Ключевые точки и построение панорамы

Компьютерное зрение

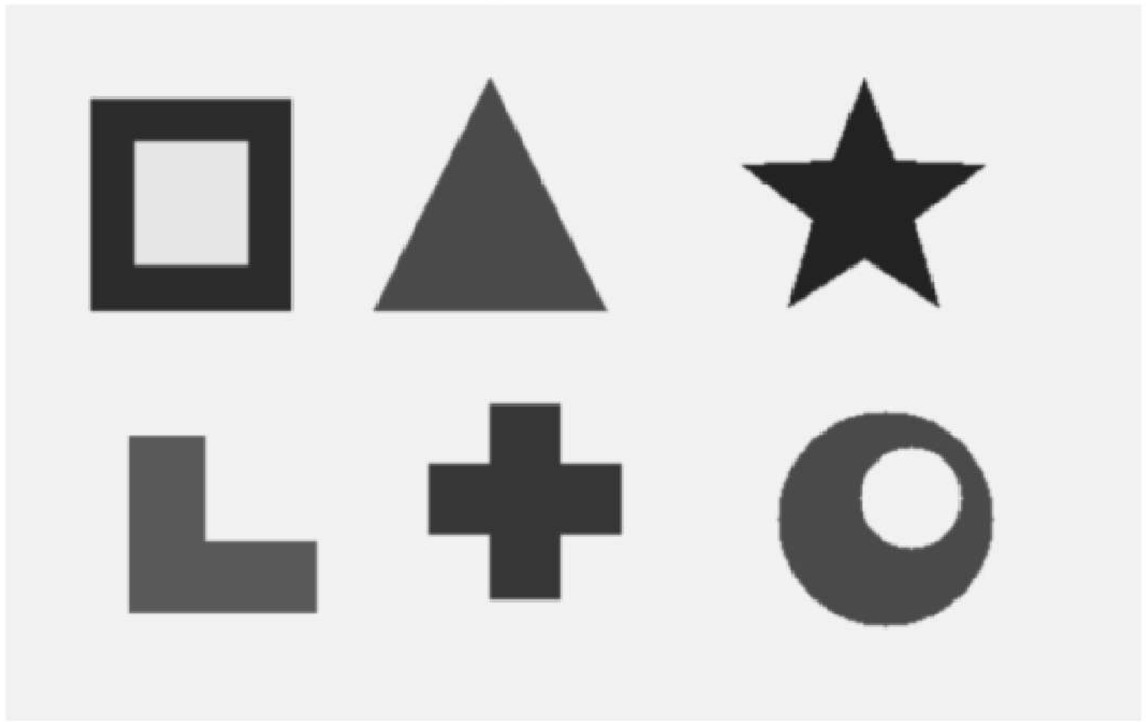
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК

2026

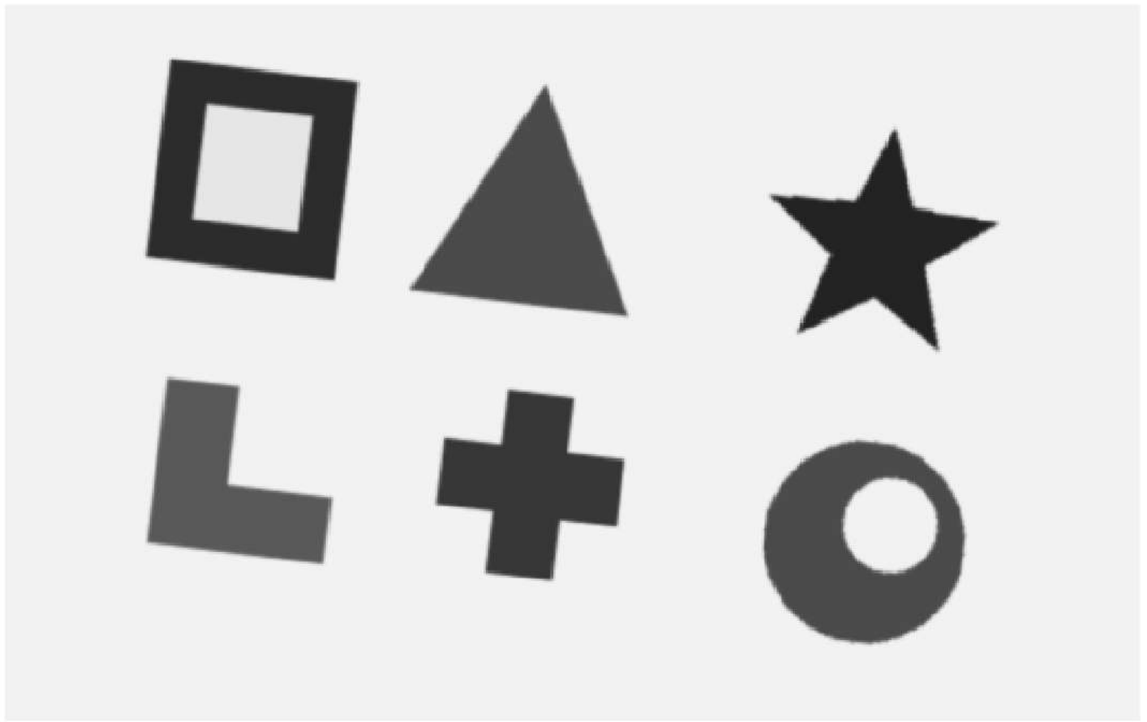
Сопоставление изображений с фигурами

Одинаковые фигуры наблюдаются в разных положениях, масштабах и ориентациях.

Изображение A

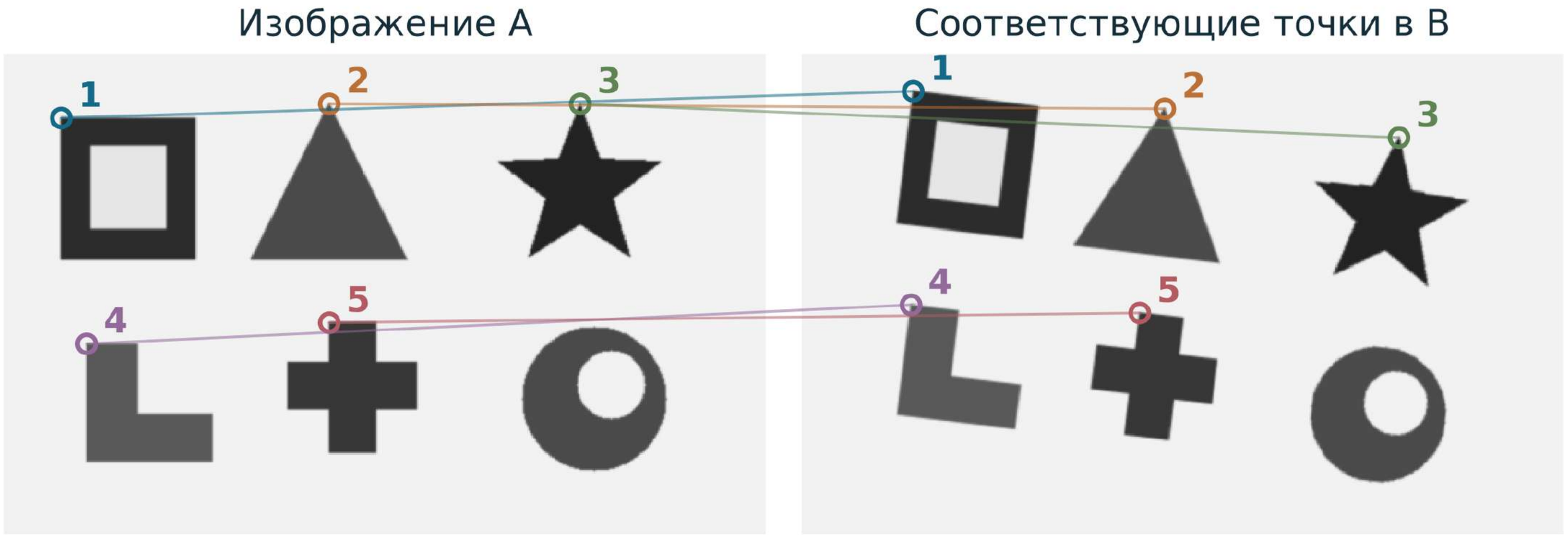


Изображение B



Соответствия ключевых точек

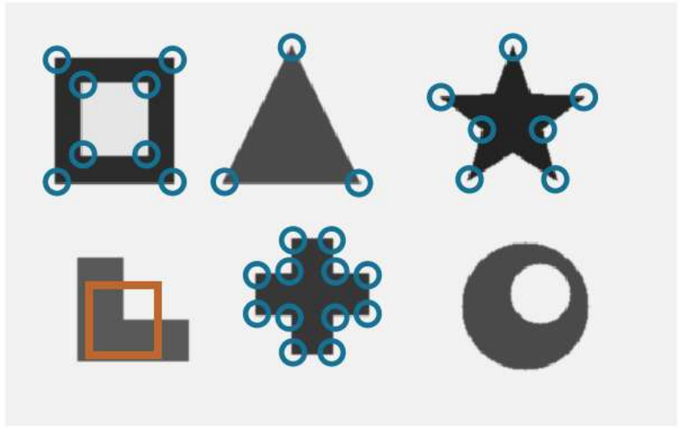
Пара точек должна описывать одно и то же место фигуры, а не просто близкие координаты.



Детектор и дескриптор

Детектор выбирает координаты.
Дескриптор преобразует окружающие пиксели в числа для сравнения.

Детектор



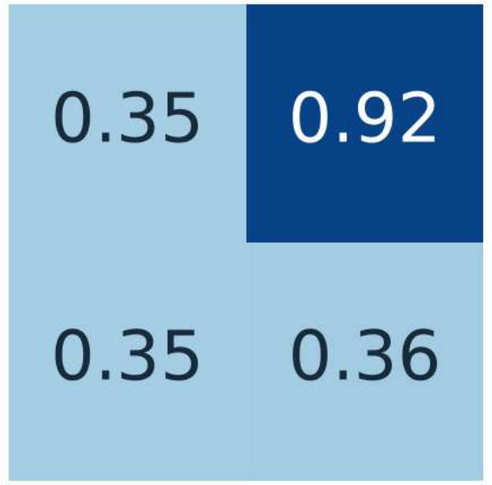
Координаты центра:
(94, 255)

Окрестность



Пиксели вокруг
выбранного центра

Дескриптор

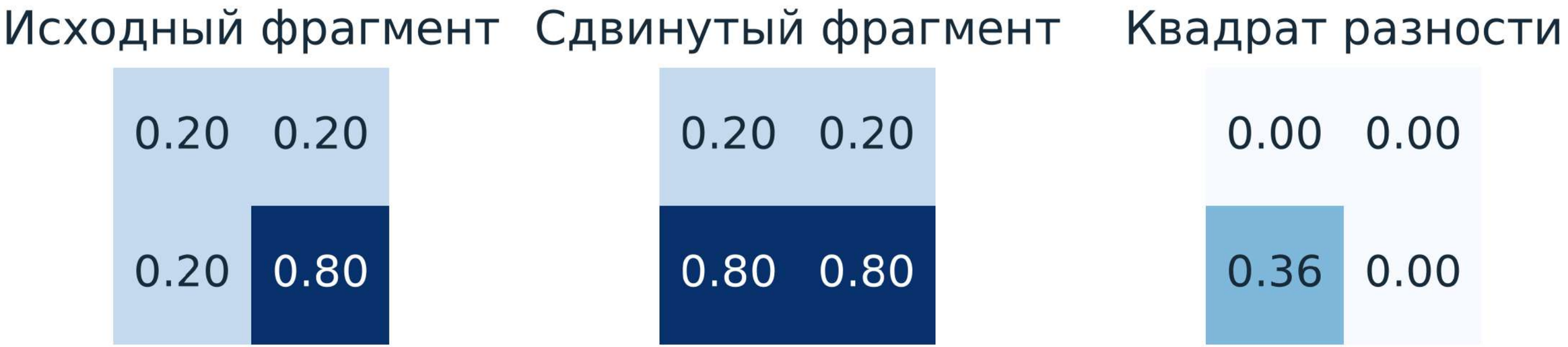


Средняя яркость
по 4 частям

Сумма квадратов различий

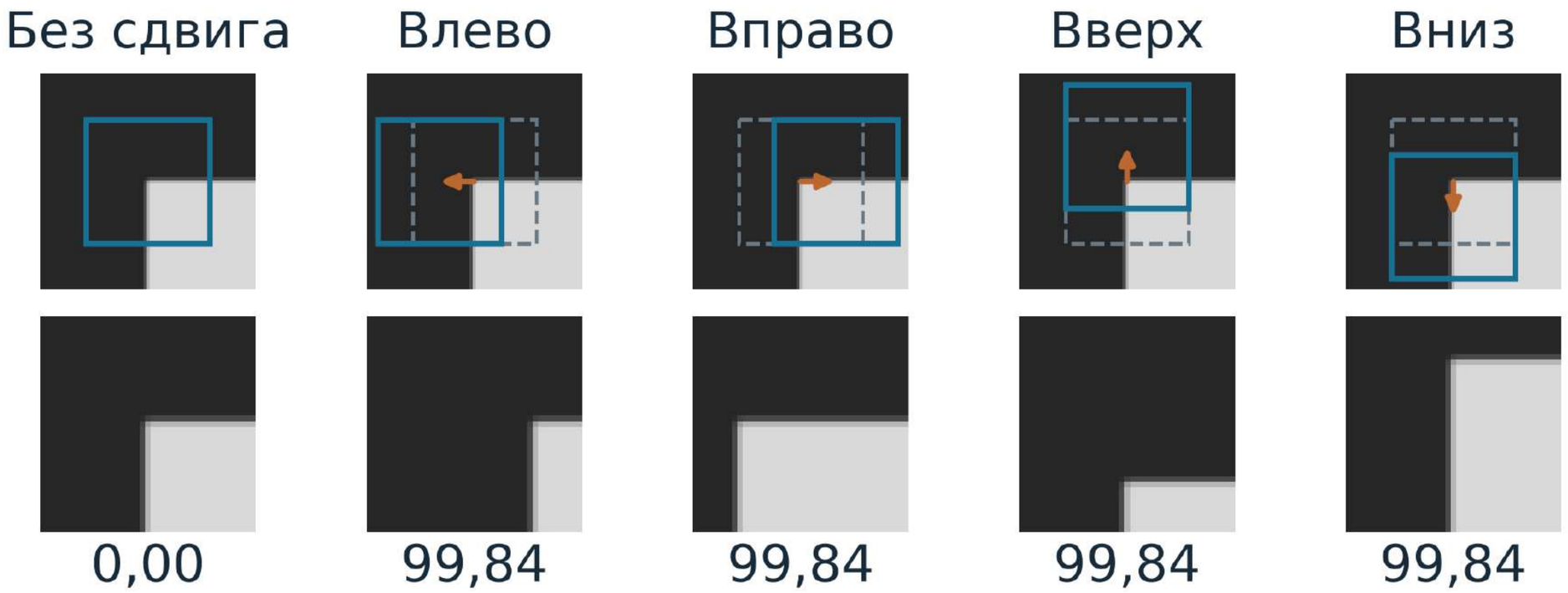
Яркости приведены к диапазону от 0 до 1. Сравниваются пиксели на одинаковых местах двух фрагментов.

$$E = \sum_{p \in W} (B_p - A_p)^2 = 0 + 0 + 0.36 + 0 = 0.36$$



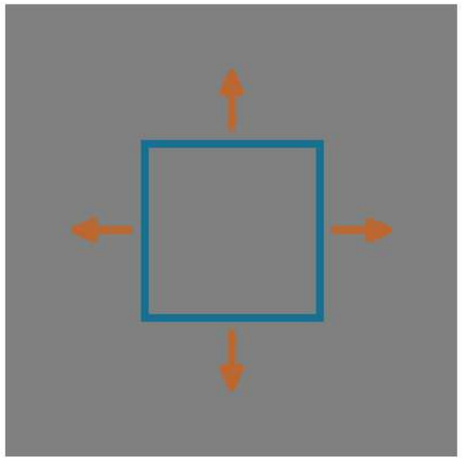
Сдвиг окна анализа

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



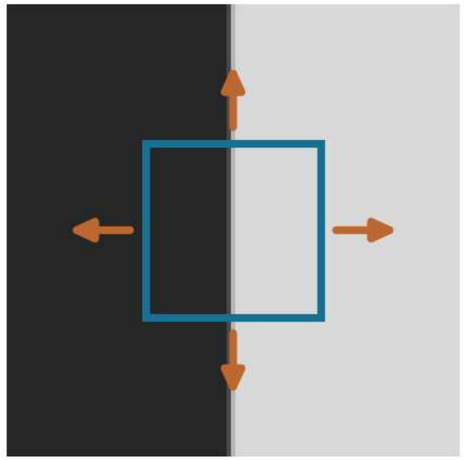
Локализуемость точки

Однородная область



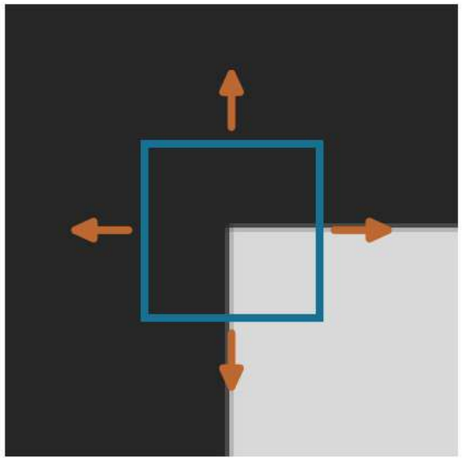
По горизонтали: 0,0
По вертикали: 0,0

Вертикальная граница



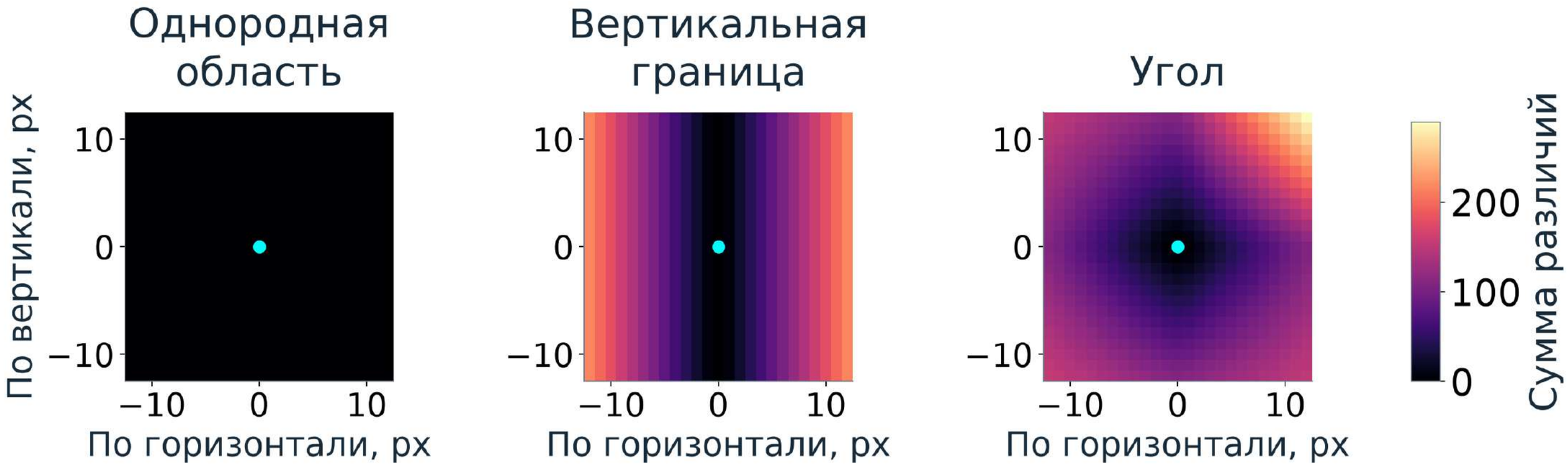
По горизонтали: 197,9
По вертикали: 0,0

Угол



По горизонтали: 99,8
По вертикали: 99,8

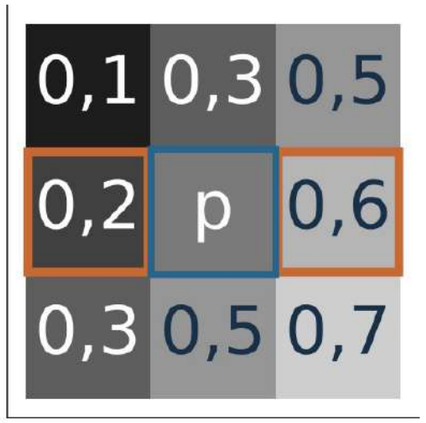
Карта ошибок сдвига



Градиент яркости

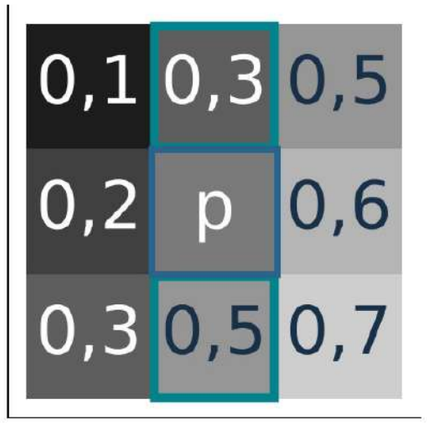
Сравниваем яркость двух соседей вокруг p. Разность делим на расстояние между ними: 2 пикселя. Получаем изменение яркости на один пиксель по x и по y.

По горизонтали



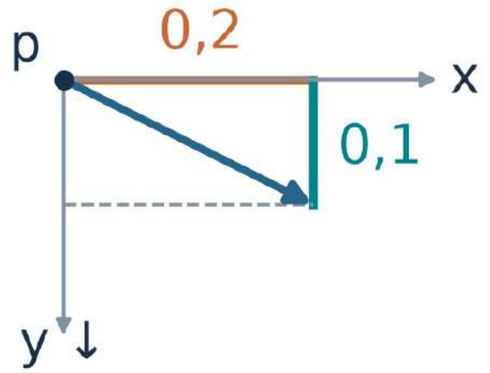
Справа − слева
Расстояние: 2 пикселя

По вертикали



Снизу − сверху
Расстояние: 2 пикселя

Вектор градиента

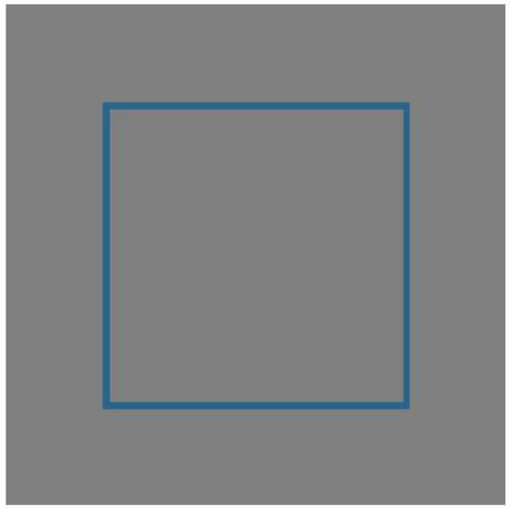


В сторону роста яркости
Ось y направлена вниз

$$I_x(p) \approx \frac{I(x+1,y) - I(x-1,y)}{2} = \frac{0{,}6 - 0{,}2}{2} = 0{,}2, \qquad I_y(p) \approx \frac{0{,}5 - 0{,}3}{2} = 0{,}1$$

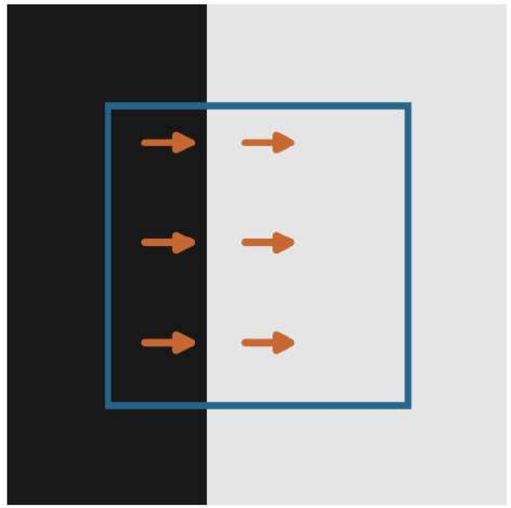
Градиенты в окрестности

Однородная область



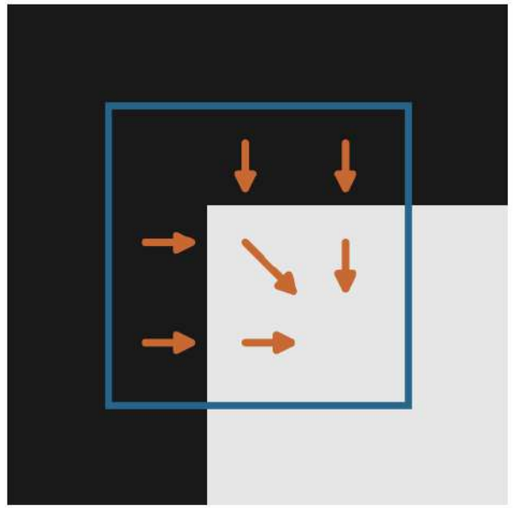
Перепадов яркости нет

Граница



Стрелки параллельны

Угол



Есть разные направления

Структурный тензор

Структурный тензор — матрица 2 × 2, составленная из градиентов окна. Она позволяет оценить, насколько изменятся пиксели при малом сдвиге.

$$\Delta I_p \approx I_x u + I_y v, \qquad E \approx \sum_{p \in W} (\Delta I_p)^2 = Au^2 + 2Buv + Cv^2$$

I_x^2

0	0	0
0,16	0,16	0
0,16	0,16	0

$I_x I_y$

0	0	0
0	0,16	0
0	0	0

I_y^2

0	0,16	0,16
0	0,16	0,16
0	0	0

Сумма A = 4 · 0,16 = 0,64 Сумма B = 1 · 0,16 = 0,16 Сумма C = 4 · 0,16 = 0,64

$$M = \begin{bmatrix} A & B \\ B & C \end{bmatrix} = \begin{bmatrix} 0{,}64 & 0{,}16 \\ 0{,}16 & 0{,}64 \end{bmatrix}$$

Собственные значения

Сравниваем сдвиги длиной 0,1 пикселя во всех направлениях. Минимальная и максимальная ошибки модели, делённые на $0{,}1^2$, — собственные значения M.

