


Sparse, at interest points

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.

[See Nowak, Jurie & Triggs, ECCV 2006]

Slide credit: K. Grauman, B. Leibe

Методы квантования/кластеризации

Традиционные решения:

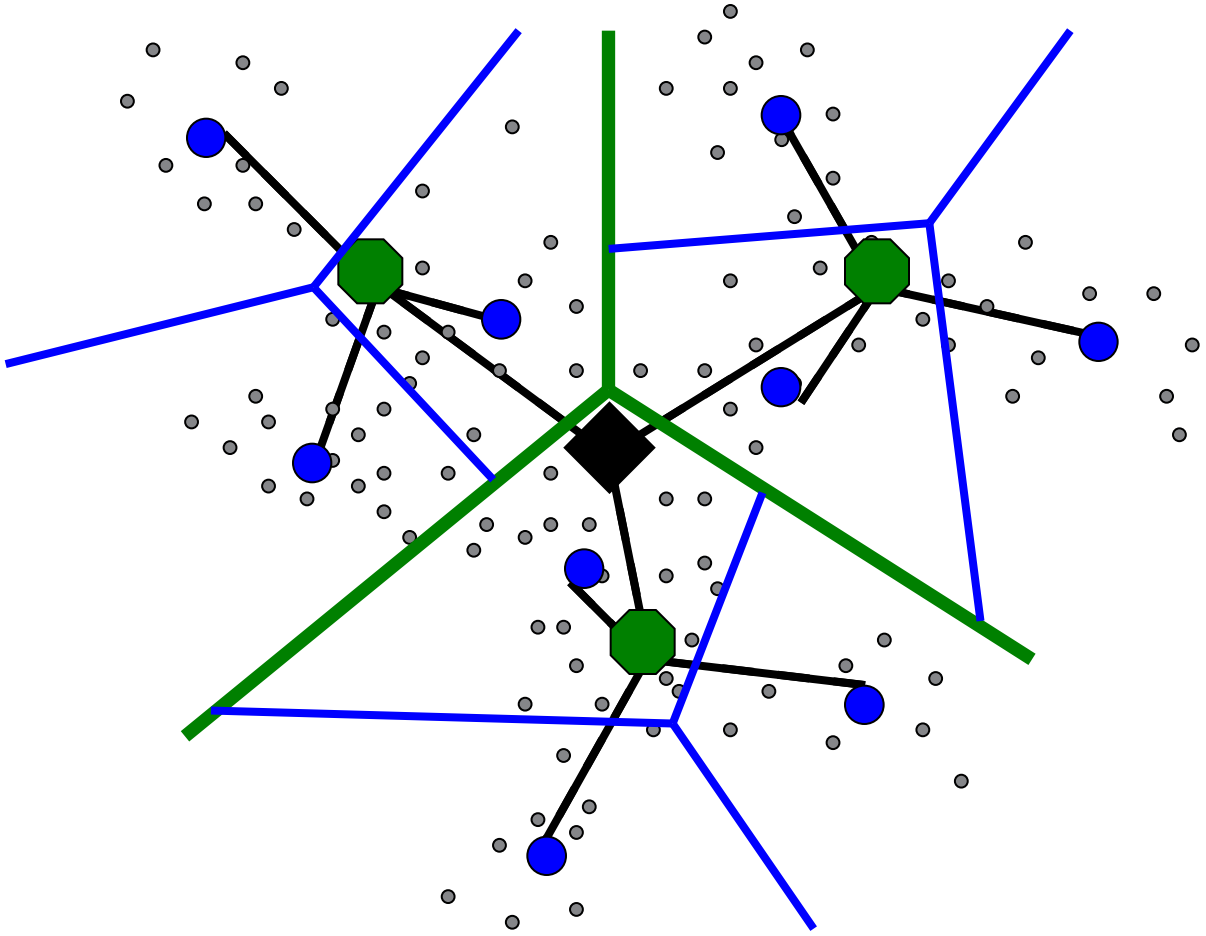
- k-means, agglomerative clustering, mean-shift,…

