академия больших данных

mail.ru group

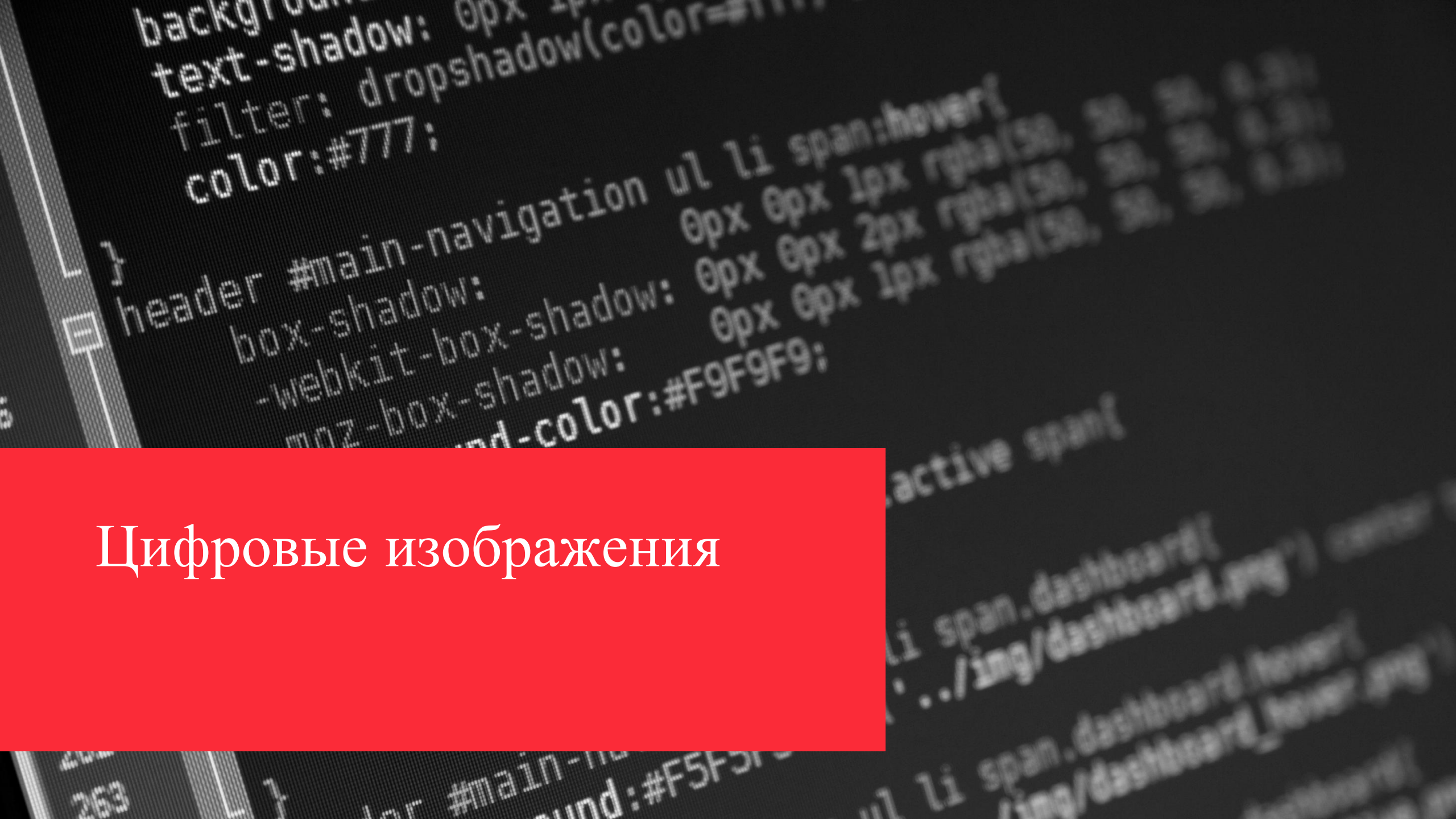
Базовые понятия обработки изображений

Андрей Савченко

Профессор НИУ ВШЭ-Нижний Новгород

Опрос

https://forms.gle/jdUkQxAsY4oxquUs7

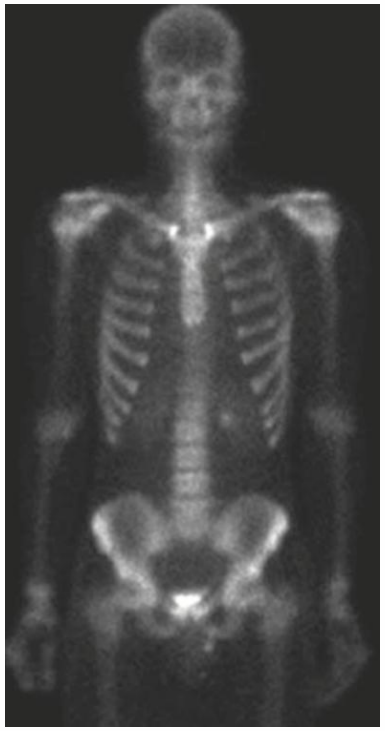


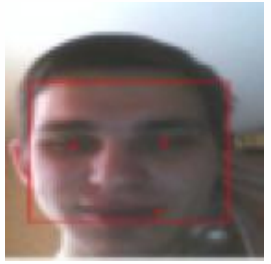
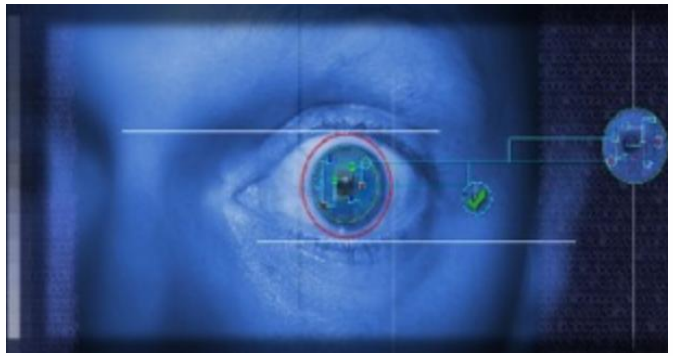
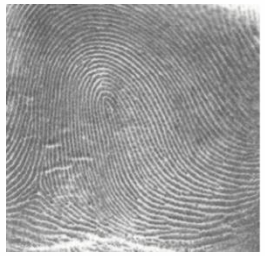
Цифровые изображения

