

Сравнение изображений

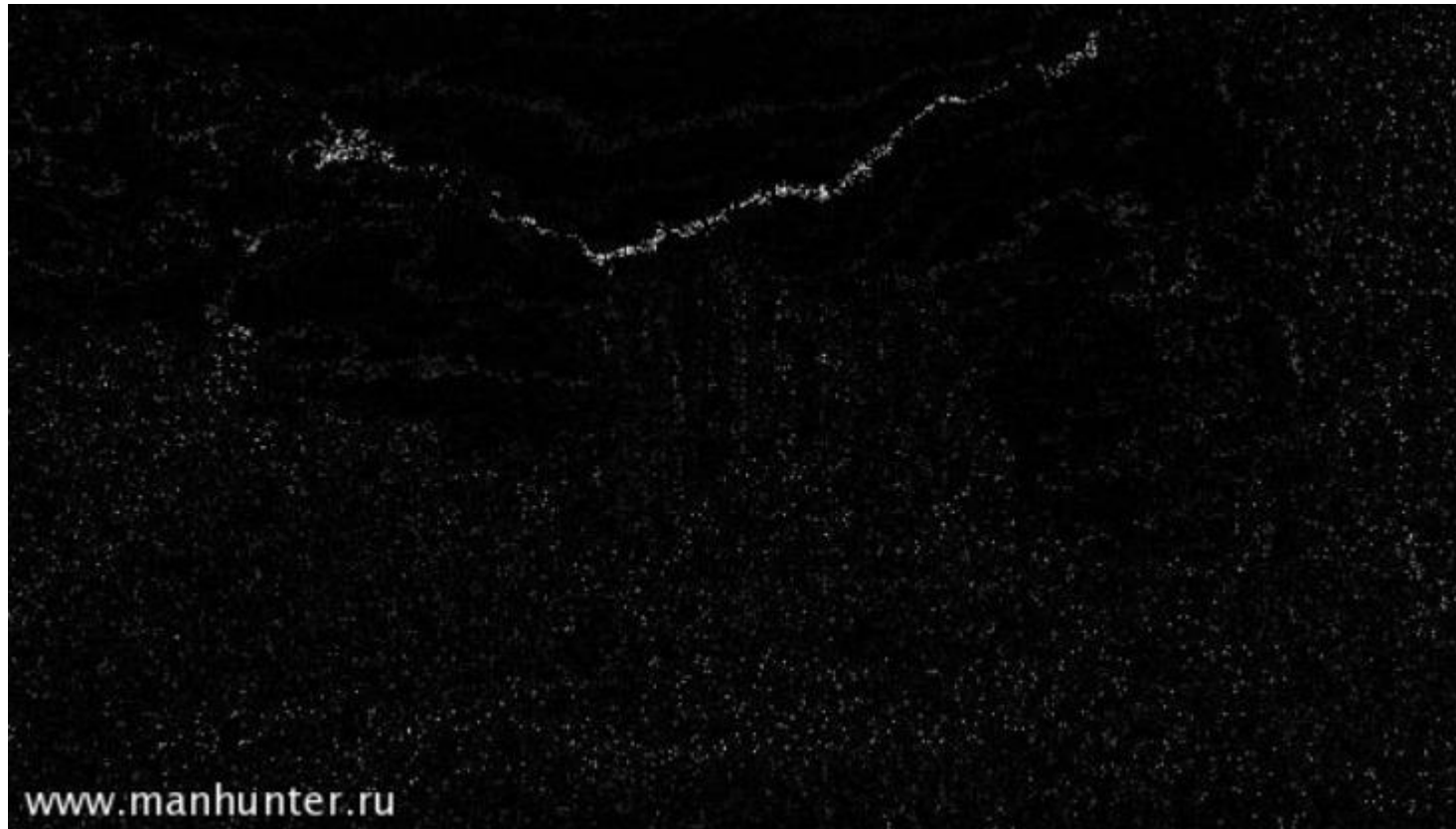


Зачем сравнивать
изображения



Поиск дубликатов
Поисковые запросы
Классификация

Попиксельное сравнение

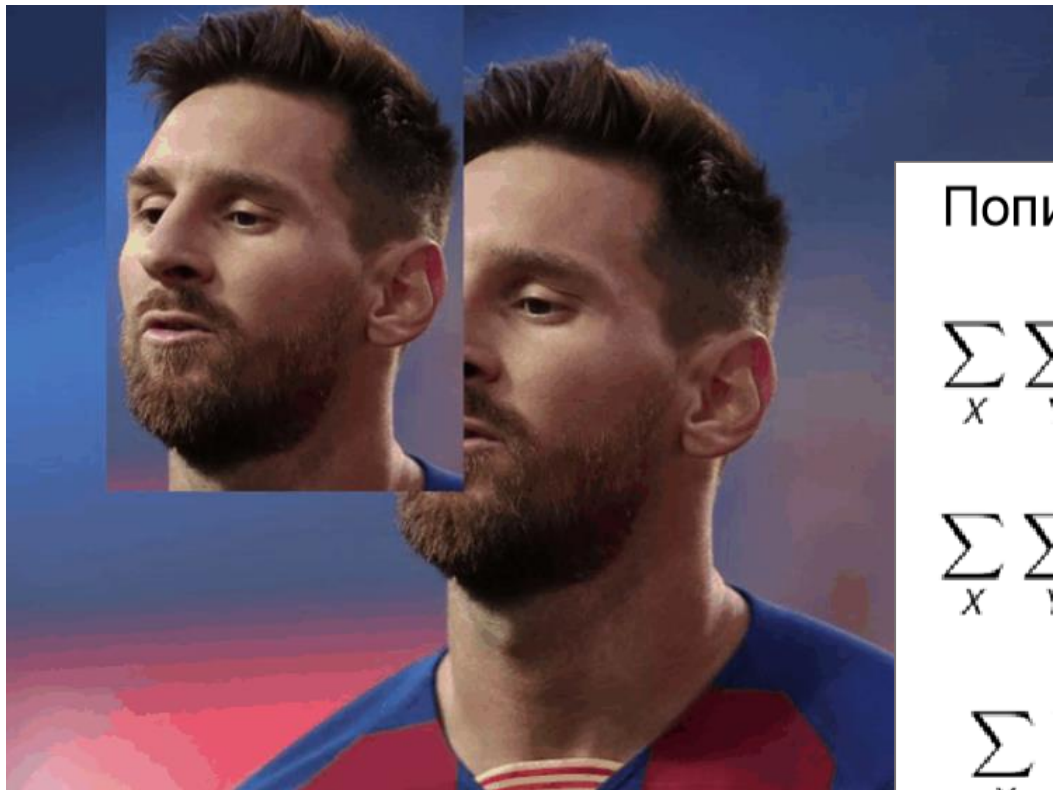


Вот так примерно выглядит изображение, полученное в результате дизъюнкции двух файлов с идентичными изображениями, но разным коэффициентом JPEG-сжатия.

Пример одного из методов:

Цвет каждой точки изображения можно представить в виде числа. Если к двум числам применить логическую операцию XOR, то совпадающие единичные биты будут обнулены. В результате, чем точнее совпадали исходные числа, тем ближе к нулю будет результат дизъюнкции. Если перенести это на нашу задачу, то чем точнее совпадают цвета на двух картинках, тем ближе к черному цвету получится результат XOR. Для большей точности сравнения исходные изображения желательно перевести в оттенки серого.

Сравнение по шаблону



Для оценки совпадения шаблона и изображения используют различные метрики



Попиксельное сравнение изображений:

$$\sum_X \sum_Y |I_1(X, Y) - I_2(X, Y)|$$
 L1 метрика (SAD - Sum of absolute differences)

$$\sum_X \sum_Y (I_1(X, Y) - I_2(X, Y))^2$$
 L2 метрика (SSD - Sum of squared differences)

$$\sum_X \sum_Y I_1(X, Y) I_2(X, Y)$$
 Кросс-корреляция (CC - Cross-correlation)

- SAD, SSD – минимизируются (0 – точное совпадение)
- CC – максимизируется (1 – точное совпадение)

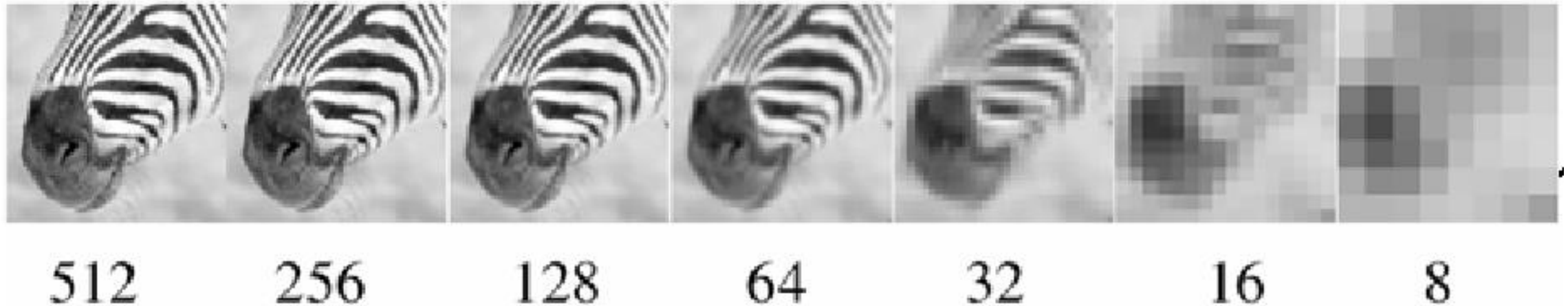
Метод неинвариантен к:

- масштабированию
- освещению
- повороту



Многомасштабное сопоставление

Вариант полного перебора параметров можно улучшить с помощью многомасштабного сопоставления.



- Строится пирамида для каждого из 2х изображений
- Ищется совпадение на самом низком уровне
- Используется как начальное приближение для уточнения на следующих уровнях

как строится пирамида •Сглаживаем с помощью фильтра Гаусса текущее изображение •Сэмплируем – берём каждый k-ый пиксель (обычно, каждый 2ой)



Как можно изображение описать набором как можно меньшего количества численных характеристик, т.е. в виде числового вектора?

Извлечение признаков (Feature Detection)

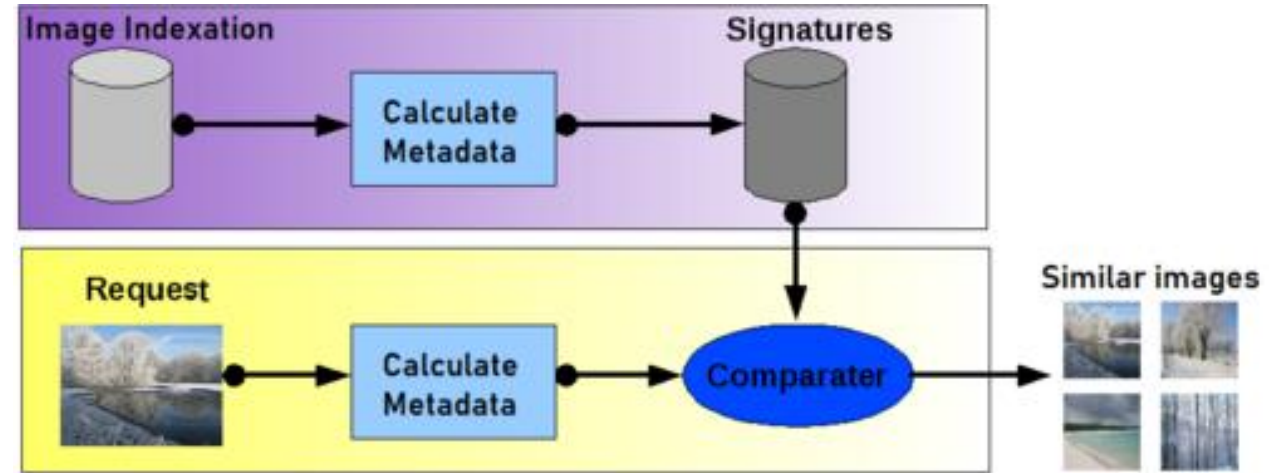
Выделяют **глобальные** признаки, которые рассчитываются по всему изображению, и **локальными** признаки, которые рассчитываются вокруг особых точек.

Можно также выделить **полулокальные** признаки, когда они извлекаются в ограниченных областях изображения в результате сегментации изображения по областям или просто в соответствии с произвольной сеткой.

