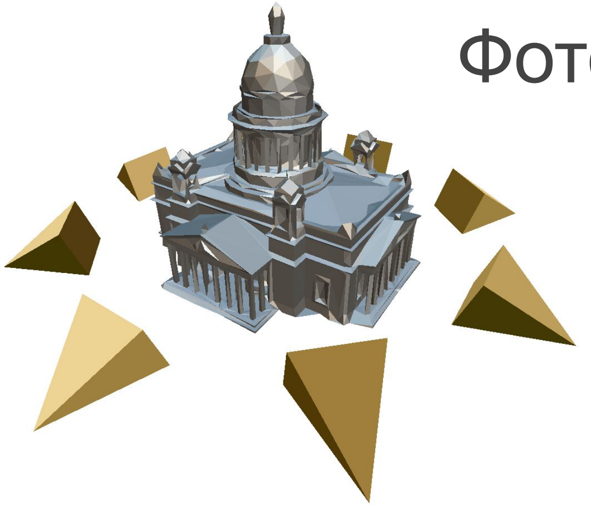
Введение в фотограмметрию Локальные ключевые точки **SIFT**



Фотограмметрия. Лекция 2

- SIFT - оставшиеся детали
- Бинарный MLDB дескриптор
- Задание на SIFT
- OpenMP, OpenCV

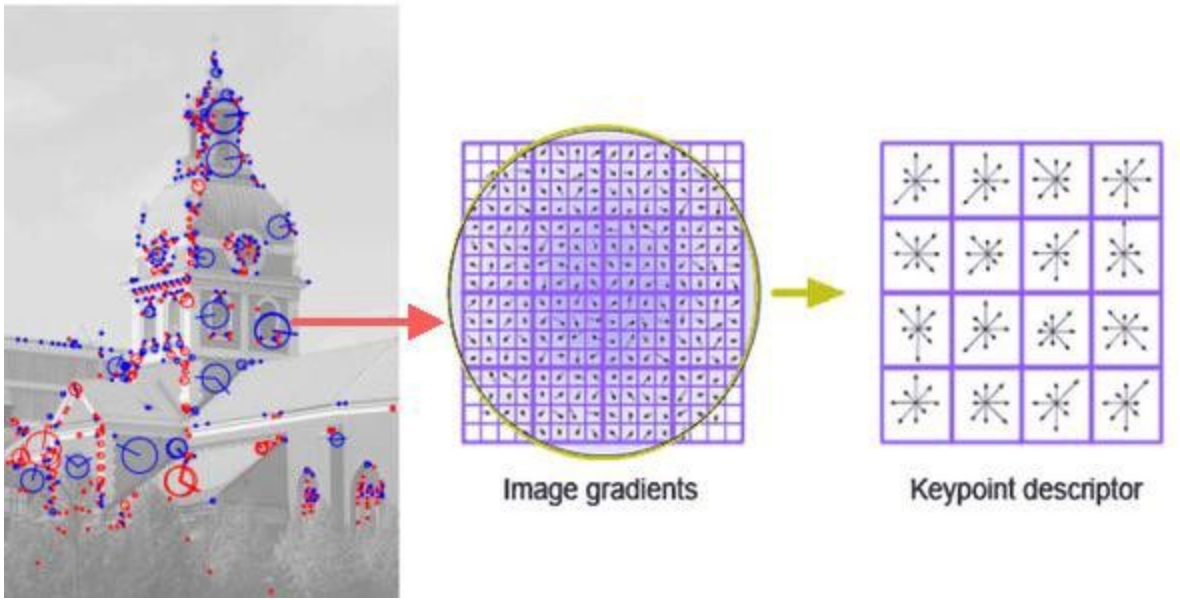
Полярный Николай
polarnick239@gmail.com

SIFT Key Points: **Detector & Descriptor**

1) Выбираем ключевыми точками “пятна”
2) У каждой точки инвариантно определяем (на базе окрестности точки):
   2.1) Поворот (голосованием за самое популярное направление градиента)
   2.2) Масштаб (за счет обнаружения “пятна” на конкретном уровне **DoG**)
3) Для каждой точки строим дескриптор из **HoG** (из гистограмм градиентов) по квадратной окрестности точки (повернутой и отмасштабированной)

Статья: Distinctive Image Features from Scale-Invariant Keypoints, G. Lowe, 2004

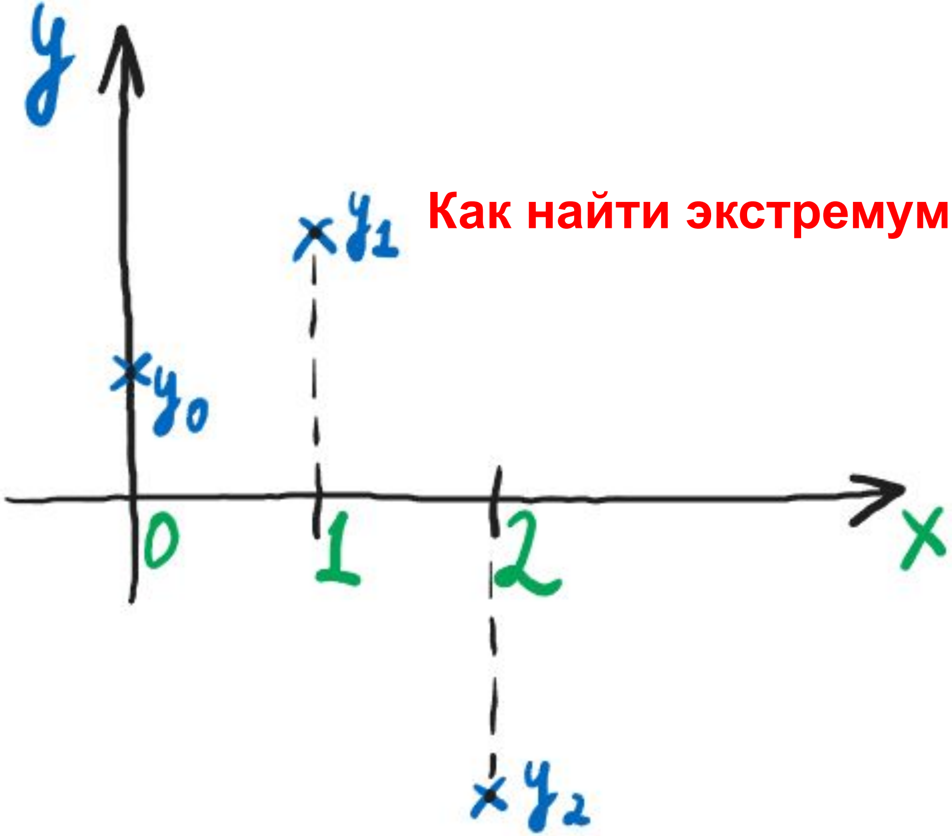
SIFT Key Points: **Detector & Descriptor**



Источник: https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774

Вопрос 1: как обнаружать экстремум еще точнее?
Вопрос 2: как обнаружать поворот точки точнее?
Вопрос 3: как ускорить алгоритм детектирования?
Вопрос 4: что в статье делается по-другому?
Вопрос 5: на что влияет размер патча?
Вопрос 6: есть ли проблемы из-за квадратности окрестности при построении дескриптора? Как исправить?
Вопрос 7: какой этап занимает больше всего времени?
Вопрос 8: какой Detector&Descriptor для **3D** облака точек?

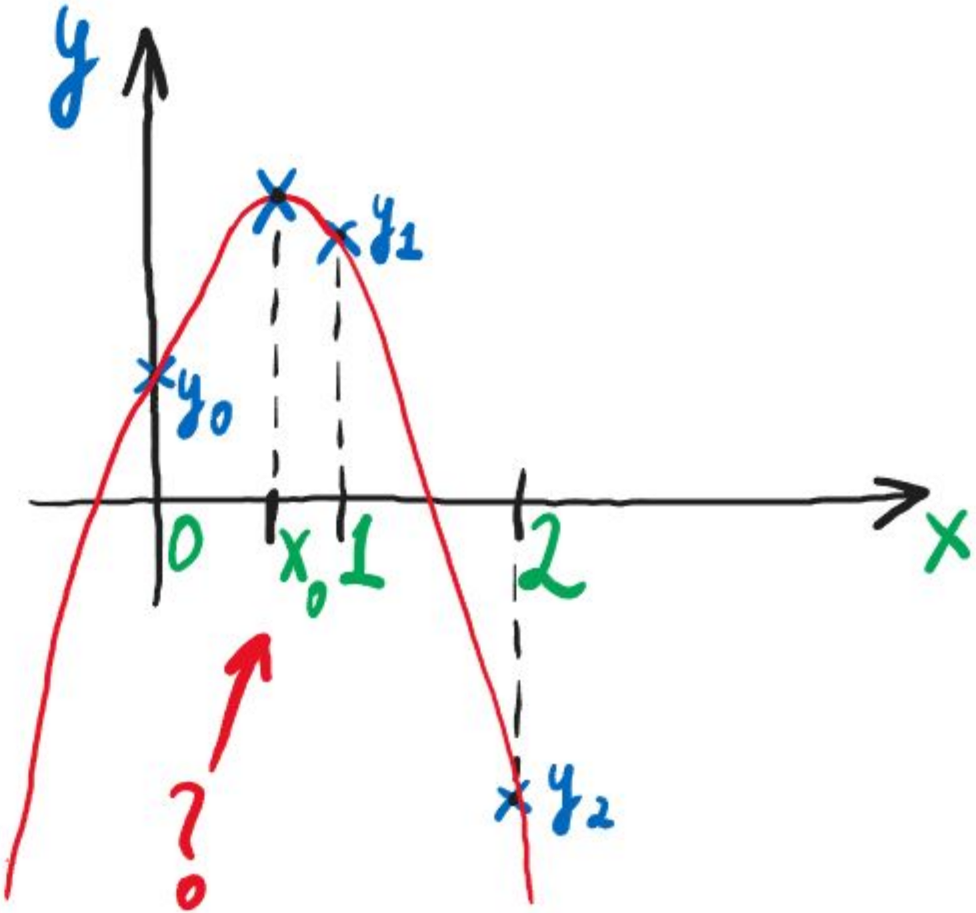
Вопрос 1: как обнаружать экстремум еще точнее?

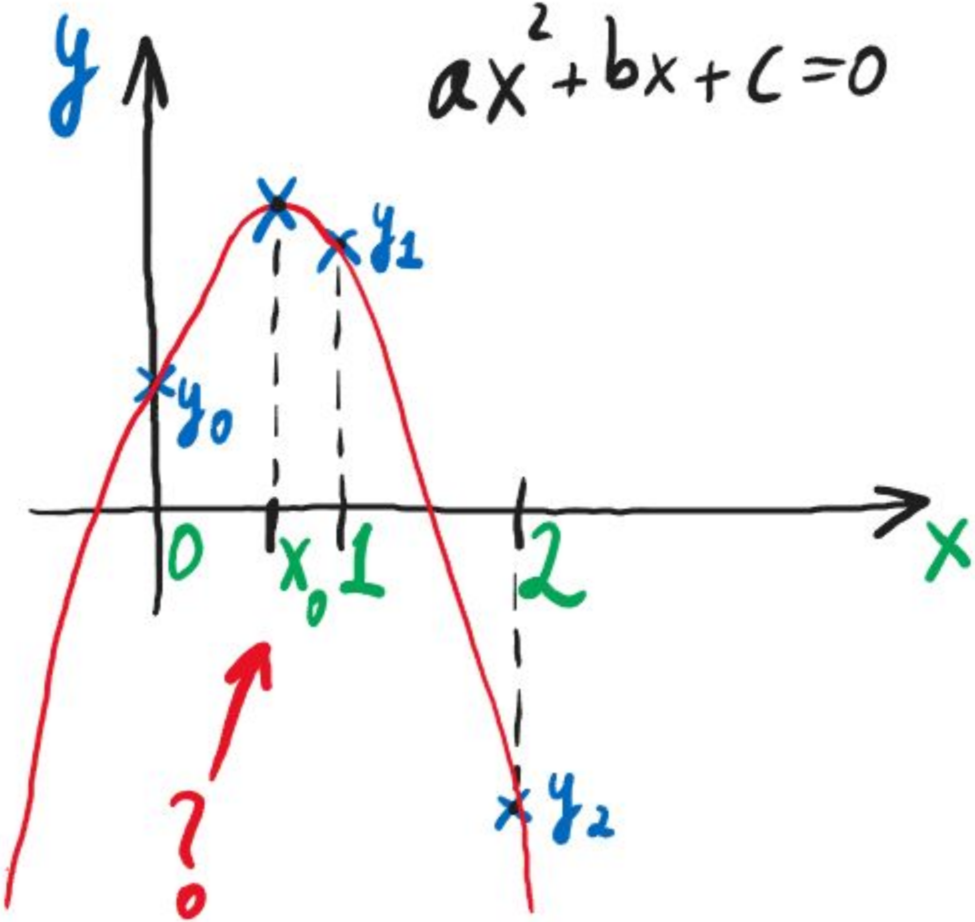


Как найти экстремум?

Вопрос 1: как обнаружать экстремум еще точнее?

Как найти экстремум?