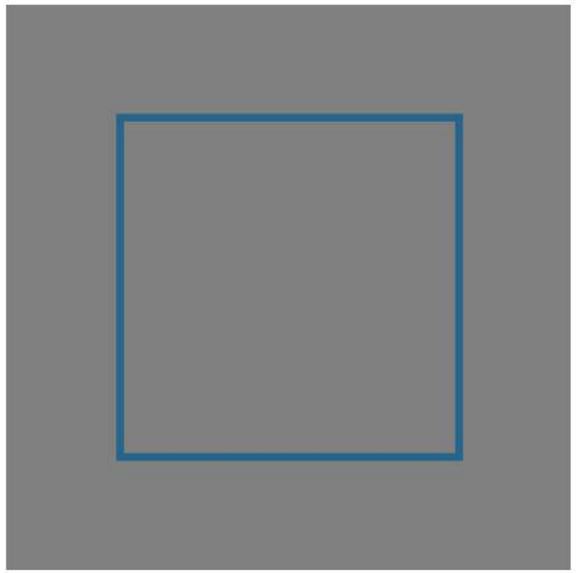


$$\lambda_{\min} = \frac{0{,}0048}{0{,}1^2} = 0{,}48, \qquad \lambda_{\max} = \frac{0{,}0080}{0{,}1^2} = 0{,}80$$

Угол, граница и однородная область

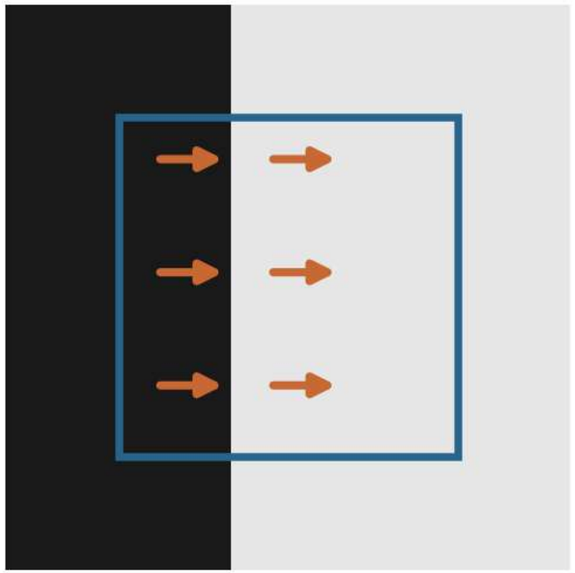
Стрелки показывают рост яркости. У границы градиенты параллельны: вдоль неё сдвиг незаметен. У угла есть разные направления.

Однородная область



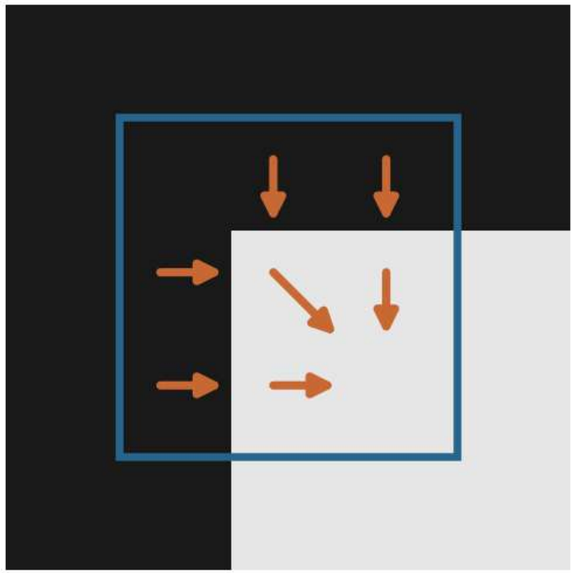
Минимум: 0,00
Максимум: 0,00

Граница



Минимум: 0,00
Максимум: 0,96

Угол

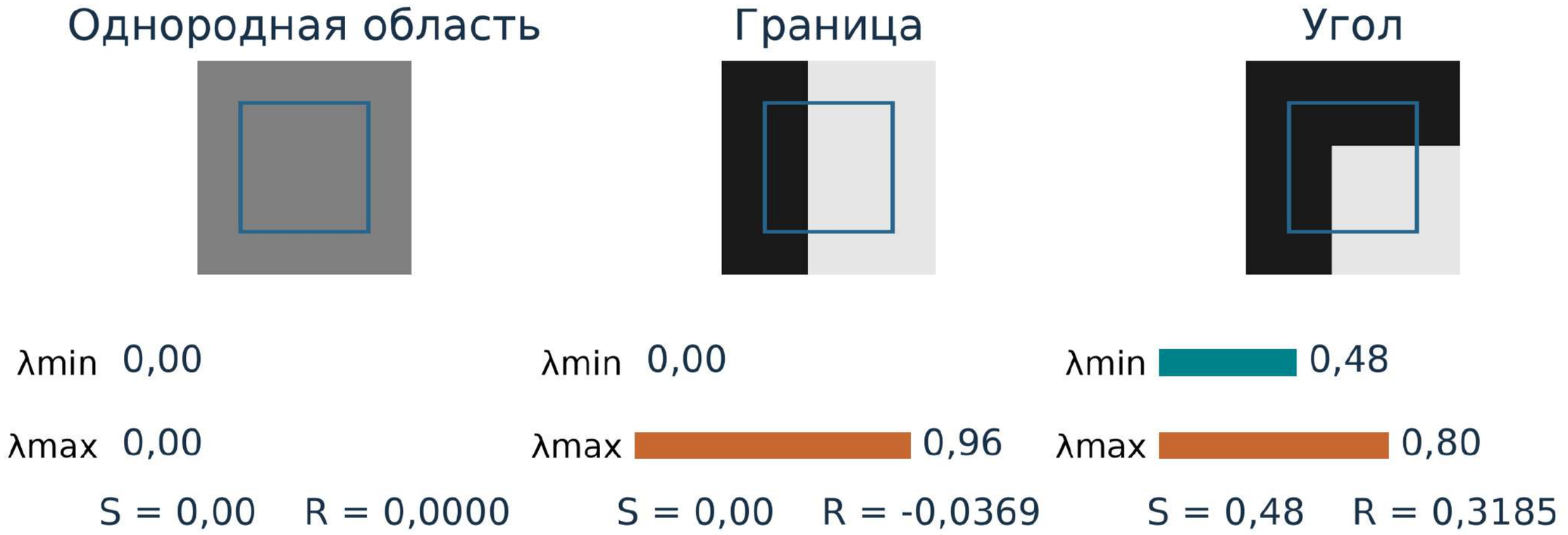


Минимум: 0,48
Максимум: 0,80

Отклики Harris и Shi–Tomasi

Отклик — одно число для выбора центра. У границы остаётся незаметное направление; у угла заметен любой малый сдвиг.

$$S = \lambda_{\min}, \qquad R = \lambda_{\min}\lambda_{\max} - 0{,}04(\lambda_{\min} + \lambda_{\max})^2$$

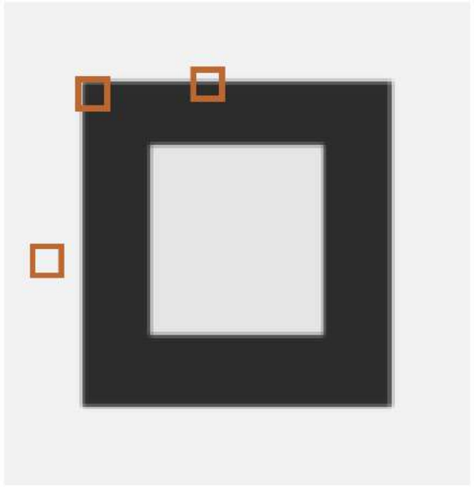


Карта отклика

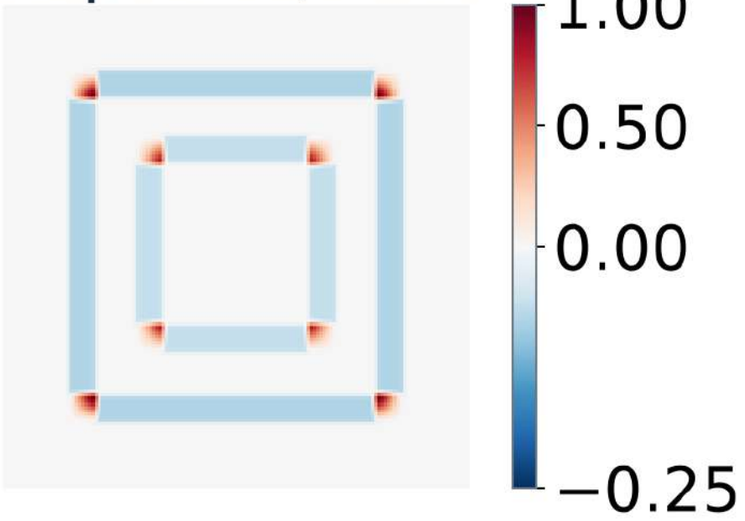
Красные пятна — сильные угловые окрестности; синие полосы — границы. Карта имеет те же координаты, что и изображение.

$$\widetilde{R}(x,y) = \frac{R(x,y)}{\max_{x,y} R(x,y)}$$

Окно 9 × 9 в каждом центре



Карта оценки

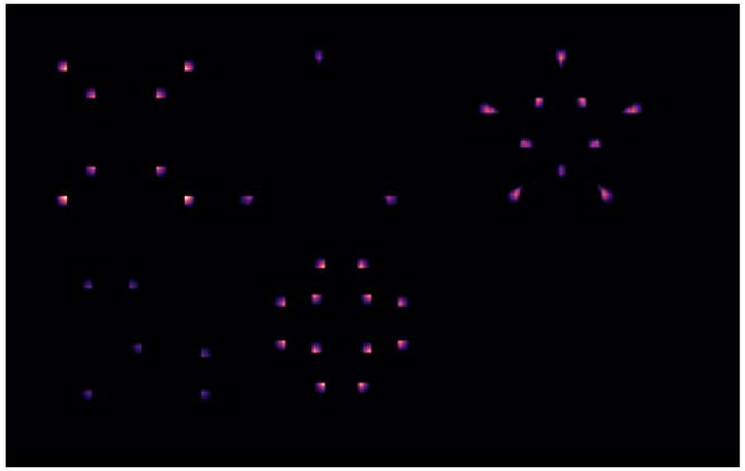


Порог и подавление немаксимумов

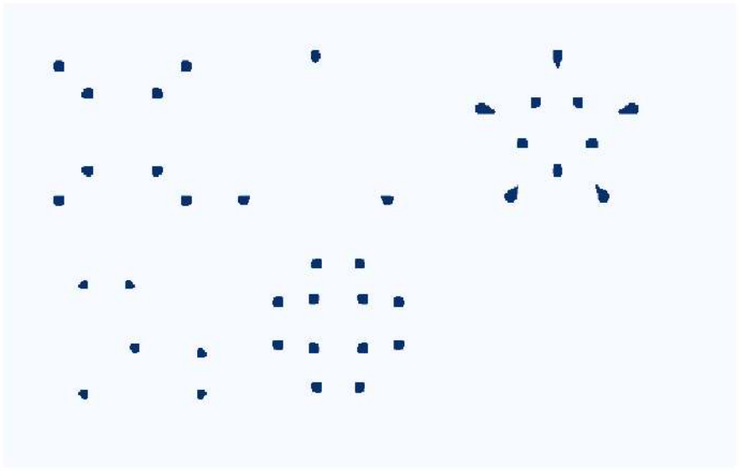
Порог 0,08 оставляет 2250 пикселей. Локальный отбор сохраняет 39 центров — с наибольшей оценкой в своей окрестности.

$$\widetilde{R}(x,y) > 0.08, \qquad \widetilde{R}(x,y) = \max_{(a,b)\in\mathcal{N}(x,y)} \widetilde{R}(a,b)$$

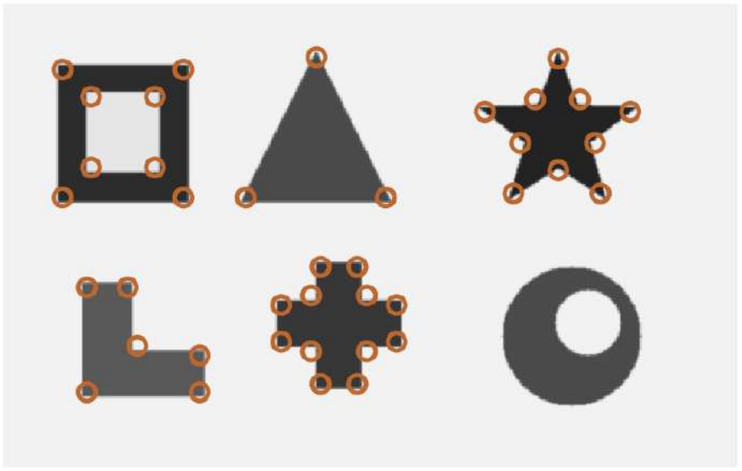
Положительная часть



Выше порога

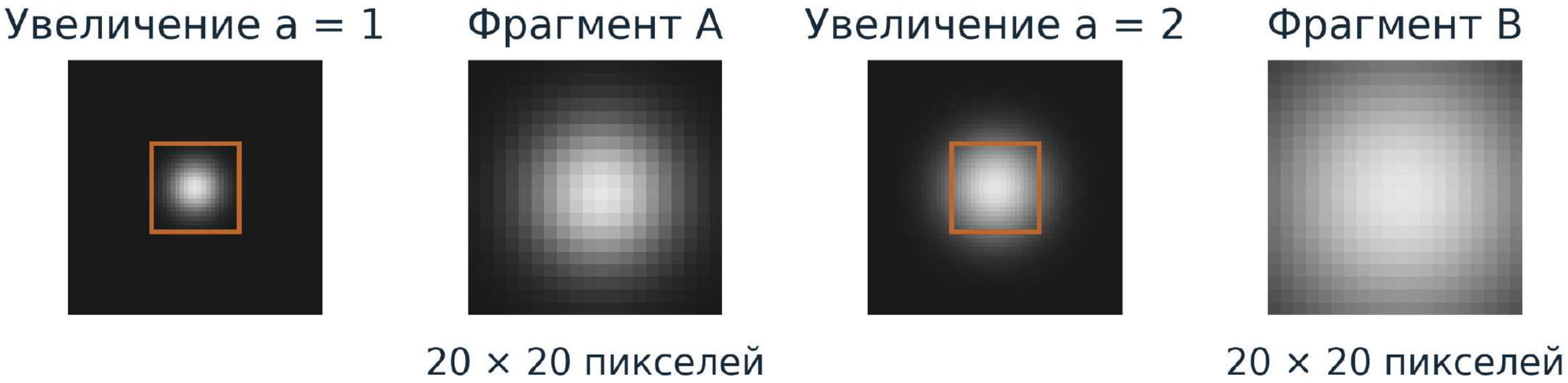


39 максимумов



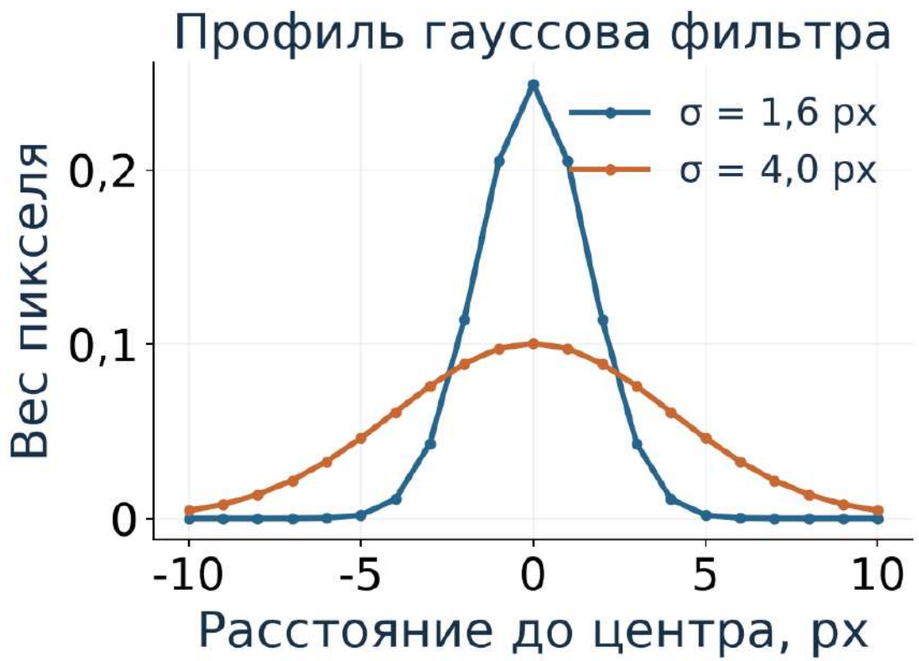
Масштаб изображения

Средние яркости в фиксированной области меняются при увеличении детали. Коэффициент увеличения a меняет её размер; для сопоставления нужно соразмерно выбирать область описания.



Гауссово сглаживание

σ задаёт ширину пространственного фильтра в пикселях. Значения σ выбирает алгоритм; это не разброс яркостей изображения.



σ = 1,6 px



Те же координаты

σ = 4,0 px



Те же координаты

$$L(x, y, \sigma) = (G_\sigma * I)(x, y), \qquad G_\sigma \propto \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Гауссова пирамида

Октава объединяет уровни, на которых σ возрастает вдвое. Один уровень — одно сглаженное изображение.

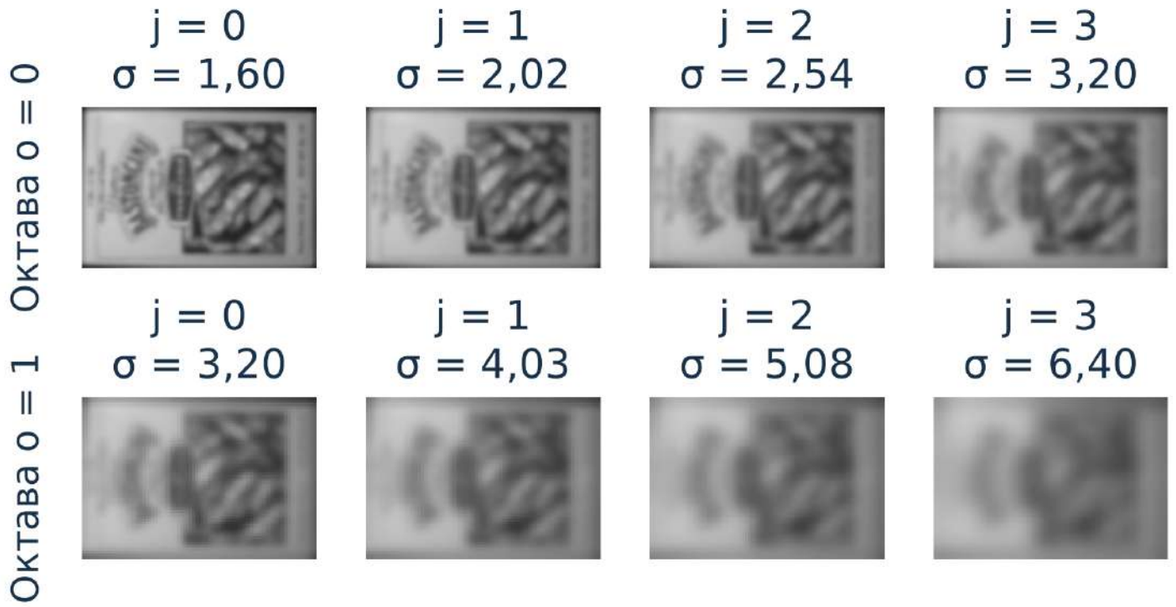
$$\sigma_{o,j} = \sigma_0\, 2^o k^j = 1{,}6 \cdot 2^o \cdot 2^{j/3}, \qquad k = 2^{1/3} \approx 1{,}26$$

$\sigma_0 = 1{,}6$ px — начальное сглаживание.

o — номер октавы. Множитель 2^o переводит σ в исходные пиксели.

j — индекс уровня внутри октавы. Здесь 3 интервала на удвоение σ.

Все σ справа — в исходных пикселях.

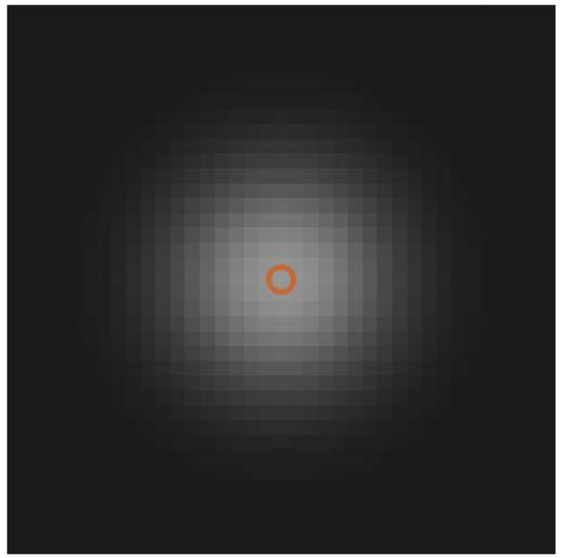


$$\sigma_{1,2} = 1{,}6 \cdot 2^1 \cdot 2^{2/3} \approx 5{,}08 \text{ px}$$

Разность гауссианов — DoG

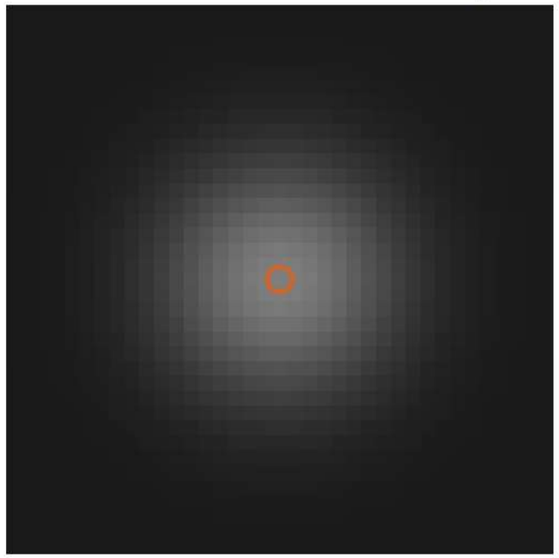
Вычитаем два размытия одного изображения попиксельно. В центре светлой детали яркость уменьшается: разность отрицательна.

Размытие σ = 3,20



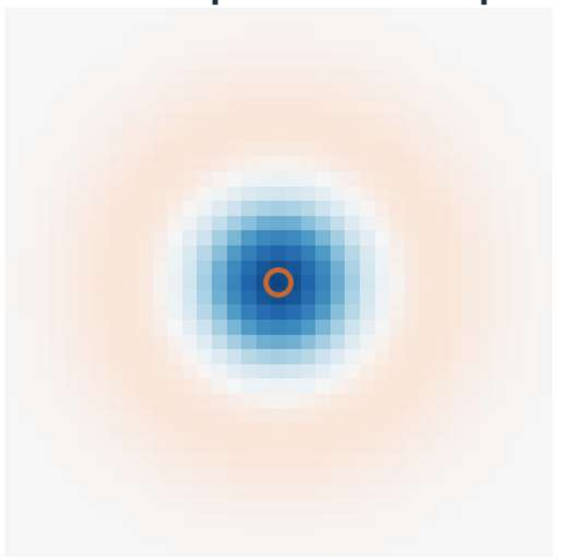
В центре: 0,5878

Размытие σ = 4,03



В центре: 0,4969

DoG: второе − первое



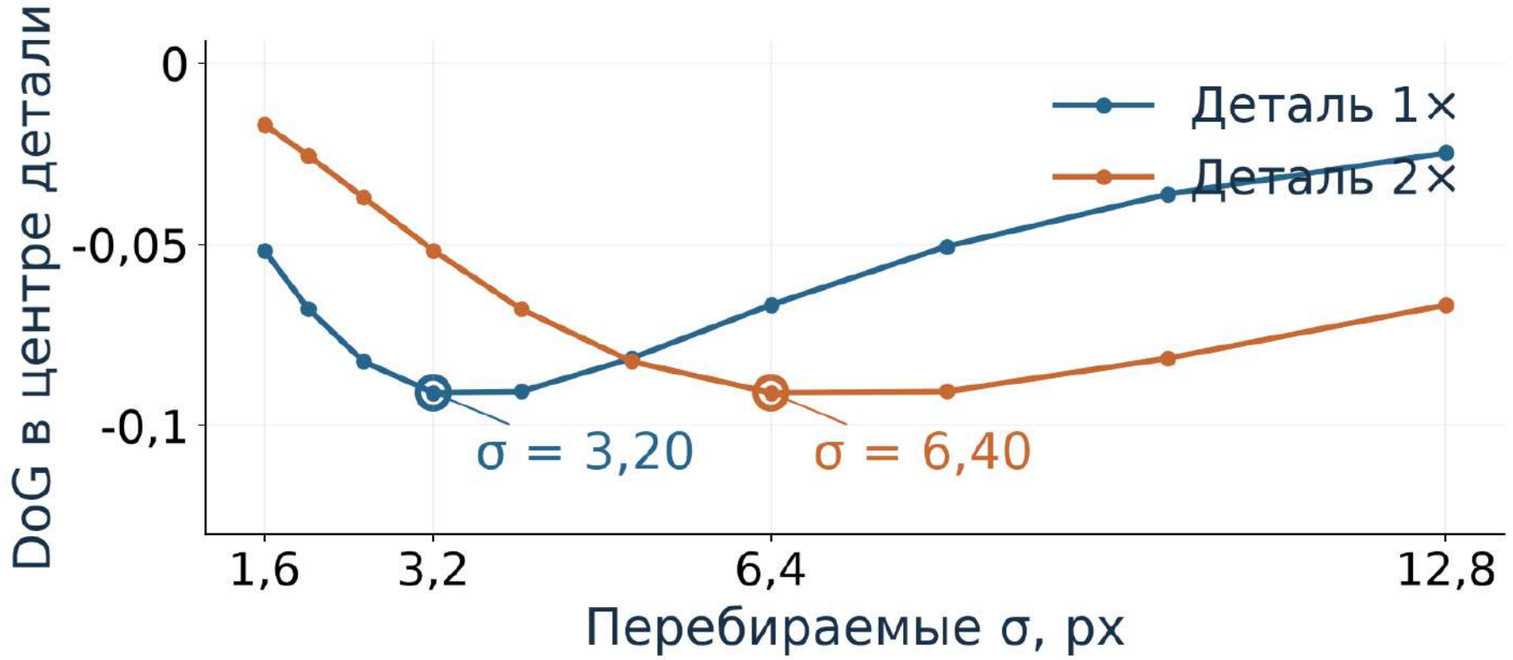
В центре: -0,0910

$$D = L(k\sigma) - L(\sigma), \qquad D(p) = 0{,}4969 - 0{,}5878 = -0{,}0910$$

Выбор характерного масштаба

Проверяем одну деталь при разных σ. Выбираем локальный экстремум DoG: для увеличенной вдвое детали он возникает при большей σ.

