

Градиенты и локальные структуры изображений

Анализ локальной структуры изображения через поведение двумерной функции яркости $I(x, y)$. Поверхность яркости как рельеф с подъёмами и падениями интенсивности.

Однородные области

Поверхность плоская, изображение выглядит однородным

Границы

Поверхность наклонена, возникает граница объекта

Углы

Наклон меняется в нескольких направлениях одновременно

Градиент как вектор максимального изменения

Градиент задаётся вектором частных производных функции изображения:

$$\nabla I(x, y) = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

Модуль градиента

$$|\nabla I| = \sqrt{I_x^2 + I_y^2}$$

Показывает силу локального изменения яркости

Ориентация

$$\theta = \arctan \left(\frac{I_y}{I_x} \right)$$

Указывает направление наибольшего роста яркости

Градиент всегда перпендикулярен линии уровня изображения — естественный инструмент выделения границ.

Операторы Собеля и Превитта

Дискретные производные вычисляются через свёртку с небольшими ядрами. Оператор Собеля сочетает дифференцирование с частичным сглаживанием.

Горизонтальный фильтр G_x

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Выделяет изменения по оси x

Вертикальный фильтр G_y

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Выделяет изменения по оси y

Собель лучше подавляет шум благодаря встроенному сглаживанию, давая более согласованную оценку градиента.

Гауссово сглаживание перед дифференцированием

Дифференцирование усиливает высокочастотные компоненты, включая шум. Поэтому вычисление градиента предваряется сглаживанием:

$$I_{\sigma}(x, y) = (I * G_{\sigma})(x, y)$$

1

Малые σ

Выделяют тонкие границы и детали, но усиливают шум

2

Большие σ

Выделяют устойчивые линии и контуры крупных объектов

Выбор σ определяет масштаб анализа изображения. Вычисление градиентов — это анализ функции яркости в заданном масштабе.

Модуль и ориентация для выделения границ

Большие значения модуля градиента соответствуют потенциальным границам — локальным максимумам вдоль направления градиента.

01

Вычисление I_x и I_y

Оценка частных производных через операторы Собеля

02

Модуль градиента

Показатель резкости изменения интенсивности в окрестности

03

Алгоритм Кэнни

Определяет пиксели с максимальным модулем вдоль ориентации

04

Пороговая фильтрация

Построение карты границ изображения

Ориентированность локальной структуры

Градиент как распределение направлений в небольшом окне вокруг точки:

Однородные области

Направления случайны и малы по величине

Границы

Направления совпадают, перпендикулярны линии разрыва

Углы

Множество направлений с большими значениями — наиболее информативные точки

Анализ локальной ориентированности лежит в основе детекторов углов: они выявляют точки, где градиент меняется во многих направлениях одновременно.

Проблемы масштаба и текстур

Чувствительность к шуму

Градиенты усиливают
высокочастотные компоненты

Текстурные области

Большие значения градиентов
создают ложные срабатывания

Масштабные эффекты

Границы резкие в одном масштабе,
сглажены в другом

❏ Анализ градиента недостаточен без учёта масштаба. Требуются интегральные структуры на окне для усреднения поведения градиентов.

Частотный смысл градиента

Дифференцирование в частотной области соответствует умножению спектра изображения на частоту:

$$\mathcal{F}\{\nabla I\} = j\omega \cdot \mathcal{F}\{I\}$$

Это означает усиление высокочастотных компонент. Градиент — высокочастотный фильтр, выделяющий резкие изменения структуры.

1

Оператор Собеля

Дискретная аппроксимация дифференцирующего фильтра с дополнительным сглаживанием

2

Устойчивость к шуму

Сглаживание улучшает качество выделения границ

Поле градиентов как глобальное описание

Векторное поле градиентов по всему изображению позволяет анализировать глобальную структуру сцены:



Протяжённые структуры

Порождают параллельные направления градиентов



Зоны скачков

Образуют линии резких изменений яркости



Пересечения направлений

Формируют потенциальные ключевые точки

Для извлечения устойчивой характеристики точки вводится матрица вторых моментов.

Вычисление градиента в OpenCV

Практическая реализация вычисления градиента с использованием оператора Собеля:

```
import cv2
import numpy as np

img = cv2.imread("image.jpg", cv2.IMREAD_GRAYSCALE)
lx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3)
ly = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)

magnitude = np.sqrt(lx**2 + ly**2)
orientation = np.arctan2(ly, lx)
```

📌 **Инженерный комментарий:** Использование CV_64F предотвращает переполнение на изображениях с высоким контрастом. Размер ядра ksize=3 — стандартный компромисс между точностью и сглаживанием.

Детекторы углов: Harris и Shi–Tomasi

Различение типов структур: однородные области, границы и углы.

Плоская область	Граница	Угол
Модуль градиента мал во всех направлениях	Модуль велик в одном направлении — перпендикулярно линии разрыва	Интенсивность изменяется существенно во многих направлениях

Углы — стабильные и хорошо различимые точки, пригодные для сопоставления между изображениями. Требуется более глубокое описание через совместный анализ градиентов в малом окне.

Матрица вторых моментов

Автокорреляционный анализ локального окна. Смещение окна на вектор (u, v) приводит к изменению интенсивности:

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

1

Плоская область

Изменение мало во всех
направлениях

2

Граница

Изменение мало вдоль границы,
велико поперёк

3

Угол

Изменение велико при любых
направлениях

Разложение в ряд Тейлора связывает геометрию локальной поверхности яркости с градиентами.

Формула матрицы вторых моментов

Линейное приближение приводит к выражению:

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

где матрица M описывает распределение градиентов внутри окна:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Диагональные элементы

Определяют силу изменения в направлении осей координат

Внедиагональные элементы

Фиксируют корреляцию изменений по двум направлениям

Матрица вторых моментов — обобщённая ковариационная матрица градиентов.

