

ДЕСКРИПТОРЫ. SIFT

Лекция 7.

Преподаватель: Сибирцева Елена
elsibirtseva@gmail.com

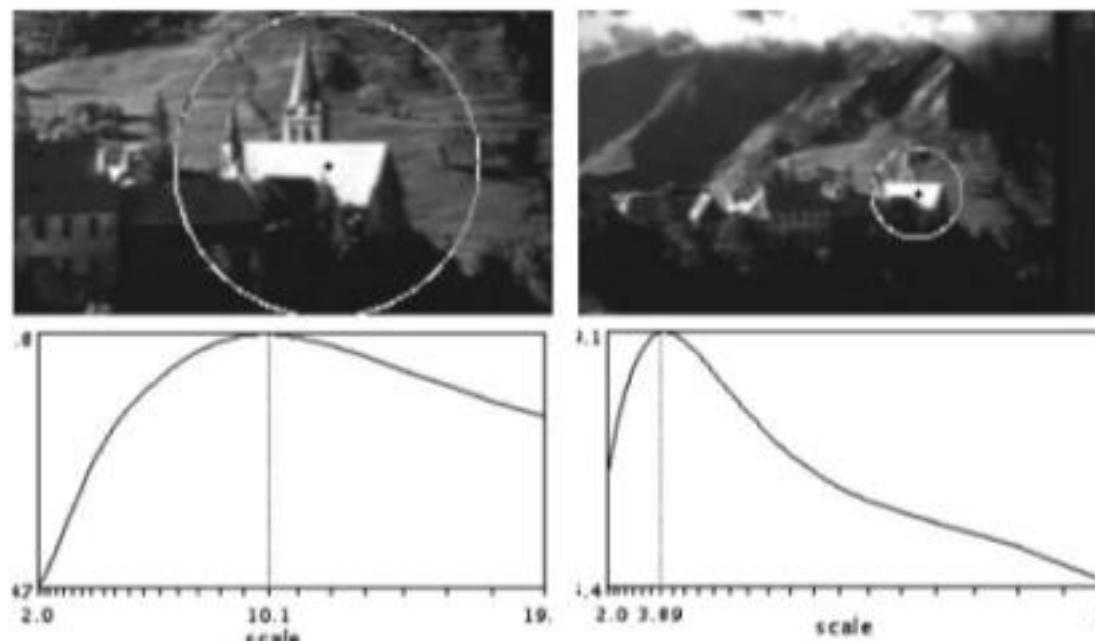
**В следующий раз
отдыхаем.**

Встречаемся 21 февраля

В предыдущих сериях...

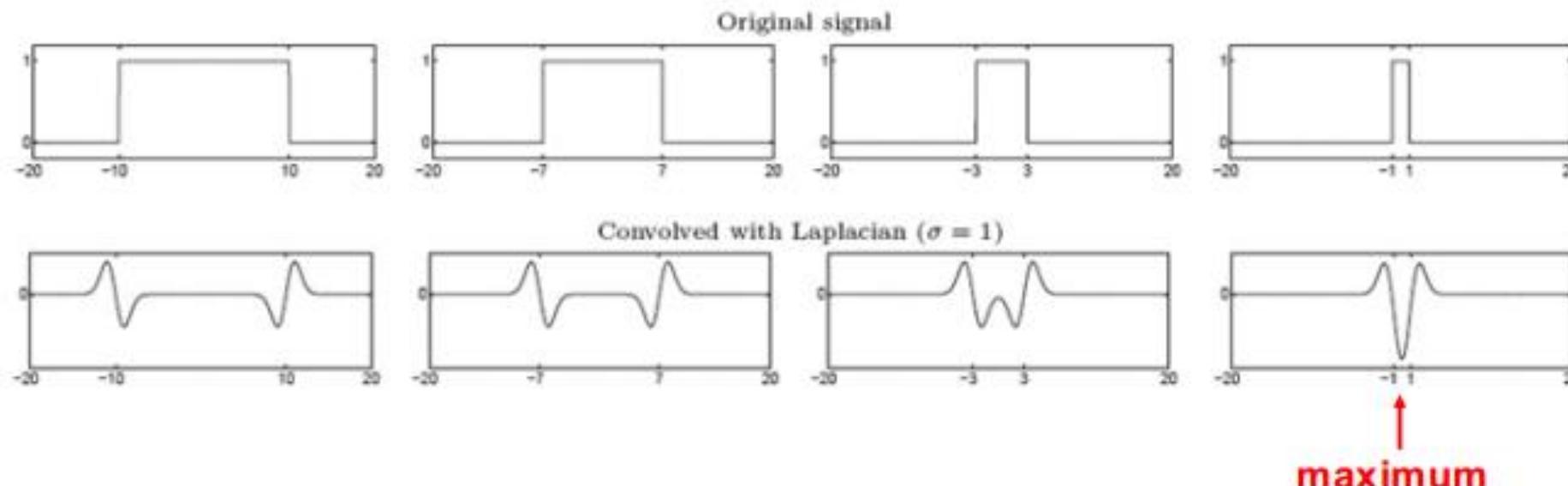
Инвариантность к масштабированию

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



От краев к блобам

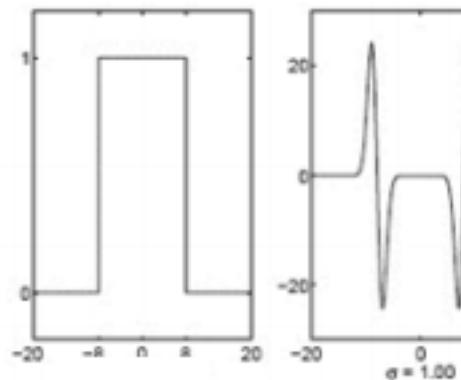
- Край = «всплеск»
- Блоб = совмещение двух «всплесков»



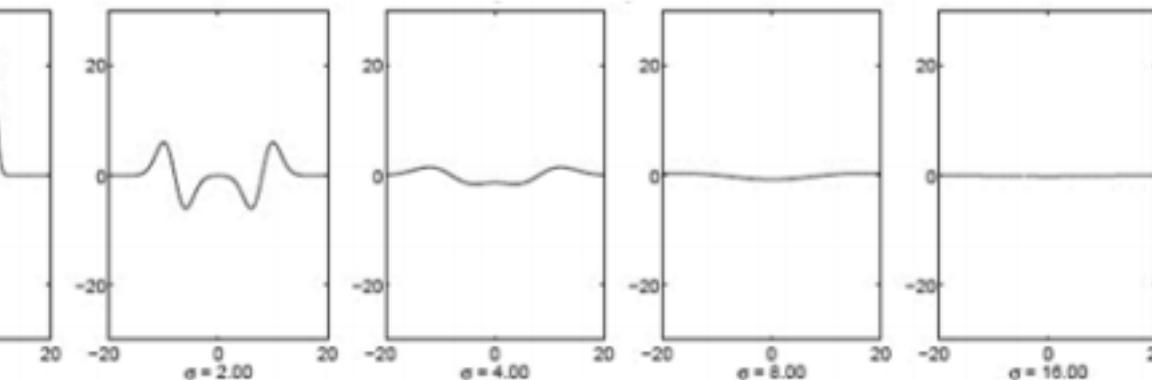
Выбор масштаба: величина отклика лапласиана Гауссиана достигает максимума в центре блоба в том случае, если размер лапласиана «соответствует» размеру блоба

Эффект нормализации

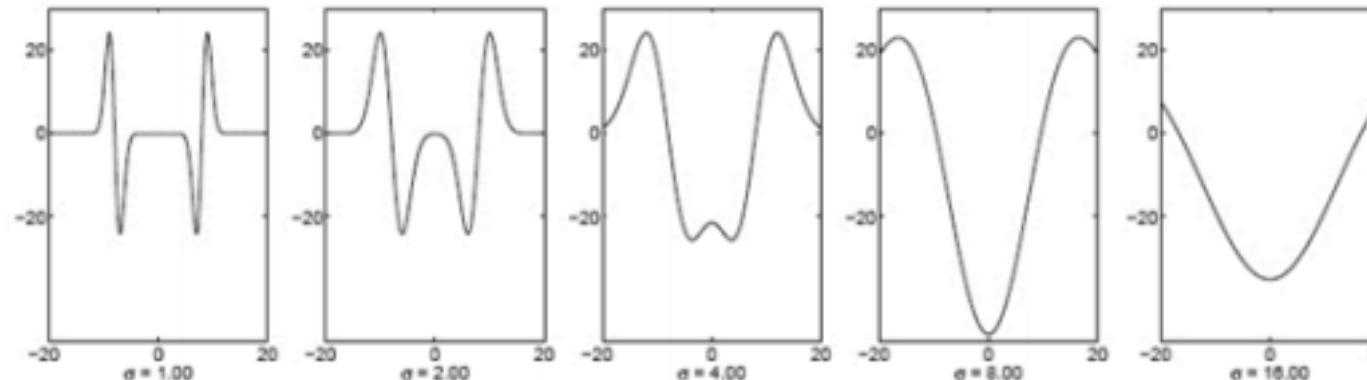
Исходный сигнал



Ненормализованный отклик Лапласиана



Нормализованный по масштабу отклик Лапласиана



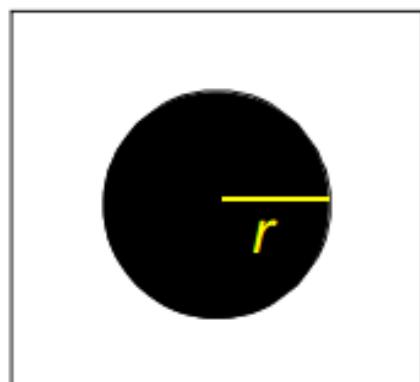
максимум

Выбор масштаба

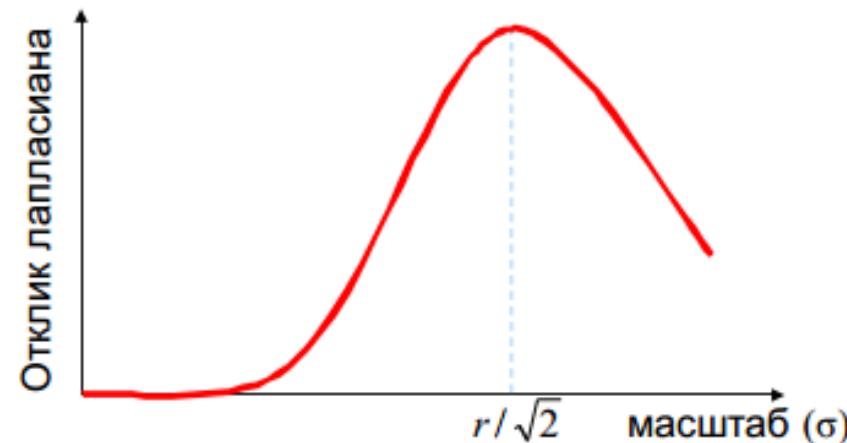
- 2D Лапласиан задается формулой:

$$(x^2 + y^2 - 2\sigma^2) e^{-(x^2+y^2)/2\sigma^2} \quad (\text{с точностью до масштаба})$$

- Для бинарного круга радиуса r , Лапласиан достигает максимума в $\sigma = r / \sqrt{2}$

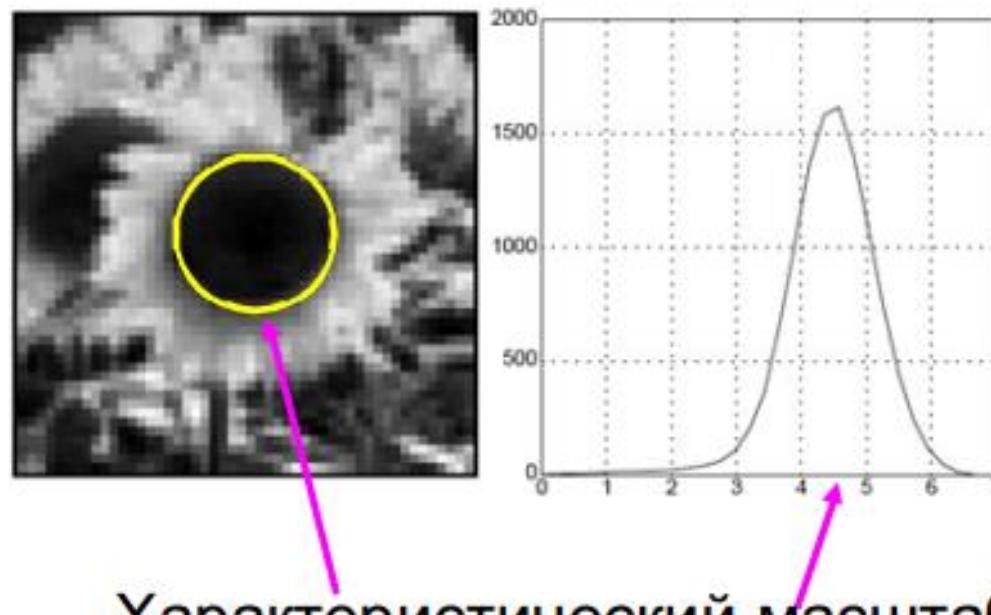


изображение



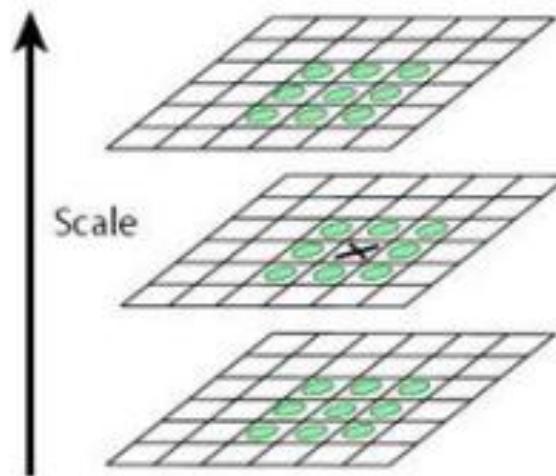
Характеристический размер

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



Многомасштабный детектор блобов

1. Свертываем изображение нормализованным фильтром Лапласианом на разных масштабах
2. Ищем максимум отклика Лапласиана в 3D



Пример



Эффективная реализация (DoG)

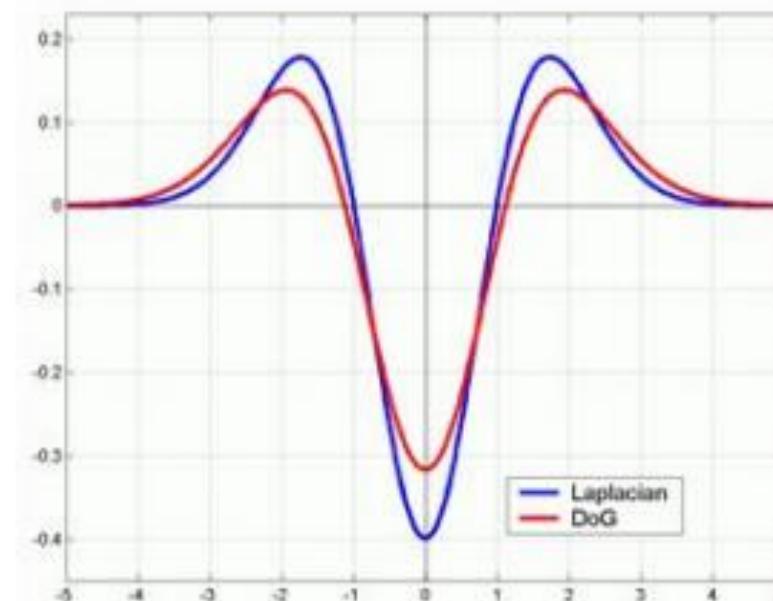
Приближение Лапласиана с помощью разницы гауссиан:

$$L = \sigma^2 (G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

(Лапласиан)

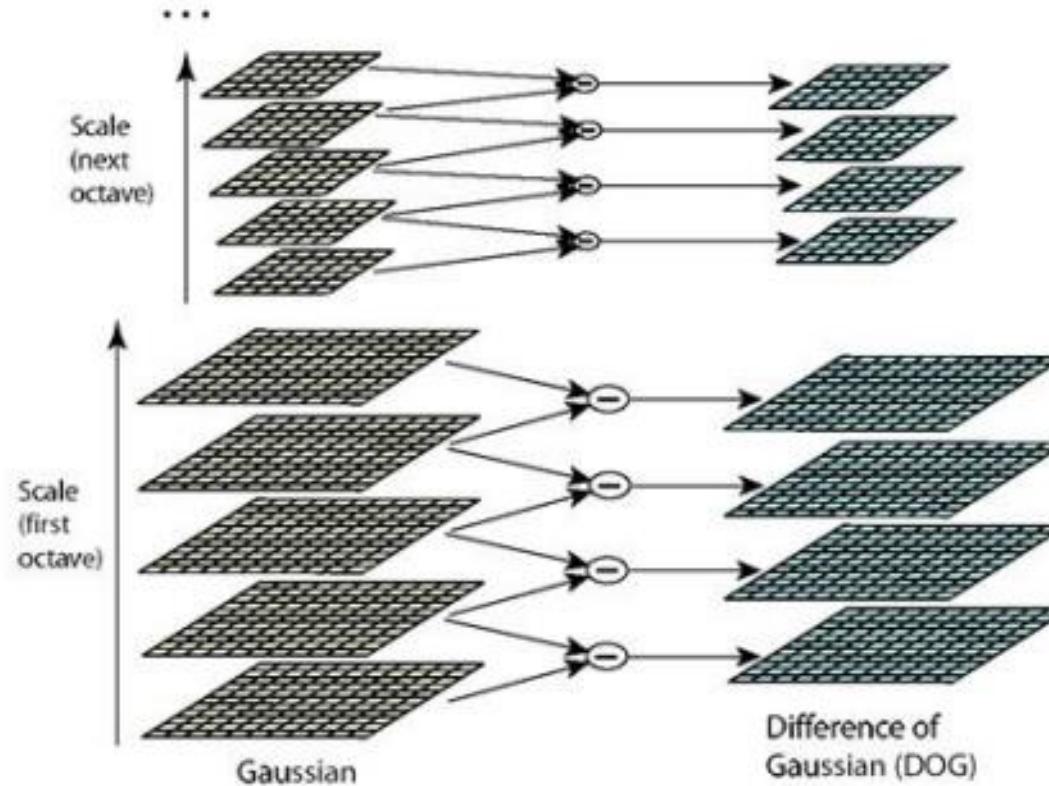
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Разница Гауссиан)