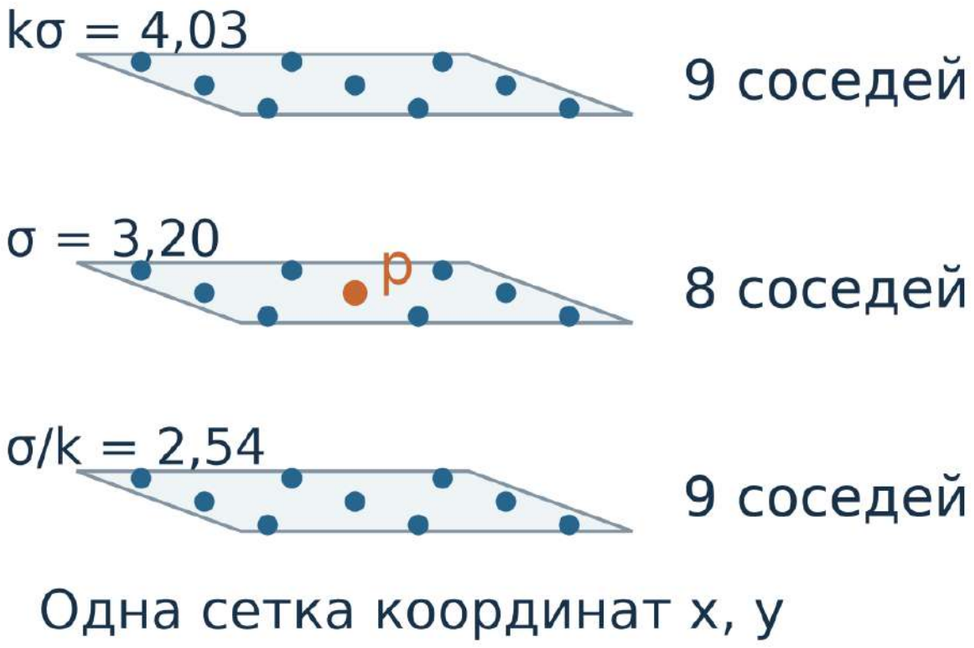




$$\sigma_1 = 3{,}20 \text{ px}, \qquad \sigma_2 = 6{,}40 \text{ px}$$

Экстремум в пространстве масштабов

Сравниваем три соседних слоя DoG одной октавы: σ/k, σ и kσ. Пиксельная сетка одинаковая; уровни сглаживания разные.



В своём слое p не сравниваем с самой собой.

В соседних слоях центральный пиксель тоже участвует.

$8 + 9 + 9 = 26$

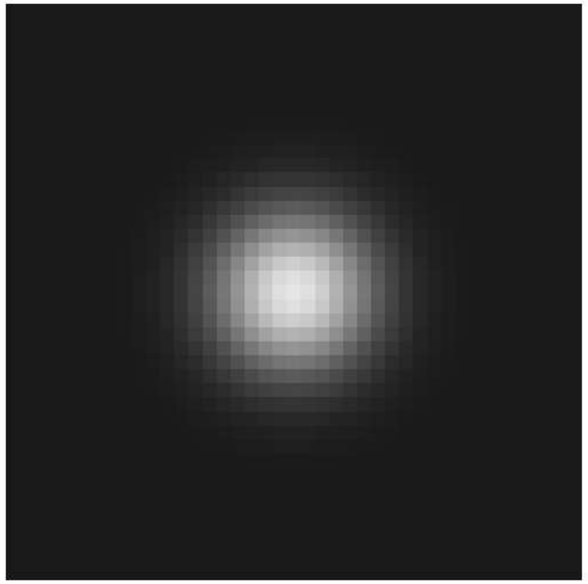
p меньше всех 26 соседей

$$D(p) = -0{,}0910 < -0{,}0907 = \min_{q \in \mathcal{N}(p)} D(q)$$

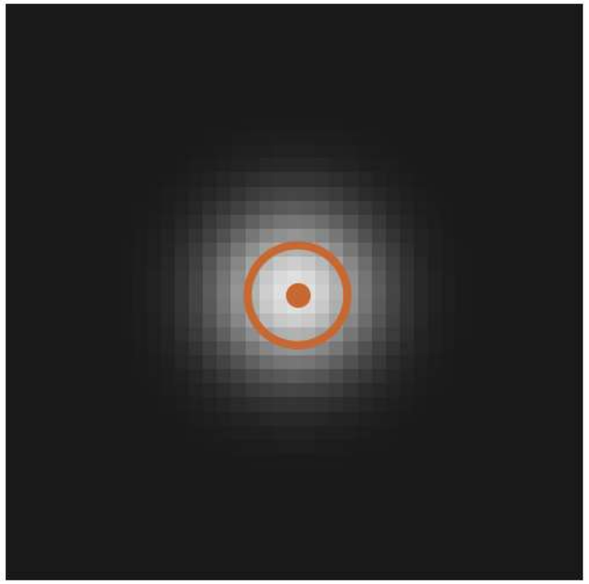
Масштаб ключевой точки SIFT

После отбора экстремума уточняются центр и σ. В OpenCV точка хранит диаметр 2σ; область для дескриптора строится по этому масштабу.

Исходная деталь



Центр и масштаб



x = 64,23 px

y = 64,23 px

σ = 3,55 px

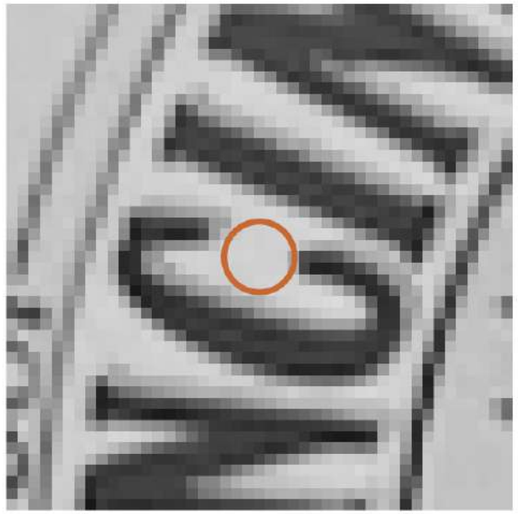
Диаметр = 7,10 px

$$\mathrm{KeyPoint.size} = 2\sigma$$

Масштаб ключевой точки

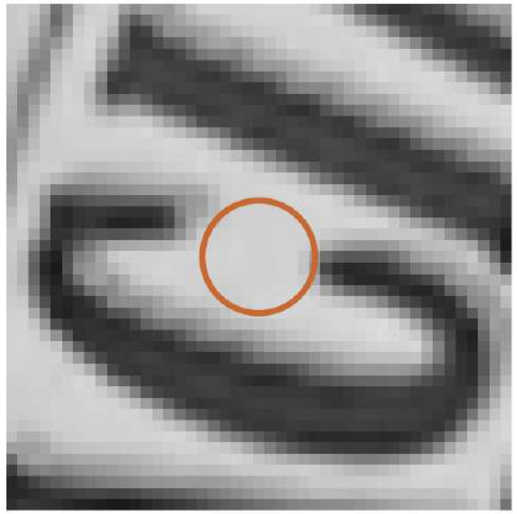
Изображение увеличено в 1,6 раза; диаметр найденной точки — в 1,59 раза. Уточнение экстремума даёт промежуточное значение σ.

Фрагмент исходного кадра



σ = 3,14 px; d = 6,29 px

После увеличения × 1,6



σ = 5,01 px; d = 10,02 px

Обе вырезки: 45 × 45 пикселей; окружность имеет диаметр d

$$\sigma_* = \sigma_0\, 2^o\, 2^{(j+\delta)/3}, \qquad d = 2\sigma_*$$

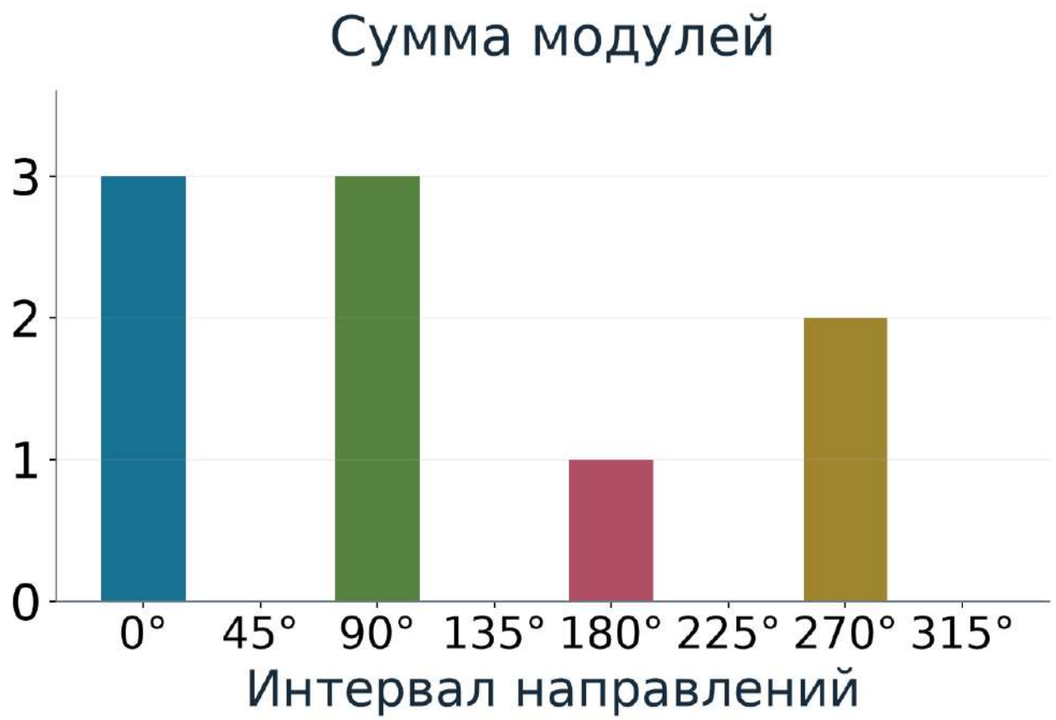
δ — уточнение уровня

Гистограмма направлений

Каждый градиент добавляет свой модуль в столбец соответствующего интервала. Два градиента направления 0° дают 2 + 1 = 3; направления 90° — 1 + 2 = 3.

$$m = \sqrt{I_x^2 + I_y^2}, \qquad \alpha = \operatorname{atan2}(I_y, I_x), \qquad h_b = \sum_{p:\, \alpha_p \in b} m_p$$

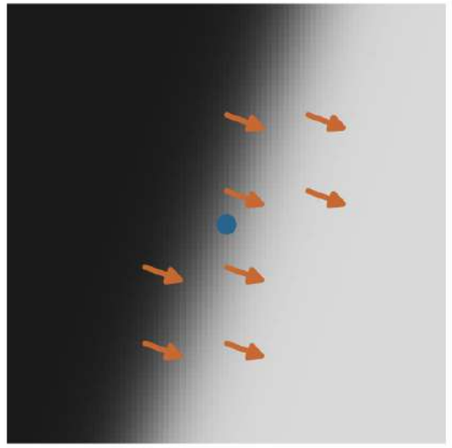




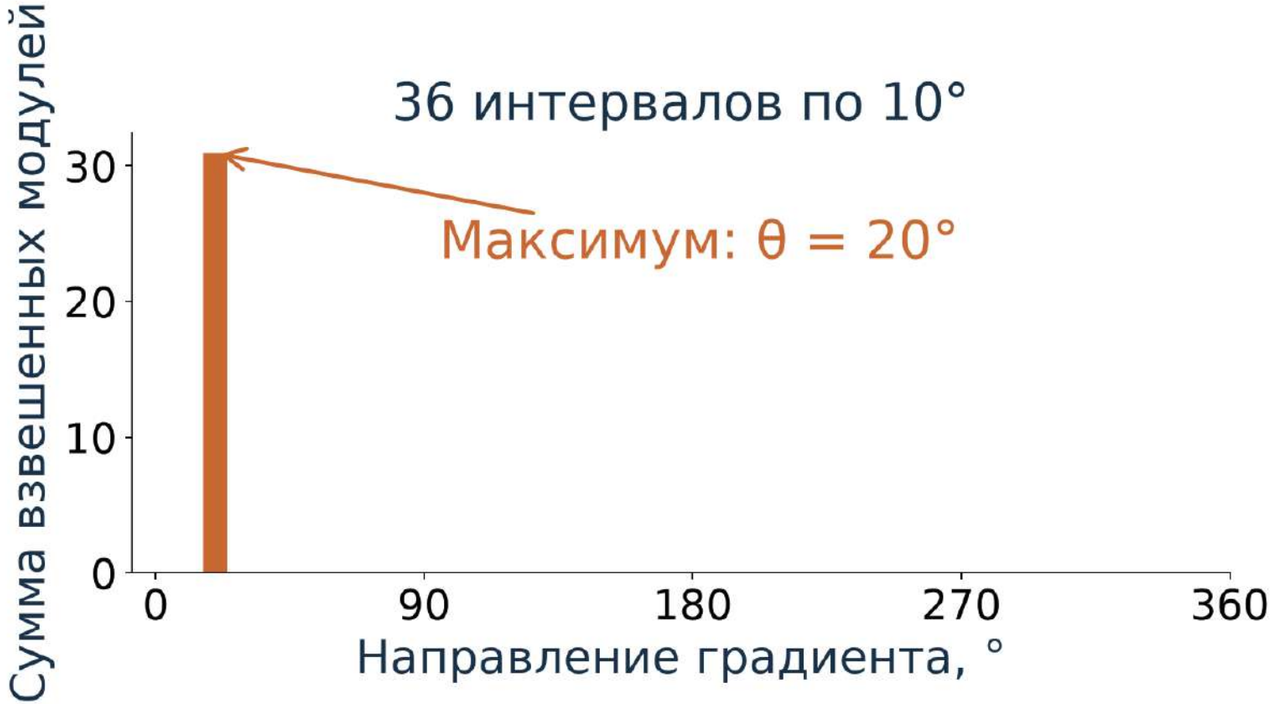
SIFT: выбор ориентации

Складываем модули градиентов по направлениям, сильнее учитывая ближние пиксели.
Направление главного пика задаёт угол θ.

Градиенты около заданного центра



Ось y направлена вниз



$$h_b = \sum_{p:\alpha_p \in b} w_p \sqrt{I_x(p)^2 + I_y(p)^2}, \qquad \theta = \arg\max_{\alpha} h(\alpha)$$

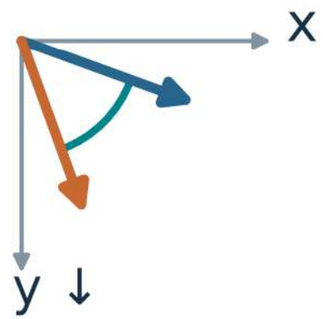
Относительное направление градиента

θ берём из пика гистограммы. Из угла каждого градиента вычитаем θ: совместный поворот детали и её ориентации сокращается.

Исходный фрагмент

θ = 20° — пик гистограммы

α = 70° — один градиент

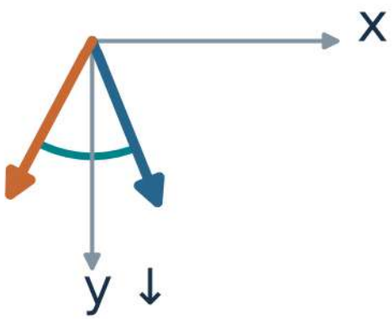


70° − 20° = 50°

Поворот фрагмента на 48°

θ = 68° — пик гистограммы

α = 118° — один градиент



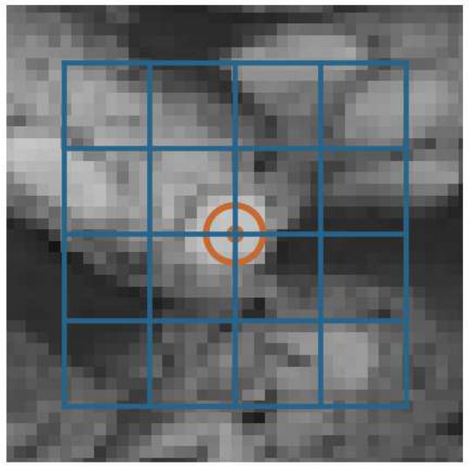
118° − 68° = 50°

$$\alpha_{\text{rel}} = (\alpha - \theta) \bmod 360°, \qquad 70° - 20° = 118° - 68° = 50°$$

SIFT: область дескриптора

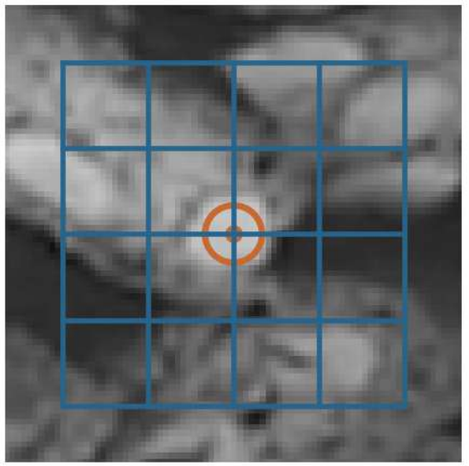
Найденная σ задаёт ширину ячейки. У более крупной детали область шире, но сетка всегда содержит 4 × 4 ячейки.

Исходная фотография



σ ≈ 2,5 px; ячейка ≈ 7,5 px
Сетка: 4 × 4; ширина ≈ 30,1 px

Фотография увеличена вдвое



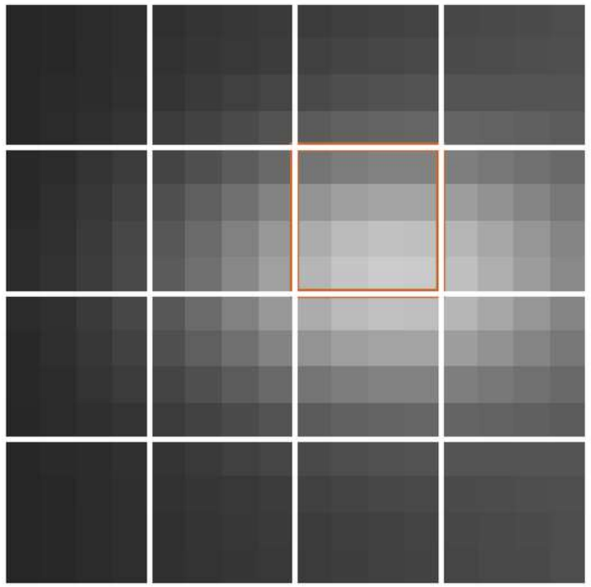
σ ≈ 5,0 px; ячейка ≈ 15,1 px
Сетка: 4 × 4; ширина ≈ 60,3 px

$$w_{\text{cell}} = 3\sigma, \qquad w_{\text{grid}} = 4 \cdot 3\sigma = 12\sigma$$

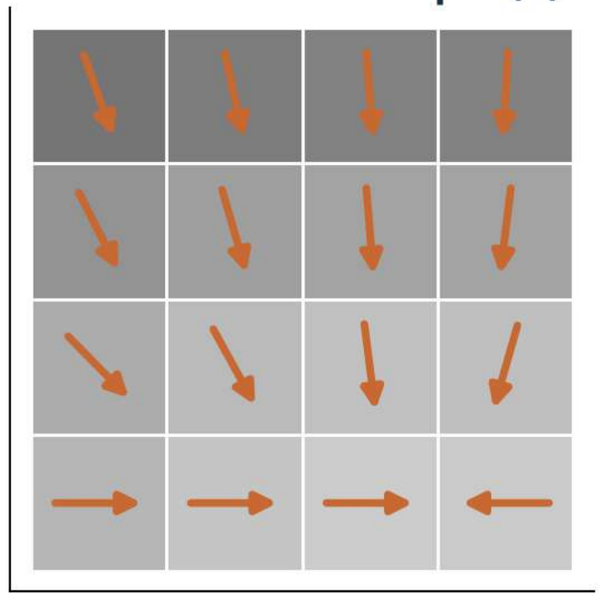
SIFT: пространственные ячейки

В каждой ячейке учитываем градиент каждого пикселя. Здесь ячейка содержит 4 × 4 отсчёта; в OpenCV их число зависит от σ.

Фрагмент: 16 × 16 отсчётов



Одна ячейка: 16 градиентов

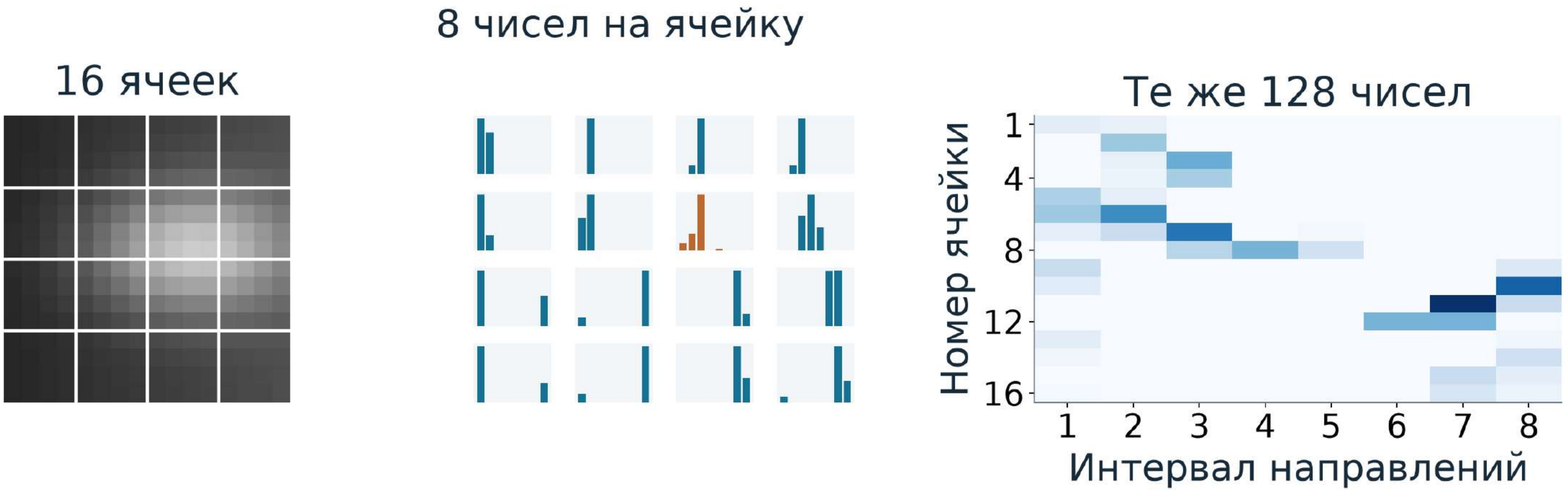


Ось y вниз; длины стрелок выровнены

SIFT: вектор дескриптора

Ячейки сохраняют положение градиентов. Их гистограммы записываются в один вектор. У мини-графиков свой вертикальный масштаб; тепловая карта справа имеет общую шкалу.

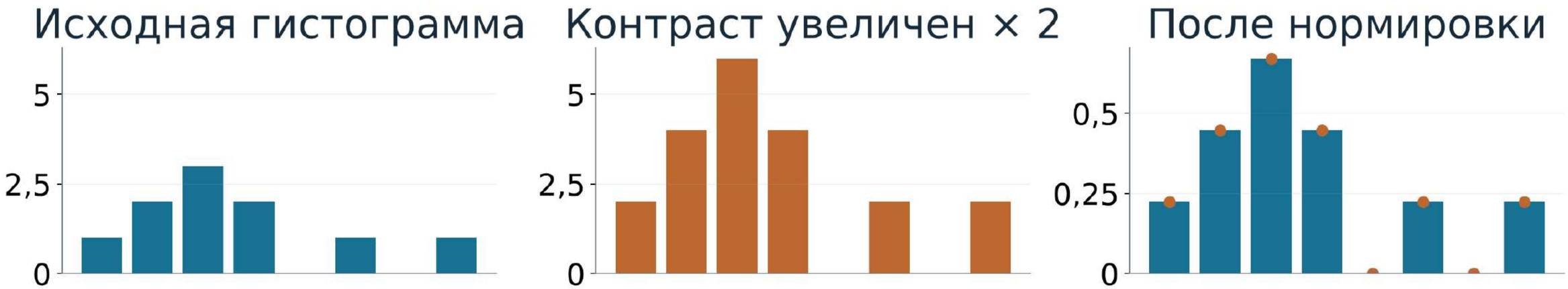
$$4 \times 4 \times 8 = 128, \qquad \mathbf{d} = (h_{1,1}, \ldots, h_{1,8}, h_{2,1}, \ldots, h_{16,8})$$



SIFT: нормировка

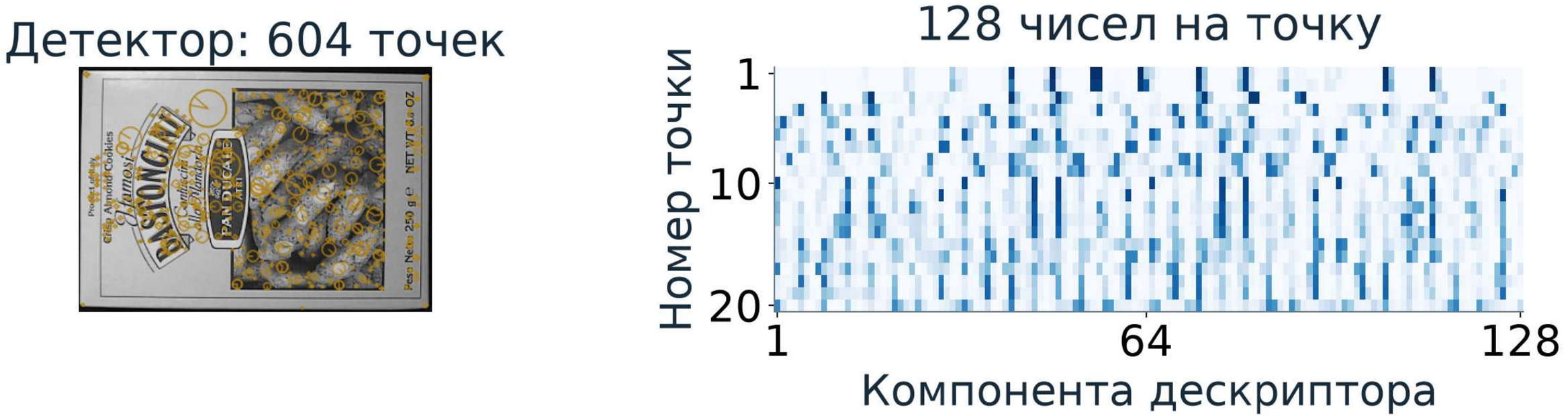
Умножение яркостей на одну константу масштабирует все градиенты. После деления вектора на его длину оба описания совпадают.

$$\widehat{\mathbf{d}} = \frac{\mathbf{d}}{\|\mathbf{d}\|_2}, \qquad \frac{2\mathbf{d}}{\|2\mathbf{d}\|_2} = \widehat{\mathbf{d}}$$



SIFT: результат вычисления

Слева координаты, масштаб и ориентация; справа первые 20 строк описаний. Строка таблицы относится к точке с тем же индексом в списке.

