Интернет-зрение

Антон Конушин

Many slides adopted from Cordelia Schmid, Li Fei-Fei, Rob Fergus, Antonio Torralba

Сколько изображений?

- Всего около ~100000 разных категорий объектов
  - И миллионы «мелких» категорий, вроде пород собак, видов насекомых и т.д.
- Сколько нам нужно изображений, чтобы научиться их всех распознавать?
- Сколько нам доступно изображений?
- Что мы про них будем знать?

Сколько всего картинок?

Число картинок на диске: **10^4**

Число картинок, виденных за 10 лет: **10^8**

(3 images/second * 60 * 60 * 16 * 365 * 10 = 630720000)

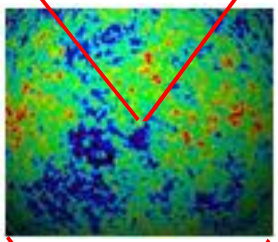
Число картинок,
виденных всем человечеством: **10^{20}**

106,456,367,669 humans[1] * 60 years * 3 images/second * 60 * 60 * 16 * 365 =

1 from http://www.prb.org/Articles/2002/HowManyPeopleHaveEverLivedonEarth.aspx

Число картинок во вселенной: **10^{243}**

10^{81} atoms * 10^{81} * 10^{81} =

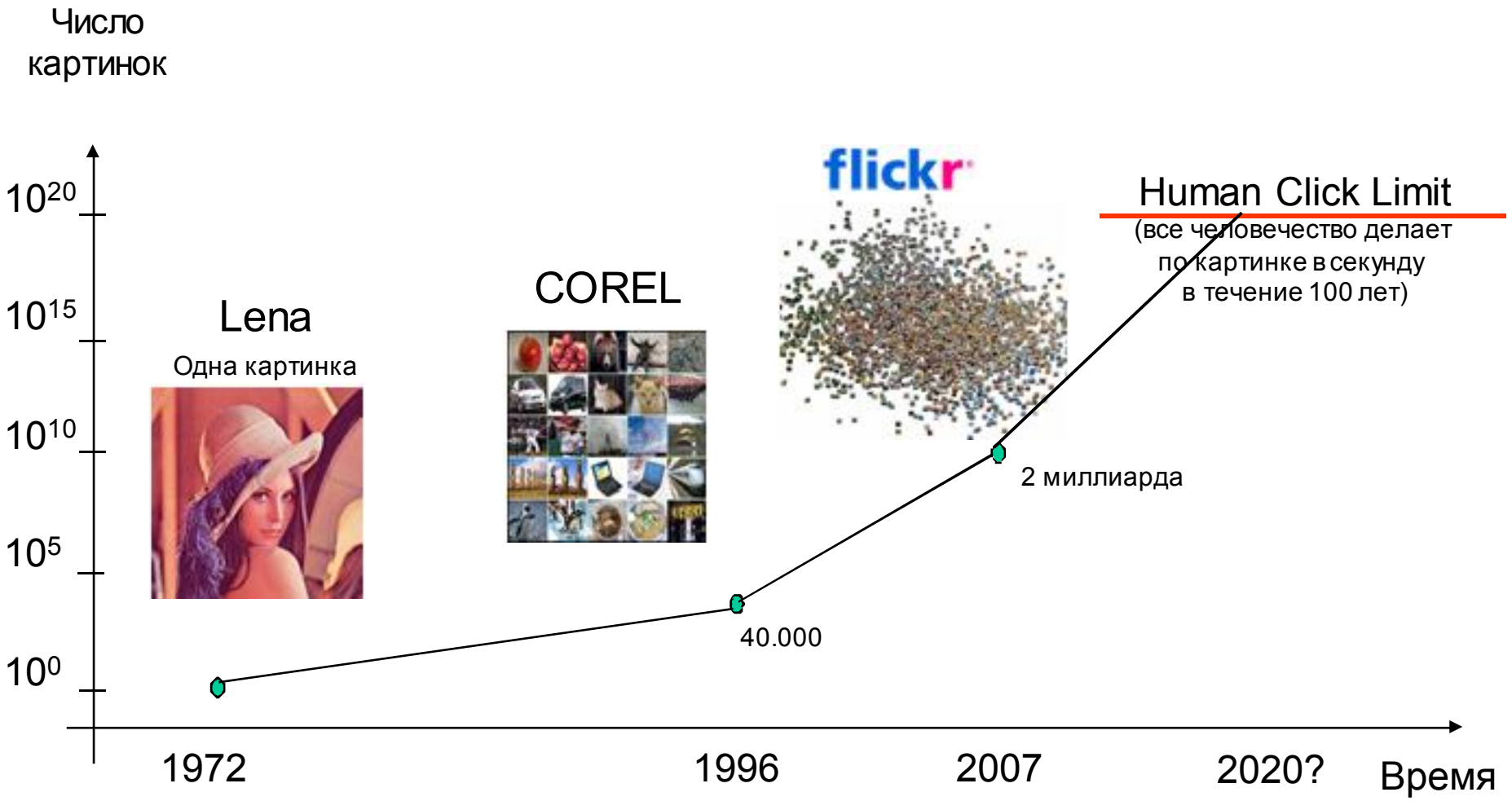


Число всех картинок 32x32 : 10^{7373}

$256^{32*32*3} \sim 10^{7373}$

Slide by Antonio Torralba

Доступные изображения



Обычно, про само изображение мы ничего не знаем, кроме параметров в EXIF из JPEG

Slide by Antonio Torralba

Что будем делать?

- Благодаря интернету, мы получили доступ к огромным коллекциям изображений
- Что мы с ними можем сделать, как они нам помогут?
- Первым делом, нужно научиться искать картинки в больших коллекциях

Поиски изображений по содержанию

- Content-based image retrieval
- Задача похожа на выделение объектов и классификацию изображений, но фокусируется в основном на масштабировании на большие коллекции с использованием приближенных методов

R. Datta, D. Joshi, J. Li, and J. Z.Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys, 2008.*

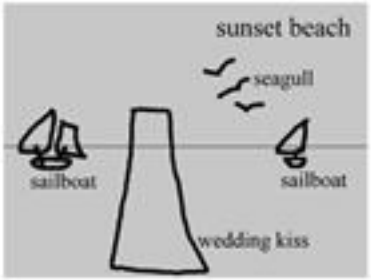
Что запрашивает пользователь?

- Запрос в виде атрибутов/текстового описания изображения
  - «Московская фотобиеннале-2012, фото» - аннотация (атрибуты) изображения
  - Нужна категоризация/аннотация изображений
- Запрос в виде изображения-примера («найди то же самое»)

- Запрос в виде некоторой характеристики содержимого

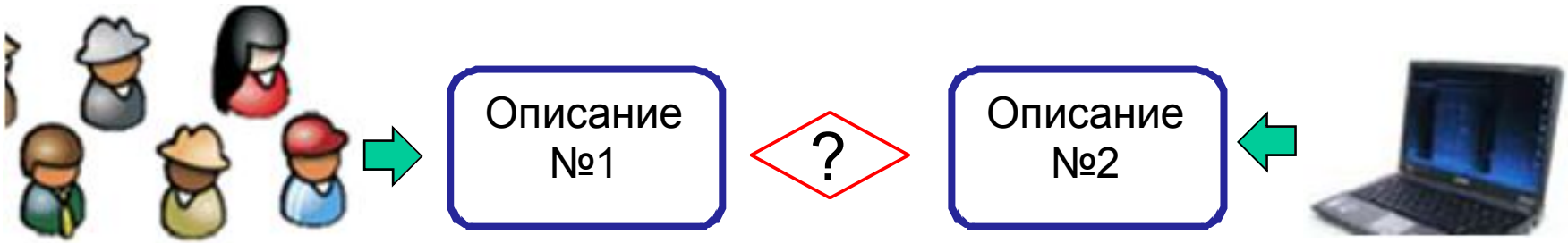
- Гистограмма цветов

- Запрос в виде рисунка-наброска



Semantic Gap

- Запрос в виде изображения-примера («найди то же самое», «найди похожее изображение»)
- Что имел пользователь в виду?
- Что значит «похожее изображение»?
- «Семантический разрыв» – несовпадение информации, которую можно извлечь из визуальных данных, и интерпретацией тех же самых данных со стороны пользователя



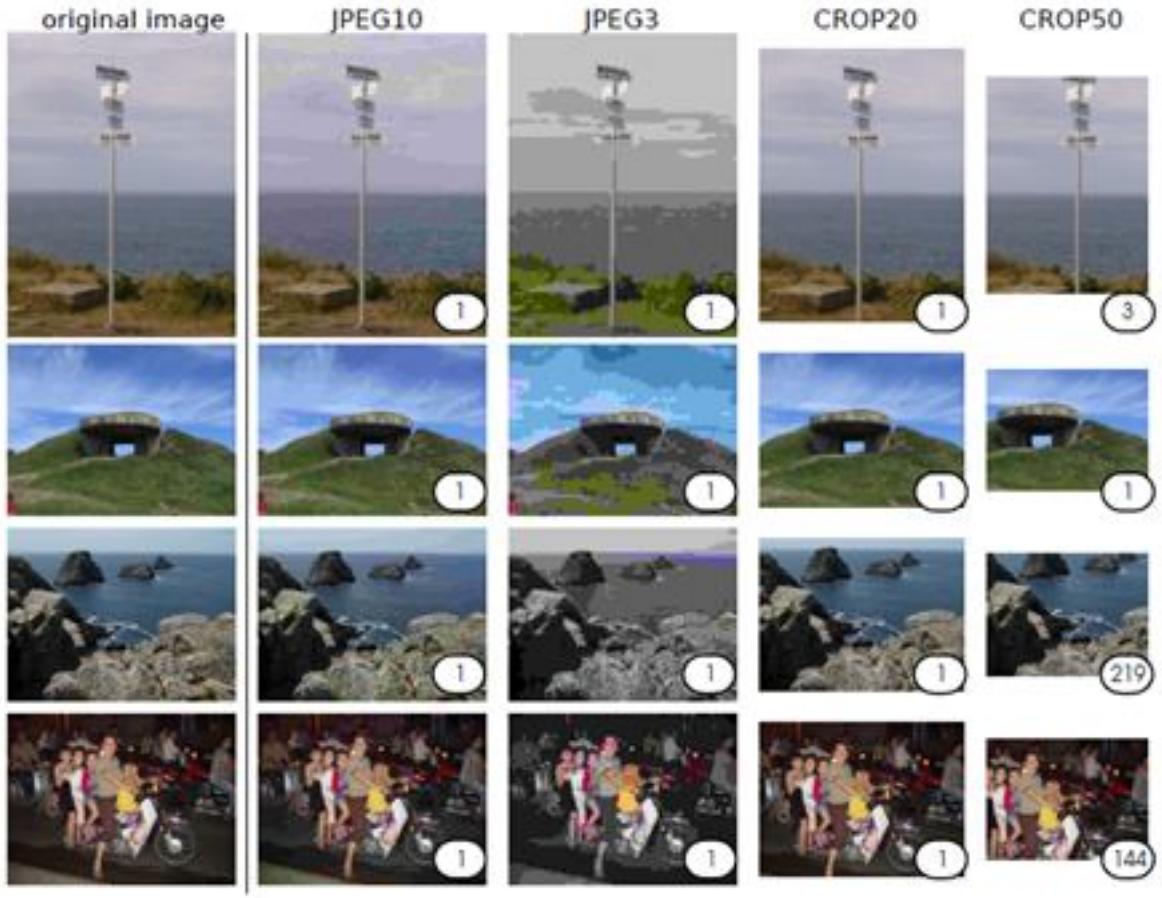
Что значит похожее?

1. Похожее по каким-то характеристикам, например, по цвету

Запрос

Что значит похожее?

2. Полудубликаты (Near-duplicates) – слегка измененная версия изображения (ракурс, цвета)



Что значит похожее?

3. Тот же самый объект или сцена («Object retrieval»)
   - Большие вариации ракурсов, фона, и т.д., чем при поиске полудубликатов

Что значит похожее?

4. Похожие визуально по геометрии сцены с учетом ракурса (могут быть разные по назначению)

Кухня

Приемная

Бар

Автобус

Самолет

Зал

Что значит похожее?

5. «Category-level classification» - изображения одного класса сцен или объектов

Пример – банкетный зал.

Например, 256 классов из базы Caltech 256

Анализ постановок задач

- Какие постановки задач внутренне схожи, а какие существенно отличаются?

Визуальное подобие

Семантическое подобие

- В последнем случае нам нужно выполнить автоматическую аннотацию изображения, определить класс сцены, присутствующие объекты, их характеристики
- Затем мы будем сравнивать метки запроса с метками изображений в базе
- Мы сегодня рассмотрим визуальное подобие для задачи «полудубликаты»

Общая схема поиска изображений

Построение индекса

Коллекция изображений

Признаки изображения → Подпись (Signature) → Индекс коллекции

Дескриптор на основе извлеченных признаков

Тестирование

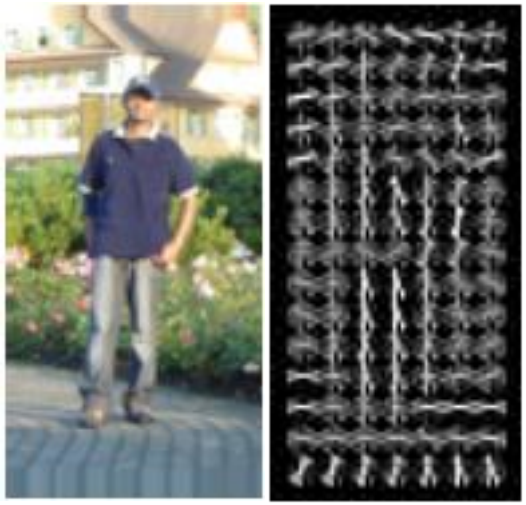
Тестовое изображение

Признаки изображения → Подпись (Signature) → Поиск по индексу → Список похожих

Ищем ближайших соседей по выбранной метрике

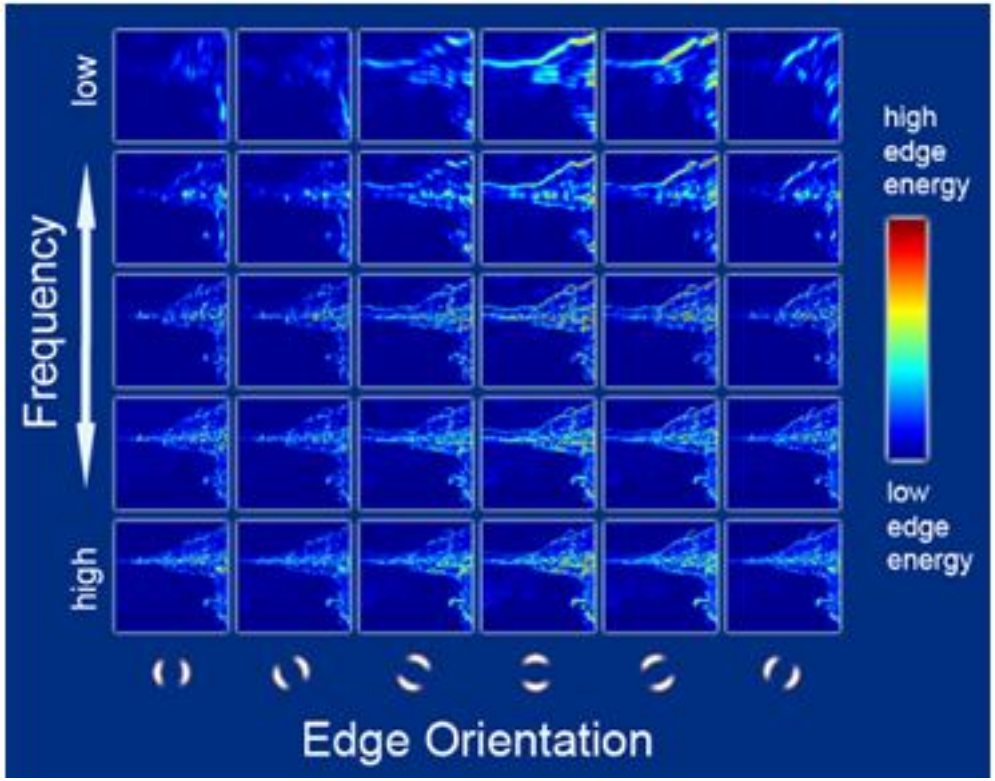
Гистограммы градиентов

- Насколько HOG подходит для сравнения двух произвольных изображений по визуальному сходству?
- Как учитывается масштаб/размер объекта?



HOG (2005)

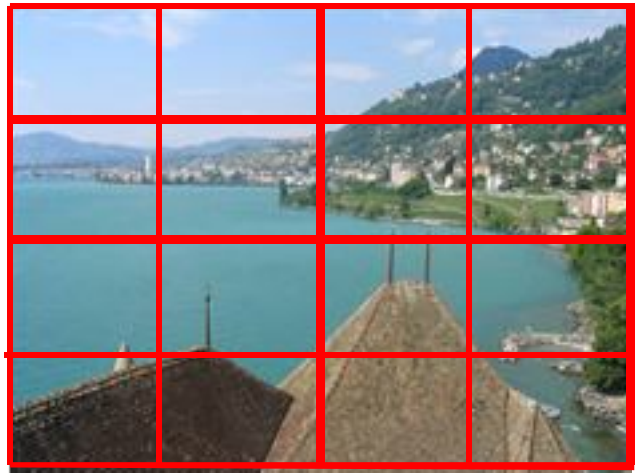
Дескриптор изображения GIST

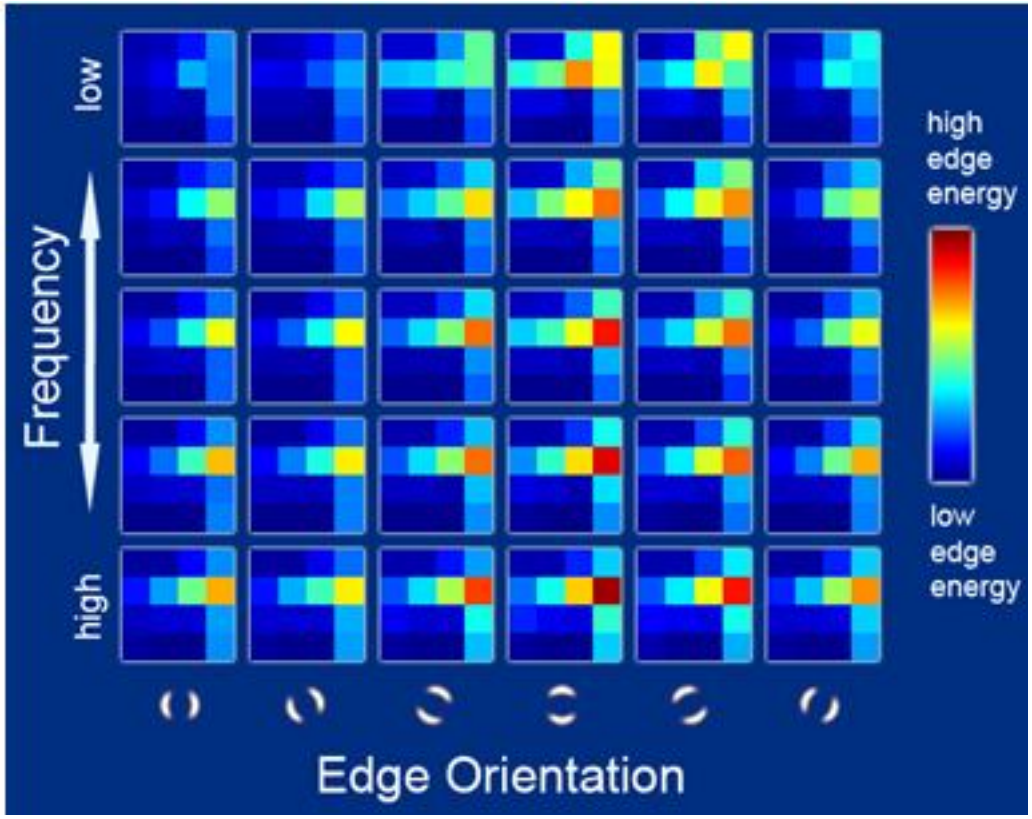


- Воспользуемся методом, похожим на HOG.
- Посчитаем отклики детекторов краёв на 5 разных масштабах и 6 ориентациях края
- Получим 33 «канала» - RGB-цвет и 30 откликов фильтров края

