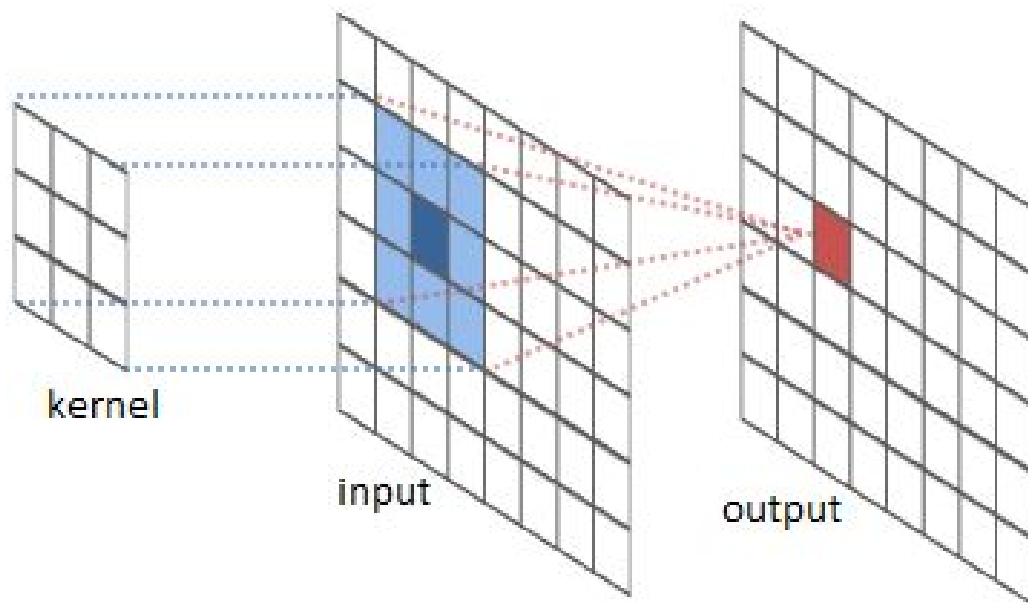


Методы и алгоритмы обработки изображений для робототехники

Лев Пролеев

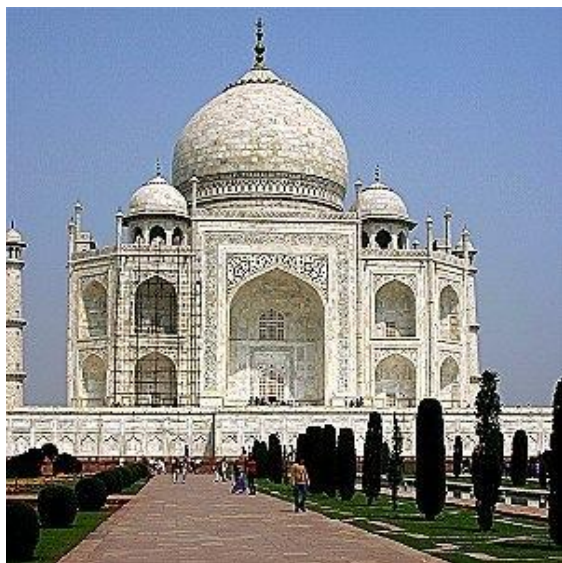
Свёртка



$$B_{ij} = A_{ij} * G_{ij} = \sum_{k=1}^n \sum_{l=1}^m A_{ij} \cdot G_{i-k,j-l}$$