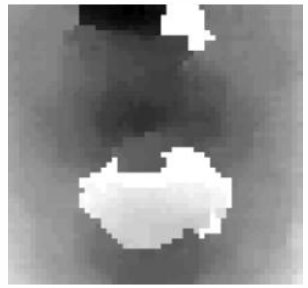
Примеры изображений

Изображение оптическое — картина, получаемая в результате действия оптической системы на лучи, испускаемые объектом, и воспроизводящая контуры и детали объекта.

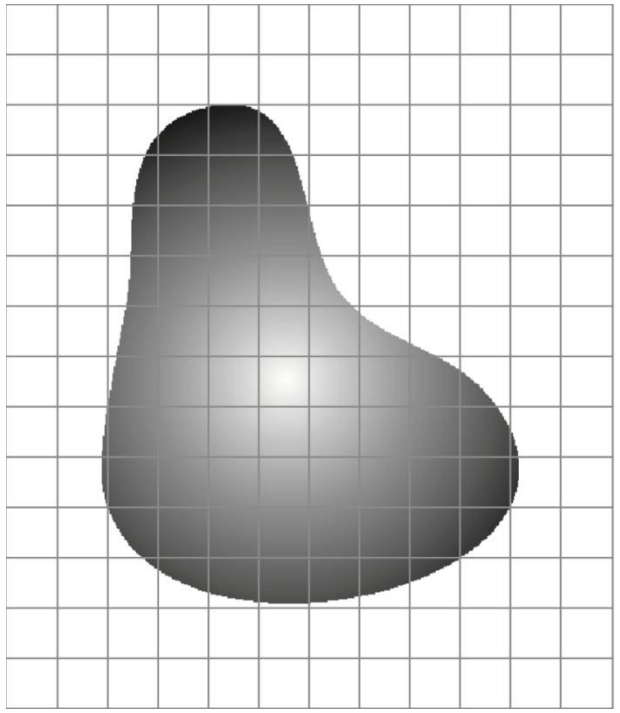


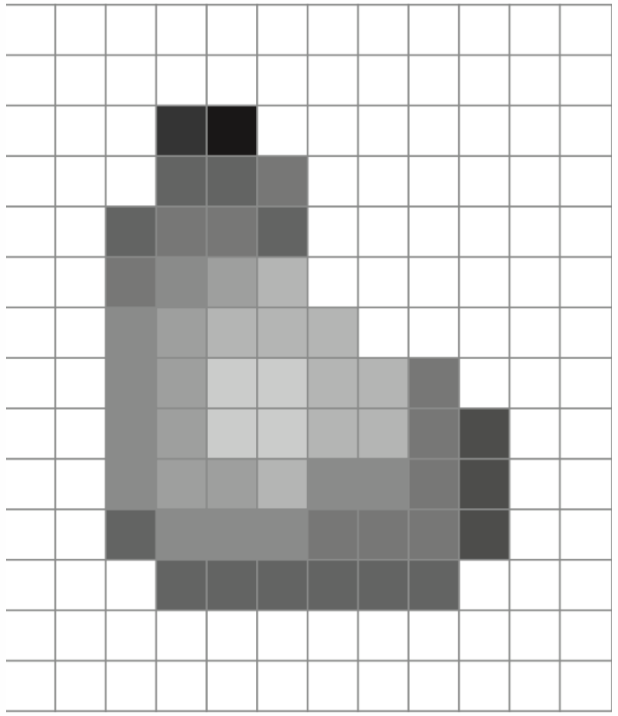


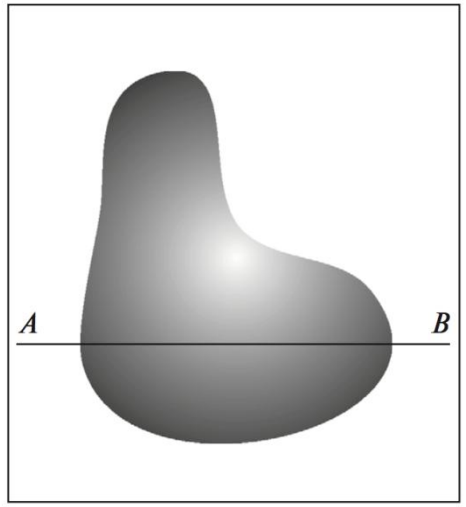





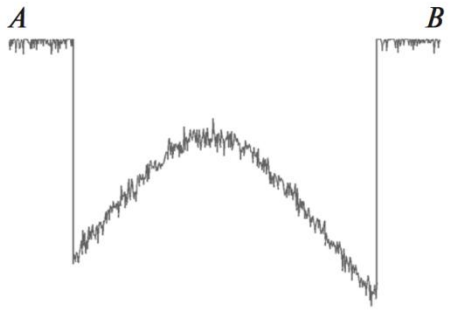



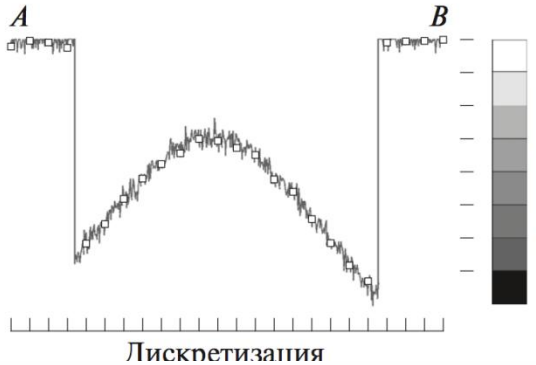
Представления цифровых изображений. Дискретизация и квантование