TORRALBA, A., MURPHY, K. P., FREEMAN, W. T., AND RUBIN. Context-based vision system for place and object recognition. In ICCV 2003

Дескриптор изображения GIST





- Разобьём изображение сеткой 4x4 на 16 ячеек
- В каждой ячейке усредним значения всех каналов
- Получим дескриптор GIST
- Можем добавить средний цвет и дисперсию для каждой ячейки

Применение GIST

Запрос

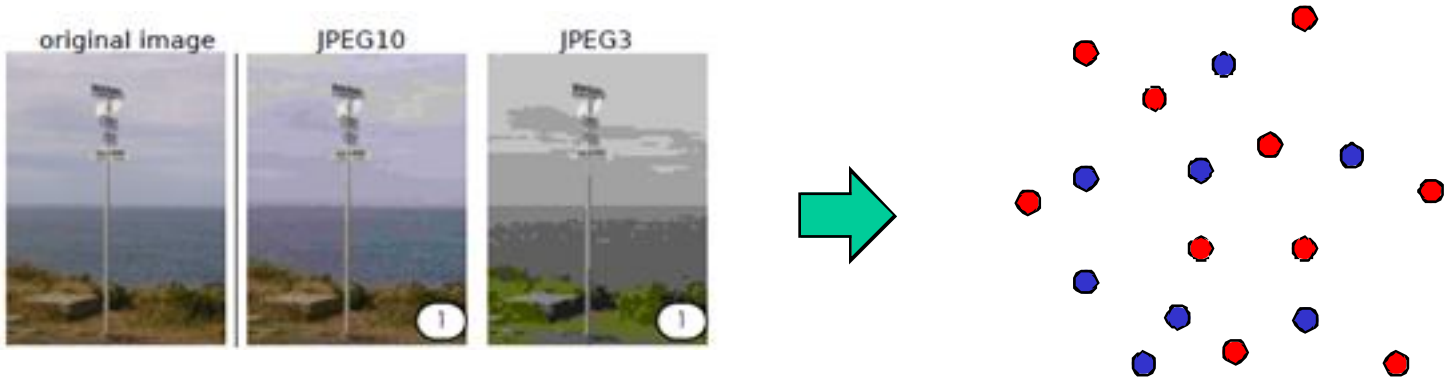
Похожие по GIST + цвету

GIST хорошо себя показал для оценки визуального подобия изображений в задаче поиска «полудубликатов»

Затраты по памяти для GIST

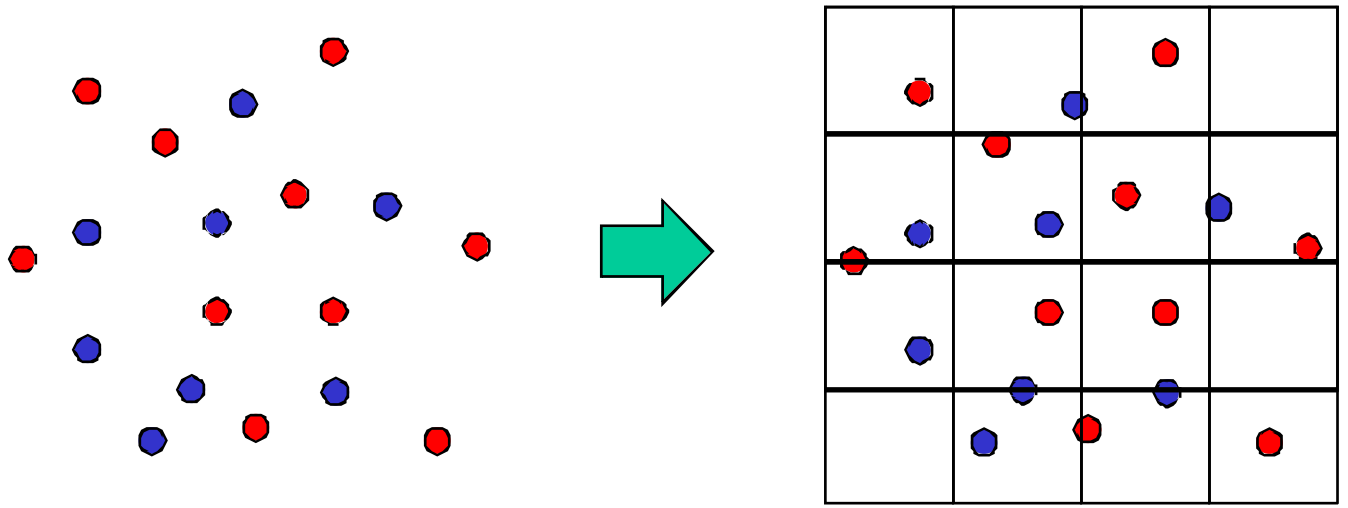
- Решетка 4x4 * 8 ориентаций * 4 масштаба = 512 параметров
- При 4 байтах на параметр – 2048 байт (16384 бита)
- 2Гб на 1M изображений
- 200Гб на 100M изображений
- Уже довольно тяжело хранить большие коллекции в RAM

Скорость работы



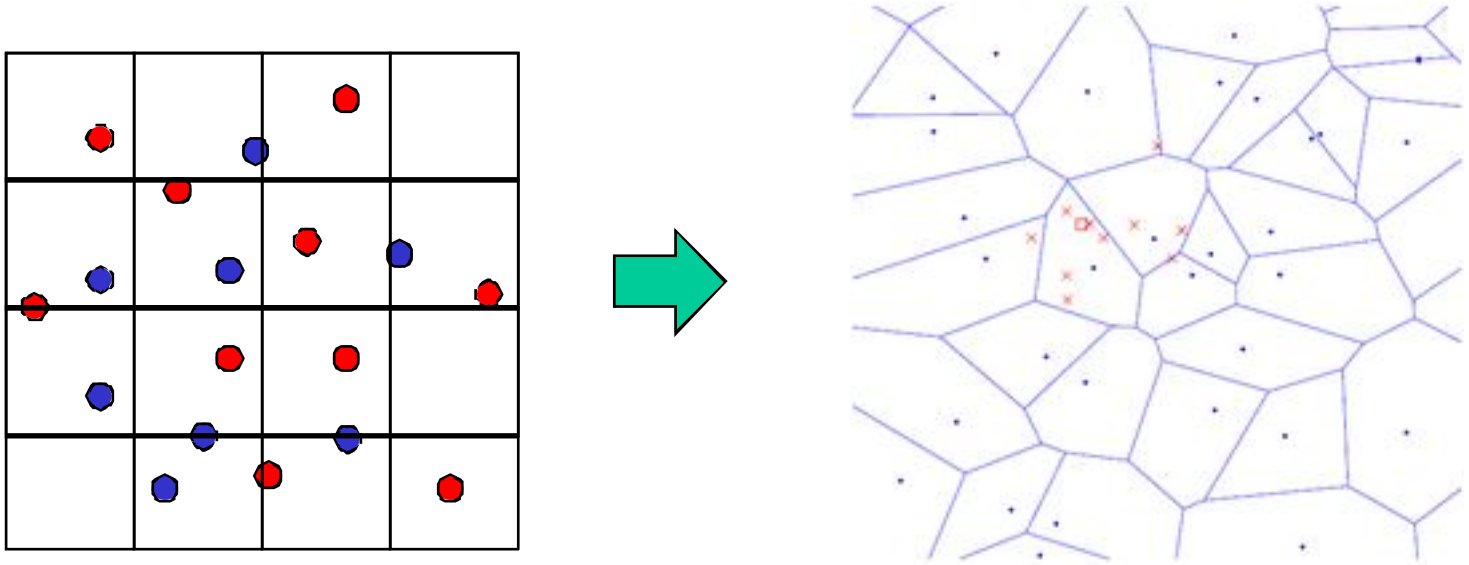
- Задача CBIR сводится к поиску ближайших соседей по дескриптору во всей коллекции (Nearest Neighbor)
- Простейшее и точное решение – линейный поиск
- Размер коллекции – миллионы и миллиарды векторов
- Нужны быстрые методы, пусть и с потерей точности

Квантование векторов



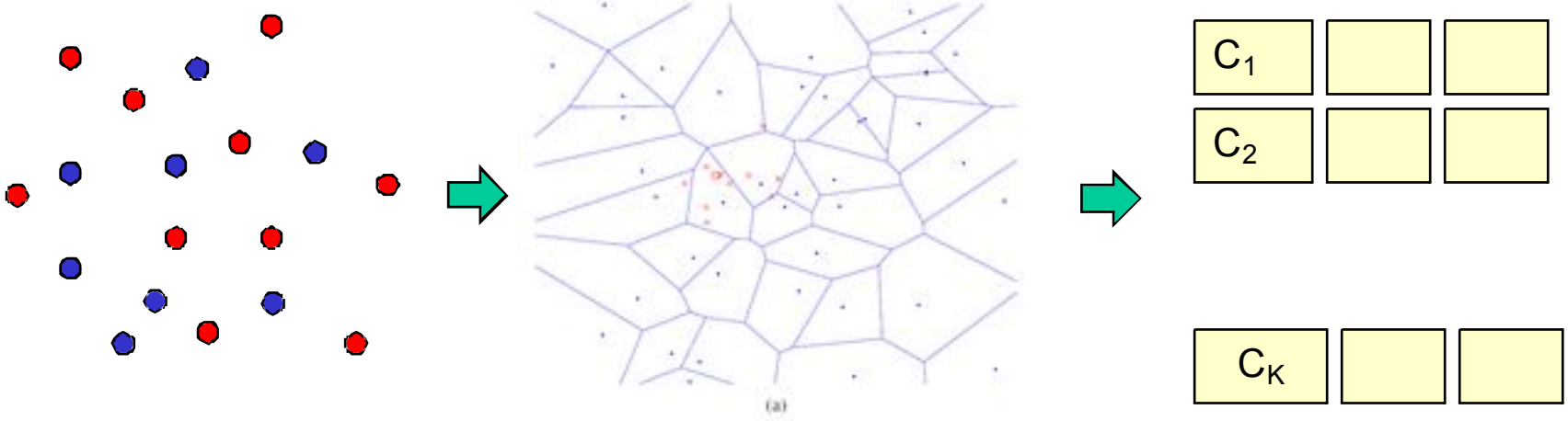
- «Vector quantization» – способ ускорения поиска за счёт уменьшения точности
  - Отображение x на q(x), где q(x) in $\{c_i\}$, c_i – центроиды, i от 1 до K
- Индекс q(x) можно записать в виде битового кода длиной $\log_2(K)$
- Фактически, сжатие исходного вектора до $\log_2(K)$
- Простейший способ квантования – разбиение пространства дескрипторов на ячейки вдоль осей координат

Адаптивное квантование



- Регулярное разбиение не учитывает структуру данных, их распределение в пространстве дескрипторов
- Адаптивно разбить пространство можем с помощью кластеризации K-средними

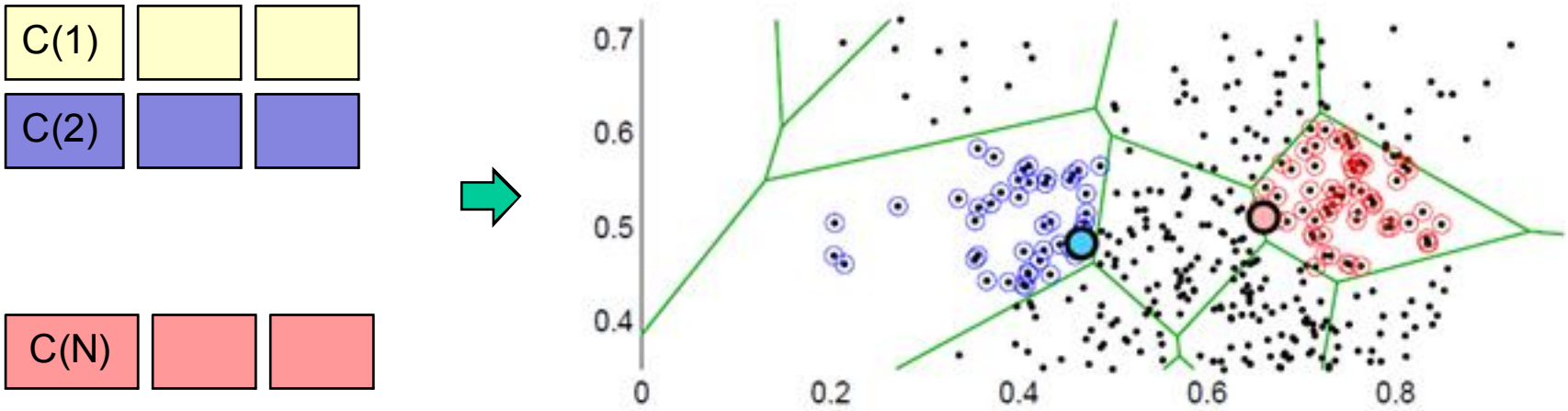
Инвертированный индекс



Квантованные вектора удобно записывать в форме «инвертированного индекса»

- K списков по числу кластеров (кодовых слов)
- В каждом списке храним индексы всех векторов, квантованных до C_i

Инвертированный индекс

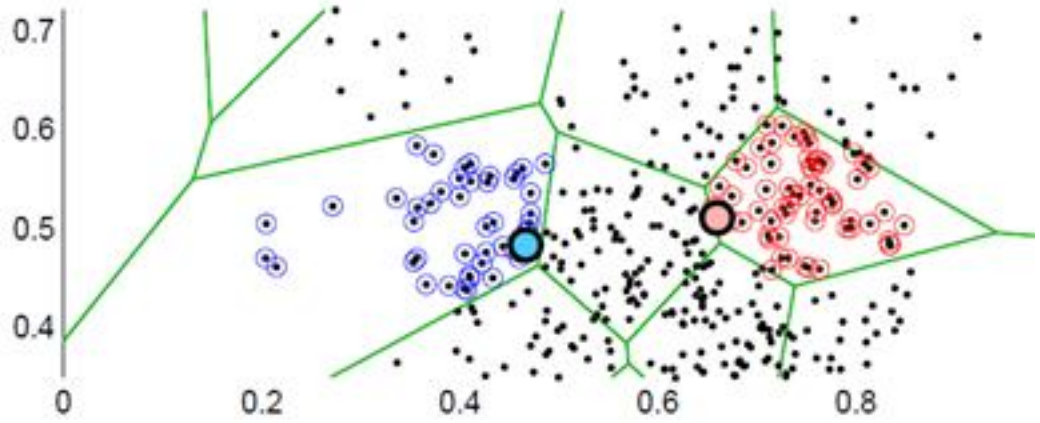


Поиск по индексу для запроса y:

- Квантуем y, получаем номер q(y)
- Выдаем все вектора, попавшие в список с номером q(y)

Проблема – не все примеры из списка одинаковы близки к q(y)

Ранжирование результатов



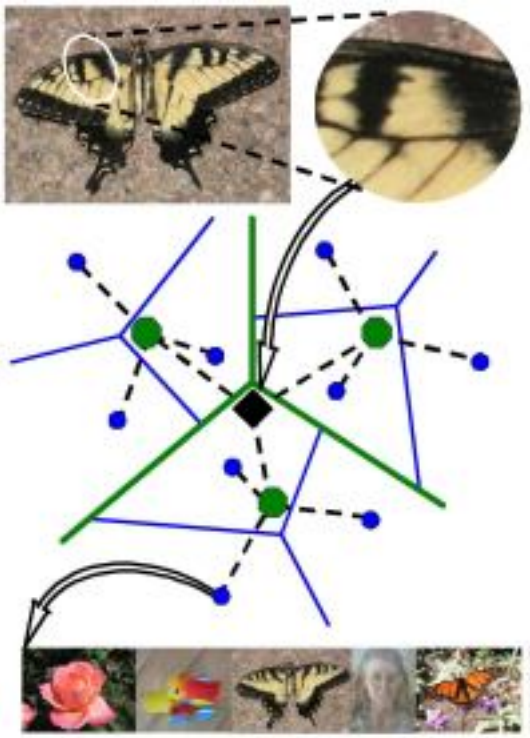
- Зачем выдавать все элементы в списке из одного кластера неупорядоченно?
- Упорядочим результаты (Re-ranking)
  - Рассчитаем расстояния от вектора-запроса до каждого элемента списка по полному дескриптору
  - Отсортируем результаты по близости (первые – ближайшие)
- Есть и другие способы ранжирования!

Проблемы квантования К-средними

- Чем больше K и длиннее код, тем меньше потеря информации при кодировании
- Для повышения точности нужно увеличивать число кластеров K
- Квантование до кодов 64 бит (0.125 бит на параметр GIST) требует кластеризации на 2^{64} кластеров, что напрямую невозможно
  - Сложность одного этапа К-средних O(NK)
  - Медленно на больших выборках (большие N)
  - Медленно при больших K
- Поэтому нужны другие, более быстрые приближенные методы, либо на замену k-means, либо в дополнение

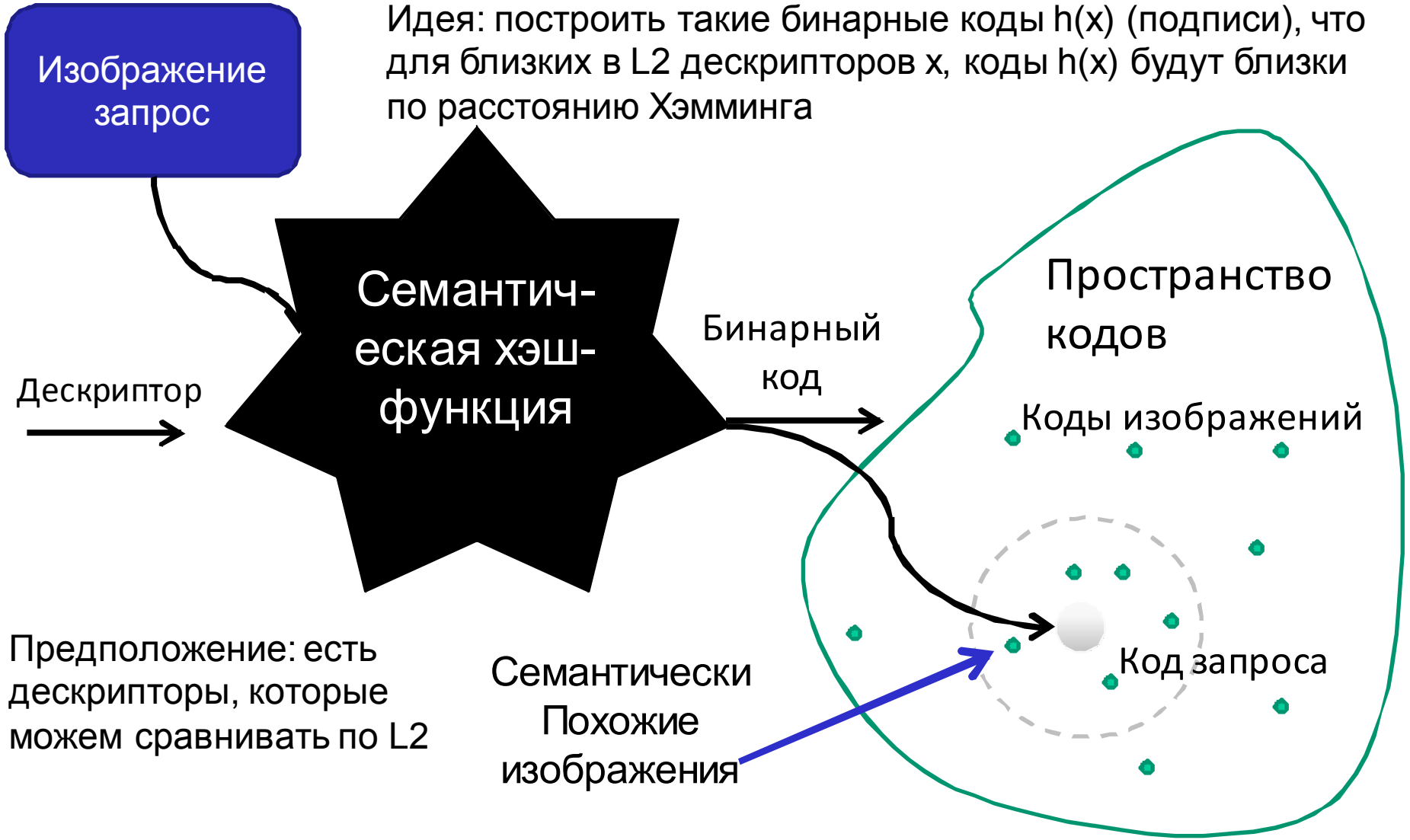
Hierarchical k-means (HKM)

- Иерархическое разбиение
  - Кластеризуем всё на K кластеров (K=10) с помощью K-средних
  - Затем данные в каждом кластере снова разбиваем на K кластеров
  - Повторяем до достижения нужной глубины
- Пример:
  - Глубина 6 даёт 1M листьев
- По точности проигрывает существенно K-средним
  - Есть другие приближённые методы (Approximate K-means)



D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In Proc. CVPR, 2006.