Алгоритм поиска изображений по содержанию

Этапе индексации состоит из извлечения визуальных характеристик изображений, таких как:

- **цвет** (цветовая гистограмма и т. д.),
- **текстура** (фильтр Габора , вейвлеты и т. д.)
- **формы** (дескрипторы Фурье и т. д.),
- **сочетание нескольких из этих характеристик.**



Эти характеристики называются низкоуровневыми, потому что они очень близки к сигналу и не передают какой-либо конкретной семантики изображения.

Как только эти характеристики были извлечены, сравнение обычно состоит в определении различных расстояний между этими характеристиками и в определении общей меры сходства между двумя изображениями. Посредством этой меры сходства и изображения запроса можно затем вычислить набор мер сходства между этим изображением запроса и набором изображений основы изображения. Затем можно упорядочить изображения базы в соответствии с их оценкой и представить результат пользователю, при этом изображения с наивысшей оценкой считаются наиболее похожими.

Глобальные признаки

Цвет

Текстура

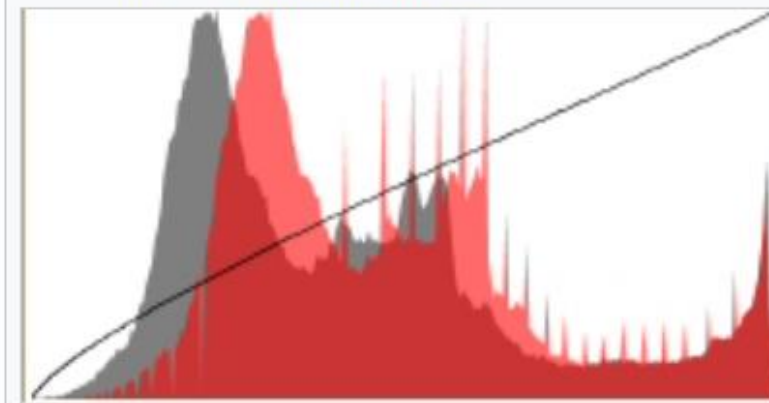
Форма

Учет цвета изображений исторически был одной из первых характеристик, используемых для исследования изображений на основе содержания.

- ✓ **Гистограмма цветов** представляет собой общее распределение цветов на изображении
- ✓ **Моменты цвета.** Обычно ограничиваются первыми двумя или тремя моментами (среднее значение , дисперсия , асимметрия или даже момент 4-го порядка, называемый эксцессом), они вычисляются в каждой из трех плоскостей выбранного цветового пространства. Одно из возможных усовершенствований - разделить изображение по сетке и вычислить эти моменты в каждой ячейке.



Изображение подсолнуха



И гистограммы трех компонентов

Цвет. Пример метода на основе гистограммы распределения цветов

В 1991 году Суэйн и Баллард предложили использовать цветную гистограмму характеристики изображений. Например, если квантуем каждую цветовую плоскость по 4 ячейкам, гистограмма цвета на 64 бита может быть представлена «кубом» :

		красный															
		0-63				64-127				128-191				192-255			
		Зеленый				Зеленый				Зеленый				Зеленый			
		0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255
Синий	0-63		Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс
	64-127	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс
	128-191	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс
	192-255	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс	Икс

Каждый цвет представляет один из 64 квантованных интервалов, а x - количество пикселей изображения, соответствующего этому интервалу. Такой дескриптор обычно представлен 64-мерным одномерным вектором.

Глобальные признаки

Цвет

Текстура

Форма

Многие дескрипторы могут быть получены из представления гистограммы, некоторые из которых были включены в экспериментальную модель стандарта MPEG-7 . Это включает:

- ✓ **доминирующий цвет** : сообщает о доминирующем цвете изображения, рассчитанном как цвет ячейки с наивысшим значением. В теории цветовое пространство может быть любым, но использование HSV дает некоторые интересные результаты.
- ✓ **масштабируемый цвет** : это преобразование Хаара цветовой гистограммы в пространстве HSV , что приводит к очень компактному коду.
- ✓ **цветовая структура** : это «локальная» цветовая гистограмма. Его конструкция идентична принципу, представленному ранее (гистограмма глобального цвета), за исключением того, что окно уменьшенного размера (8×8 пикселей) проходит через изображение при подсчете цветов в каждом пикселе. Таким образом, он отражает некоторую информацию о пространственной структуре распределения цвета.
- ✓ **цветовая схема** : вычисляется дискретное косинусное преобразование представления гистограммы, и только часть коэффициентов может использоваться для подписи. Это дает возможность получить относительно грубый, но очень компактный дескриптор или более тонкий дескриптор, но занимающий больше места. Такой дескриптор интересен для приложений, требующих определенной масштабируемости.

Глобальные признаки

Цвет

Текстура

Форма

✓ Стохастический подход

Текстуру можно описать статистически, учитывая, как уровни серого изображения организуются по отношению друг к другу.

✓ Структурный подход

Этот подход предполагает наличие фундаментальных примитивов для описания текстур. Одним из первых примеров таких примитивов были *Текстоны*, предложенные Джулесом. Данная текстура описывается как сложная комбинация этих примитивов, выраженная, например, с помощью графиков. Эти графики затем могут служить дескрипторами текстуры.

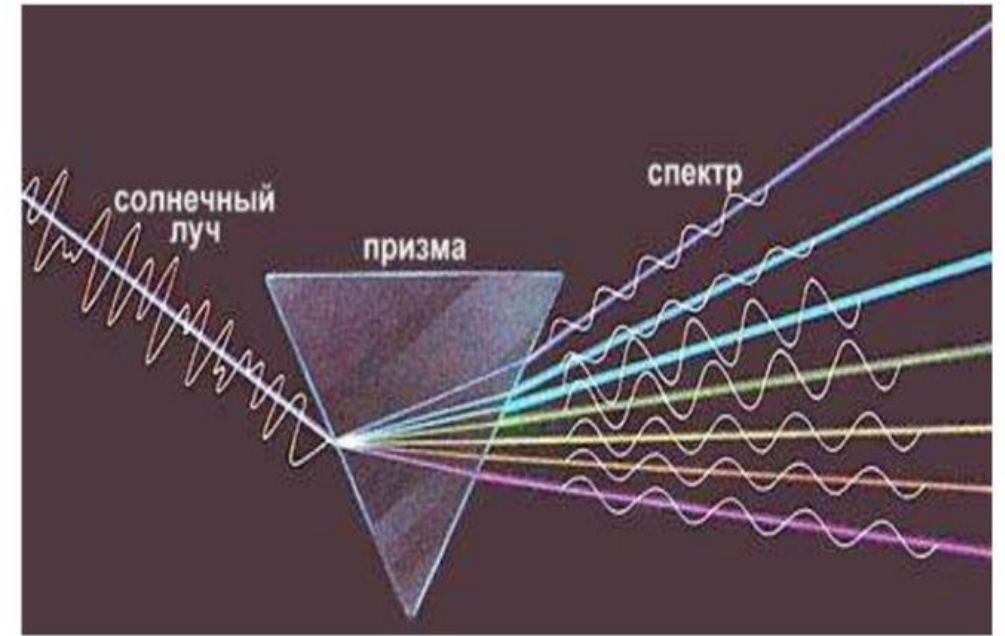
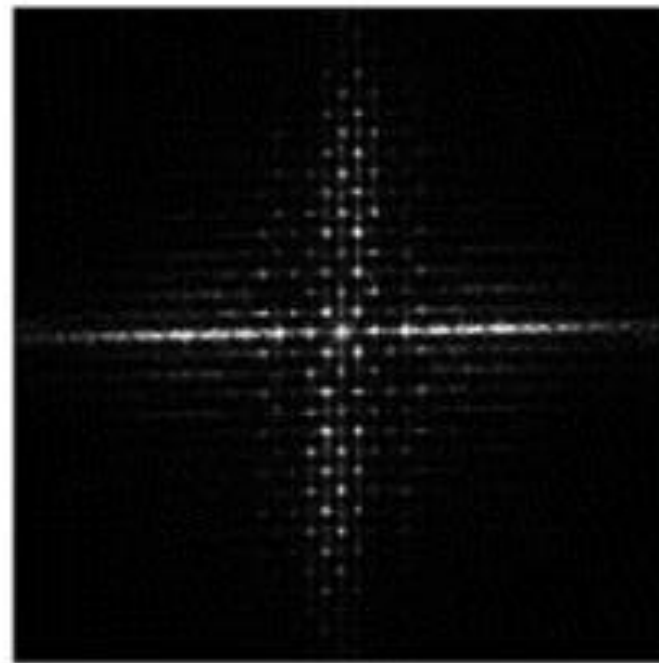
✓ Спектральный подход

Выражение периодичностей и других закономерностей в изображении или в сигнале естественно происходит в рамках спектрального анализа. Таким образом, дискретное преобразование Фурье изображения может быть основой для создания дескрипторов. Однако чаще используется набор фильтров Габора, расположенных в нескольких масштабах и ориентациях.

Альтернативой преобразованию Фурье является вычисление дискретного косинусного преобразования (DCT). На практике дескрипторы, основанные на коэффициентах DCT, позволили различать внутренние и внешние изображения, изображения городских ландшафтов и естественные ландшафты и, в сочетании с другими дескрипторами, одновременно несколько категорий природных сцен.

Выявление текстур
спектральным методом

Разложение в ряд Фурье



Например, радуга – это
разложение света в
ряд Фурье.

Общие характеристики низкого уровня

Цвет

Текстура

Форма

Описание форм требует предварительного определения регионов. Это может быть результатом сегментации изображений или обнаружения их контуров. Затем можно охарактеризовать регионы с помощью различных показателей, таких как их основная ориентация (например, их первая собственная ось), их симметрия или их моменты.

В частности, используется метод описания, предложенный Мохтарианом, который называется Пространство шкалы кривизны. Он заключается в описании области как функции изменения кривизны ее контура.

Этот метод инвариантен к вращению и сдвигу. Что касается масштабирования, то необходимо стандартизовать хотя бы по длине контура. Что еще более важно, дескриптор обычно вычисляется в разных масштабах, каждый из которых соответствует свертке по гауссовскому ядру.

Задача сопоставления изображений

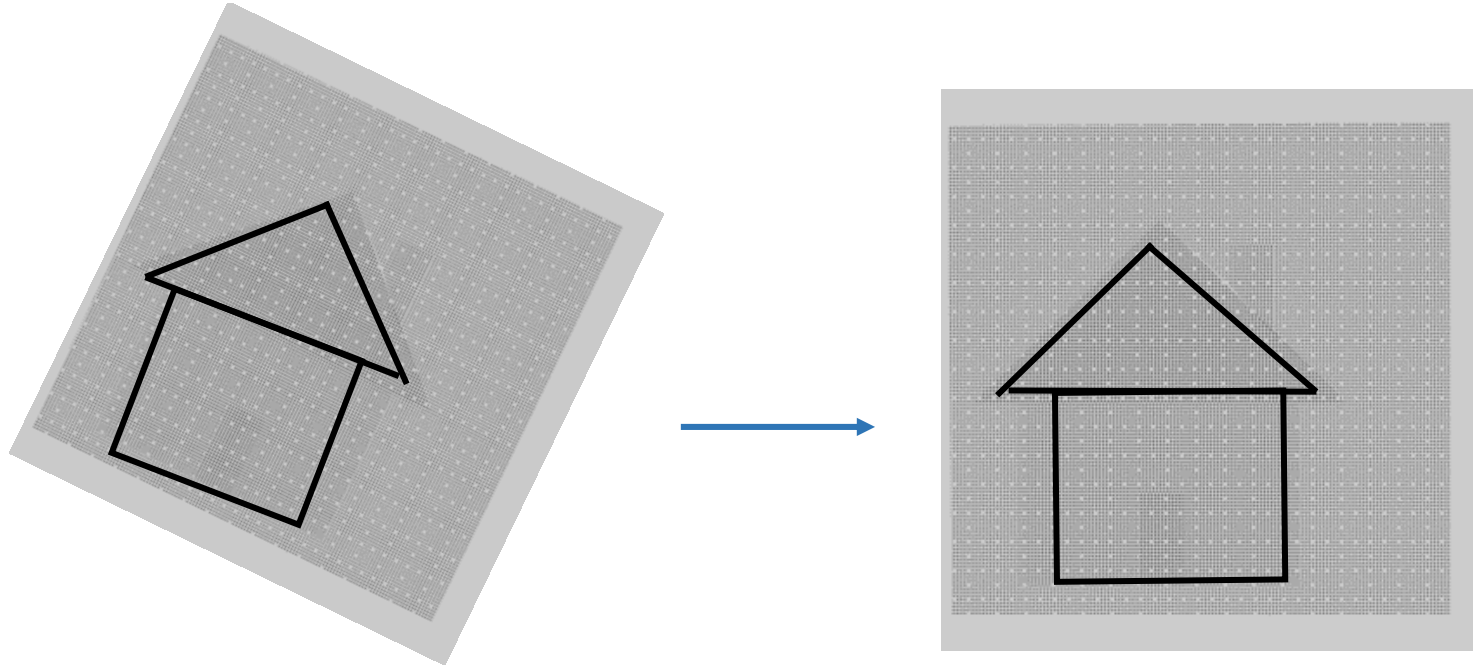


Когда вообще решается **задача сопоставления изображений**.