Геометрическая интерпретация собственных значений

Собственные значения матрицы M равны λ_1 и λ_2 . Классификация структур:

Плоская область

$$\lambda_1 \approx 0, \lambda_2 \approx 0$$

Оба значения малы

Граница

Одно значение велико, второе мало

Угол

Оба значения велики

Функция смещения $E(u,v)$ принимает эллиптическую форму. В случае угла эллипс близок к окружности с большим радиусом — высокая вариативность во всех направлениях.

Критерий Харриса

Детектор Харриса использует инвариантную комбинацию собственных значений:

$$R = \det M - k(\text{trace } M)^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

где параметр k обычно выбирают в диапазоне 0.04–0.06.


$$R > 0$$

Угол

Обе компоненты велики, детерминант велик


$$R < 0$$

Граница

Детерминант мал, след велик


$$R \approx 0$$

Плоская область

Всё мало

Критерий показал хорошую устойчивость и эффективность, оставаясь стандартом в компьютерном зрении.


Устойчивость Харриса

Сглаживание градиентов

Градиенты сглаживаются перед вычислением матрицы вторых моментов, снижая влияние шума

Усреднение в окне

Суммирование внутри окна стабилизирует значения матрицы и её собственные значения

 **Ограничение:** Детектор не является масштабно-инвариантным. При увеличении изображения параметры окна и пороги становятся несоответствующими.

Требуется адаптация к изменению масштаба — это привело к появлению последующих методов.

Ограничения метода Харриса

Детектор Харриса чувствителен к выбору параметров:

Ширина окна

Слишком маленькое окно → ложные срабатывания в текстурах.

Слишком большое → размывание локальных структур

Параметр k

Определяет компромисс между выделением границ и углов.

Малое k → смешивание, большое k → пропуск углов

Локализация

Отклик может быть большим, но локализация неустойчива из-за несимметричной формы окна

Детектор Shi–Tomasi

Альтернативный критерий, основанный на минимуме собственных значений:

$$R_{ST} = \min(\lambda_1, \lambda_2)$$

Углы выбираются как точки, где это значение превышает порог.

Прямолинейность

Не участвует параметр k , более простая интерпретация

Точность локализации

Часто обеспечивает лучшую точность, чем Харрис

Применение

Предпочтителен при построении трекинга, например в алгоритме Лукаса–Канаде

Хорошая ключевая точка — та, где обе компоненты структуры значимы.

Реализация в OpenCV

```
import cv2
import numpy as np

img = cv2.imread("image.jpg", cv2.IMREAD_GRAYSCALE)

# Harris
harris = cv2.cornerHarris(img, blockSize=3,
                          ksize=3, k=0.04)

# Shi-Tomasi
shi = cv2.goodFeaturesToTrack(img,
                              maxCorners=300,
                              qualityLevel=0.01,
                              minDistance=5)
```

📌 **Инженерный комментарий:** `cornerHarris` чувствителен к нормировке — изображение желательно приводить к `float32`. Параметры `qualityLevel` и `minDistance` сильно влияют на плотность найденных точек.

Переход к инвариантным признакам

Детекторы Харриса и Shi–Tomasi дают устойчивые ключевые точки, но только в фиксированном масштабе.

Проблема масштаба

При увеличении или уменьшении объекта отклик меняется, точки перестают совпадать

Масштабная нормализация

Необходимость анализа структуры градиентов в многомасштабном пространстве

Октавы и пирамиды

Масштабно-инвариантные детекторы строят пирамиды изображений

Это приводит к идее масштабно-инвариантных детекторов признаков.

Масштабно-инвариантные детекторы

Фундаментальная проблема масштаба

Локальная структура изображения зависит от масштаба наблюдения. Угол, различимый при одном разрешении, может исчезнуть при изменении масштаба.

Разные расстояния

Объект снят с разных расстояний до
камеры

Разное фокусное расстояние

Изменение оптических параметров
съёмки

Геометрическая конфигурация

Объект появляется в разных размерах

Требуется характеристика, эволюционирующая предсказуемо при изменении масштаба — масштабное пространство.

Масштабное пространство $L(x, y, \sigma)$

Семейство сглаженных версий исходного изображения, параметризованное масштабом σ :

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

где G — двумерный гауссиан.

📄 **Теорема о единственности:** Гауссиан — единственная функция сглаживания, удовлетворяющая аксиомам каузальности и стабильности (отсутствие появления новых структур при увеличении σ).

L — строгая математическая конструкция, обеспечивающая однонаправленную эволюцию признаков. Трёхмерная поверхность, где ключевые точки — экстремумы, устойчивые к вариациям размера.

Логарифмическая параметризация масштаба

Автомодельность масштабного пространства: если объект увеличивается в k раз, его представление воспроизводится при увеличении σ также в k раз.

$$\sigma_i = \sigma_0 k^i$$

где $k = 2^{1/s}$ — шаг между масштабами внутри октавы, s — количество уровней.



Относительные отношения между масштабами остаются стабильными.

Лапласиан Гаусса (LoG)

Оптимальный оператор выделения блобов — пятен повышенной или пониженной яркости:

$$\nabla^2 L = L_{xx} + L_{yy}$$

Лапласиан обнаруживает места, где поверхность яркости сильно прогибается. Но с увеличением σ амплитуда убывает.

Нормированная версия для сопоставимых откликов:

$$\sigma^2 \nabla^2 L$$

- ❑ Нормировка делает возможным поиск экстремумов по σ . Но вычисление второй производной для множества уровней слишком дорого.