Семантическое хеширование



R. R. Salakhutdinov and G. E. Hinton. Semantic hashing. In SIGIR workshop on Information Retrieval and applications of Graphical Models, 2007.

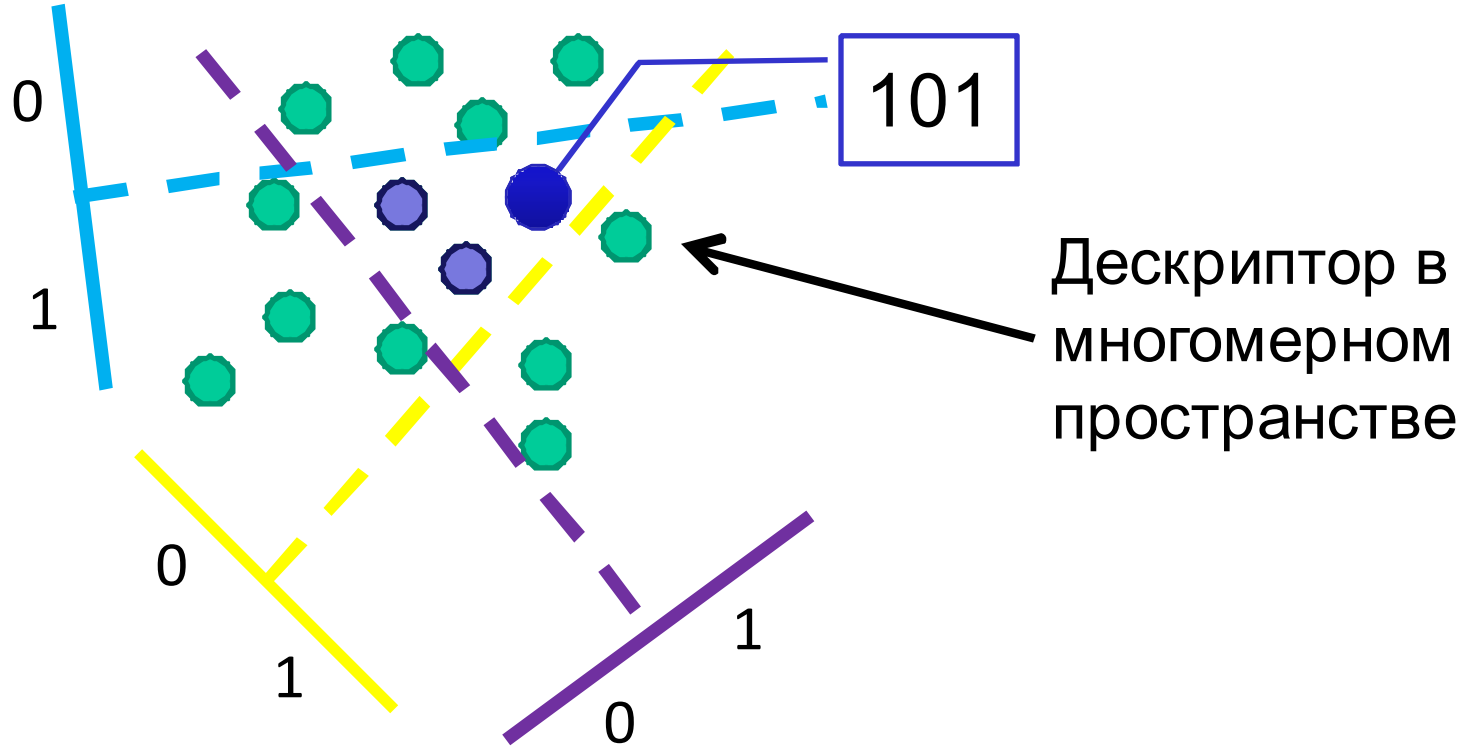
Отличие от квантования?

- И при квантовании k-средними и при семантическом хэшировании получаем бинарный код некоторой длины

- Разница в том, что при обычном квантовании разные номера никак не связаны друг с другом, а тут расстояние Хэмминга между сжатыми бинарными кодами коррелирует с расстоянием между векторами-прототипами

Locality Sensitive Hashing (LSH)

- Возьмем случайную проекцию данных на прямую
- Выберем порог близко к медиане проекций на прямую
- Пометим проекции 0 или 1 (1 бит подписи)
- С увеличением числа бит подпись приближает L2-метрику в исходных дескрипторах



A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In FOCS, 2006

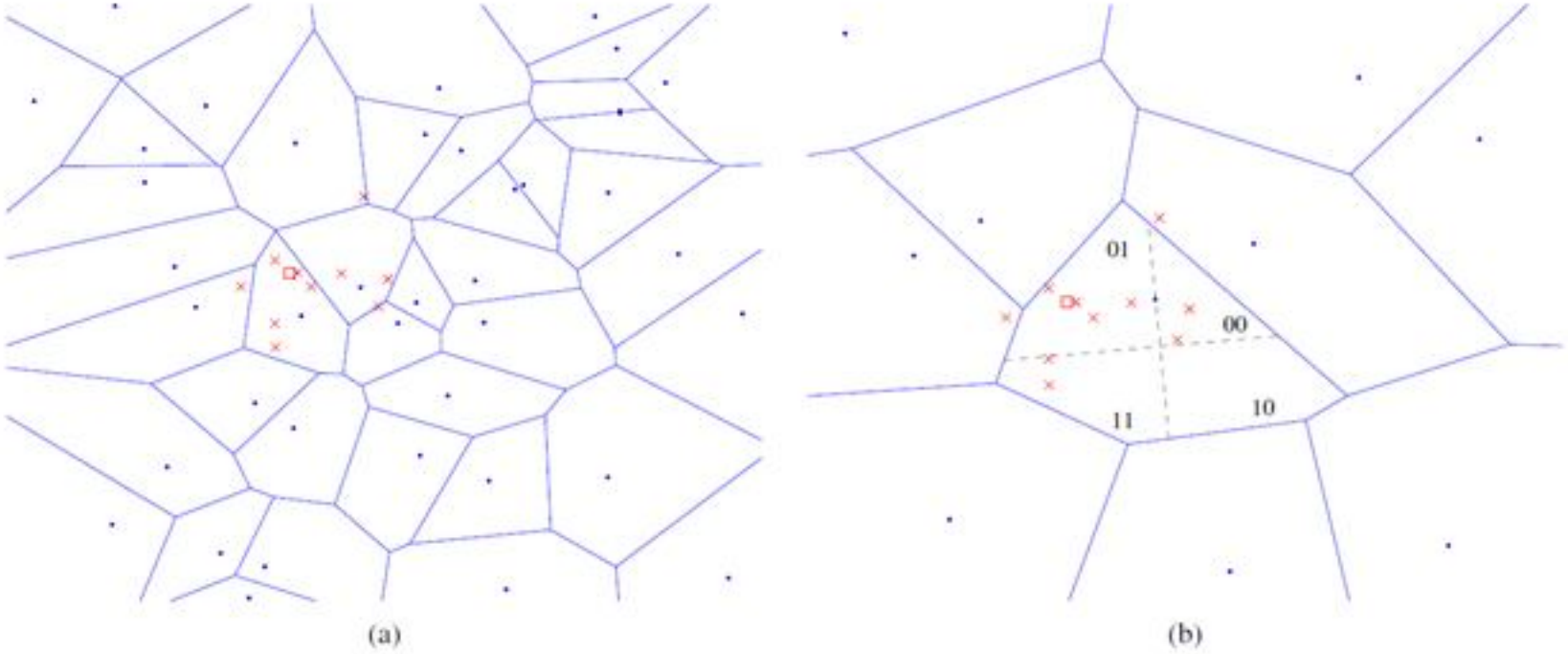
Locality Sensitive Hashing (LSH)

- Плюсы
  - Приближенный способ вычисления ближайшего соседа
  - Быстрое квантование (быстро вычисляем бинарный код)

- Недостатки:
  - Приближение L2 лишь асимптотическое
  - На практике может потребоваться слишком много бит для подписи

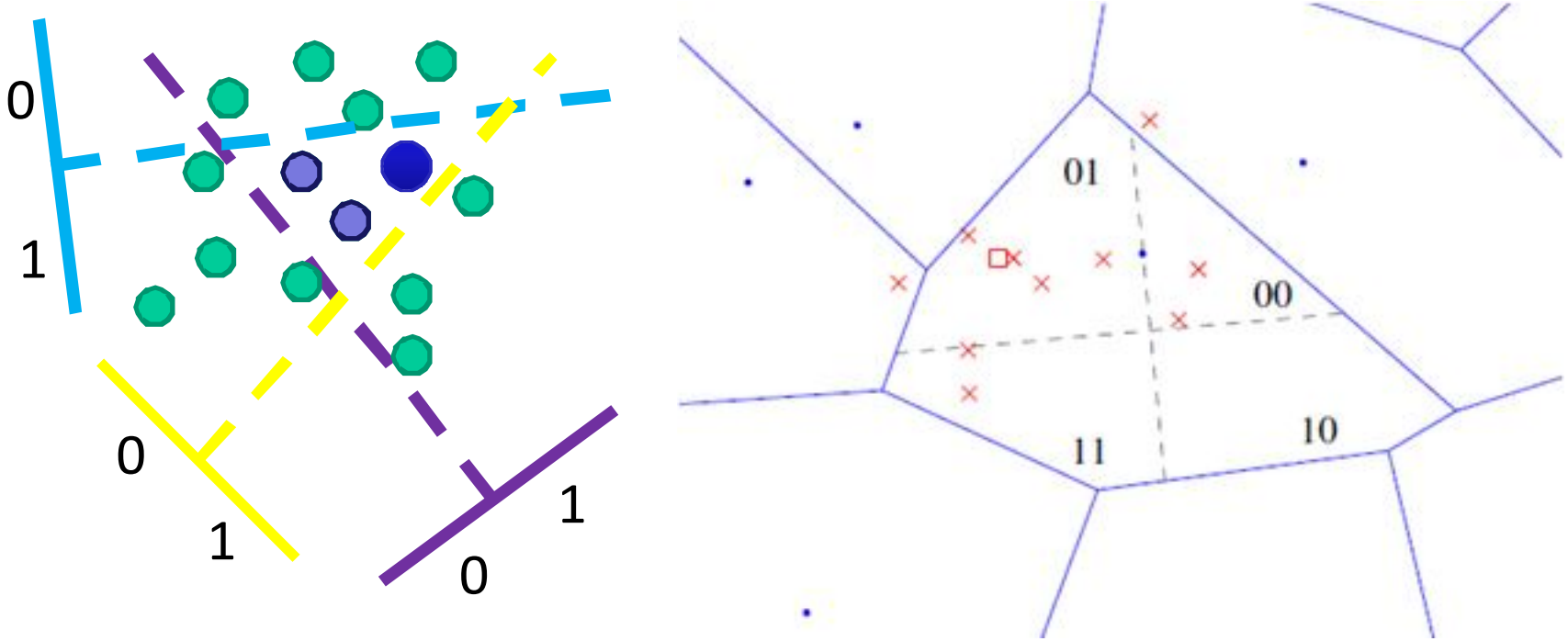
- Вывод:
  - Использовать как замену K-средних нельзя
  - Можно использовать в дополнение

Проблемы K-средних



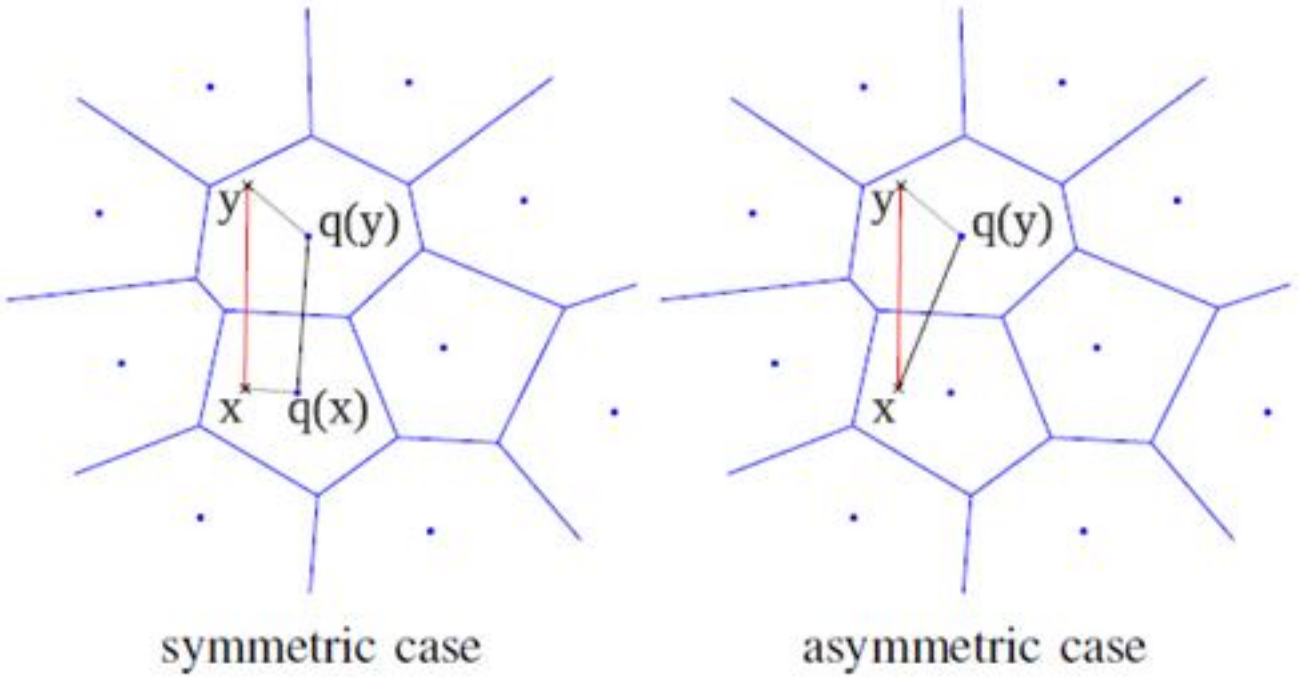
- Маленькое K – большие ячейки
  - Слишком грубый порог на сравнение!
- Большое K – маленькие ячейки
  - Медленнее кластеризация и квантование
  - Ячейка может оказаться слишком маленькой, не все соседи попадут

«Hamming Embedding»



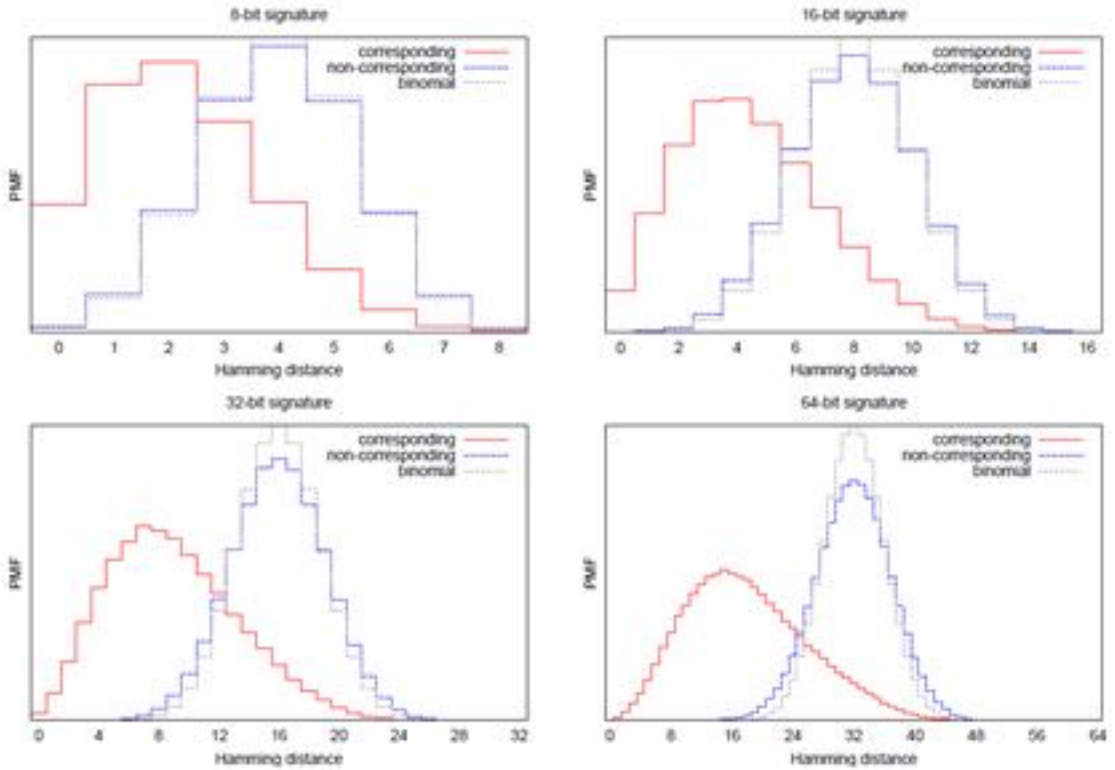
- Построим с помощью LSH коды, квантующие вектора (x-q(x)) для каждой ячейки независимо от остальных
- Будем использовать эти коды для сравнения дескрипторов, попавших в одну ячейку
- В итоге получается:
  - Вначале K-средние, а внутри каждой ячейки – квантование через LSH или другой метод семантического хэширования
  - Оказалось лучше 2х этапного иерархического K-среднего