Difference of Gaussian = DoG

Эффективная реализация (DoG)



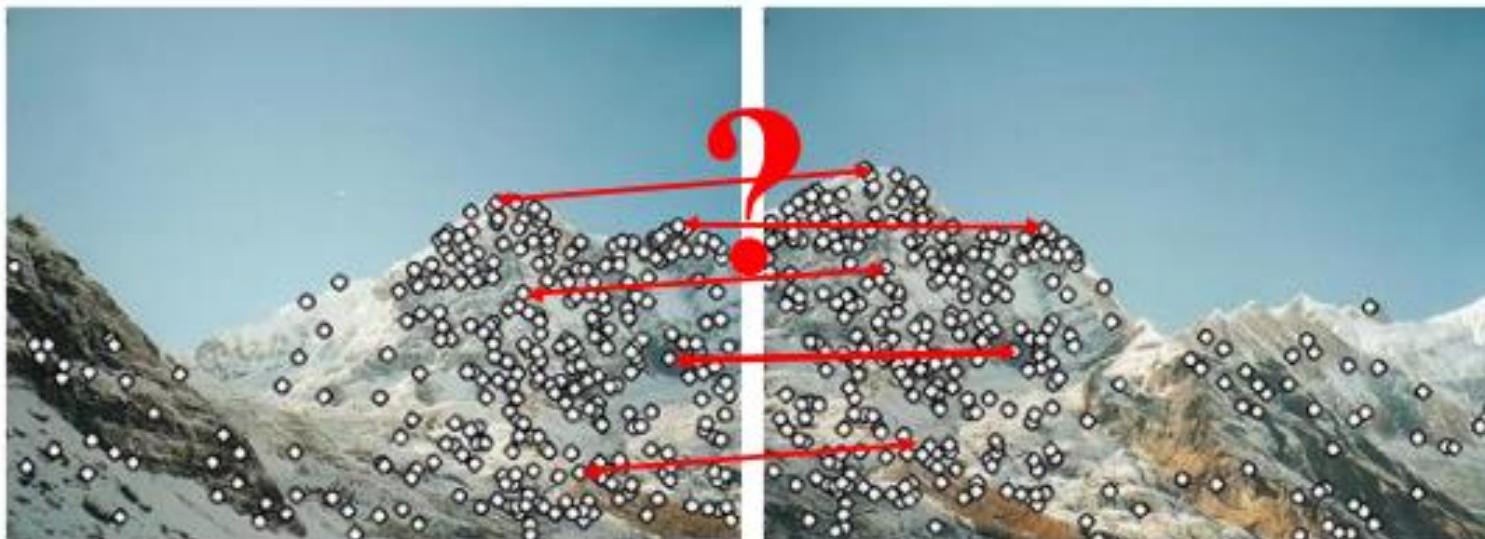
Детектор DoG также выделяет «блобы» на изображении

David G. Lowe. "[Distinctive image features from scale-invariant keypoints.](#)"
IJCV 60 (2), pp. 91-110, 2004.

ДЕСКРИПТОРЫ

Дескрипторы

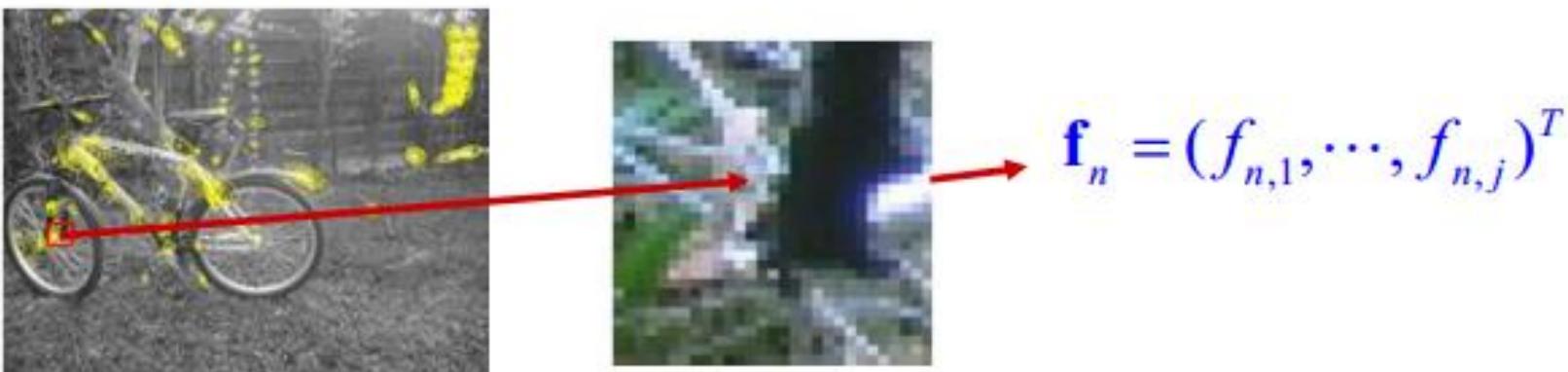
Точки найдены – как их сопоставить?



- Нужно как-то описать каждую точку, чтобы можно было отличать одну от другой!
- Дескриптор (Descriptor) - вектор признак окрестности точки

Дескрипторы

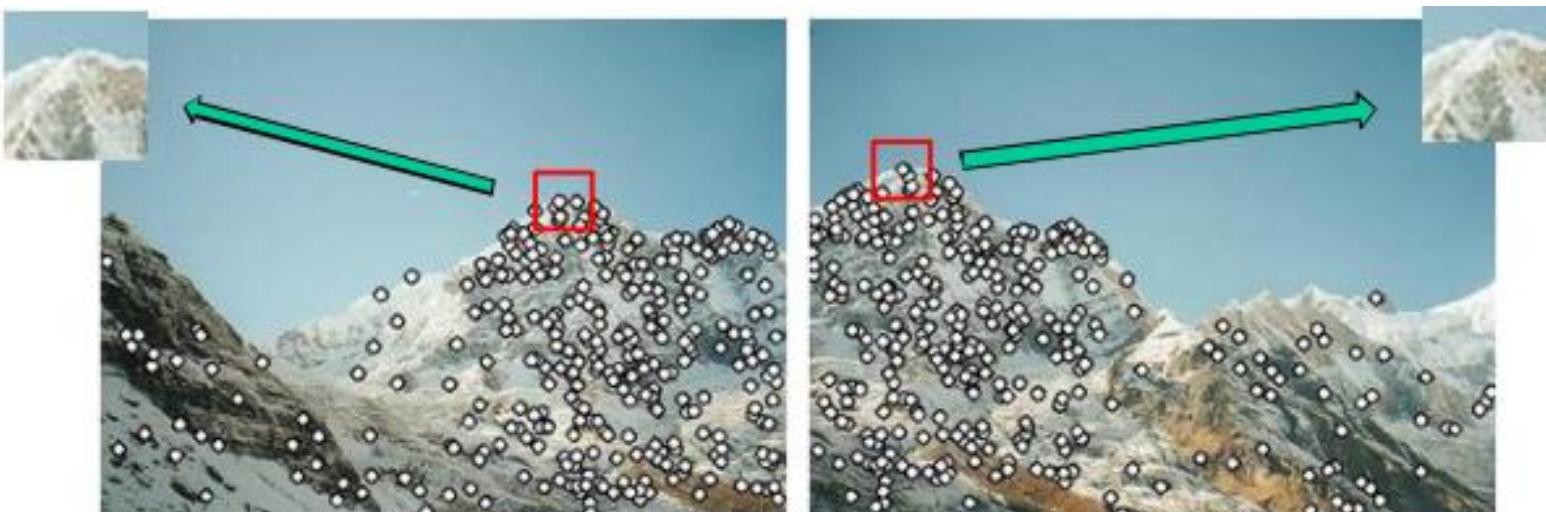
Необходимо каждую интересную точку описать набором параметров:



Как будем поступать:

- Возьмём окрестность точки
 - Какой формы?
 - Какого размера?
- Вычислим по окрестности набор признаков
 - Какие?

Простейший подход

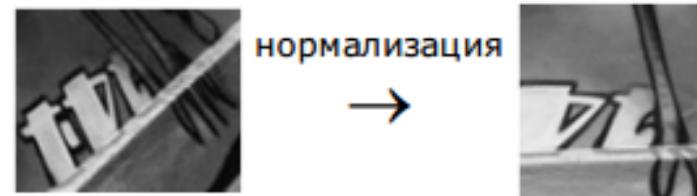


- Возьмём квадратные окрестности, со сторонами, параллельными строкам и столбцам изображения
- Яркости пикселов будут признаками
- Сравнивать будем как два изображения - попиксельно (SAD, SSD)
- Такая окрестность инвариантна только к сдвигу изображения

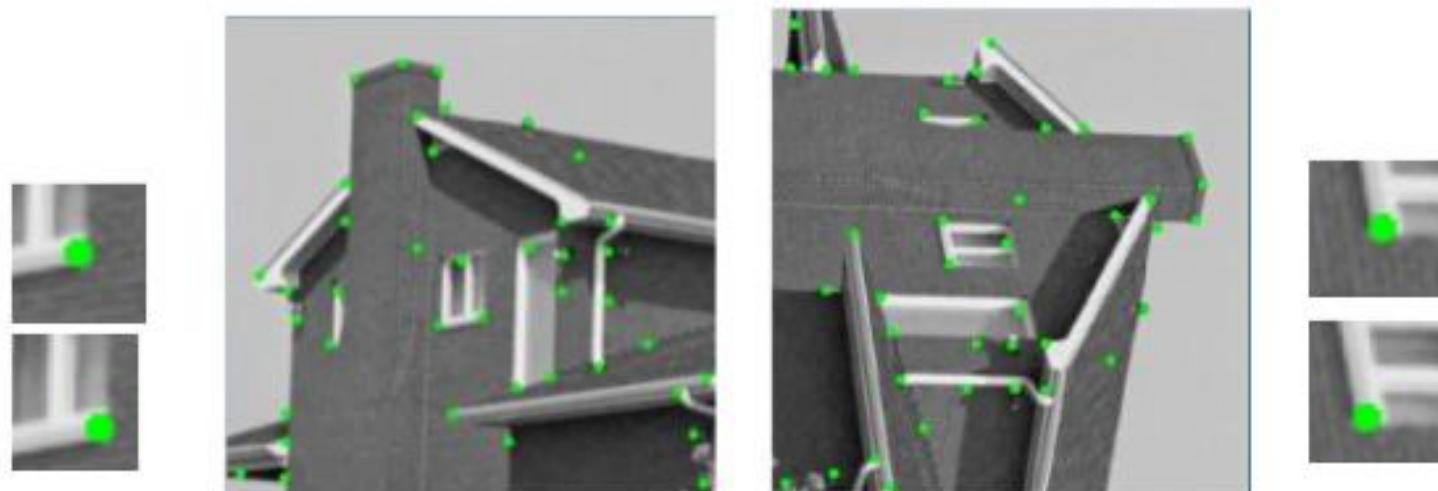
Инвариантность к яркости

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

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



Недостаток простой окрестности



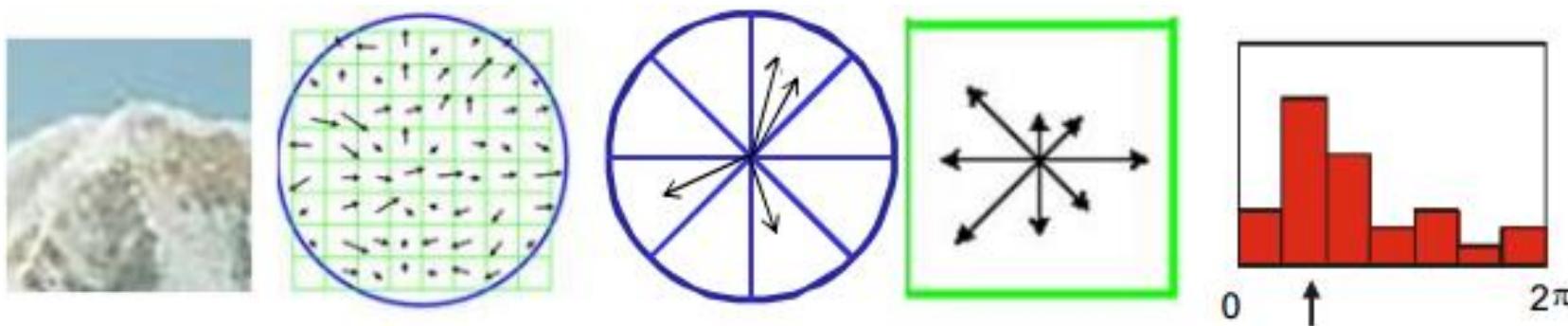
- Детектор точек инвариантен к повороту, а окрестность нет
- Небольшие сдвиги, т.е. ошибки в нахождении точки делают невозможным попиксельное сравнение

Метод SIFT

- Scale-Invariant Feature Transform:
 - Детектор DoG
 - Определение положения и характерного масштаба особенности
 - Ориентация
 - Определение доминантной ориентации особенности по градиентам
 - Дескриптор
 - Использование статистик по направлению градиентам
- Устойчив к изменениям освещенности и небольшим сдвигам

David G. Lowe. "[Distinctive image features from scale-invariant keypoints.](#)" *IJCV* 60 (2), pp. 91-110, 2004.

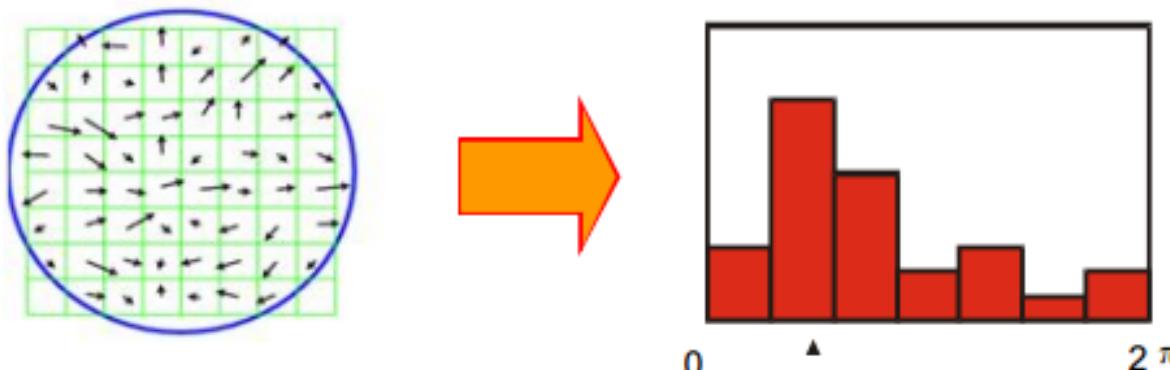
Гистограмма ориентаций градиентов



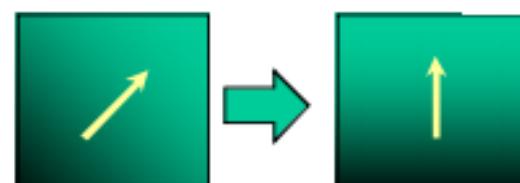
- Основа дескриптора SIFT – подсчёт гистограммы ориентаций градиентов
 - Вычислим направление градиента в каждом пикселе
 - Квантуем ориентации градиентов на 8 ячеек (направлений)
 - Пометим каждый пиксель номером ячейки
 - Посчитаем гистограмму направлений градиентов
 - Для каждой ячейки посчитаем количество пикселов с номером этой ячейки

Ориентация фрагмента

- Идея: найти основное (доминантное) направление градиентов пикселей в окрестности точки
- Выберем в гистограмме ячейку с максимальным значением, возьмём это направление как доминирующее



- Повернем фрагмент так, чтобы доминантное направление градиента было направлен вверх



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

Окрестность особенности

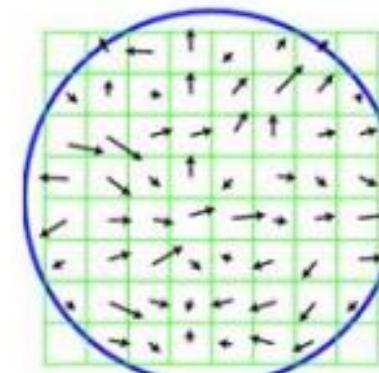
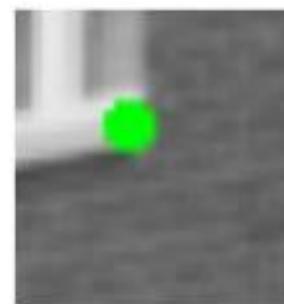
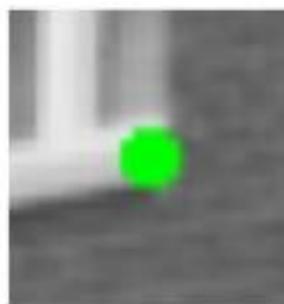