Свёртка

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

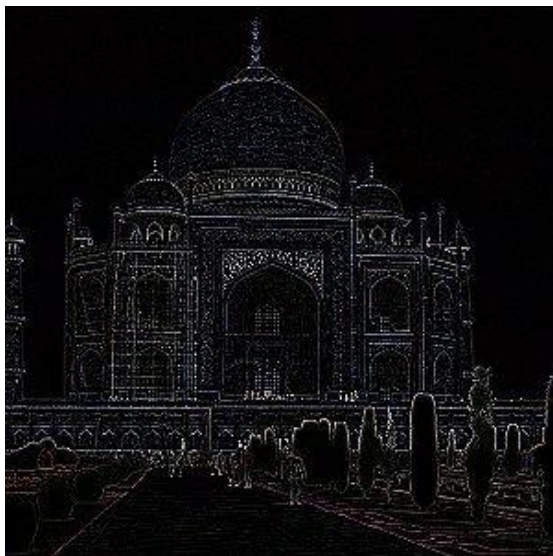


0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0



Свёртка

	0	1	0	
	1	-4	1	
	0	1	0	



	-2	-1	0	
	-1	1	1	
	0	1	2	



Шумы



Оригинал



Гауссовский шум



Импульсивный шум

Усреднение

$$K = \left(\frac{1}{N^2}\right)_{i,j=1}^N$$



Гауссовский фильтр

$$G_{ij} = e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$



Медианный фильтр



Детекторы признаков изображения

Признаки

- углы
- границы объектов
- прямые
- блобы

SIFT (Scale Invariant Feature Transform)

Один из первых детекторов стабильных относительно масштабирования.

Действует в четыре этапа:

- 1) Нахождение локальных экстремумов в пространстве масштаба
- 2) Локализация ключевых точек
- 3) Ориентация ключевых точек
- 4) Составление дескриптора

Масштабируемое пространство (scale space)

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

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



t = 0



t = 1

Масштабируемое пространство (scale space)