Сравнение векторов



- Можем применять семантическое хэширование и к записям в базе *y*, и к запросу *x* (симметричное сравнение)
- Можем сравнивать вектор-запрос x с реконструированной версий квантованных векторов (ассиметричное сравнение)
- Ассиметричный вариант точнее

Влияние кодов



- Посмотрим распределение расстояний Хэмминга между правильными и неправильными парами
- Сравнение по кодам показывает, что при длине кода > 32 бит наблюдается заметная разница между правильными и неправильными сопоставлениями

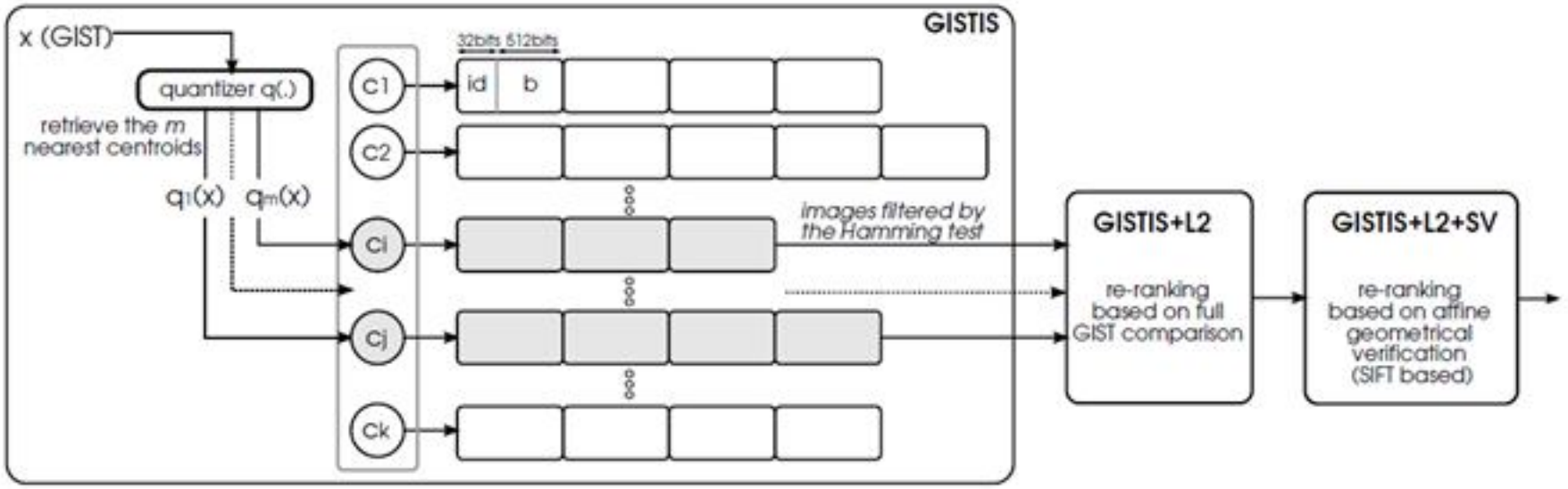
Алгоритм GISTIS

- GIST Indexing Structure
  - Добавим в наш простой алгоритм для поиска полудубликатов hamming embedding

- Схема метода:
  - Строим GIST для каждого изображения
  - Кластеризуем все дескрипторы с помощью k-means на k=200 слов
  - Применяем LSH к каждому кластеру и для всех векторов в кластере считаем бинарную подпись
  - Идентификатор картинки и бинарная подпись хранится в индексе в RAM

M.Douze et.al., Evaluation of gist descriptors for web-scale image search. In International Conference on Image and Video Retrieval. ACM,2009.

Схема метода



- В индексе в RAM хранится только бинарная подпись изображения (512 бит) и идентификатор
- Сам GIST хранится на жестком диске
- Можем проводить сортировку несколько раз:
  - Вначале по бинарным подписям
  - Затем по GIST с жёсткого диска

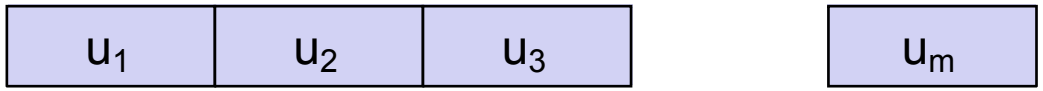
Результаты

	Байт на изображение в RAM	Время на построение дескриптора	Время поиска в базе из 110M изображений
GIST	3840	35мс	1.26с
GISTIS	68	36мс	2мс
GISTIS + L2	68	36мс	6/192мс

M.Douze et.al., Evaluation of gist descriptors for web-scale image search. In International Conference on Image and Video Retrieval. ACM,2009.

Квантование через произведение

- Вспомним:
  - Для векторов SIFT квантование до кодов 64 бита (0.5 бита на параметр) требует подсчёта и хранения 2^{64} центроидов, что невозможно
- Идея “product quantization”:
  - Разобьём вектор x длины D на m частей
  - Квантуем каждый подвектор u_j независимо от остальных



  - Пусть каждый подвектор квантуем на k* центроидов, тогда всего центроидов $(k^*)^m$
  - Длина кода $l = m \log_2 k^*$
- Сравнение с K-средними по памяти
  - Память: kD (K-средние) и $mk^*(D/m)=k^{1/m}D$

H.Jégou et. al. Product quantization for nearest neighbor search, PAMI 2011

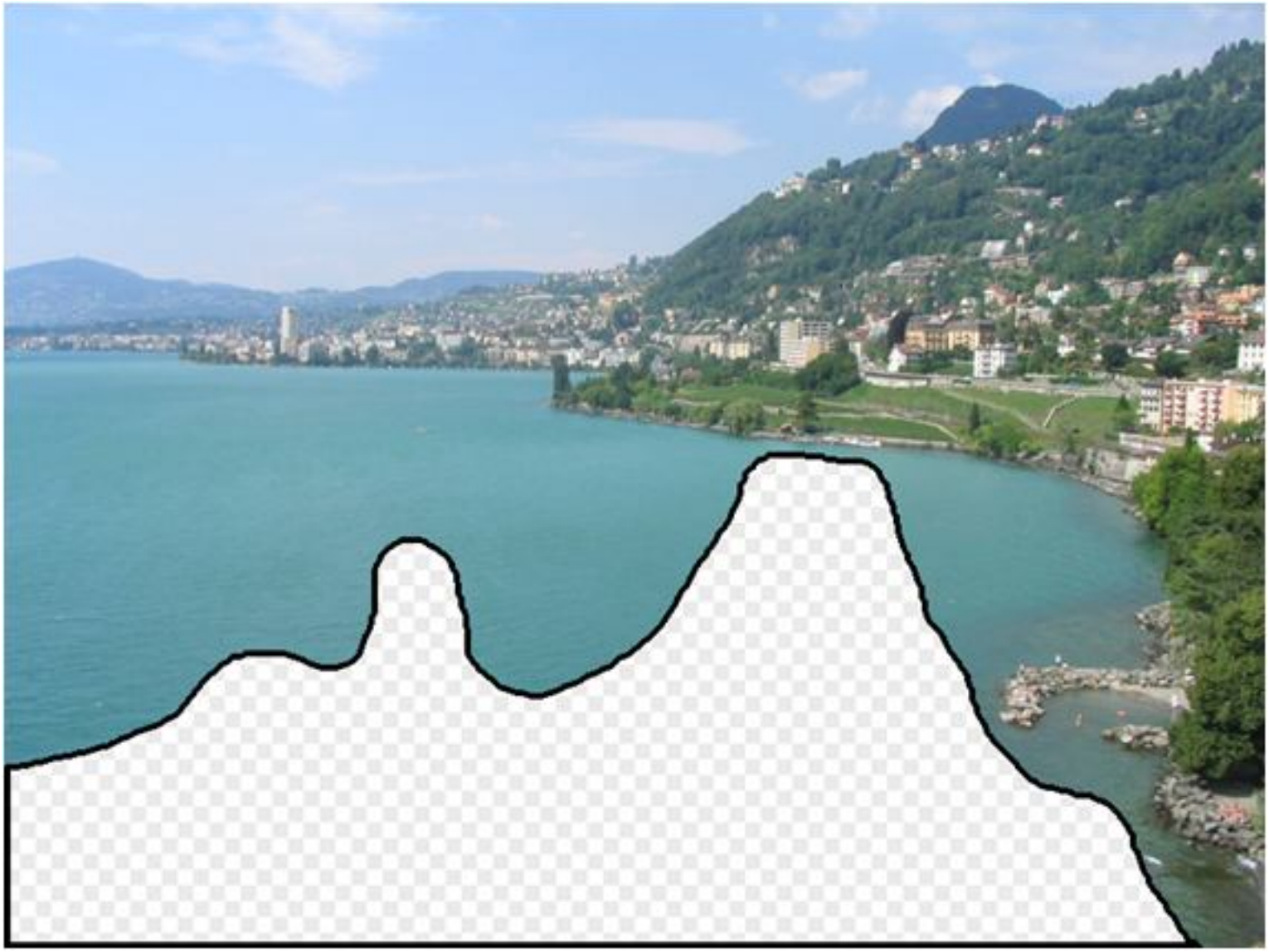
Резюме поиска полудубликатов

- Для каждой картинки строим дескриптор и записываем его в инвертированный индекс
- GIST – неплохой дескриптор для поиска полудубликатов
- Размер GIST слишком большой для поиска по большим коллекциям, поэтому нужно делать ускоренные приближенные методы поиска ближайших соседей.
- Мы рассмотрели:
  - Квантование k-средними и иерархический метод k-средних
  - Семантическое хеширование – LSH, квантование произведения

Попробуем решить такую задачку:

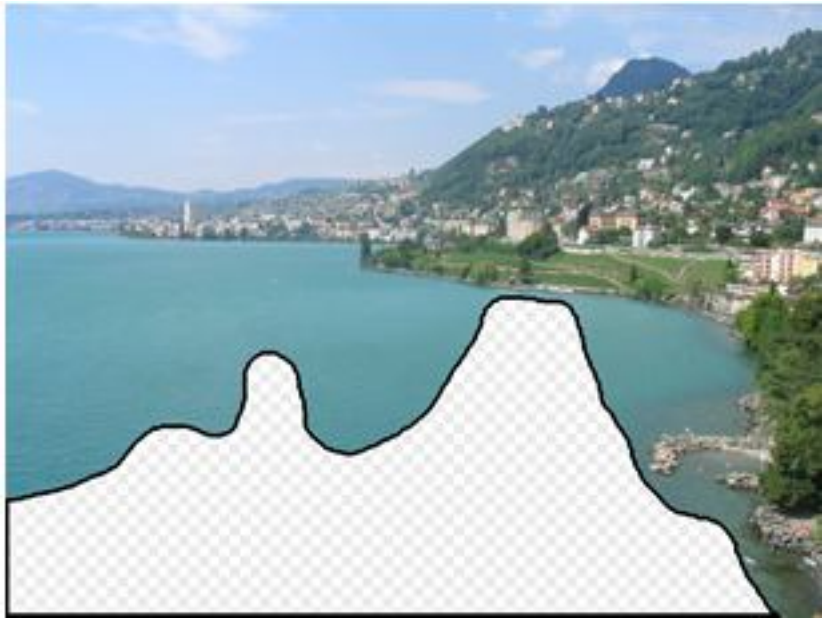
James Hays, Alexei A. Efros Scene Completion using Millions of Photographs, *SIGGRAPH 2007*

«Inpainting»



Реконструкция изображения в невидимой области

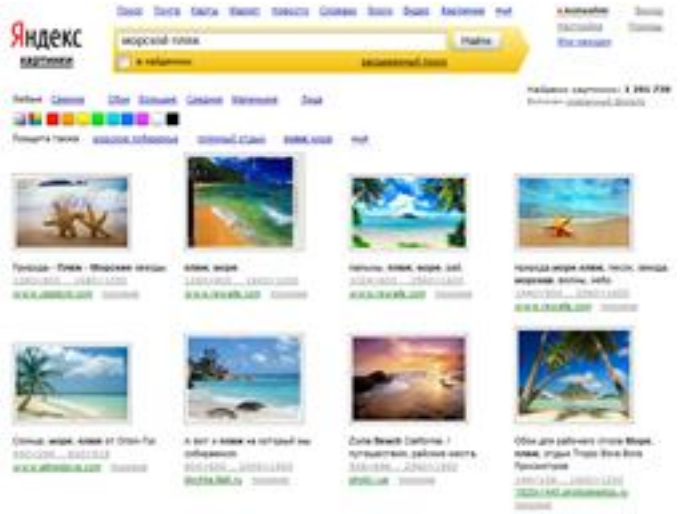
Как будем решать?



- Если бы мы нашли в интернете подходящую картинку, мы бы могли просто ей просто «закрыть дырку»

Как будем решать?

- Скачаем из интернета множество пейзажей по разным ключевым словам
- Очень много – миллион и больше картинок
- Будем искать похожие картинки в этой базе
- Воспользуемся методом поиска полудубликатов



Изображение-запрос



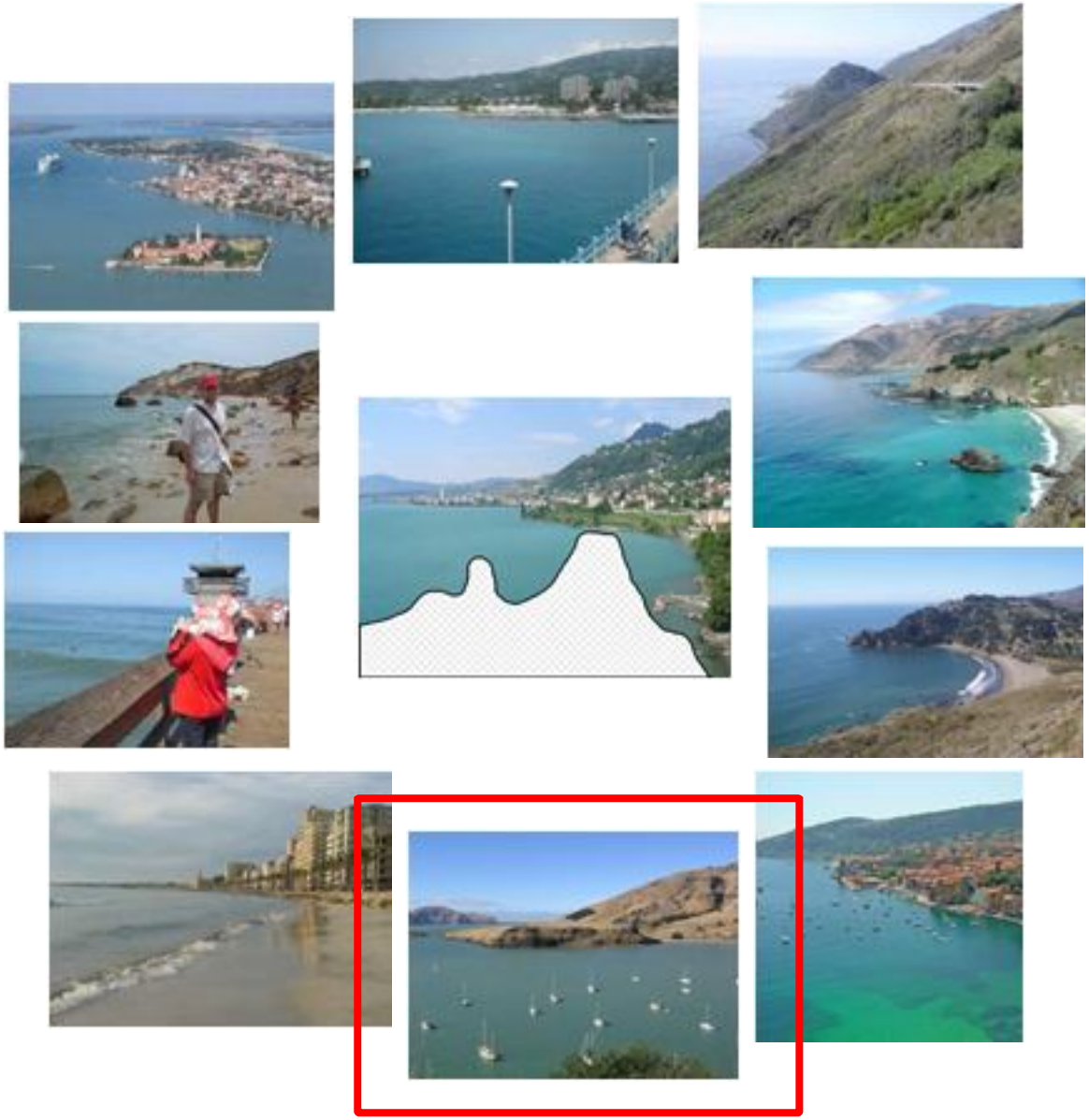








10 ближайших из 20,000 изображений



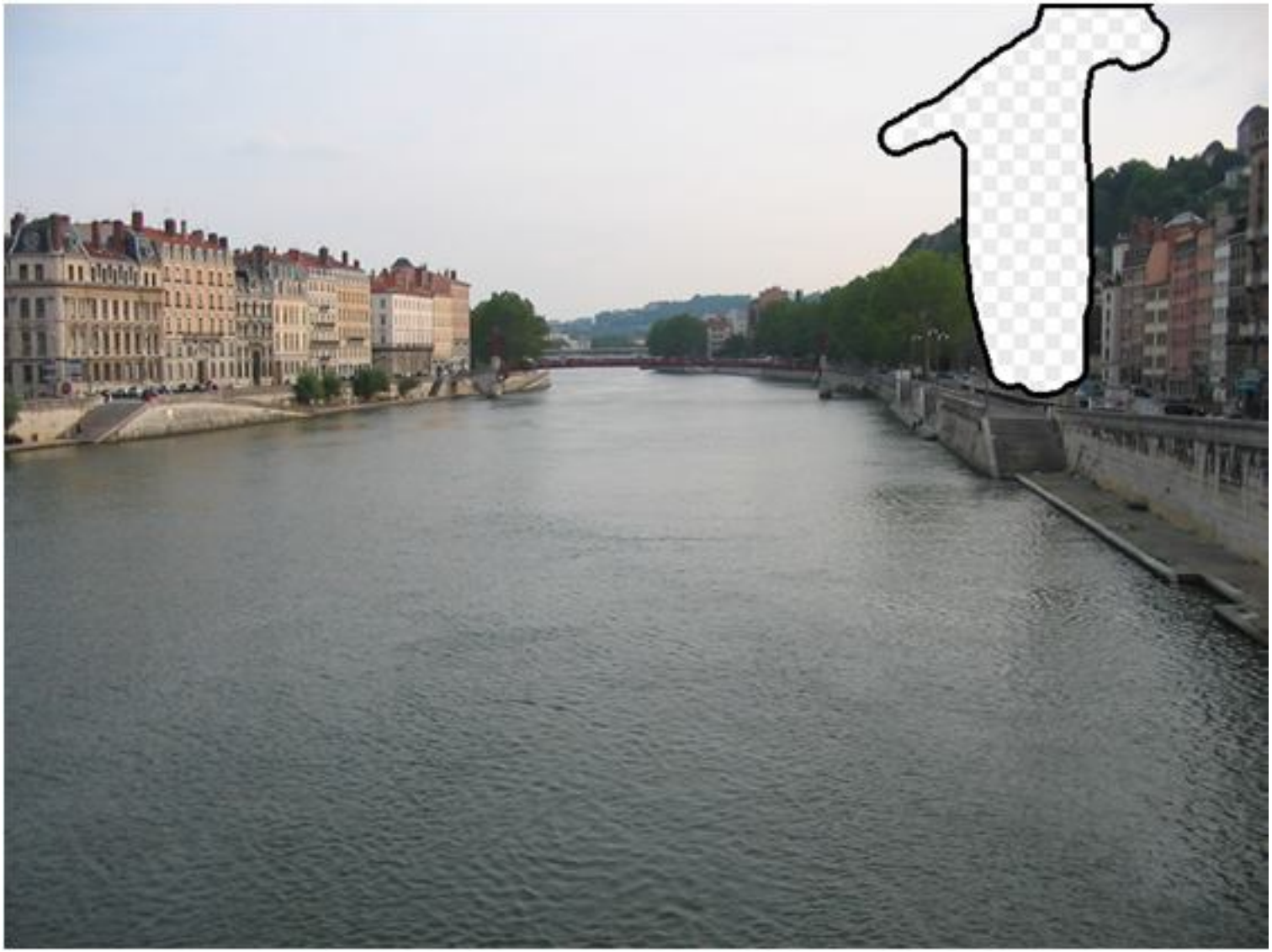
10 ближайших из 2х миллионов изображений