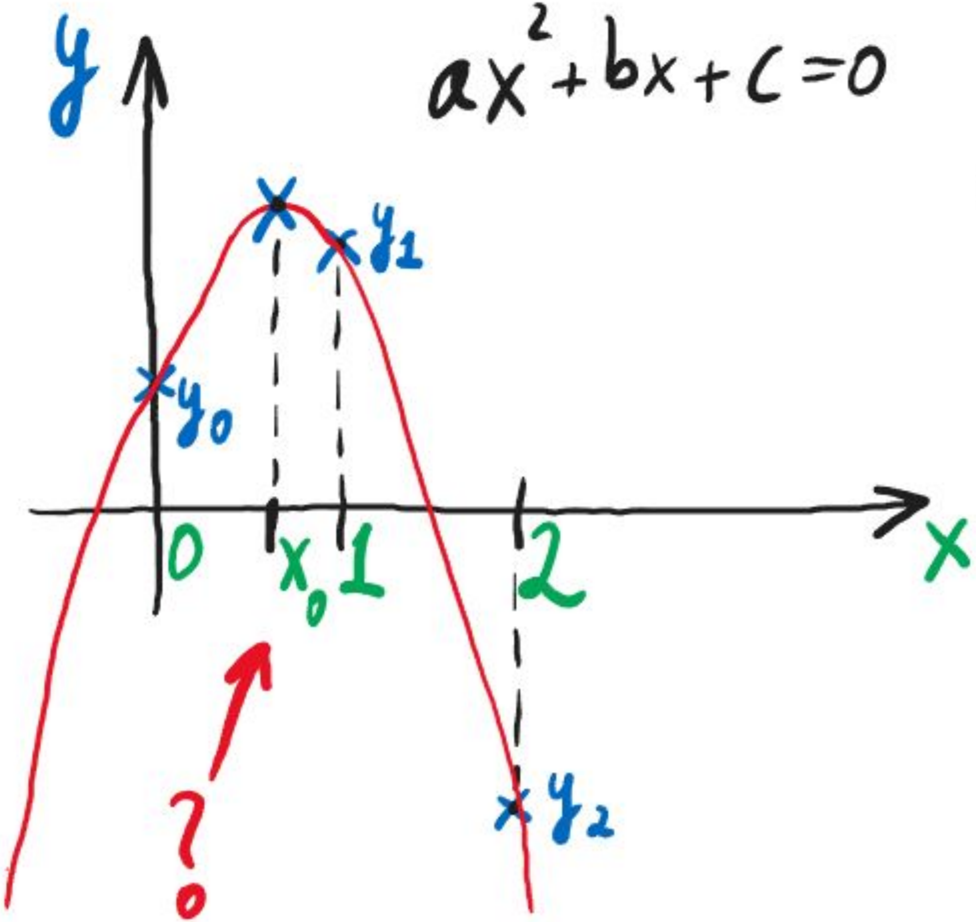

y
y_1
y_0
0
x_0
1
2
x
?
y_2



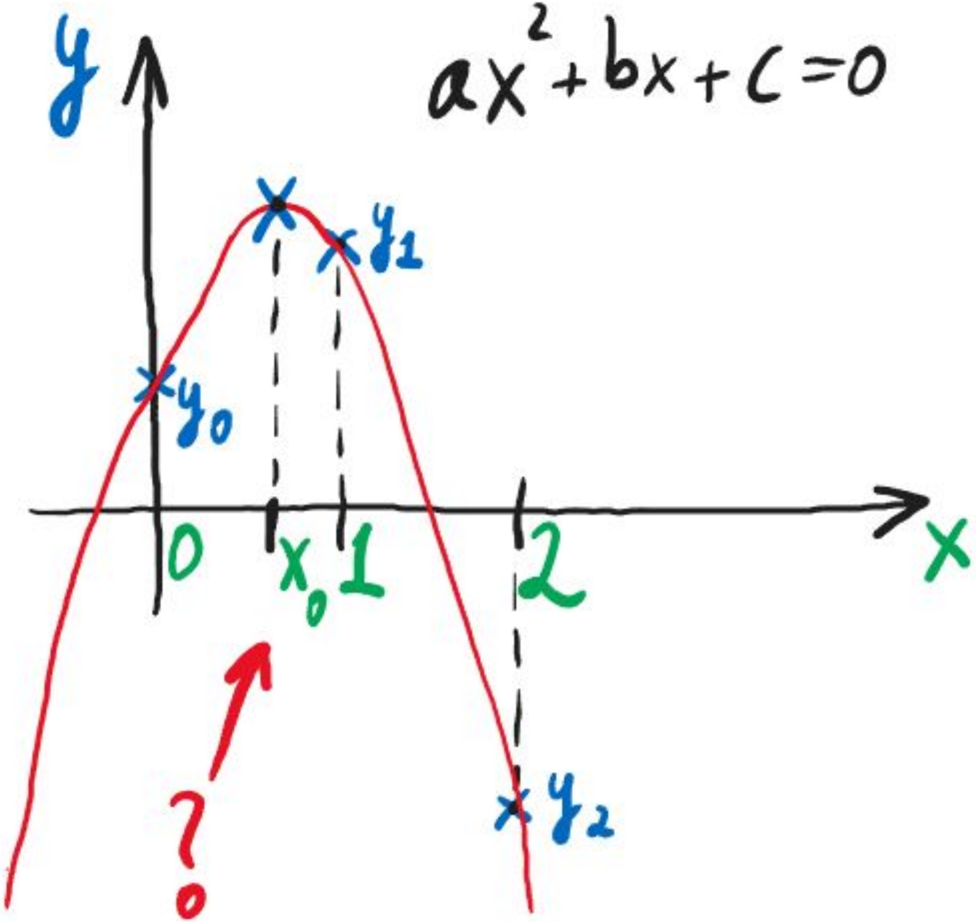
$ax^2+bx+c=0$
y
x
0
x_0
1
2
y_0
y_1
y_2
?

$ax^2+bx+c=0$

$$\begin{cases} a\cdot ? + b\cdot ? + c = ? \\ a\cdot ? + b\cdot ? + c = ? \\ a\cdot ? + b\cdot ? + c = ? \end{cases}$$

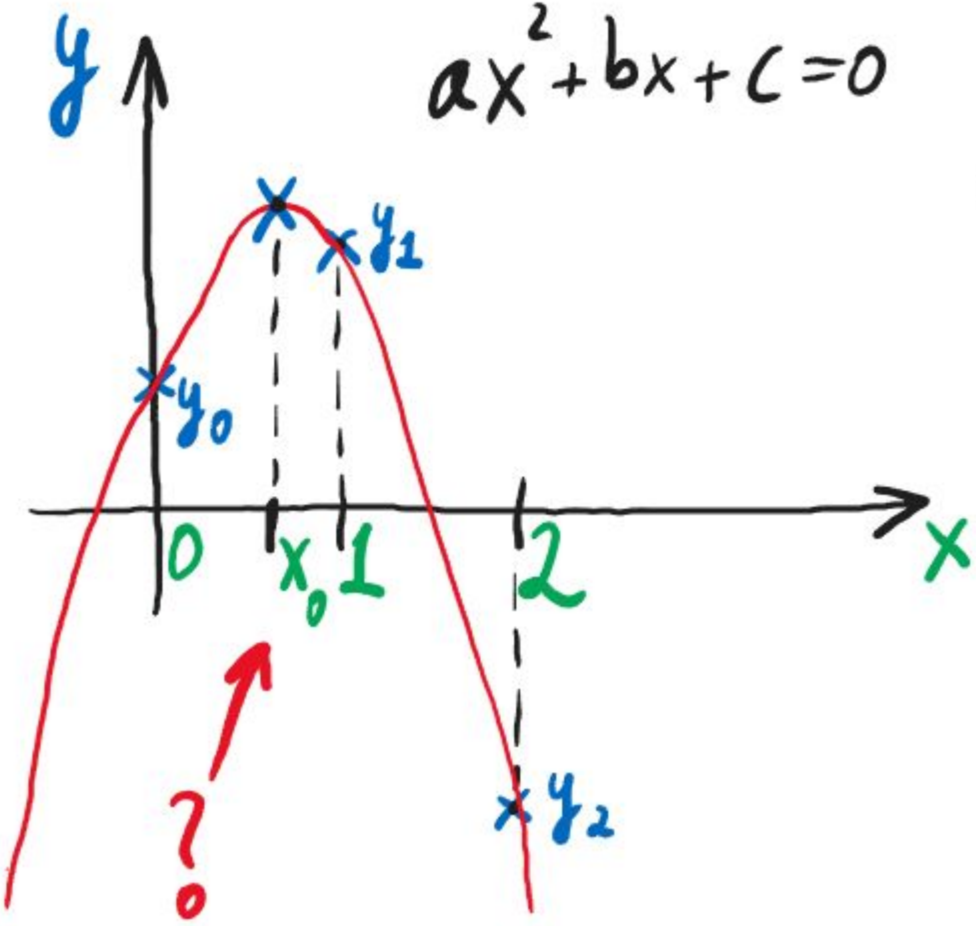


$$ax^2+bx+c=0$$

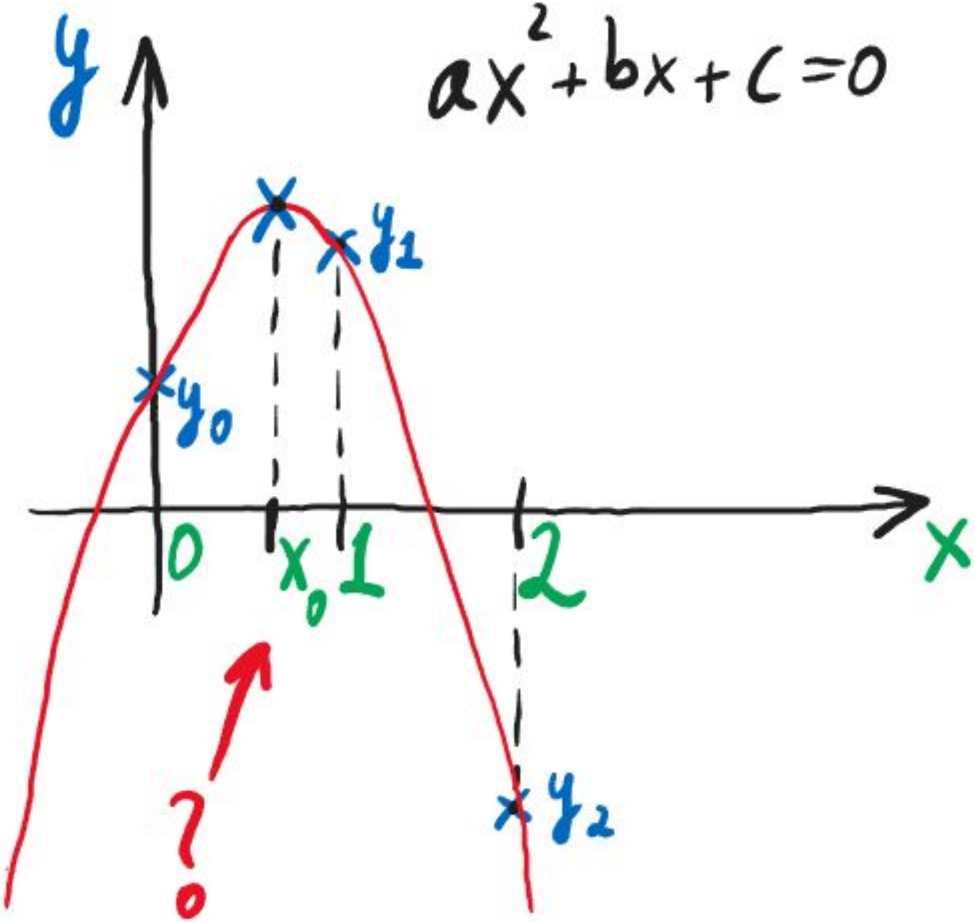


$$\begin{cases} a\cdot 0^2+b\cdot 0+c=y_0 \\ a\cdot 1^2+b\cdot 1+c=y_1 \\ a\cdot 2^2+b\cdot 2+c=y_2 \end{cases}$$

$$ax^2+bx+c=0$$



$$\begin{cases} \cancel{a\cdot 0^2}+\cancel{b\cdot 0}+\boxed{c=y_0} \\ a\cdot 1^2+b\cdot 1+c=y_1 \\ a\cdot 2^2+b\cdot 2+c=y_2 \end{cases}$$

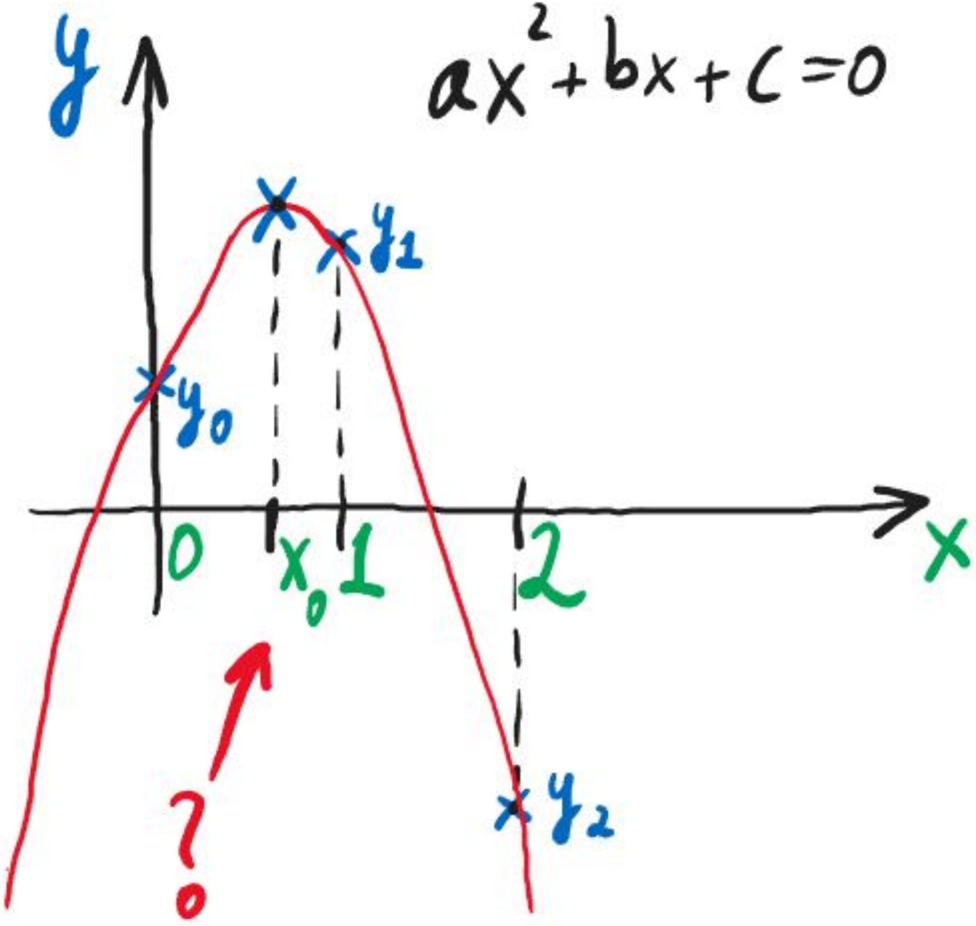


$$ax^2+bx+c=0$$

$$\begin{cases} \cancel{a\cdot 0^2} + \cancel{b\cdot 0} + \boxed{c = y_0} \\ a\cdot 1^2 + b\cdot 1 + c = y_1 \quad (2) \\ a\cdot 2^2 + b\cdot 2 + c = y_2 \quad (3) \end{cases}$$