Использование иерархической кластеризации,
организация словаря в виде дерева:

- Vocabulary tree [Nister & Stewenius, CVPR 2006]

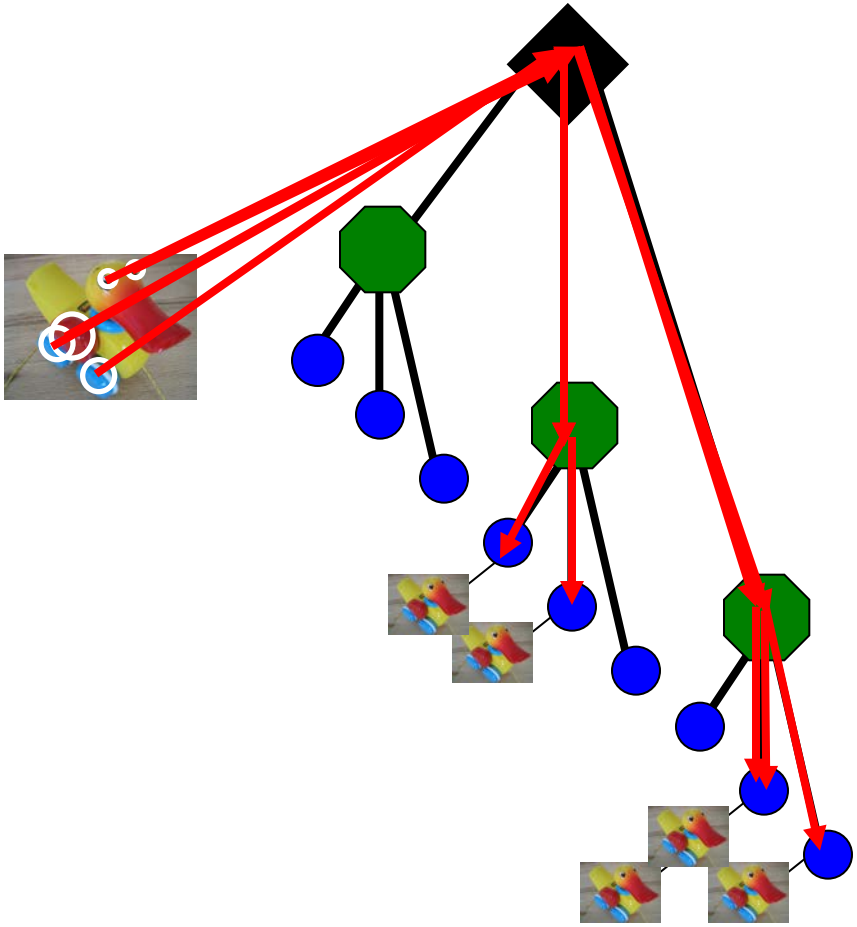
Более быстрая индексация и поиск при большем размере словаря!