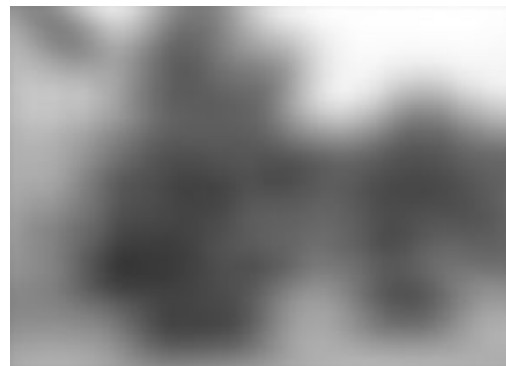
$t = 4$



$t = 16$



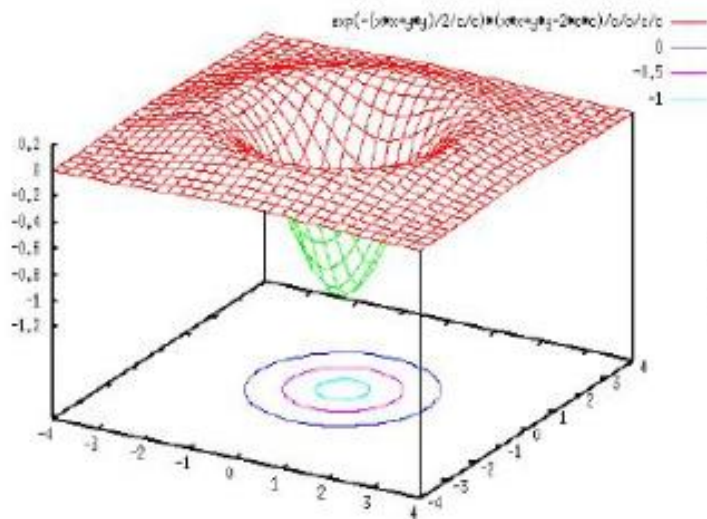
$t = 64$



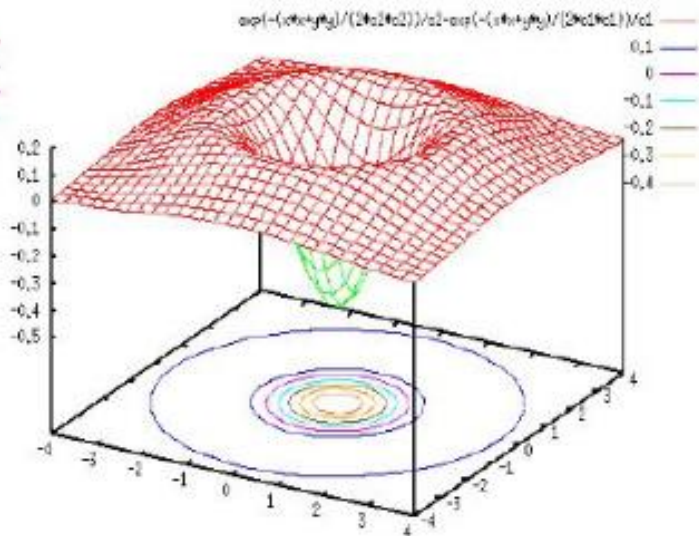
$t = 256$

LoG и DoG

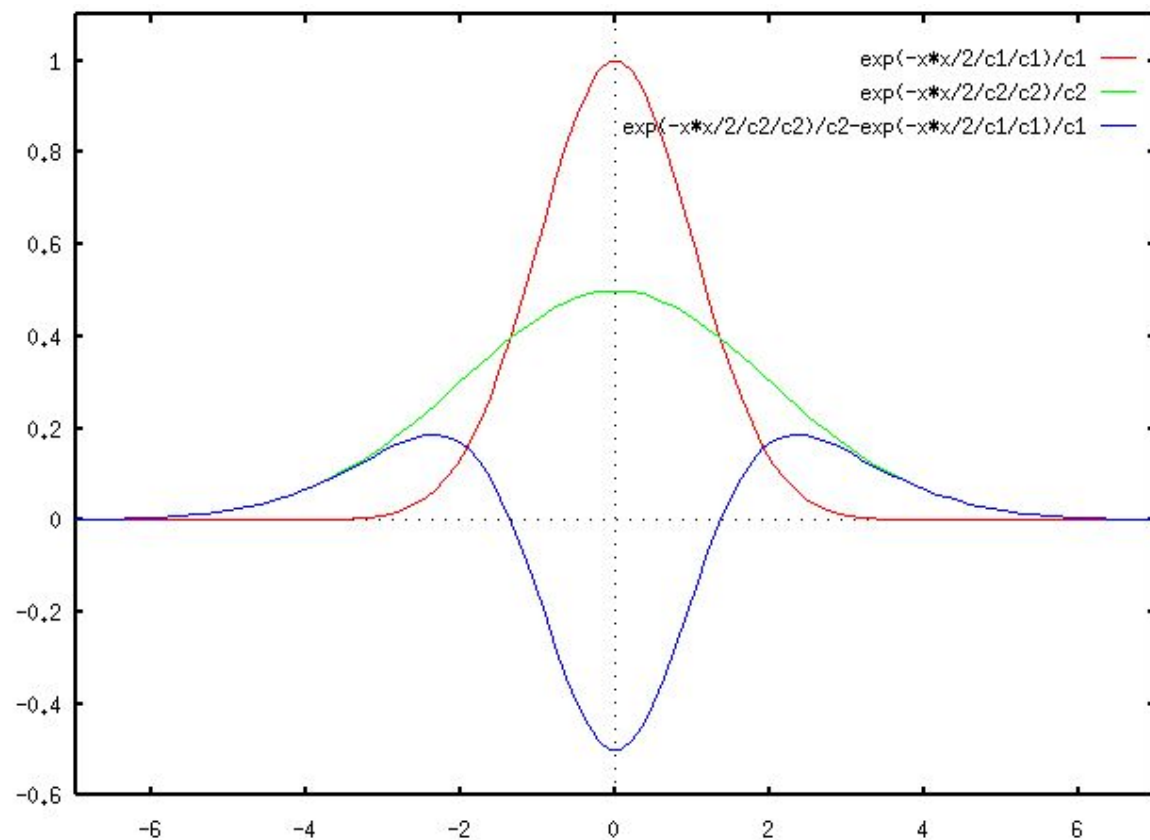
Laplacian of Gaussian



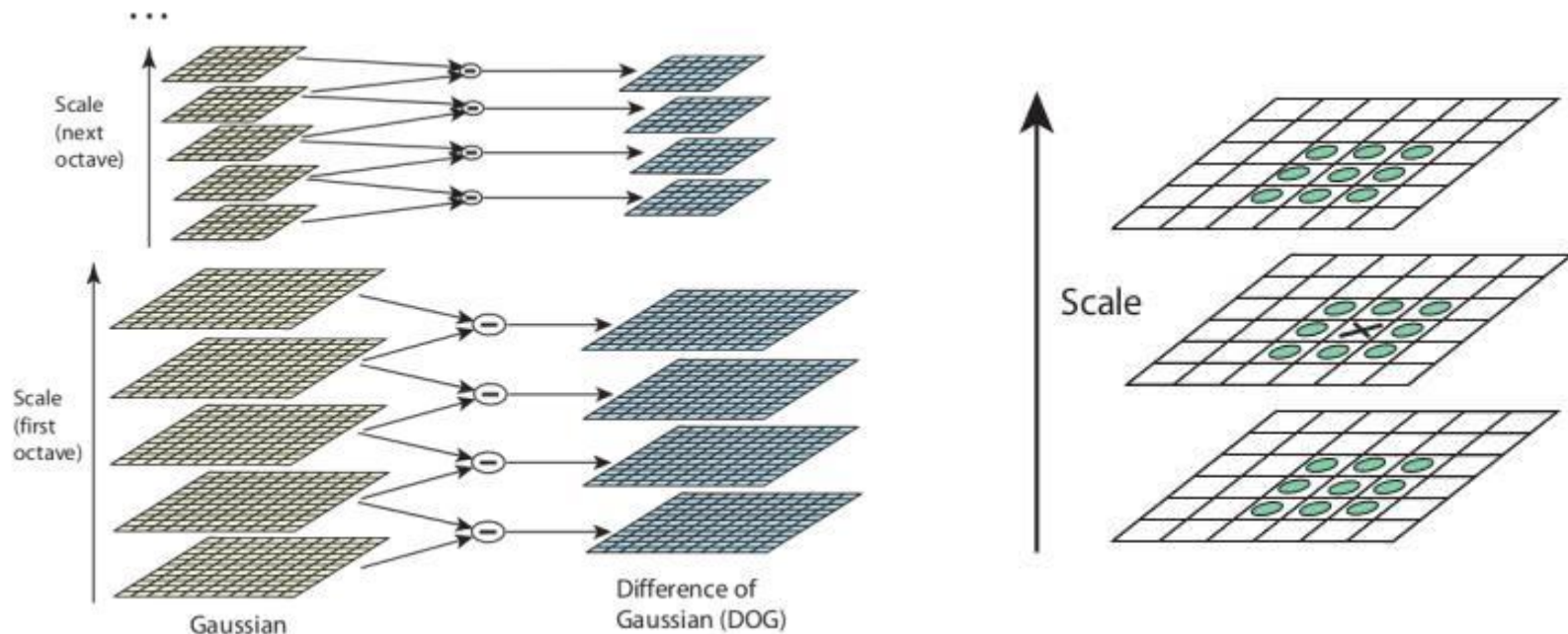
Difference of Gaussians



DoG



Нахождение локальных экстремумов DoG



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

Локализация экстремумов

Локализация экстремумов и отбрасывание
низкоконтрастных точек:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \text{ где } \mathbf{x} = (x, y, \sigma)^T$$