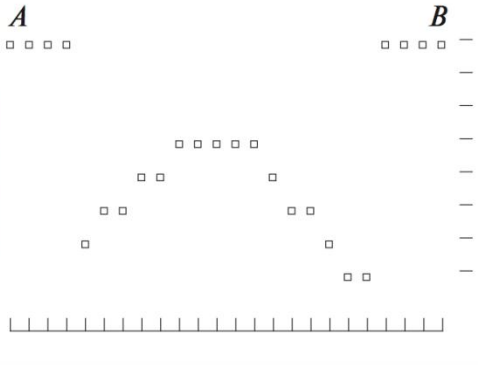












Гонсалес, Вудс «Цифровая обработка изображений»

Квантование (число бит на пиксель)

2 levels - binary

4 levels

256 levels – 1 byte

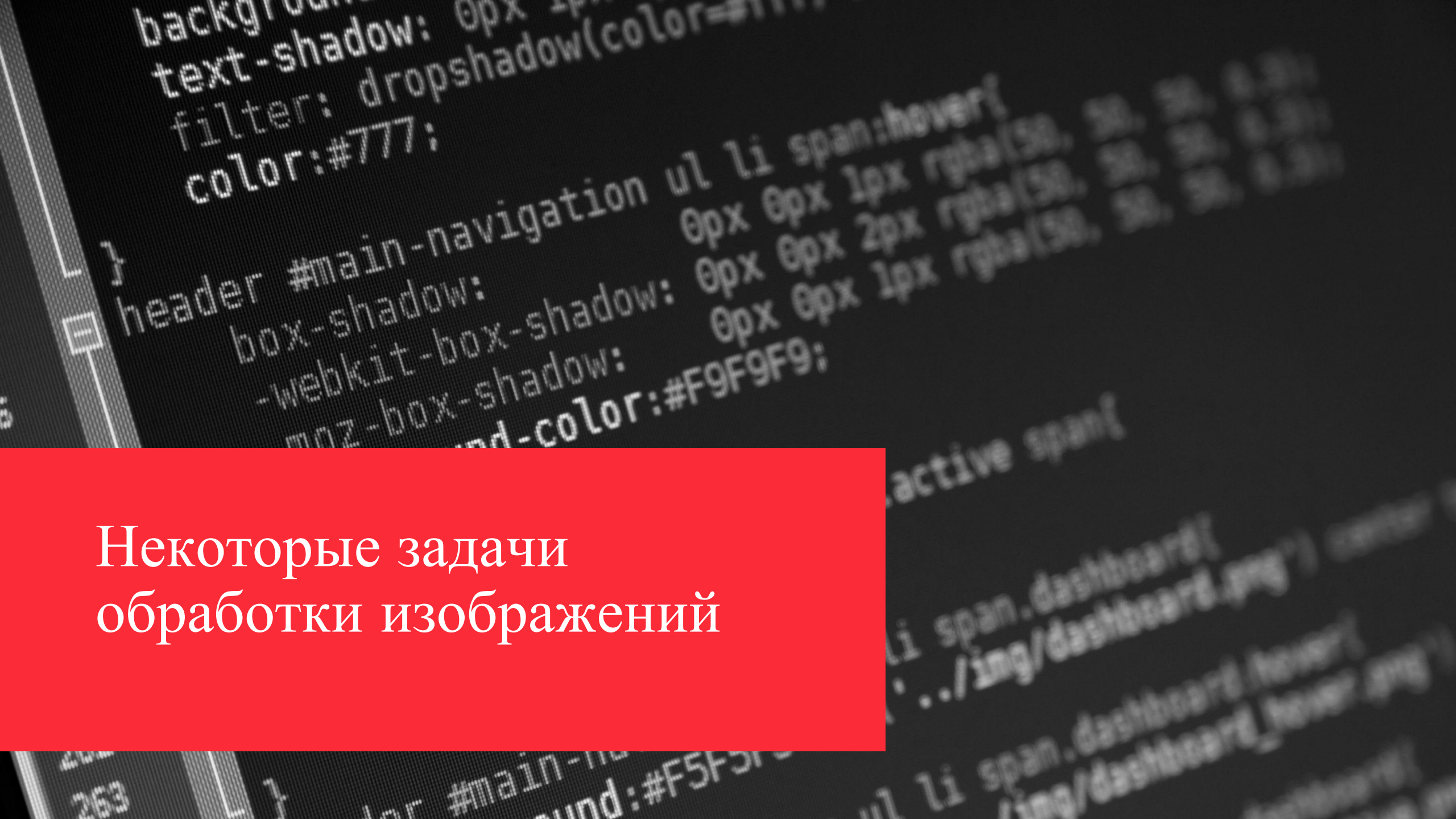
http://www.robots.ox.ac.uk/~az/lectures/ia/lect1.pdf