Например:

- создание панорамных снимков,
- создание стереопары и реконструкция трехмерной модели объекта по его двумерным проекциям,
- распознавание объектов и поиск по образцу из какой-то базы,
- слежение за движением объекта по нескольким снимкам,
- реконструкция аффинных преобразований изображений.

Выравнивание изображений



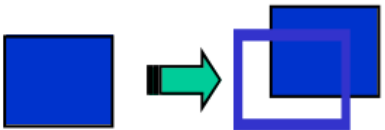

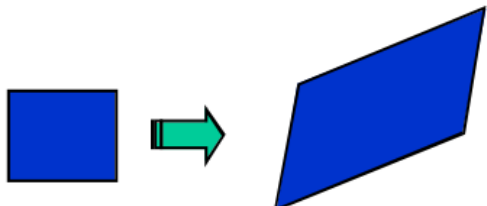
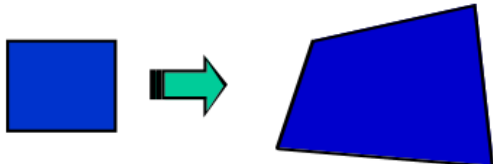
Есть два изображения
одного и того же
объекта. Как нам
совместить
изображения
автоматически?

Найдём такое преобразование (совмещение изображений), при котором изображения больше всего совпадут

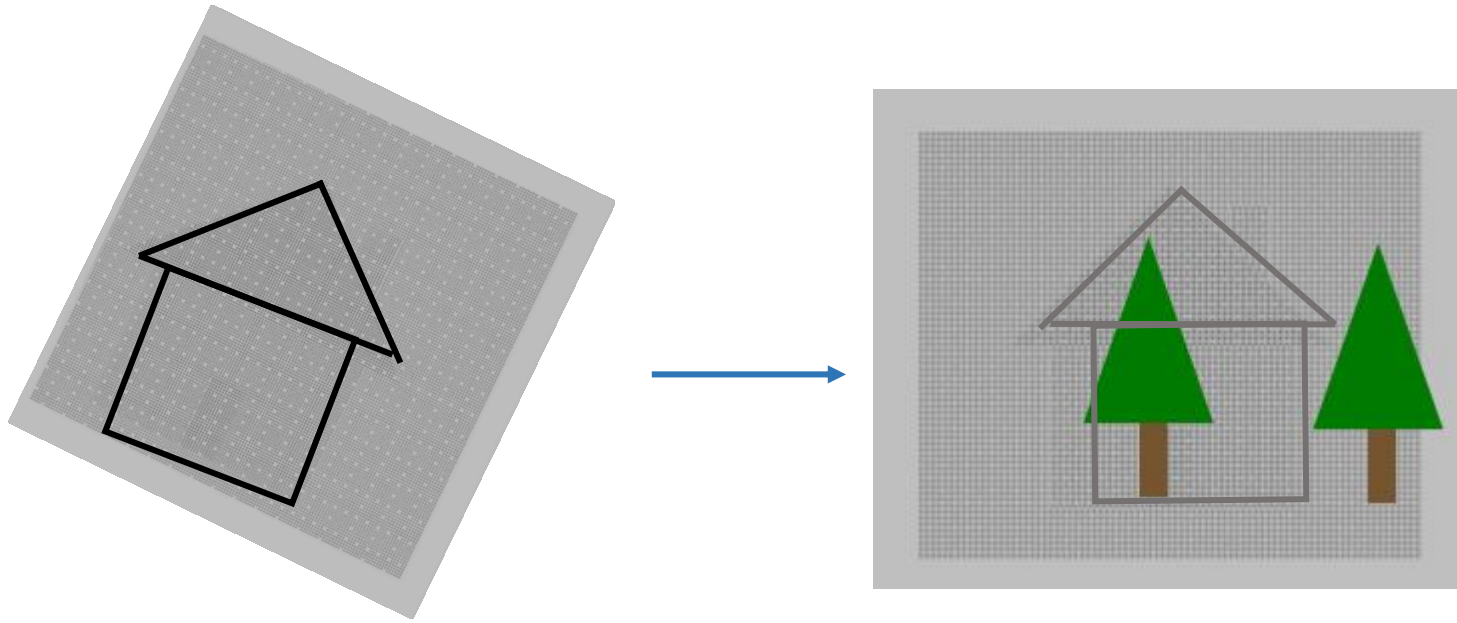
Что нужно определить:

- Какое преобразование будем использовать?
- Как оценить совпадение (похожесть изображений)?

Геометрические преобразования

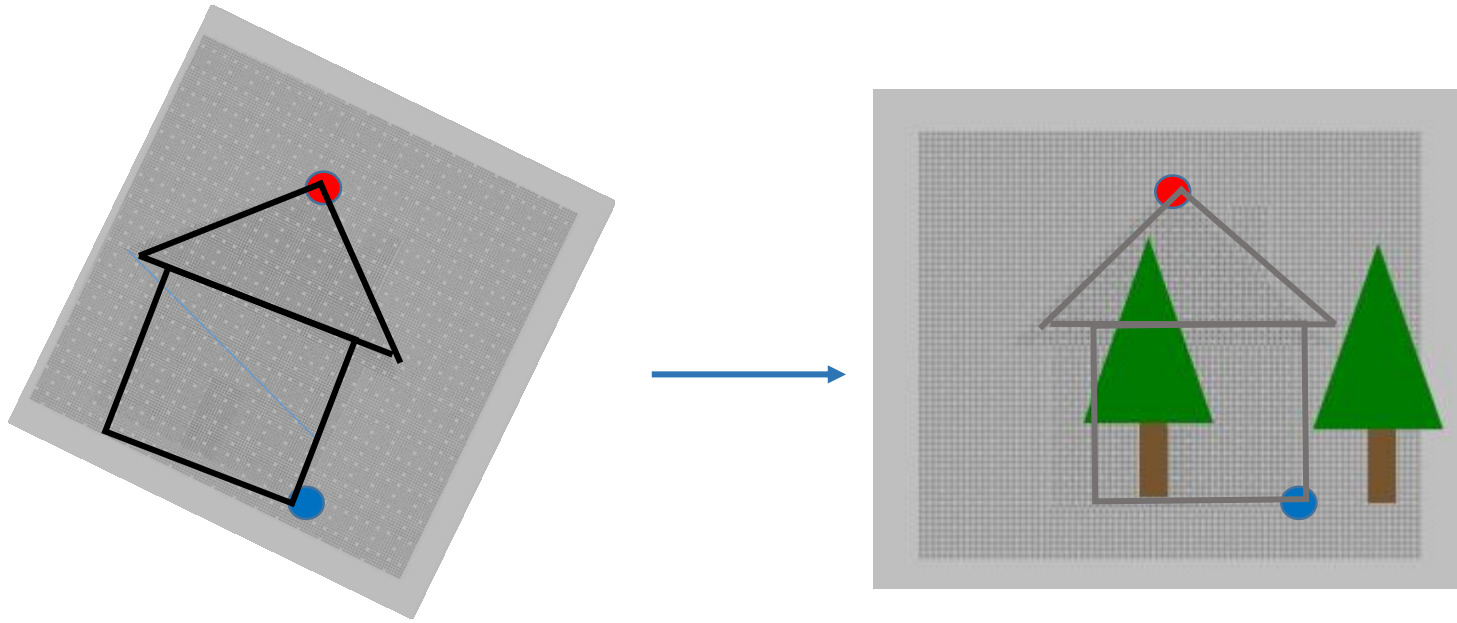
- Параллельный перенос 
- Подобие (перенос, масштаб, поворот) 
- Аффинное 
- Проективное (гомография) 

Перекрывание объекта



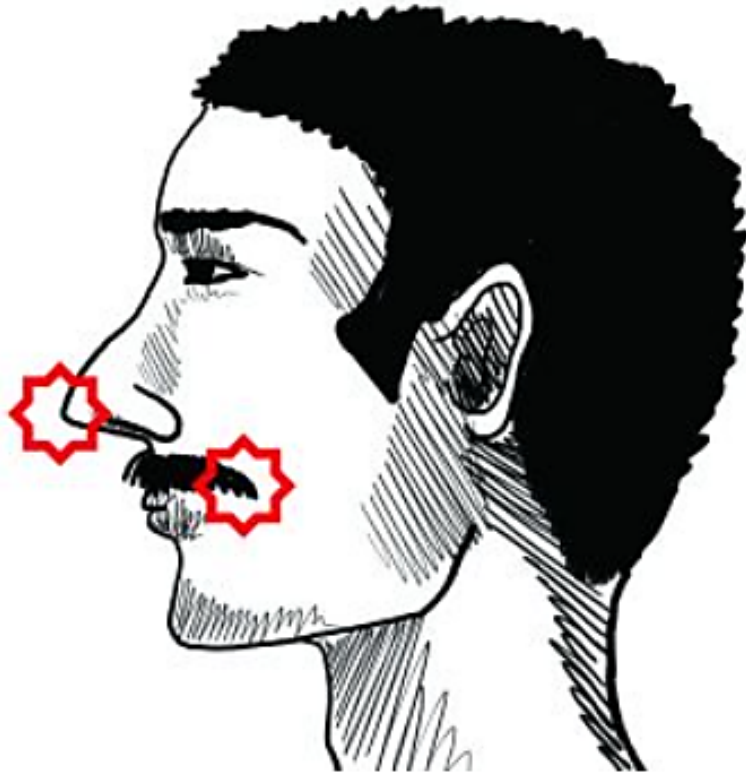
- Как быть, если объект («дом») частично перекрыт другими объектами («ёлками»)
- Положение «ёлок» относительно дома разное на разных ракурсах
- **Прямое попиксельное сопоставление изображений не даст хорошего результата**

Локальные особенности



- Найти хорошо различимые точки на изображениях (**«особенности» «локальный особые точки» «характеристические точки»**)
- Определить, какой точке на одном изображении соответствуют какая точка на втором изображении
- Найти такое преобразование, которое совмещает найденные точки

Локальные особенности



features – особые точки

Особые точки (в разных источниках – *features/characteristic points/local feature points/interest point/локальные особенности*) – говоря неформально – “хорошо различимые” фрагменты изображения. Это точки (пиксели) с характерной (особой) окрестностью – т.е. отличающиеся своей окрестностью от всех соседних точек. Классический пример локальной особенности – вершина угла (а не какая-то произвольная точка на прямой или на однородном фоне). Описываются вектором признаков вычисляемых на основе интенсивности/градиентов или других характеристик точек окрестности. Используя особые точки можно анализировать как изображения целиком так и объекты на них. Хорошие характерные точки

позволяют справиться с изменением масштаба, ракурса и перекрытиями сцены или объекта.

Детекторы локальных особенностей

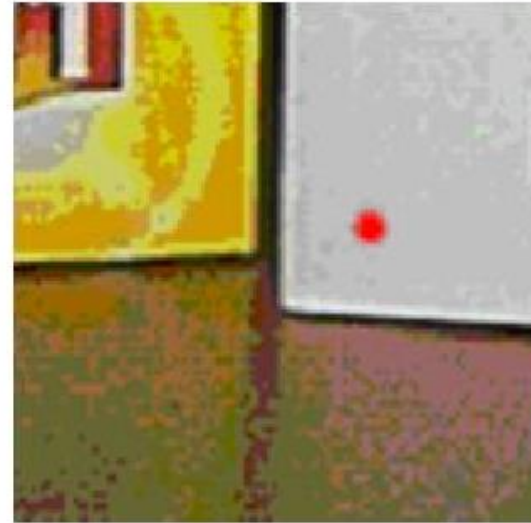
- Детекторы углов
- Детектор Моравеца
- Детектор Харриса
- Алгоритм FAST
- Блобы
- Детектор на основе DoG (Difference of Gaussian)
- Детектор Harris-Laplacian
- Детектор областей IBR
- Детектор MSER

Дескрипторы

Локальные особенности



Пример особой
точки



Пример точки, не
являющейся особой

Локальная (особая) точка изображения - это точка с характерной (особой) окрестностью, т.е. отличающаяся от всех других точек в некоторой окрестности.