- Для каждой найденной особенности теперь знаем характеристические масштаб и ориентацию
- Выберем соответствующую прямоугольную окрестность
 - (Rotation Invariant Frame)
- Приведем окрестность к стандартному размеру (масштабируем)

Пример локальных особенностей



Устойчивость к сдвигам



- За счёт чего можно дескриптор сделать устойчивым к небольшим сдвигам?
- При расчёте гистограммы будем взвешивать вклад пикселей
- Веса рассчитываем в зависимости от близости к центру, по Гауссине
- Небольшие ошибки в положении, масштабе и ориентации будут приводить к небольшим изменениям дескриптора

Использование цвета

- RGB-SIFT

- 3 дескриптора SIFT для каждого канала

- C-SIFT

- Каналы O_1 и O_2

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}$$

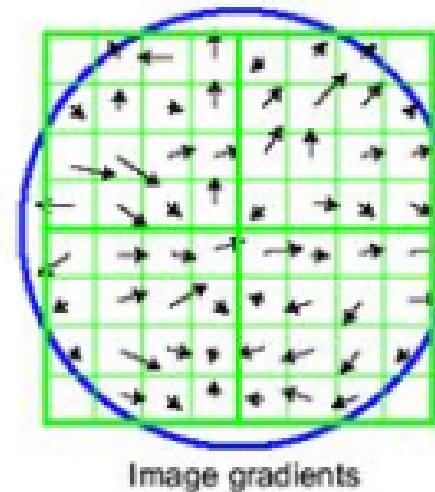
- rgSIFT

- Каналы r и g

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}$$

Резюме SIFT

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



Резюме локальных особенностей

- Локальные характерные особенности - один из основных инструментов для анализа изображений
- Особенности должны быть устойчивы к изменению положения, масштаба, ракурса и освещения изображения
- Мы рассмотрели несколько методов:
 - Детекторы: Harris, LoG, DoG, Harris-Laplace
 - Дескрипторы: цветовая окрестность, SIFT, C-SIFT

The logo consists of the word "SIFT" in a bold, white, sans-serif font. The letter "I" is uniquely designed with a vertical bar on the left and a shorter vertical bar on the right, creating a stepped appearance. The letter "F" has a single vertical bar with a horizontal cutout in the middle. The letter "T" is a standard vertical bar. The entire logo is set against a dark gray rectangular background that is positioned at the bottom of the slide.

SIFT

scale invariant feature transform

SIFT basics

- SIFT = Scale Invariant Feature Transform
- Запатентован в университете UBC (University of British Columbia)
- Похож по принципам работы на естественные визуальные системы человека, обезьян и т.п.
- Преобразует данные с изображения в инвариантные к масштабу координаты



David Lowe, UBC

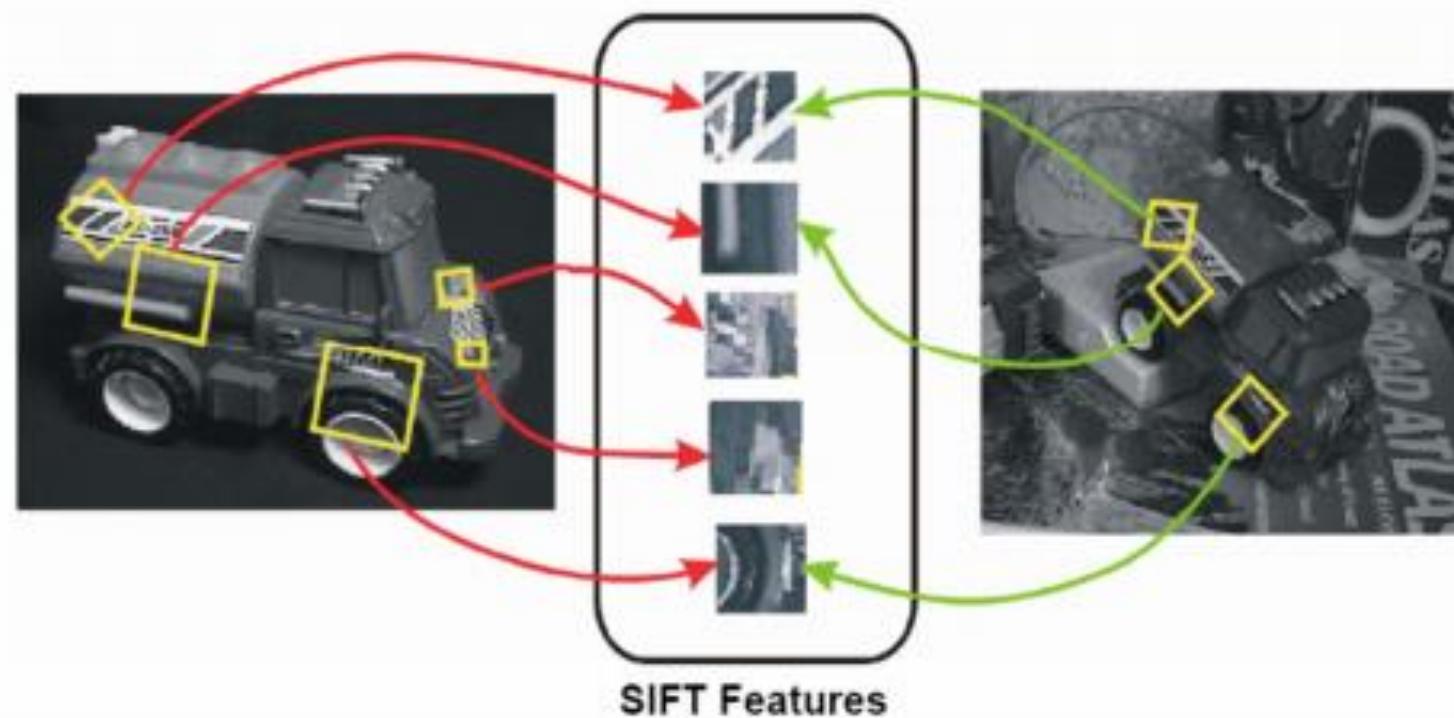
Цель

- Выделить отличимые инвариантные особенности
 - Верно соотнесенные с большим набором особенностей любого изображения
- Инвариантность к масштабу и повороту
- Устойчивость к
 - Аффинным преобразованиям
 - Изменению ракурса в 3D
 - Добавлению шума
 - Изменению освещения

Плюсы

- **Локальность:** робастны к перекрытию и шумам
- **Узнаваемость:** отдельные особенности можно соотнести и распознать
- **Количество:** многие особенности могут быть найдены даже для небольших объектов
- **Скорость обработки:** приближенное к реальному времени (bullshit)

Инвариантные локальные особенности



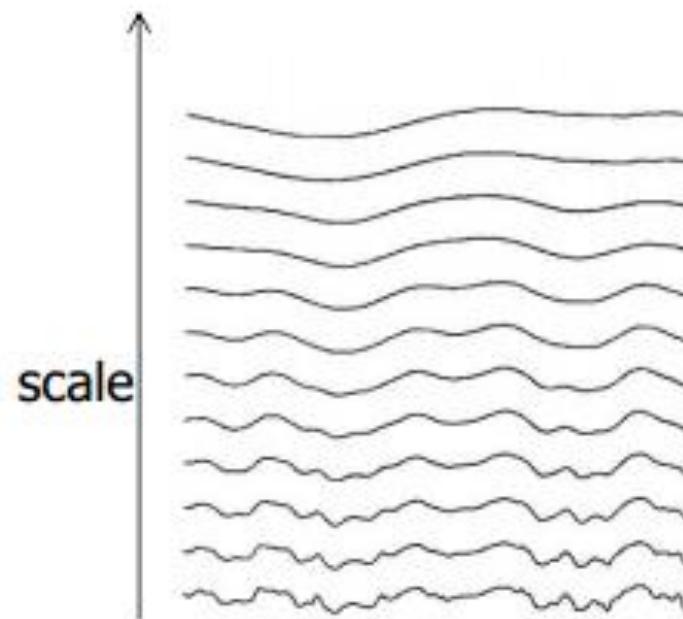
Шаги по выявлению key points

- Выбрать пики в пространстве масштаба
 - Т.е. потенциальные области для обнаружения характеристических признаков
- Локализация основных точек интереса
 - Уточнение координат, удаление и фильтрация ненужных точек
- Присвоение ориентации
- Построение дескрипторов для характеристических точек
 - Описание точек как вектор высокой размерности

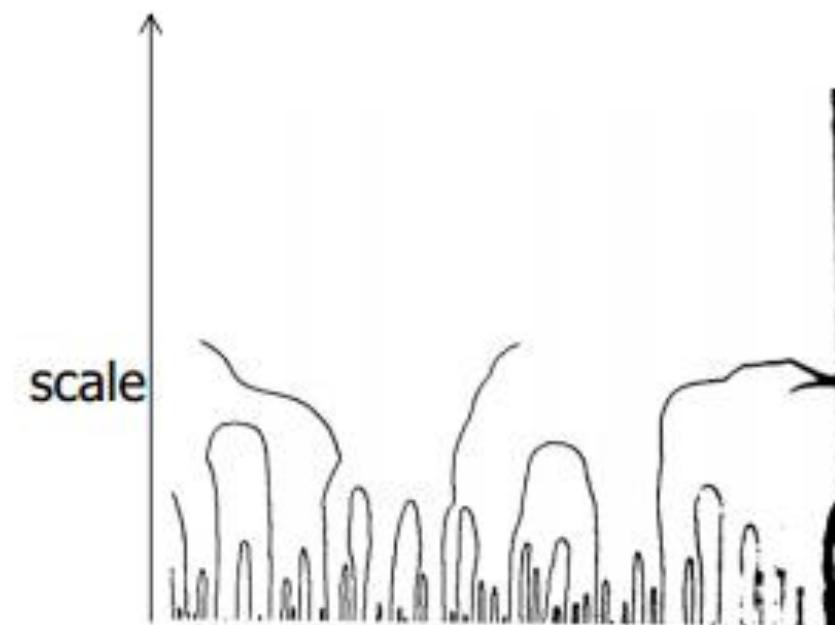
Масштаб

- Какой должна быть сигма для Canny и для LG детекторов?
- Если использовать множество значений сигма (масштабов), как потом совместить в одно изображение полученные карты краев?

Пространство масштабов

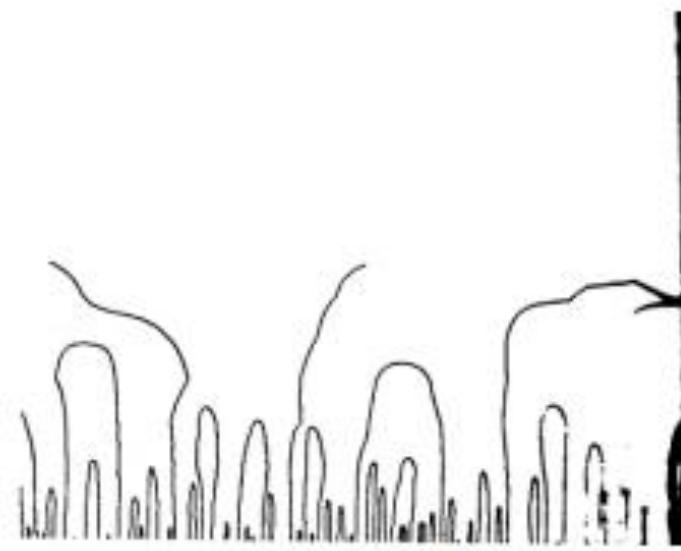


Multiple smooth versions of a signal

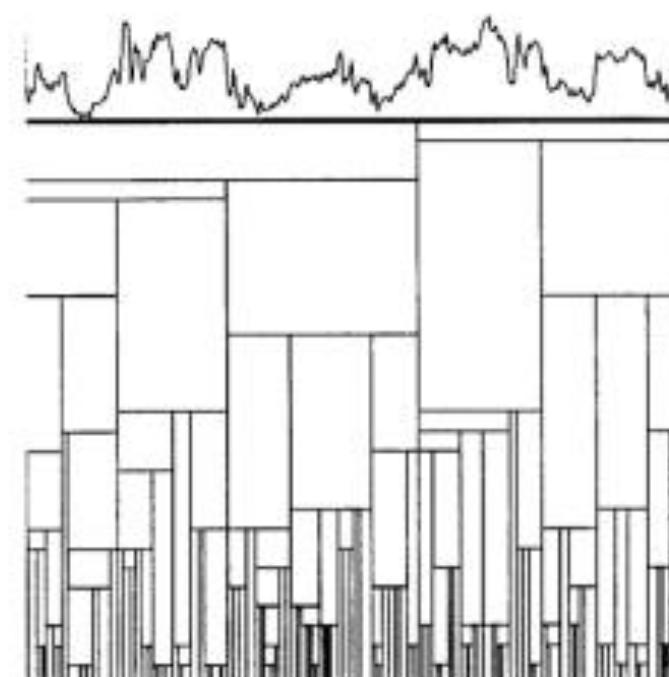


Zerocrossings at multiple scale

Пространство масштабов



Scale Space



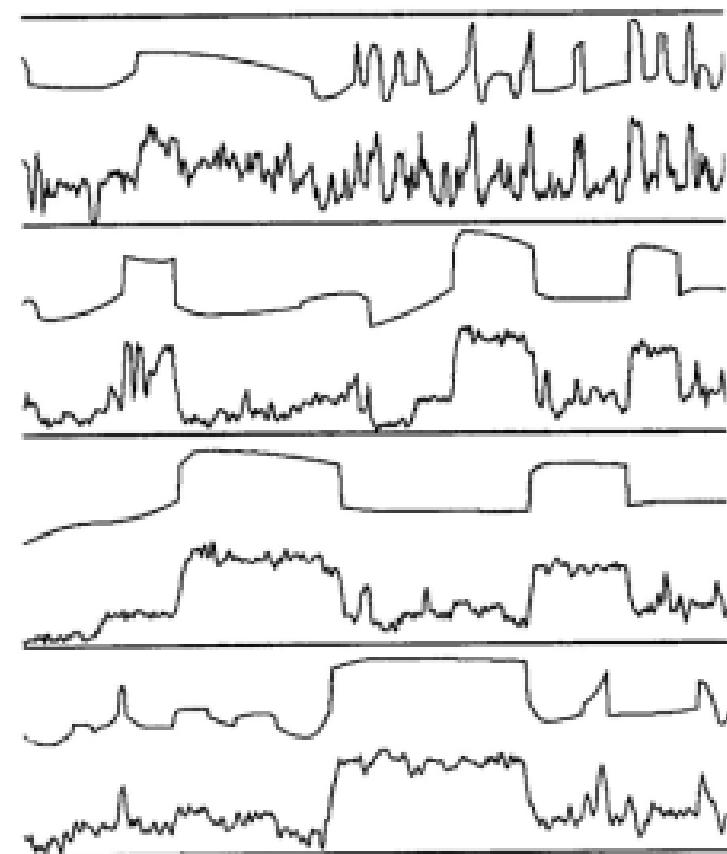
Interval Tree

**Пространство
масштабов.**

Зачем???

Пространство масштабов

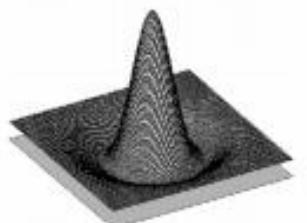
A top level description of several signals using stability criterion.