Разрешение

64x64

256x256

https://link.springer.com/article/10.1186/s13640-019-0436-5/figures/1

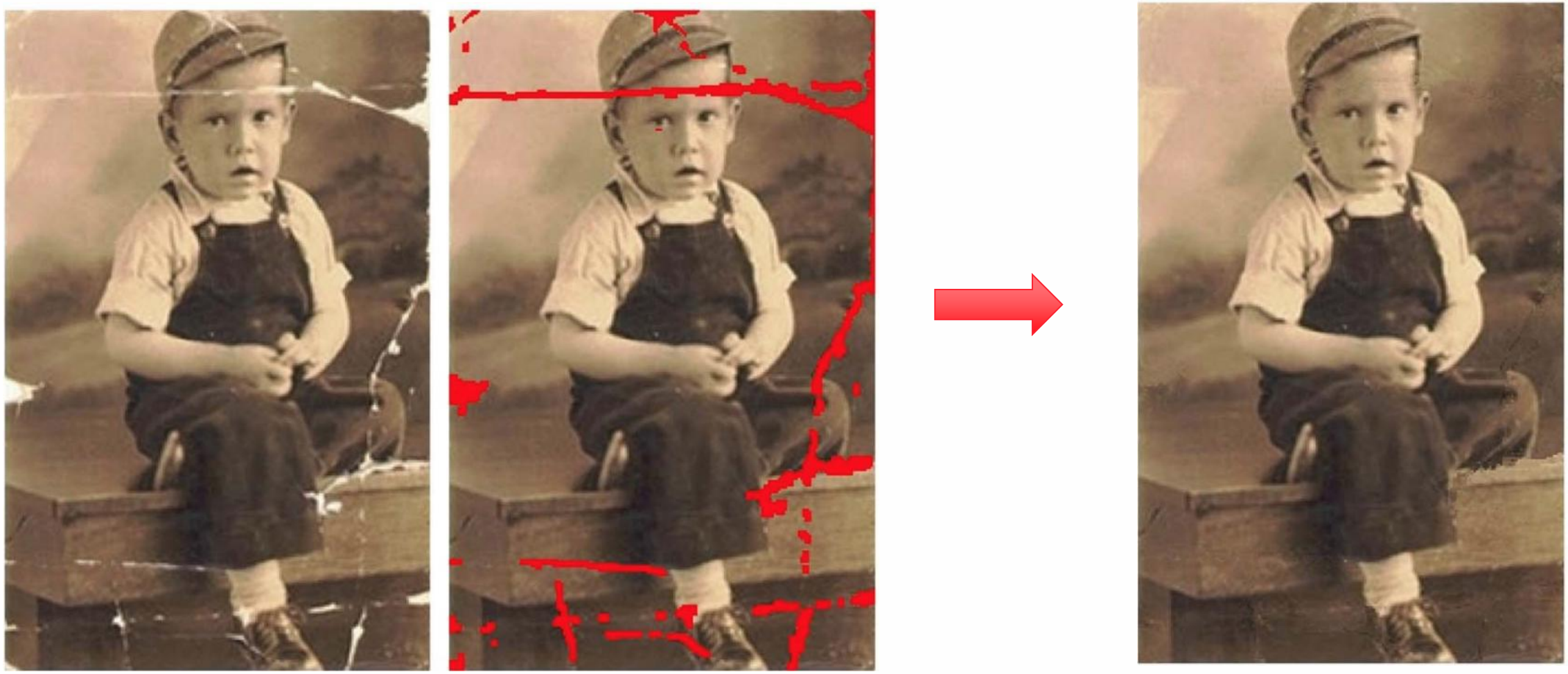


Некоторые задачи обработки изображений

Шумоподавление

https://www.mathworks.com/help/wavelet/examples/denoising-signals-and-images.html

Ретуширование (inpainting)