Построение дерева



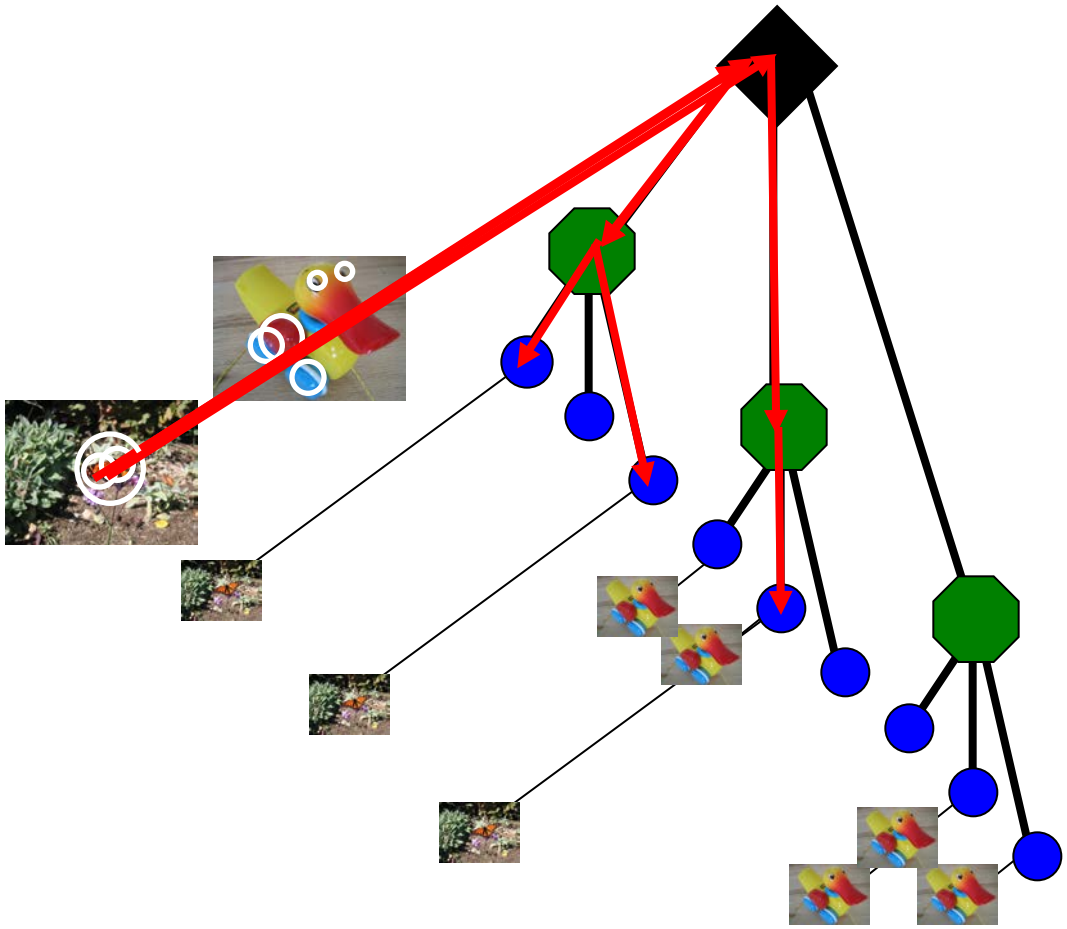
Slide credit: D. Nister

Индексирование



Slide credit: D. Nister

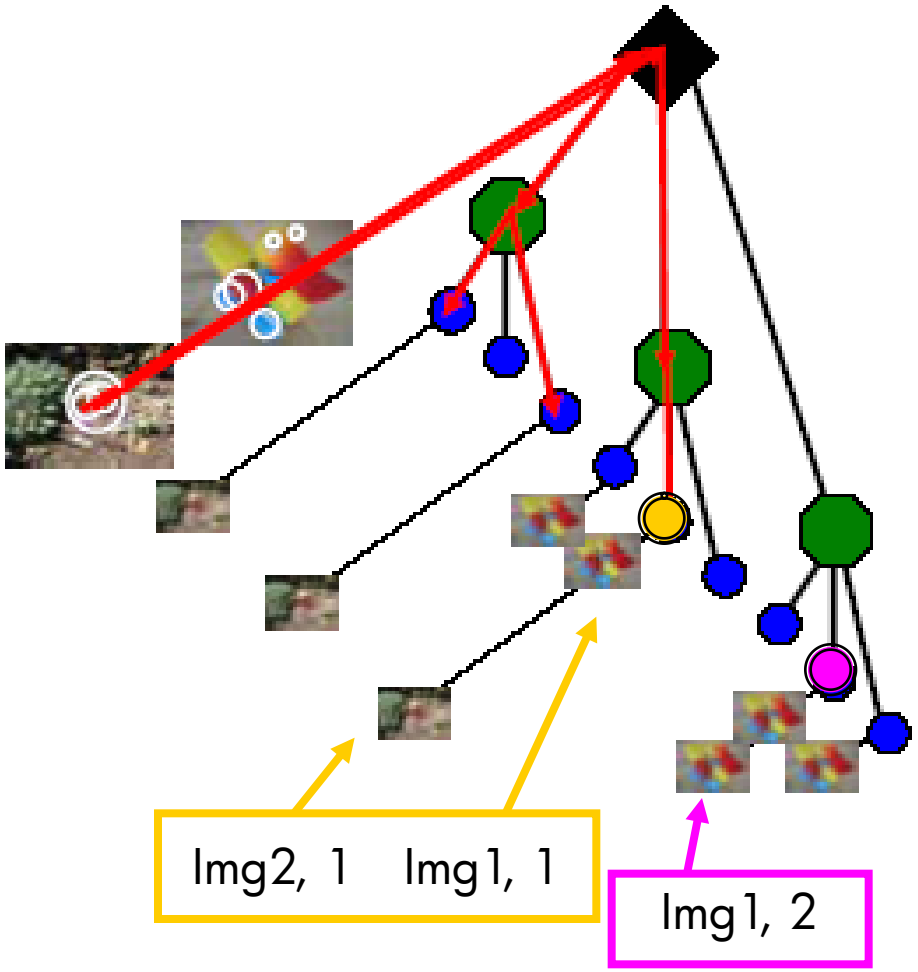
Индексирование



Slide credit: D. Nister

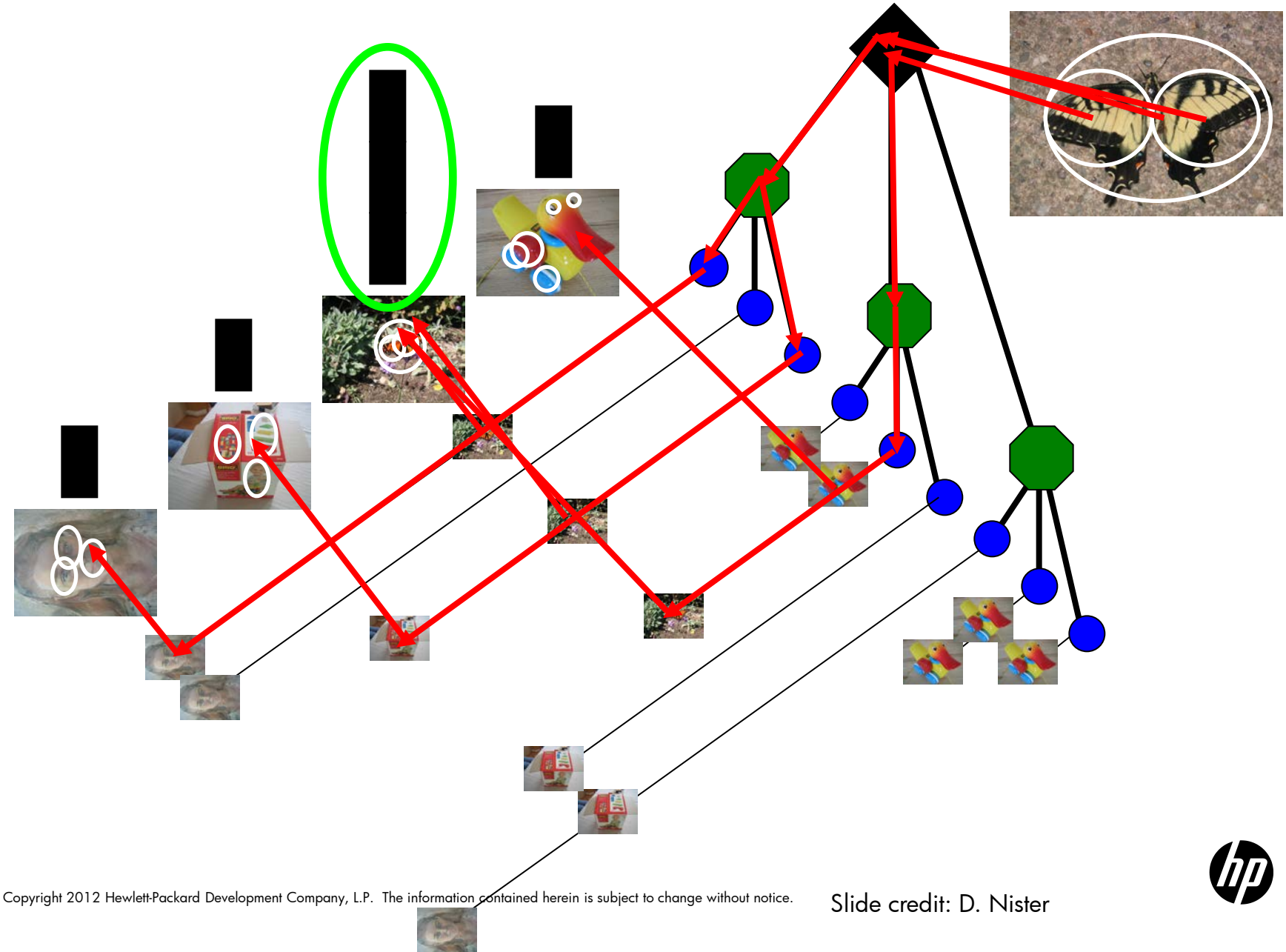
Индексирование

- Каждый узел дерева – обратный индекс
- Каждый узел дерева хранит идентификаторы изображений и частоту соответствующего слова в изображении (или tf-idf)



Slide credit: T. Tommasi

Поиск



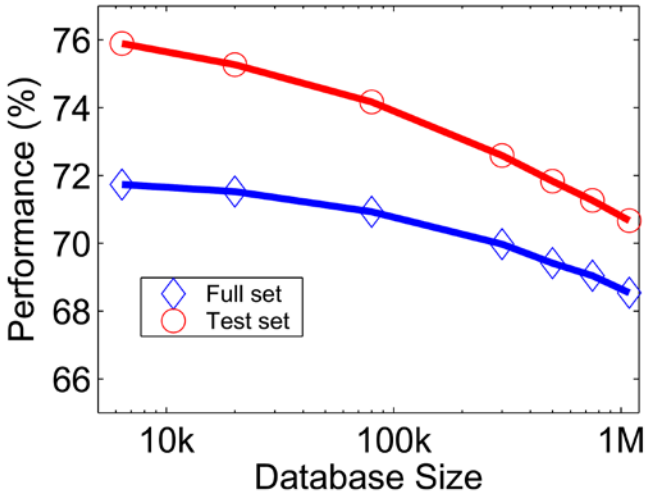
Slide credit: D. Nister