Закрываем дырку

Цифровой фотомонтаж (разрезы графов + смешение по Пуассону)

Пример

Пример



Пример

Пример

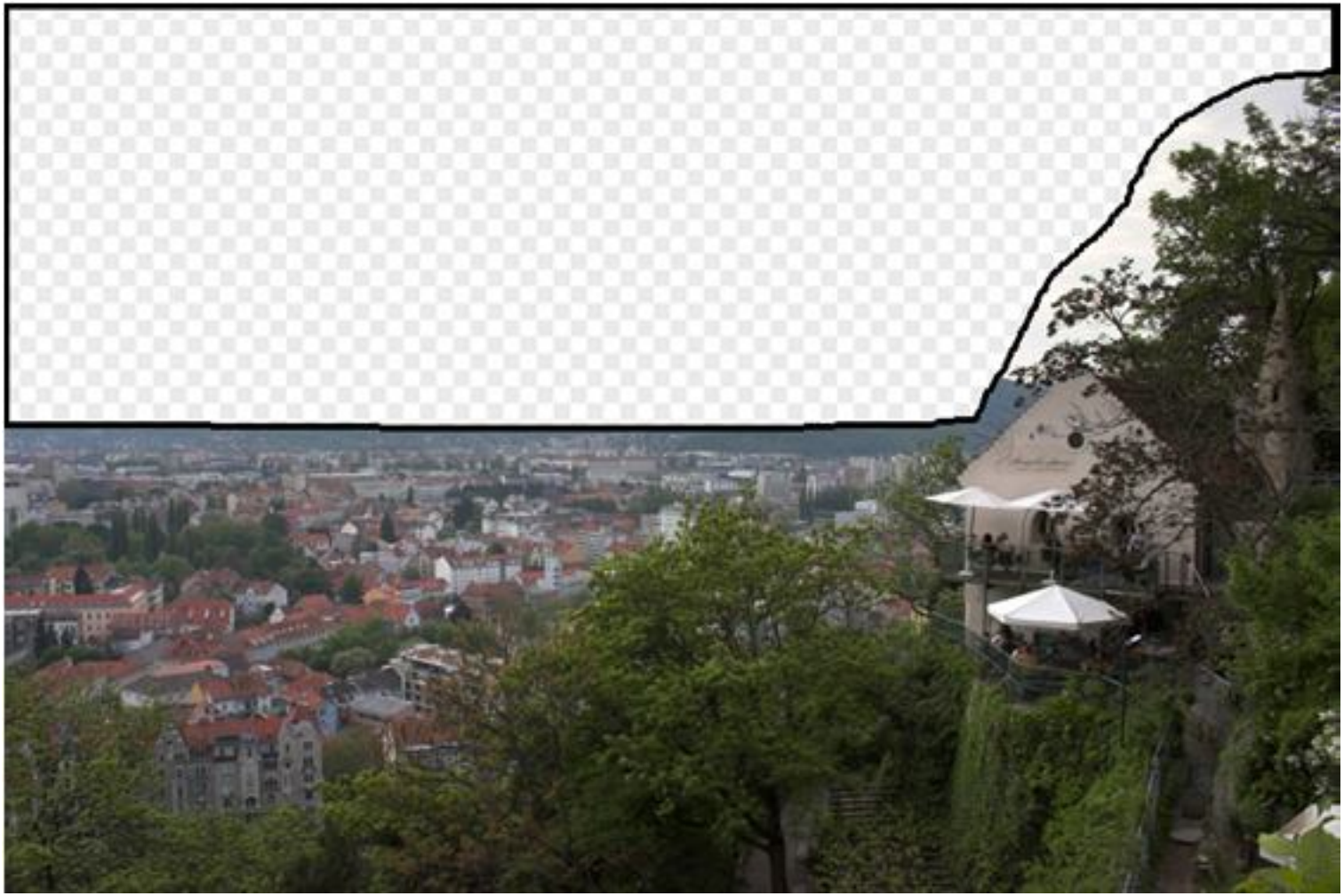
… 200 ближайших

Пример

Пример

Пример

Пример

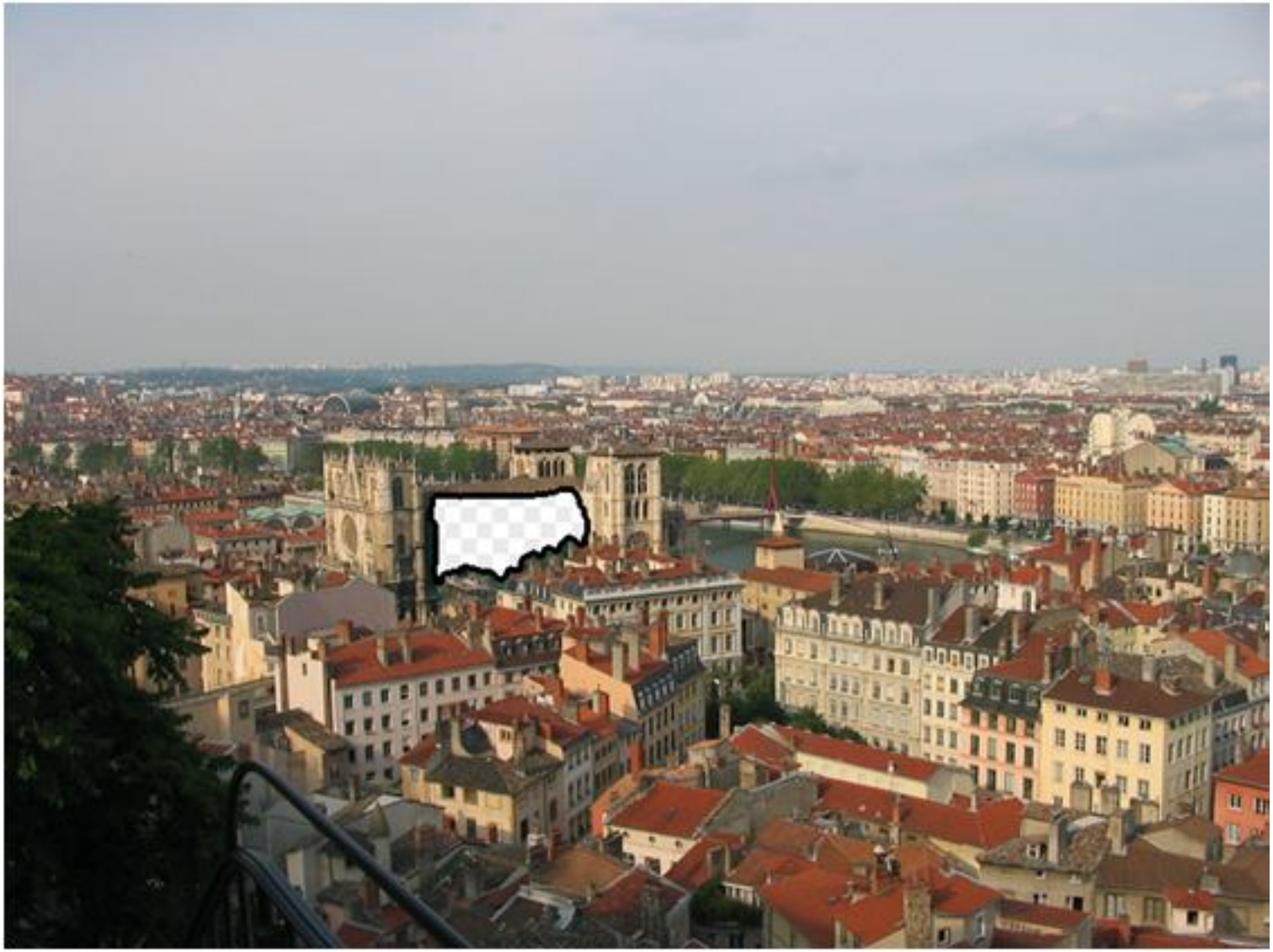


Пример

Пример

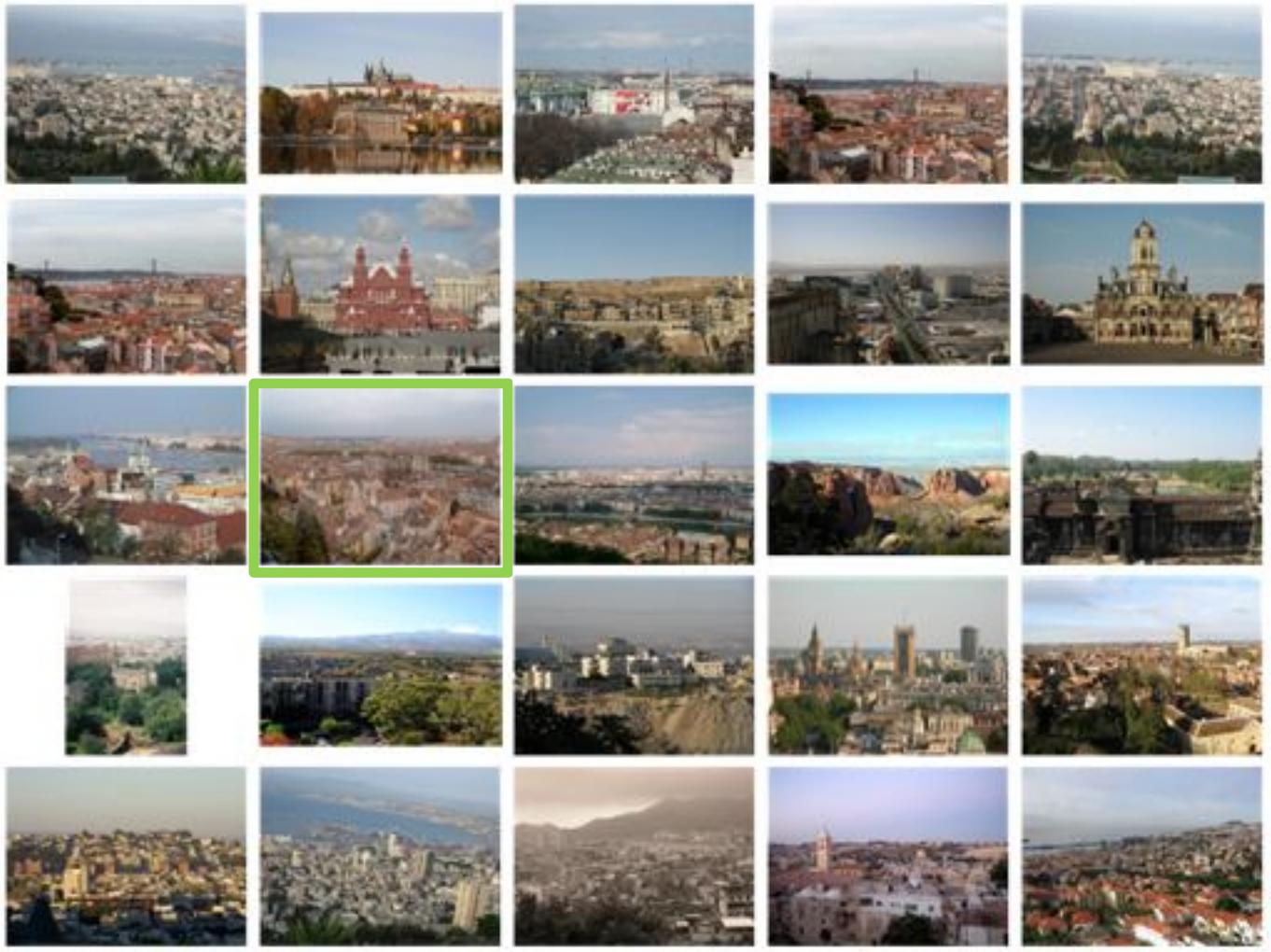
Пример

Пример



Пример

Пример



… 200 ближайших

Изображение с того же города!

im2gps

- Собрали 6M картинок из Flikr с проставленными GPS-метками
- К дескриптору GIST добавили кучу дополнительной информации, требующей долгого обсчёта
- Кластер из 400 машин для аннотации всех 6M изображений
- Теперь по комбинированному дескриптору можем искать
- Найдем, вот эту картинку:

James Hays, Alexei A. Efros **im2gps: estimating geographic information from a single image**, *CVPR 2008*.


Paris


Paris


Paris


Paris


Paris


Paris


Paris


Madrid


Rome


Paris


Cuba


Paris


Paris


Poland


Paris


Paris

Отображение результатов на карте

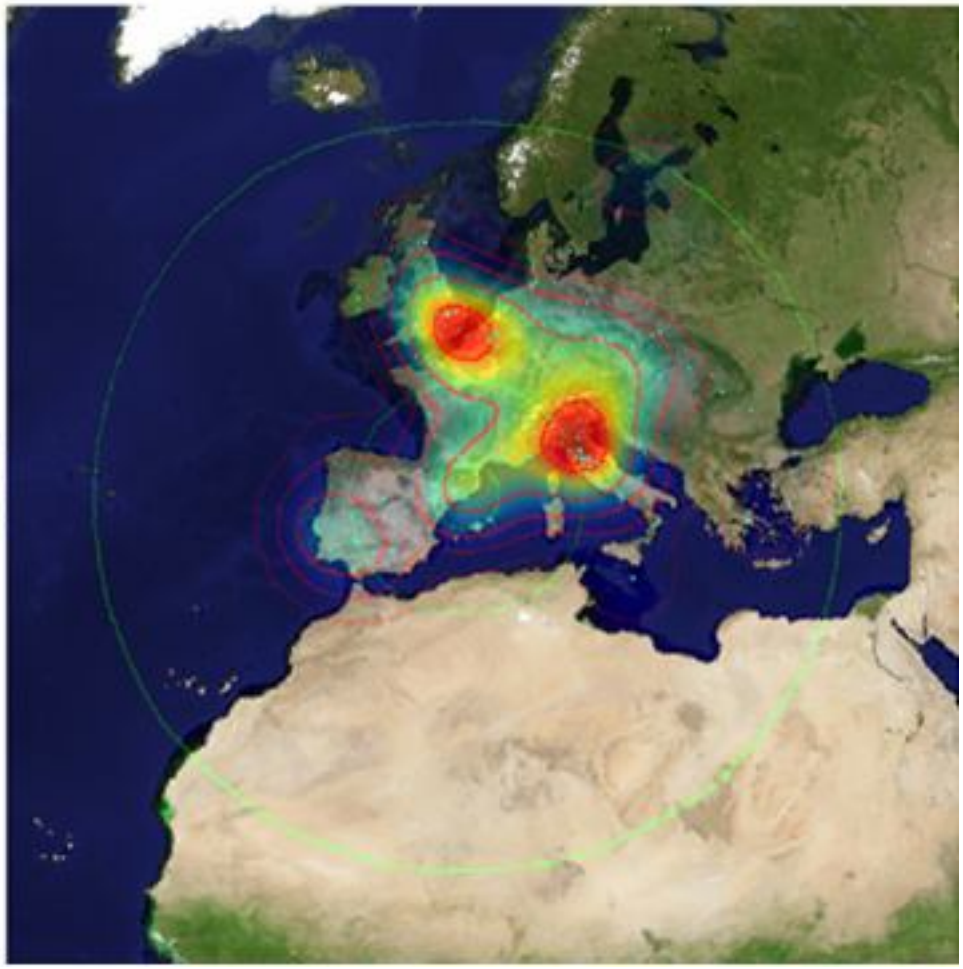
Найдем теперь такую:

Что мы нашли:

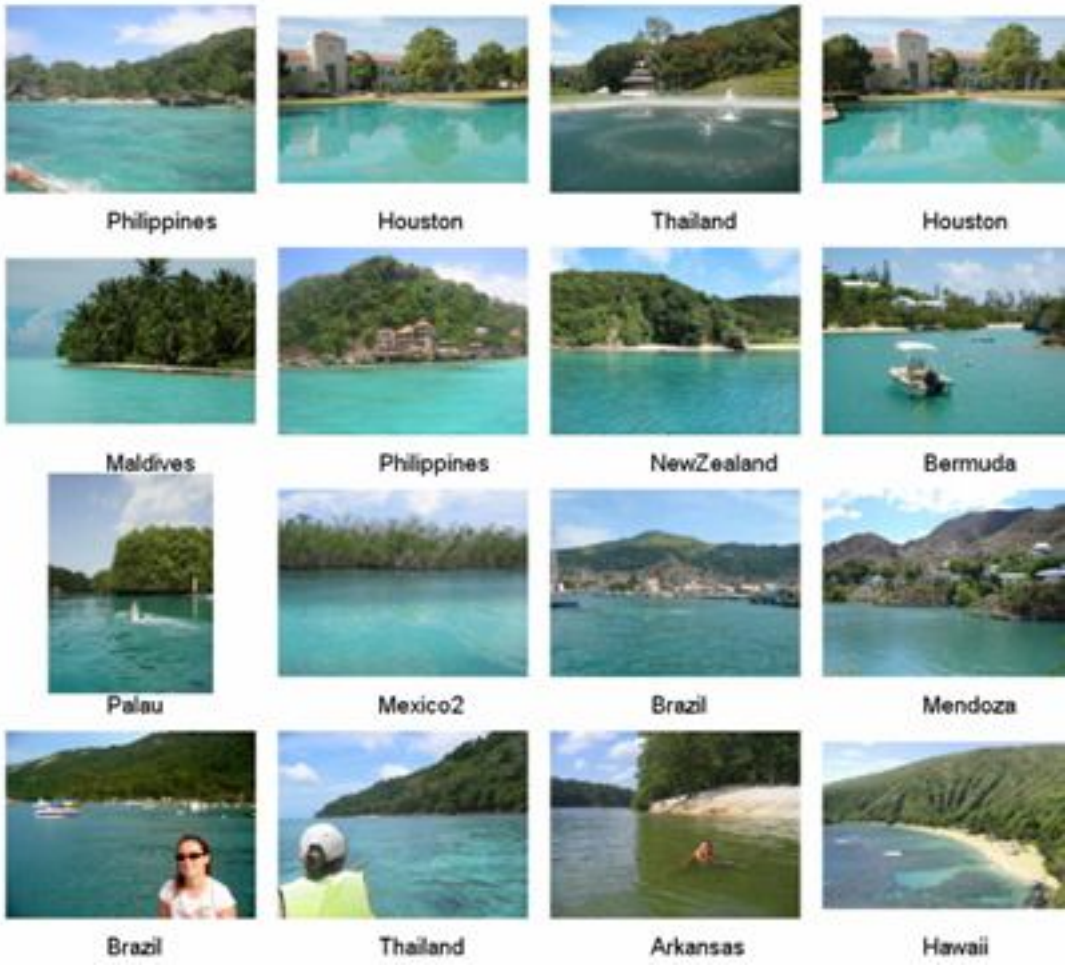

Madrid
england
France
Paris
Croatia
heidelberg
Macau
Malta
Cairo
Italy
Italy
Italy
Latvia
europe
Barcelona
Austria