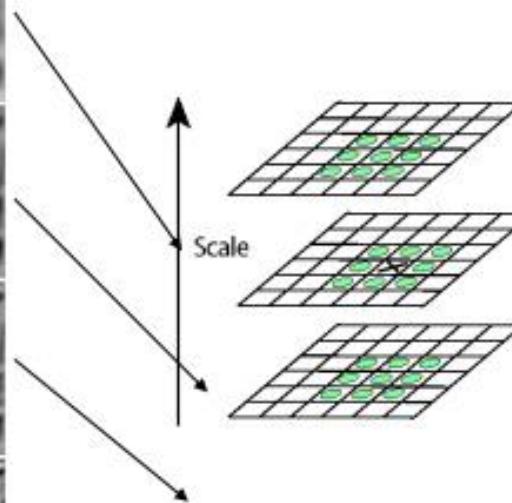
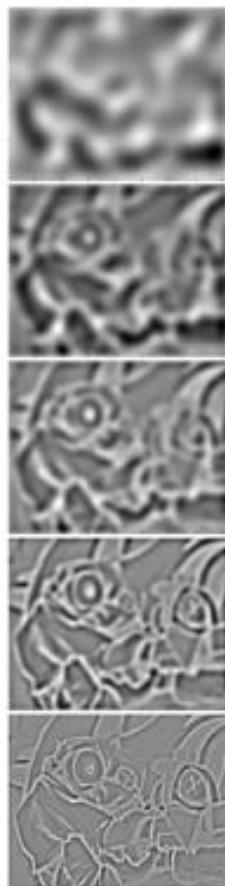
LoG

Interest points:

**Local maxima in scale
space of Laplacian-of-
Gaussian**

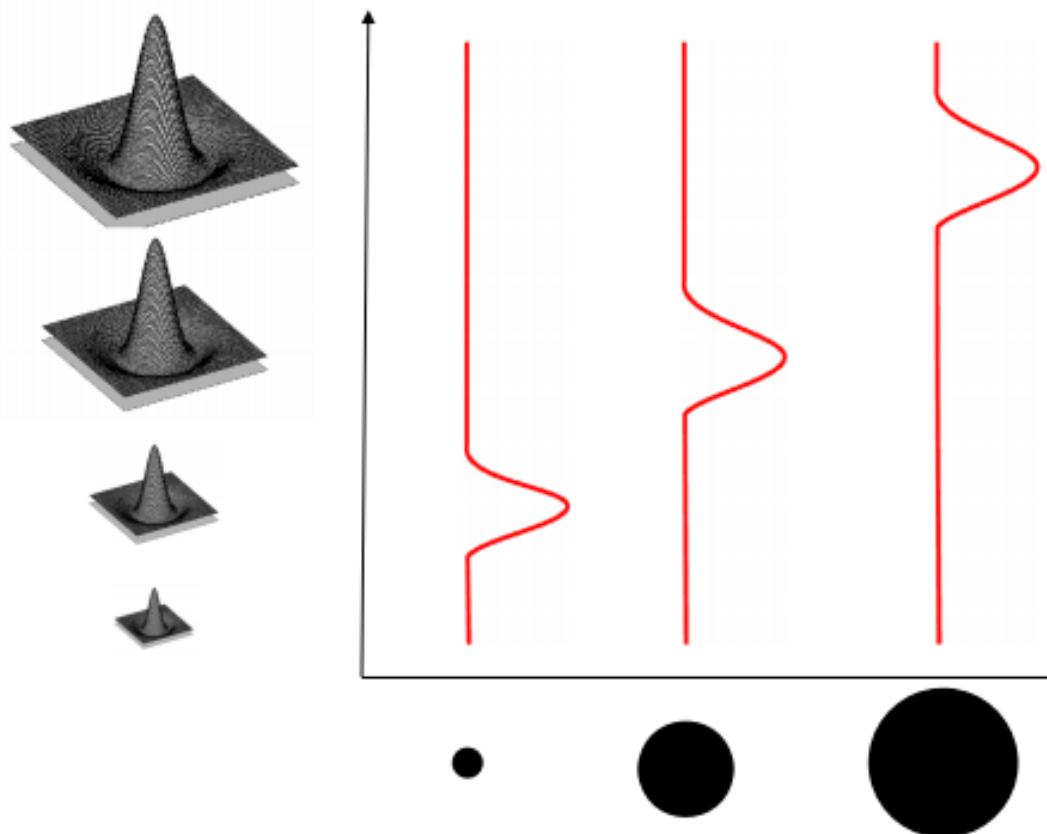


$$L_{xx}(\sigma) + L_{yy}(\sigma) \rightarrow \sigma^3$$
$$\sigma^4$$
$$\sigma^2$$
$$\sigma$$



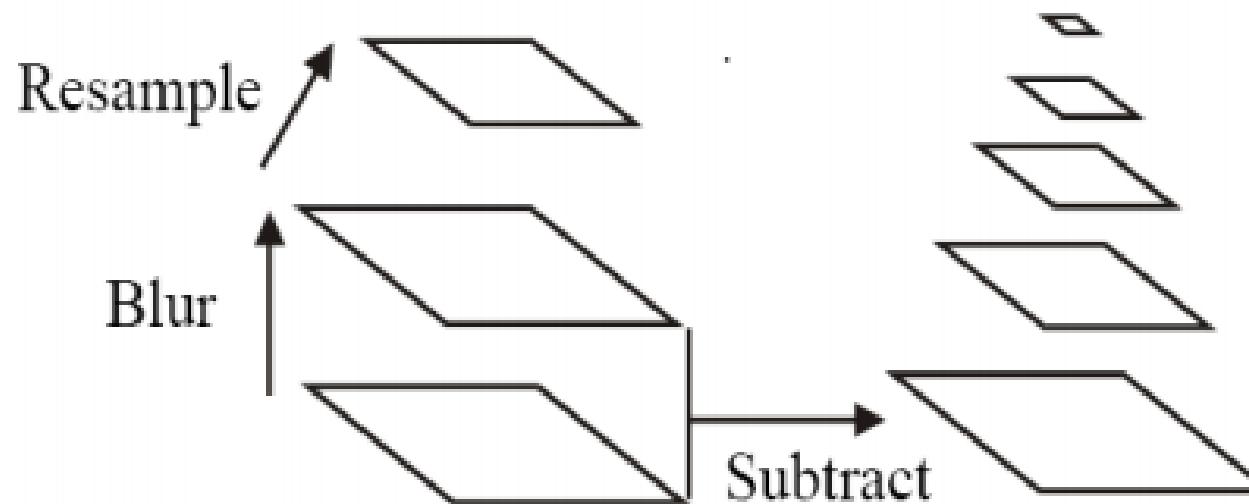
⇒ List of
 (x, y, σ)

- Laplacian-of-Gaussian = “blob” detector



Построение пространства масштабов

- All scales must be examined to identify scale-invariant features
- An efficient function is to compute the Laplacian Pyramid (Difference of Gaussian) (Burt & Adelson, 1983)



Приближение LoG по DoG

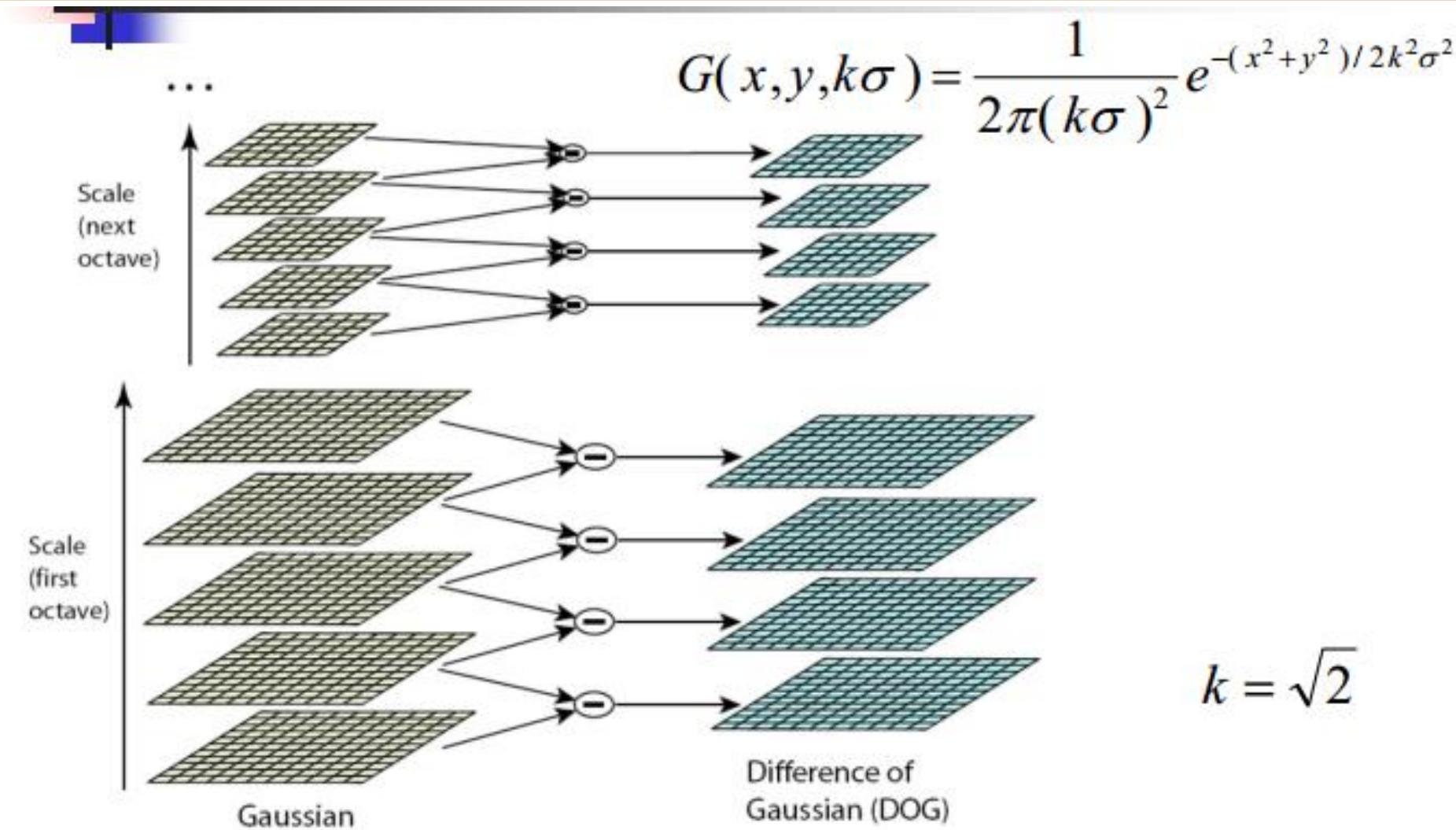
$$\frac{\partial G}{\partial \sigma} = \sigma \Delta^2 G \quad \text{Heat Equation}$$

$$\sigma \Delta^2 G = \frac{\partial G}{\partial \sigma} = \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

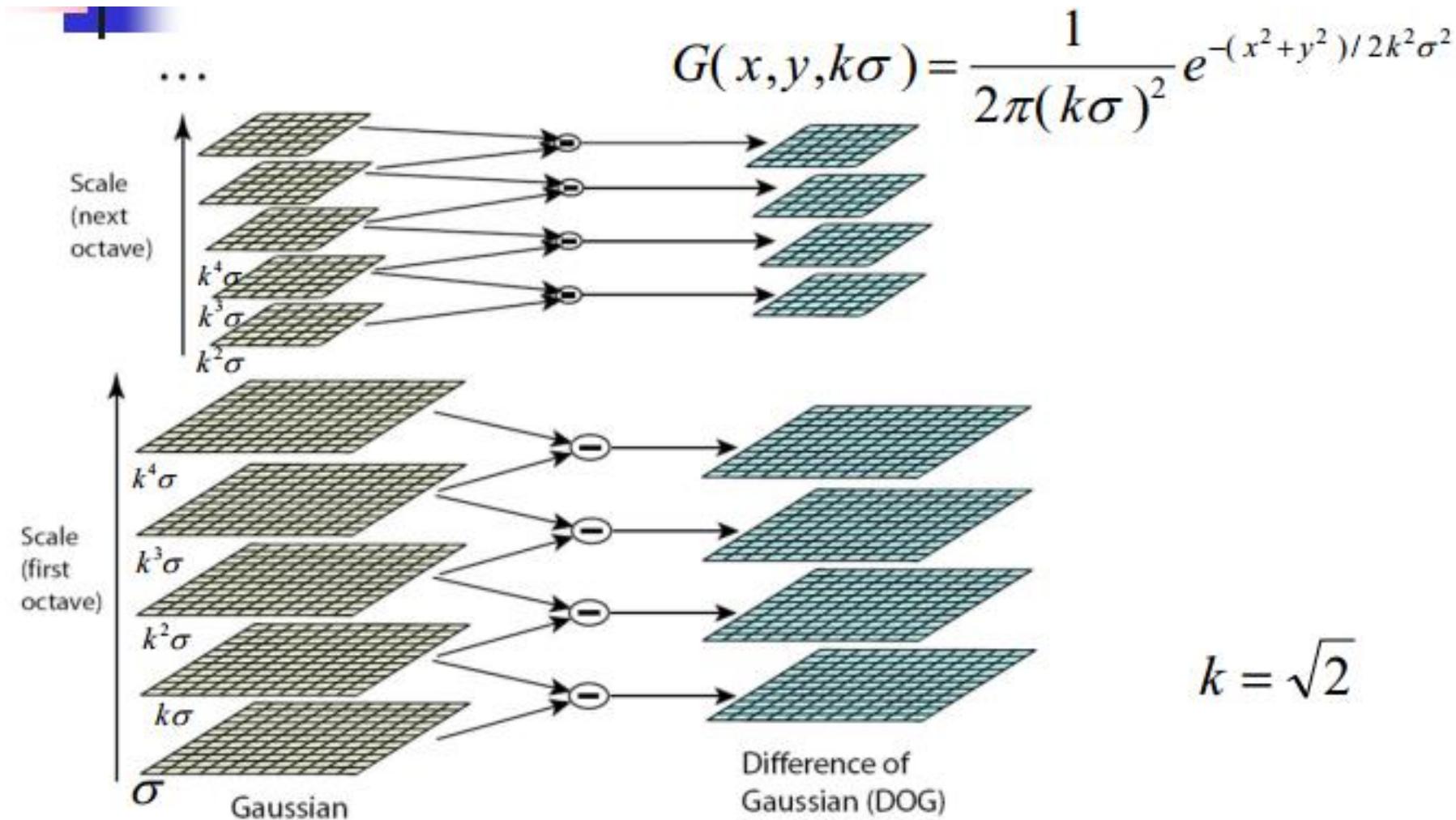
$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \Delta^2 G$$