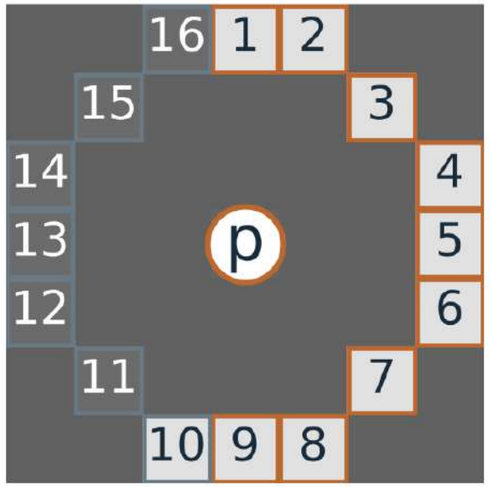


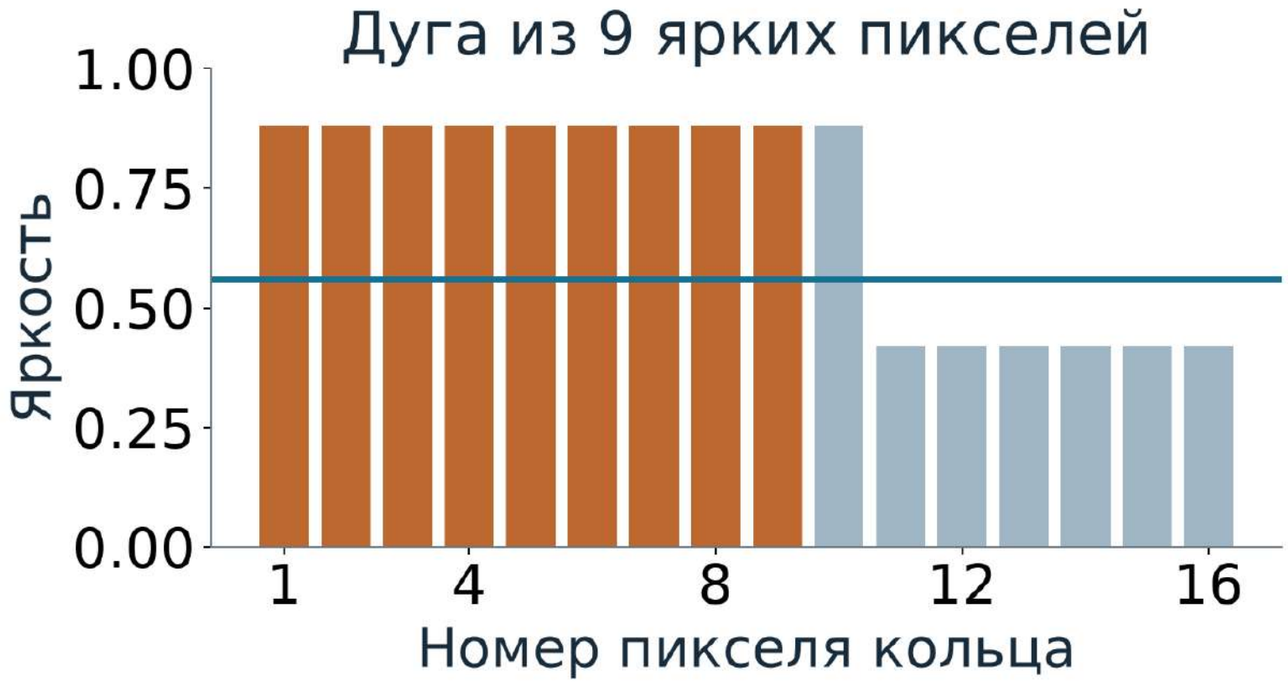
FAST: сегментный тест

Центр имеет яркость 0,40. Девять подряд идущих пикселей кольца должны быть ярче 0,56 либо все темнее 0,24.

$$I(p) = 0.40, \quad t = 0.16, \quad \underbrace{I(q) > 0.56}_{I(p)+t} \quad \vee \quad \underbrace{I(q) < 0.24}_{I(p)-t}$$

Кольцо вокруг центра

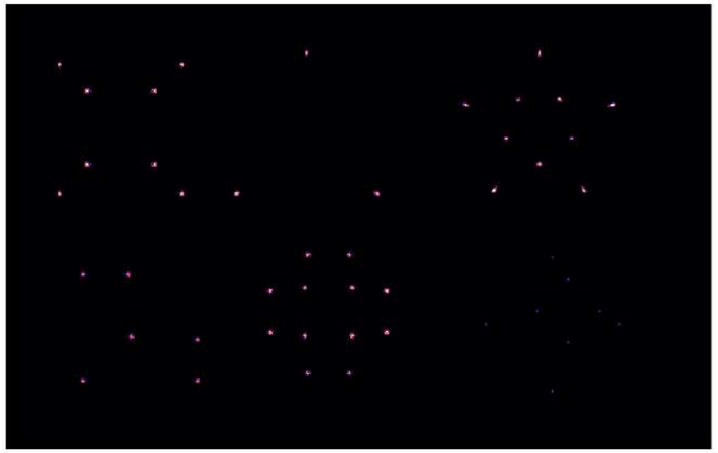




FAST: координаты точек

Оценка показывает, какой порог выдерживает лучшая непрерывная дуга. После порога соседние кандидаты прореживаются по локальным максимумам.

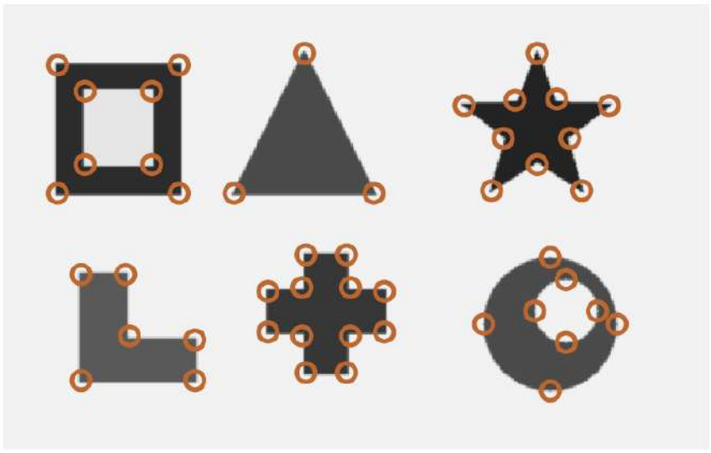
Карта оценки



Выше порога



47 центров

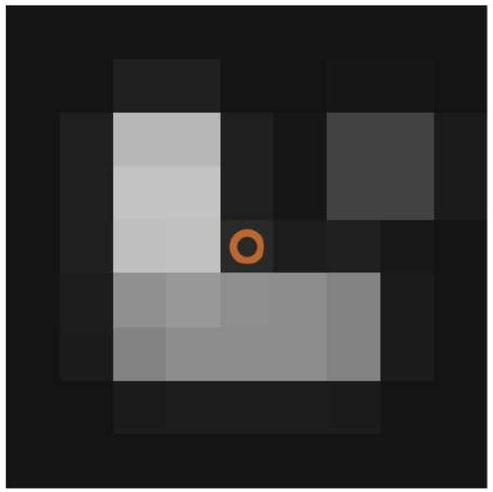


BRIEF: сравнение яркостей

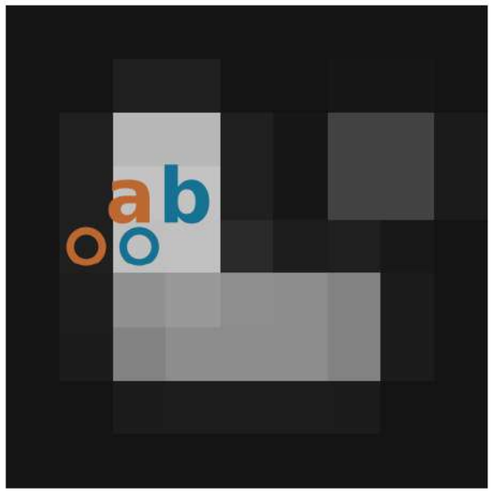
Две позиции заданы относительно центра. Если первая темнее второй, записываем 1; иначе 0. Перед сравнениями фрагмент слегка сглаживается.

$$b_j = \mathbf{1}[I(\mathbf{p} + \mathbf{a}_j) < I(\mathbf{p} + \mathbf{b}_j)] \in \{0, 1\}$$

Заданный центр



Сравнение яркостей



Получается один бит

Яркость a: 33.4

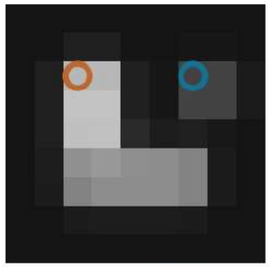
Яркость b: 192.2

a темнее b → 1

BRIEF: бинарное описание

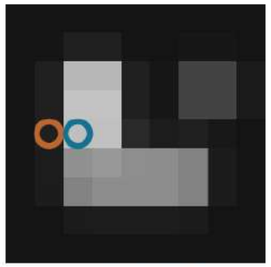
В каждой паре сравниваем оранжевую и синюю позиции. Порядок сравнений фиксирован; полученные биты записываются слева направо.

Сравнение 1



$183 \geq 67 \rightarrow 0$

Сравнение 2



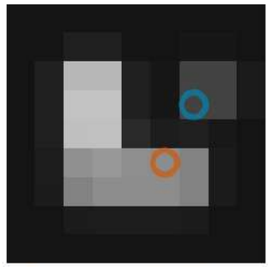
$33 < 192 \rightarrow 1$

Сравнение 3



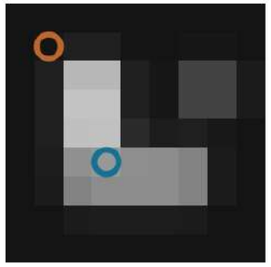
$196 \geq 24 \rightarrow 0$

Сравнение 4



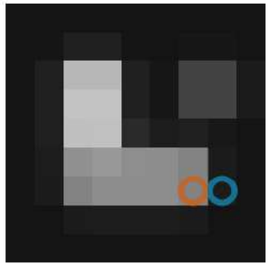
$141 \geq 67 \rightarrow 0$

Сравнение 5



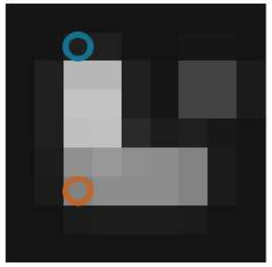
$21 < 153 \rightarrow 1$

Сравнение 6



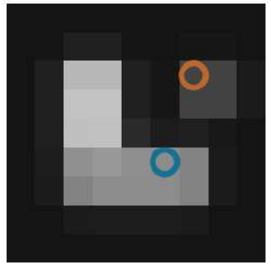
$132 \geq 29 \rightarrow 0$

Сравнение 7



$132 \geq 33 \rightarrow 0$

Сравнение 8

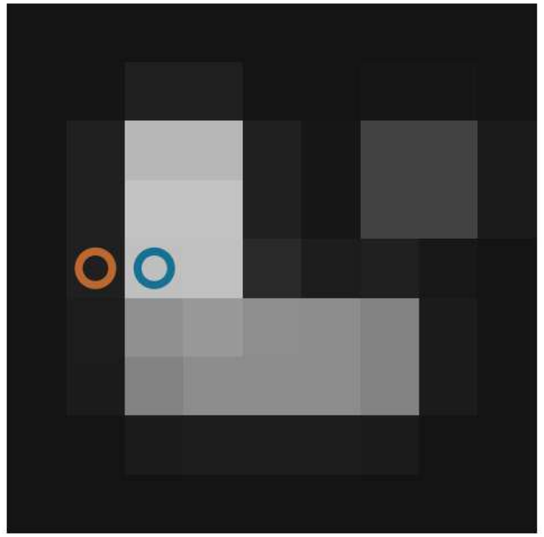


$67 < 141 \rightarrow 1$

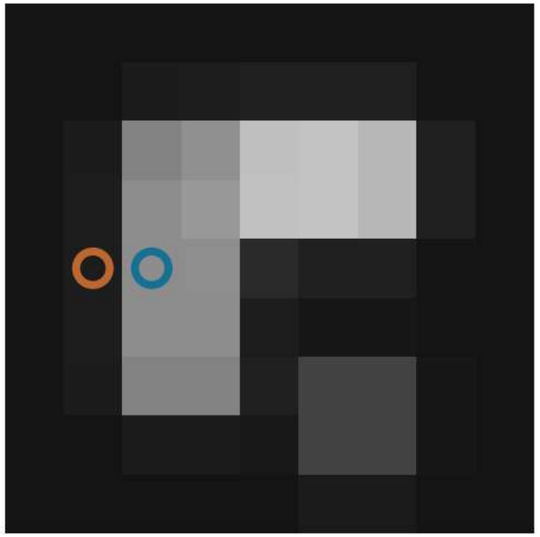
BRIEF: чувствительность к повороту

Центр совпадает, но прежние позиции попадают на другие части фигуры. Из-за этого сравнения могут дать другие биты для той же детали.

Фрагмент A



Фрагмент B: 90°



Пары не повёрнуты

A	0	1	0	0	1	0	0	1
B	1	1	1	1	1	0	0	0

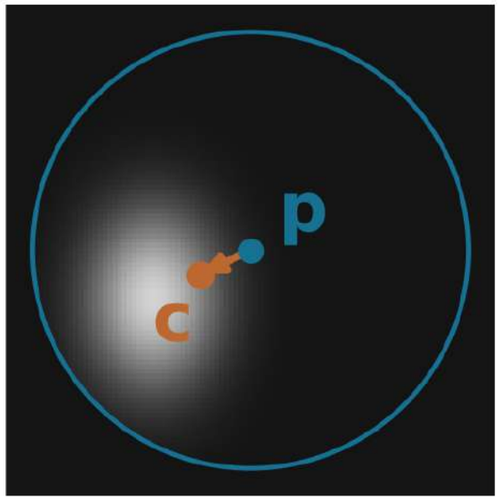
Различаются 4 из 8 битов

ORB: ориентация

p — центр ключевой точки; c — центр яркости круговой окрестности. Односторонний вектор от p к c задаёт угол ориентации; более яркие пиксели сильнее смещают c.

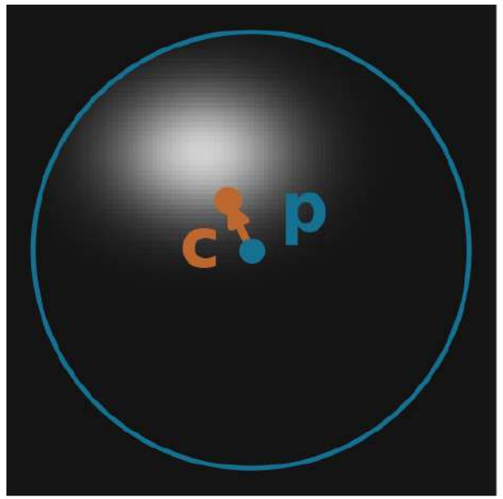
$$\mathbf{c} = \frac{\sum_{\mathbf{q}\in W} I(\mathbf{q})\,\mathbf{q}}{\sum_{\mathbf{q}\in W} I(\mathbf{q})}, \qquad \theta = \operatorname{atan2}(c_y - p_y, c_x - p_x)$$

Исходная окрестность



Ориентация: 154.3°

Поворот на 90°

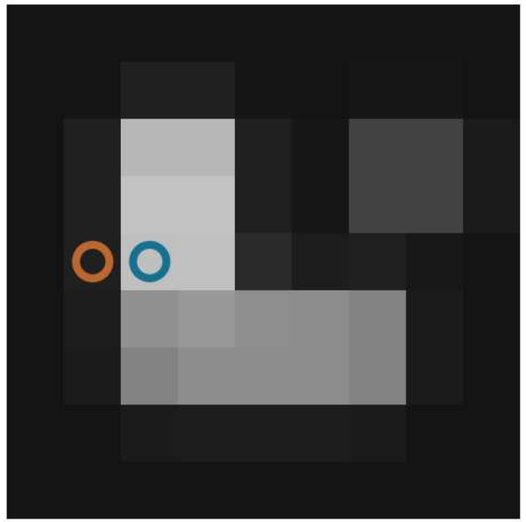


Ориентация: 244.3°

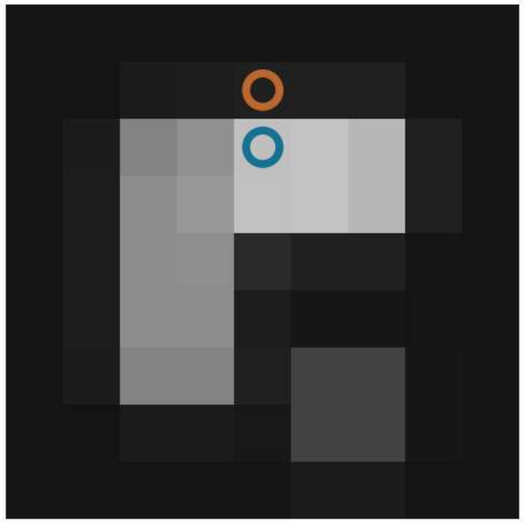
ORB: поворот пар пикселей

Обе позиции каждой пары меняются согласованно. Они снова попадают на соответствующие части детали, и в этом точном примере биты совпадают.

Пара в A



Пара в B: повёрнута



Пары согласованы

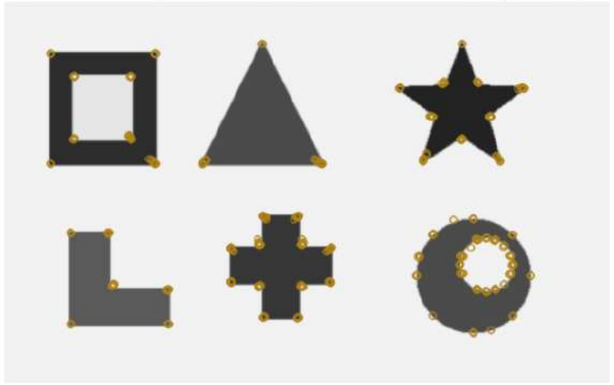
A	0	1	0	0	1	0	0	1
B	0	1	0	0	1	0	0	1

Различаются 0 из 8 битов

ORB: результат вычисления

FAST проверяет кольцо фиксированного радиуса и сам не устойчив к масштабу. ORB применяет его на уровнях пирамиды, отбирает точки по Harris и вычисляет ориентированные бинарные описания.

Детектор: 299 координат





Детекторы и дескрипторы

Метод	Выбор координат	Описание окрестности
Harris / Shi–Tomasi	Карта оценки → порог → максимумы	Не вычисляет
FAST	Дуга на кольце → прореживание	Не вычисляет
BRIEF	Принимает готовые точки	Бинарные сравнения яркостей
SIFT	Экстремумы DoG по координатам и масштабу	128 компонент гистограмм
ORB	FAST в пирамиде; отбор точек	Ориентированные сравнения rBRIEF

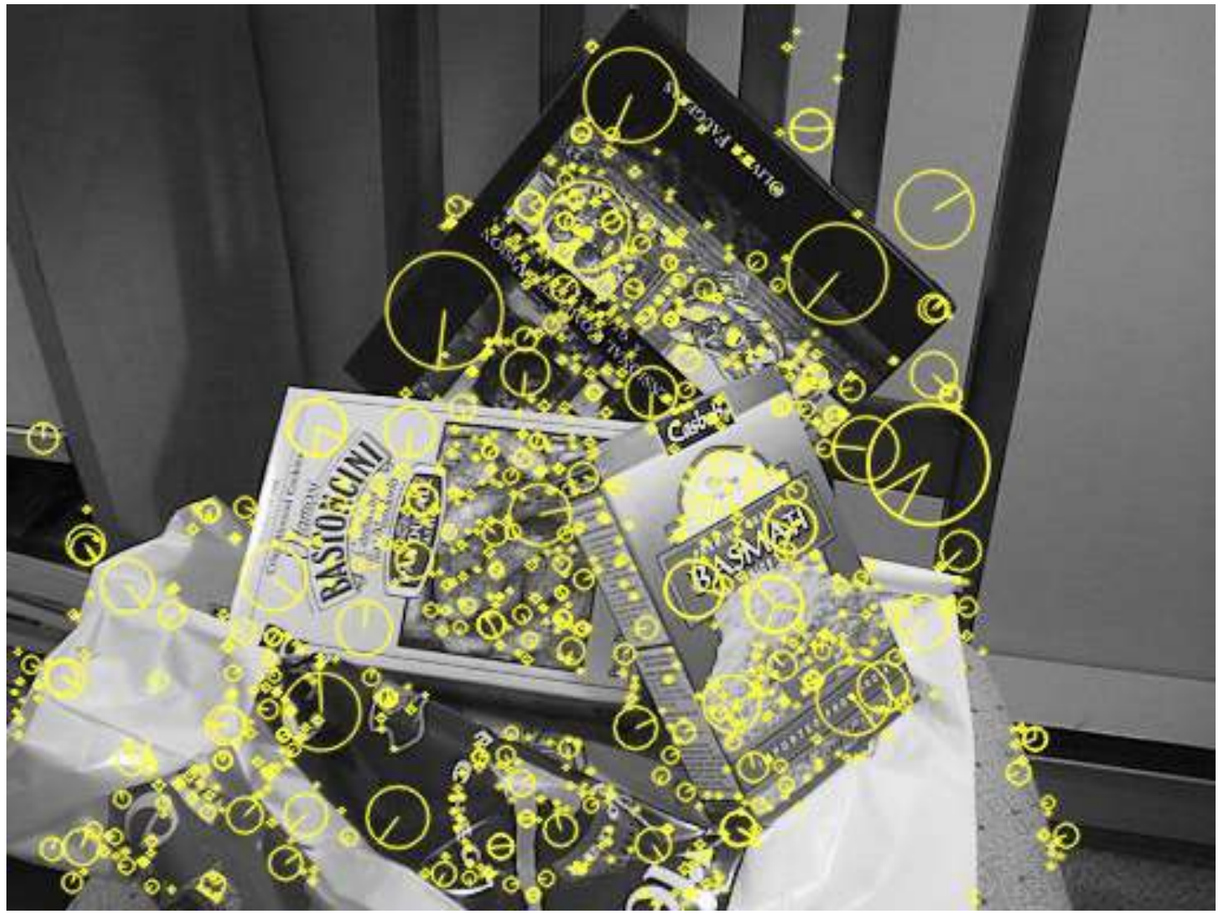
Вычислительная стоимость локальных признаков

SIFT вычисляет градиенты и гистограммы для каждого описания. Для большого числа кадров полезно сократить объём данных и заменить часть вычислений сравнениями яркостей.

Операция	SIFT	Бинарные признаки
Поиск центра	Экстремумы DoG	FAST: сравнения на кольце
Описание	Гистограммы градиентов	BRIEF: сравнения яркостей
Одна точка	128 чисел float32: 512 байт	256 бит: 32 байта
Сравнение описаний	Разности числовых компонент	Число несовпавших битов

Сопоставление фотографий объекта

Для каждой точки уже получено описание окрестности. Теперь по этим описаниям ищем одинаковые детали в двух фотографиях: координаты и порядок точек могут различаться.



Расстояние между дескрипторами

Для числовых описаний сравниваем величины компонент; для бинарных — считаем несовпавшие разряды. В примере векторы сокращены для ручного расчёта.

$$d_2 = \sqrt{0^2 + 3^2 + 4^2} = 5, \qquad d_H = 2$$

Числовые компоненты

A	1	2	3
B	1	5	7

Разности: 0, 3, 4

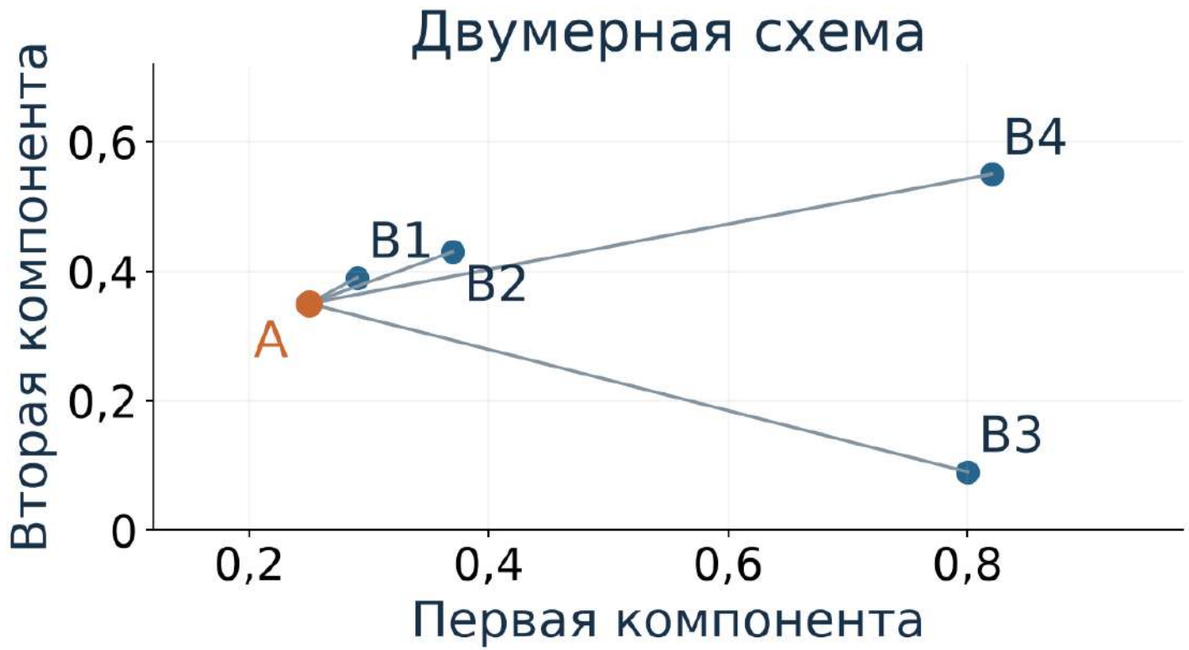
Бинарные сравнения

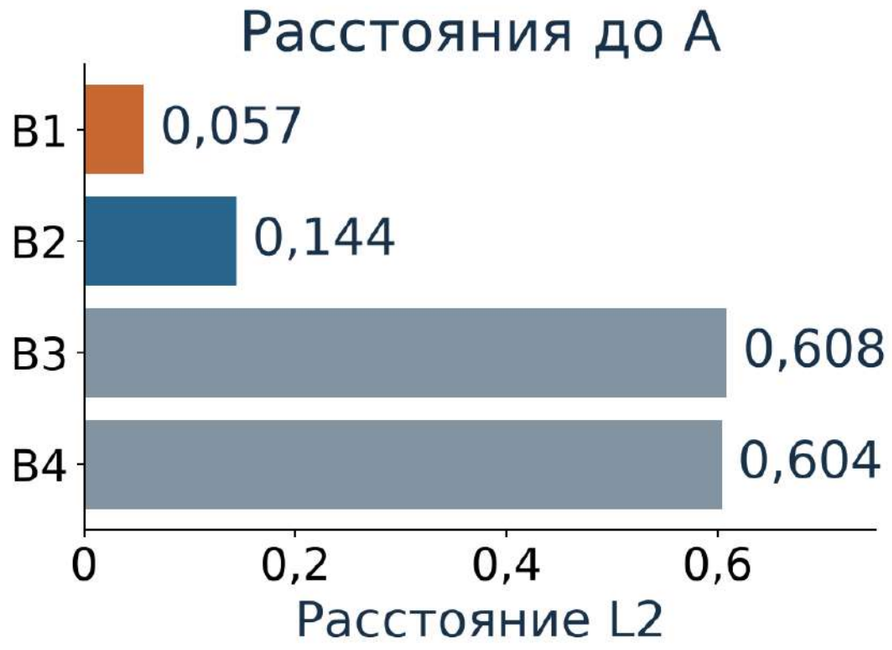
A	0	1	1	0	1	0
B	0	1	0	1	1	0

Различаются два разряда

Ближайшие дескрипторы

Сравниваем описание A со всеми кандидатами B. Две координаты на схеме — компоненты дескриптора, а не положение точки на фотографии.