$$(3)-2\cdot(2):\ 2\cdot a - c = y_2 - 2y_1$$

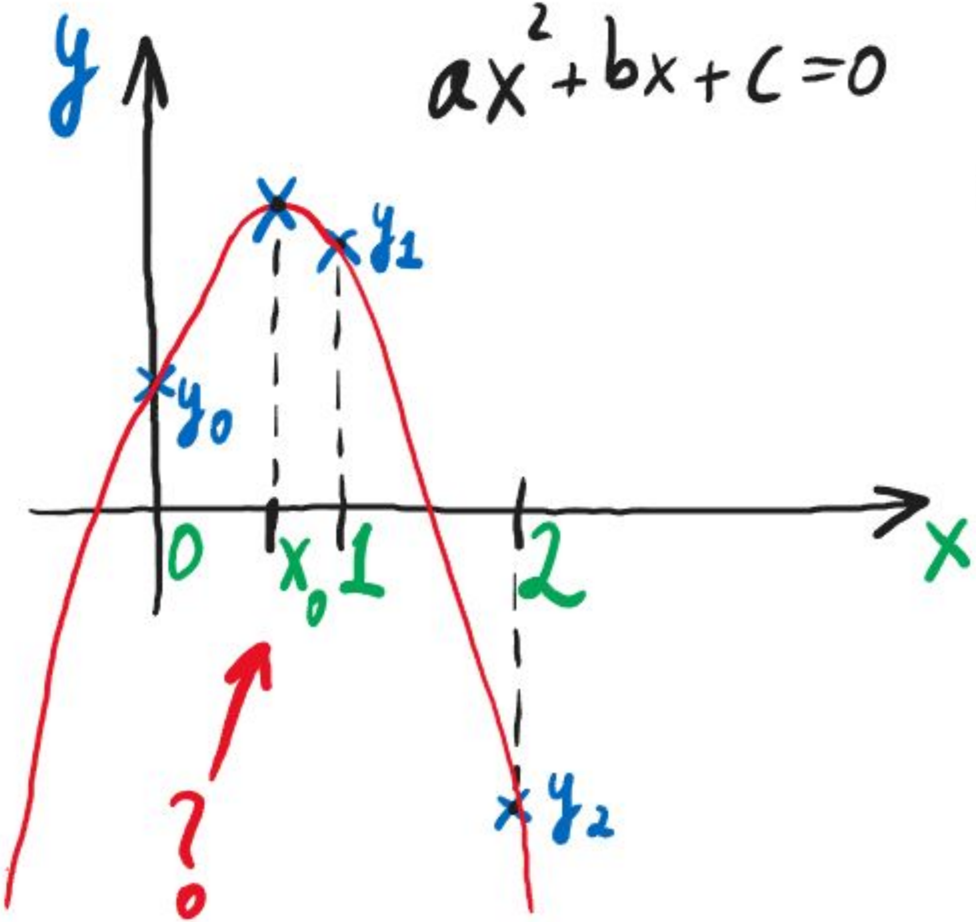
$$ax^2+bx+c=0$$



$$\begin{cases} \cancel{a\cdot 0^2} + \cancel{b\cdot 0} + \boxed{c = y_0} \\ a\cdot 1^2 + b\cdot 1 + c = y_1 \quad (2) \\ a\cdot 2^2 + b\cdot 2 + c = y_2 \quad (3) \end{cases}$$

$$(3)-2\cdot(2):\ 2\cdot a - \boxed{c} = y_2 - 2y_1$$

$$a = \frac{y_2 - 2y_1 + y_0}{2}$$



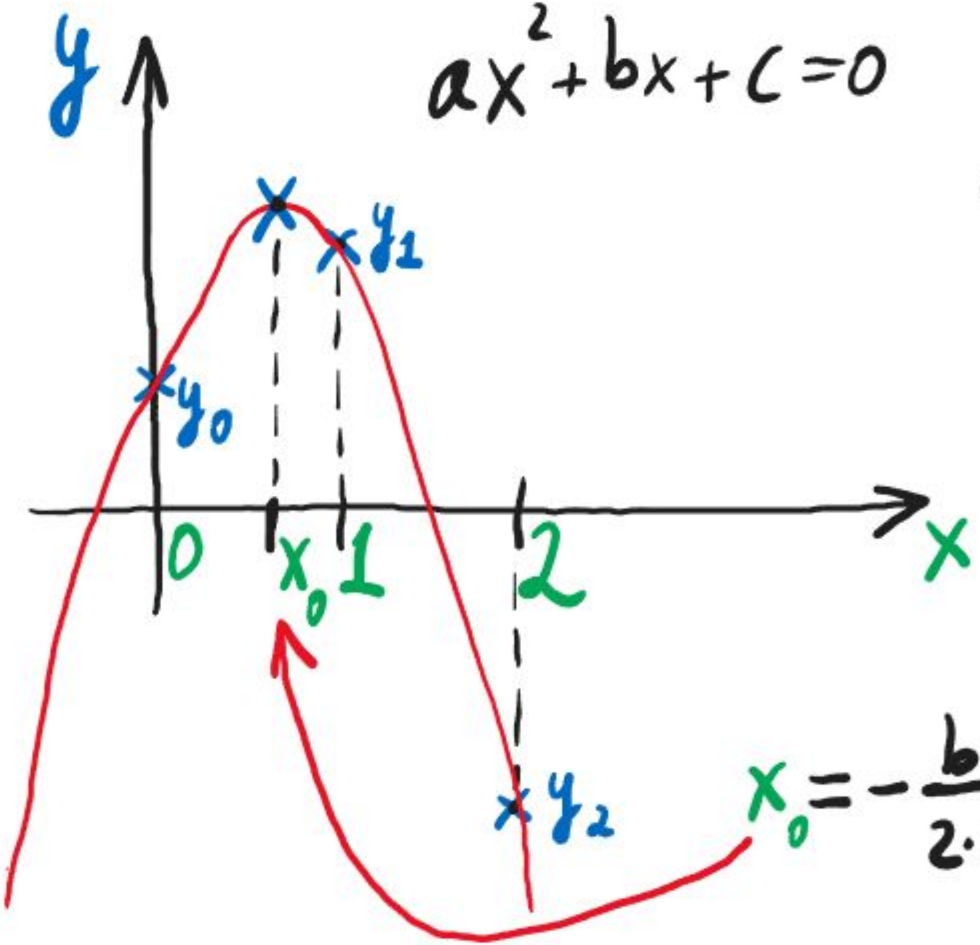
$$ax^2+bx+c=0$$

$$\begin{cases} \cancel{a\cdot 0^2} + \cancel{b\cdot 0} + \boxed{c = y_0} \\ a\cdot 1^2 + b\cdot 1 + c = y_1 \quad (2) \\ a\cdot 2^2 + b\cdot 2 + c = y_2 \quad (3) \end{cases}$$

$(3)-2\cdot(2)$: $2\cdot a - \boxed{c} = y_2 - 2y_1$

$$a = \frac{y_2 - 2y_1 + y_0}{2}$$

(2): $b = y_1 - y_0 - a$



$$ax^2+bx+c=0$$

$$\begin{cases} \cancel{a\cdot 0^2} + \cancel{b\cdot 0} + \boxed{c = y_0} \\ a\cdot 1^2 + b\cdot 1 + c = y_1 & (2) \\ a\cdot 2^2 + b\cdot 2 + c = y_2 & (3) \end{cases}$$

$$(3)-2\cdot(2):\ 2\cdot a - \boxed{c} = y_2 - 2y_1$$

$$a = \frac{y_2 - 2y_1 + y_0}{2}$$

$$(2):\ b = y_1 - y_0 - a$$

$$x_0 = -\frac{b}{2\cdot a}$$

Вопрос 1: как обнаружать экстремум еще точнее?

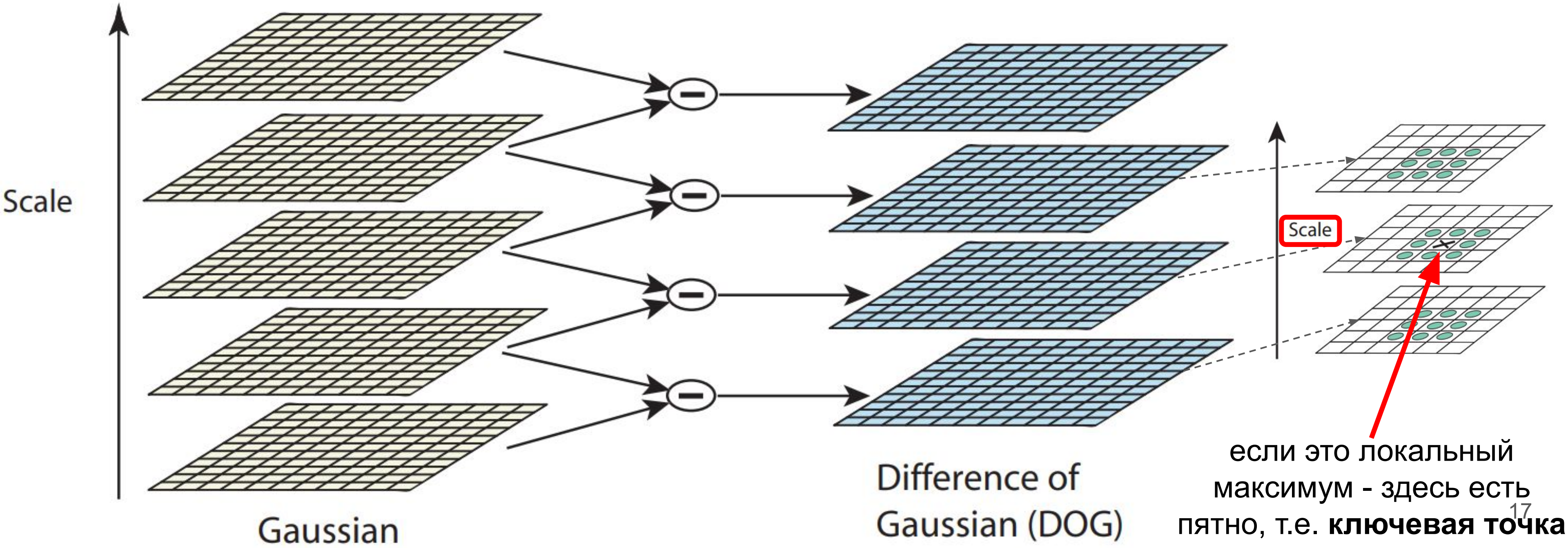
Но это 2D случай: **f(x) = y**

Такой ли вид имеет наша функция?

Вопрос 1: как обнаружать экстремум еще точнее?

Но это 2D случай: **f(x) = y**

Такой ли вид имеет наша функция?

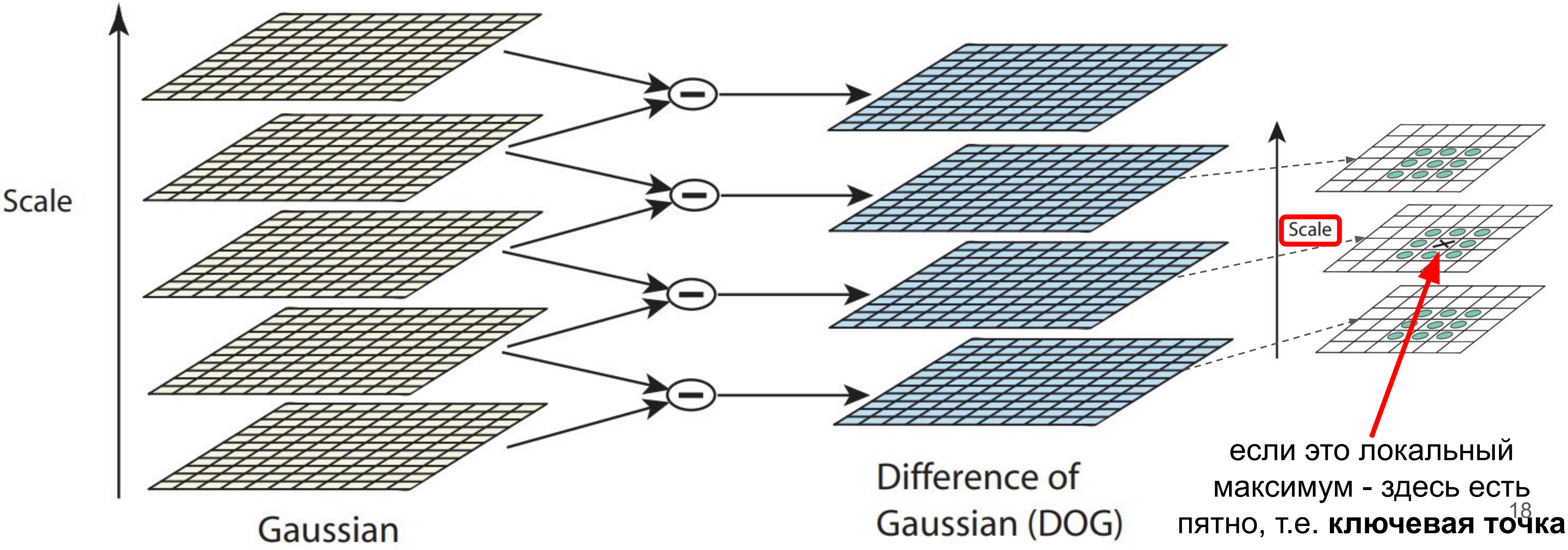


если это локальный максимум - здесь есть пятно, т.е. **ключевая точка**

Вопрос 1: как обнаружать экстремум еще точнее?