откуда
$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} < 0.5$$

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} < 0.03$$

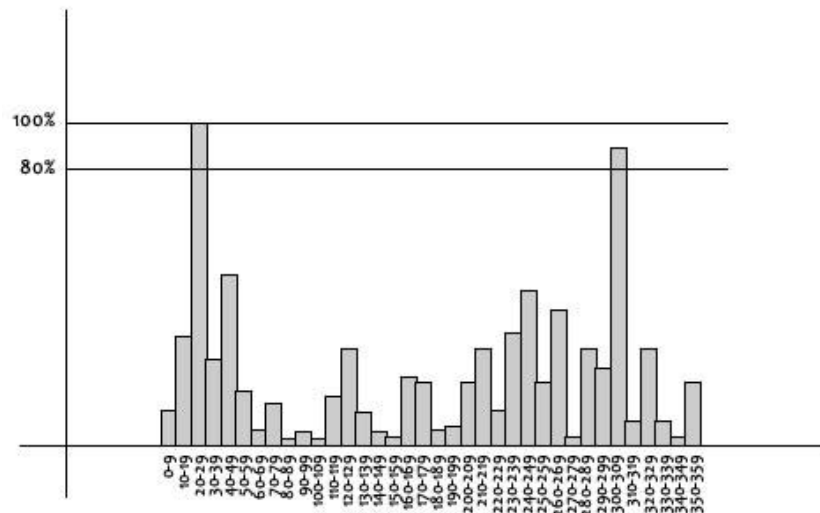
Отбрасывание точек, лежащих на гранях
объектов:

$$\mathbf{H} = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$$

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r}$$

где r -- параметр (у автора 10)