Typical values: $\sigma = 1.6$; $k = \sqrt{2}$

Построение пространства масштабов

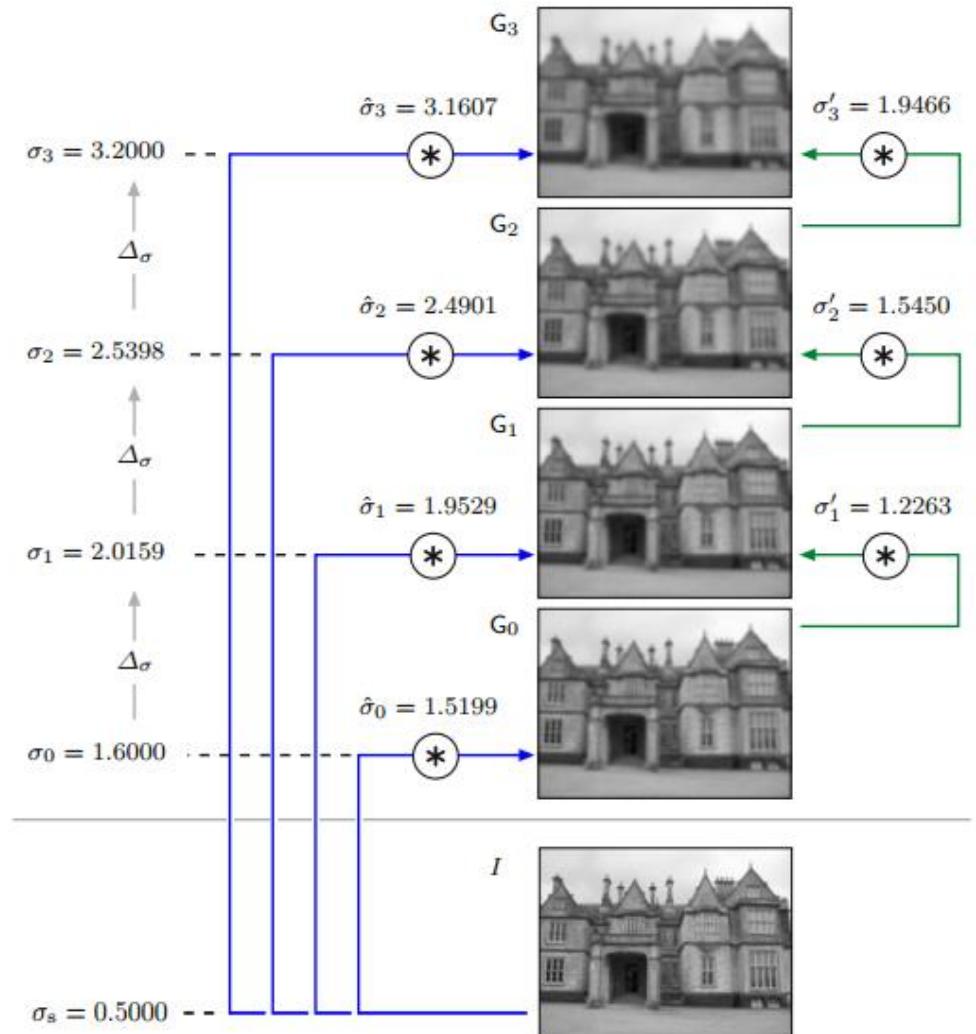


Построение пространства масштабов



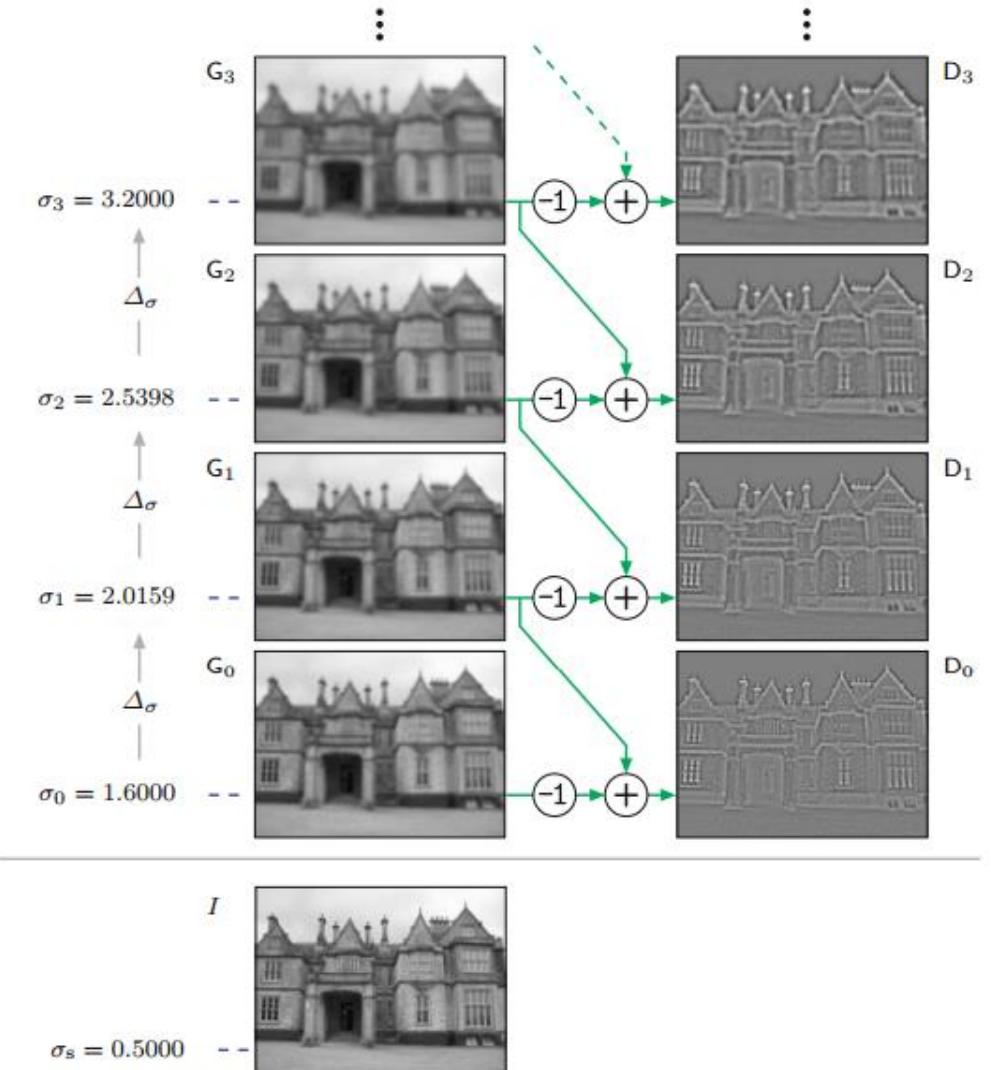
Gaussian scale space

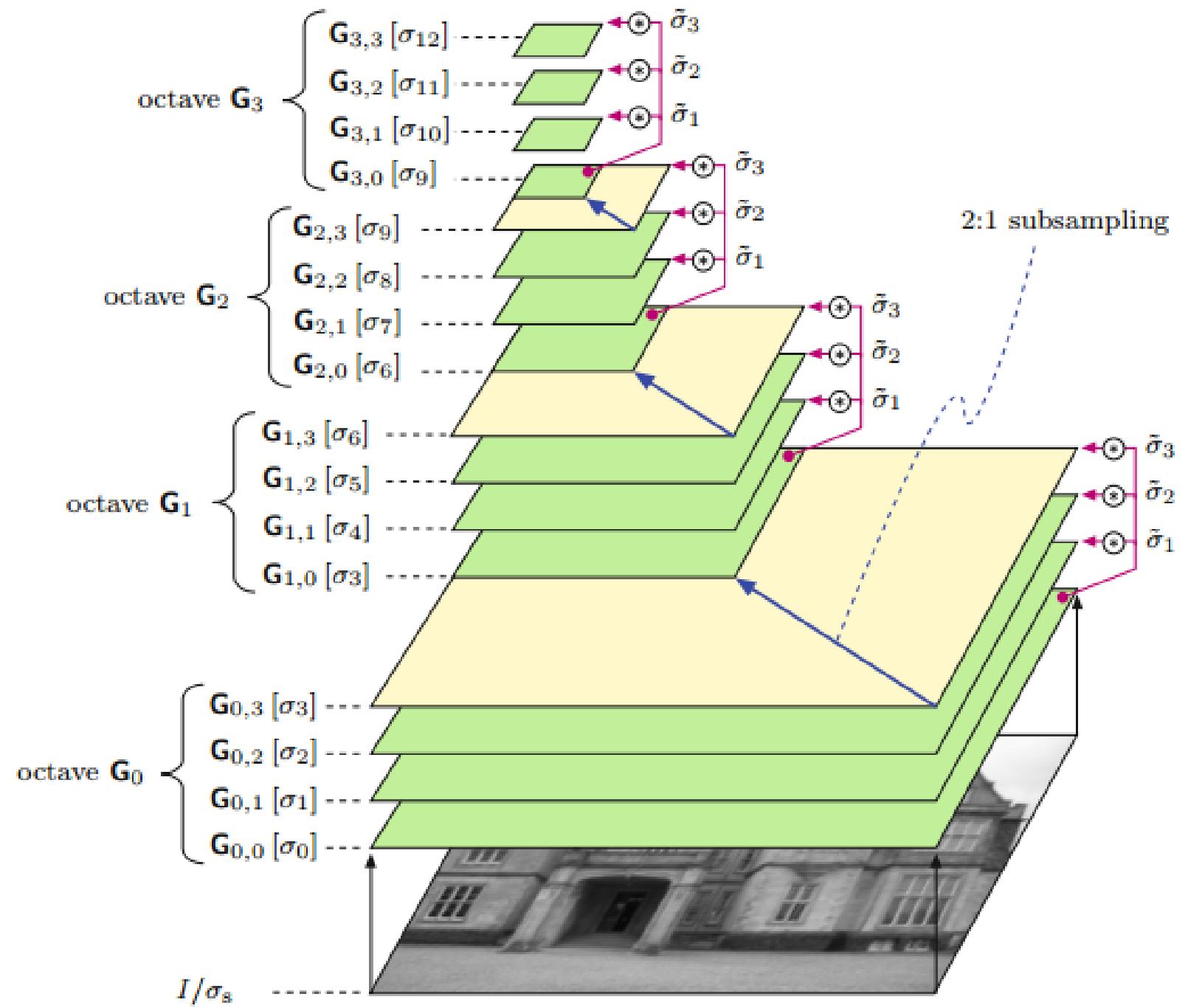
- Gaussian scale space construction (first four levels).
The discrete input image I is assumed to be pre-filtered with a Gaussian of width $\sigma_s = 0.5$; the scale of the initial level (base scale offset) is set to $\sigma_0 = 1.6$. The discrete scale space levels G_0, G_1, \dots (at absolute scales $\sigma_0, \sigma_1, \dots$) are slices through the continuous scale space. Scale levels can either be calculated by filtering directly from the discrete image I with Gaussian kernels of width $\hat{\sigma}_0, \hat{\sigma}_1, \dots$ (blue arrows) or, alternatively, by recursively filtering with $\sigma_1, \sigma_2, \dots$ (green arrows).



DoG scale space

- Difference-of-Gaussians (DoG) scale space construction. The differences of successive levels G_0, G_1, \dots of the Gaussian scale space (see Fig. 7.6) are used to approximate an LoG scale space. Each DoG level D_m is calculated as the point-wise difference between Gaussian levels $G_{m+1} - G_m$. The values in D_0, \dots, D_3 are scale-normalized (see Eqn. (7.14)) and mapped to a uniform intensity range for viewing





Algorithm for hierarchical Gaussian scale space

- Building a hierarchical Gaussian scale space. The input image I is first blurred to scale σ_0 by filtering with a Gaussian kernel of width σ^0 . In each octave G_p , the scale levels $G_{p,q}$ are calculated from the base level $G_{p,0}$ by filtering with a set of Gaussian filters of width $\sigma^1, \dots, \sigma^Q$. The base level $G_{p,0}$ of each higher octave is obtained by subsampling the top level $G_{p-1,Q}$ of the previous octave.

1: **BUILDGAUSSIANSCALESPACE**($I, \sigma_s, \sigma_0, Q, P$)

Input: I , source image; σ_s , sampling scale; σ_0 , reference scale of the first octave; Q , number of scale steps per octave; P , number of octaves.
Returns a hierarchical Gaussian scale space representation \mathbf{G} of the image I .

- 2: $\hat{\sigma}_0 \leftarrow (\sigma_0^2 - \sigma_s^2)^{1/2}$ ▷ Eqn. (7.26)
- 3: $\mathbf{G}_{0,0} \leftarrow I * H^{\mathbf{G}, \hat{\sigma}_0}$ ▷ apply 2D Gaussian filter of width $\hat{\sigma}_0$
- 4: $\mathbf{G}_0 \leftarrow \text{MAKEGAUSSIANOCTAVE}(\mathbf{G}_{0,0}, 0)$ ▷ create octave \mathbf{G}_0
- 5: **for** $p \leftarrow 1, \dots, P-1$ **do** ▷ octave index p
- 6: $\mathbf{G}_{p-1,Q} \leftarrow \mathbf{G}_{p-1}(Q)$
- 7: $\mathbf{G}_{p,0} \leftarrow \text{DECIMATE}(\mathbf{G}_{p-1,Q})$
- 8: $\mathbf{G}_p \leftarrow \text{MAKEGAUSSIANOCTAVE}(\mathbf{G}_{p,0}, p)$ ▷ create octave \mathbf{G}_p
- 9: $\mathbf{G} \leftarrow (\mathbf{G}_0, \dots, \mathbf{G}_{P-1})$
- 10: **return** \mathbf{G} . ▷ hierarchical Gaussian scale space \mathbf{G}

```

11: MAKEGAUSSIANOCTAVE( $\mathbf{G}_{p,0}, p$ )
    Input:  $\mathbf{G}_{p,0}$ , octave base level;  $p$ , octave index.
12:   for  $q \leftarrow 1, \dots, Q$  do                                ▷ level index  $q$ 
13:      $\tilde{\sigma}_q \leftarrow \sigma_0 \cdot \sqrt{2^{2q}/Q - 1}$           ▷ Eqn. (7.39)
14:      $\mathbf{G}_{p,q} \leftarrow \mathbf{G}_{p,0} * H^{\mathbf{G}, \tilde{\sigma}_q}$       ▷ apply 2D Gaussian filter of width  $\tilde{\sigma}_q$ 
15:    $\mathbf{G}_p \leftarrow (\mathbf{G}_{p,0}, \dots, \mathbf{G}_{p,Q})$ 
16:   return  $\mathbf{G}_p$ .                                         ▷ scale space octave  $\mathbf{G}_p$ 

```

```

17: DECIMATE( $\mathbf{G}_{\text{in}}$ )
    Input:  $\mathbf{G}_{\text{in}}$ , Gaussian octave level of size  $M \times N$ .
18:    $(M, N) \leftarrow \text{SIZE}(\mathbf{G}_{\text{in}})$ 
19:    $M' \leftarrow \lfloor \frac{M}{2} \rfloor, \quad N' \leftarrow \lfloor \frac{N}{2} \rfloor$ 
20:    $\mathbf{G}_{\text{out}} \leftarrow$  new image of size  $M' \times N'$ 
21:   for all image coordinates  $(u, v) \in M' \times N'$  do
22:      $\mathbf{G}_{\text{out}}(u, v) \leftarrow \mathbf{G}_{\text{in}}(2u, 2v)$            ▷ 2:1 sub-sampling
23:   return  $\mathbf{G}_{\text{out}}$ .                                     ▷ decimated octave level  $\mathbf{G}_{\text{out}}$ 

```

Octave \mathbf{G}_2
(100×75)

Gaussian



DoG



Octave \mathbf{G}_1
(200×150)

$\mathbf{G}_{1,3}$



$\mathbf{G}_{1,2}$



$\mathbf{G}_{1,1}$



$\mathbf{G}_{1,0}$



$\mathbf{D}_{1,0}$



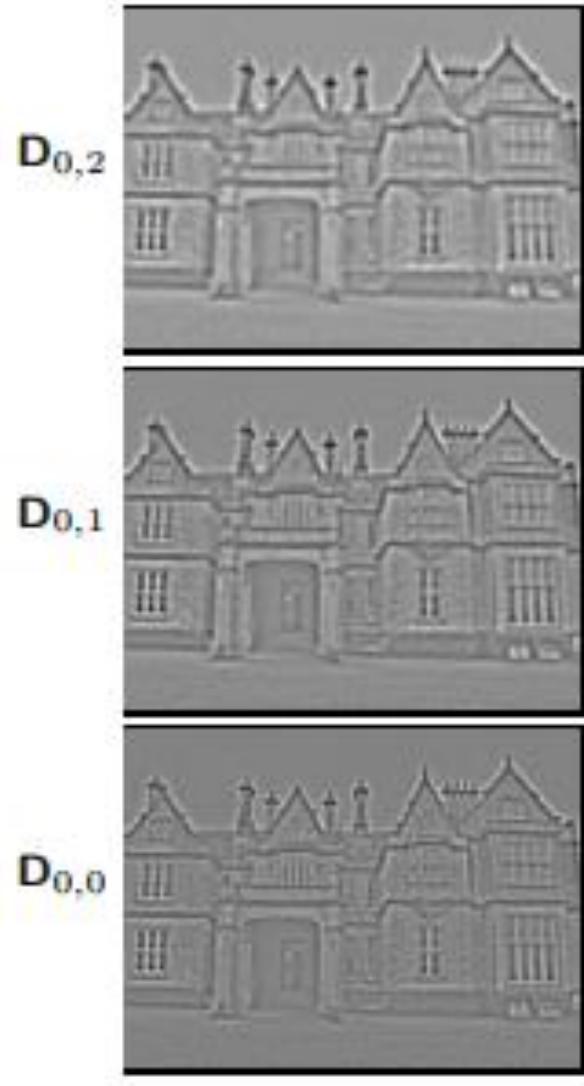
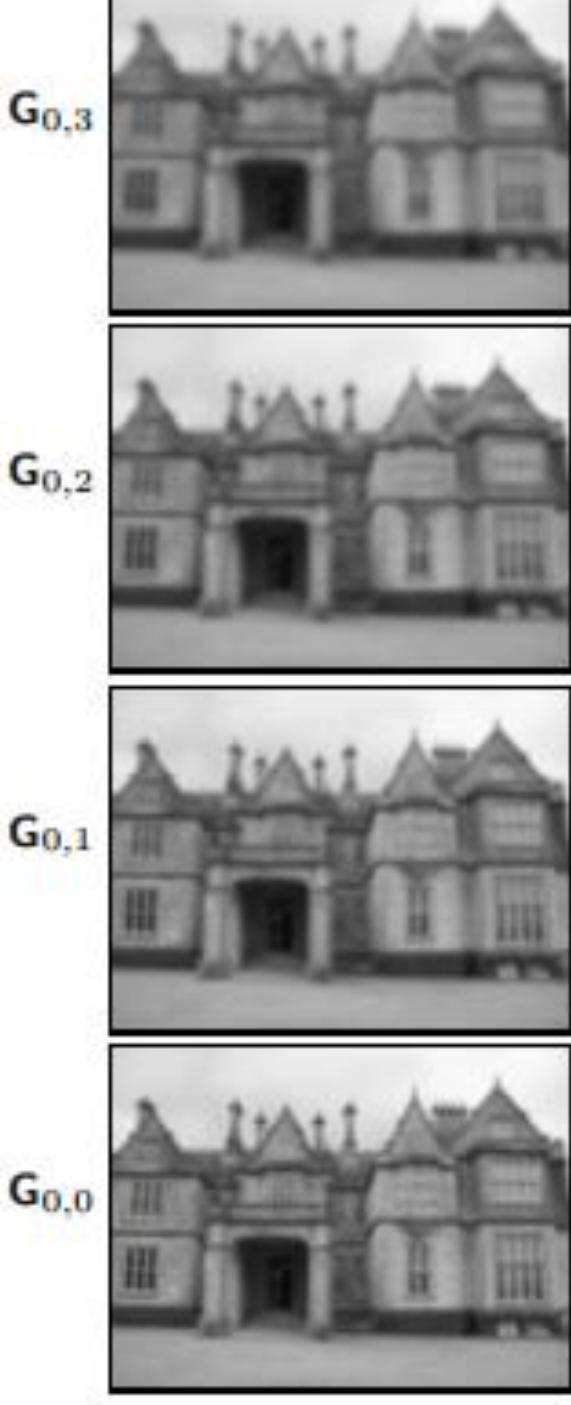
$\mathbf{D}_{1,1}$



$\mathbf{D}_{1,2}$



Octave \mathbf{G}_0
(400 × 300)



Octave **G₀**
(400 × 300 pixels)



Octave **G₁**
(200 × 150 pixels)



Octave **G₂**
(100 × 75 pixels)



G_{0,3}



G_{1,3}



G_{2,3}



G_{0,2}



G_{1,2}



G_{2,2}



G_{0,1}



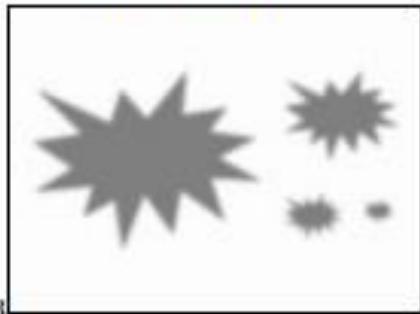
G_{1,1}



G_{2,1}

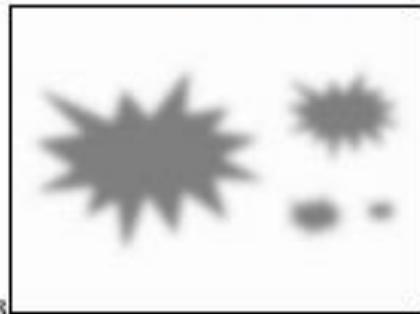


Octave \mathbf{G}_0
(400 × 300 pixels)