Difference-of-Gaussians (DoG)

SIFT использует DoG — вычислительную аппроксимацию нормированного LoG:

$$\text{DoG}(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Это приближение нормированного Лапласиана:

$$L(x, y, k\sigma) - L(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 L(x, y, \sigma)$$

1

Быстрое вычисление

DoG вычисляется гораздо быстрее, чем прямой LoG

2

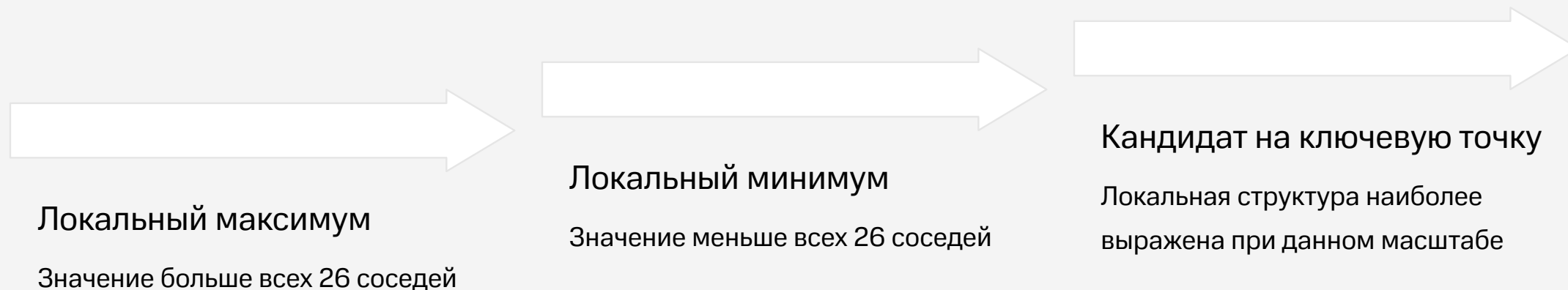
Точность

Экстремумы DoG совпадают с экстремумами нормированного LoG

Фундаментальный инженерный компромисс для работы в реальном времени.

Экстремумы в пространстве (x, y, σ)

Каждая точка DoG сравнивается с 26 соседями: 8 на том же уровне и 9 на уровнях выше и ниже.



Трёхмерный поиск экстремумов делает ключевые точки инвариантными к масштабу: при увеличении объекта экстремум переместится в соседний слой пирамиды.

Точная локализация экстремумов

Найденные экстремумы уточняются с помощью разложения DoG в трёхмерном пространстве Тейлора:

$$D(\mathbf{x}) \approx D + \frac{\partial D}{\partial \mathbf{x}}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

01

Решение системы уравнений

Нахождение точного положения
экстремума

02

Оценка контрастности

Точки с малым контрастом
отбрасываются

03

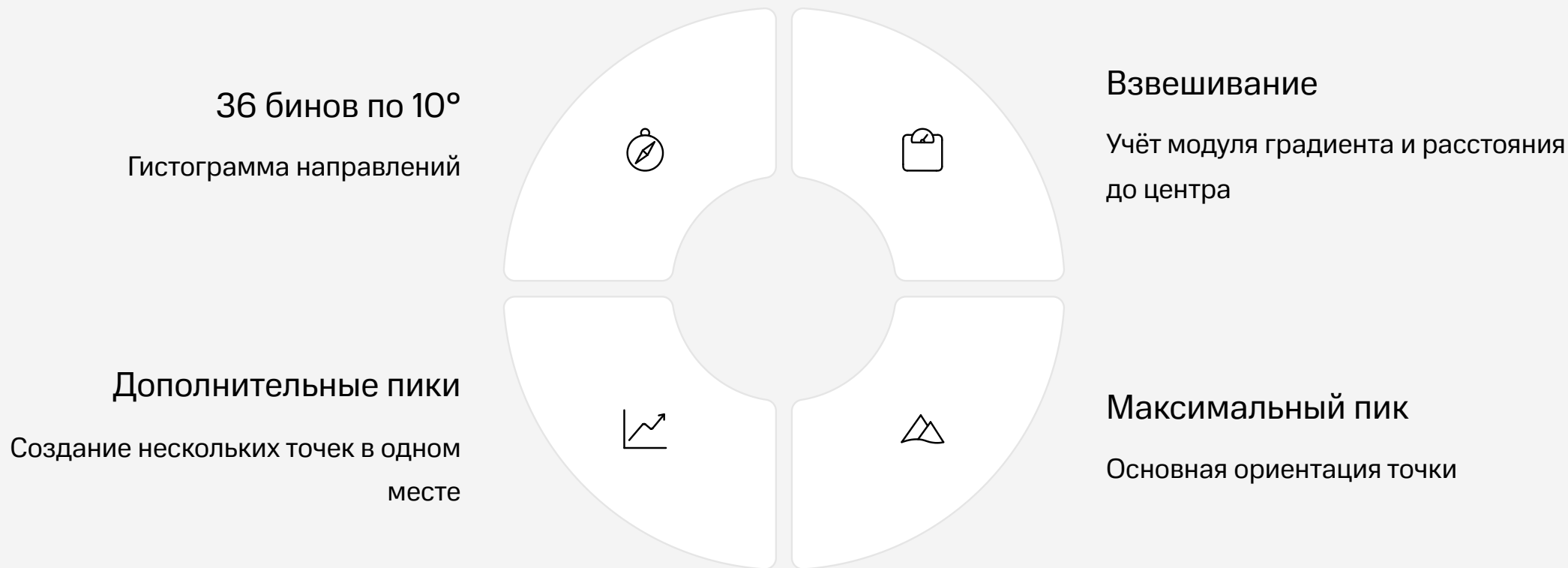
Критерий Гессе

Исключение точек на яркостных рёбрах
с вытянутой структурой

Эти процедуры устраняют до половины ложных кандидатов.

Оценка ориентации

Для поворотной инвариантности SIFT строит гистограмму направлений градиента в окрестности ключевой точки:



Каждая ключевая точка получает локальную ось для сравнения структур независимо от абсолютного поворота.

Инвариантность SIFT

SIFT-ключевая точка описывается тройкой параметров: (x, y) — положение, σ — масштаб, θ — ориентация.