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



Slide credit: K. Grauman, B. Leibe

Locality Sensitive Hashing (LSH): motivation

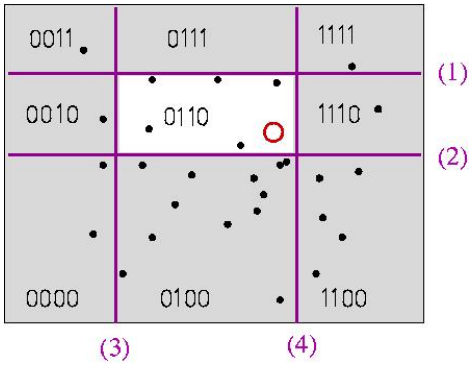
- Similarity Search over High-Dimensional Data
  - Image databases, document collections etc
- Curse of Dimensionality
  - All space partitioning techniques degrade to linear search for high dimensions
- Exact vs. Approximate Answer
  - Approximate might be good-enough and much-faster
  - Time-quality trade-off

Need *sub-linear* dependence on the data-size for high-dimensional data!

LSH: Key idea

Hash the data-point using several LSH functions so that probability of collision is higher for closer objects

- Idea: construct hash functions $g: R^d \rightarrow U$ such that for any points p,q:
  - If $||p-q|| \leq r$, then $Pr[g(p)=g(q)]$ is ~~"high"~~ "not-so-small"
  - If $||p-q|| > cr$, then $Pr[g(p)=g(q)]$ is "small"
- Then we can solve the problem by hashing



Indyk & Motwani, 1998

Fig. credit: P. Indyk, K. Grauman, B. Leibe

LSH [Indyk-Motwani'98]