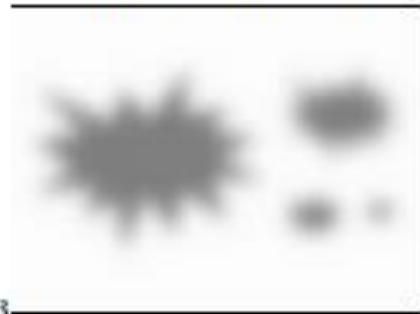
$\mathbf{G}_{0,3}$

Octave \mathbf{G}_1
(200 × 150 pixels)

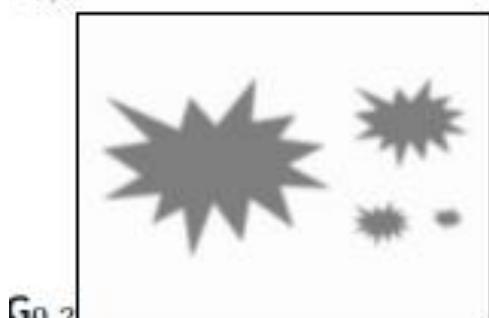


$\mathbf{G}_{1,3}$

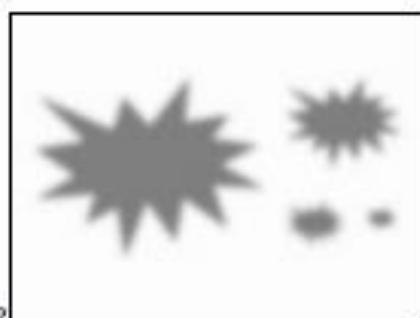
Octave \mathbf{G}_2
(100 × 75 pixels)



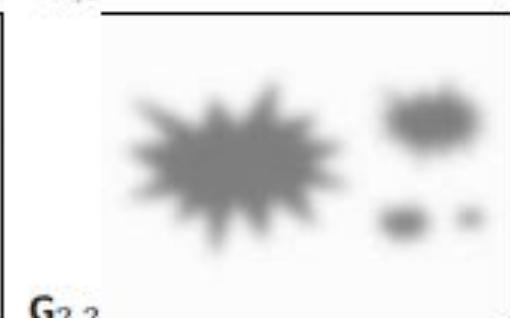
$\mathbf{G}_{2,3}$



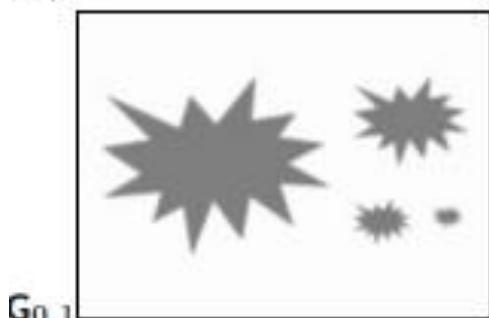
$\mathbf{G}_{0,2}$



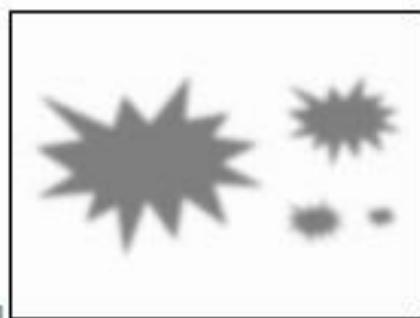
$\mathbf{G}_{1,2}$



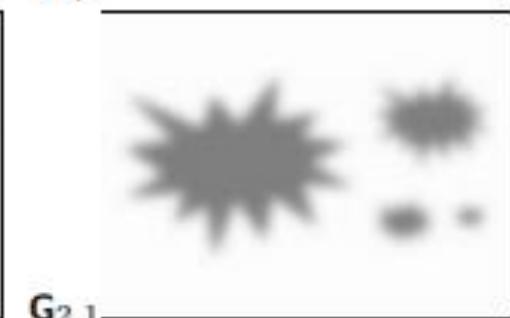
$\mathbf{G}_{2,2}$



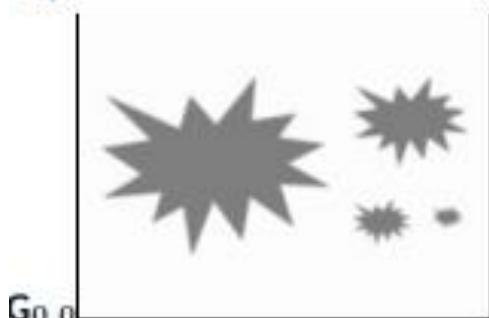
$\mathbf{G}_{0,1}$



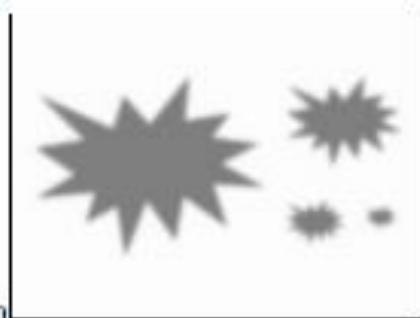
$\mathbf{G}_{1,1}$



$\mathbf{G}_{2,1}$



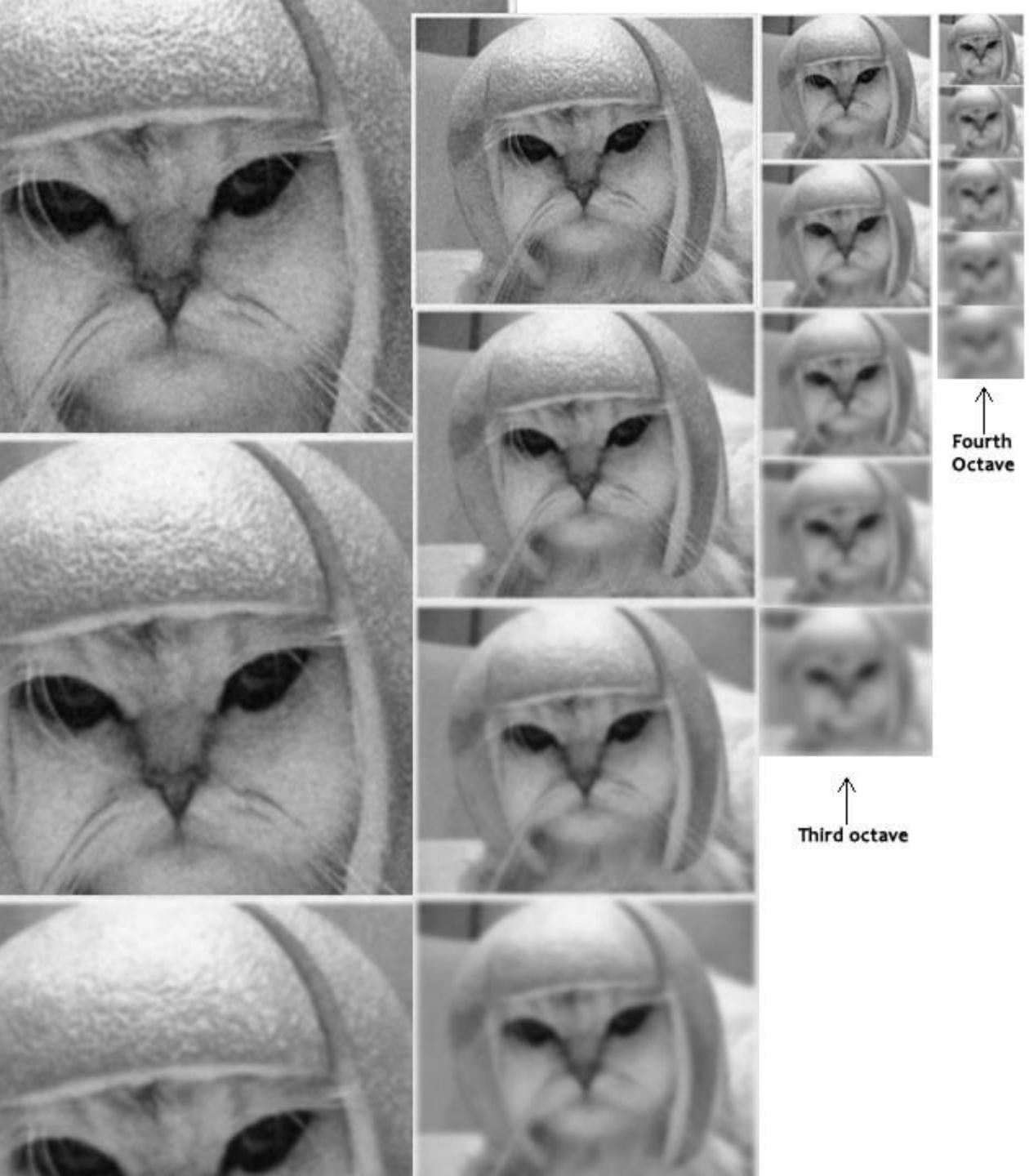
$\mathbf{G}_{0,0}$



$\mathbf{G}_{1,0}$



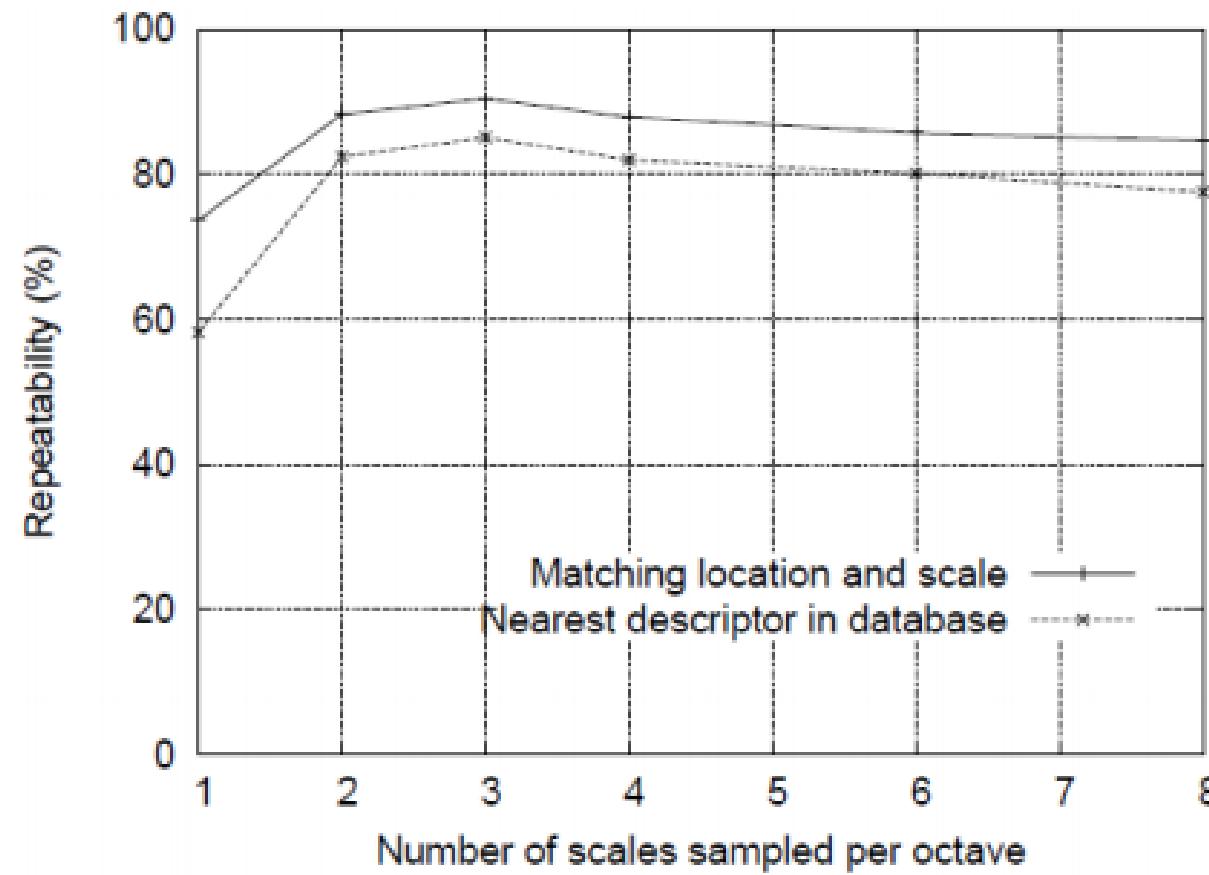
$\mathbf{G}_{2,0}$



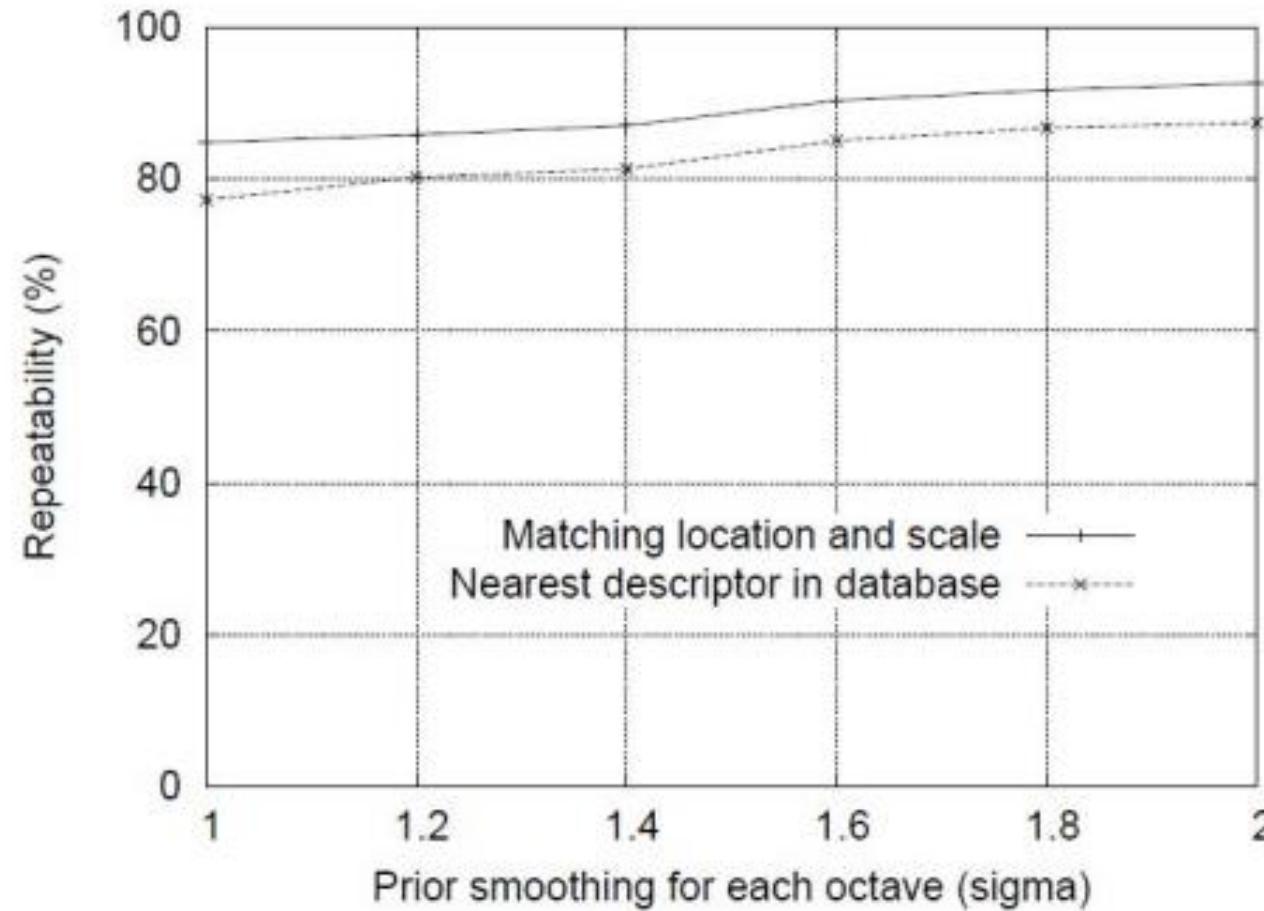
	scale	→			
octave	0.707107	1.000000	1.414214	2.000000	2.828427
	1.414214	2.000000	2.828427	4.000000	5.656854
	2.828427	4.000000	5.656854	8.000000	11.313788
	5.656854	8.000000	11.313788	16.000000	22.627417

$$\sigma = .707187.6; k = \sqrt{2}$$

Сколько масштабов на октаву?