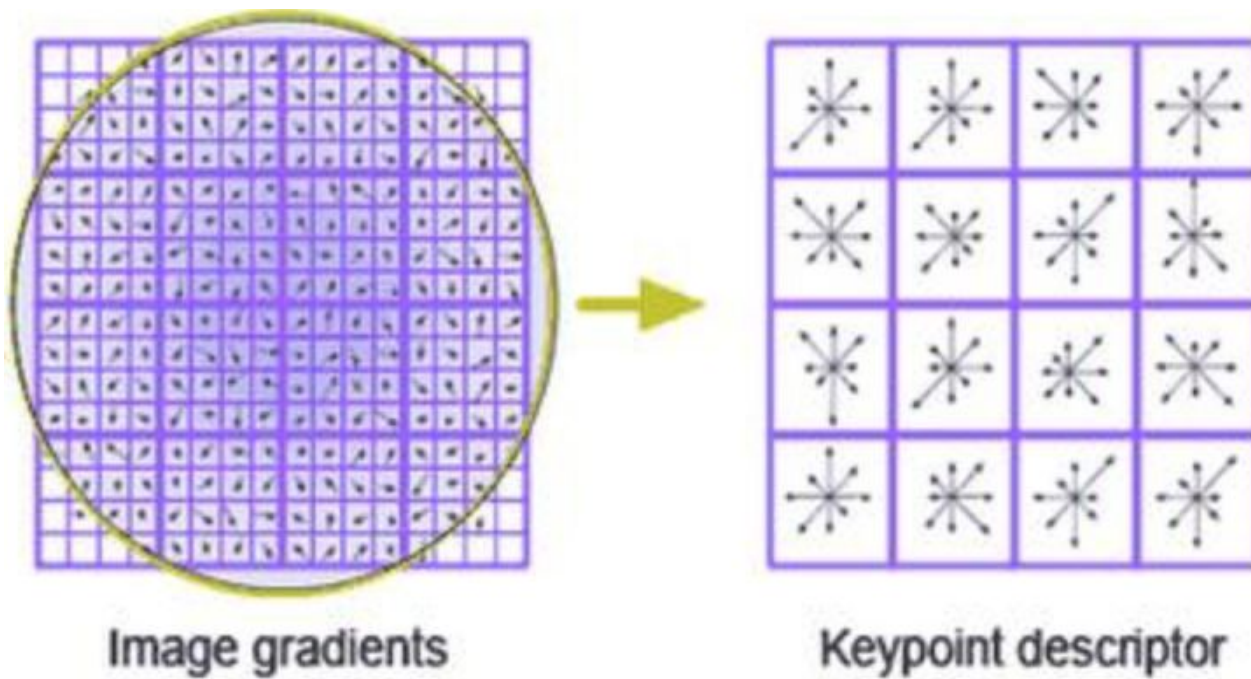
Ориентация ключевых точек



магнитуда: $m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$

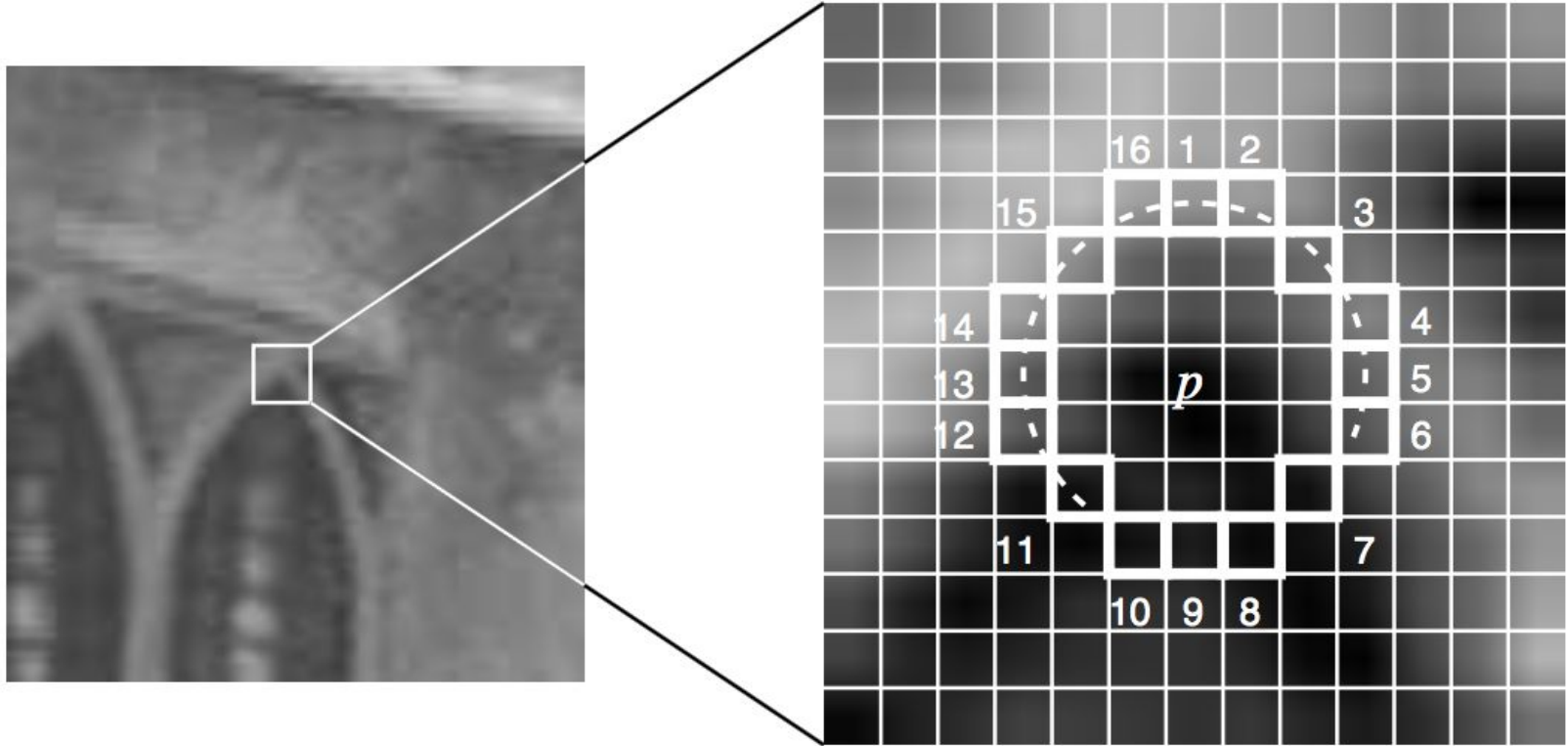
ориентация: $\theta(x, y) = \arctan\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$