3

Геометрические параметры

Положение, масштаб, ориентация

128

Размер дескриптора

Компактное представление окрестности

Геометрическая инвариантность

- Масштаб
- Поворот
- Небольшие сдвиги

Фотометрическая инвариантность

- Изменения освещённости
- Контраст
- Текстурные регионы


SIFT остаётся одним из наиболее надёжных детекторов в широком диапазоне условий.

Построение DoG-пирамиды

```
import cv2
import numpy as np

img = cv2.imread("image.jpg", 0).astype(np.float32) / 255.0
k = 2 ** (1/3) # три уровня на октаву

sigmas = [1.6, 1.6*k, 1.6*k*k, 1.6*k*k*k]
gauss = [cv2.GaussianBlur(img, (0,0), s)
          for s in sigmas]
dogs = [gauss[i+1] - gauss[i]
        for i in range(len(gauss)-1)]
```

 **Инженерный комментарий:** Выбор $k = 2^{1/3}$ даёт три уровня на октаву — хороший компромисс между вычислительностью и точностью. Нормированные значения $[0,1]$ снижают численные ошибки при вычитании.

Дескрипторы признаков

От точек к описаниям

Ключевая точка — это лишь положение, масштаб и ориентация. Требуется компактное численное представление окрестности.

Информативность

Точки можно отличить друг от друга

Компактность

Быстрый поиск соответствий

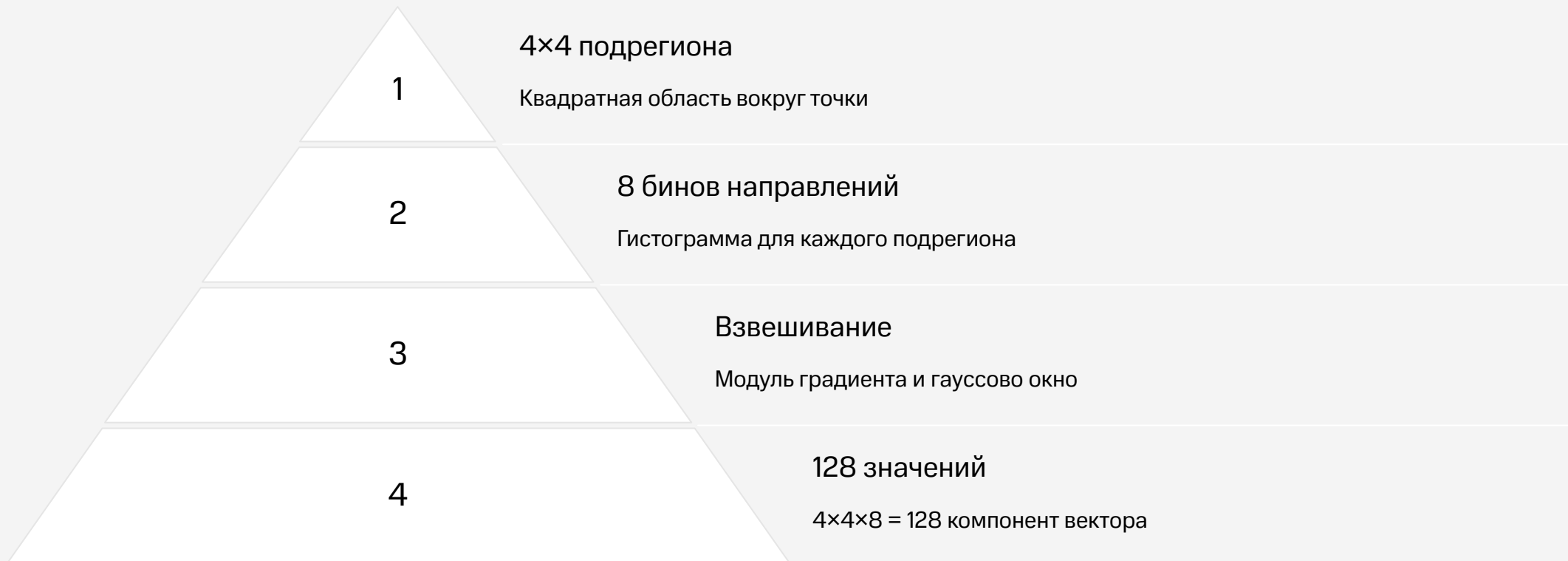
Инвариантность

Сравнение изображений в разных условиях

Дескриптор — «язык», в котором локальная структура превращается в объективный код для сопоставления.

SIFT-дескриптор

Построение ориентированных гистограмм градиентов в локальной системе координат.



Ориентации измеряются относительно главной ориентации ключевой точки — поворотная инвариантность. Оптимальный компромисс между выразительностью и компактностью.

Математическая структура SIFT

Вклад в гистограмму подрегиона (i,j) и бина k:

$$h_{ijk} = w(x, y) \cdot m(x, y) \cdot \delta_k(\theta'(x, y))$$

где δ_k — билинейная интерполяция между соседними бинами, $w(x, y)$ — двумерное гауссово окно.

После вычисления всех гистограмм дескриптор нормируется:

$$\mathbf{d} = \frac{\mathbf{h}}{|\mathbf{h}|_2}$$

Крупные значения обрезаются и нормируются повторно для устойчивости к интенсивным перепадам яркости. Двойная нормировка делает SIFT фотометрически инвариантным.

Устойчивость SIFT к преобразованиям

Линейная фотометрическая трансформация:

$$I'(x, y) = aI(x, y) + b$$

1

Модули градиентов

Масштабируются на коэффициент a

2

Ориентации

Остаются прежними

3

Нормировка

Устраняет влияние множителя a

Отсечение пиков улучшает устойчивость к локальным выбросам (яркие блики). SIFT стабилен при сильных фотометрических вариациях — важно для робототехники и SLAM.