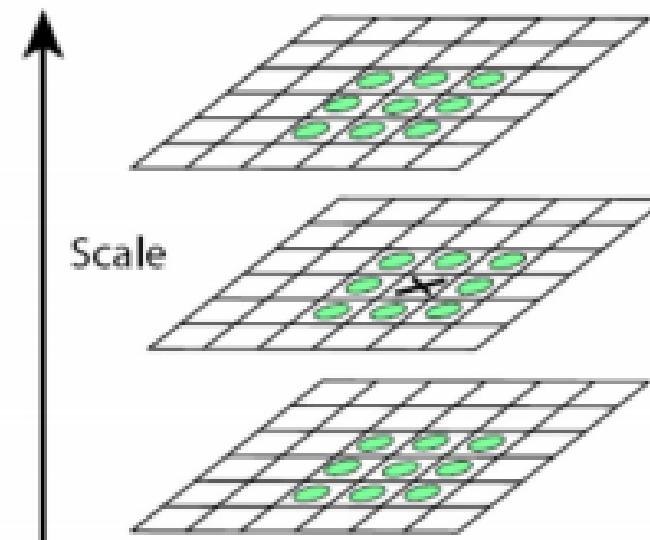


Изначальное значение сигмы



Выявление пика

- Compare a pixel (**X**) with 26 pixels in current and adjacent scales (**Green Circles**)
- Select a pixel (**X**) if larger/smaller than all 26 pixels
- Large number of extrema, computationally expensive
 - Detect the most stable subset with a coarse sampling of scales
- Note that keypoints are not detected in the lowermost and topmost scales. There simply aren't enough neighbours to do the comparison. So simply skip them!

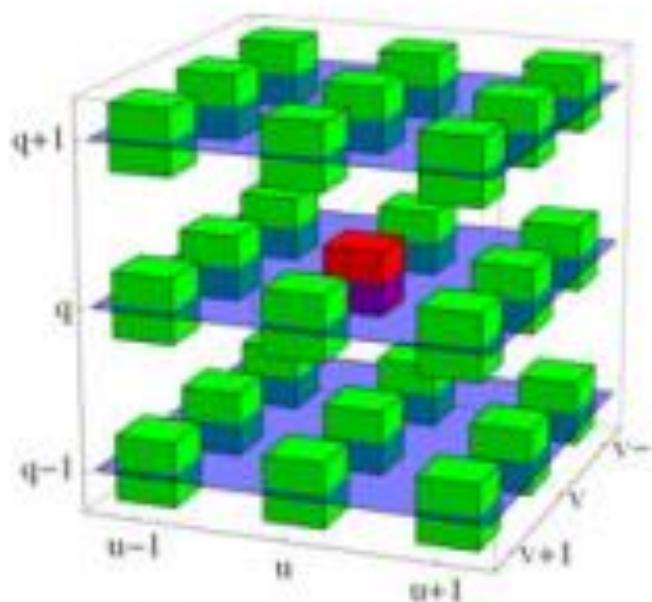


Key point selection and refinement

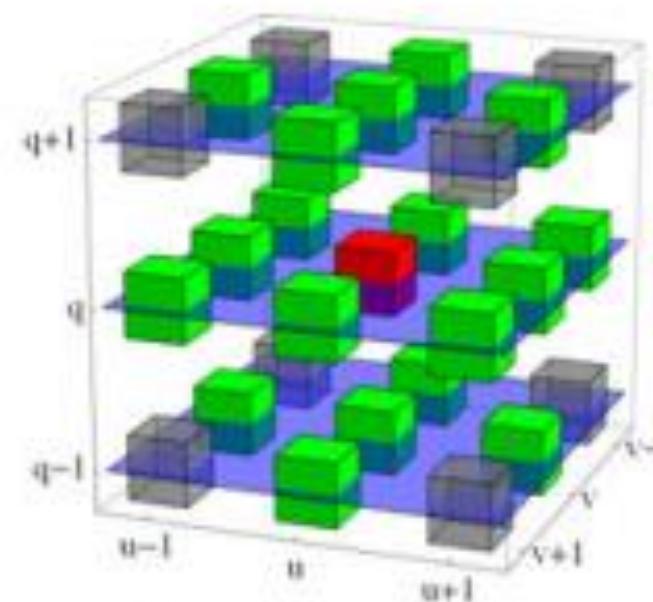
3 шага нахождения key points

- (1) найти экстремумы в DoG scale space,
- (2) уточнить координаты точек с помощью локальной интерполяции
- (3) выпилить все края

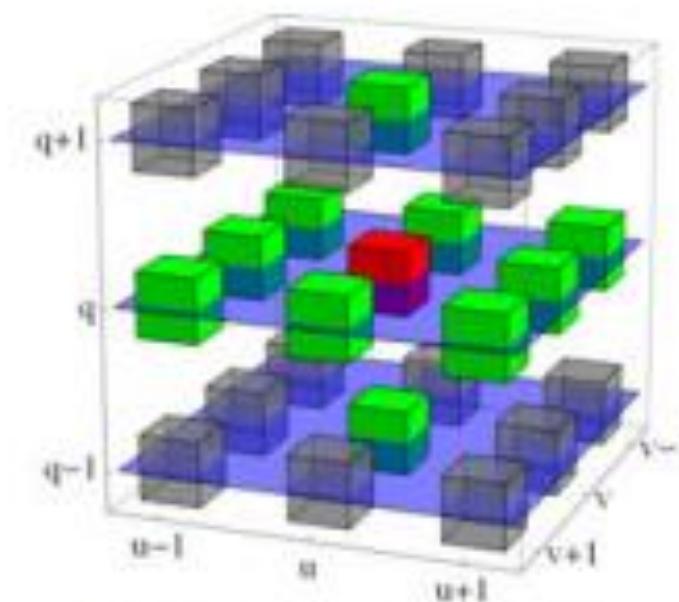
(1) 3D окрестности для выявления экстремумов в DoG scale space



(a) 26-neighborhood



(b) 18-neighborhood



(c) 10-neighborhood

- For a given position c at some octave p and level q in the hierarchical DoG scale space D , the set $ND(c)$ contains the complete 3D neighborhood of 27 cells. The red cube marks the center coordinate $c = (u, v, q)$ in scale space octave D_p (u, v are the relative space coordinates and q is the scale level index). **A local maximum/minimum is detected if the DoG value at the center is greater/smaller than any of the neighborhood values (green cubes).**

(2) Уточнение координат

At any extremal position $\mathbf{c} = (u, v, q)$ in octave p of the hierarchical DoG scale space \mathbf{D} , the corresponding $3 \times 3 \times 3$ neighborhood $\mathcal{N}_D(\mathbf{c})$ is used to estimate the elements of the continuous 3D gradient, that is,

$$\nabla_D(\mathbf{c}) = \begin{pmatrix} d_x \\ d_y \\ d_\sigma \end{pmatrix} = 0.5 \cdot \begin{pmatrix} D(\mathbf{c} + \mathbf{e}_i) - D(\mathbf{c} - \mathbf{e}_i) \\ D(\mathbf{c} + \mathbf{e}_j) - D(\mathbf{c} - \mathbf{e}_j) \\ D(\mathbf{c} + \mathbf{e}_k) - D(\mathbf{c} - \mathbf{e}_k) \end{pmatrix}, \quad (7.54)$$

with $D(\mathbf{c})$ as defined in Eqn. (7.51). Similarly, the 3×3 Hessian matrix for position \mathbf{c} is obtained as

$$\mathbf{H}_D(\mathbf{c}) = \begin{pmatrix} d_{xx} & d_{xy} & d_{x\sigma} \\ d_{xy} & d_{yy} & d_{y\sigma} \\ d_{x\sigma} & d_{y\sigma} & d_{\sigma\sigma} \end{pmatrix}, \quad (7.55)$$

(2) Уточнение координат

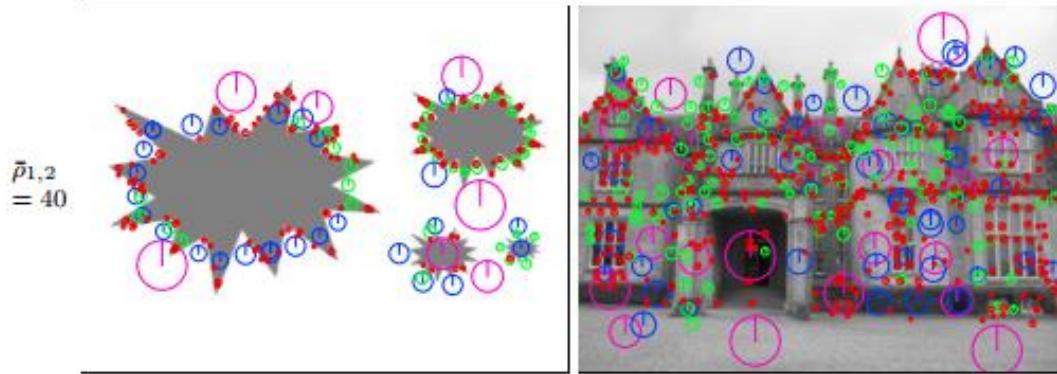
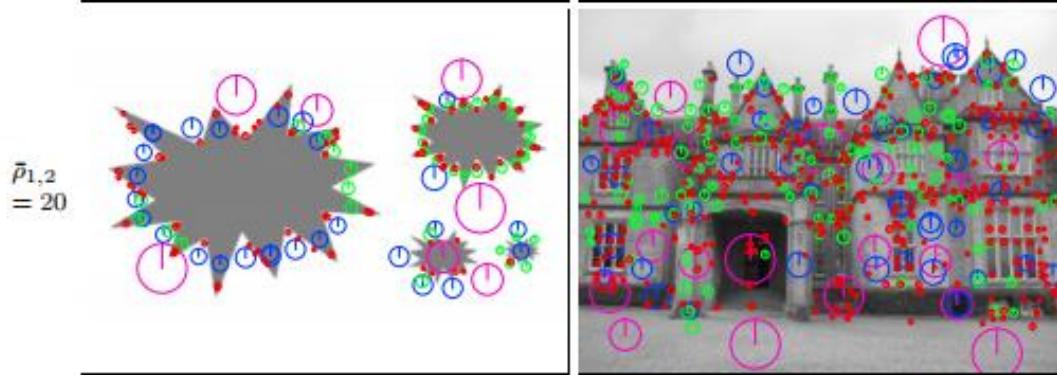
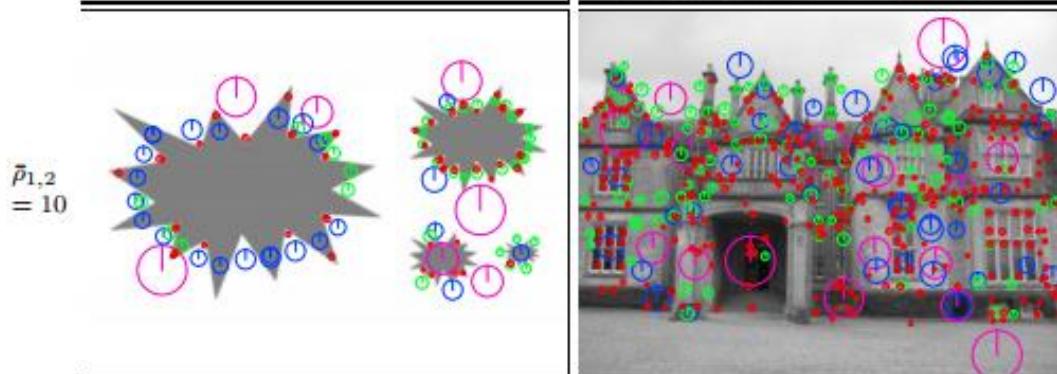
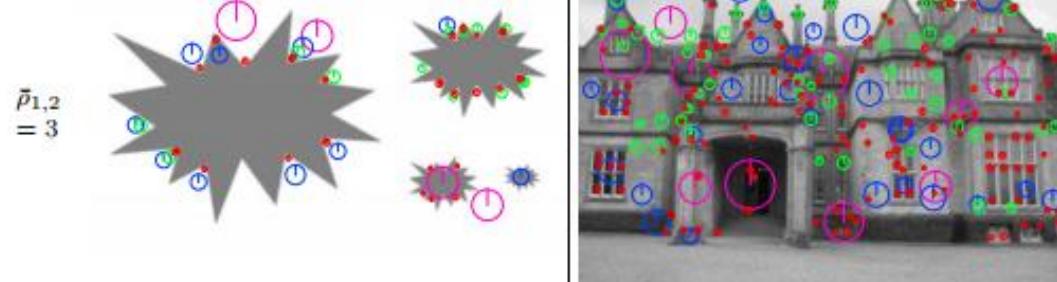
$$\begin{aligned} d_{xx} &= D(c - e_i) - 2 \cdot D(c) + D(c + e_i), \\ d_{yy} &= D(c - e_j) - 2 \cdot D(c) + D(c + e_j), \\ d_{\sigma\sigma} &= D(c - e_k) - 2 \cdot D(c) + D(c + e_k), \\ d_{xy} &= \frac{1}{4} \cdot [D(c + e_i + e_j) - D(c - e_i + e_j) - D(c + e_i - e_j) + D(c - e_i - e_j)], \\ d_{x\sigma} &= \frac{1}{4} \cdot [D(c + e_i + e_k) - D(c - e_i + e_k) - D(c + e_i - e_k) + D(c - e_i - e_k)], \\ d_{y\sigma} &= \frac{1}{4} \cdot [D(c + e_j + e_k) - D(c - e_j + e_k) - D(c + e_j - e_k) + D(c - e_j - e_k)]. \end{aligned} \quad (7.56)$$

(3) Долой края

For a particular lattice point $c = (u, v, q)$ in DoG scale space, with neighborhood $N_D(c)$, the 2×2 Hessian matrix for the spatial coordinates is

$$\mathbf{H}_{xy}(c) = \begin{pmatrix} d_{xx} & d_{xy} \\ d_{xy} & d_{yy} \end{pmatrix}, \quad (7.63)$$

with d_{xx} , d_{xy} , d_{yy} as defined in Eqn. (7.56), i.e., these values can be extracted from the corresponding 3×3 Hessian matrix $\mathbf{H}_D(c)$ (see Eqn. (7.55)).



Локализация key point

- Candidates are chosen from extrema detection



original image



extrema locations

Первичная фильтрация

- 1. Low contrast candidates
- 2. Poorly localized candidates along an edge
- Taylor series expansion of DOG, D .

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

- Minima or maxima is located at $\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$
- Value of $D(\mathbf{x})$ at minima/maxima must be large, $|D(x)| > th$.

Первичная фильтрация



from 832 key points to 729 key points, th=0.03.

Последующая фильтрация

- Analogous to Harris corner detector
- Compute Hessian of D

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad Tr(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2$$
$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \lambda_1\lambda_2$$

- Remove outliers by evaluating

$$\frac{Tr(H)^2}{Det(H)} = \frac{(r+1)^2}{r} \qquad r = \frac{\lambda_1}{\lambda_2}$$

Последующая фильтрация

- Following quantity is minimum when $r=1$
- It increases with r

$$\frac{Tr(H)^2}{\text{Det}(H)} = \frac{(r+1)^2}{r}$$
$$r = \frac{\lambda_1}{\lambda_2}$$

- Eliminate key points if $r > 10$

Последующая фильтрация

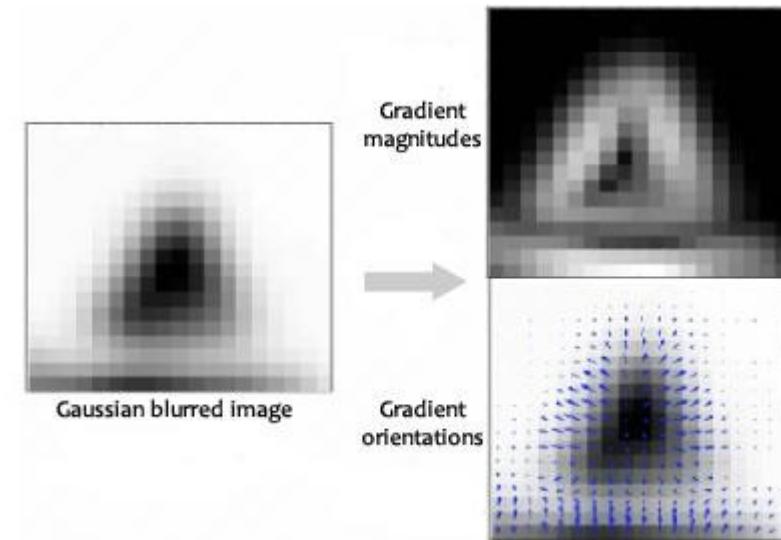


from 729 key points to 536 key points.