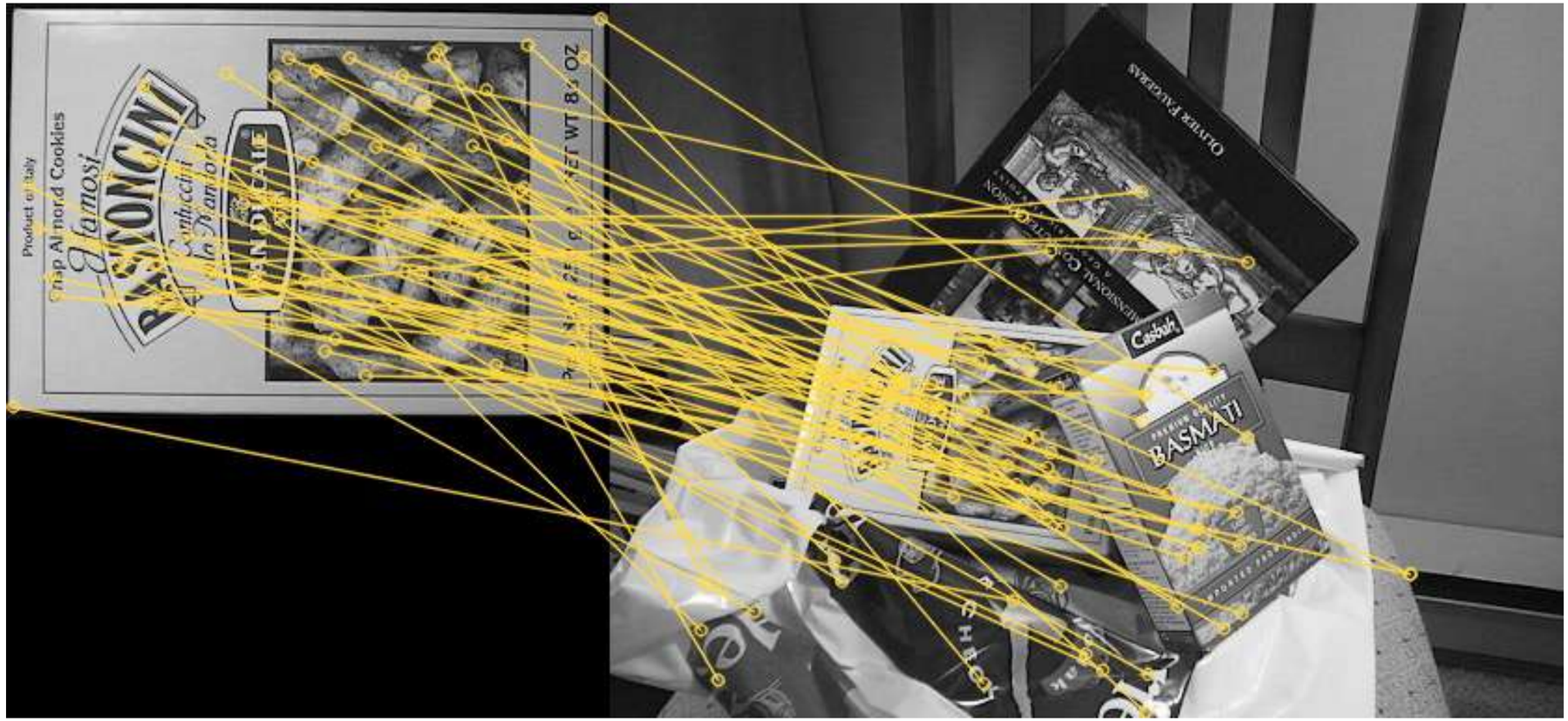


$$d_j = \|\mathbf{a} - \mathbf{b}_j\|_2$$

Неотфильтрованные соответствия

604 дескриптора дают 604 ближайших кандидата; на рисунке показана часть линий.

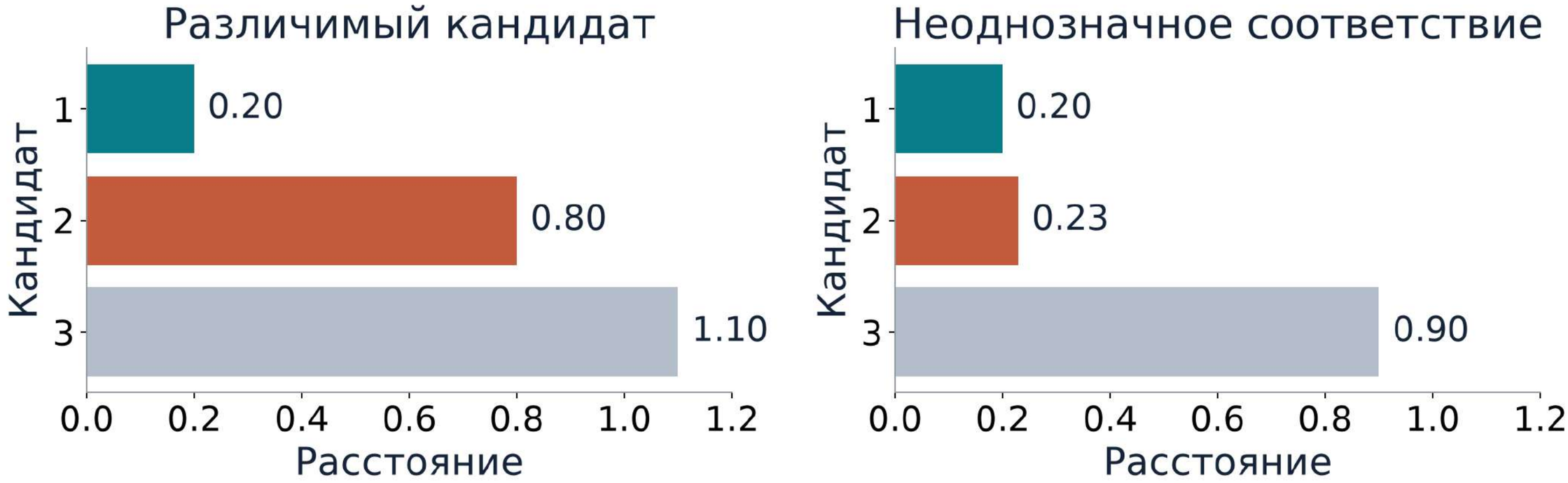
Ближайший кандидат может быть неверным: повторяющиеся детали и фон создают похожие описания.



Критерий Лоу

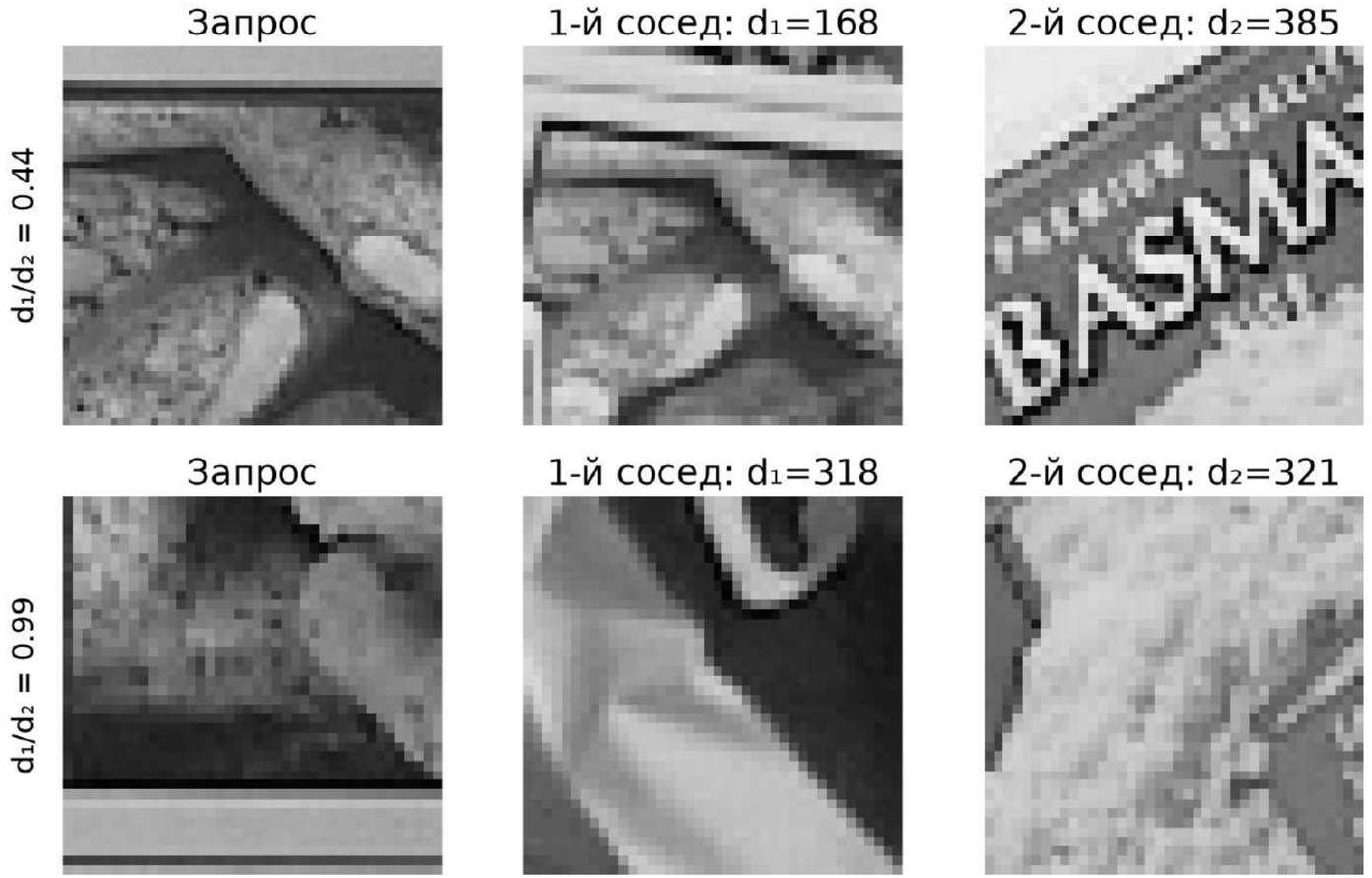
Первый кандидат принимается, если он заметно ближе второго. Порог задаёт требуемую различимость.

$$\frac{d_1}{d_2} < \tau, \qquad d_1 \le d_2, \quad \tau = 0.75$$



Ближайшие кандидаты: пример

Для каждой выбранной точки показаны исходная окрестность и два ближайших кандидата.
Сравнение выполняется по дескрипторам.



Соответствия после критерия Лоу

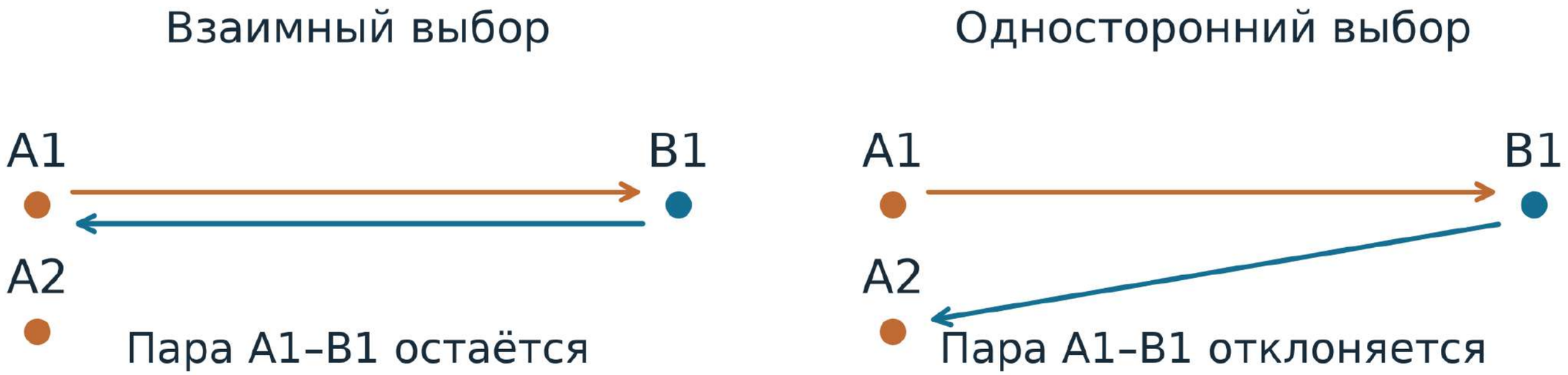
После фильтрации осталось 80 соответствий из 604.

Критерий устраняет многие неоднозначные пары; оставшиеся соответствия требуют геометрической проверки.



Взаимная проверка соответствий

Ищем ближайшее описание в обе стороны. Пару оставляем, если обе точки выбрали друг друга. Критерий Лоу дополнительно проверяет отличие от второго кандидата.



KNN и критерий Лоу: Python

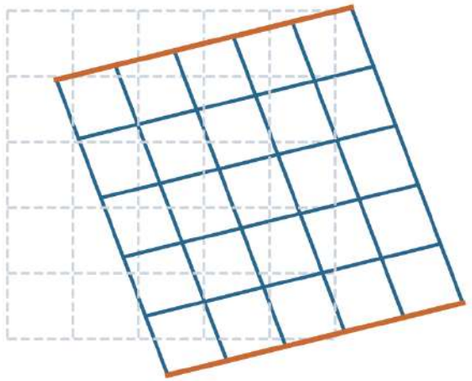
```
# SIFT: расстояние L2 между числовыми описаниями
if des_a is None or des_b is None:
    raise ValueError("На одном из кадров нет дескрипторов")
matcher = cv2.BFMatcher(cv2.NORM_L2)
pairs = matcher.knnMatch(des_a, des_b, k=2)

good = []
for pair in pairs:
    if len(pair) < 2:              # второго кандидата может не быть
        continue
    first, second = pair
    if first.distance < 0.75 * second.distance:
        good.append(first)      # сохраняем индексы двух точек
```

Геометрические модели

Аффинная модель описывает линейные изменения и перенос. Гомография дополнительно описывает перспективное отображение плоскости.

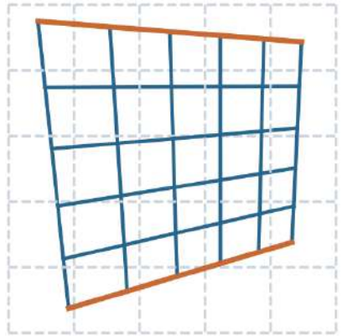
Аффинное преобразование



Поворот, перенос, масштабирование и скос (shear / skew).

$$A = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Перспективное преобразование



Дополнительно — перспектива: параллельные прямые могут сходиться.

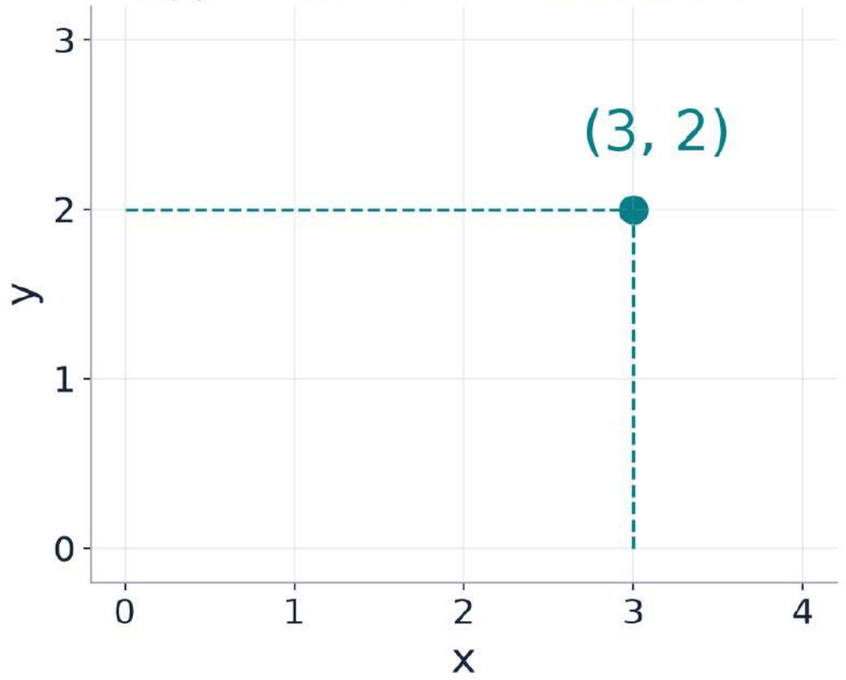
$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Однородные координаты

Однородные координаты — тройка чисел, которая задаёт точку после деления первых двух чисел на третье.

$$(x, y) \longleftrightarrow [\lambda x, \lambda y, \lambda]^{\mathsf{T}}, \qquad \lambda \neq 0$$

Одна точка на плоскости



Несколько записей той же точки

(3, 2, 1) → 3 / 1 = 3; 2 / 1 = 2

(6, 4, 2) → 6 / 2 = 3; 4 / 2 = 2

(9, 6, 3) → 9 / 3 = 3; 6 / 3 = 2

Гомография

- Гомография умножает однородные координаты на матрицу. Полученная тройка переводится обратно в пиксельные координаты делением на её третью компоненту.

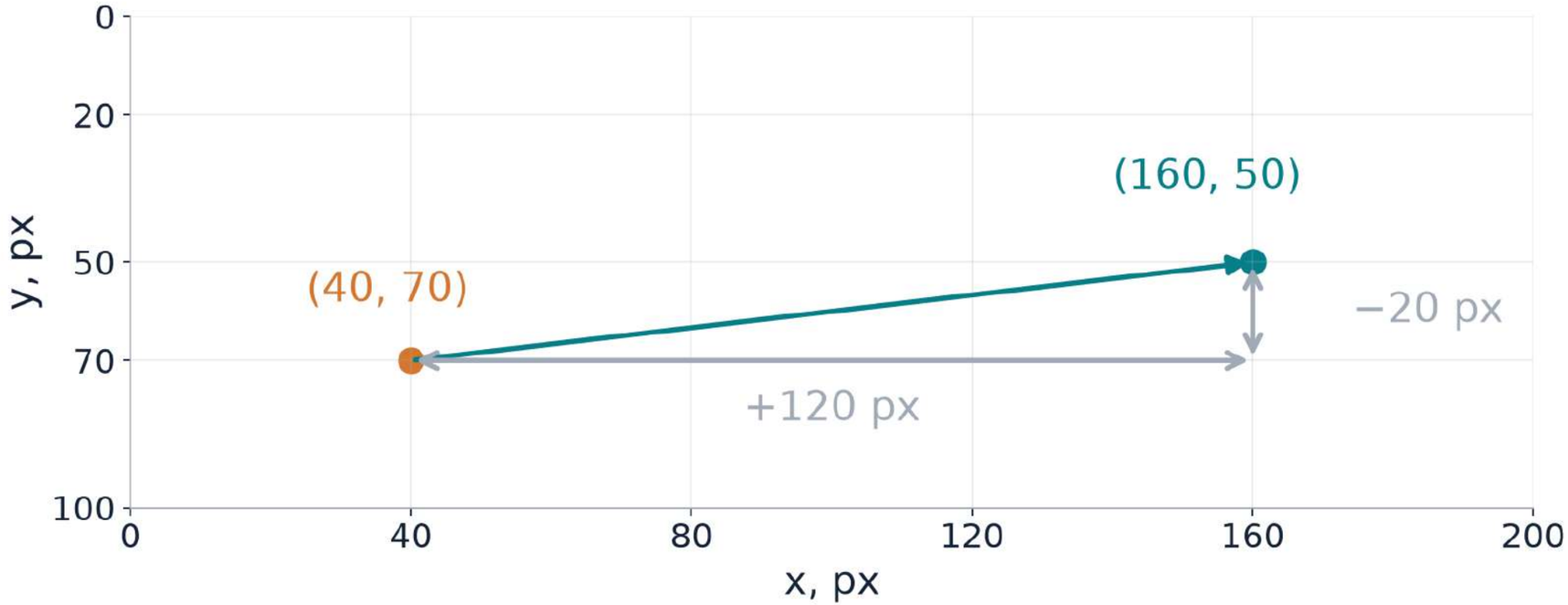
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = \frac{u}{w}, \qquad y' = \frac{v}{w}$$

Перенос точки

Горизонтальная координата увеличивается вправо, вертикальная — вниз.

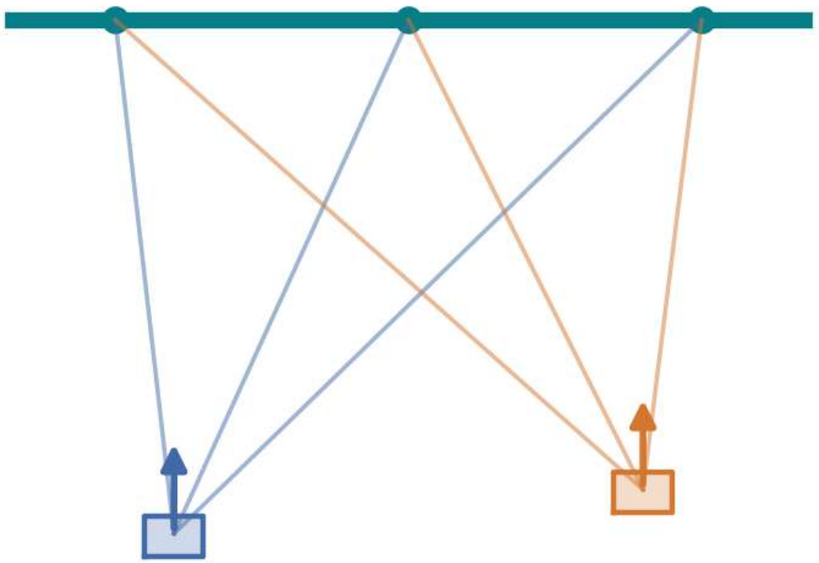
$$\begin{bmatrix} 1 & 0 & 120 \\ 0 & 1 & -20 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 40 \\ 70 \\ 1 \end{bmatrix} = \begin{bmatrix} 160 \\ 50 \\ 1 \end{bmatrix}$$



Условия для одной гомографии

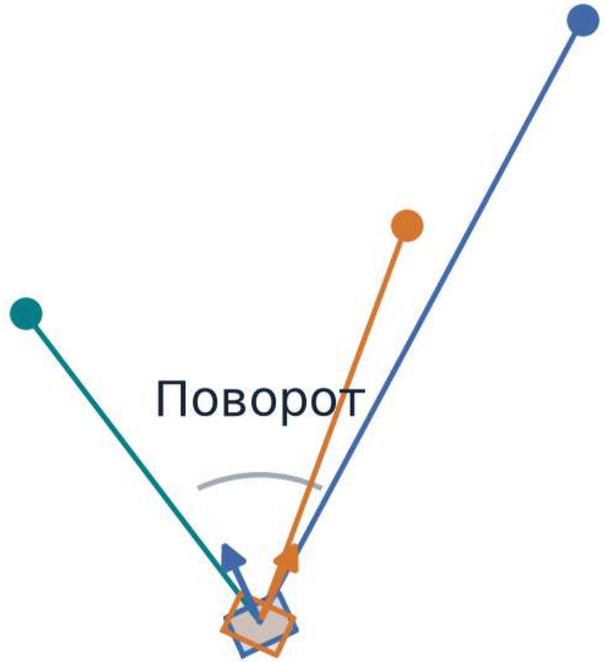
Плоская сцена

Общая плоскость



Положение камеры может меняться

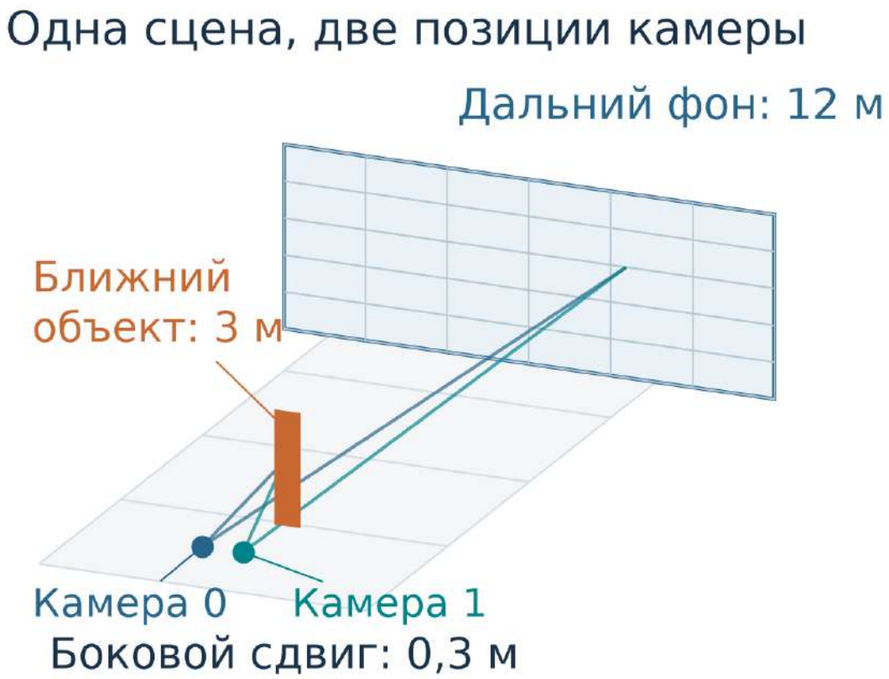
Один центр камеры

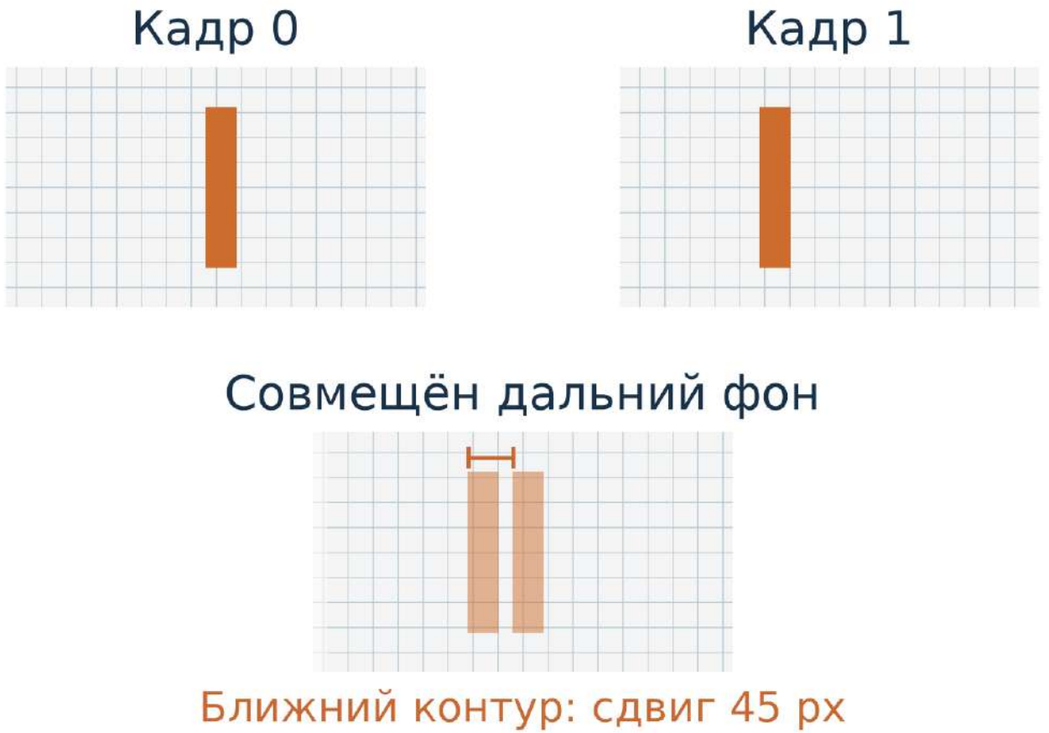


Объекты могут иметь разную глубину

Параллакс

При боковом сдвиге камеры ближний объект смещается сильнее фона. После выравнивания фона его двойной контур остаётся.