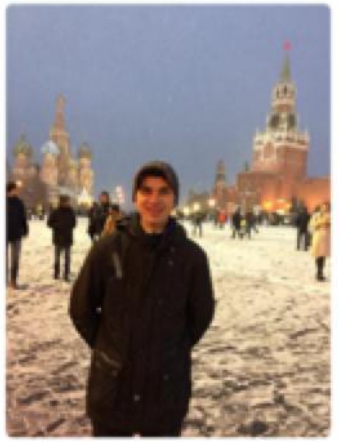


http://www.robots.ox.ac.uk/~az/lectures/ia/lect1.pdf

Распознавание изображений

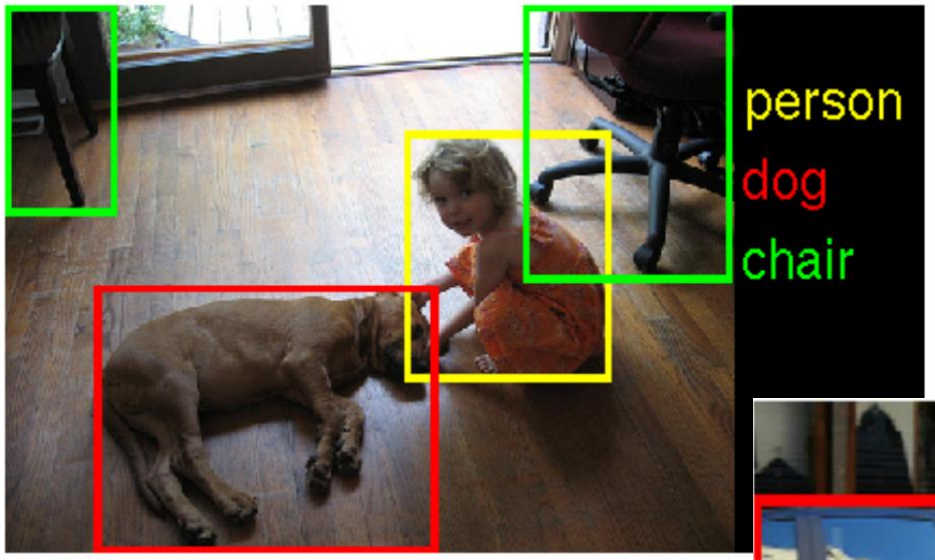
(a) Siberian husky

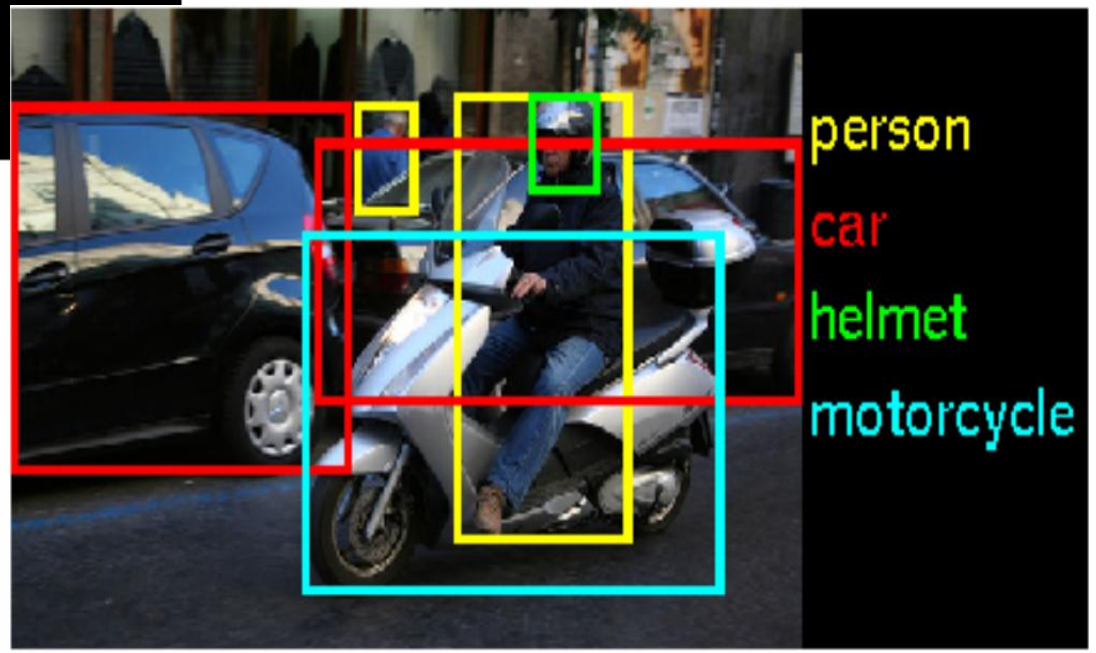
(b) Eskimo dog

ImageIdentify[]