Отображение результатов на карте

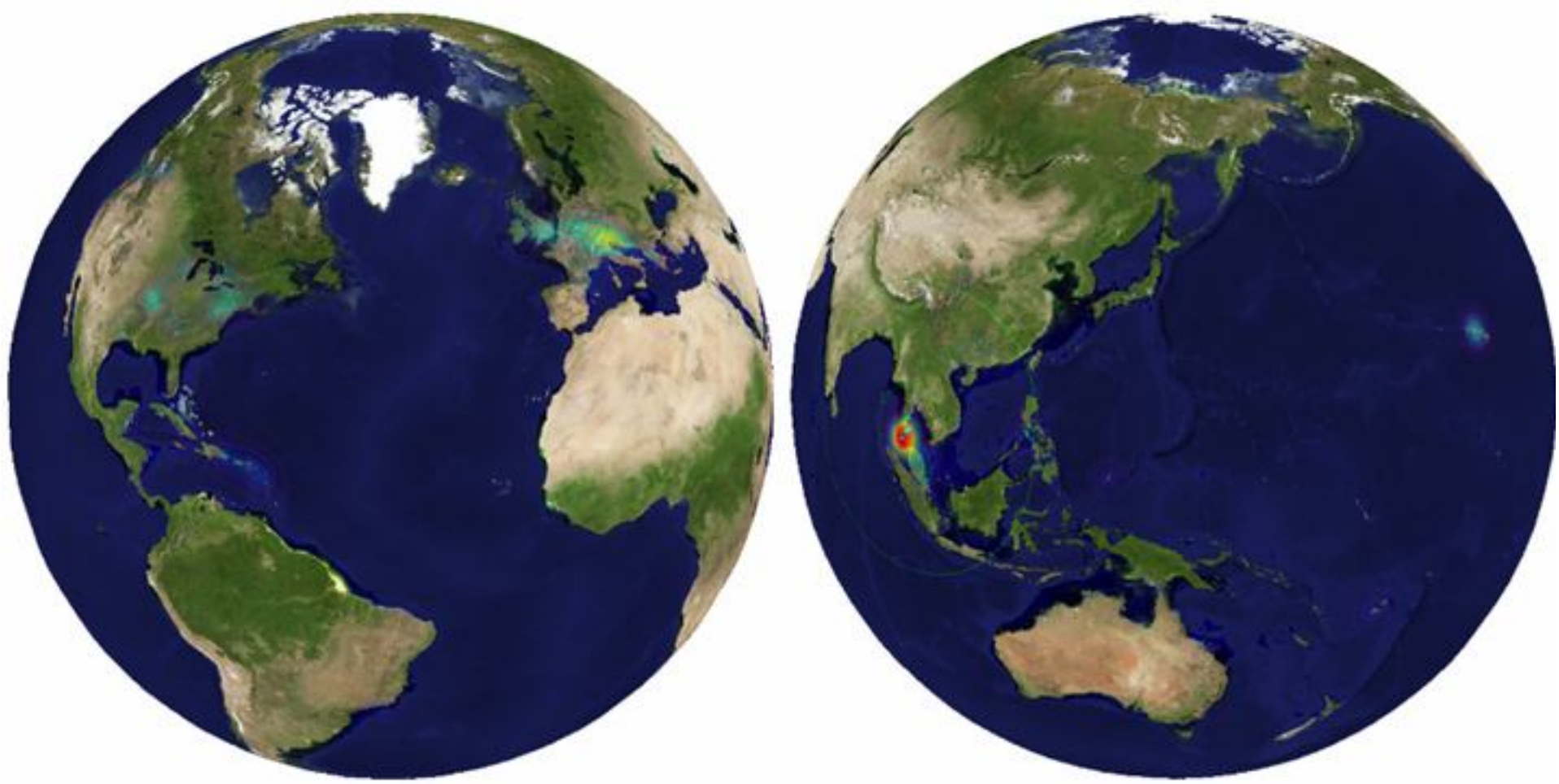
200 результатов, кластеризуем и покажем центры
и распределение картинок



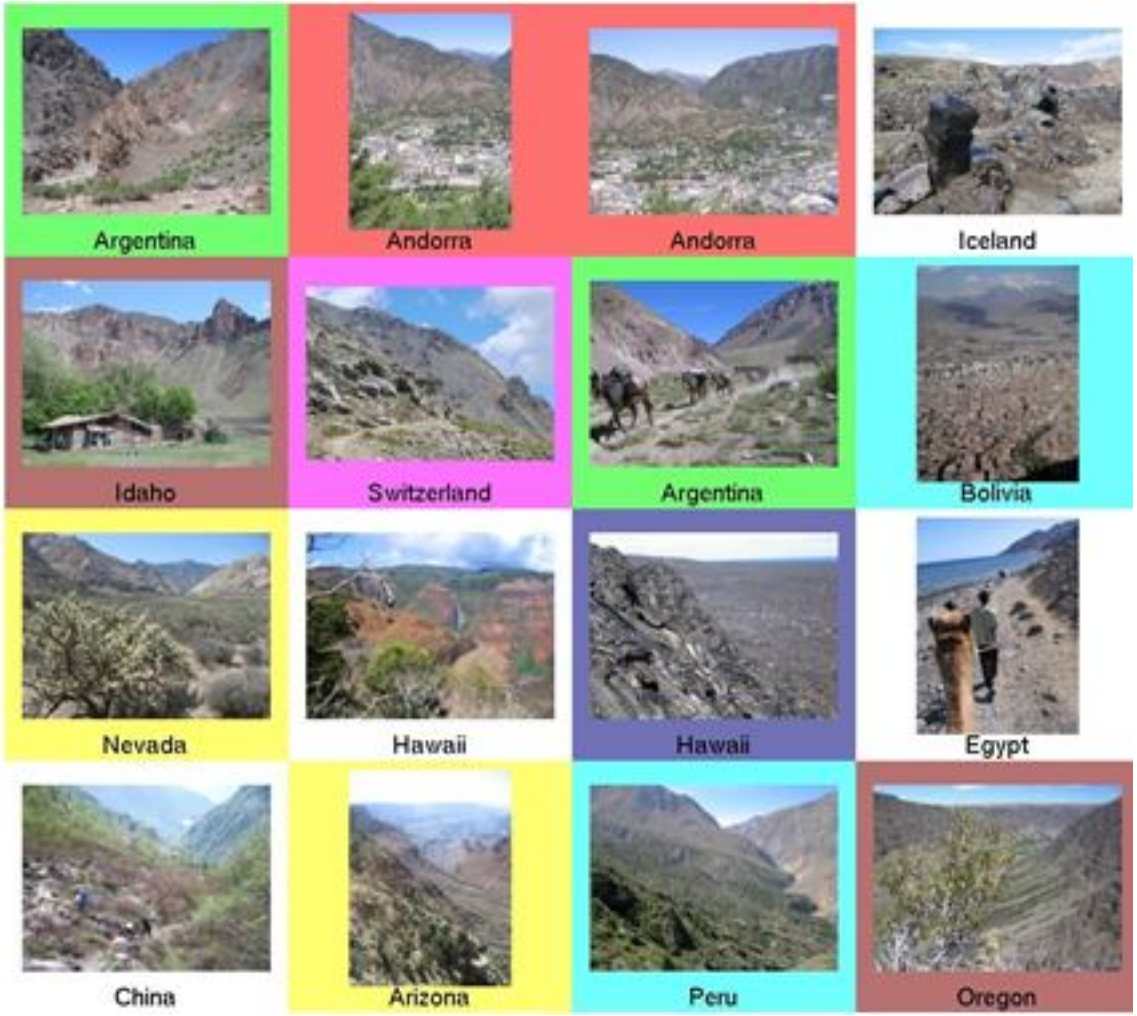
Пример


Philippines
Houston
Thailand
Houston
Maldives
Philippines
NewZealand
Bermuda
Palau
Mexico2
Brazil
Mendoza
Brazil
Thailand
Arkansas
Hawaii

Пример

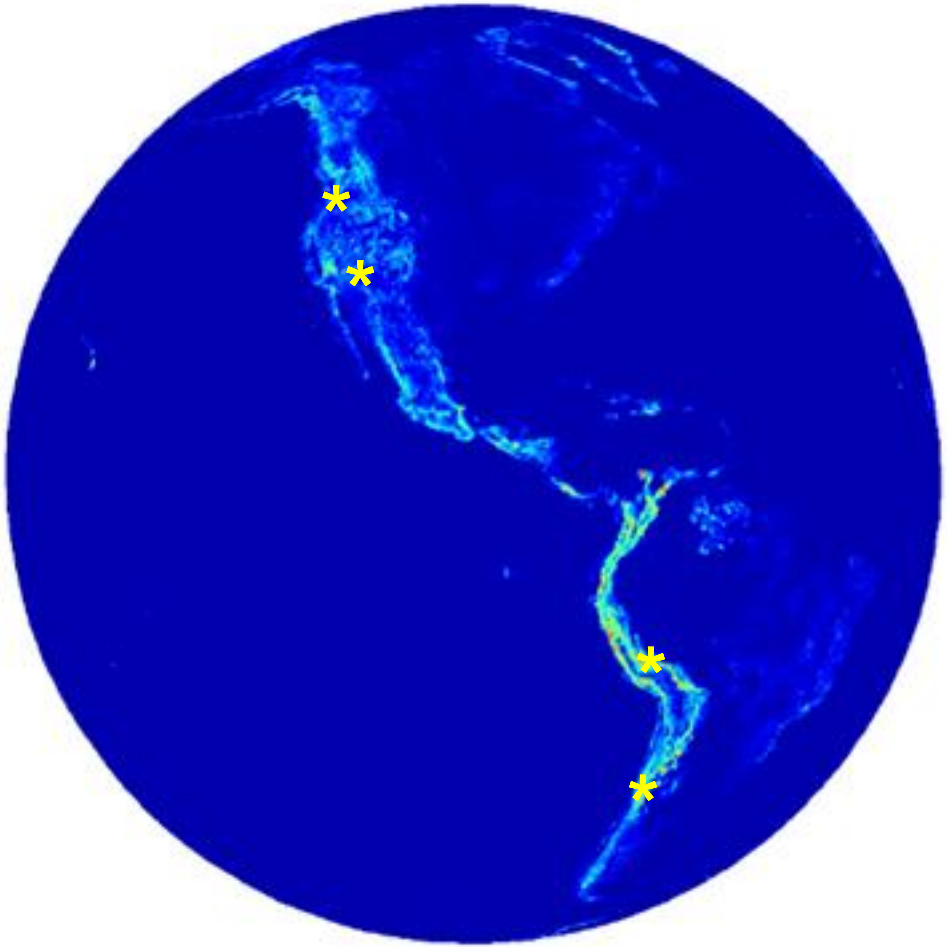


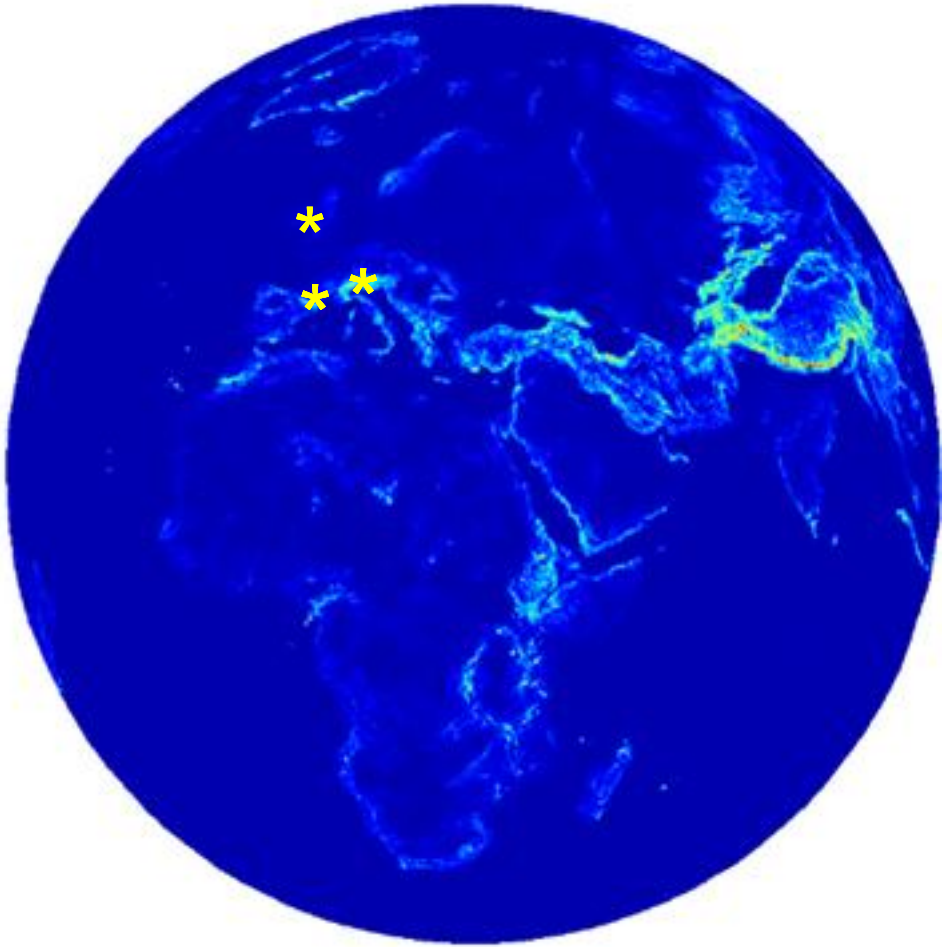
Категоризация, регрессия



Если для изображений есть дополнительная аннотация, то можем сделать «kNN» классификацию, используя похожесть изображений как метрику близости

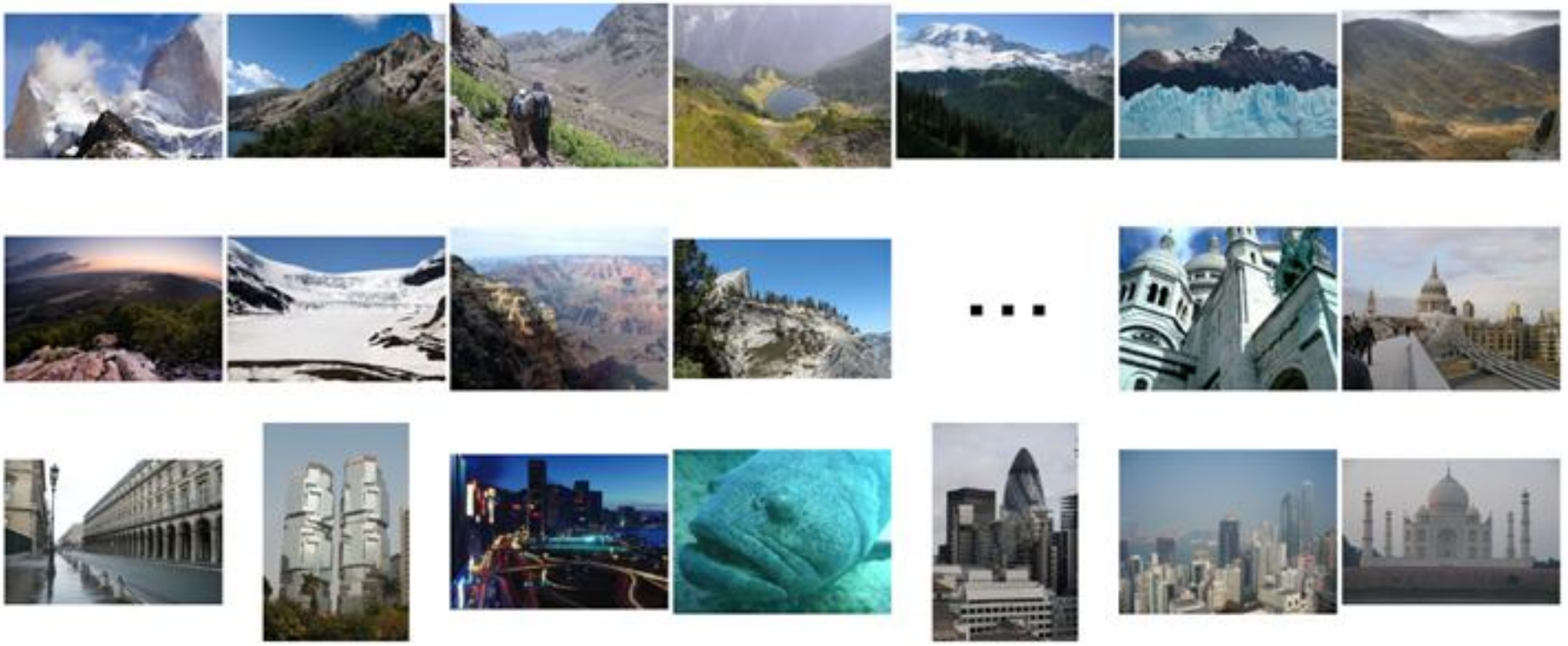
Скорость = 112 м / км





Уклон

Ранжирование изображений по уклону (от макс к мин)