Составление дескриптора





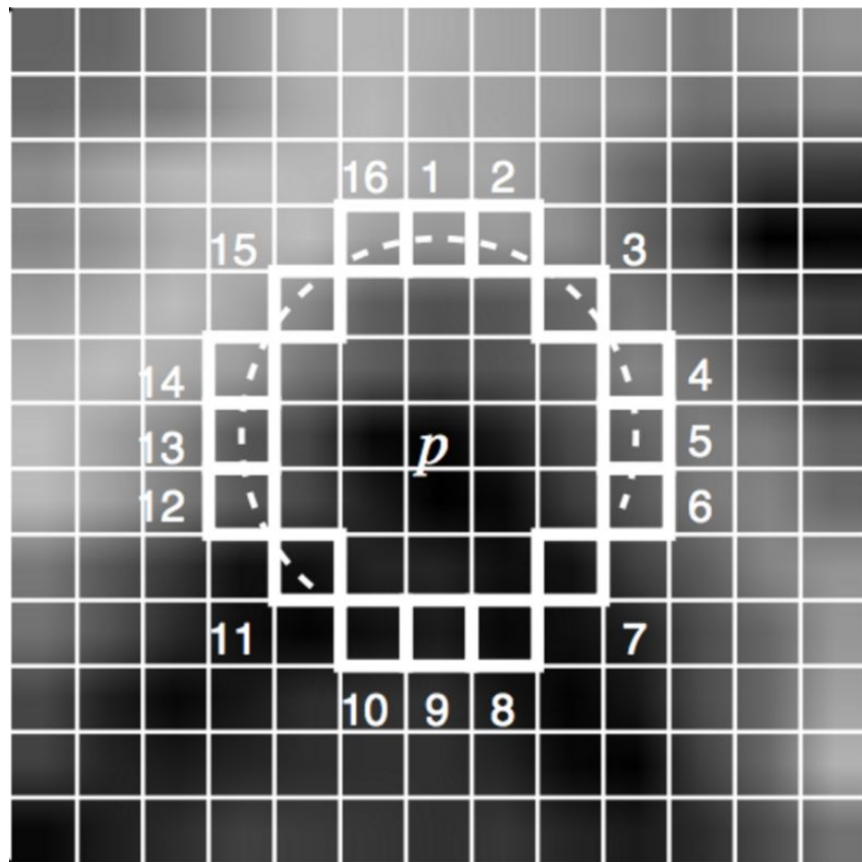
FAST: Features from Accelerated Segment Test



Segment test

Точка p считается углом, если существует n соприкасающихся точек окружности, каждая из которых ярче, чем $(I_p + t)$, либо темнее, чем $(I_p - t)$, где I_p — яркость в точке p .

Заметим, что при $n \geq 12$, можно отбросить много кандидатов быстрее.



Проблемы быстрого segment test

- Быстрый тест не обобщается на случай $n < 12$
- Выбор пикселей для быстрого теста неявно предполагает знание распределения частоты появления особой точки
- Знания из первых 4 проверок не используются в дальнейшем
- Достаточно близкие точки могут помечаться углами вместе

FAST

Каждая точка из окружности может находиться в трёх состояниях:

1. Ярче, чем $I_p + t$
 2. Темнее, чем $I_p - t$
 3. Между этими значениями
- Соберём картинки, похожие на те, в которых нужно будет искать углы.
 - Проведём для каждого пикселя полный segment test.
 - Получим выборку из шестнадцатимерных векторов со значением каждой компоненты из $\{1, 2, 3\}$ и лейблами угол/не угол.
 - Обучим решающее дерево.