SURF: ускорение вычислений

Speeded-Up Robust Features — более быстрый аналог SIFT. Замена гауссовых производных фильтрами Хаара через интегральное изображение:

$$II(x, y) = \sum_{u \leq x} \sum_{v \leq y} I(u, v)$$

Сумма интенсивностей в любом прямоугольнике вычисляется за четыре операции.

64

Размер дескриптора

Вдвое меньше SIFT

5-8x

Ускорение

По сравнению с SIFT

Снижение вычислительных затрат на порядок при умеренной устойчивости.

Интегральные изображения

Математическое основание ускорения SURF. Вычисление суммы в прямоугольнике:

$$S = II(x_2, y_2) - II(x_1, y_2) - II(x_2, y_1) + II(x_1, y_1)$$

Линейная стоимость

Затраты растут линейно с количеством фильтров, а не с площадью окна

Встроенные системы

Особенно важно для мобильных устройств, где SIFT слишком тяжёлый

Инвариантность

Достаточно устойчив к масштабным и ориентационным изменениям

ORB: FAST + BRIEF

Oriented FAST and Rotated BRIEF — быстрый, свободный от патентов аналог SIFT и SURF.

Детектор FAST

Определяет углы по паттерну яркостей вокруг точки

Дескриптор BRIEF

Бинарный вектор на основе сравнений интенсивностей в парах точек

Локальная ориентация через моменты для инвариантности к вращению:

$$\theta = \arctan \frac{\sum x \cdot I(x, y)}{\sum y \cdot I(x, y)}$$

После ротирования паттерна выборка относительно θ бинарный дескриптор становится поворотно-инвариантным. Десятки тысяч дескрипторов в секунду.

Бинарные тесты BRIEF

BRIEF строится как вектор длиной 256 или 512 из бинарных тестов:

$$\tau_i = \begin{cases} 1, & I(p_i) < I(q_i) \\ 0, & I(p_i) \geq I(q_i) \end{cases}$$

где пары точек (p_i, q_i) выбираются по распределению внутри окна.

1

Сортировка по Фишеру

Пары сортируются для повышения различимости дескриптора

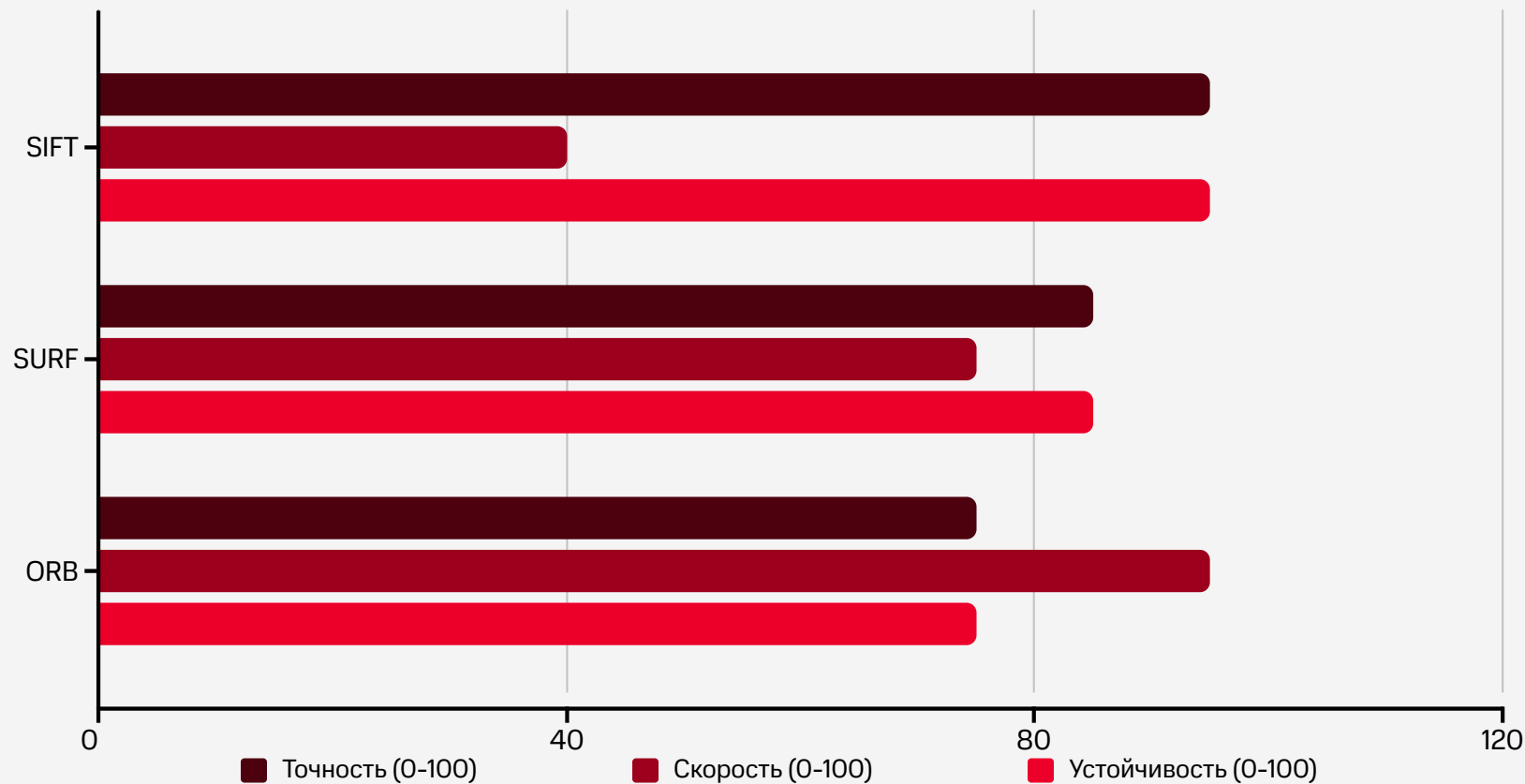
2

Расстояние Хэмминга

Сравнение через XOR и подсчёт единиц — аппаратное ускорение

Тысячи точек можно сравнить за миллисекунды.