Плотность населения

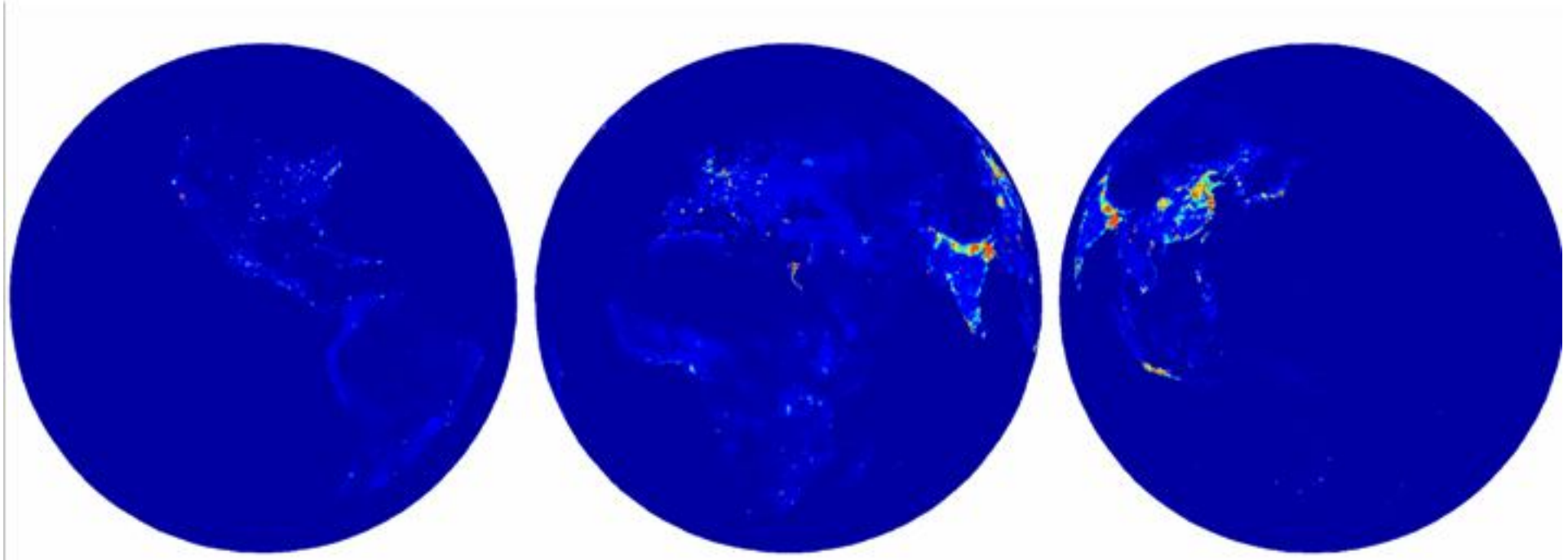
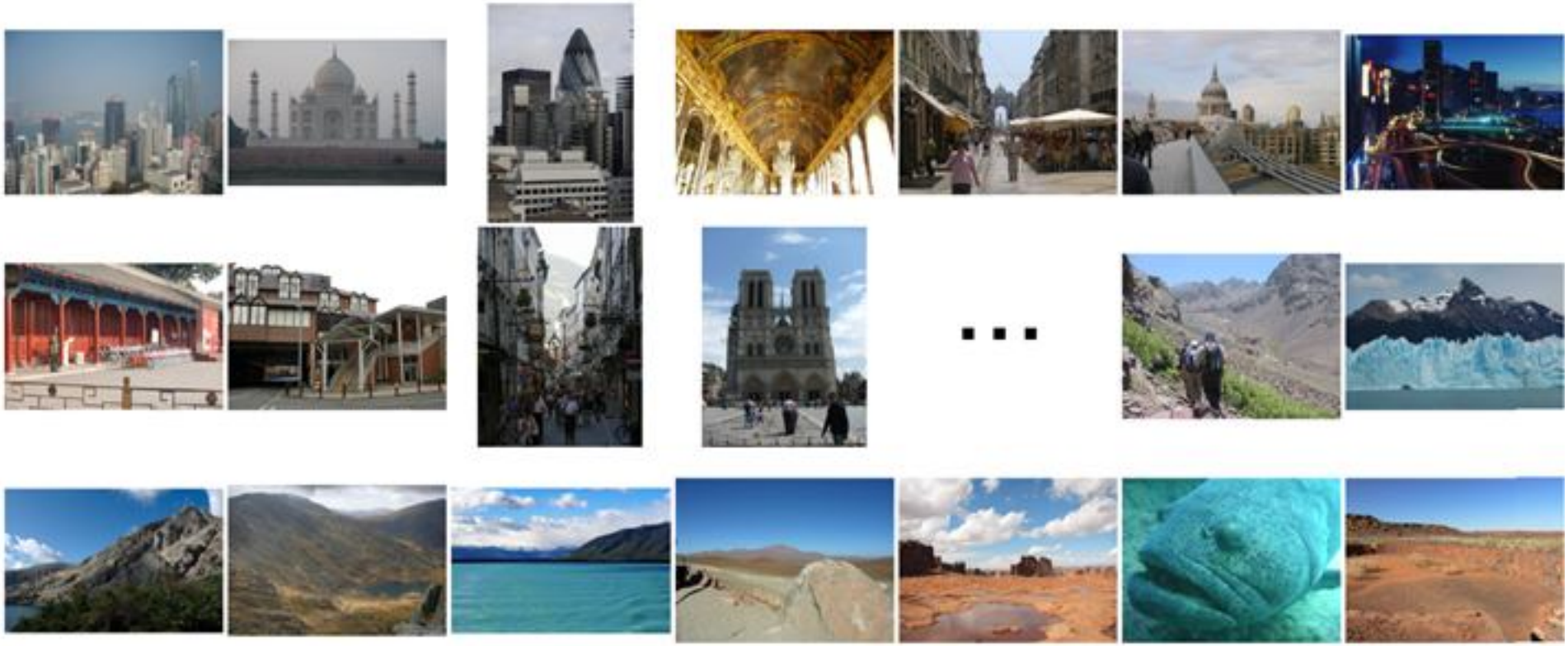


Figure 2. Global population density map.

Ранжирование по населению



Промежуточное резюме

- В больших коллекциях изображений можно найти очень близкие изображения (практически полудубликаты)

- Если есть дополнительная аннотация изображений базы, то KNN поиск по большой базе позволяет неплохо решать задачи регрессии по изображениям

Крошки-картинки (Tiny images)

А что мы можем сделать, если у нас есть сто миллионов картинок?

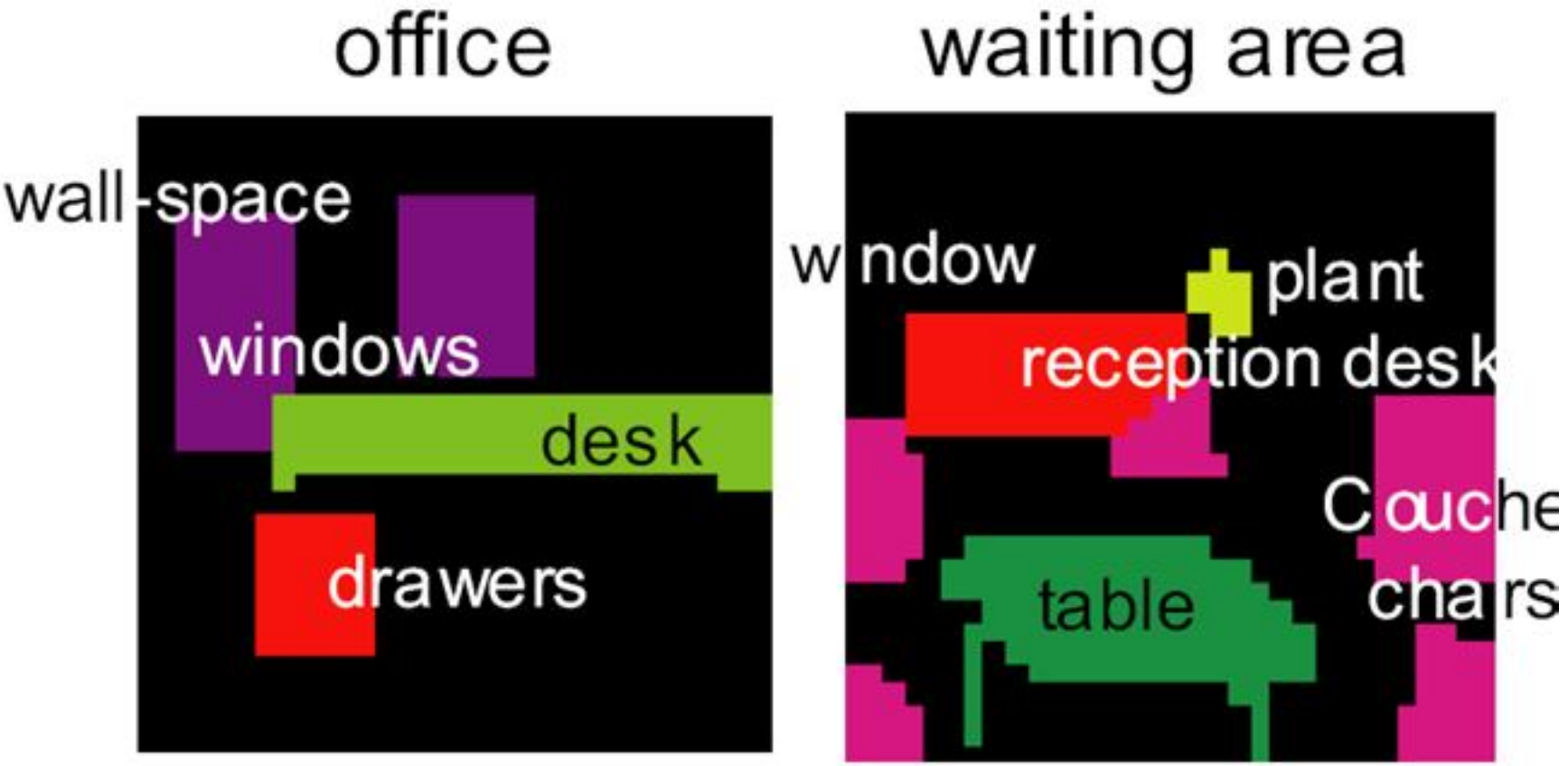
A. Torralba, R. Fergus, W. T. Freeman 80 million tiny images: a large dataset for non-parametric object and scene recognition IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.30(11), pp. 1958-1970, 2008.

Примеры изображений

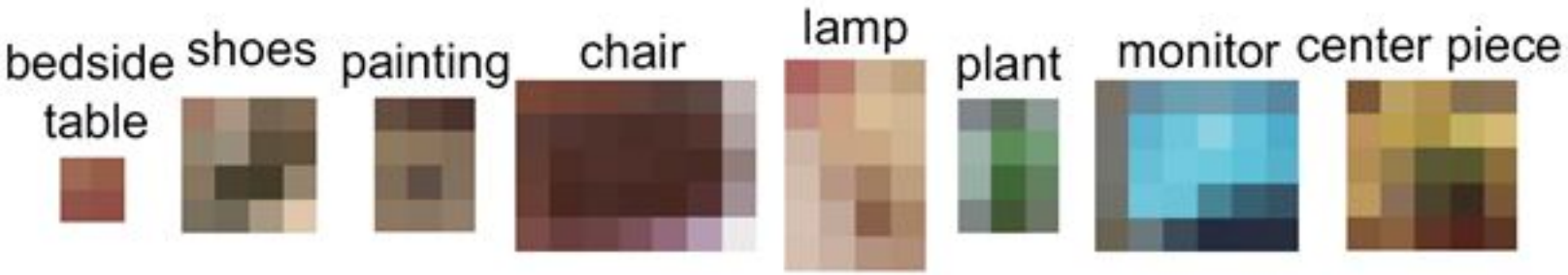
Какого размера нам потребуются картинки, чтобы их можно было использовать для распознавания? Попробуем на человеке...

Примеры изображений

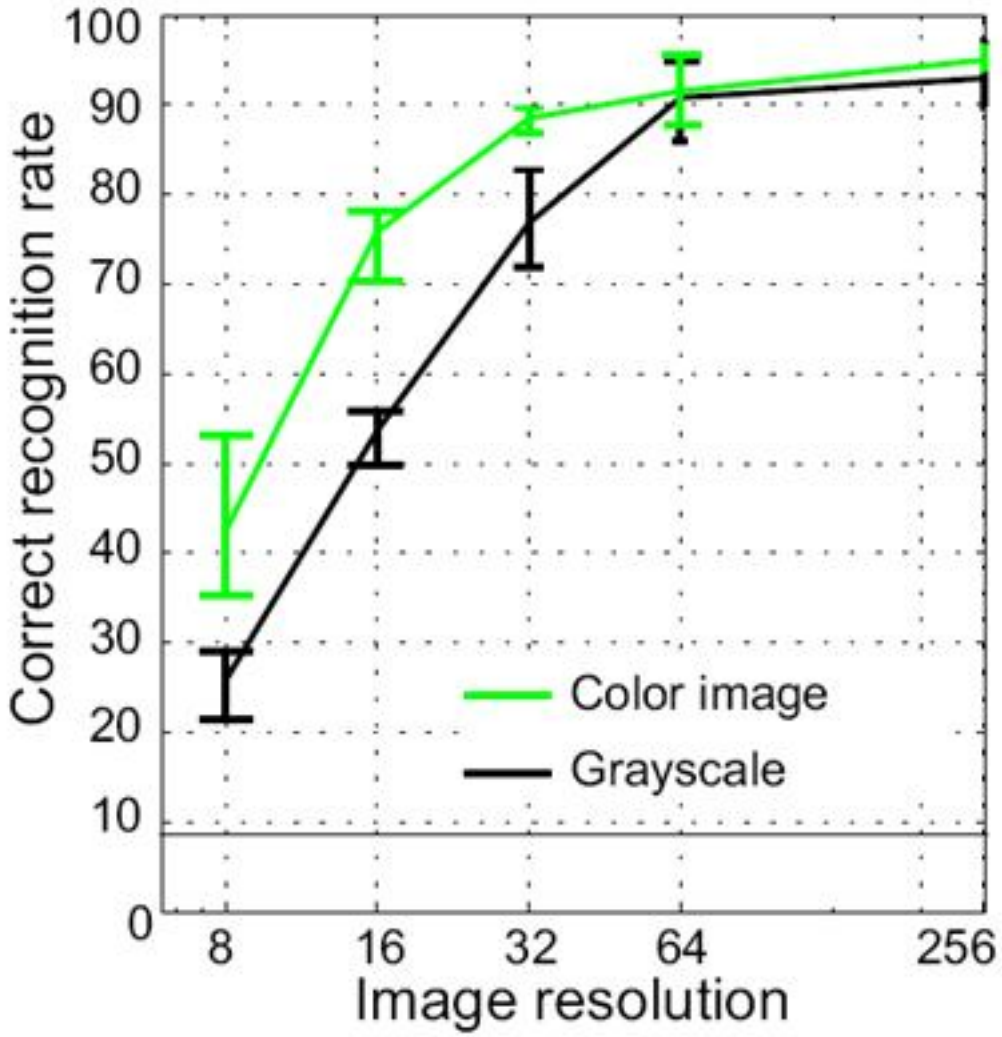
Сегментация



Отдельные объекты



Распознавание человеком



WordNet
A lexical database for English

http://wordnet.princeton.edu/

About WordNet

- About WordNet

Use WordNet online

Download

Frequently Asked Questions

Related projects

WordNet documentation

WordNet

WordNet® is a large lexical database of English, developed under the direction of George A. Miller (Emeritus). Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

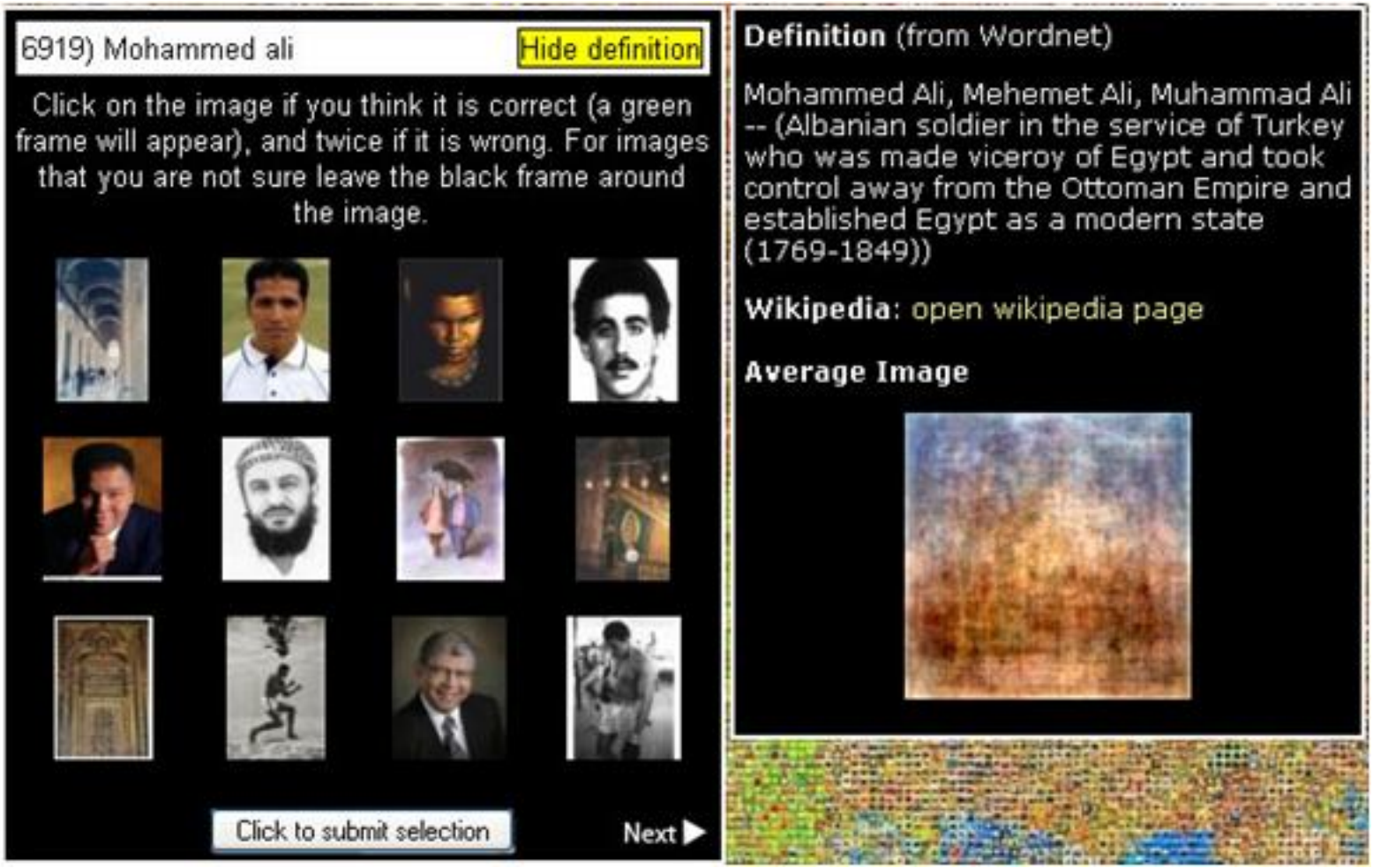
Over the years, many people have contributed to the development of WordNet. Currently, the WordNet team includes the following members, and the WordNet project is housed in the Department of Computer Science:

We appreciate your comments and suggestions, especially when they are constructive and help us improve WordNet. Please contact us at [email].

Our staff examines all mail and tries to make appropriate changes, but we hope you understand that due to time constraints we cannot always respond to the sender.