Но это 2D случай: **f(x) = y**

Наша функция имеет вид **f(x, y) = z** (яркость в каждом пикселе)



если это локальный максимум - здесь есть пятно, т.е. **ключевая точка**

$$f(x) = ax^2 + bx + c$$

$$x_{min} = \arg\min_x f(x)$$

Как по-другому найти экстремум?

$$f(x) = ax^2 + bx + c$$

$$x_{min} = \underset{x}{\operatorname{argmin}} f(x)$$

$$f'(x_{min}) = 0$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c \qquad x_{min} = \arg\min_x f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2} f''(x_0) dx^2$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c \qquad x_{min} = \underset{x}{\arg\min}\, f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2} f''(x_0)dx^2$$

$$f'(x) \sim 0 + f'(x_0) + \frac{1}{\cancel{2}} f''(x_0)dx \cdot \cancel{2}$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c \qquad x_{min} = \arg\min_x f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2}f''(x_0)dx^2$$

$$f'(x) \sim 0 \quad + f'(x_0) \quad + \frac{1}{\cancel{2}}f''(x_0)dx \cdot \cancel{2} = 0$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c$$

$$x_{min} = \underset{x}{\arg\min}\, f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)\,dx + \frac{1}{2} f''(x_0)\,dx^2$$

$$f'(x) \sim 0 + f'(x_0) + \frac{1}{\cancel{2}} f''(x_0)\,dx \cdot \cancel{2} = 0$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c$$

$$x_{min} = \underset{x}{\operatorname{argmin}} f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2} f''(x_0) dx^2$$

$$f'(x) \sim 0 + f'(x_0) + \frac{1}{\cancel{2}} f''(x_0) dx \cdot \cancel{2} = 0$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = ?$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c \qquad x_{min} = \underset{x}{\operatorname{argmin}} f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2} f''(x_0) dx^2$$

$$f'(x) \sim 0 + f'(x_0) + \frac{1}{\cancel{2}} f''(x_0) dx \cdot \cancel{2} = 0$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора

$$f(x) = ax^2 + bx + c$$

$$x_{min} = \underset{x}{\operatorname{argmin}} f(x)$$

$$f'(x_{min}) = 0$$

$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2} f''(x_0) dx^2$$

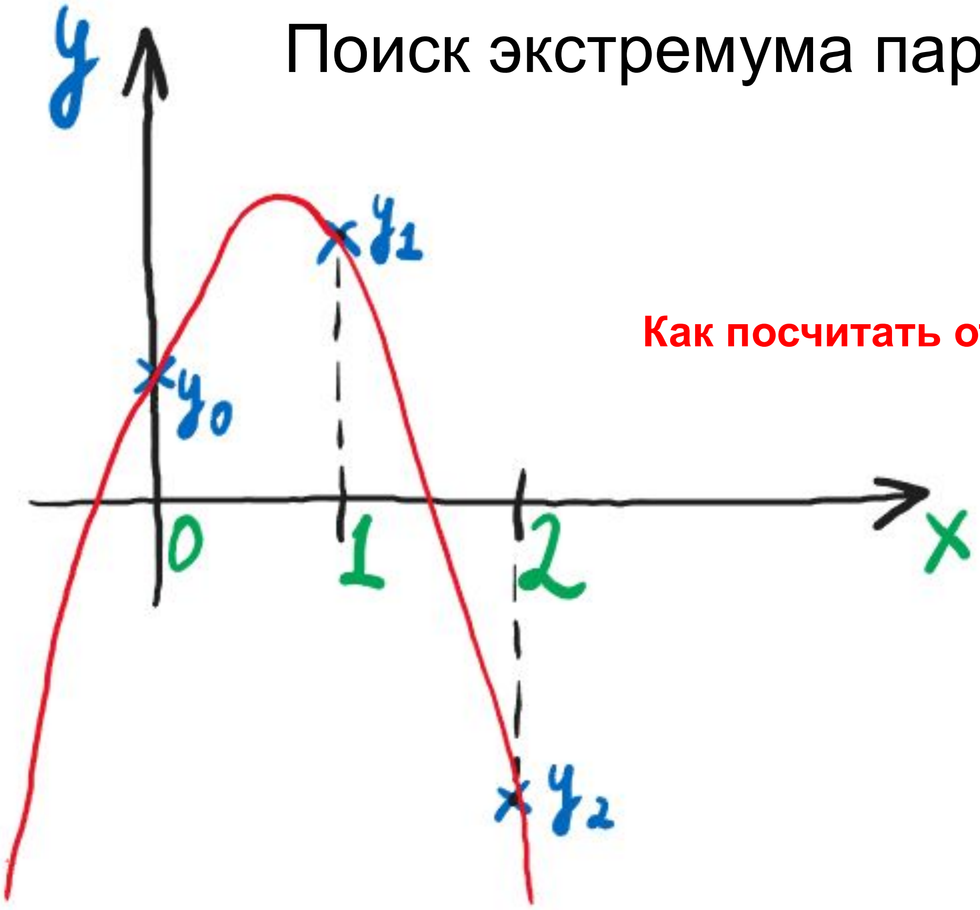
$$f'(x) \sim 0 + f'(x_0) + \frac{1}{\cancel{2}} f''(x_0) dx \cancel{2} = 0$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

Можем ли мы по этим формулам написать найти экстремум?

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора

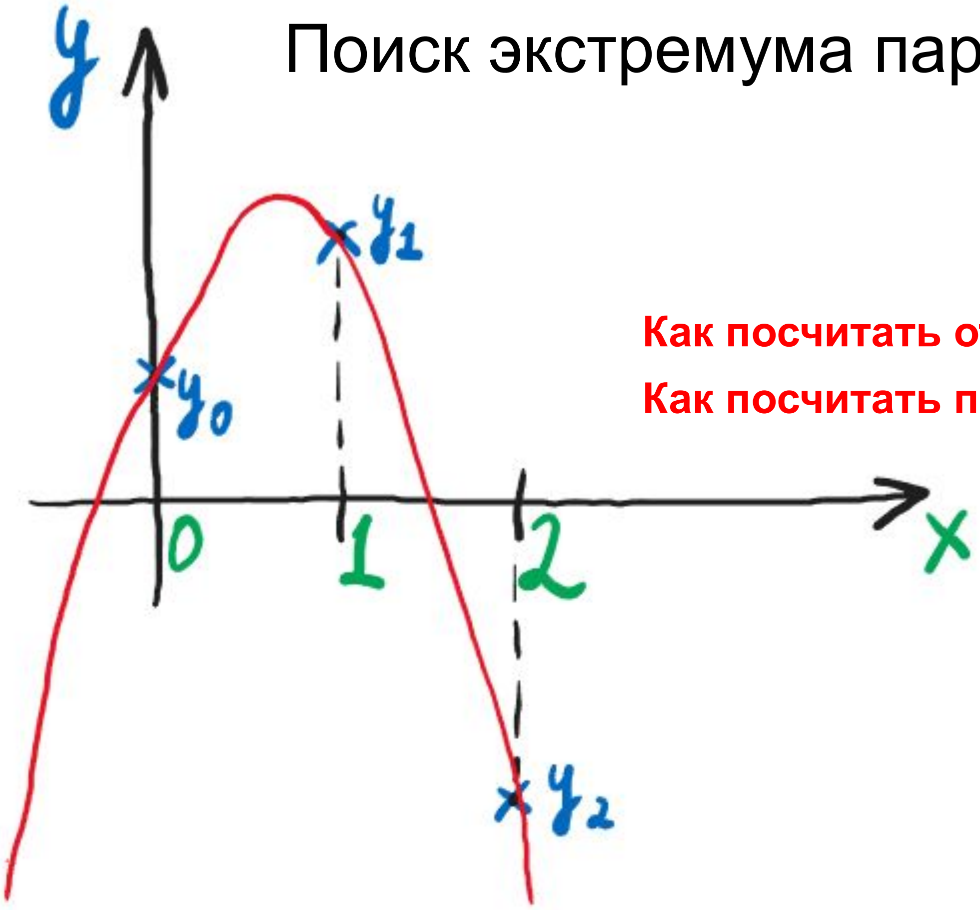


Как посчитать ответ если мы знаем эти три точки?

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



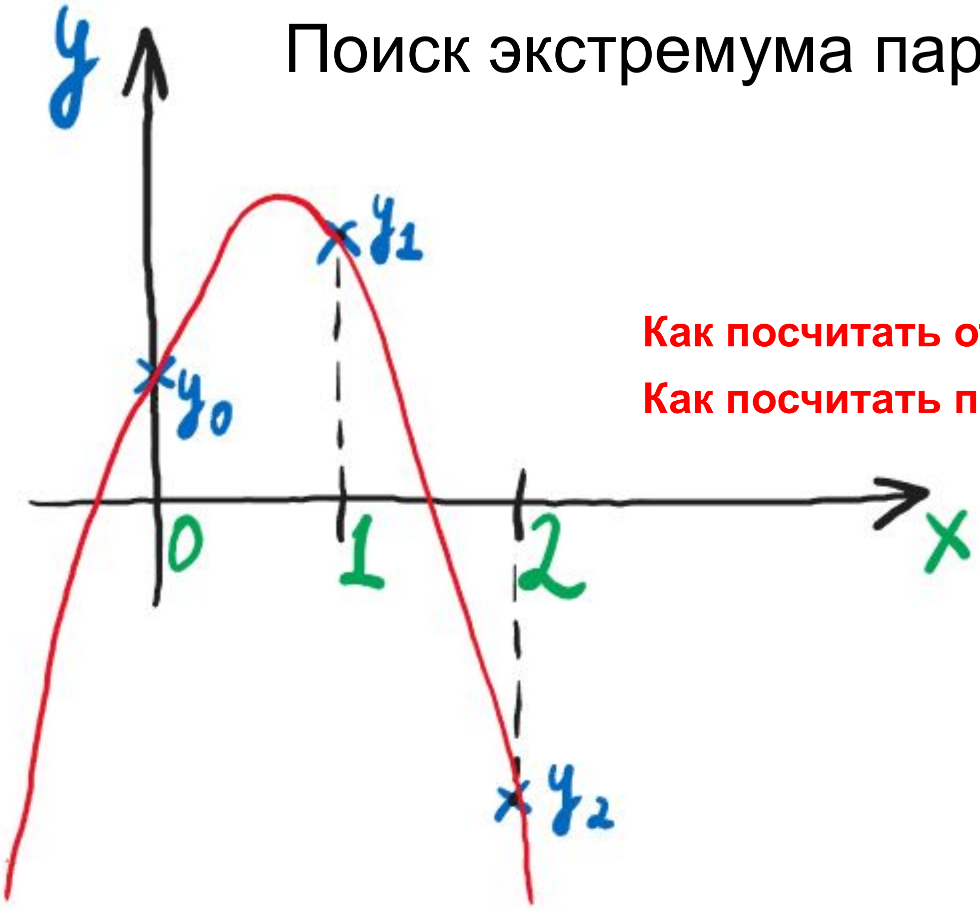
Как посчитать ответ если мы знаем эти три точки?

Как посчитать производные?

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



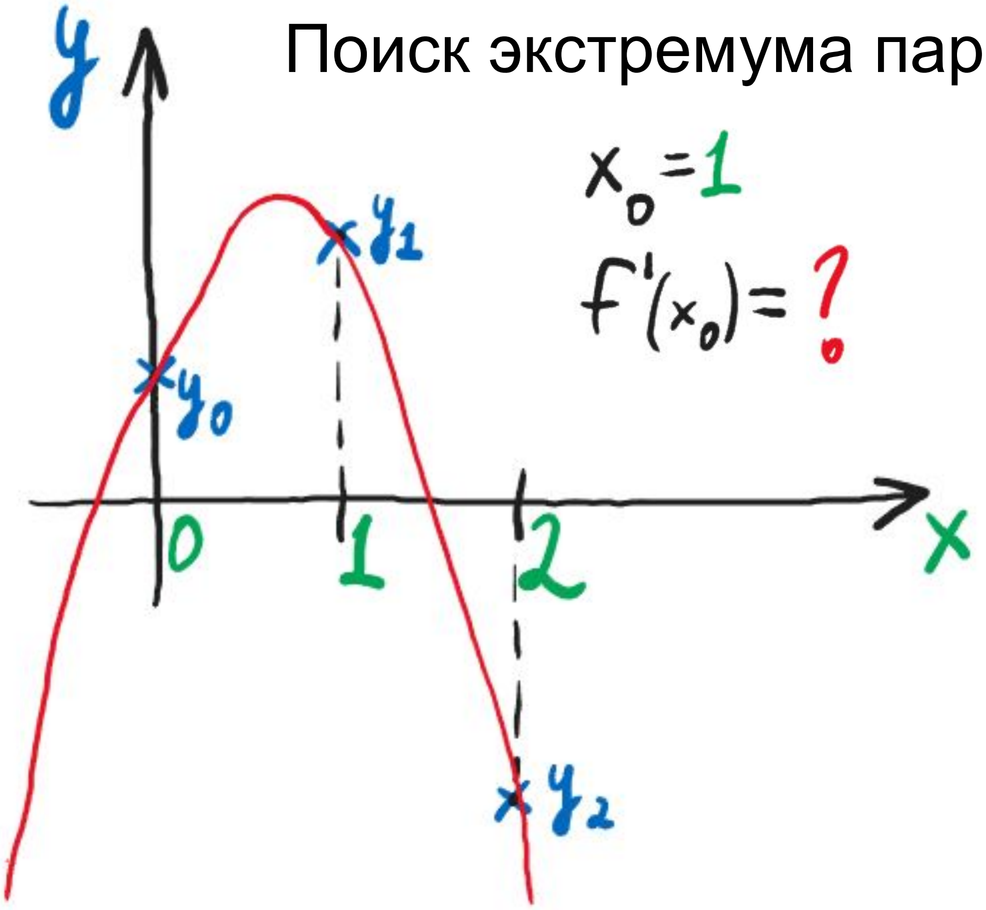
Как посчитать ответ если мы знаем эти три точки?

Как посчитать производные?