Сравнение SIFT, SURF и ORB



Выбор зависит от задачи: SLAM и real-time — ORB; панорамная сшивка — SIFT/SURF; мобильные приложения — ORB.

Получение дескрипторов в OpenCV

```
import cv2

img = cv2.imread("image.jpg", 0)

# --- SIFT ---
sift = cv2.SIFT_create()
kp_s, des_s = sift.detectAndCompute(img, None)

# --- SURF ---
surf = cv2.xfeatures2d.SURF_create(400)
kp_u, des_u = surf.detectAndCompute(img, None)

# --- ORB ---
orb = cv2.ORB_create(nfeatures=1000)
kp_o, des_o = orb.detectAndCompute(img, None)
```

📌 **Инженерный комментарий:** SIFT и SURF доступны в opencv-contrib. ORB встроен по умолчанию. Количество ключевых точек (nfeatures, hessianThreshold) сильно влияет на стабильность сопоставления.

Сопоставление признаков

Метрики расстояния

Дескрипторы SIFT и SURF — векторы в вещественном пространстве. Задача — поиск пар (d_i, d'_j) с минимальным расстоянием.

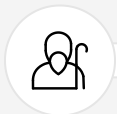
Для SIFT используется L2-норма:

$$|d_i - d'_j|_2 = \sqrt{\sum_k (d_{ik} - d'_{jk})^2}$$

SIFT-дескрипторы нормированы, поэтому L2 расстояние отражает различие структур окрестностей, а не абсолютную яркость. Два дескриптора совпадают даже при значительном отличии освещения — при условии сохранения локального распределения градиентов.

Быстрый поиск ближайших соседей

Поиск ближайшего соседа среди тысяч дескрипторов — дорогостоящая операция. Используются специализированные структуры данных.



kd-деревья

Рекурсивное разбиение
пространства по координатам



FLANN

Fast Library for Approximate Nearest
Neighbors



k-means деревья

Комбинация случайного разбиения
и кластеризации

FLANN ускоряет поиск на порядок при минимальной потере точности — неотъемлемая часть систем с десятками тысяч дескрипторов.

Фильтр Lowe (ratio test)

Критерий отношения для отбраковки ложных совпадений:

$$\frac{|d_i - d'_j|_2}{|d_i - d'_k|_2} < r$$

где d'_j — ближайший сосед, d'_k — второй ближайший, r — порог (обычно $r=0.75$).

Надёжное совпадение

Отношение расстояний мало — первый сосед значительно ближе второго

Ненадёжное совпадение

Отношение велико — оба соседа примерно одинаково близки

Ratio test резко снижает количество ложных срабатываний, особенно в текстурных областях. Обязательный шаг в большинстве систем сопоставления изображений.

Необходимость геометрической проверки

Даже после ratio test остаётся значительное число несовпадений:



Повторяющиеся текстуры

Множество похожих
дескрипторов в регулярных
паттернах



Перспективные искажения

Изменение формы объектов
при изменении угла обзора



Шум и динамика

Шум изображения и
движущиеся объекты

Необходимо учитывать глобальные геометрические отношения между изображениями — сочетание локальных и глобальных признаков.

Эпиполярная геометрия

Если два изображения — проекции одной сцены, соответствующие точки связаны фундаментальным уравнением:

$$x'^T F x = 0$$

где F — фундаментальная матрица, x и x' — гомогенные координаты точек.

Эпиполярная линия

Каждой точке на первом изображении соответствует линия на втором

1

2

3

Отбраковка

Несоответствующие пары являются ложными

Геометрическая проверка

Пара должна удовлетворять эпиполярному условию

Вычисление фундаментальной матрицы требует устойчивости к выбросам — используется RANSAC.

Метод RANSAC

Random Sample Consensus — итерационный алгоритм оценки параметров модели при наличии большого числа выбросов.

Случайная выборка

Выбор подмножества соответствий

Выбор лучшей модели

Максимум инлаеров



Восстановление
преобразования

Вычисление параметров модели

Подсчёт инлаеров

Число согласованных точек

RANSAC — центральный метод для панорам, визуальной одометрии и реконструкции сцены.

Гомография и аффинные преобразования

Геометрические преобразования зависят от сцены.

Гомография

Для плоских объектов (страницы, стены):

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}$$

Отображение точек: $x' = Hx$

Аффинное преобразование

Для более общих случаев:

- Параллельный перенос
- Вращение
- Масштабирование
- Сдвиги

Выбор модели определяется задачей: сшивка панорам — гомография, локальное сопоставление — аффинная модель.

Построение панорамы

Сопоставление признаков — ключевой элемент алгоритмов построения панорамы.

1

Выделение ключевых точек

SIFT/SURF/ORB детекторы

2

Сопоставление дескрипторов

Поиск соответствий с ratio test

3

RANSAC оценка гомографии

Устойчивое вычисление преобразования

4

Проекция и слияние

Объединение изображений с экспозиционной коррекцией

Устойчивость SIFT/SURF/ORB + RANSAC позволяет строить панорамы на смартфонах за доли секунды.

Ограничения сопоставления признаков

Ситуации, где сопоставление признаков работает плохо:



Области без текстуры

Небо, стены — ключевых точек мало
или они плохо различимы



Повторяющиеся паттерны

Окна небоскрёба — множество
похожих дескрипторов, массовые
ложные совпадения



Требуются дополнительные методы

Более сложная модель сцены,
линии, контуры, обучаемые
дескрипторы, нейросетевые
архитектуры

Сопоставление признаков — мощная, но не универсальная технология. Эффективность определяется разнообразием и выразительностью структуры изображения.