- A family H of functions $h: R^d \rightarrow U$ is called (P_1,P_2,r,cr)-sensitive, if for any p,q:
  - if $||p-q|| < r$ then $Pr[\ h(p)=h(q)\ ] > P_1$
  - if $||p-q|| > cr$ then $Pr[\ h(p)=h(q)\ ] < P_2$

- Example: Hamming distance
  - LSH functions: $h(p)=p_i$, i.e., the i-th bit of p
  - Probabilities: $Pr[\ h(p)=h(q)\ ] = 1-D(p,q)/d$

p=10010010
q=11010110

Slide credit: P. Indyk

LSH (альтернативное определение)

Пусть *P* множество объектов (точек в пространстве выскокой размерности),
sim – функция подобия объектов: $sim : P \times P \to [0,1]$

Тогда схема LSH – это семейство хэш-функций *H*, имеющих распределение *D*, таких что при выборе функции h ∈ H согласно распределению D:

$$\Pr_{h \in H}[h(q) = h(p)] = sim(q, p) \quad \forall q, p \in P$$

LSH: индексирование

Вход:

- Set of N points $\{ p_1 , \ldots\ldots\ p_n \}$
- L – длина хэша
- M – число хэш-таблиц

Выход:

- Хэш-таблицы T_i , $i = 1 , 2, \ldots. M$

Алгоритм:

Foreach $i = 1 , 2, \ldots. M$

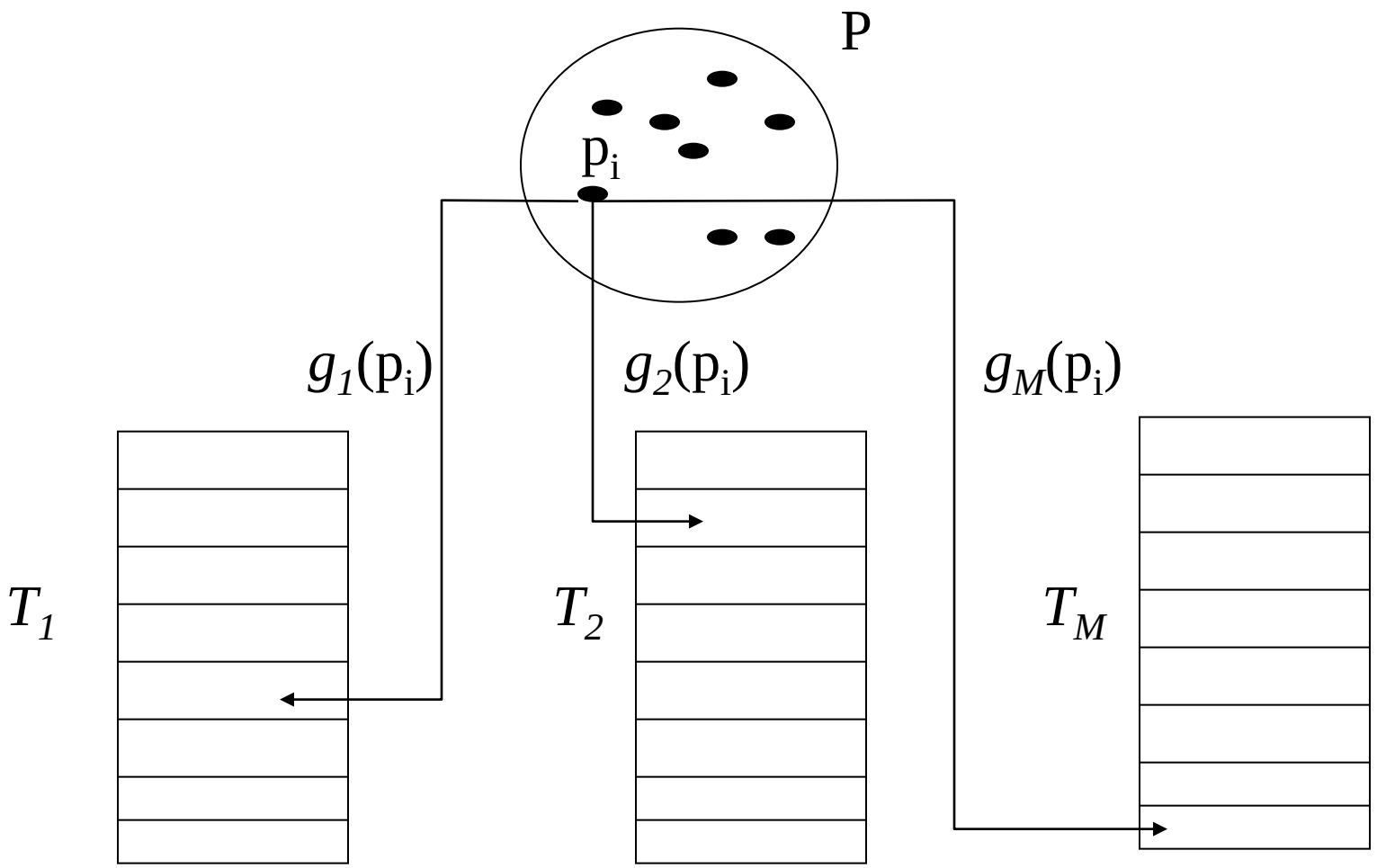
  Initialize T_i with a random hash function $g_i(p)=<h_1(p), h_2(p),\ldots h_k(p)>$