Процесс определения достигается путём использования детектора и дескриптора.

Детектор – это метод извлечения особых точек из изображения.

Дескриптор – идентификатор особой точки, выделяющий её из остального множества особых точек.

Требования к особенностям

Повторимость (Repeatability)

- Особенность находится в том же месте сцены не смотря на изменения точки обзора и освещения

Локальность (Locality)

- Особенность занимает маленькую область изображения, поэтому работа с ней нечувствительна к перекрытиям

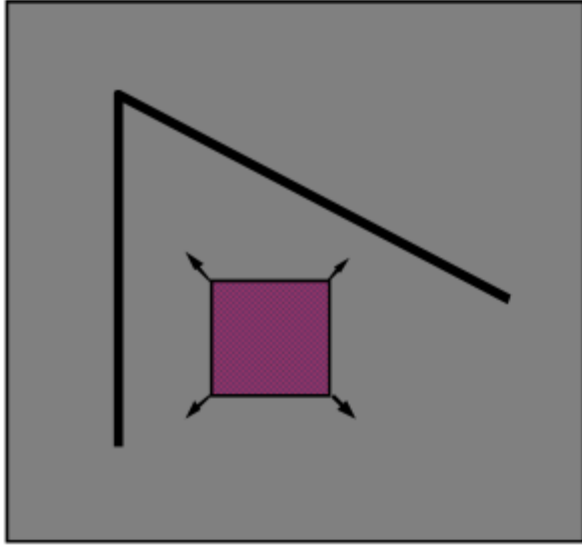
Значимость (Saliency)

- Каждая особенность имеет уникальное описание

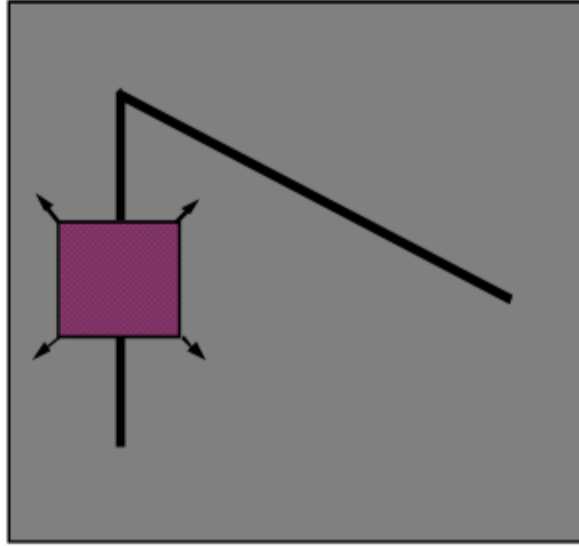
Компактность и эффективность

- Количество особенностей существенно меньше числа пикселей изображения

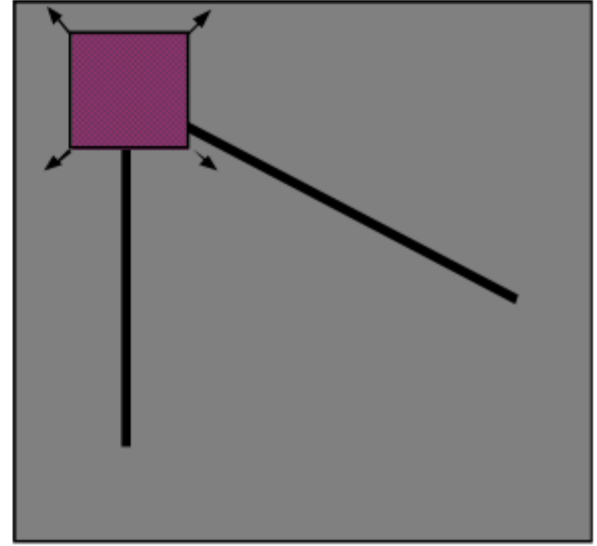
Локальные особенности



монотонный регион:
в любом направлении
изменений нет



«край»:
вдоль края
изменений нет



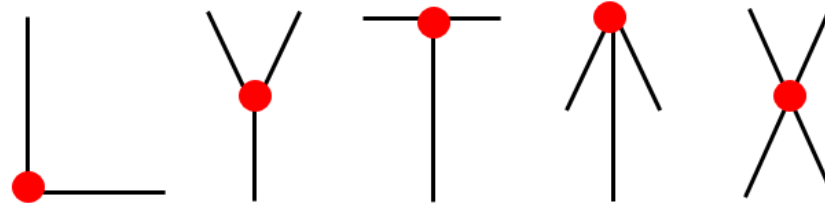
«уголок»:
изменения при
перемещении
в любую сторону

Углы

Углы (corners) – особые точки, которые формируются из двух или более граней. Или **угол** – это точка, у которой в окрестности интенсивность изменяется относительно центра (x, y) .

Главное свойство таких точек заключается в том, что в области вокруг угла у градиента изображения преобладают два доминирующих направления, что делает их различимыми.

В зависимости от количества пересекаемых граней существуют разные виды уголков на рис. Различные детекторы углов по-разному реагируют на каждый из таких видов уголков.



Существуют различные подходы к определению особых точек.

Это могут быть методы, основанные на:

- измерении интенсивности изображения
- поиске максимальной кривизны контуров
- определяющие пересечение контуров и т. д.

Детекторы углов

Работа с использованием особых точек началась с детектора Моравеца (Moravec, 1977). **Детектор Моравеца** – самый простой из существующих. Автор рассматривает изменение яркости квадратного окна W (обычно размера 3×3 , 5×5 , 7×7 пикселей) относительно интересующей точки при сдвиге окна W на 1 пиксель в 8-ми направлениях.

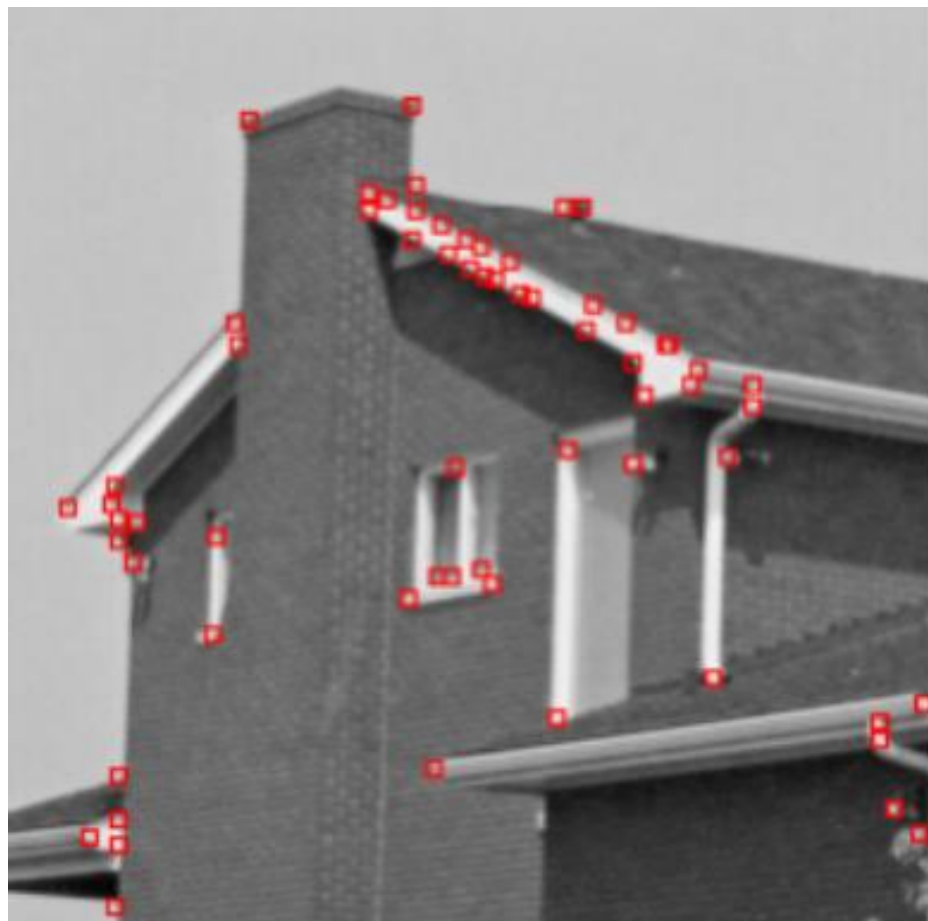
Харрис и Стефенс улучшили детектор Моравеца (Moravec)(1988), они рассмотрели производные яркости изображения для исследования изменений яркости по множеству направлений. **Детектор Харриса** является наиболее оптимальным детектором углов и широко применяется.

Детектор Харриса инвариантен к поворотам, частично инвариантен к аффинным изменениям интенсивности.

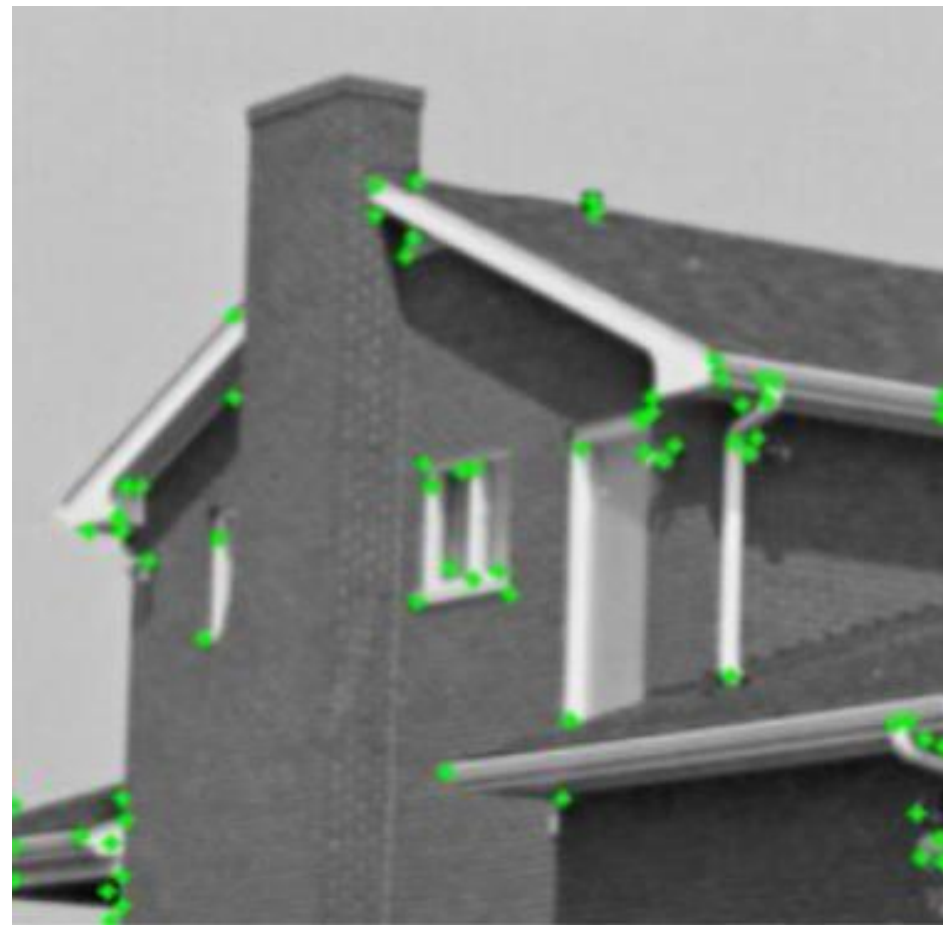

```
void cornerHarris (  
    InputArray src,    // входное одноканальное изображение  
    OutputArray dst,   // выходной вектор углов  
    int blockSize,     // размер окрестности  
    int ksize,         // параметр апертуры для оператора Собеля  
    double k,          // свободный параметр детектора Харриса  
    int borderType = BORDER_DEFAULT //метод экстраполяции  
);
```

```
int blockSize = 2;  
int apertureSize = 3;  
double k = 0.04;  
// определение углов  
Mat dst = Mat::zeros(src.size(), CV_32FC1);  
cornerHarris(src_gray, dst, blockSize, apertureSize, k);  
...
```

Результат работы детекторов



Детектор Моравеца



Детектор Харриса

Детектор Ши – Томаси

Функция **goodFeaturesToTrack()** реализует метод Харриса с усовершенствованиями Ши-Томаси. Она вычисляет необходимые операторы производных, анализирует их и возвращает список особых точек:

```
void goodFeaturesToTrack(  
Mat& image,                // входное изображение  
OutputArray corners,        // выходной вектор углов  
int maxCorners,             // сколько углов сохранять  
double qualityLevel,        // отношение к наилучшему углу  
double minDistance,         // минимальное расстояние между углами  
InputArray mask = noArray(), // игнорировать углы в точка, где  
mask=0  
int blockSize = 3,          // размер окрестности  
bool useHarrisDetector = false, // false°='метрика Ши-Томаси'  
double k = 0.04             // используется для метрики Харриса  
);
```

Количество возвращенных углов можно ограничить, задав аргумент **maxCorners**, а минимальное расстояние между соседними углами – с помощью **minDistance**. Аргумент **blockSize** задает размер окрестности при вычислении угла; типичное значение равно 3, но для изображений с высоким разрешением его можно немного увеличить.