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



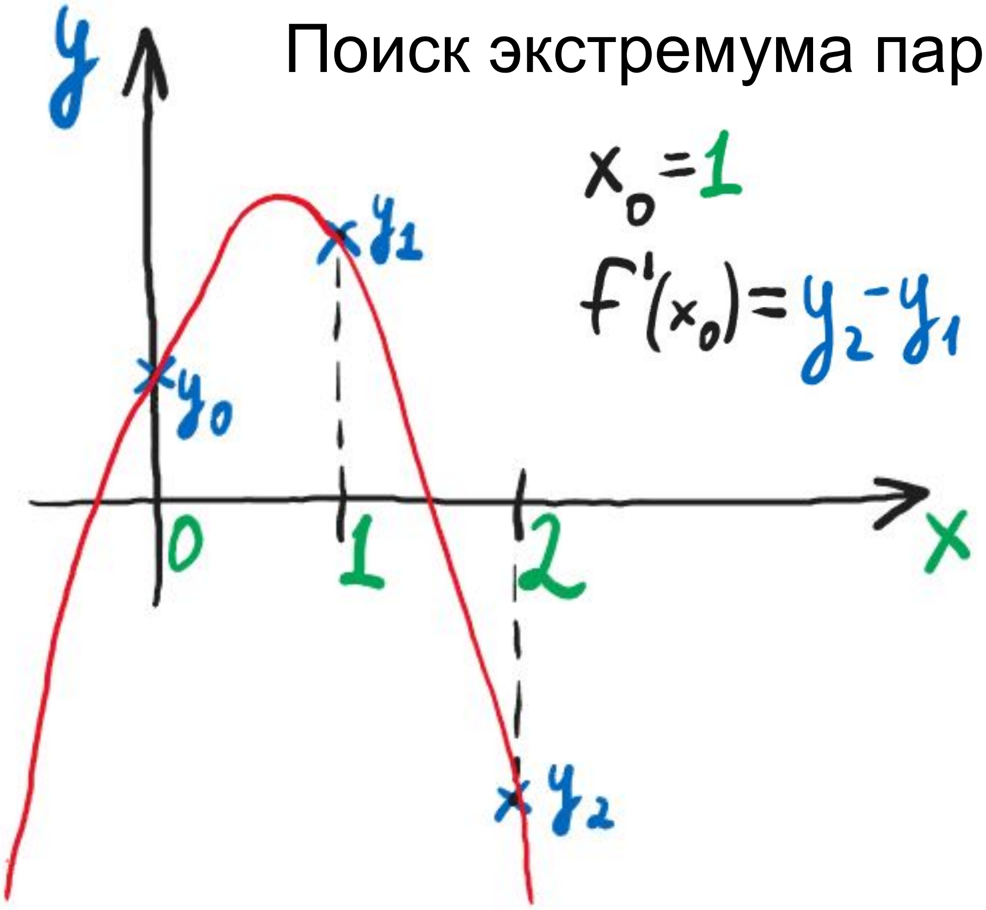
$x_0 = 1$

$f'(x_0) = ?$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



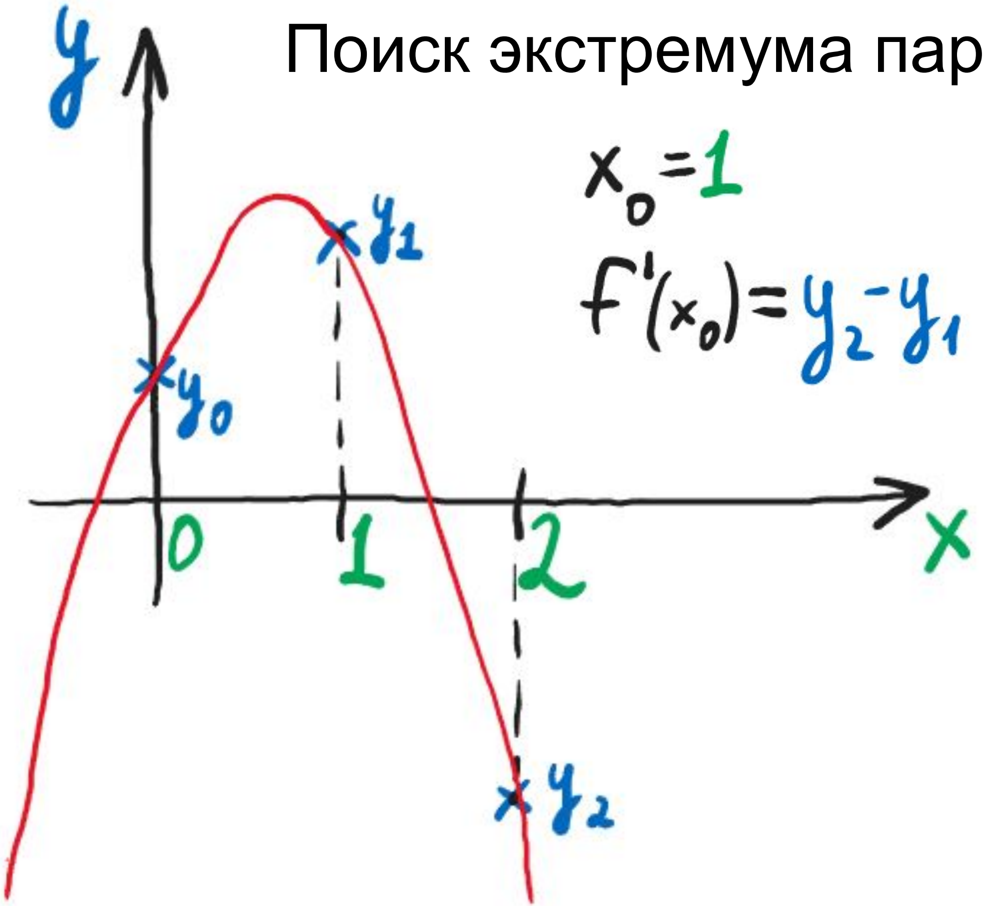
$$x_0 = 1$$

$$f'(x_0) = y_2 - y_1$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



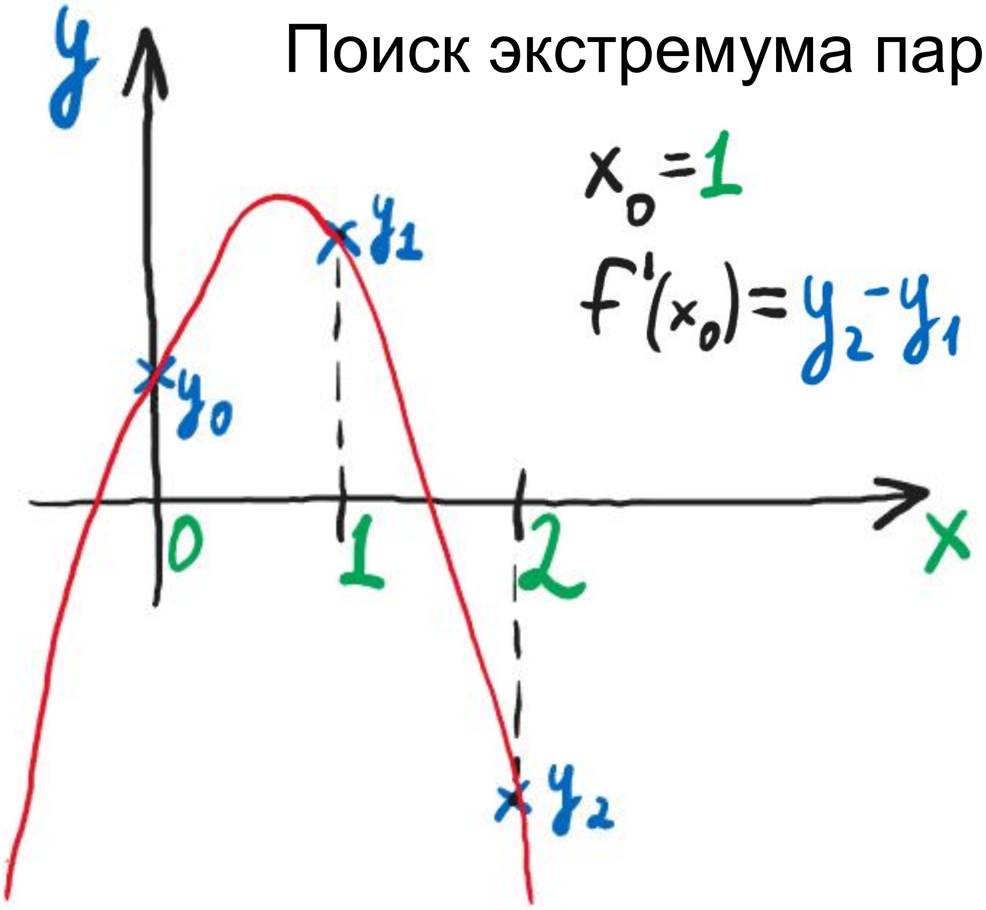
$x_0 = 1$

$f'(x_0) = y_2 - y_1 \quad f''(x_0) = ?$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



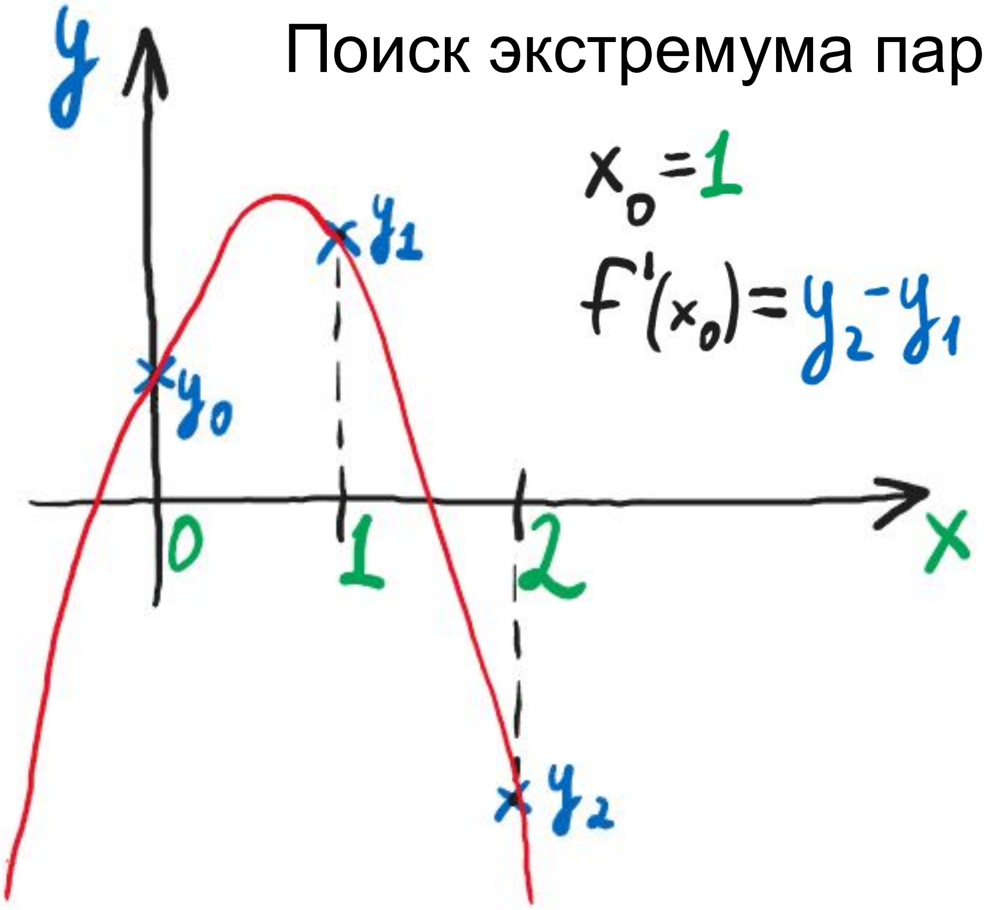
$$x_0 = 1$$

$$f'(x_0) = y_2 - y_1 \qquad f''(x_0) = (f'(x_0))' = ?$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



$$x_0 = 1$$

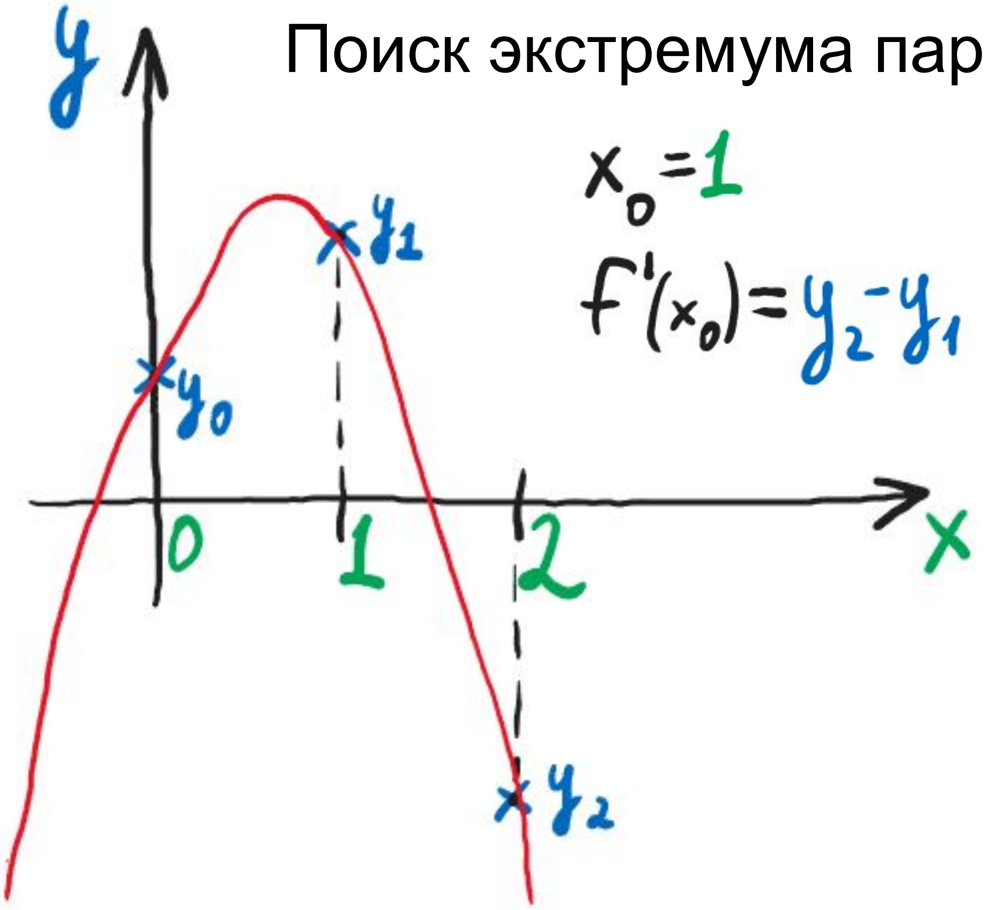
$$f'(x_0) = y_2 - y_1$$

$$f''(x_0) = (f'(x_0))' = f'(1) - f'(0) =$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