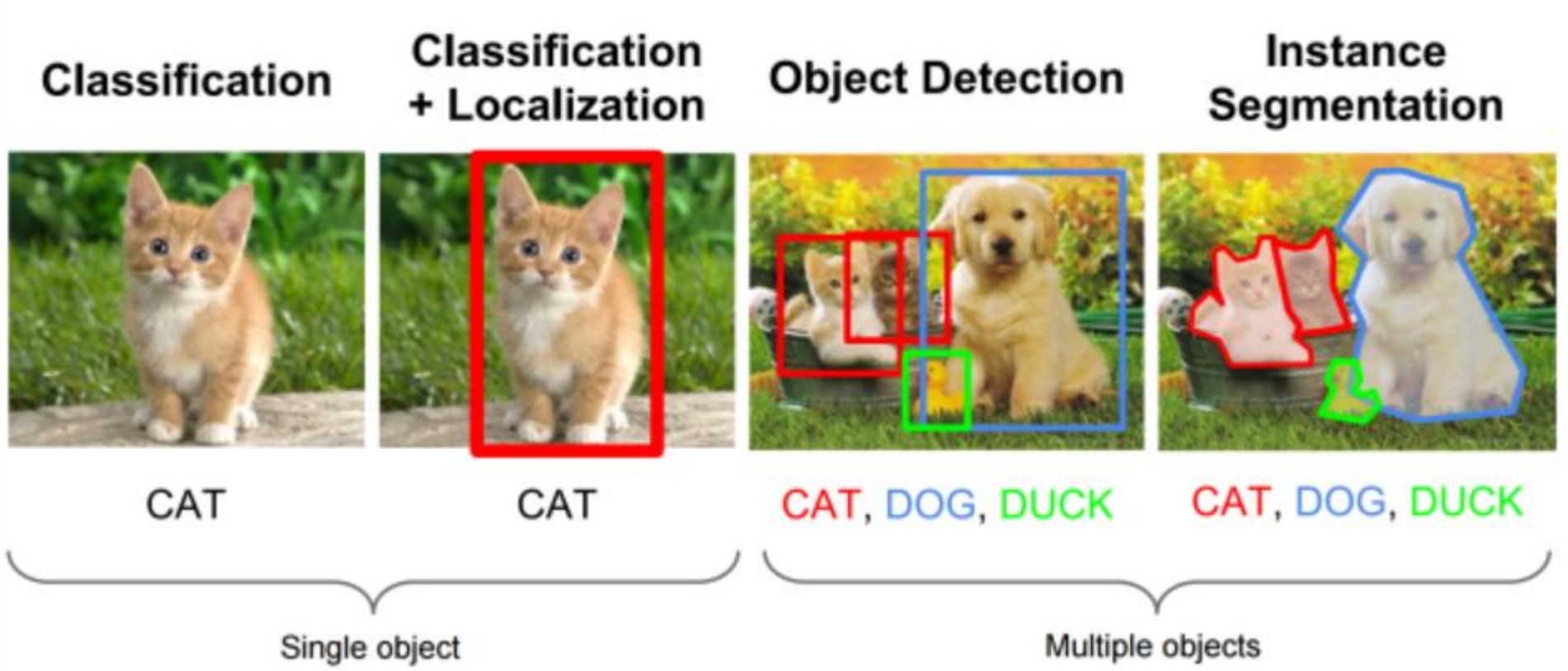
bastion

Детектирование объектов на изображениях





Классификация, детектирование, локализация...



https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/object_localization_and_detection.html

Семантическая сегментация

Текстовое описание изображений

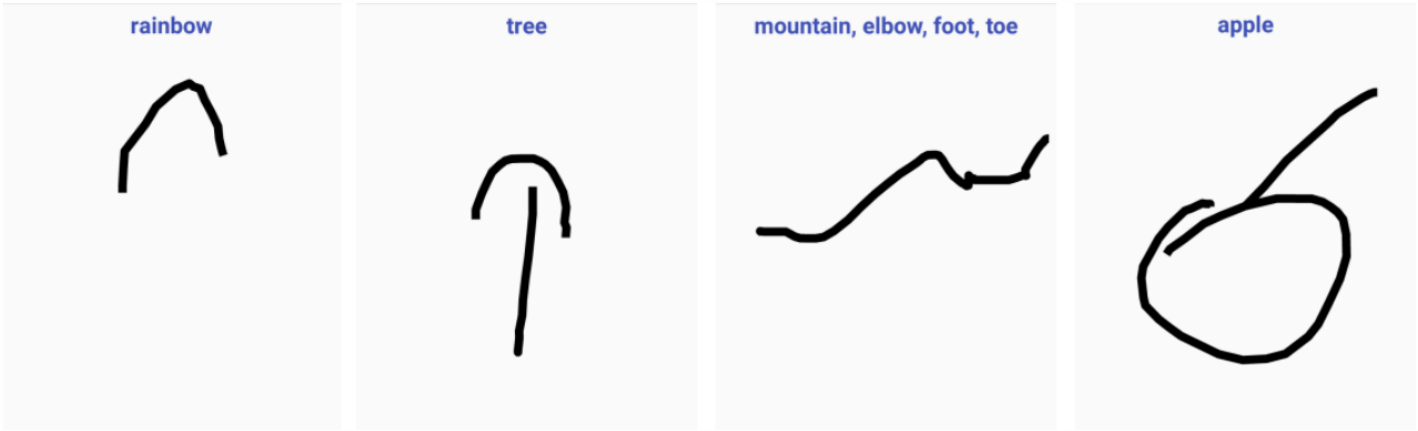
http://arxiv.org/abs/1411.4555 “Show and Tell: A Neural Image Caption Generator”

Генерация изображений. Перенос стиля

Prisma

Другие задачи

Распознавание рисунков



https://quickdraw.withgoogle.com/

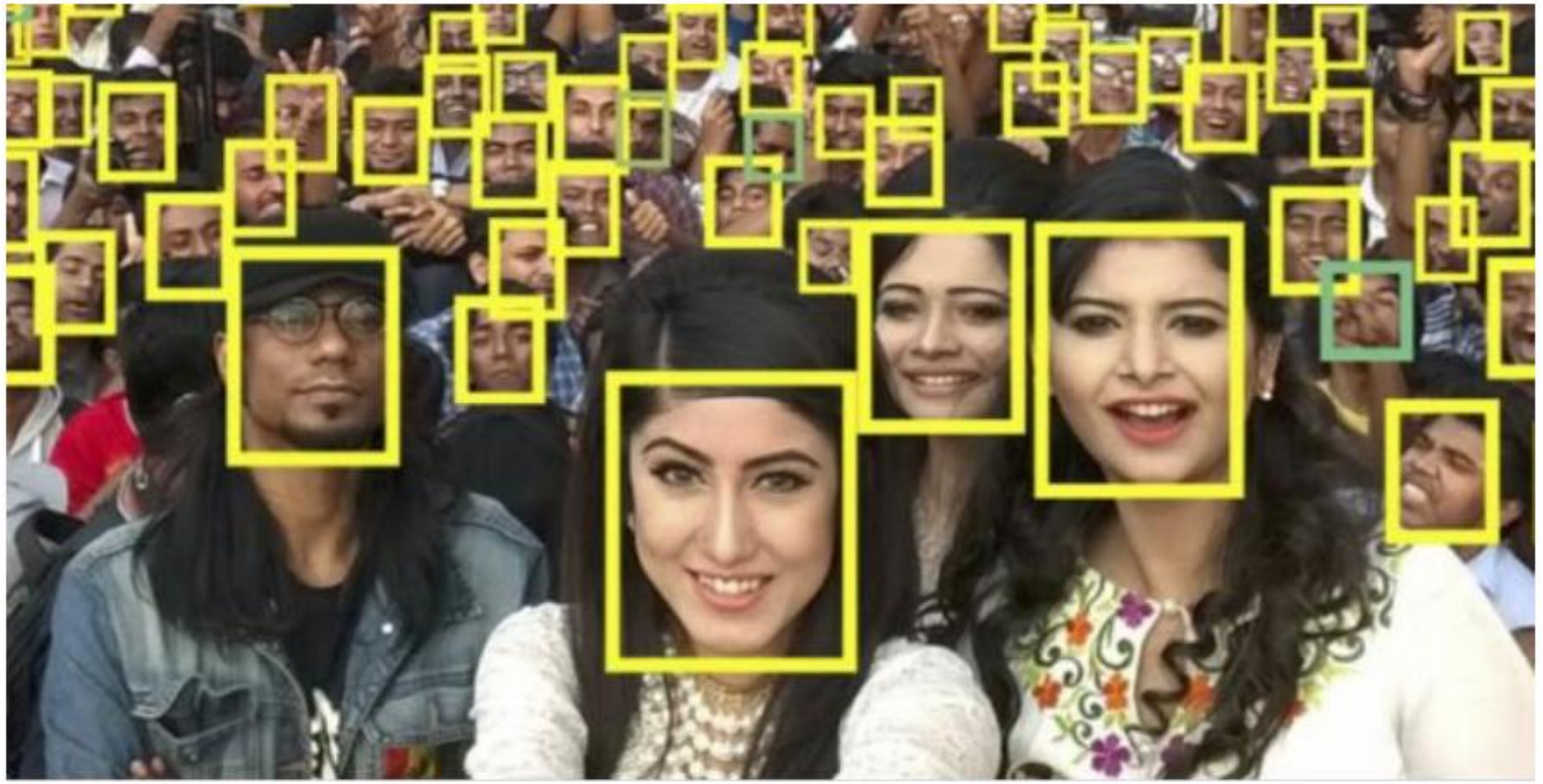
https://github.com/tensorflow/docs/blob/master/site/en/r1/tutorials/sequences/recurrent_quickdraw.md