FAST: подавление немаксимумов

Выберем точку с максимальным значением функции V :

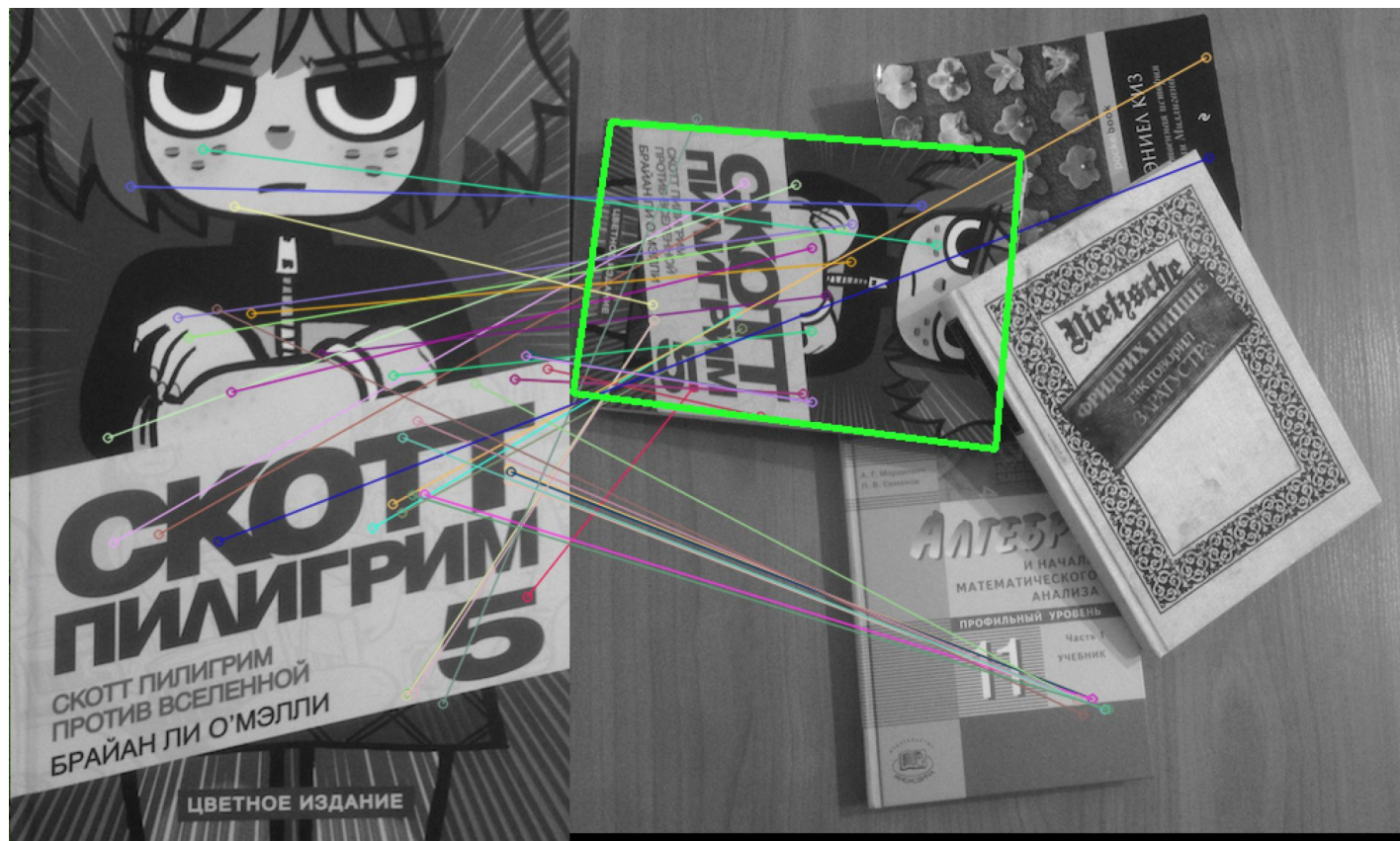
$$V = \max \left(\sum_{x \in S_{bright}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \rightarrow x}| - t \right)$$

где

$$S_{bright} = \{x | I_{p \rightarrow x} \geq I_p + t\}$$

$$S_{dark} = \{x | I_{p \rightarrow x} \leq I_p - t\}$$





Что почитать

- Про признаки в целом
 - http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html#features-meaning
- SIFT
 - http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html#sift-intro
 - <https://habrahabr.ru/post/106302/>
- FAST
 - http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html#fast