$$x_0 = 1$$

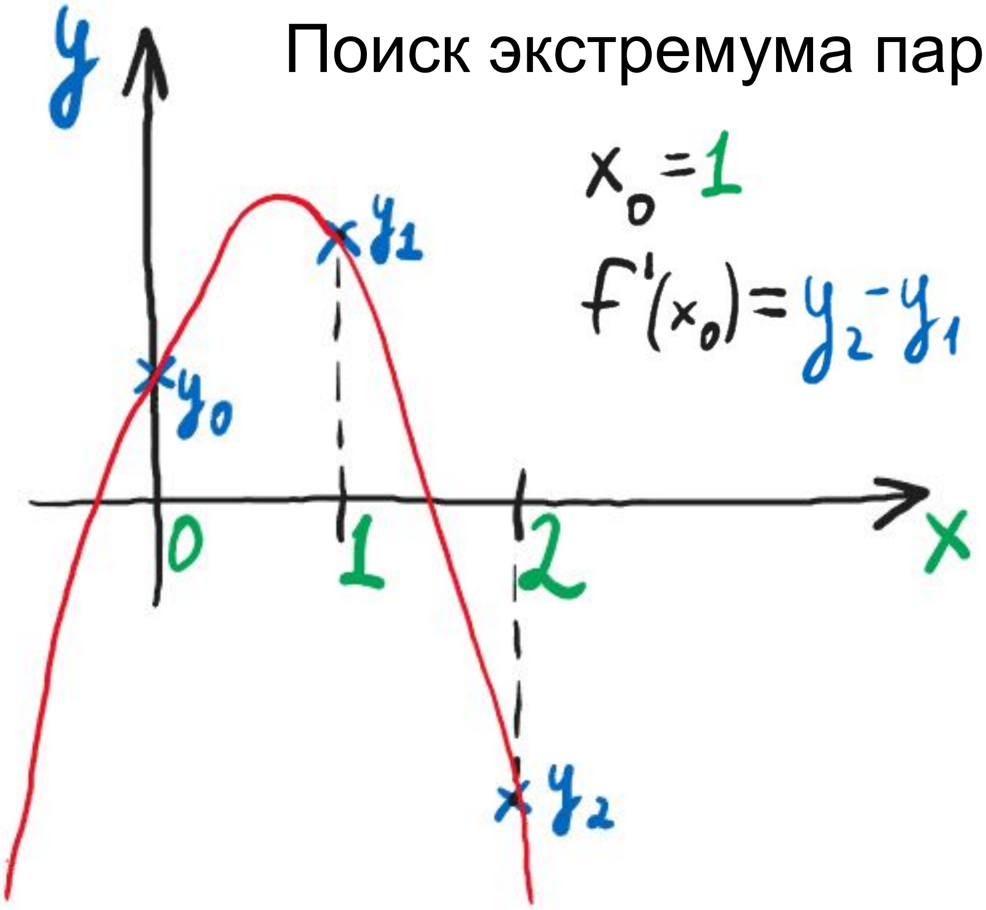
$$f'(x_0) = y_2 - y_1$$

$$f''(x_0) = \left(f'(x_0)\right)' = f'(1) - f'(0) = (y_2 - y_1) - (y_1 - y_0)$$

$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Поиск экстремума параболы через ряд Тейлора



$$x_0 = 1$$

$$f'(x_0) = y_2 - y_1$$

$$f''(x_0) = \left(f'(x_0)\right)' = f'(1) - f'(0) = (y_2 - y_1) - (y_1 - y_0) = y_2 - 2 \cdot y_1 + y_0$$

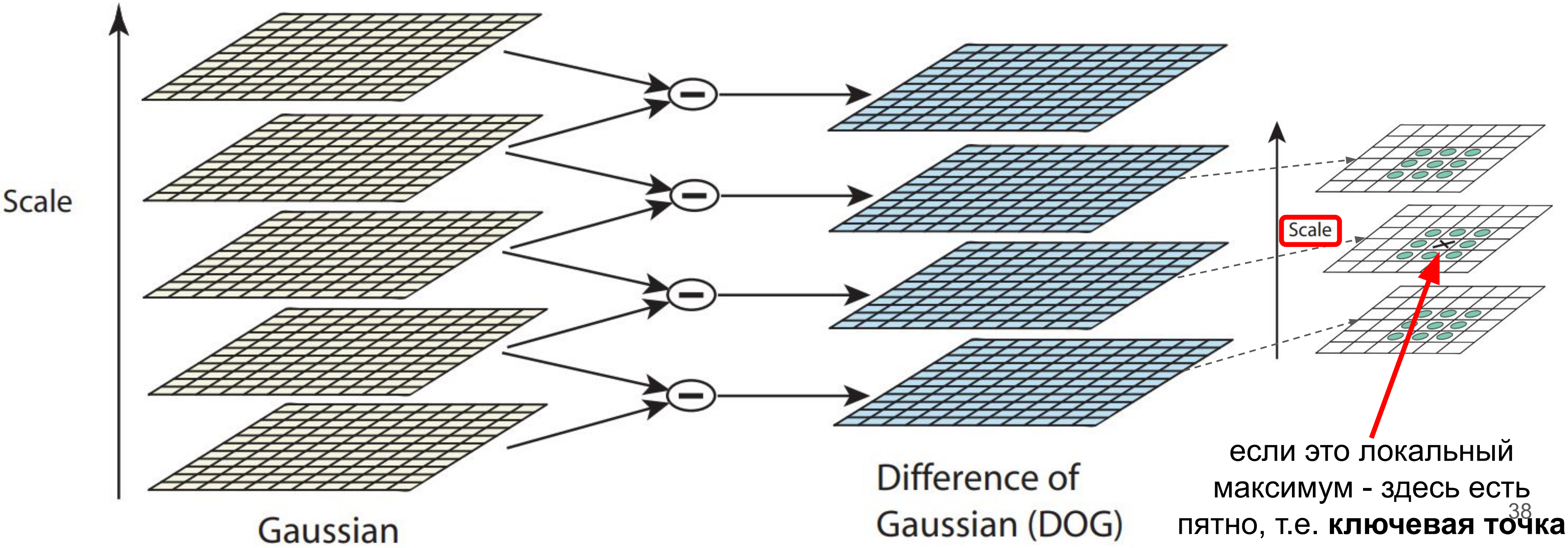
$$dx = -\frac{f'(x_0)}{f''(x_0)}$$

$$x_{min} = x_0 + dx$$

Вопрос 1: как обнаружать экстремум еще точнее?

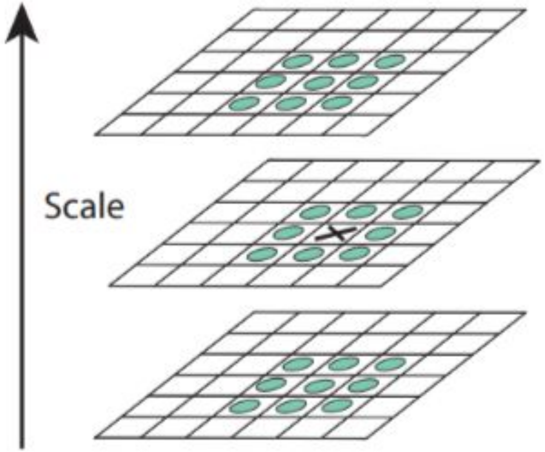
Но это 2D случай: **f(x) = y**

Наша функция имеет вид **f(x, y) = z** (яркость в каждом пикселе)



2D	3D
f'	∇f
f''	матрица Гессе

2D	3D
f'	∇f
f''	матрица Гессе



2D

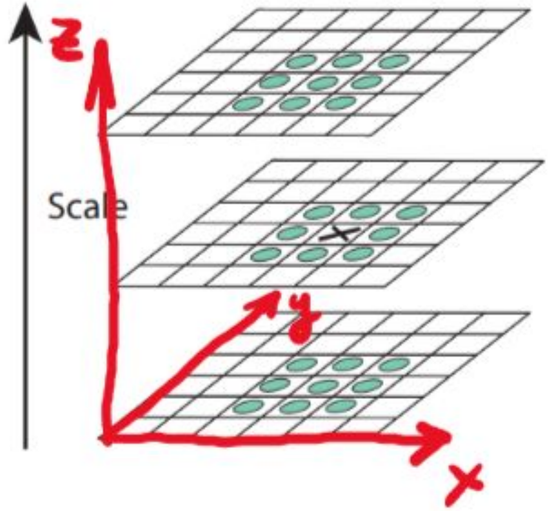
3D

f'

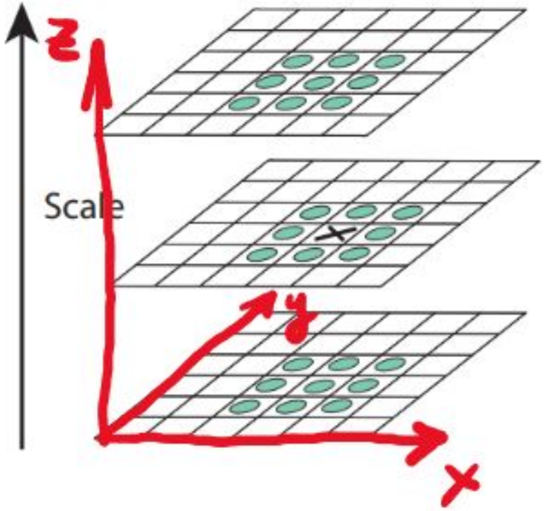
∇f

f''

матрица Гессе



2D	3D
f'	$\nabla f \rightarrow \left[\; ? \; \right]$
f''	матрица Гессе



2D

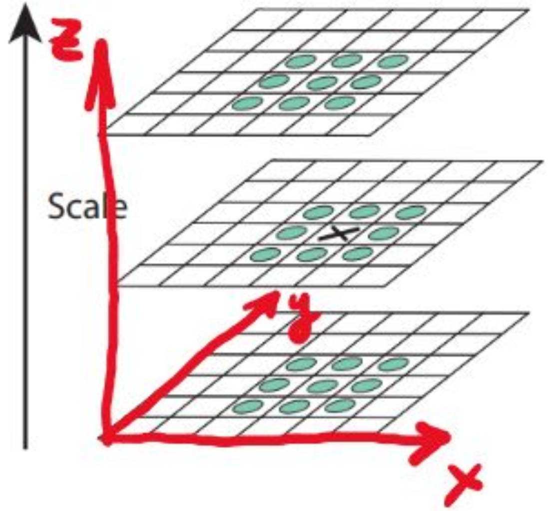
f'

f''

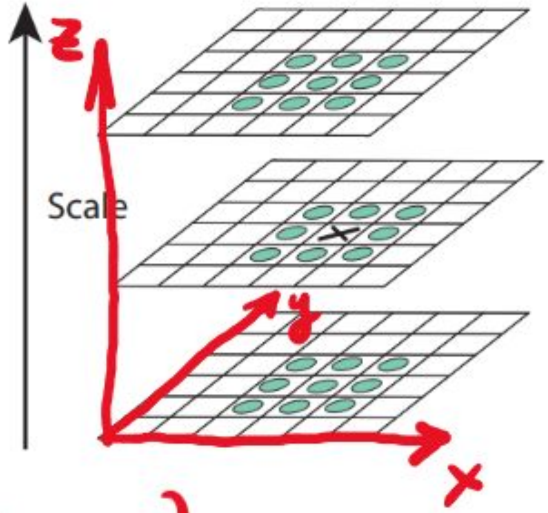
3D

$$\nabla f \rightarrow \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \\ \frac{df}{dz} \end{bmatrix}$$

матрица Гессе



2D	3D
f'	$\nabla f \rightarrow \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \\ \frac{df}{dz} \end{bmatrix}$
f''	матрица Гессе $\rightarrow \left\{ \; ? \; \right\}$



2D

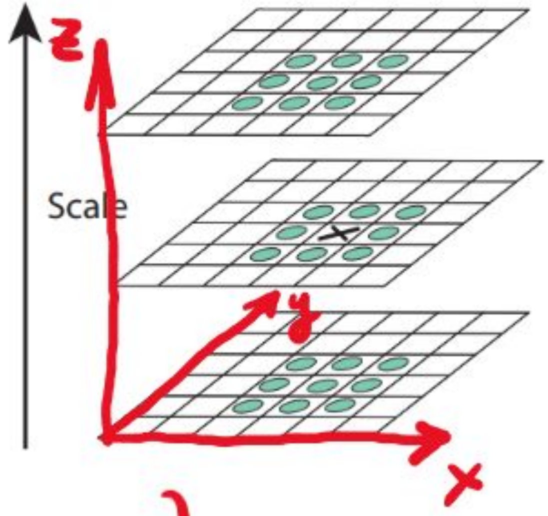
3D

f'

f''

$\nabla f \rightarrow \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \\ \frac{df}{dz} \end{bmatrix}$