MemNet, https://arxiv.org/abs/1708.02209

LaMem, http://memorability.csail.mit.edu/demo.html

Запоминаемость изображения

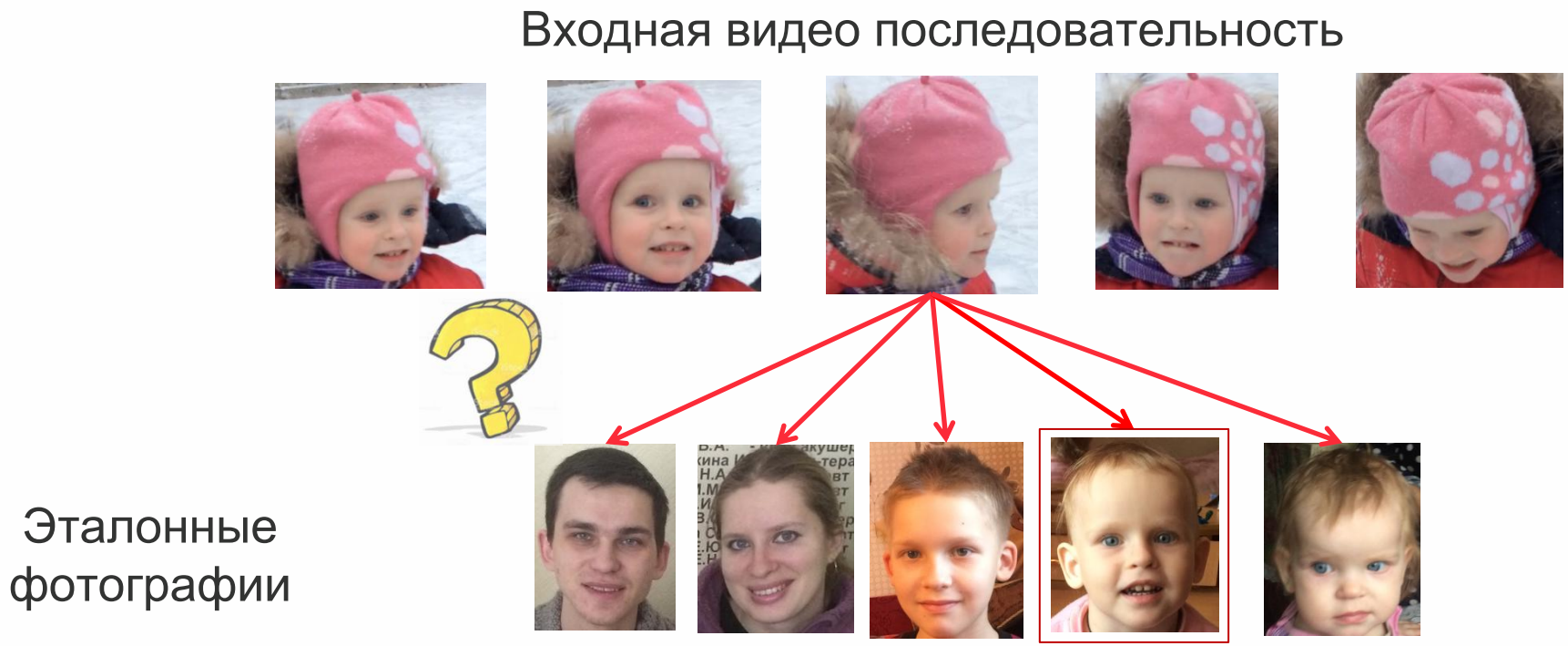
Обработка изображений лиц (1). Детектирование