Сопоставление SIFT + BFMatcher + RANSAC

```
import cv2
import numpy as np

img1 = cv2.imread("a.jpg", 0)
img2 = cv2.imread("b.jpg", 0)

sift = cv2.SIFT_create()
kp1, d1 = sift.detectAndCompute(img1, None)
kp2, d2 = sift.detectAndCompute(img2, None)

bf = cv2.BFMatcher(cv2.NORM_L2)
matches = bf.knnMatch(d1, d2, k=2)

# Ratio test
good = []
for m, n in matches:
    if m.distance < 0.75 * n.distance:
        good.append(m)

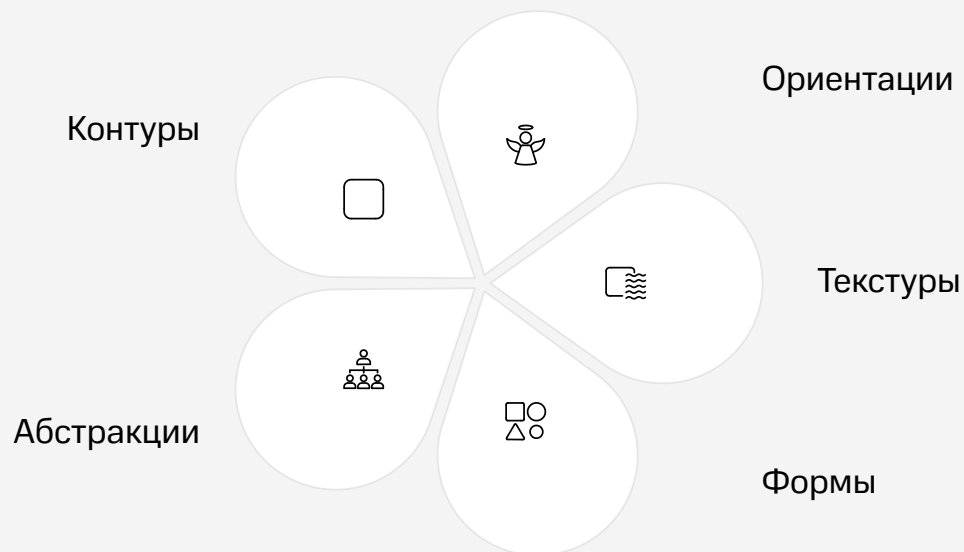
# RANSAC
pts1 = np.float32([kp1[m.queryIdx].pt for m in good])
pts2 = np.float32([kp2[m.trainIdx].pt for m in good])
H, mask = cv2.findHomography(pts1, pts2, cv2.RANSAC)
```

 **Инженерный комментарий:** ransacReprojThreshold определяет чувствительность модели. Используют значения 3–5 пикселей, при высоком разрешении — увеличивают.

Признаки как язык изображения

От классики к нейросетям

Классические признаки — градиенты, углы, blobs, локальные гистограммы — выражение фундаментальной идеи: изображение описывается через иерархию локальных структур.



В классических методах признаки создаются вручную. В нейросетевых — обучаются, но основаны на тех же принципах: локальность, инвариантность, иерархия.

Переоткрытие локальных фильтров в CNN

Первые слои свёрточных сетей перерабатывают изображение почти так же, как классические операторы Собеля и Превитта.

Горизонтальные границы

Весовые фильтры первых слоёв

Вертикальные границы

Детекторы ориентированных
структур

Диагональные границы

Выделение текстурных частот

Это не случайно — следствие оптимизации. CNN «учатся» на данных то, что инженеры закладывали вручную, но масштабнее, глубже и точнее, в контексте целевой задачи и глобальной структуры изображения.

Первые слои CNN как детекторы

Активации первых слоёв современных архитектур (ResNet, VGG, ConvNeXt) похожи на карты модуля и направления градиента.