матрица Гессе $\rightarrow \left\{ \frac{d^2f}{dx\,dx} \right\}$



2D

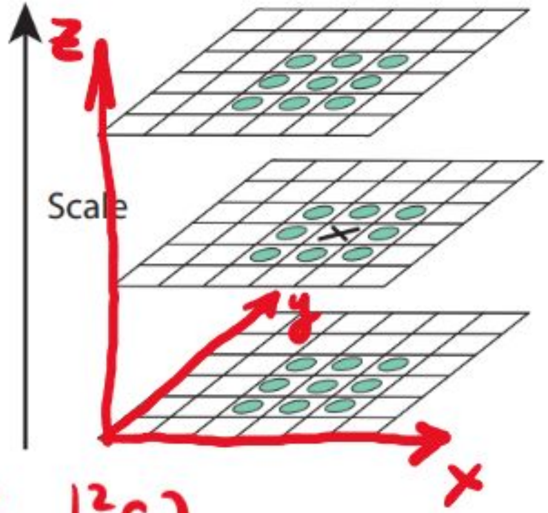
3D

f'

f''

$\nabla f \rightarrow \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \\ \frac{df}{dz} \end{bmatrix}$

матрица Гессе $\rightarrow \begin{Bmatrix} \frac{d^2f}{dxdx} & \frac{d^2f}{dxdy} & \frac{d^2f}{dxdz} \\ \frac{d^2f}{dydx} & \frac{d^2f}{dydy} & \frac{d^2f}{dydz} \\ \frac{d^2f}{dzdx} & \frac{d^2f}{dzdy} & \frac{d^2f}{dzdz} \end{Bmatrix}$



$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$\nabla f(x) \sim 0 + \nabla f(x_0) + \cancel{2} \cdot \frac{1}{\cancel{2}} \left[H f(x_0) \right] dx$$

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$\nabla f(x) \sim 0 + \nabla f(x_0) + \cancel{2} \cdot \frac{1}{\cancel{2}} \left[H f(x_0) \right] dx = 0$$

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$\nabla f(x) \sim 0 + \nabla f(x_0) + \cancel{2} \cdot \cancel{\frac{1}{2}} \left[H f(x_0) \right] dx = 0$$

1) $dx = \left[H f(x_0) \right]^{-1} \left(- \nabla f(x_0) \right)$

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$\nabla f(x) \sim 0 + \nabla f(x_0) + \cancel{2} \cdot \frac{1}{\cancel{2}} \left[H f(x_0) \right] dx = 0$$

1) $dx = \left[H f(x_0) \right]^{-1} \left(-\nabla f(x_0) \right)$

2) $\left[H f(x_0) \right] \cdot dx = -\nabla f(x_0)$

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$\nabla f(x) \sim 0 + \nabla f(x_0) + \cancel{2} \cdot \frac{1}{\cancel{2}} \left[H f(x_0) \right] dx = 0$$

1) $dx = \left[H f(x_0) \right]^{-1} \left(-\nabla f(x_0) \right)$

2) $\left[H f(x_0) \right] \cdot dx = -\nabla f(x_0)$

Решать уравнение (2) численно устойчивее чем обращать и умножать матрицу (1)!

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T \left[H f(x_0) \right] dx$$

$$\nabla f(x) \sim 0 + \nabla f(x_0) + \cancel{2} \cdot \cancel{\frac{1}{2}} \left[H f(x_0) \right] dx = 0$$

$$1)\quad dx = \left[H f(x_0) \right]^{-1} \left(-\nabla f(x_0) \right)$$

$$2)\quad \left[H f(x_0) \right] \cdot dx = -\nabla f(x_0)$$

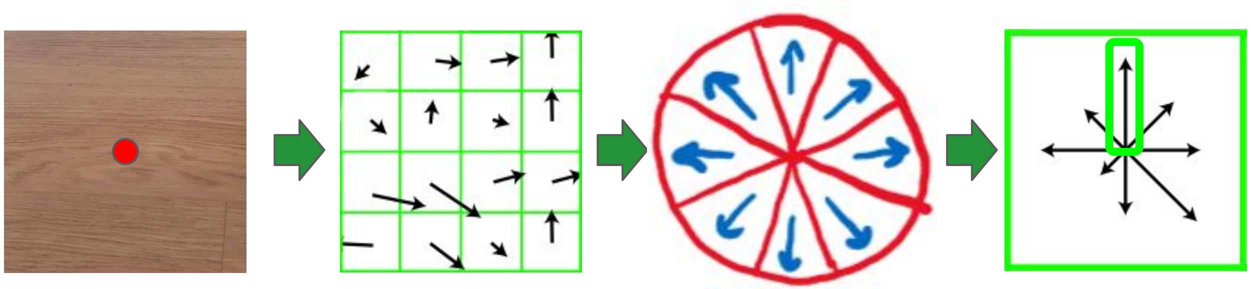
Решать уравнение (2) численно устойчивее чем обращать и умножать матрицу (1)!

Пример как это реализовано в OpenCV:

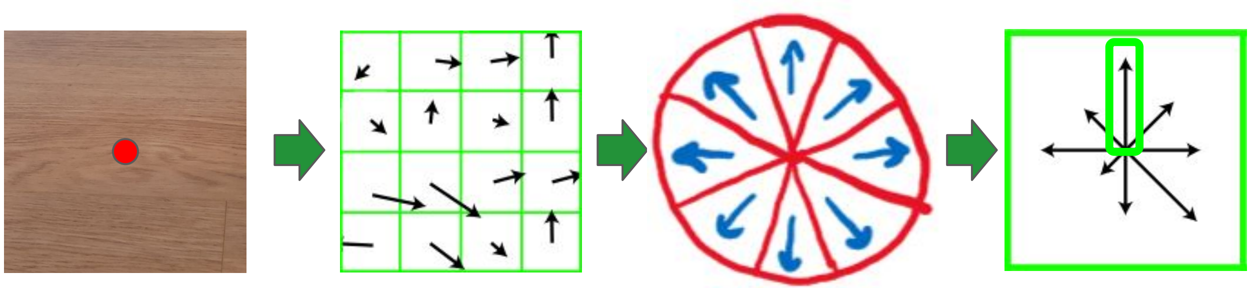
```
Vec3f X = H.solve(dD, DECOMP_LU);
```

opencv/features2d/src/sift.simd.hpp#L334

Вопрос 2: как обнаружать поворот точки точнее?



Вопрос 2: как обнаружать поворот точки точнее?



1) Суммируя вектора которые за нас проголосовали (т.н. **soft-binning**)
2) **Sub-pixel** точность на базе параболы (точнее sub-bin)

Вопрос 3: как ускорить алгоритм детектирования?

Вопрос 3: как ускорить алгоритм детектирования?

Cascade filtering: чем раньше стало ясно что пиксель - не ключевая точка, тем раньше пропускаем его.

Вопрос 3: как ускорить алгоритм детектирования?

Cascade filtering: чем раньше стало ясно что пиксель - не ключевая точка, тем раньше пропускаем его.

Значит **чем быстрее** работает этап фильтрации - **тем раньше** он идет.

Значит **чем медленнее** работает этап фильтрации - **тем позже** он идет.

Вопрос 4: что в статье делается по-другому?

Вопрос 4: что в статье делается по-другому?

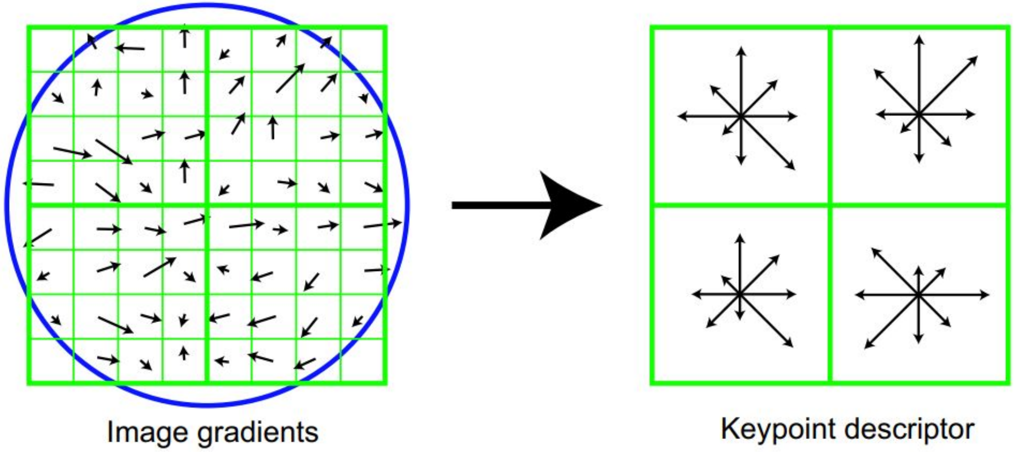


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

Вопрос 4: что в статье делается по-другому?

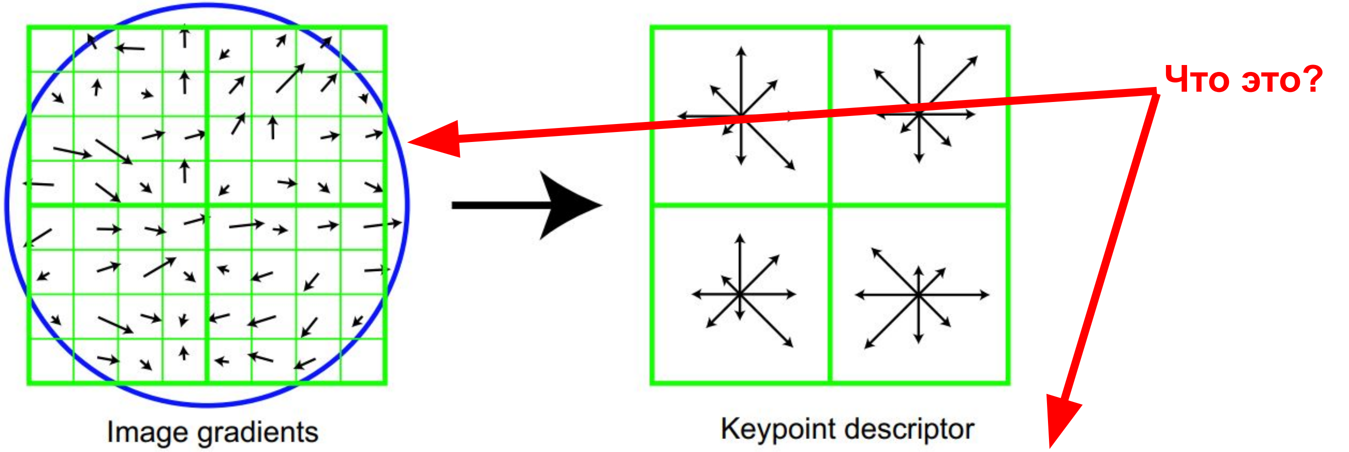


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

Вопрос 4: что в статье делается по-другому?

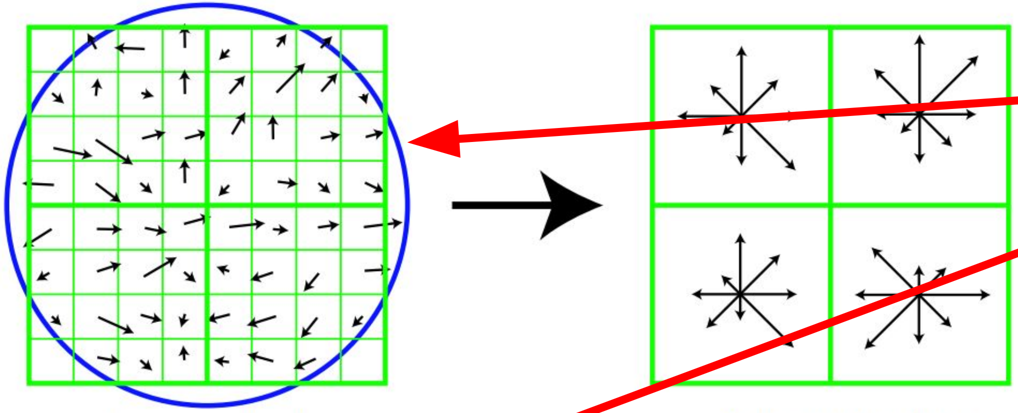


Image gradients Keypoint descriptor

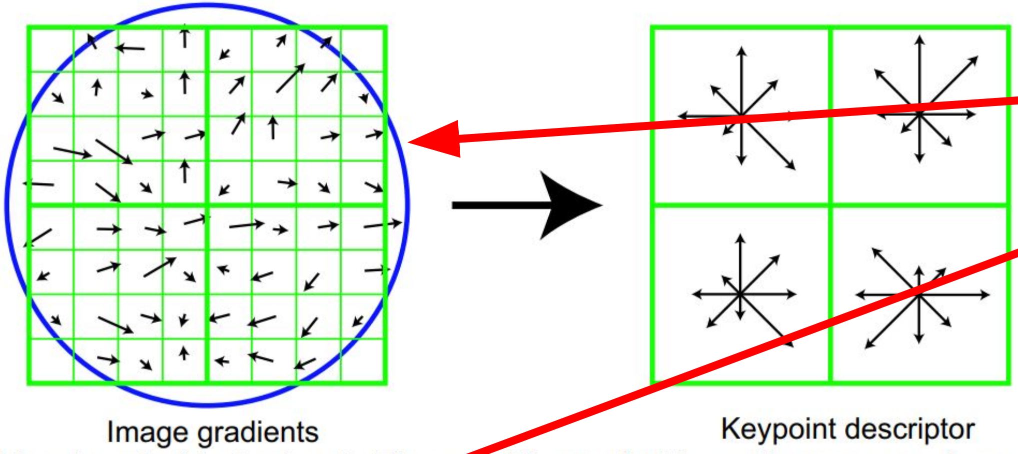
Давайте найдем где в коде эта Figure 7 упоминается:

pyramid as described in Section 5. These are illustrated with small arrows at each sample location on the left side of Figure 7.

A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. This is illustrated with a circular window on the left side of Figure 7, although, of course, the weight falls off smoothly. The purpose of this Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far from the center of the descriptor, as these are most affected by misregistration errors.

The keypoint descriptor is shown on the right side of Figure 7. It allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The

Вопрос 6: есть ли проблемы из-за квадратности окрестности при построении дескриптора? Как исправить?



Давайте найдем где в коде эта Figure 7 упоминается:

pyramid as described in Section 5. These are illustrated with small arrows at each sample location on the left side of Figure 7.