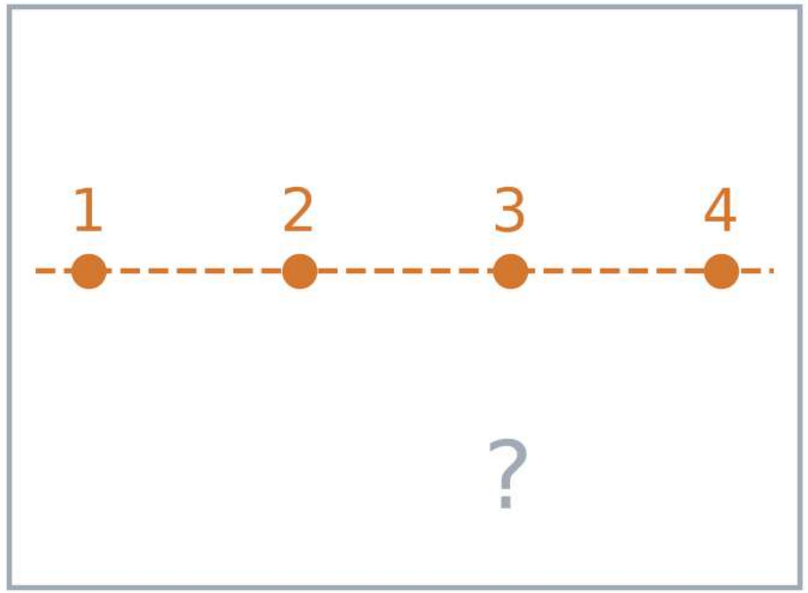


$$|\Delta x| = fB/Z: \quad 600 \cdot 0{,}3/3 = 60 \text{ px}, \quad 600 \cdot 0{,}3/12 = 15 \text{ px}$$

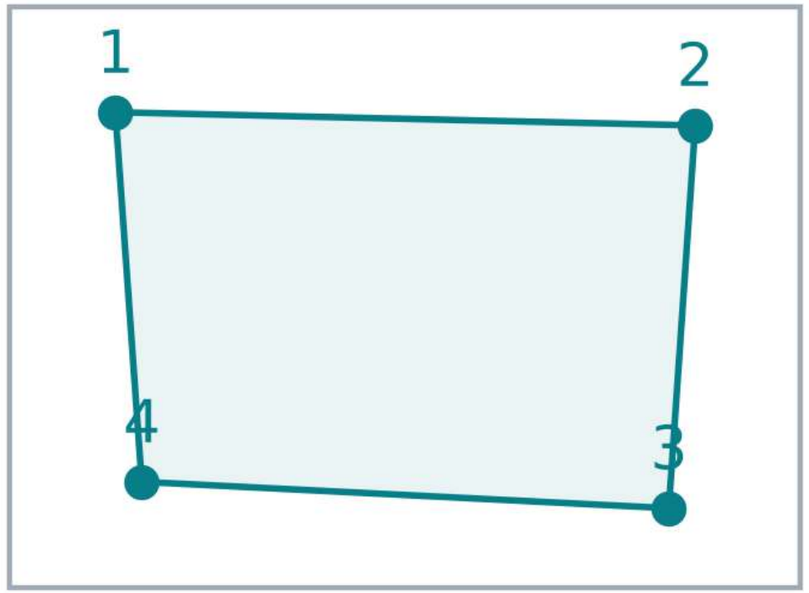
Расположение опорных точек

Минимум — четыре пары: никакие три точки этой четвёрки не должны лежать на одной прямой. Для устойчивой оценки точки распределяют по общей области.

Четыре точки на одной прямой



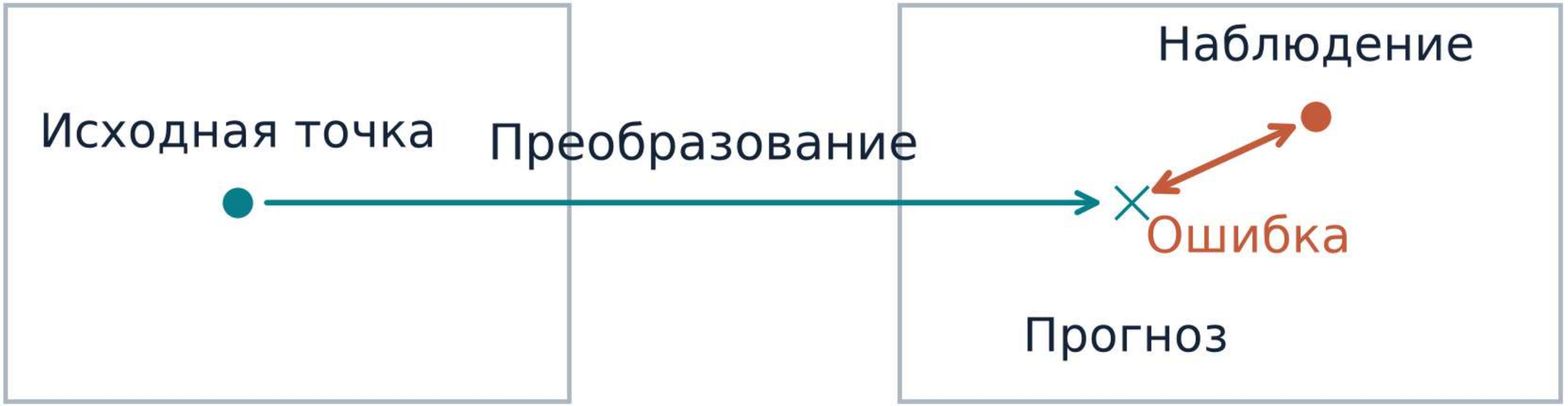
Четыре точки охватывают площадь



Ошибка переноса точки

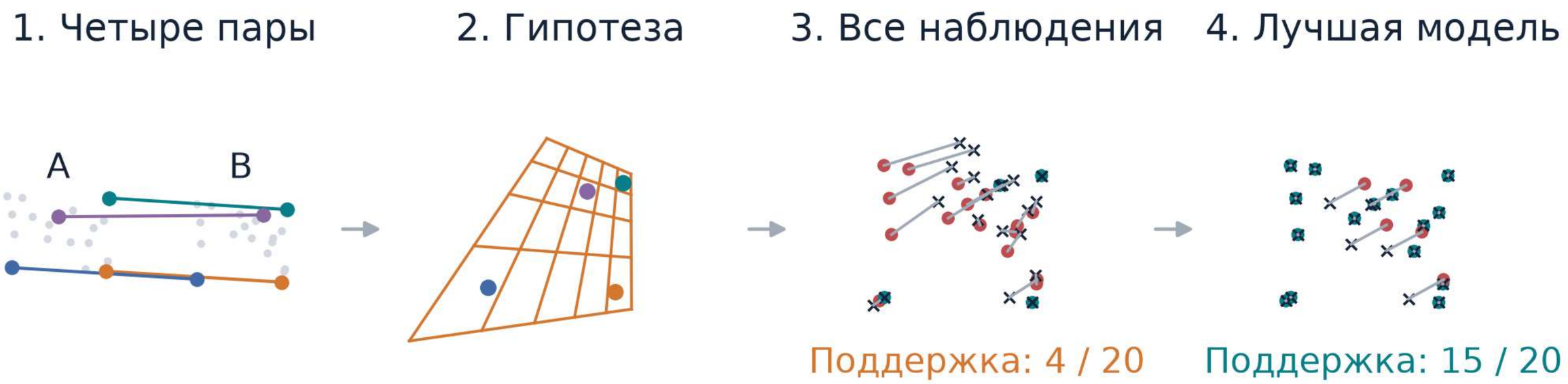
Гомография предсказывает положение исходной точки в целевом изображении. Ошибка — расстояние от прогноза до наблюдаемой точки.

$$e_k = \|\mathbf{x}'_k - \pi(H\widetilde{\mathbf{x}}_k)\|_2$$



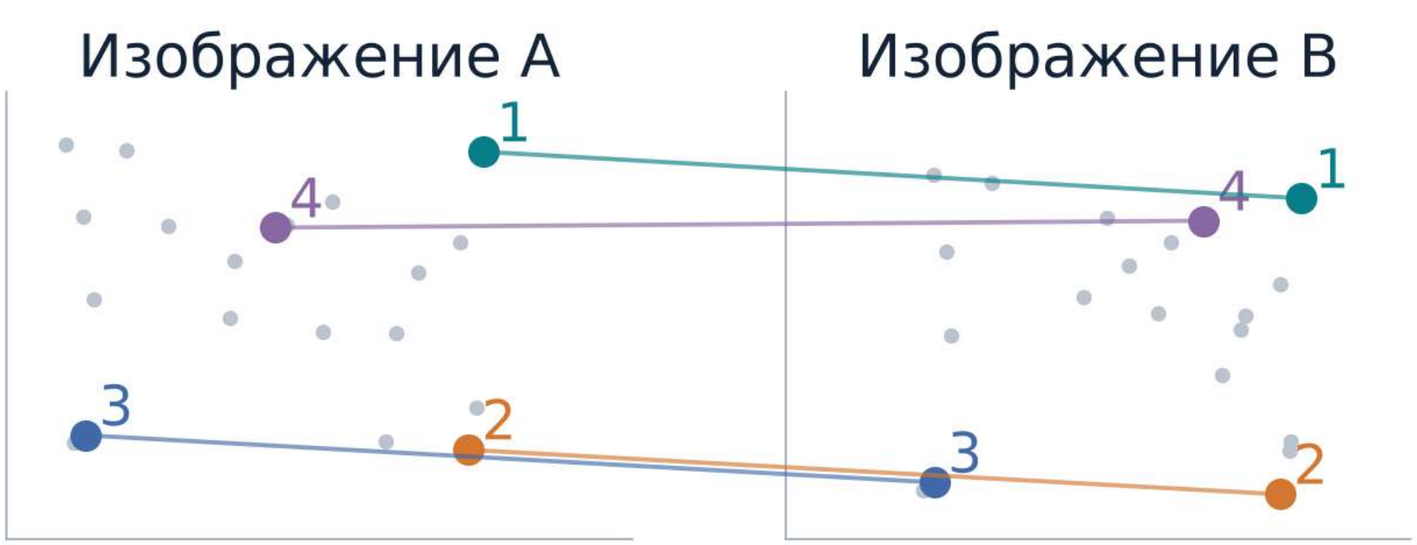
Алгоритм RANSAC

Разные четвёрки дают разные гипотезы. Лучшую выбирают по числу согласованных пар и уточняют по всем её inliers.



Ошибочная гипотеза RANSAC

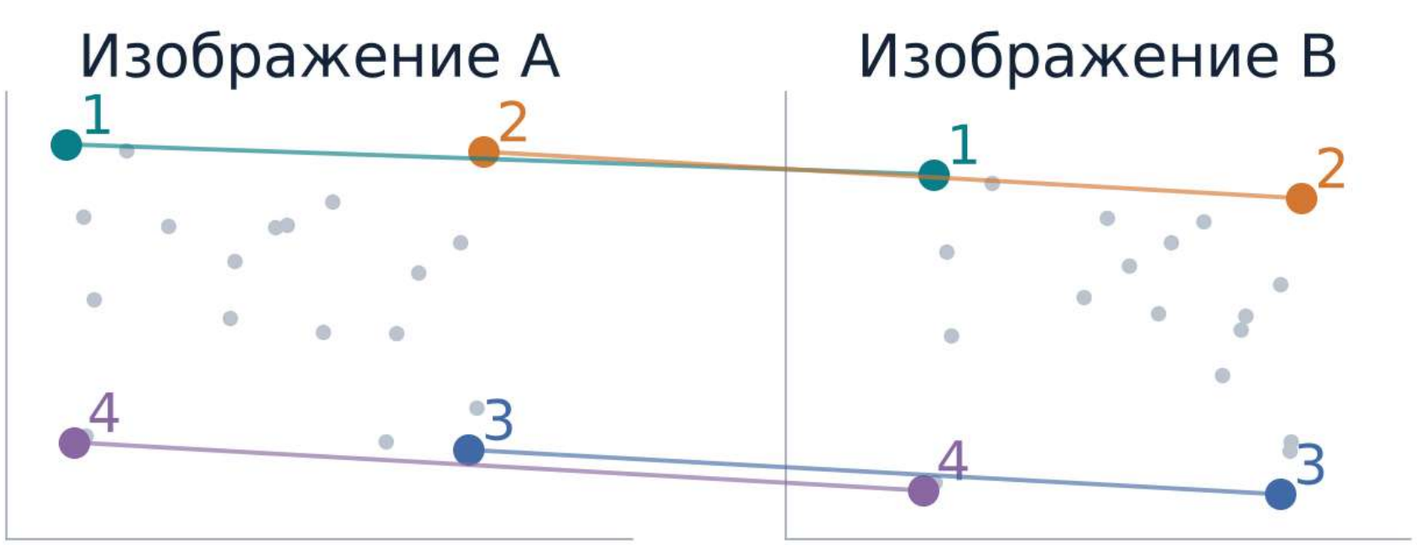
Четыре цветные линии задают выбранные пары. Справа показаны наблюдения, прогнозы модели и соединяющие их остатки.





Согласованная гипотеза RANSAC

Четвёрка верных пар получает поддержку 15 из 20 наблюдений. Модель проверяется по всем парам, а не только по выбранным четырём.





Оценка гомографии: Python

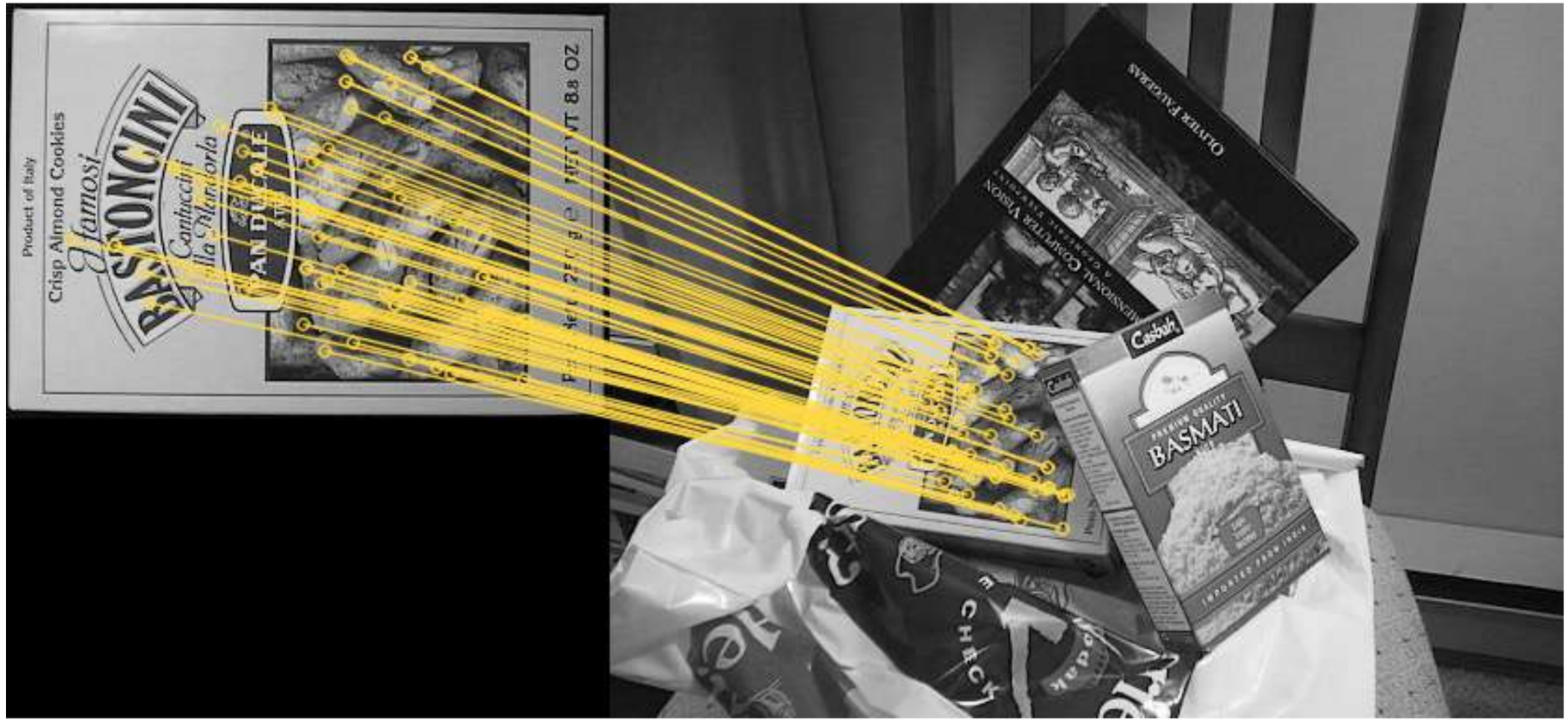
```
# Матчи искали A -> B; геометрию оцениваем в направлении B -> A
if len(good) < 4:
    raise ValueError("Для гомографии нужны хотя бы четыре пары")
src = np.float32([kp_b[m.trainIdx].pt for m in good])
dst = np.float32([kp_a[m.queryIdx].pt for m in good])

# Порог 3 пикселя относится к системе координат кадра A
H, inliers = cv2.findHomography(
    src, dst, cv2.RANSAC, 3.0,
    maxIters=10000, confidence=0.999
)
if H is None:
    raise ValueError("Не удалось оценить преобразование")
# Далее проверяем число, распределение точек и полученный контур
```

Соответствия после RANSAC

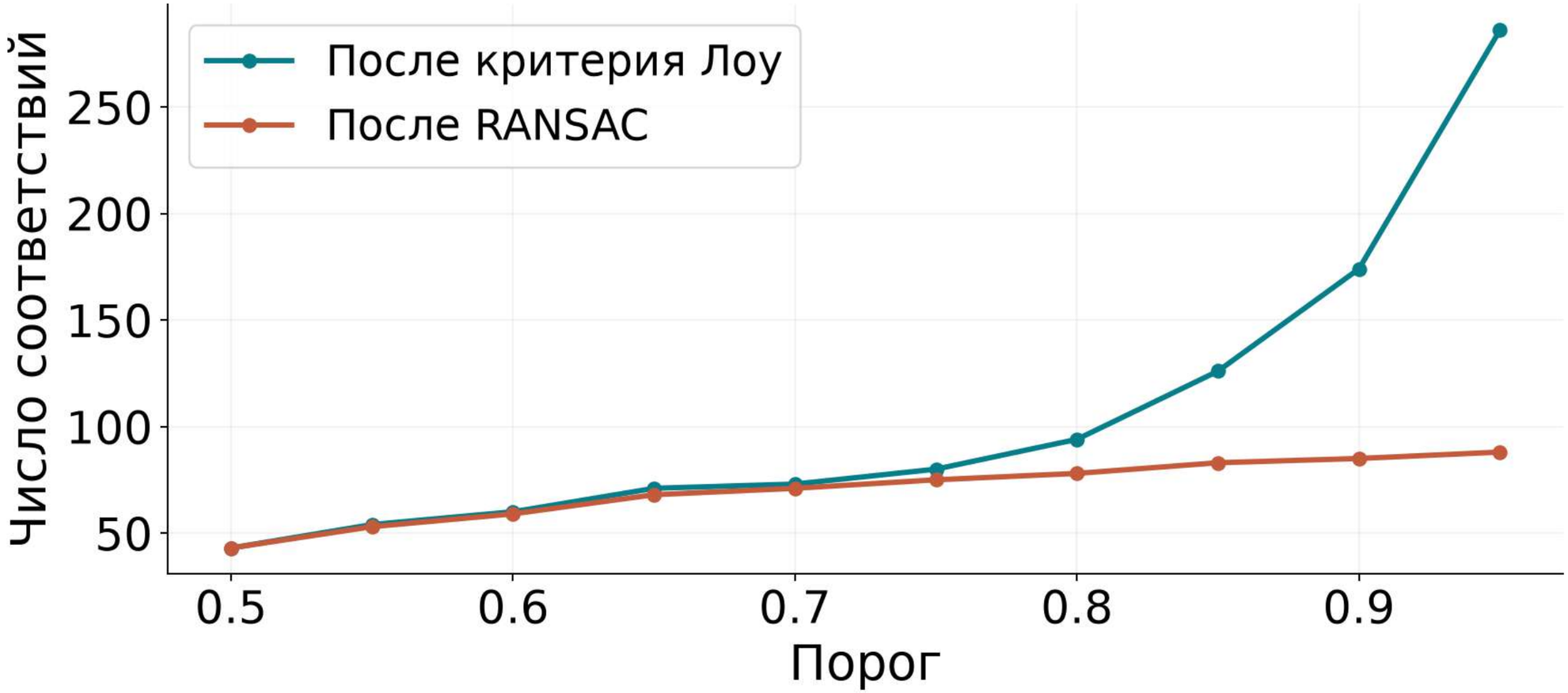
Из 80 пар после Лоу RANSAC сохранил 75. Медианная ошибка переноса составила 0.50 px.

Остаток на принятых парах характеризует подгонку модели; для оценки надёжности нужны дополнительные проверки.



Порог критерия Лоу

Меньший порог уменьшает число принятых пар. Больший порог сохраняет больше кандидатов и допускает больше неоднозначности.



Совмещение контура объекта

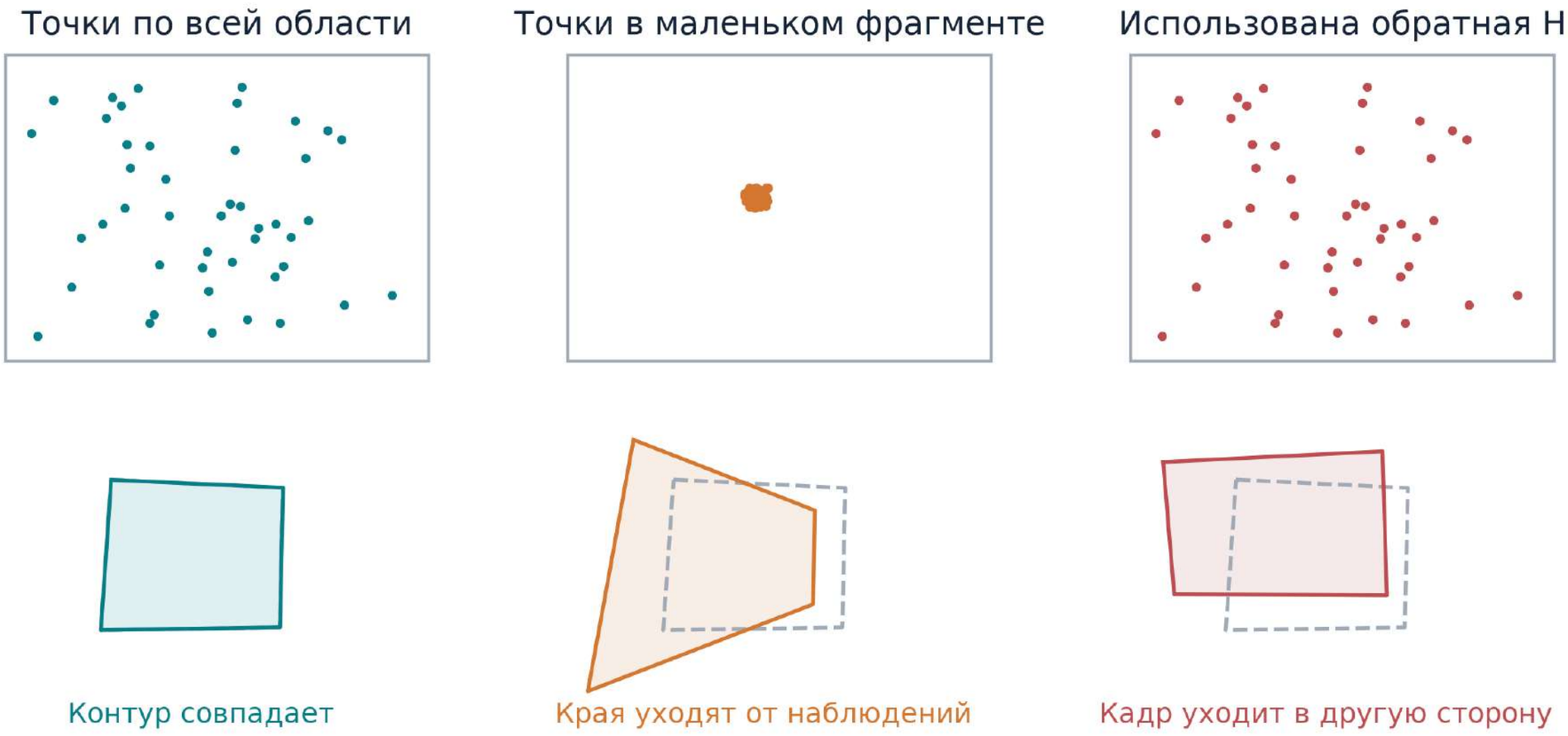
Оценённое преобразование применяется к новым координатам — четырём углам изображения объекта.

Корректная рамка подтверждает, что модель описывает область объекта, а не только отдельные использованные точки.



Проверка преобразования

Верхний ряд: расположение использованных точек. Нижний ряд: перенесённый контур кадра; пунктир обозначает правильное положение.



Склейка двух соседних кадров

Преобразование можно использовать для совмещения двух фотографий одной сцены. Общие детали в перекрытии связывают кадры, а крайние области расширяют поле зрения.

Кадр 0



Кадр 1



Общее изображение



Соответствия соседних кадров

Для кадров 0 и 1 получено 551 соответствие после критерия Лоу и 513 согласованных пар после RANSAC. По ним оценивается преобразование координат.



Панорама из пяти кадров

Тот же способ совмещения применяется к ряду из пяти фотографий. Каждая соседняя пара имеет общие детали; все кадры нужно разместить на одном холсте.

Кадр 0



Кадр 1



Кадр 2



Кадр 3



Кадр 4



Панорама



Съёмка горизонтального ряда

Соседние фотографии должны содержать общие неподвижные детали. Камеру поворачивают вокруг одного центра; фокус и настройки цвета фиксируют.