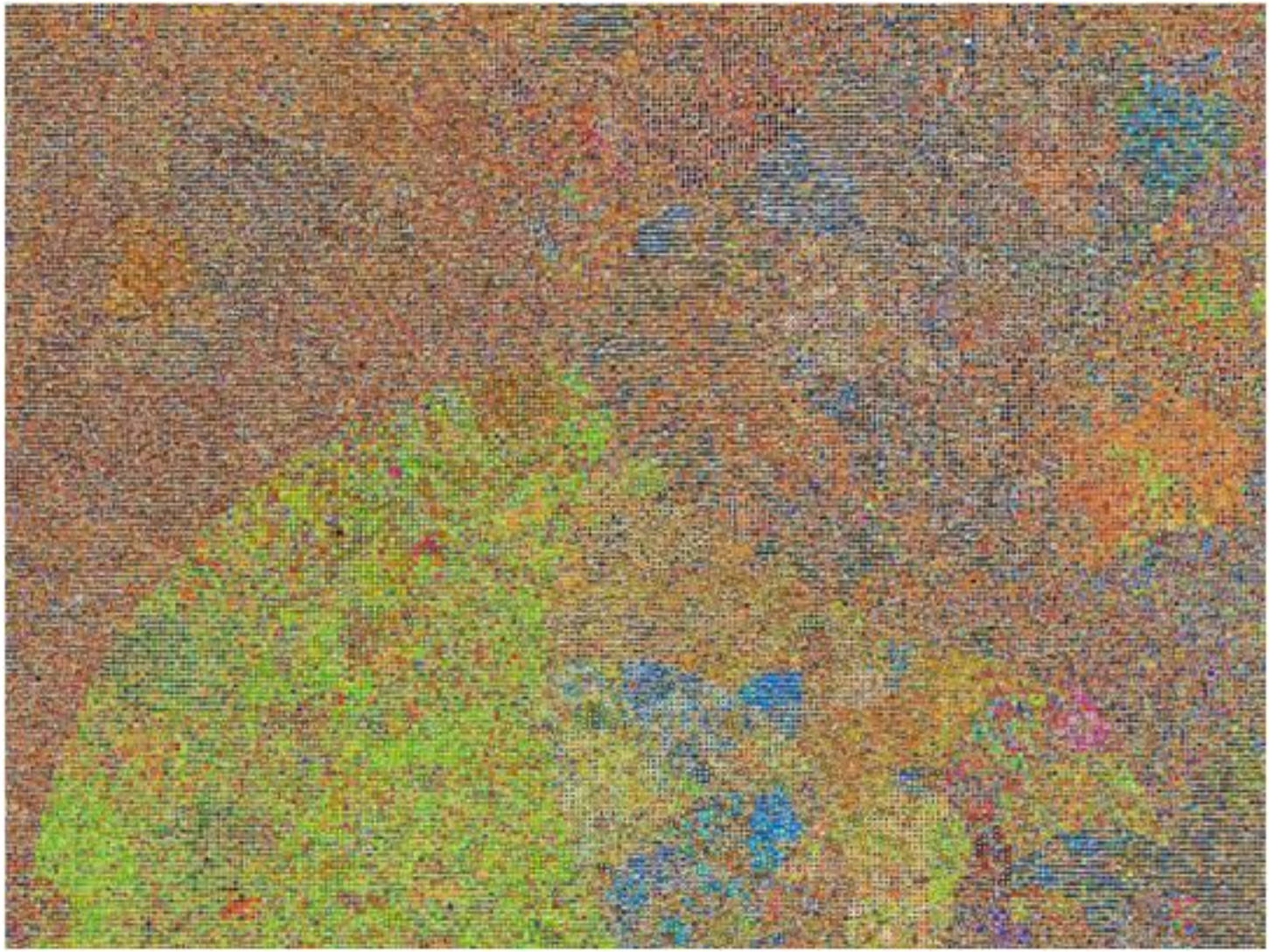
Please note that changes made to the database are not reflected until a new version

80 миллионов изображений



http://people.csail.mit.edu/torralba/tinyimages/

80 миллионов изображений



http://people.csail.mit.edu/torralba/tinyimages/

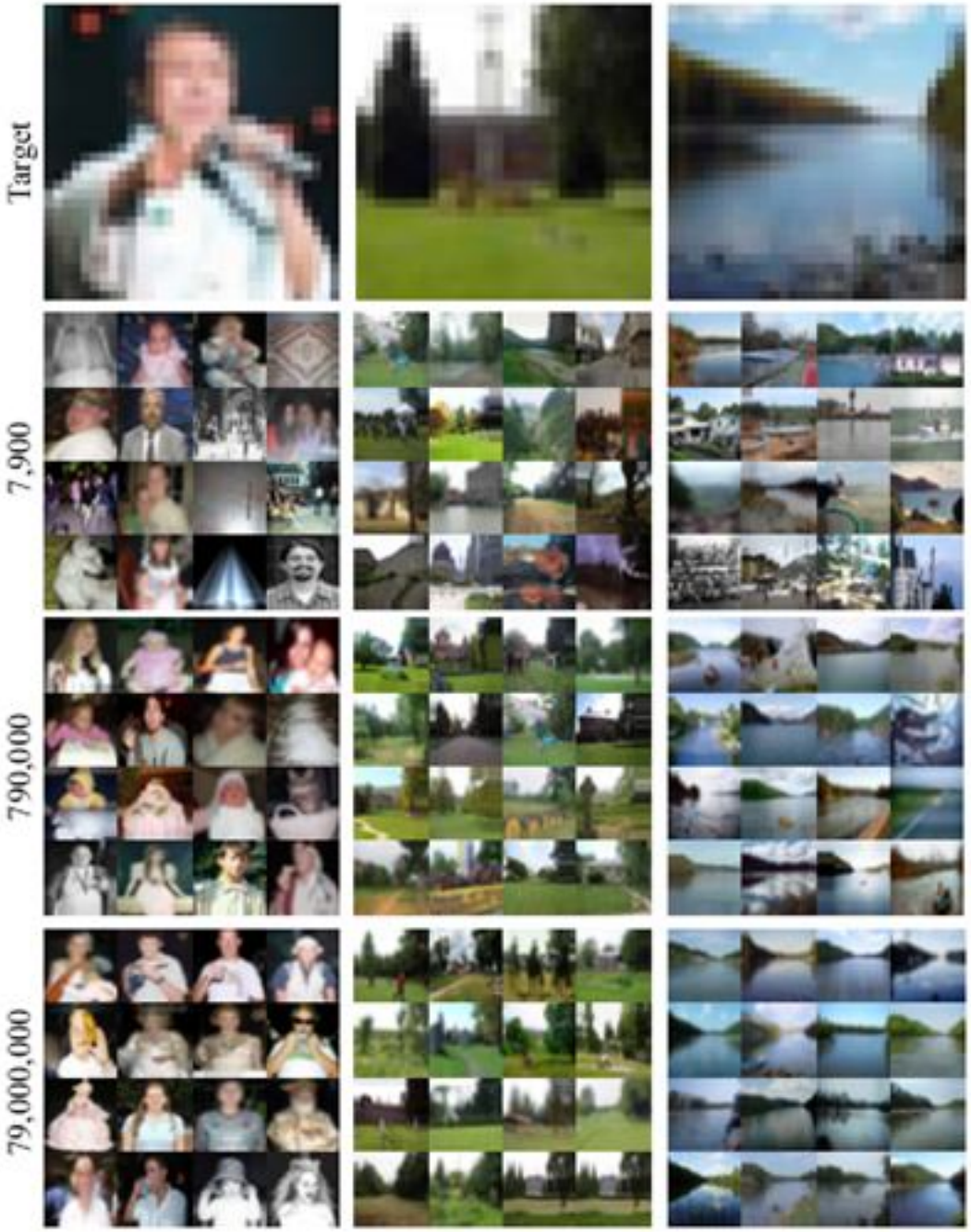
Метрики для поиска

$$D^2_{SSD} = \sum_{x,y,c} (I_1(x,y,c) - I_2(x,y,c))^2$$

SSD-метрика

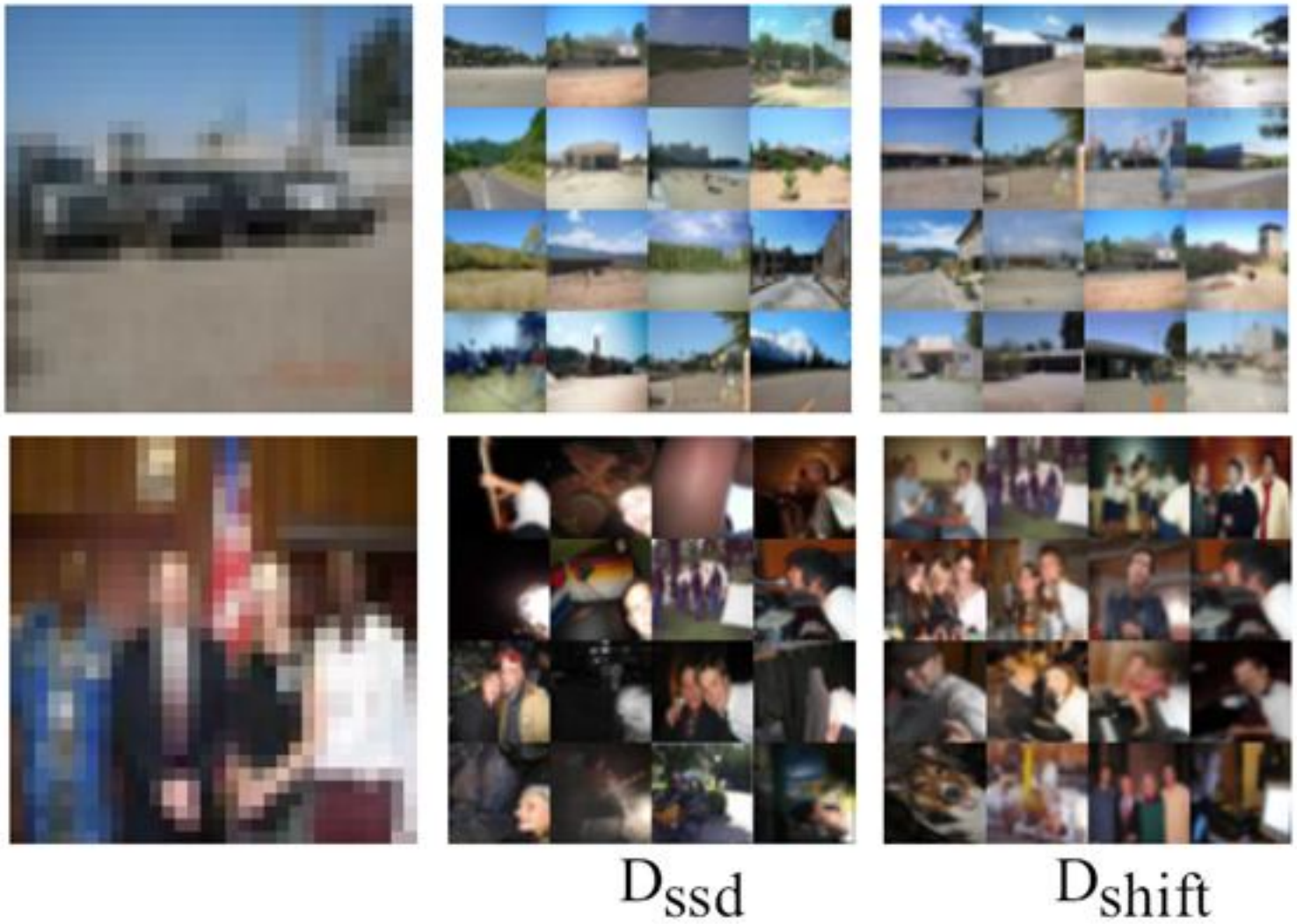
$$D^2_{warp} = \min_{\theta} \sum_{x,y,c} (I_1(x,y,c) - T_\theta I_2(x,y,c))^2$$

Сравнение с искажением



Наиболее похожие по SSD в зависимости от размеров коллекции

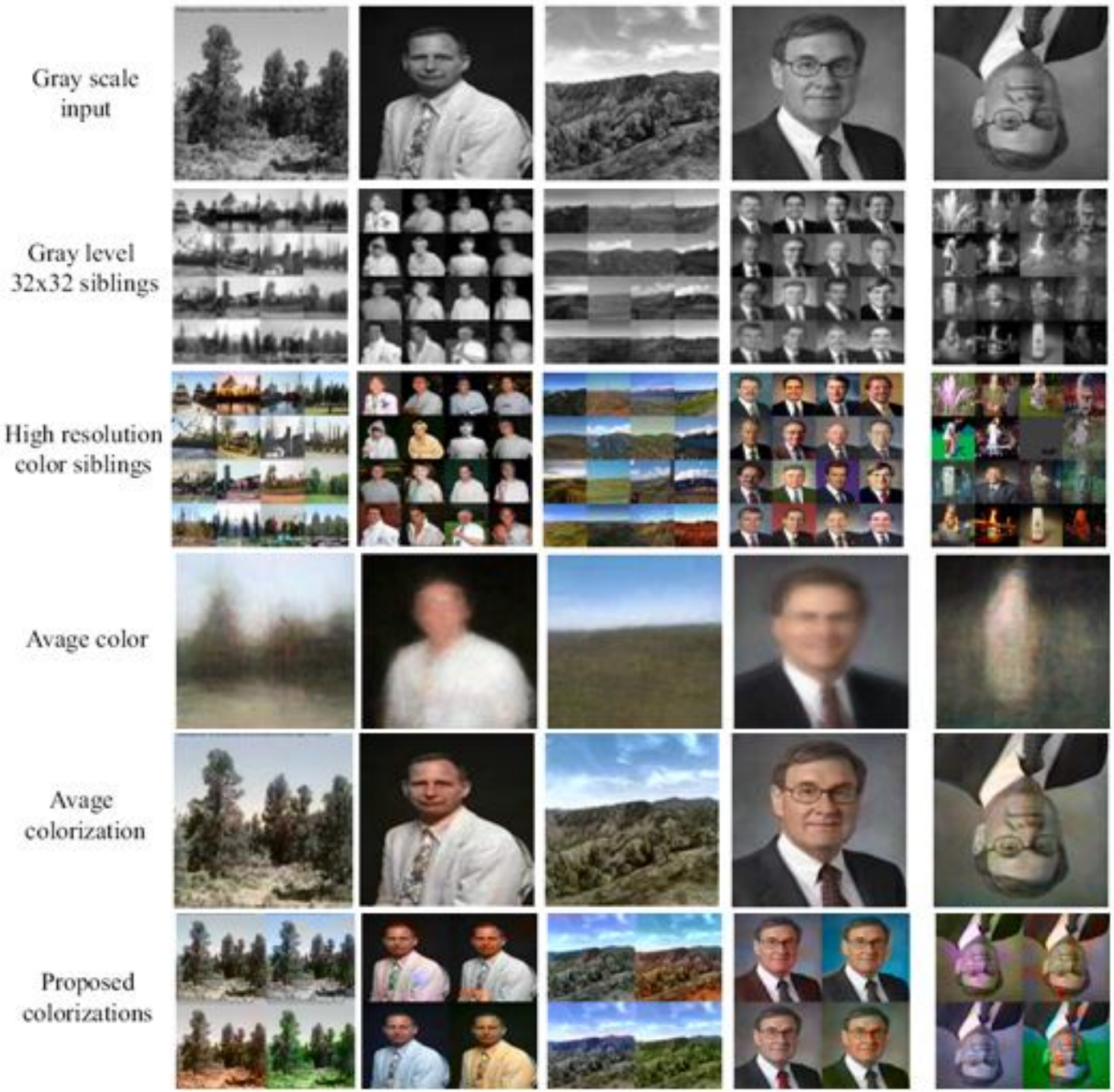
Результат улучшенной метрики



Результат улучшенной метрики

Поиск лиц на основе коллекции


25
27
20


Gray scale input
Gray level 32x32 siblings
High resolution color siblings
Avage color
Avage colorization
Proposed colorizations

Выводы из крошек-картинок

- Человек может распознать сцену с 90% правильностью по цветному изображению 32x32
- Если есть достаточно большая выборка, то мы можем для практически любого изображения найти очень близкое и по содержанию, и по ракурсу, и т.д..
- Метод «ближайшего соседа» при больших выборках может неплохо работать

Расширенная аннотация

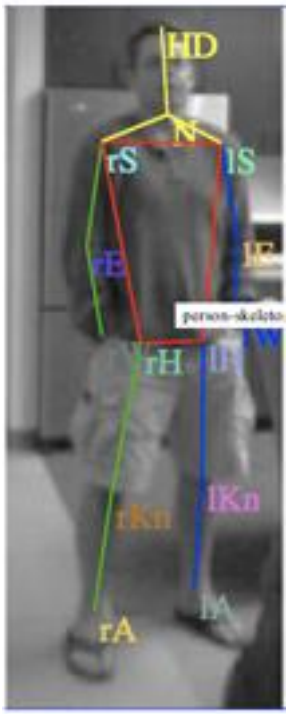
- Хотелось бы больше информации про изображения в коллекции
  - GPS-метка и атлас уже дали интересные результаты
- Нужна дополнительная аннотация
- Как бы нам её получить?

Mechanical Turk (1770)

- Automaton Chess Player – робот, игравший в шахматы
  - Автоматон двигает фигуры, говорит «Чек» и обыгрывает всех!
- С 1770 по 1854 развлекал публику, только в 1820 году раскрыли обман

Human Intelligence Task

- Есть много задач, простых для человека, но крайне сложных для компьютера
- Пример: классификация и разметка изображений



Galaxy Zoo

GALAXY ZOO http://www.galaxyzoo.org/

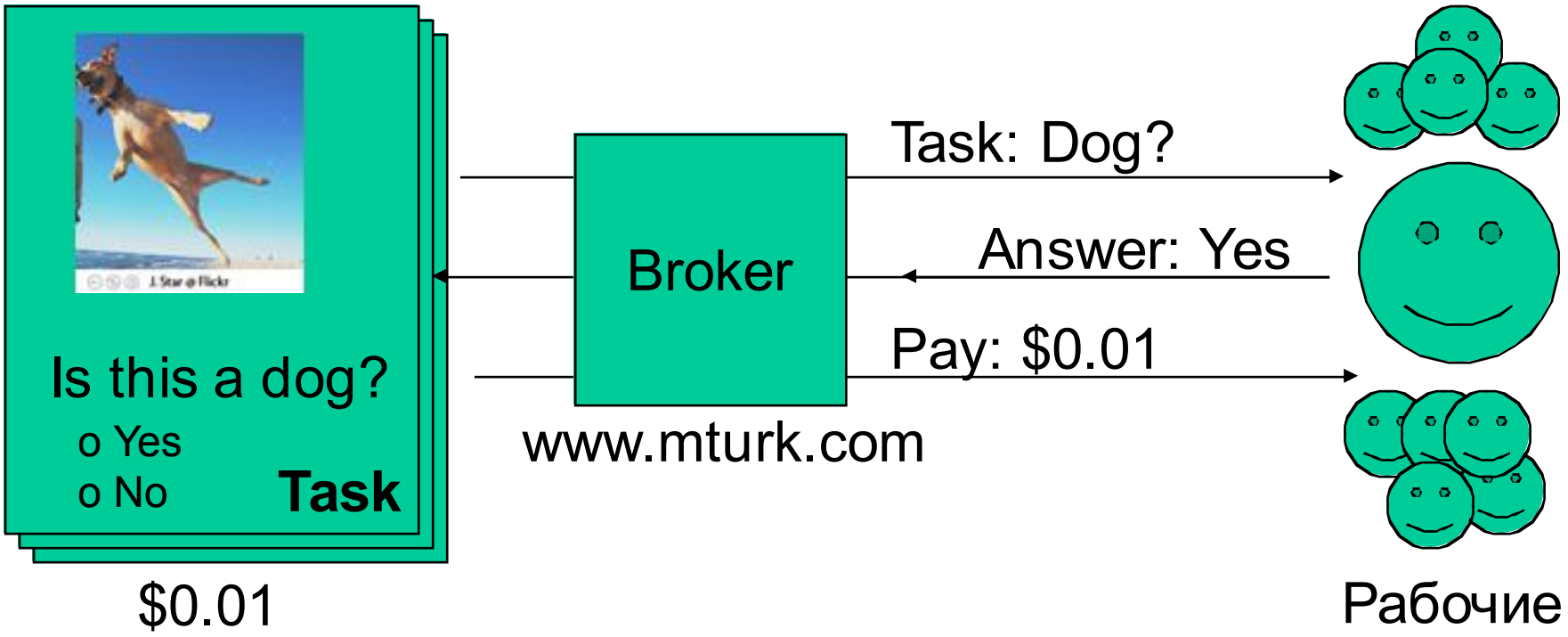
- Классификация изображений галактик
- Первый масштабный проект такого рода
- Более 150000 волонтёров за первый год бесплатно сделали более 60 млн. меток

Система LabelMe

- Просим пользователей выделять любые объекты замкнутыми ломаными и давать им название

B. Russell, A. Torralba, K. Murphy, W. T. Freeman LabelMe: a database and web-based tool for image annotation, IJCV, 2008

Фрилансеры



• Возникли интернет-брокеры для сведения заказчиков и работников
• Пример известного – Amazon Mechanical Turk
• Разметчики – просто подработка (студенты, домохозяйки), люди из более бедных стран

Резюме лекции

- Данные – один из краеугольных элементов для решения задачи компьютерного зрения
- Нужно уметь собирать большие массивы данных и быстро искать в них
  - GIST, квантование, семантическое хэширование, инвертированный индекс
- Интернет и глобализация дают нам удобный инструмент для сбора и обработки данных