Foreach $i = 1 , 2, \ldots. M$

  Foreach $j = 1 , 2, \ldots. N$

    Store point p_j on bucket $g_i(p_j)$ of hash table T_i

LSH: индексирование



LSH: поиск билижайшего соседа

Вход:

- Запрос q

Выход:

- Приблизительный ближайший сосед p для q

Алгоритм:

Foreach $i = 1, 2, \ldots M$

Foreach { p: точа из ячейки $g_i(q)$ хэш-таблицы T_i }

IF $\| q - p \| \leq cr$ THEN return p

Использование LSH для поиска нечетких дубликатов

Шаг 1: Для каждого изображения коллекции вычислить набор локальных дескрипторов (например, набор SIFT-дескрипторов)

Шаг 2: Вычислить хэш-значение для каждого дескриптора

Шаг 3: Для каждой пары изображений вычислить колчиество совпадающих хэшей

Шаг 4: Дубликаты: пары изображений с числом совпадений > threshold

Оценка методов поиска

Что оценивать?

Быстродействие (efficiency)

- Важно в связи с большими объемами данных

Качество (effectiveness)

- Как померять?

Проблемы оценки

Необходимость наличия общей тестовой коллекции

- Corel Photo CDs
- Brodatz texture collection: http://www.ux.uis.no/~tranden/brodatz.html
- CoPhIR: http://cophir.isti.cnr.it/whatis.html
- Участие в ImageCLEF, TRECVID, imageEVAL, ROMIP

Как получить оценку релевантности?

- Использование коллекций с выделенными группами (Corel collection)
- Оценка пользователями
  - Метод общего котла (Pooling)
  - Различные виды оценки (relevant – not relevant, ranking, …)

Оценка качества

"You can see, that our results are better"

Численная оценка качестве

Точность (precision), полнота (recall)

$$precision = \frac{\text{No. relevant documents retrieved}}{\text{Total No. documents retrieved}},$$

$$recall = \frac{\text{No. relevant documents retrieved}}{\text{Total No. relevant documents in the collection}}$$

- Average recall/precision
- Recall at N, Precision at N
- F-measure

Численная оценка

- Error rate

$$\text{Error rate} = \frac{\text{No. non-relevant images retrieved}}{\text{Total No. images retrieved}}$$

- Retrieval efficiency

$$\text{Retrieval efficiency} = \begin{cases} \dfrac{\text{No. relevant images retrieved}}{\text{Total No. images retrieved}} \\ \quad \text{if No. retrieved} > \text{No. relevant}, \\ \dfrac{\text{No. relevant images retrieved}}{\text{Total No. relevant images}} \quad \text{otherwise.} \end{cases}$$