Кадр 0



Общая неподвижная деталь

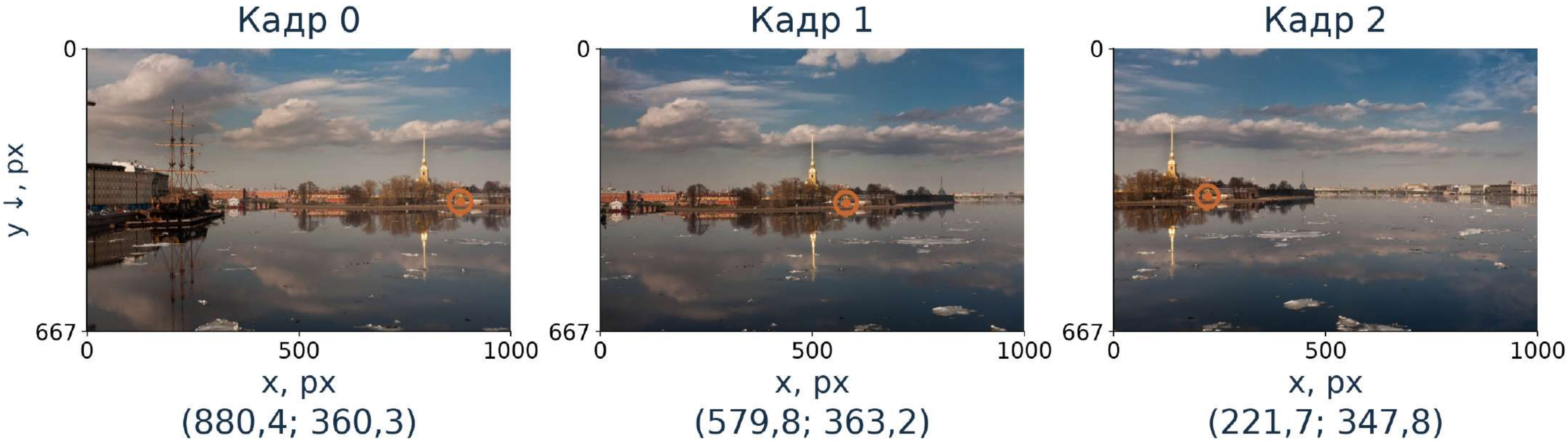
Кадр 1



Общая неподвижная деталь

Координаты одной детали в трёх кадрах

Начало координат каждого кадра — его левый верхний угол. Поэтому одно и то же место сцены имеет разные пиксельные координаты.



$$\widetilde{\mathbf{x}}_i \sim H_{i \leftarrow j} \widetilde{\mathbf{x}}_j$$

Композиция преобразований

Сначала переносим точку из кадра 2 в кадр 1, затем — из кадра 1 в кадр 0. Правая матрица в произведении действует первой.



$$\widetilde{\mathbf{x}}_0 \sim H_{0\leftarrow 1}\left(H_{1\leftarrow 2}\widetilde{\mathbf{x}}_2\right)$$

Порядок умножения матриц

На той же детали правильная последовательность совмещает наблюдения. Перестановка матриц меняет прогноз положения точки.

Область в кадре 0

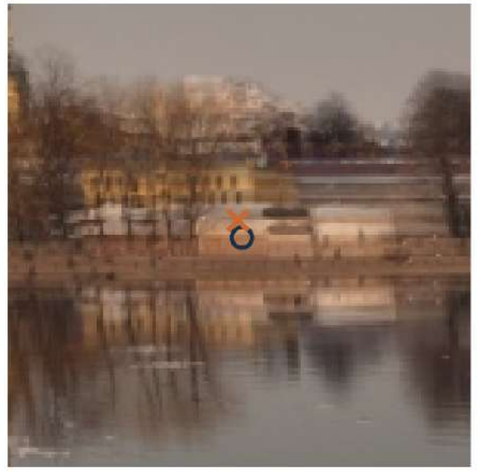


Порядок 2 → 1 → 0



Ошибка: 0,23 px

Множители переставлены



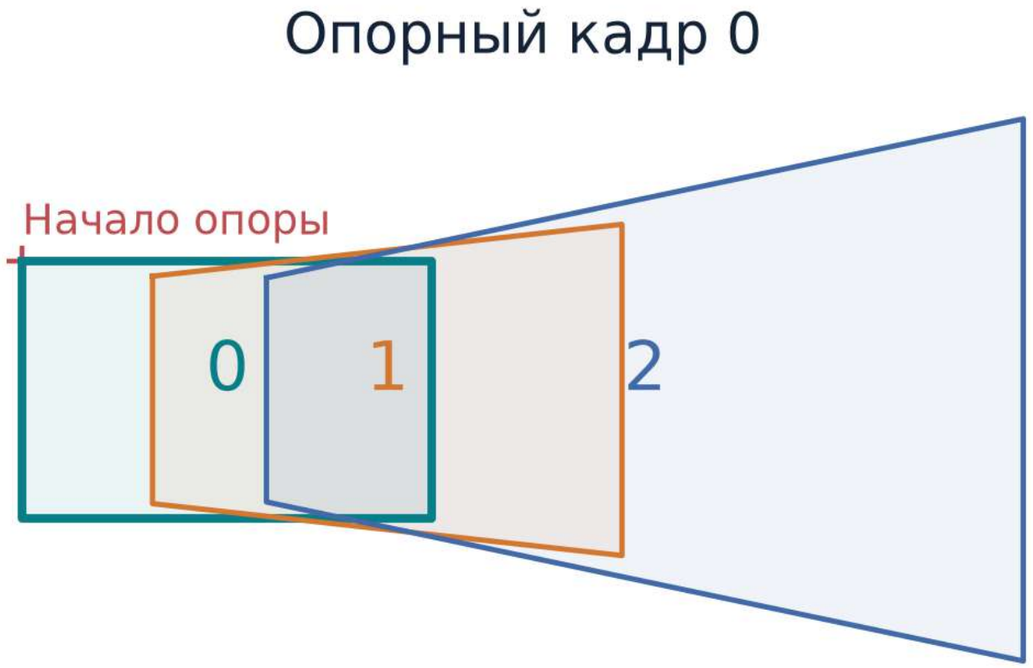
Ошибка: 5,70 px

$$H_{0\leftarrow 1} H_{1\leftarrow 2} \neq H_{1\leftarrow 2} H_{0\leftarrow 1}$$

Смена опорного кадра

Стрелка в индексе указывает направление перехода между кадрами. Для новой опоры r выполняем переходы i → 0 → r.

$$H_{r\leftarrow i} = H_{r\leftarrow 0} H_{0\leftarrow i} = \left(H_{0\leftarrow r}\right)^{-1} H_{0\leftarrow i}, \qquad H_{r\leftarrow r} = I$$





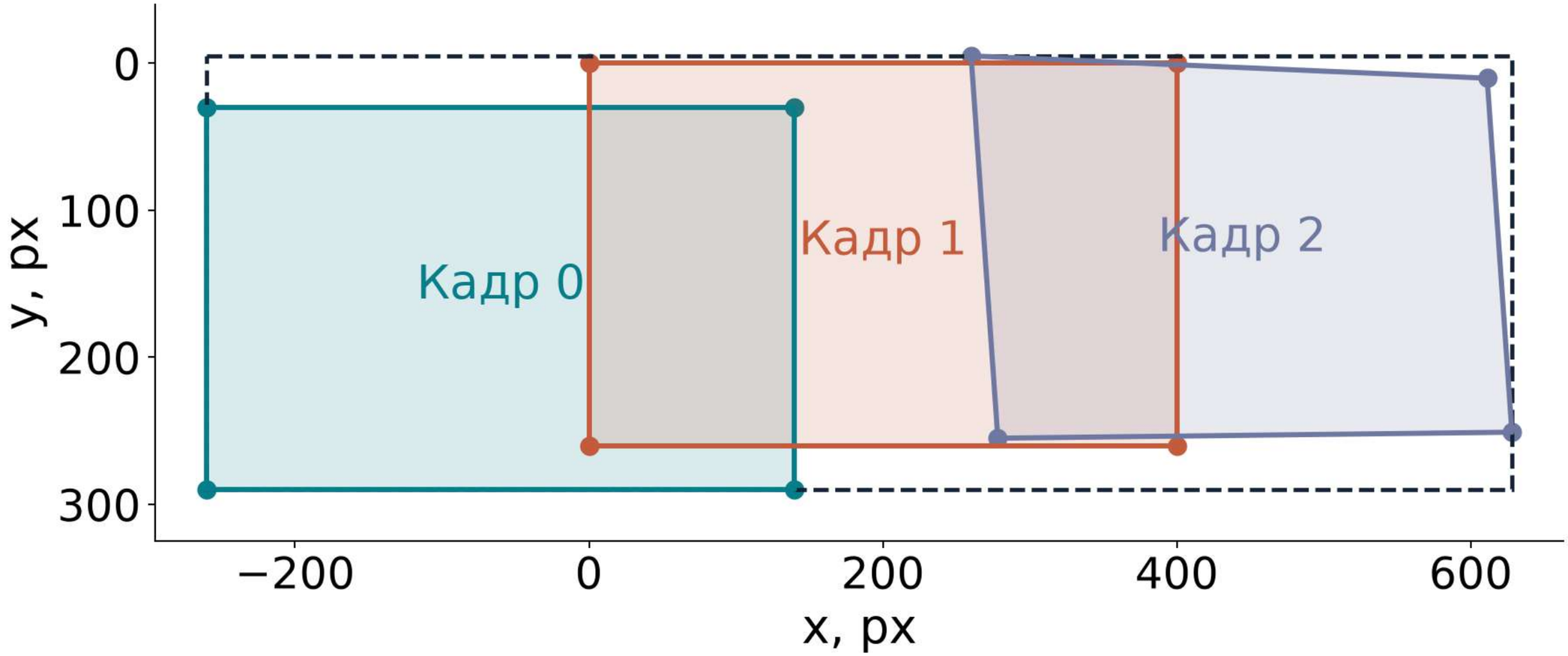
Композиция преобразований: Python

```python
# Вход: каждая H переводит следующий кадр в текущий
G = [np.eye(3)]                    # кадр 0 пока служит опорой
for H_next_to_current in pairwise_H:
    transform = G[-1] @ H_next_to_current
    transform /= np.linalg.norm(transform)
    G.append(transform)

# Меняем общую систему координат на центральный кадр
reference = len(G) // 2
change_frame = np.linalg.inv(G[reference])
G = [change_frame @ transform for transform in G]
# Каждый исходный кадр затем переносится на холст один раз
```

Границы кадров на холсте

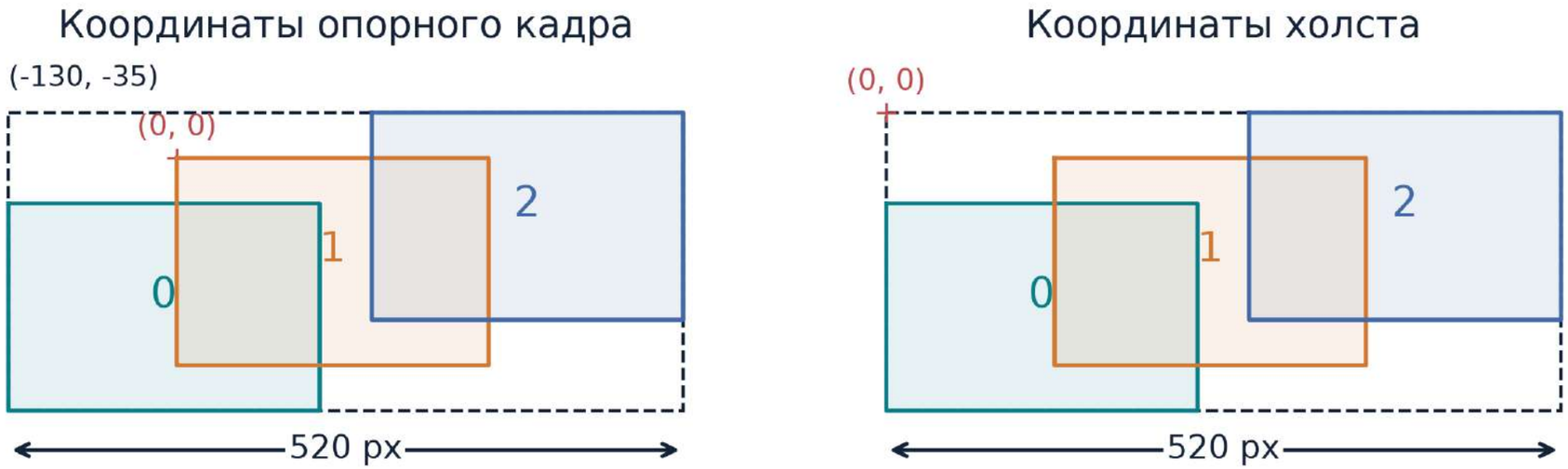
Четыре угла каждого кадра преобразуются собственной накопленной матрицей. Общие минимумы и максимумы задают границы будущего холста.



Холст и начало координат

Границы всех кадров задают общий прямоугольник. Перенос его левого верхнего угла в (0, 0) устраняет отрицательные координаты.

$$T = \begin{bmatrix} 1 & 0 & -x_{\min} \\ 0 & 1 & -y_{\min} \\ 0 & 0 & 1 \end{bmatrix}, \qquad W_i = T\widehat{G}_i$$

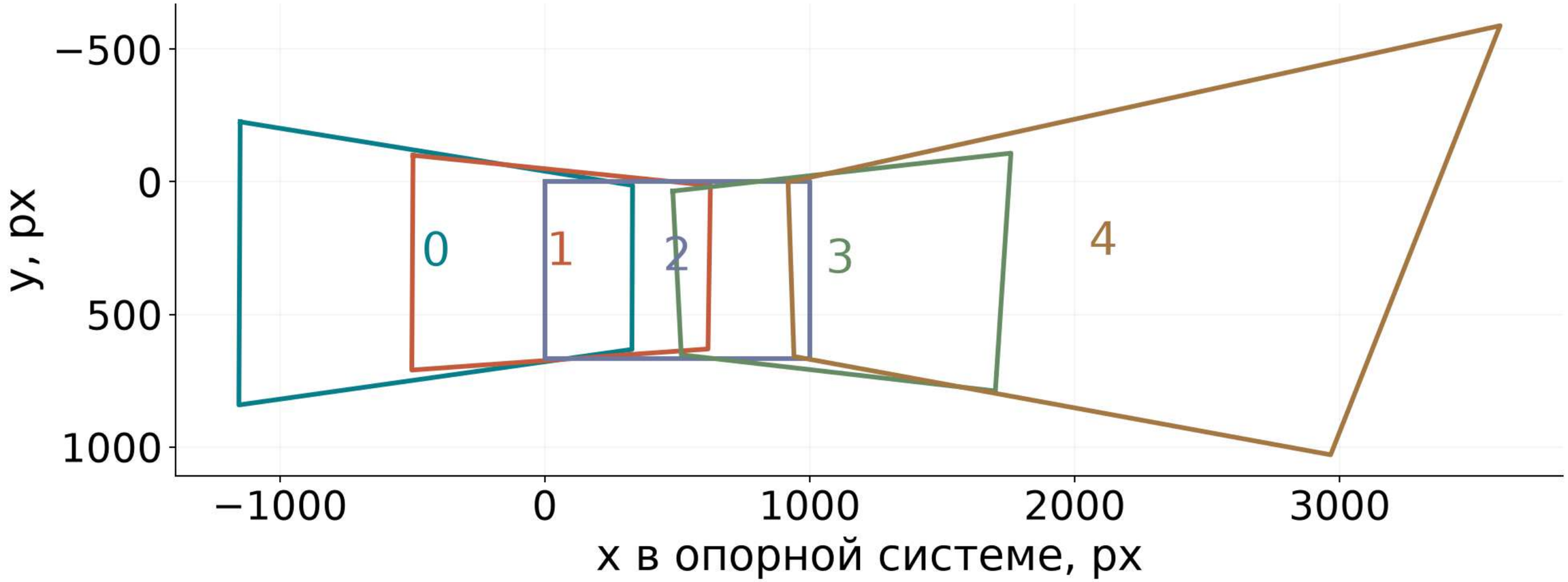


Общий сдвиг: +130 по x, +35 по y

Холст для пяти кадров

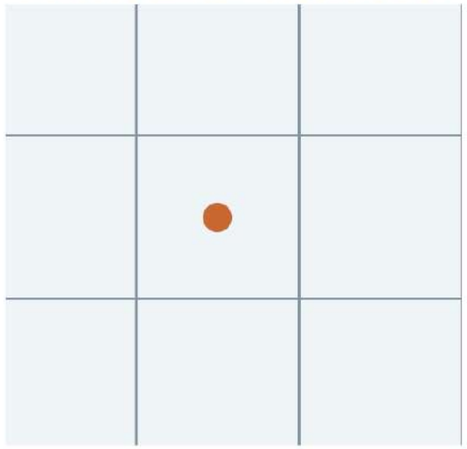
Опорный кадр имеет индекс 2. Размер общего холста — 4761 × 1619 px.

Широкий угловой охват приводит к заметному растяжению крайних кадров в плоской проекции.

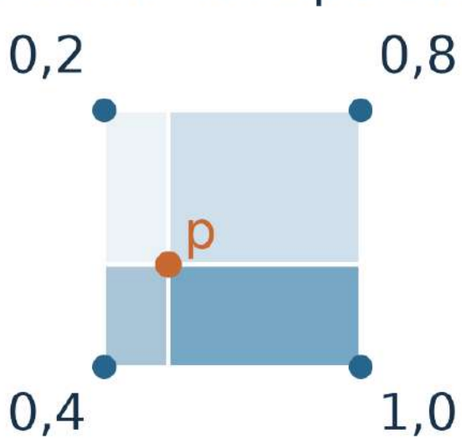


Интерполяция при переносе изображения

Обратное отображение даёт дробную координату. Цвет вычисляем по четырём соседям: их веса зависят от расстояния до точки.



Обратное отображение
даёт дробную координату



x = 0,25; y = 0,60
Ось y направлена вниз

Вес × яркость

0,30 × 0,2
0,10 × 0,8
0,45 × 0,4
0,15 × 1,0

Сумма: 0,47

$$w_{00} = (1 - 0{,}25)(1 - 0{,}60) = 0{,}30, \qquad I(p) = \sum_{a,b \in \{0,1\}} w_{ab} I_{ab} = 0{,}47$$

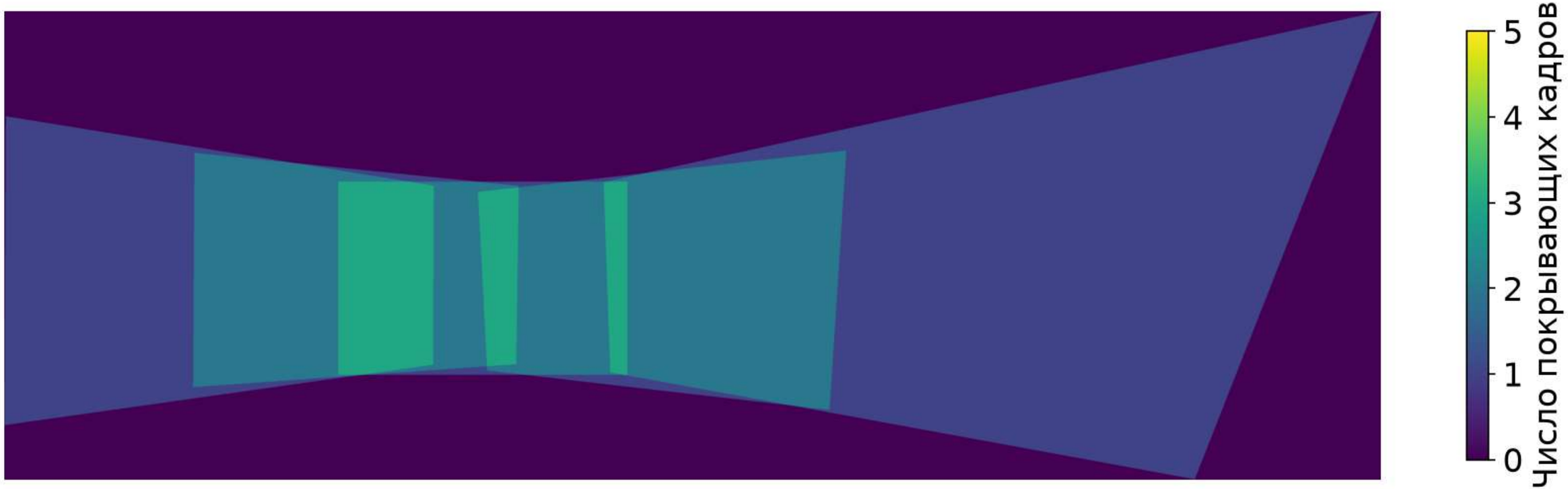
Перенос изображения и маски: Python

```
# Цвет переносим с билинейной интерполяцией
warped = cv2.warpPerspective(
    image, W, (width, height), flags=cv2.INTER_LINEAR
)

# Маска описывает наличие данных, независимо от их цвета
source_mask = np.ones(image.shape[:2], np.uint8)
valid = cv2.warpPerspective(
    source_mask, W, (width, height), flags=cv2.INTER_NEAREST
) > 0
# Чёрный объект тоже является валидной частью изображения
```

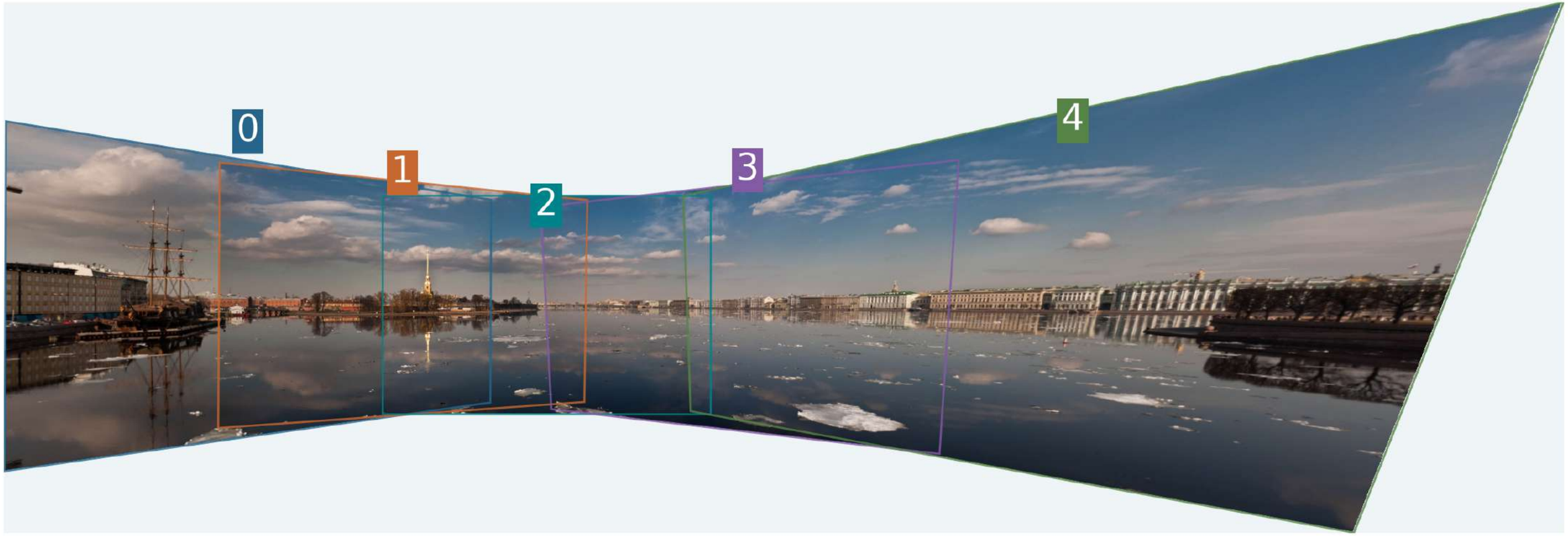
Карта покрытия

Значение маски покрытия равно числу наблюдающих кадров. Ноль обозначает пустоту; два и более — область выбора или смешивания цветов.



Прямое наложение кадров

На общем холсте видны границы кадров и различия яркости и оттенка. Кадры перенесены и последовательно наложены в исходных цветах.



Плавное смешивание в перекрытии

В общей области постепенно уменьшаем вклад одного кадра и увеличиваем вклад другого. Переход смягчается, но общая разница яркости и цвета остаётся.

$$I(\mathbf{x}) = \frac{w_1(\mathbf{x})I_1(\mathbf{x}) + w_2(\mathbf{x})I_2(\mathbf{x})}{w_1(\mathbf{x}) + w_2(\mathbf{x})}$$

Прямое наложение: заметны границы



Плавные веса: переход смягчён

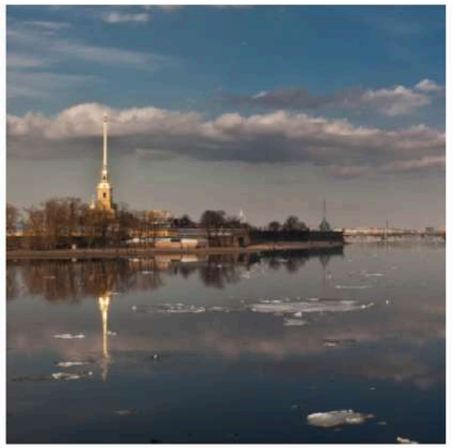


Компенсация по общим деталям

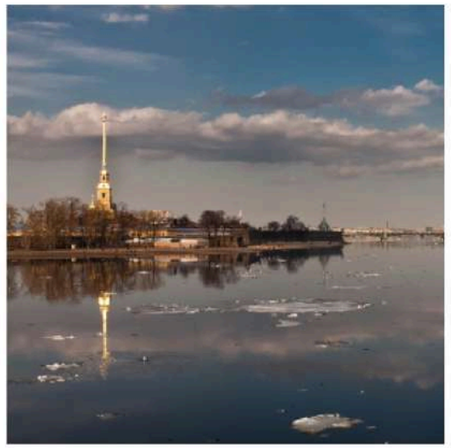
В перекрытии видны одни и те же поверхности. По их яркостям оцениваем поправки к каналам каждого кадра; цвета опорного кадра сохраняем.

$$g_{i,c} I_{i,c}(\mathbf{x}) \approx g_{j,c} I_{j,c}(\mathbf{x}), \qquad c \in \{R, G, B\}, \qquad g_{2,c} = 1$$

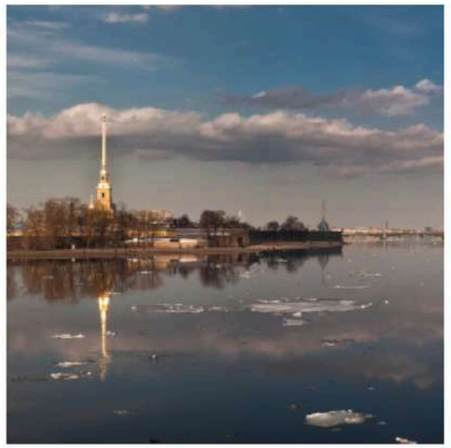
Кадр 1
до коррекции



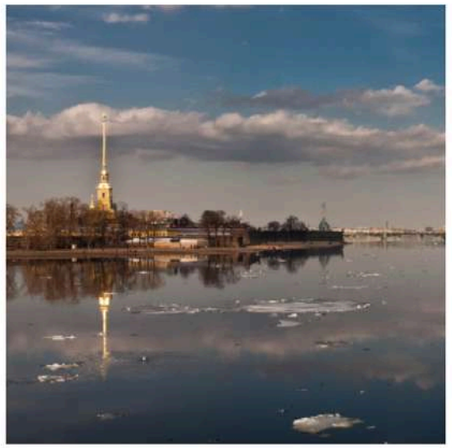
Кадр 2
опорный



Кадр 1
после коррекции



Кадр 2
без изменений



Компенсация и смешивание

Сначала согласуем яркость и цвет по перекрытиям. Затем смешиваем исправленные изображения с плавными весами.