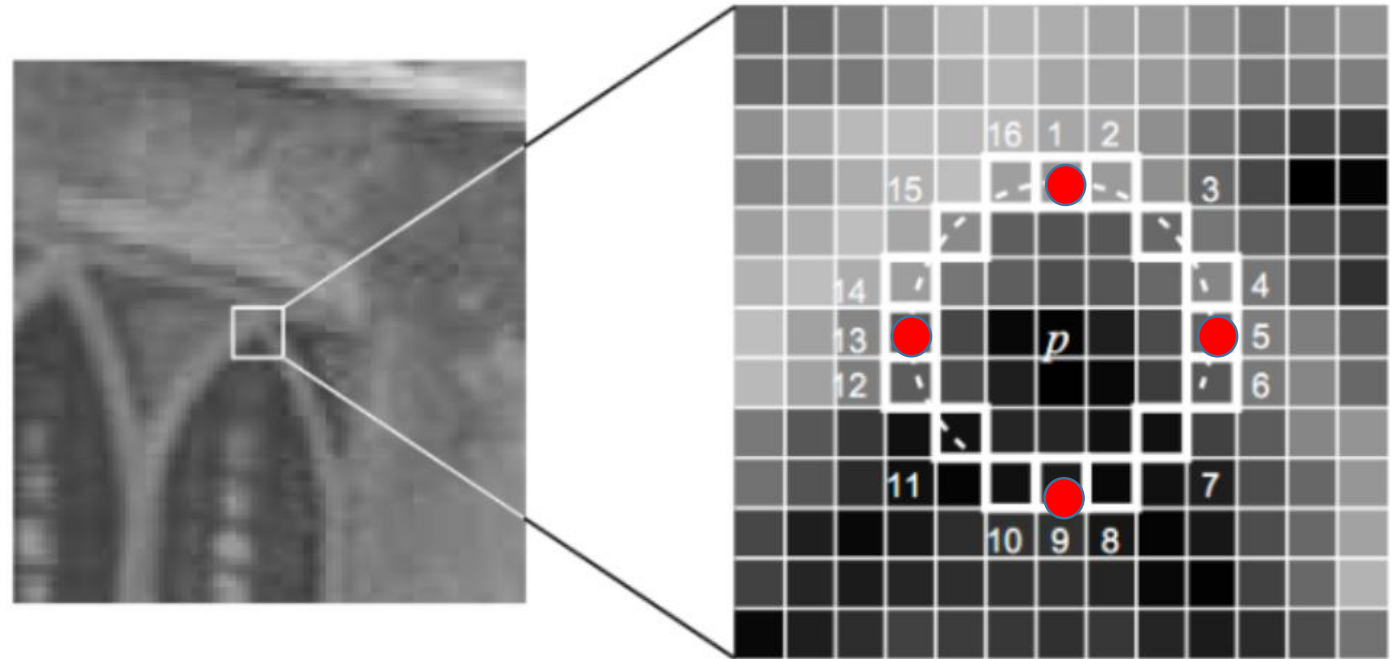
Алгоритм FAST

В середине 2000-х годов появились алгоритмы быстрого поиска особенных точек. Наиболее ярким представителем данного класса алгоритмов является алгоритм **FAST** (*features from accelerated segment test* — особенности, полученные из ускоренной проверки сегментов).

<https://habr.com/ru/post/244541/>

В алгоритме рассматривается окружность из 16 пикселей. Яркость пикселей, лежащих на окружности, сравнивается с яркостью центральной точки и на основании ряда проверок принимается решение, является ли центральная точка характерной.

Последовательность проверок и их общее число подбираются и оптимизированы заранее на основе обширной обучающей выборки изображений.

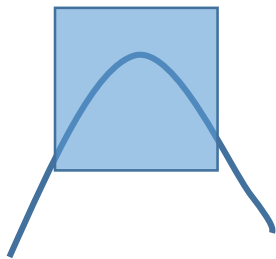


Алгоритм FAST хорошо зарекомендовал себя в приложениях, осуществляющих слежение за объектами в реальном времени.

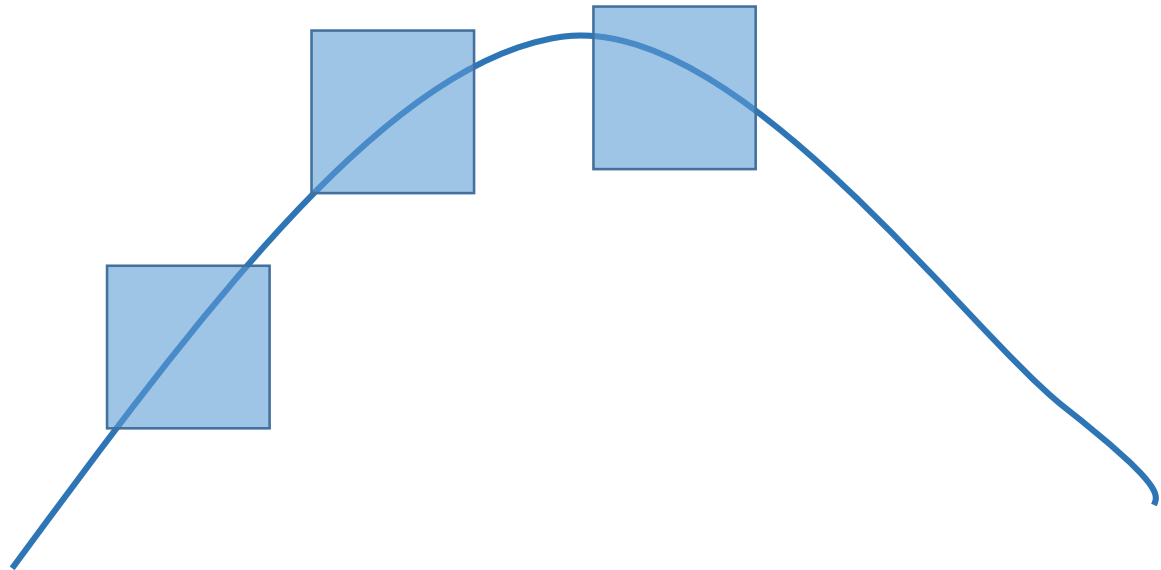
В OpenCV реализован алгоритм ORB [алгоритм ORB \(Oriented FAST and Rotated BRIEF\)](#)

Масштабирование

К недостаткам детекторов углов стоит отнести чувствительность к шуму и зависимость детектора от масштаба изображения (для устранения этого недостатка используют многомасштабный детектор Харриса (multi-scale Harris detector)).



Угол



Край

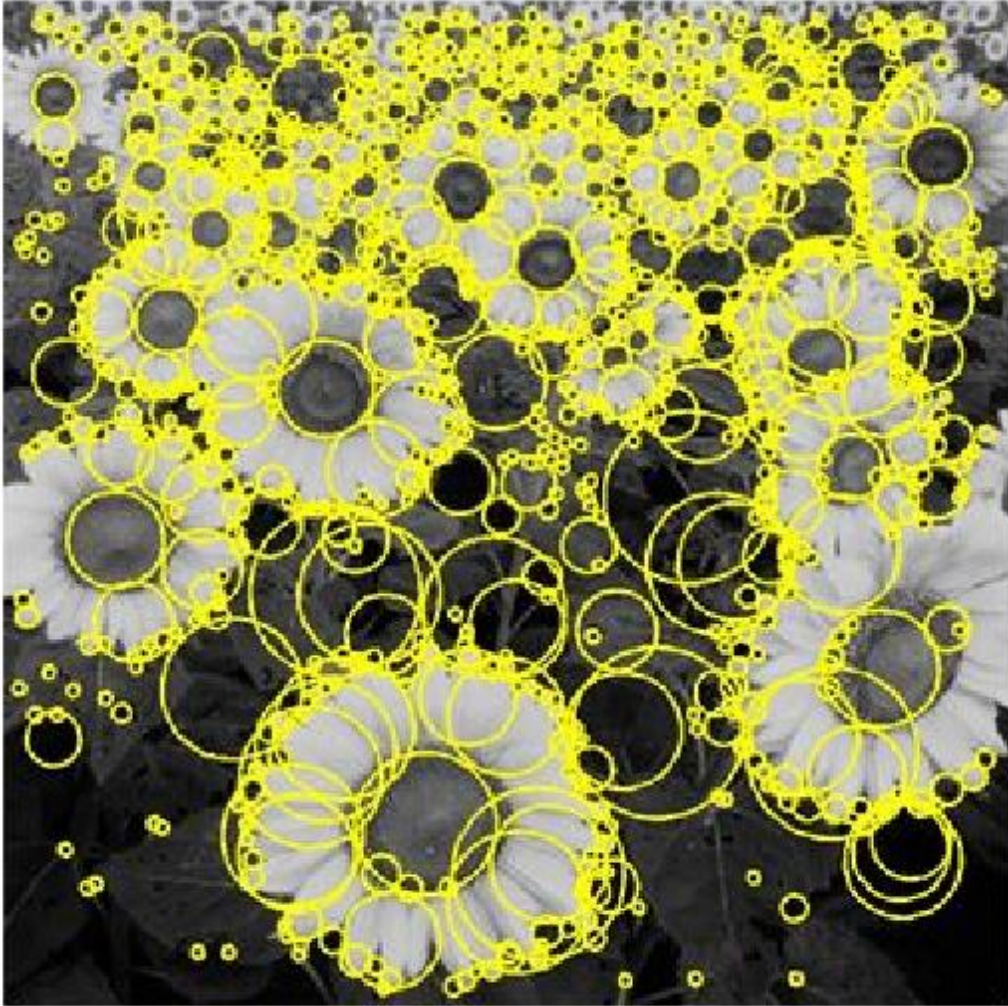
Инвариантность к масштабированию



Цель:

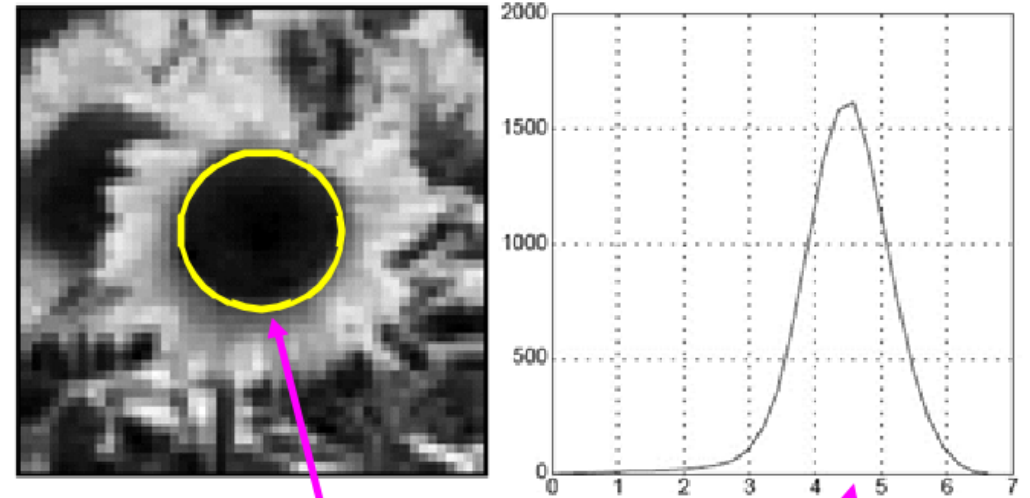
- определять размер окрестности особой точки в масштабированных версиях одного и того же изображения
- требуется метод выбора размера характеристической окрестности

Блобы



Первый такой метод нахождения блобов был предложен в конце 90-х гг.