Картиночные дорожки на РОМИП СЕМИНАР

http://romip.ru

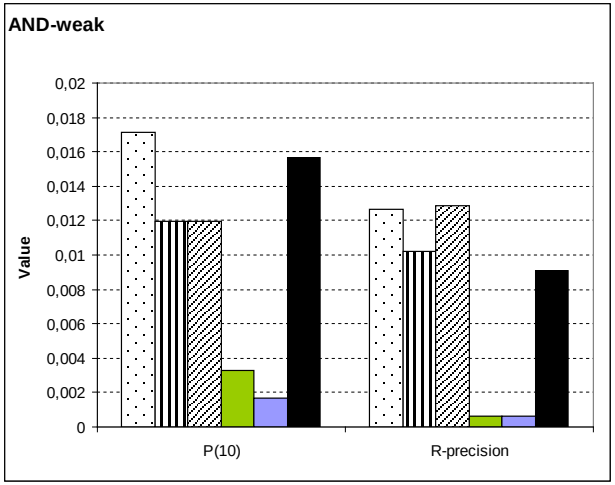
- Поиск по визуальному подобию (с 2008)
  - Коллекция Flickr (20000 изображений)
  - 750 tasks labeled
- Поиск нечетких дубликатов (с 2008)
  - Коллекция: набор кадров из видео потока (37800 изображений)
  - ~1500 clusters
- Построение текстовых меток (с 2010)
  - Коллекция Flickr (20000 изображений)
  - ~ 1000 labeled images

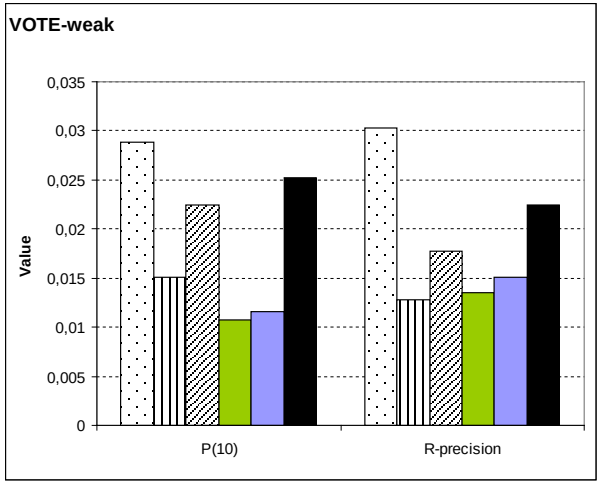
Поиск по визуальному подобию: наблюдения

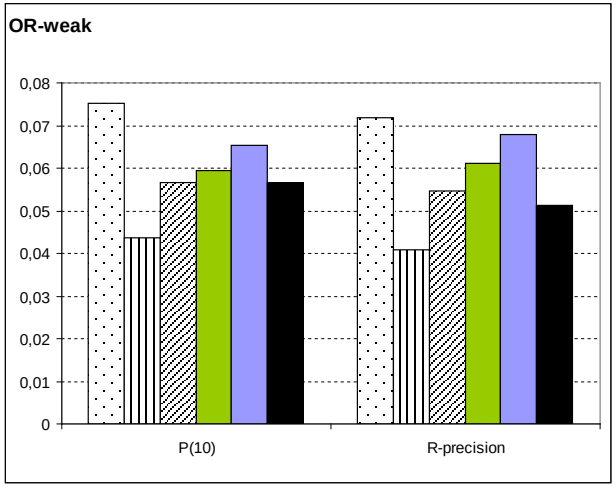
Слабая согласованность оценок асессоров

- Число слаборелевантных 377 (AND), 1086 (VOTE) и 3052 (OR)

Низкое качество результатов у всех участников







Заключение

Поиск изображений: необходимость индексировать данные высокой размерности

- Использование классических методов многомерного индексирования (k-мерные деревья, метрические деревья)
- Обратный индекс
- «Мешок визуальных слов», vocabulary tree
- Locality Sensitive Hashing

Оценка методов поиска

- Единые тестовые коллекции, единые метрики оценки
- Качество поиска пока все еще далеко от идеала
- РОМИП – требуются добровольцы!