Активаторы границ

Некоторые фильтры работают как детекторы ярких границ

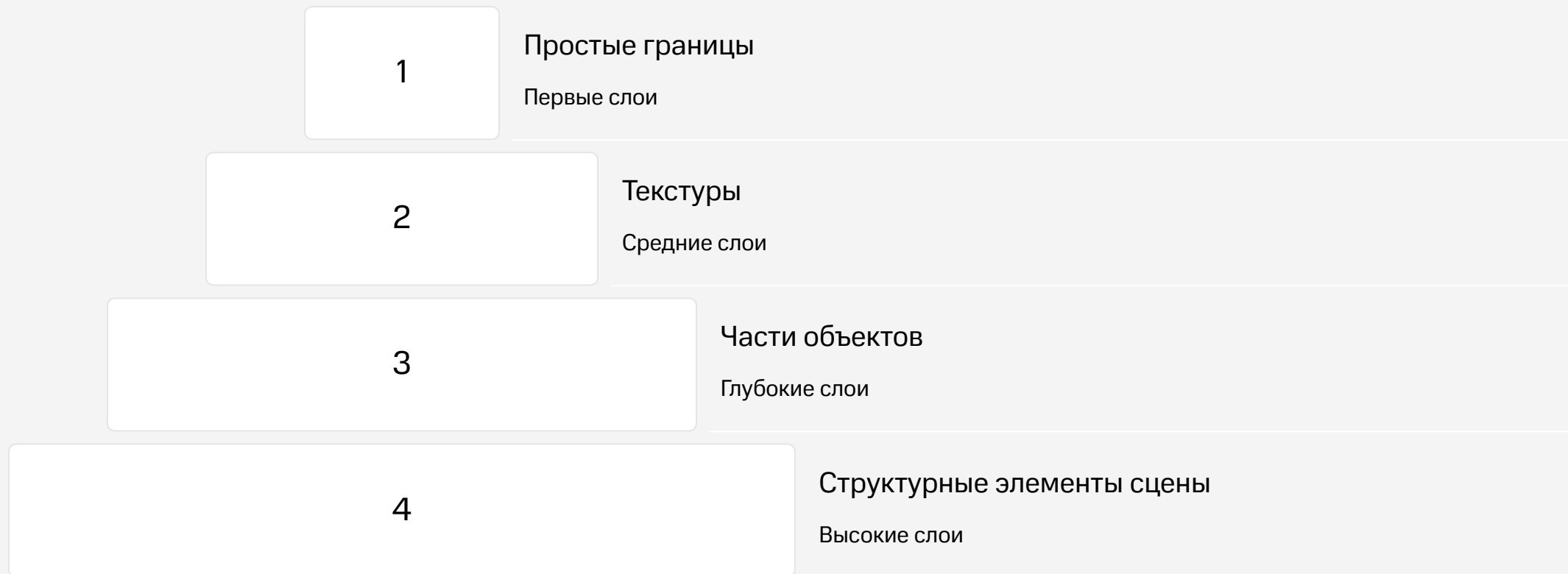
Детекторы углов

Другие похожи на детекторы углов Харриса — реагируют на многонаправленные изменения

Такие фильтры обеспечивают локальную инвариантность: при небольшом сдвиге, повороте или масштабировании объекта ответы первых слоёв остаются похожими. Автоматическое построение градиентных признаков, стабилизированных через обучение.

От локальных к глобальным признакам

CNN продолжают обработку признаков на десятки слоёв, повышая уровень абстракции:



На высоких слоях признаки обладают большим рецептивным полем, кодируя глобальные отношения между частями объекта. Глубокое непрерывное признаковое пространство.

Пирамиды признаков в CNN

Современные архитектуры (Feature Pyramid Networks, HRNet, U-Net) явно используют многомасштабный анализ, аналогичный гауссовой пирамиде, но обучаемый.

1

Высокие разрешения

Мелкие детали объектов

2

Средние разрешения

Части объектов

3

Глубокие уровни

Крупные структуры

4

Объединение

Единое признаковое пространство

Идея восходит к SIFT, где каждый дескриптор соответствует определённому масштабу. В CNN масштабы формируются в процессе обучения — адаптивная и контекстно зависящая система.

Признаки среднего уровня

На промежуточных слоях CNN появляются признаки, отражающие части объектов:



Глаз

Биологические структуры



Колесо автомобиля

Механические части



Окно здания

Архитектурные элементы



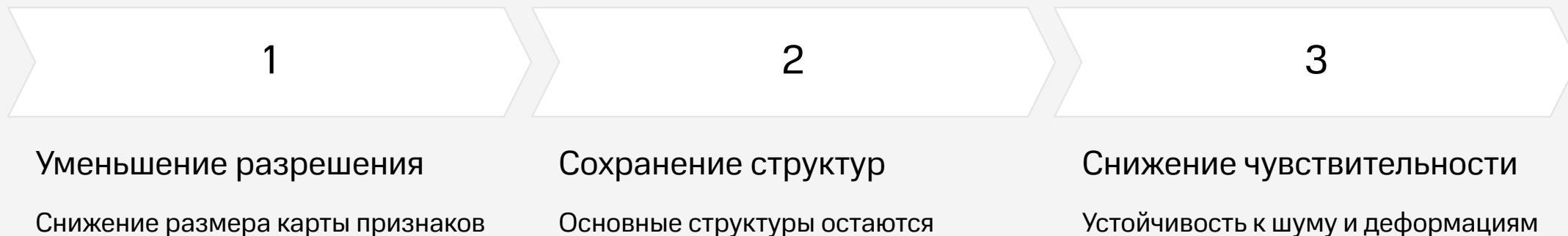
Угол стола

Геометрические формы

Эти признаки инвариантны к вариациям цвета, освещения и текстуры, но схватывают геометрию. В классических подходах таких признаков не существует.

Пространственная инвариантность

Инвариантность к небольшим сдвигам достигается через операции пулинга или сглаженного свёрточного субсемплинга.



В классической обработке аналогичную функцию служат масштабные фильтры и нормировки в дескрипторах (гауссово окно в SIFT). В CNN такие механизмы обучаются или настраиваются архитектурой.

Классические vs обучаемые признаки

Классические дескрипторы

- Определены жёстко — формулы заданы инженером
- Не адаптируются к задаче
- SIFT одинаков для всех применений
- Ограничены локальными градиентами

CNN-признаки

- Формируются данными и целевой функцией
- Оптимальны для конкретной проблемы
- Различны для разных задач
- Включают глобальный контекст

CNN создаёт значительно более богатое признаковое пространство, став стандартом в задачах распознавания с достаточным количеством данных.

Роль локальных признаков сегодня

Локальные признаки не исчезли. В робототехнике, SLAM, 3D-реконструкции классические признаки SIFT/ORB используются благодаря:

Интерпретируемость

Понятная геометрическая интерпретация

Устойчивость

Детерминированное поведение

Независимость от данных

Не требуют больших обучающих выборок

Современные методы совмещают CNN-признаки и локальные дескрипторы: сети обучают дескрипторы, аналогичные SIFT (SuperPoint, R2D2), используемые в классическом процессе сопоставления. Гибридная схема объединяет преимущества обоих подходов.

Переход к анализу движения

Все рассмотренные методы — от градиентов до CNN-признаков — формируют векторное представление изображения, где локальная структура выражена количественно.

Локальные признаки

Векторное представление структуры

Согласованность изменений

Определяет поле оптического потока



The diagram consists of a horizontal line with three white circular nodes. The first node is connected by a vertical line to the text 'Локальные признаки' and 'Векторное представление структуры'. The second node is connected by a vertical line to the text 'Изменения между кадрами' and 'Движение отражается в изменениях признаков'. The third node is connected by a vertical line to the text 'Согласованность изменений' and 'Определяет поле оптического потока'.

Изменения между кадрами

Движение отражается в изменениях признаков

Если локальные структуры можно сопоставить между двумя изображениями, то можно определить, как они сместились во времени. Это составляет основу оптического потока — тема следующей лекции.