Характеристический размер определяется как масштаб, на котором достигается максимум отклика Лапласиана



Характеристический масштаб

DoG (Difference of Gaussia)

Разность гауссианов (Difference of Gaussian, DoG). Гауссианом (или изображением, размытым гауссовым фильтром) является изображение

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Здесь L — значение гауссиана в точке с координатами (x, y) , а σ — радиус размытия. G — гауссово ядро, I — значение исходного изображения, $$ — операция свертки.*

Разностью гауссианов называют изображение, полученное путем попиксельного вычитания одного гауссиана исходного изображения из гауссиана с другим радиусом размытия.

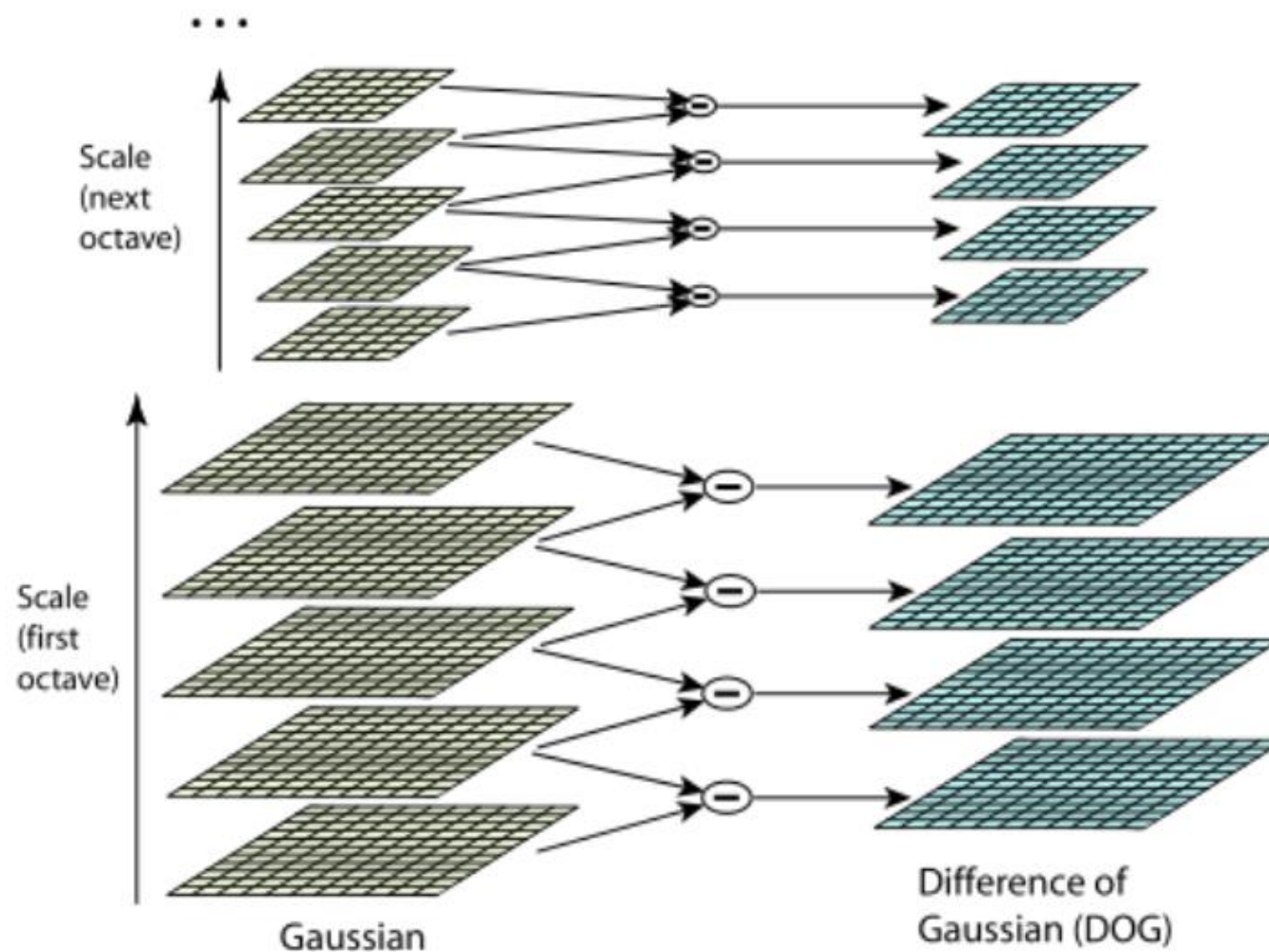
$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

Различная степень размытия изображения гауссовым фильтром может быть принята за исходное изображение, взятое в некотором масштабе.

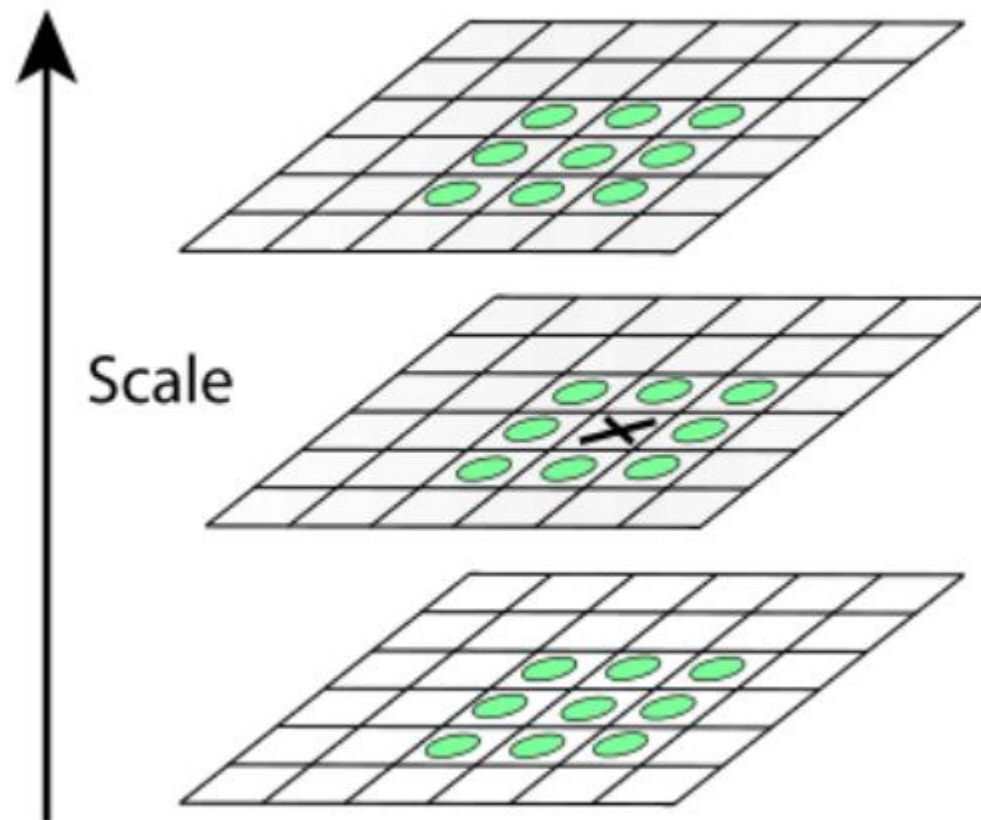
Инвариантность относительно масштаба достигается за счет нахождения ключевых точек для исходного изображения, взятого в разных масштабах. Для этого строится пирамида гауссианов

DoG (Difference of Gaussia)

Строится пирамиды гауссианов и разностей гауссианов



DoG (Difference of Gaussia)

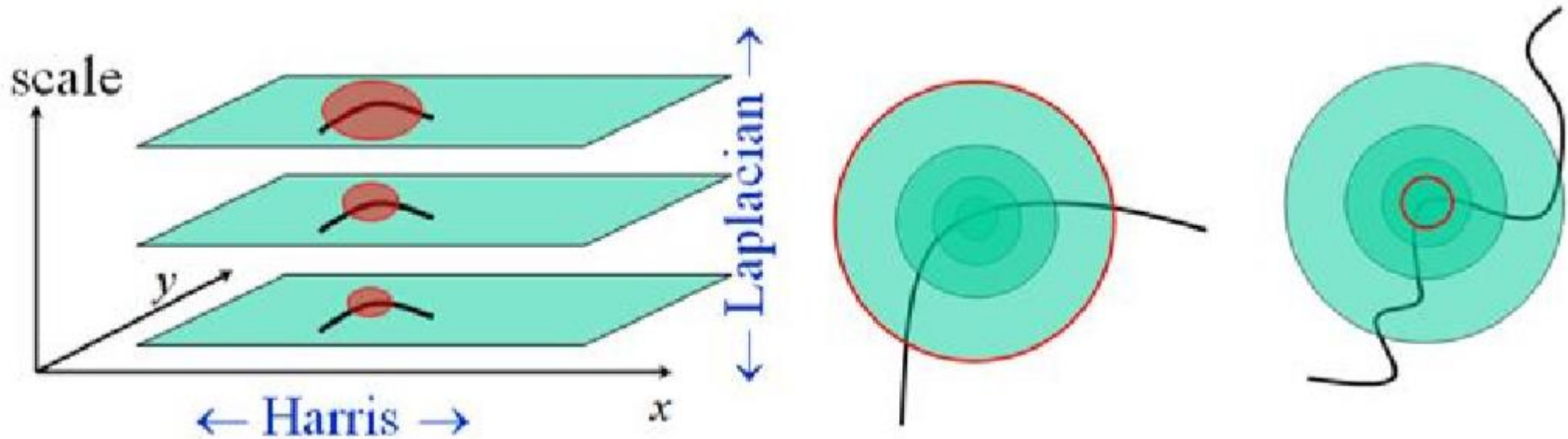


Будем считать точку особой, если она является локальным экстремумом разности гауссианов.

В каждом изображении из пирамиды DoG ищутся точки локального экстремума. Каждая точка текущего изображения DoG сравнивается с её восемью соседями и с девятью соседями в DoG, находящихся на уровень выше и ниже в пирамиде. Если эта точка больше (меньше) всех соседей, то она принимается за точку локального экстремума.

Детектор Harris-Laplacian

Разные варианты чередования вычисления функции Харриса и Лапласиана позволяют выделить углы на изображении, но с характеристическим размером



Детекторы областей

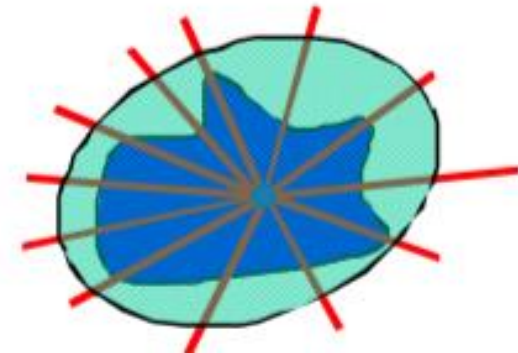
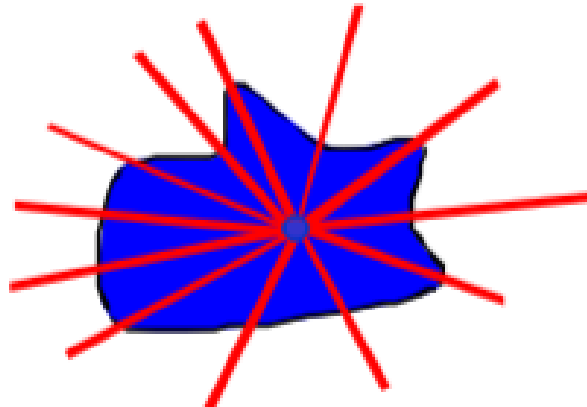
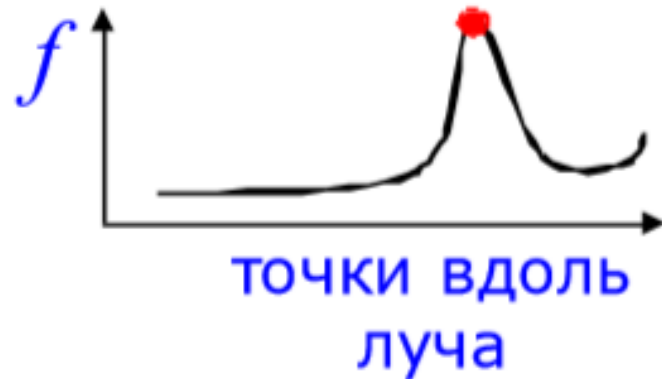
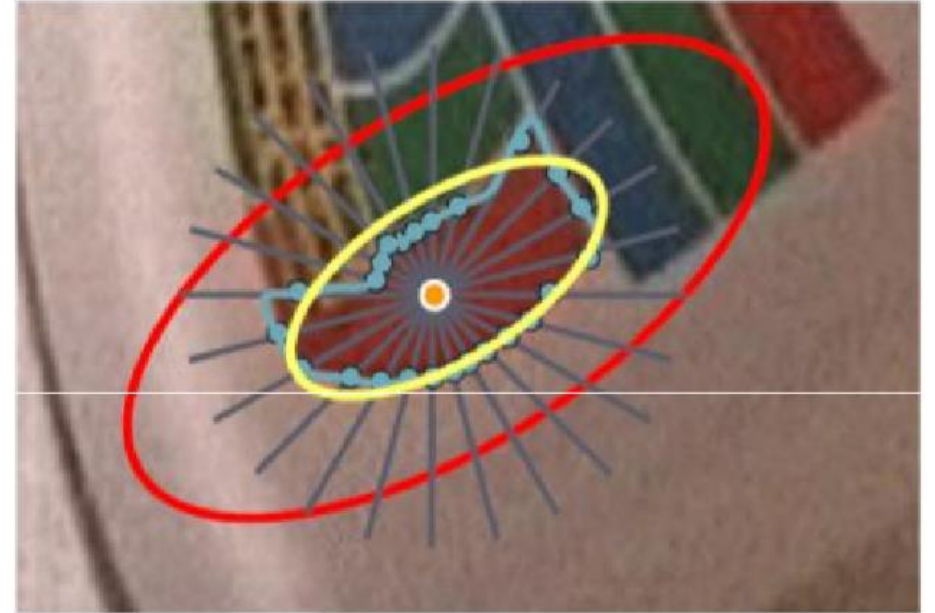
Можно попробовать работать с более уникальными характеристиками изображения – с **областями**. Экстремальных областей гораздо меньше, но они более точно характеризуют сцену или объект.