ВОПРОС ОРИЕНТАЦИИ

Назначение ориентации

- To achieve rotation invariance
- Compute central derivatives, gradient magnitude and direction of L (smooth image) at the scale of key point (x,y)

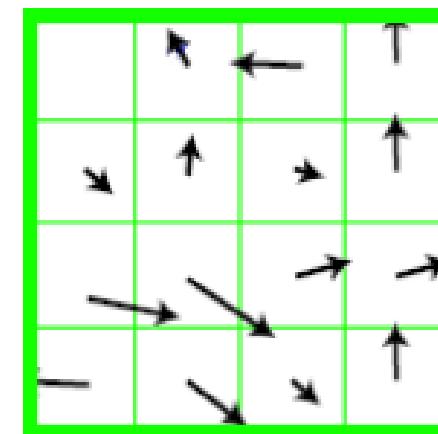


$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y)))$$

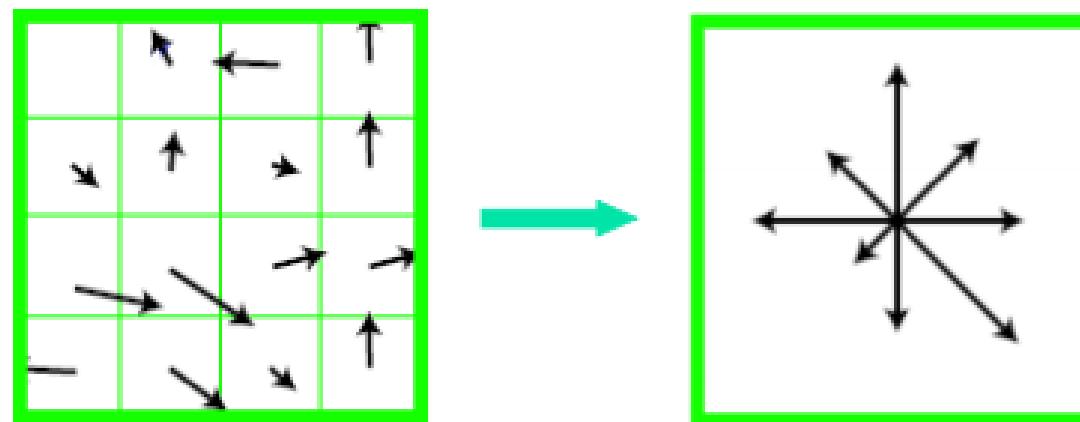
Назначение ориентации

- Create a weighted direction histogram in a neighborhood of a key point (36 bins)
- Weights are
 - Gradient magnitudes
 - Spatial gaussian filter with $\sigma=1.5 \times \text{scale of key point}$

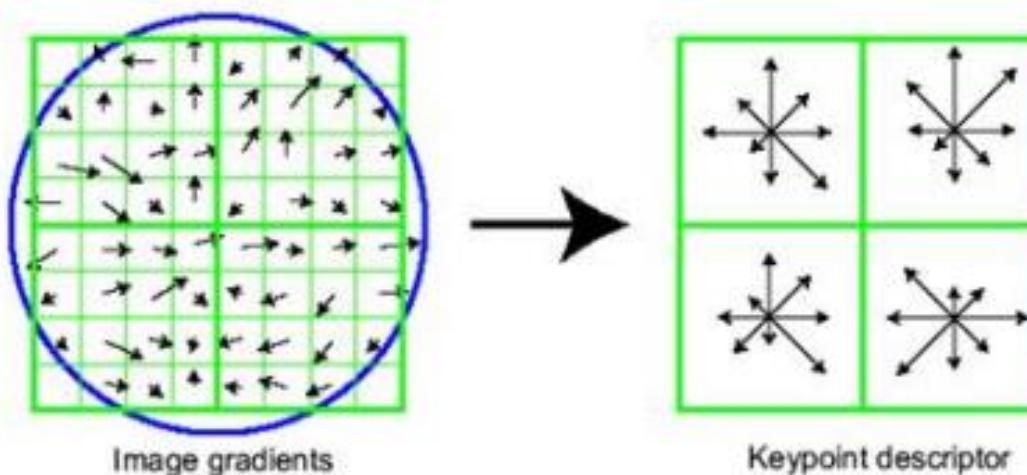


Назначение ориентации

- Select the peak as direction of the key point
- Introduce additional key points (same location) at local peaks (within 80% of max peak) of the histogram with different directions

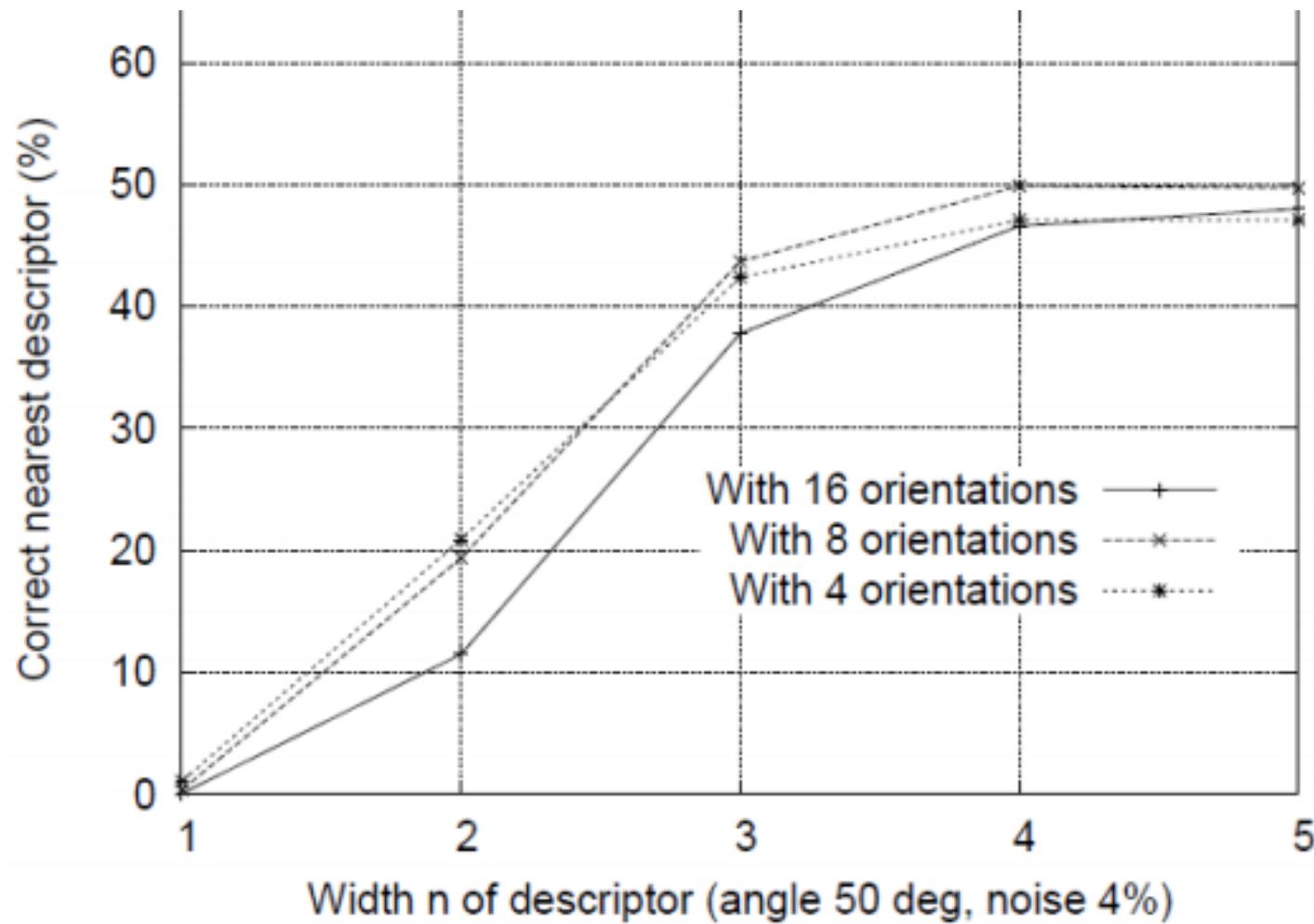


Построение дескриптора



- Для учета локальных свойств разделим окрестность на блоки сеткой, в каждом блоке посчитаем свою гистограмму градиентов
 - Обычно – сетка 4×4 , в каждой гистограмма с 8ю ячейками
 - Стандартная длина вектора-дескриптора – 128 ($4 \times 4 \times 8$)
 - Можем использовать обычную меру SSD для сравнения дескрипторов
 - Можем использовать другие метрики, учитывающие, что дескриптор SIFT – это гистограмма

Регионы дескрипторов ($n \times n$)



Выделение локальных дескрипторов

- Store numbers in a vector
- Normalize to unit vector (**UN**)
 - Illumination invariance (affine changes)
- For non-linear intensity transforms
 - Bound **Unit Vector** items to maximum 0.2 (remove larger gradients)
 - Renormalize to unit vector

Matching

- Match the key points against a database of that obtained from training images.
- Find the nearest neighbor i.e. a key point with minimum Euclidean distance.
 - Efficient Nearest Neighbor matching
 - Looks at ratio of distance between best and 2nd best match (.8)

Matching



Matching

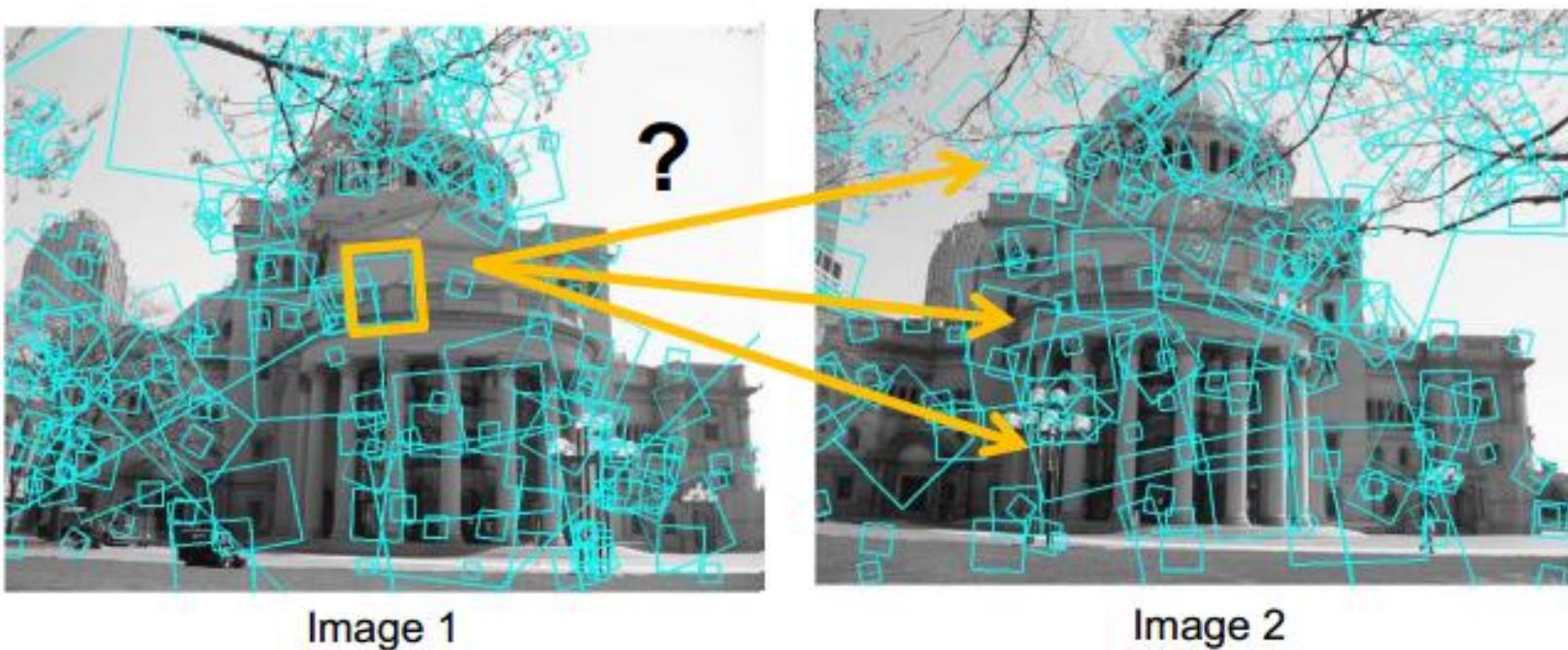


Image 1

Image 2

- To generate **candidate matches**, find patches that have the most similar appearance or SIFT descriptor
- Simplest approach: compare them all, take the closest (or closest k , or within a thresholded distance)

Неоднозначность



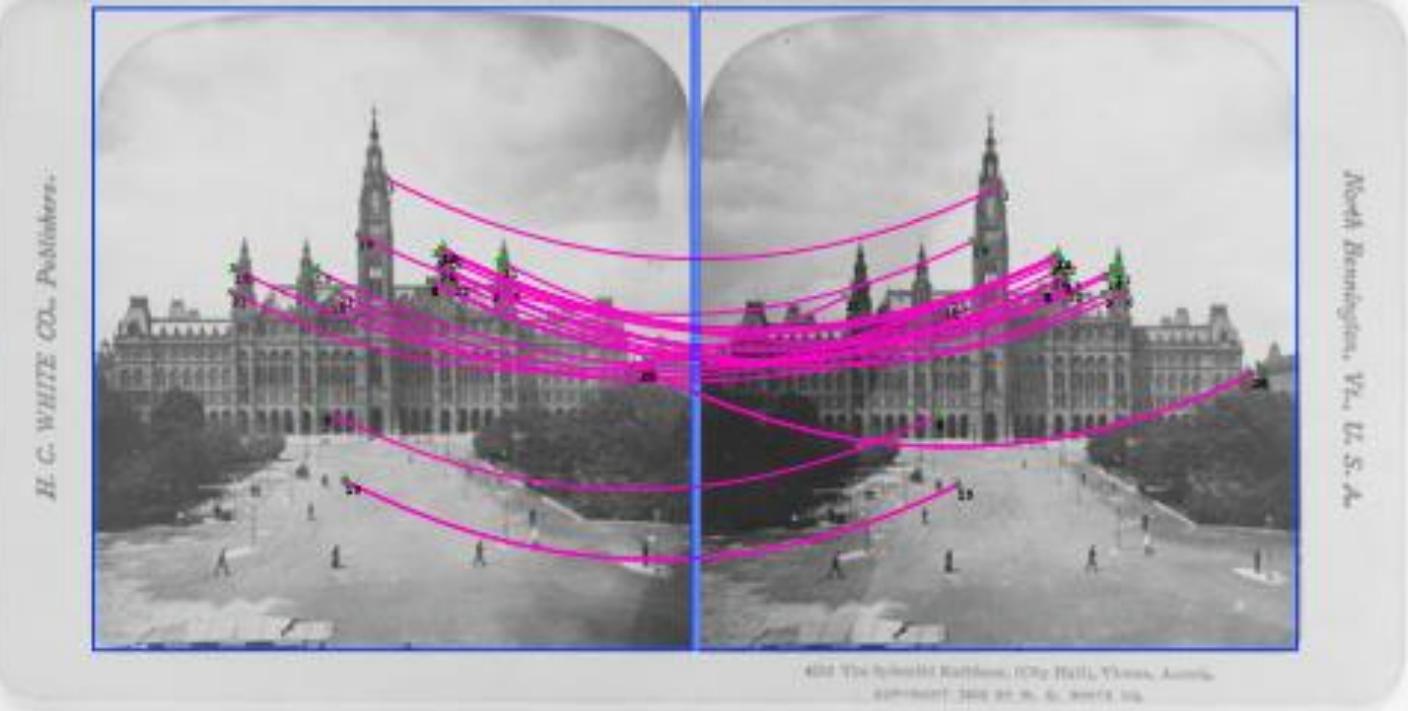
Image 1



Image 2

- At what distance do we have a good match?
- To add robustness to matching, can consider **ratio** : $\text{distance to best match} / \text{distance to second best match}$
- If low, first match looks good.
- If high, could be ambiguous match.

L_2
(b)



L_∞
(c)



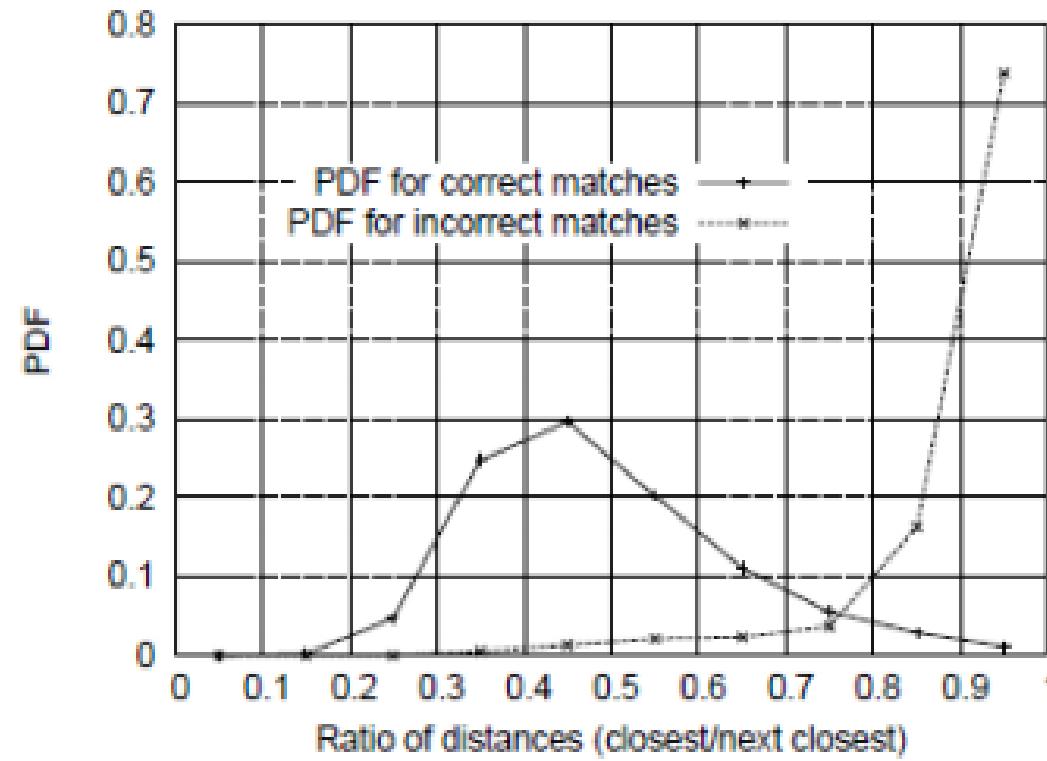
(b)



(c)



Неоднозначность



В следующих сериях...