A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. This is illustrated with a circular window on the left side of Figure 7, although, of course, the weight falls off smoothly. The purpose of this Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far from the center of the descriptor, as these are most affected by misregistration errors.

The keypoint descriptor is shown on the right side of Figure 7. It allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The

Вопрос 5: на что влияет размер патча?

Вопрос 5: на что влияет размер патча?

Данные предоставил Stéphane Prodent

Вопрос 7: какой этап занимает больше всего времени?

1) Строим Gaussian пирамиду и **DoG** пирамиду
2) Проверяем является ли пиксель локальным экстремумом
3) Проверяем значение (**response**) пикселя в **DoG** больше ли оно чем некоторый порог
4) Суб-пиксельно уточняем положение пикселя
5) Определяем поворот ключевой точки
6) Определяем масштаб ключевой точки
7) Для каждой точки строим дескриптор из **HoG** (гистограммы градиентов) по квадратной окрестности точки (повернутой и отмасштабированной)

Вопрос 7: какой этап занимает больше всего времени?

1) Строим Gaussian пирамиду и **DoG** пирамиду
   [можно ли уменьшить общее число пикселей пирамиды?]
   [ради скорости и экономии памяти]
2) Проверяем является ли пиксель локальным экстремумом
3) Проверяем значение (**response**) пикселя в **DoG**
   больше ли оно чем некоторый порог
4) Суб-пиксельно уточняем положение пикселя
5) Определяем поворот ключевой точки
6) Определяем масштаб ключевой точки
7) Для каждой точки строим дескриптор из **HoG** (гистограммы градиентов)
   по квадратной окрестности точки (повернутой и отмасштабированной)

Вопрос 7: какой этап занимает больше всего времени?

1) Строим Gaussian пирамиду и **DoG** пирамиду
2) Проверяем является ли пиксель локальным экстремумом
3) Проверяем значение (**response**) пикселя в **DoG** больше ли оно чем некоторый порог **[можно ли/стоит ли сделать раньше?]**
4) Суб-пиксельно уточняем положение пикселя
5) Определяем поворот ключевой точки
6) Определяем масштаб ключевой точки
7) Для каждой точки строим дескриптор из **HoG** (гистограммы градиентов) по квадратной окрестности точки (повернутой и отмасштабированной)

Вопрос 7: какой этап занимает больше всего времени?

1) Строим Gaussian пирамиду и **DoG** пирамиду
2) Проверяем является ли пиксель локальным экстремумом
3) Проверяем значение (**response**) пикселя в **DoG** больше ли оно чем некоторый порог **[можно ли/стоит ли сделать раньше?]**
4) Суб-пиксельно уточняем положение пикселя
5) Определяем поворот ключевой точки
6) Определяем масштаб ключевой точки
7) Для каждой точки строим дескриптор из **HoG** (гистограммы градиентов) по квадратной окрестности точки (повернутой и отмасштабированной)
8) Для каждой точки **A** (и ее дескриптора) первой картинки ищем точку **B** на второй картинке, у которой дескриптор самый похожий на **A**.

Вопрос 7: какой этап занимает больше всего времени?

1) Строим Gaussian пирамиду и **DoG** пирамиду
2) Проверяем является ли пиксель локальным экстремумом
3) Проверяем значение (**response**) пикселя в **DoG** больше ли оно чем некоторый порог
   [можно ли/стоит ли сделать раньше?]
4) Суб-пиксельно уточняем положение пикселя
5) Определяем поворот ключевой точки
6) Определяем масштаб ключевой точки
7) Для каждой точки строим дескриптор из **HoG** (гистограммы градиентов) по квадратной окрестности точки (повернутой и отмасштабированной)
8) Для каждой точки **A** (и ее дескриптора) первой картинки ищем точку **B** на второй картинке, у которой дескриптор самый похожий на **A**.

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128} (A_i - B_i)^2}$$

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes **MLDB Descriptor:** 512 Bits = 64 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128} (A_i - B_i)^2}$$

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Какая метрика разницы дескрипторов?

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) = ???**

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) =** число битов-единиц

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128} (A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) =** число битов-единиц

Как тогда посчитать расстояние Хэмминга?

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) =** число битов-единиц

MLDB = Modified-Local Difference Binary

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128} (A_i - B_i)^2}$$

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) =** число битов-единиц

MLDB = Modified-Local Difference Binary

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

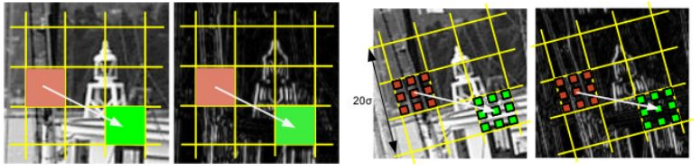
Разбиваем патч на четыре части, считаем среднюю яркость в каждой. Сравниваем попарно - получили первые 6 бит дескриптора. И т.д..

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) =** число битов-единиц

MLDB = Modified-Local Difference Binary



(a) LDB binary test (b) M-LDB binary test

Figure 1: LDB [20] and proposed M-LDB binary tests between grid divisions around a keypoint, shown for the intensity and the gradients in x. M-LDB includes rotation and sub-sampling that depends on the scale.

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Бинарный дескриптор: **MLDB**

SIFT Descriptor: 128 Float = 512 bytes

$$dist(A, B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2}$$

Разбиваем патч на четыре части, считаем среднюю яркость в каждой. Сравниваем попарно - получили первые 6 бит дескриптора. И т.д.

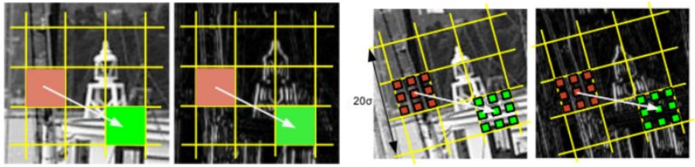
А как выбрать число бит?
Можно ли его вообще поменять?

MLDB Descriptor: 512 Bits = 64 bytes

Расстояние Хэмминга: число разных бит

popcount32(4 bytes**) =** число битов-единиц

MLDB = Modified-Local Difference Binary



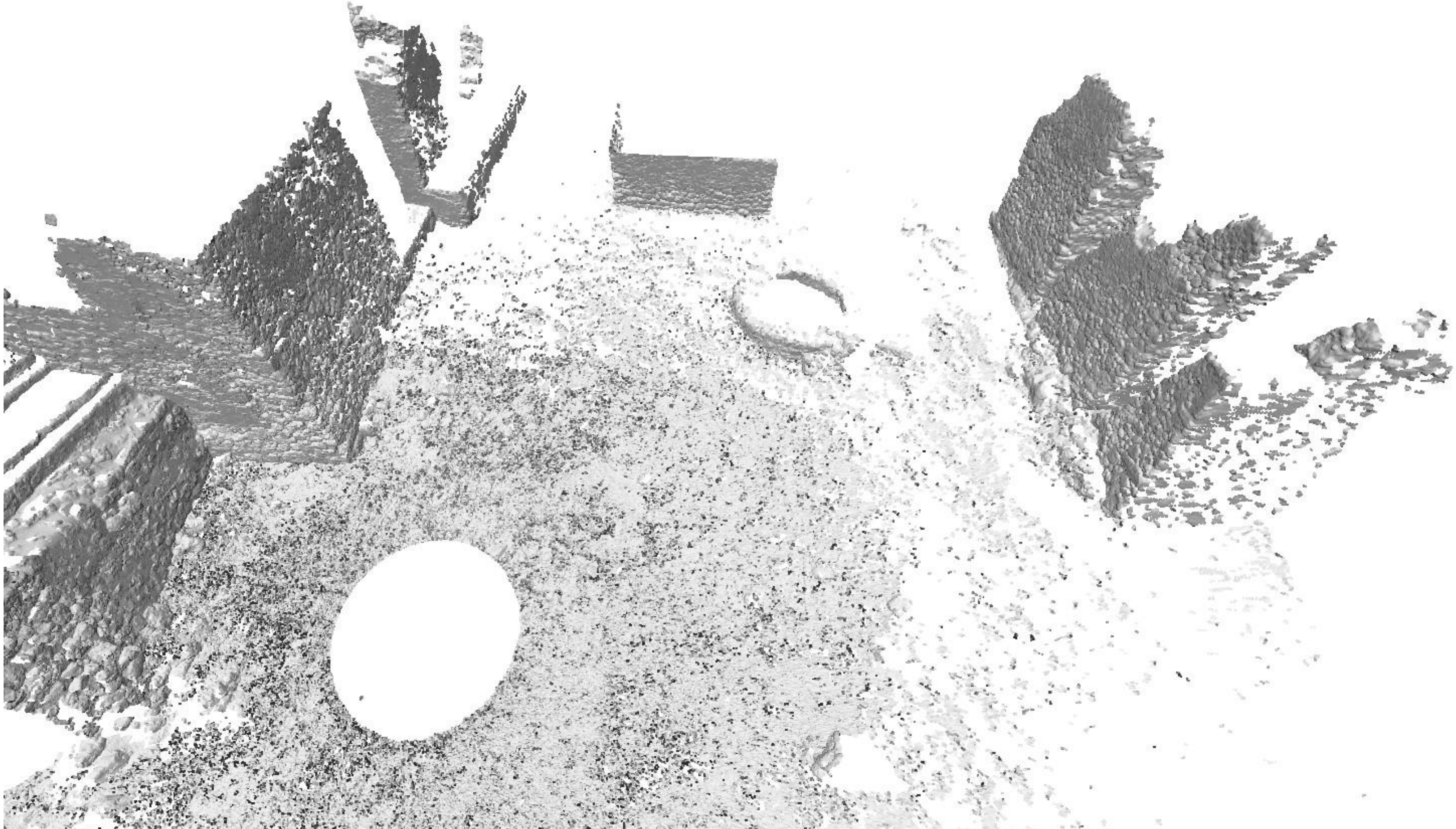
(a) LDB binary test (b) M-LDB binary test

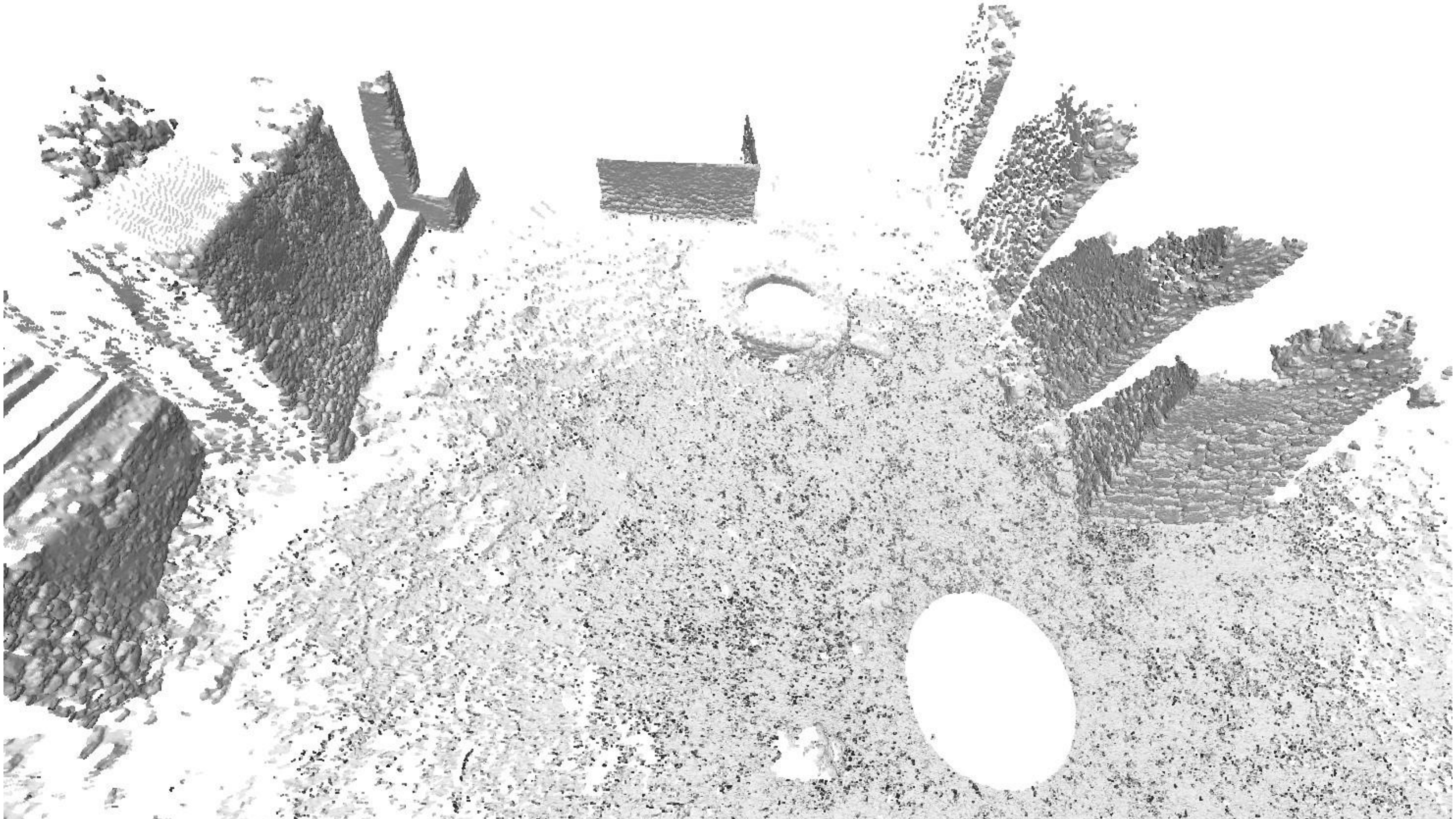
Figure 1: LDB [20] and proposed M-LDB binary tests between grid divisions around a keypoint, shown for the intensity and the gradients in x. M-LDB includes rotation and sub-sampling that depends on the scale.

Статья: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, P. Alcantarilla et al., 2013

Вопрос 8: какой Detector&Descriptor для **3D** облака точек?

B ?





Вопросы?

Полярный Николай
polarnick239@gmail.com

SIFT. Задание

Задание: https://github.com/PhotogrammetryCourse