У экстремальных (особых) областей в этом контексте есть два важных свойства:

- непрерывное преобразование координат изображения. Это означает, что он является аффинно-инвариантным, и не имеет значения, будет ли изображение искажено или искажено.
- монотонное преобразование интенсивности изображения. Подход, конечно, чувствителен к естественным световым эффектам, таким как изменение дневного света или движущихся теней.

Детектор областей IBR (Intensity-extrema based regions)

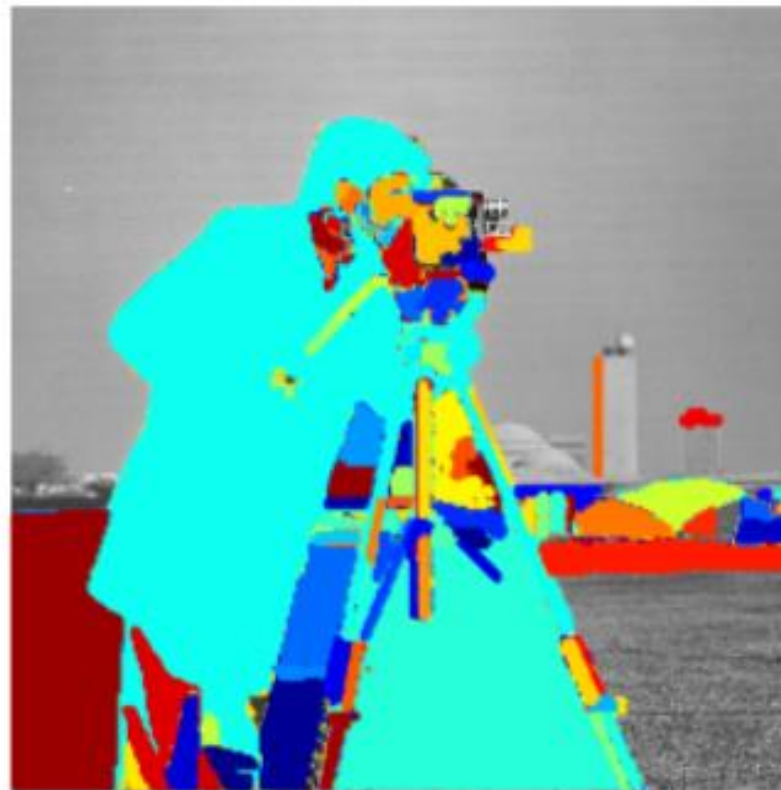
- Идти от локального экстремума яркости по лучам, считая некоторую величину f
- Остановка при достижении пика f



MSER = Maximally Stable Extremal Regions

Метод находит соответствия между элементами изображения в двух изображениях с **разных точек зрения**.

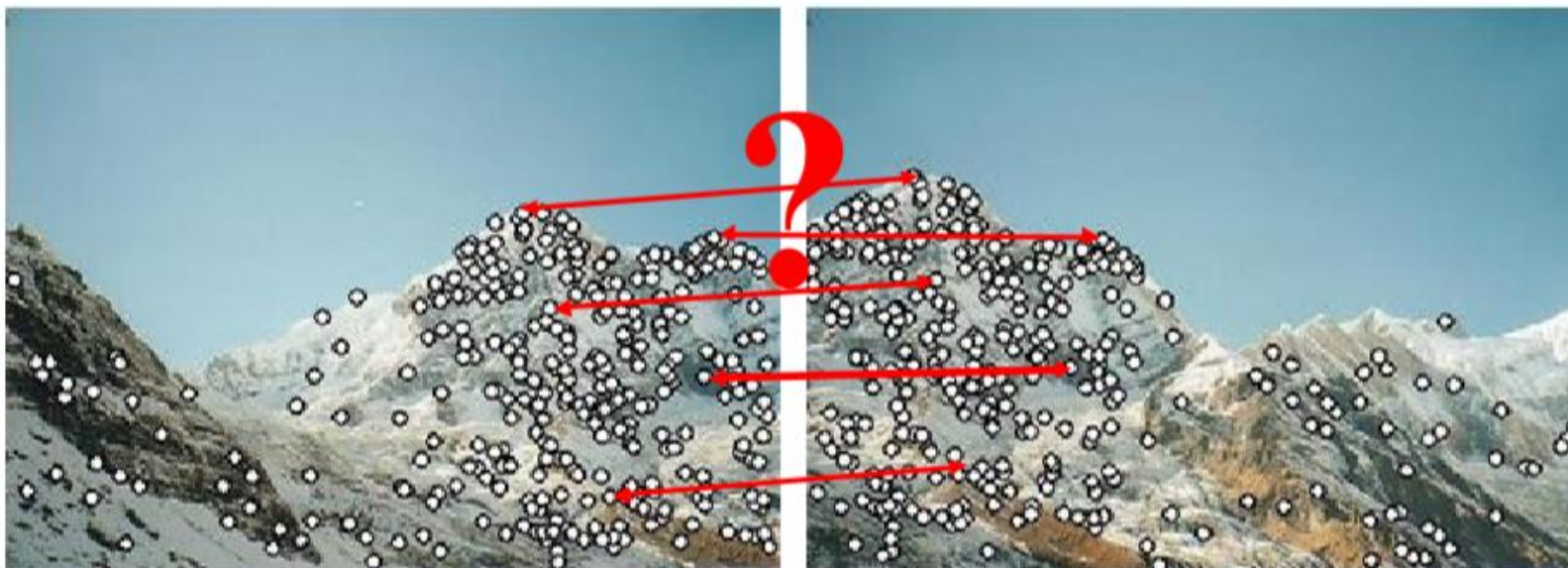
- Задать порог яркости T
- Провести сегментацию
- Извлечь области
- Для каждой области найти порог, при котором рост площади минимален



Резюме

- Локальные особенности – один из основных инструментов анализа изображений
- Рассмотрели основные детекторы особенностей:
 - Детекторы углов: Harris (Forstner), Harris-Laplace)
 - Детекторы Блобов: LoG (Laplacian of Gaussian), DoG (Difference of Gaussians)
 - Детекторы областей: IBR (Intensity-extrema based regions), MSER (Maximally Stable Extremal Regions)

Точки найдены, а как их отличить друг от друга. Для этой цели служат дескрипторы



Дескриптор — идентификатор ключевой точки, выделяющий её из остальной массы особых точек.

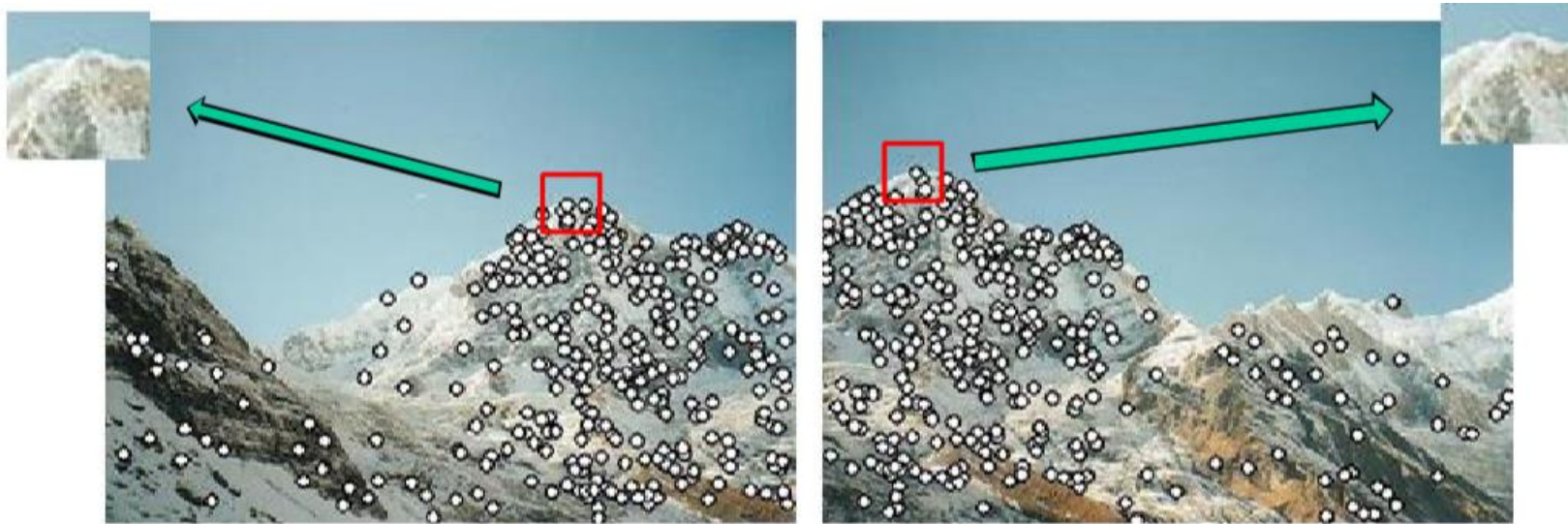
Дескриптор представляет собой запись фрагмента картинки в числовом виде, с помощью дескрипторов можно достаточно точно сравнивать фрагменты без использования самих фрагментов, затем проводится кластеризация, т.е. распределение похожих дескрипторов по кластерам

Дескрипторы

Дескрипторы должны быть:

- специфичны (различные для разных точек)
- локальны (зависеть только от небольшой окрестности)
- инвариантны (к искажениям, изменениям освещенности)
- просты в вычислении

Дескрипторы. Простейший подход



- Возьмём квадратные окрестности, со сторонами, параллельными строкам и столбцам изображения
- Яркости пикселей будут признаками
- Сравнивать будем как изображения попиксельно (SAD, SSD)
- Такая окрестность инвариантна только к сдвигу изображения