Только плавное смешивание



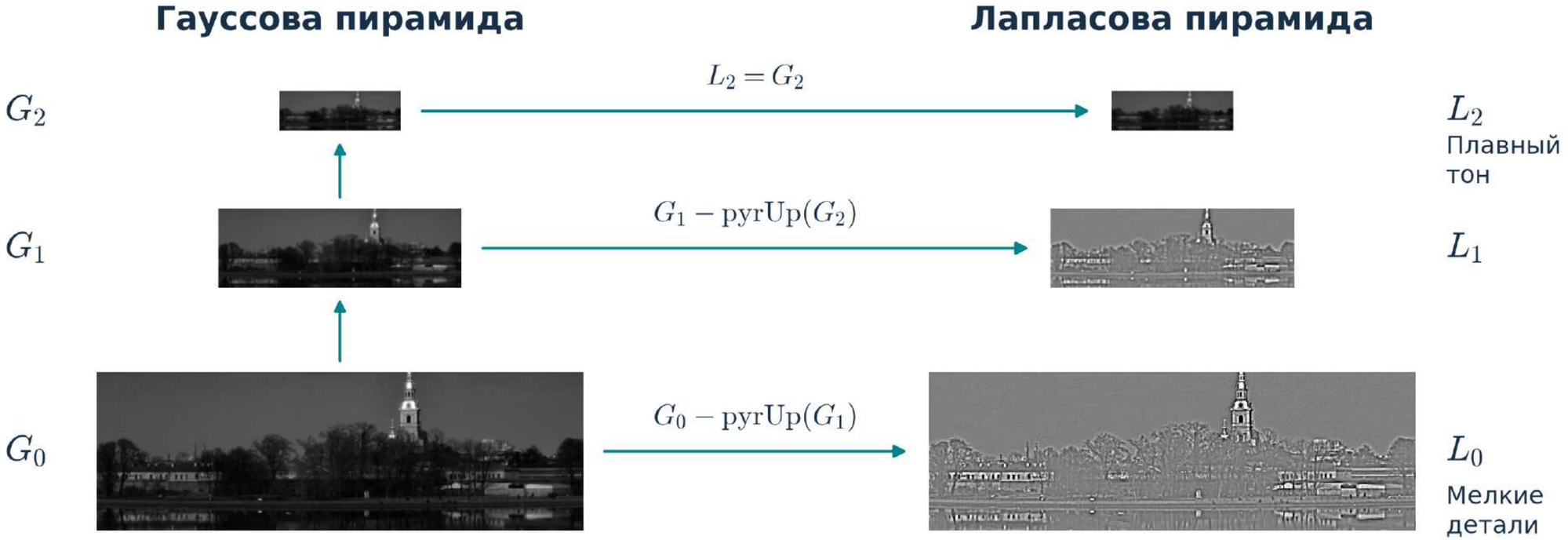
Компенсация цвета и яркости, затем смешивание



Лапласова пирамида

Сглаживаем и уменьшаем текущий уровень вдвое. Результат увеличиваем до размера текущего уровня и вычитаем из него. Разность сохраняет потерянные детали.

$$G_{\ell+1} = \mathrm{pyrDown}(G_\ell), \qquad L_\ell = G_\ell - \mathrm{pyrUp}(G_{\ell+1})$$

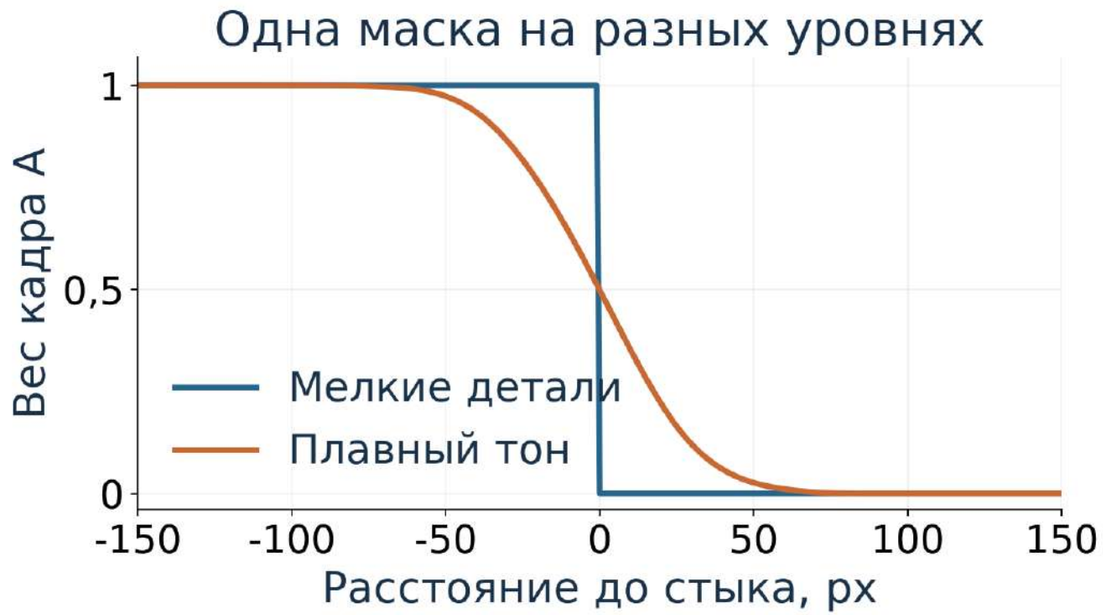


На рисунке разность показана с усилением ×4; серый цвет соответствует нулю.

Многополосное смешивание

Изображения раскладываем в лапласовы пирамиды, маску весов — в гауссову. Каждый уровень смешиваем отдельно, затем собираем изображение обратно.

$$B_\ell = w_\ell L_\ell^A + (1 - w_\ell) L_\ell^B$$



Жёсткая граница



Смешивание по 6 уровням



Собираем от грубого к мелкому.
R — восстановленный уровень.

$$R_\ell = B_\ell + \mathrm{pyrUp}(R_{\ell+1}), \qquad R_N = B_N$$

Результат многополосного смешивания

В обоих вариантах цвета предварительно согласованы. Сравнивать нужно и общий переход яркости, и детали около границы кадров.

Компенсация и плавные веса



Компенсация и многополосное смешивание



Диагностика швов

Разные дефекты требуют разных действий. Плавный переход яркости сам по себе не гарантирует правильного совмещения деталей.

Двойной контур



Проверить геометрию

Яркостный скачок



Согласовать цвета и шов

Плавный неверный цвет

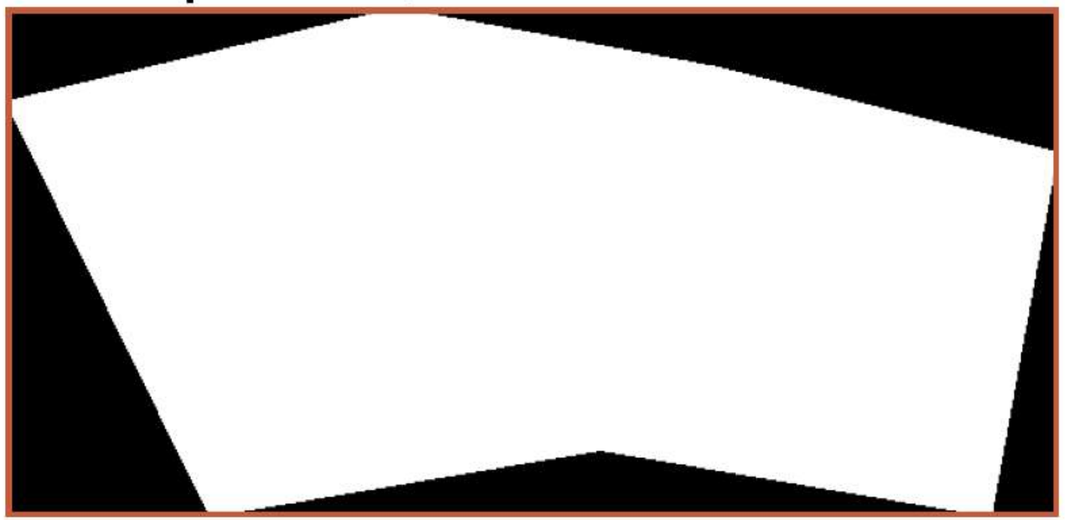


Оценить цветовые поправки

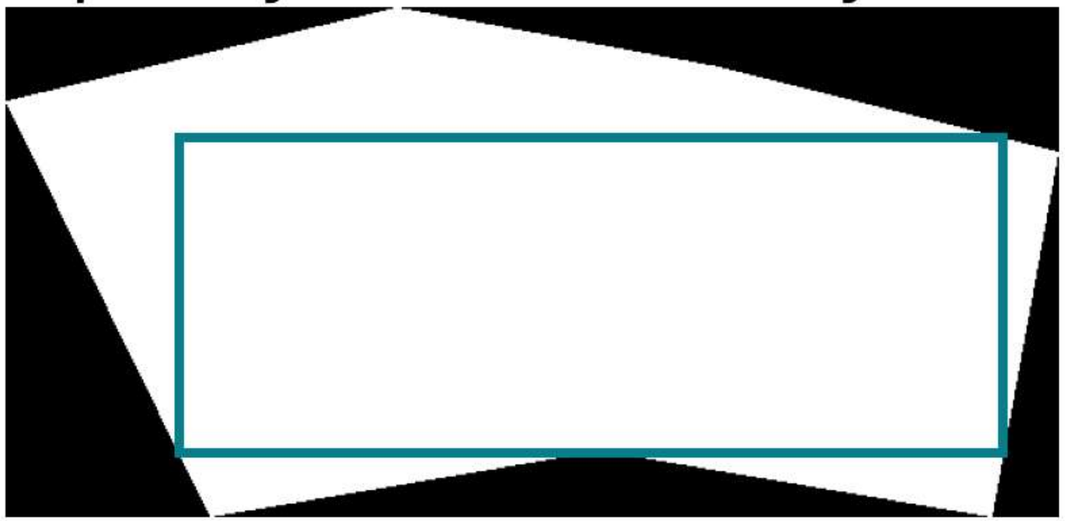
Обрезка панорамы

Ограничивающий прямоугольник маски может содержать пустые углы. Для кадра без пустот нужен прямоугольник целиком внутри валидной области.

Границы всей маски



Прямоугольник без пустот



Итоговая панорама

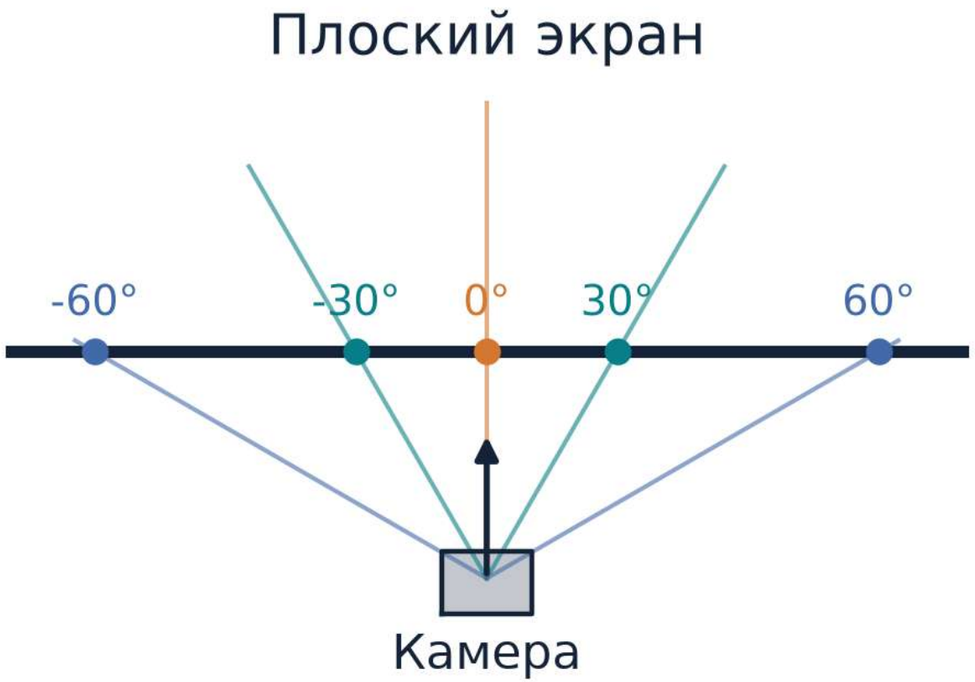
Размер результата — 4255 × 667 px. Каждый пиксель внутри обрезанного кадра покрыт валидными наблюдениями.

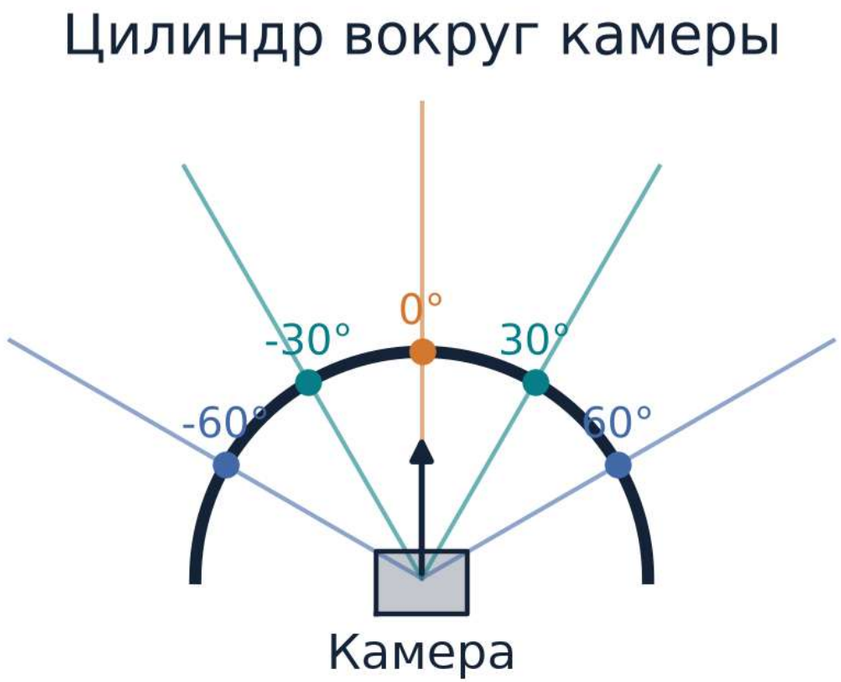
Обрезка уменьшает поле зрения; исходный холст и маску сохраняют для оценки потерь.



Поверхность панорамы

Вид сверху: одни и те же лучи пересекают плоскость или поверхность цилиндра. На плоскости точки у края расходятся сильнее.





Растяжение при широком охвате

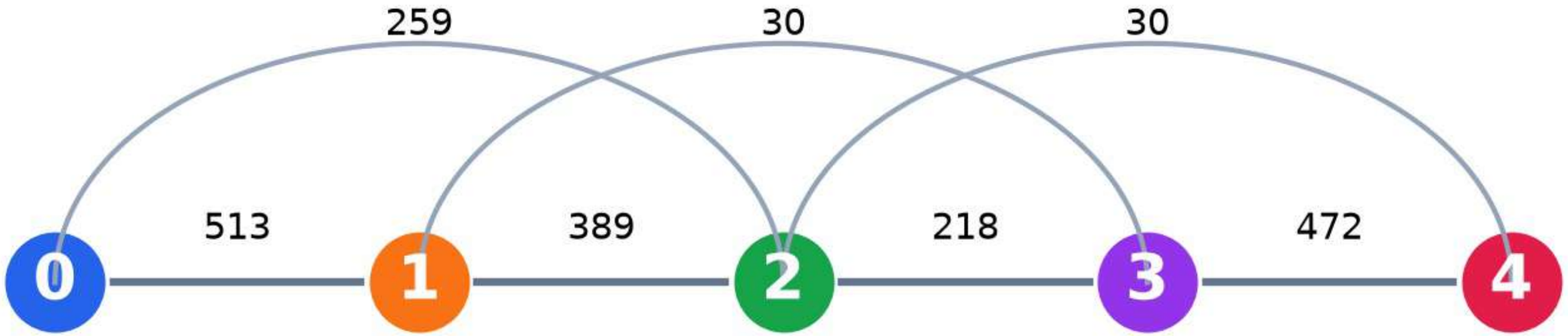
Одна синтетическая сцена показана в двух проекциях при охвате от −70° до +70°. Номера обозначают одни и те же детали.

$$x_{\text{plane}} = f \tan\theta, \qquad x_{\text{cyl}} = f\theta$$



Граф перекрытий: пример

При последовательном совмещении погрешности накапливаются. В этой серии общие детали есть и через один кадр: дополнительные связи позволяют проверить и уточнить общую геометрию.



Накопление ошибок

Синтетический пример: истинный сдвиг между соседями равен 100 px, измеренный — 102 px.

После четырёх переходов ошибка положения достигает 8 px.



Граф перекрытий ряда

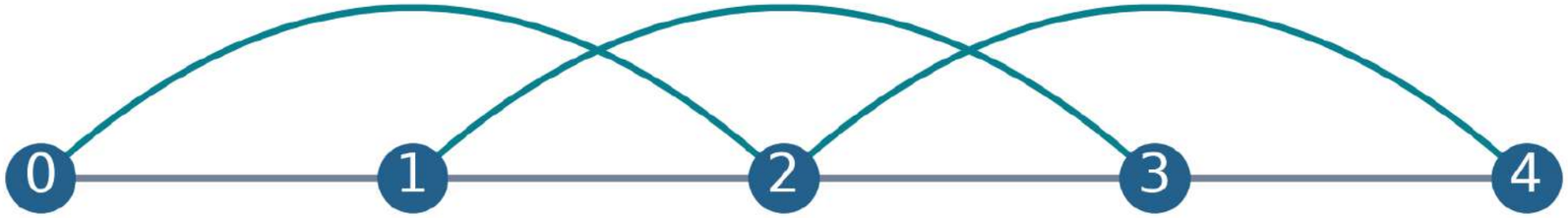
Вершина соответствует кадру; ребро — независимо проверенному набору соответствий.

Перекрытия через кадр создают дополнительные пути между одними и теми же изображениями.

Только соседние кадры

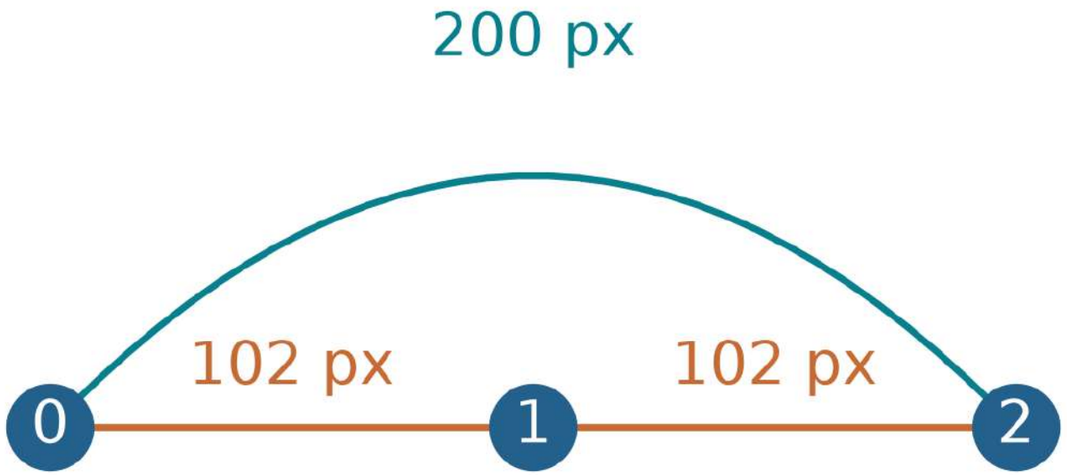


Соседи и кадры через один



Согласованность цикла

- Два соседних перехода дают 204 px; прямое измерение через кадр даёт 200 px.
- Согласовать все измерения точно невозможно: требуется компромисс с учётом их надёжности.

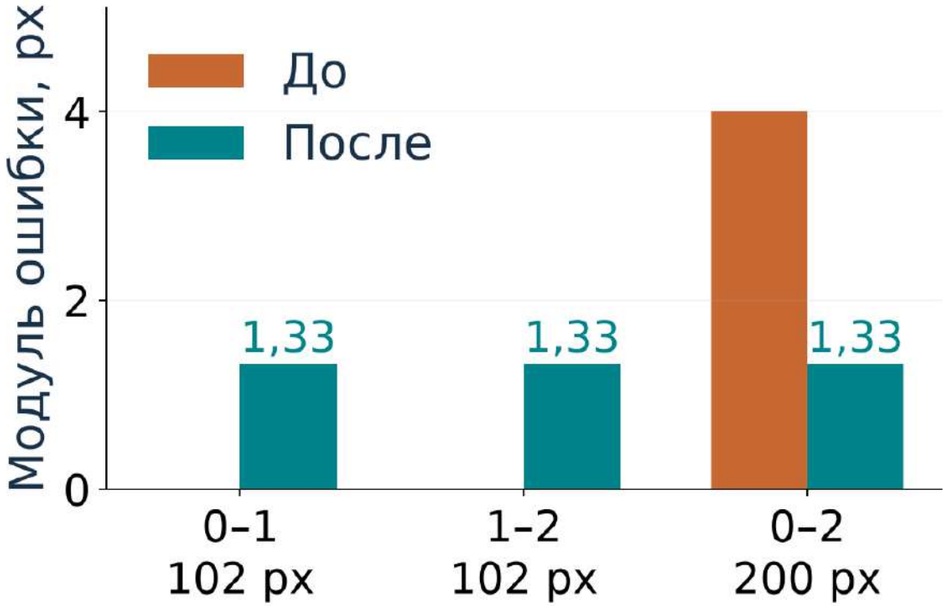


$$102 + 102 - 200 = 4 \text{ px}$$

Совместная оценка положений

Измерения 102 + 102 и 200 противоречат друг другу. Совместная оценка немного изменяет положения кадров, распределяя ошибку между связями.





$$\min_{s_1,s_2}(s_1 - 102)^2 + (s_2 - s_1 - 102)^2 + (s_2 - 200)^2, \qquad s_0 = 0$$

Последовательность построения панорамы

Каждый этап даёт проверяемый результат: точки → пары → преобразования → общий холст → итоговое изображение.

1. Кадры



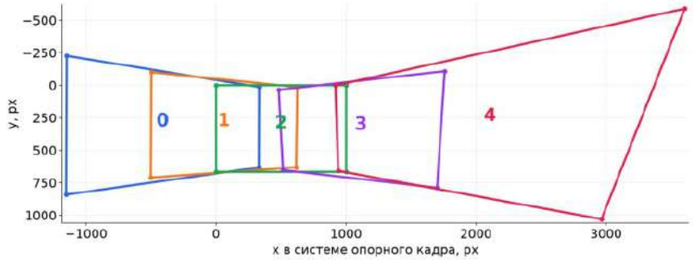
2. Локальные признаки



3. Соответствия



4. Геометрия



5. Объединение

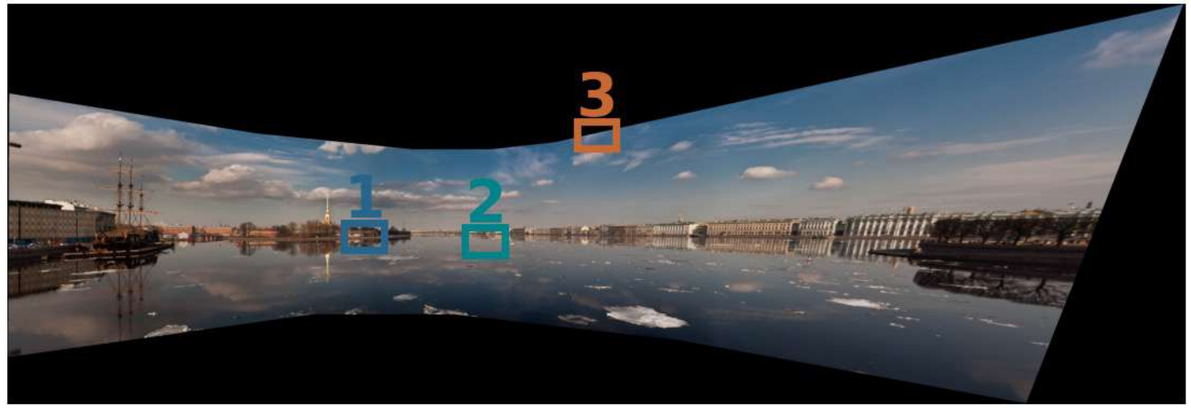


6. Кадрирование

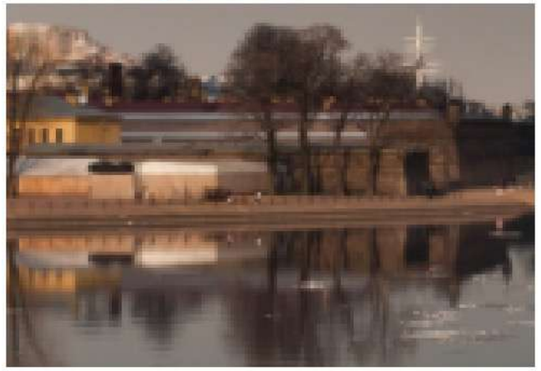


Визуальная проверка панорамы

Контуры выявляют ошибки геометрии, переходы яркости — проблемы смешивания, маска — незаполненные области.



Контуры объектов



Переходы яркости



Граница покрытия



Первоисточники: локальные признаки и геометрия

- Harris & Stephens, 1988 — локализация углов. Rosten & Drummond, 2006 — FAST.
- Lowe, 2004 — масштабно-инвариантные признаки SIFT и критерий различимости соответствий.
- Rublee et al., 2011 — ORB. Fischler & Bolles, 1981 — RANSAC.
- OpenCV: Features2D, findHomography и геометрические преобразования.

Первоисточники: панорамы и оптимизация

- Brown & Lowe, 2007 — автоматическое построение панорам и совместное уточнение.
- Szeliski — Image Alignment and Stitching: A Tutorial.
- OpenCV: калибровка камеры и модуль stitching. SciPy: нелинейные наименьшие квадраты.
- Исходные фотографии: официальные наборы OpenCV boat, box и шахматные доски.