Дескрипторы. Инвариантность к яркости

- Можем добиться следующим образом:
 - Локальная нормализация гистограммы
 - Дескрипторы, основанные на градиенте яркости, инвариантны к сдвигу яркости
 - Нормирование яркости - вычесть среднее значение, поделить на дисперсию

$$I' = (I - \mu) / \sigma$$

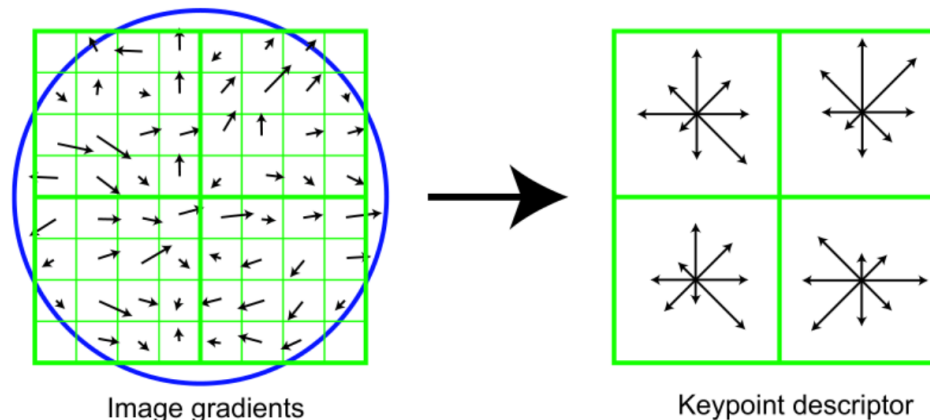


нормализация



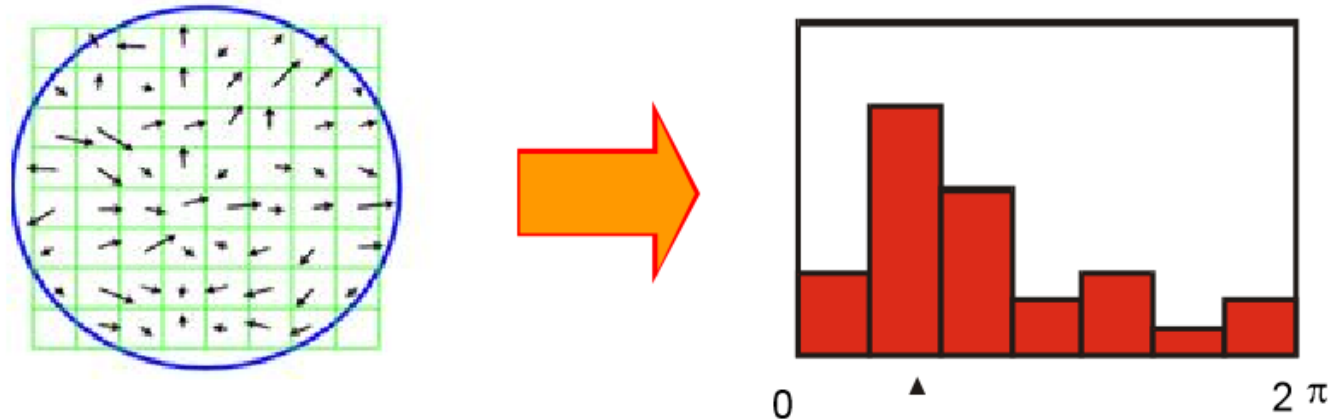
Дескрипторы. SIFT

- Scale-Invariant Feature Transform:
 - Детектор DoG
 - Определение положения и масштаба особенности
 - Ориентация
 - Определение доминантной ориентации по градиентам
 - Дескриптор
 - Использование статистик по направлению градиентам

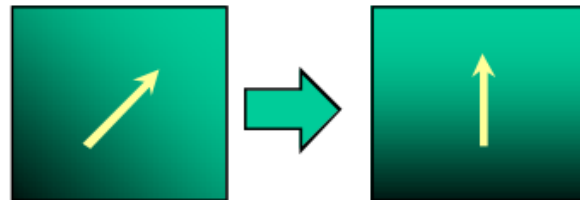


SIFT. Ориентация

- Идея: найти основное (доминантное) направление градиентов пикселей в окрестности точки
- Посчитаем гистограмму, взвешивая вклад по гауссиане с центром в точке



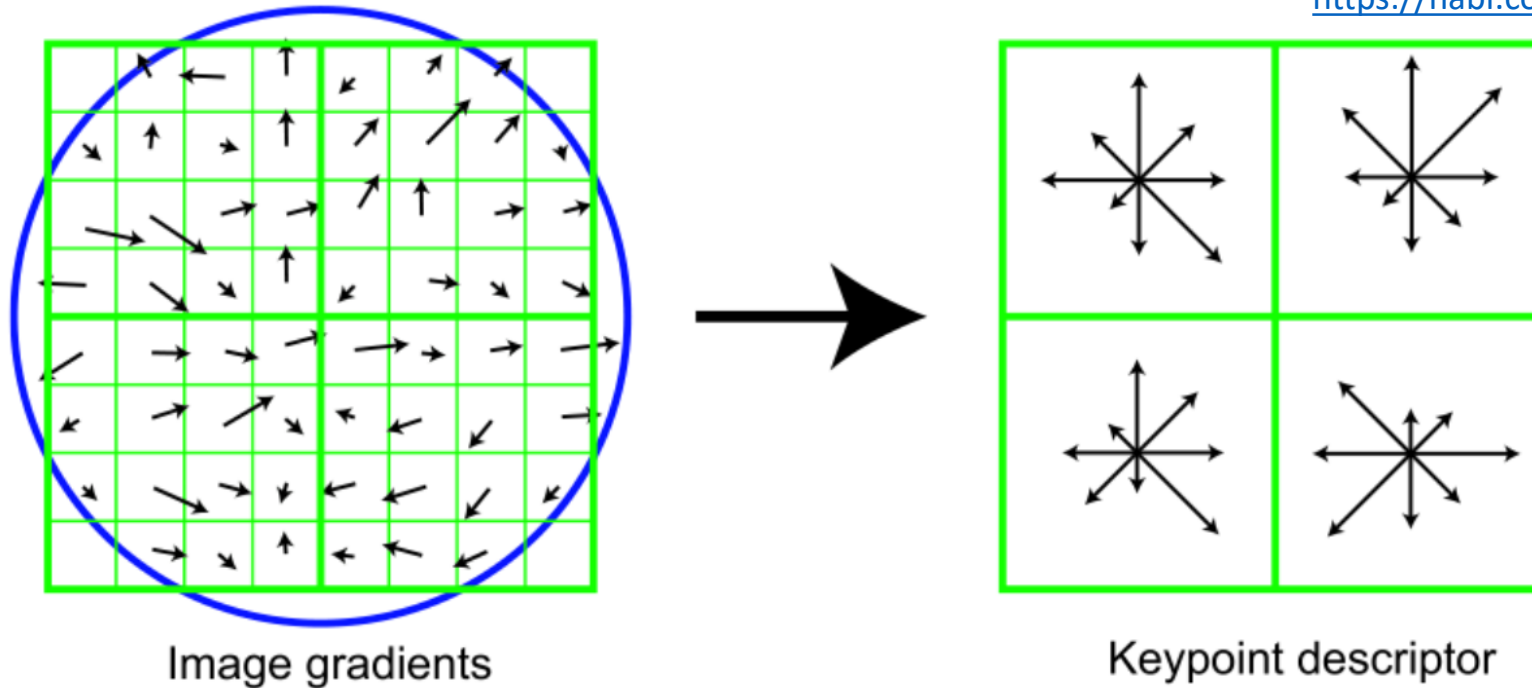
- Повернуть фрагмент так, чтобы доминантное направление градиента было направлено вверх



- Если локальных максимумов несколько – считаем, что несколько точек с разной ориентацией

SIFT. Дескриптор

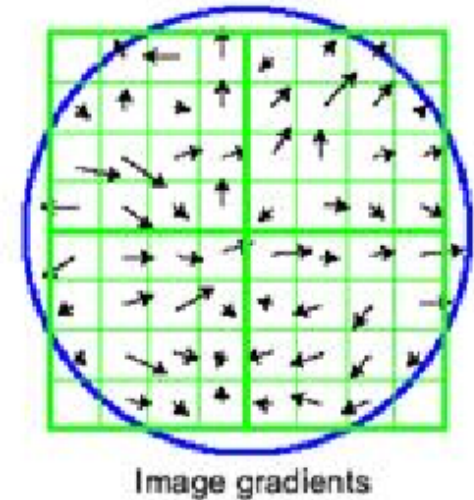
<https://habr.com/ru/post/106302/>



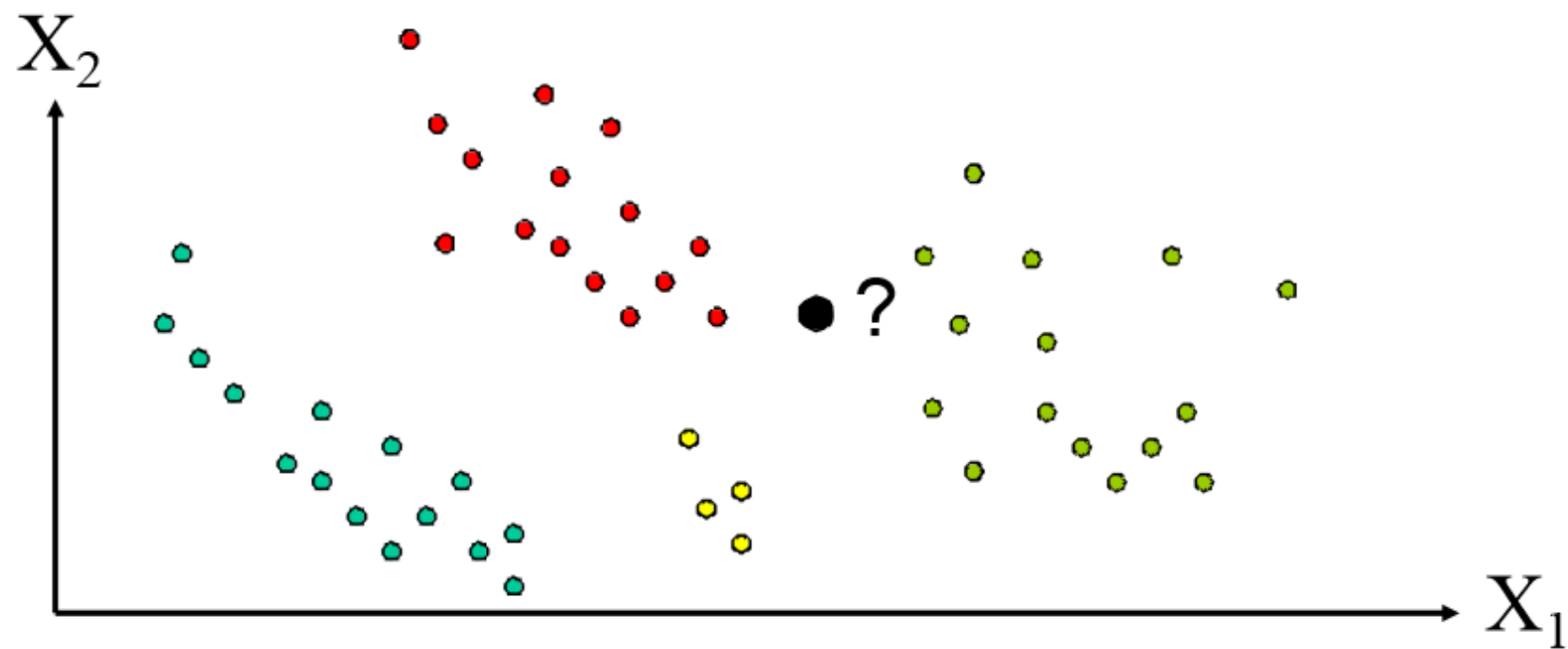
- Для учета локальных свойств разделим окрестность на блоки сеткой, в каждом блоке посчитаем свою гистограмму градиентов
- Обычно – сетка 4x4, в каждой гистограмма с 8ю ячейками
- Стандартная длина вектора-дескриптора – 128 ($4 \cdot 4 \cdot 8$)
- Сравниваем как вектор (разные метрики)

Резюме SIFT

- Детектор SIFT весьма специфичен, устойчив к изменениям освещения, небольшим сдвигам
- Вся схема SIFT (детектор, выбор окрестностей, дескриптор) оказалась очень эффективным инструментом для анализа изображений
- Очень широко используется



Классификация по признакам



Все необходимое для реализации этого метода есть в модуле features2d. Он позволяет находить особые точки, строить их дескрипторы, применять методы сравнения дескрипторов.

Детекторы



- SIFT
- SURF
- FAST
- STAR
- MSER
- HARRIS
- GFTT (Good Features To Track)

Дескрипторы



- SIFT
- SURF
- HoG
- ORB
- FREAK, BRISK
- ...

Сравнение дескрипторов

- BruteForce
- FlannBased
- Bag-Of-Words

Пост-обработка

- Cross check
- Ratio check

Учебный курс
«Разработка мультимедийных приложений
с использованием библиотек OpenCV и IPP»
Писаревский В.Н.

D:\ПМММП\!Литература\Курс-Мультимед
_OpenCV\Lec06_OpenCV_text.pdf