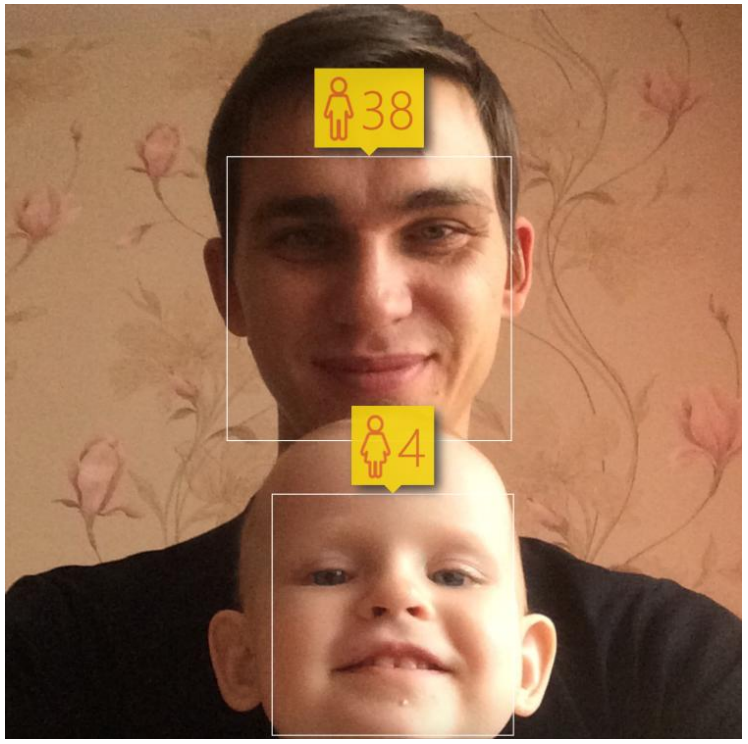
Обработка изображений лиц (2). Верификация

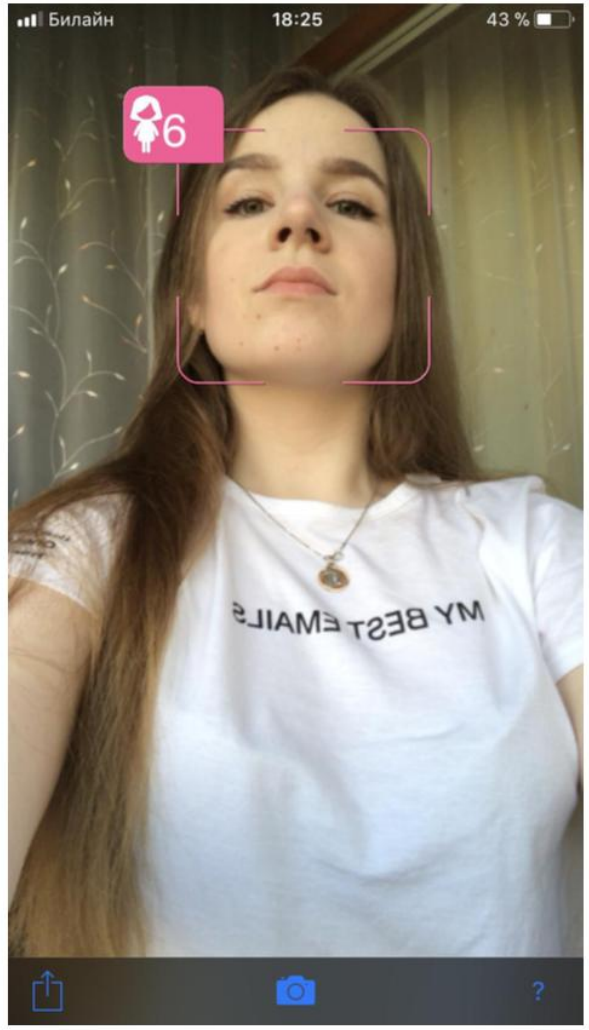
Kumar et al, TPAMI, 2011

Обработка изображений лиц (3). Идентификация (распознавание)



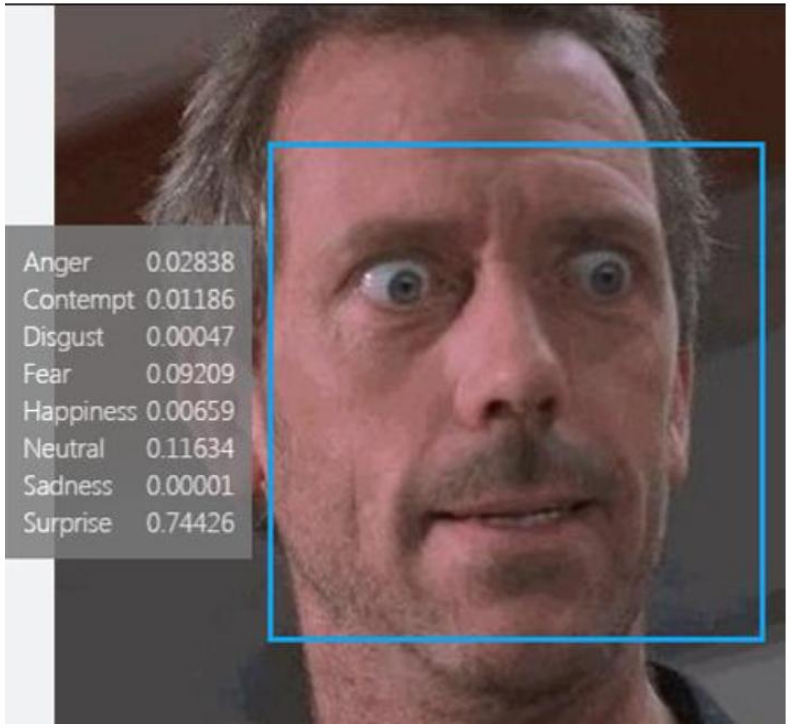
Обработка изображений лиц (4). Распознавание пола и возраста





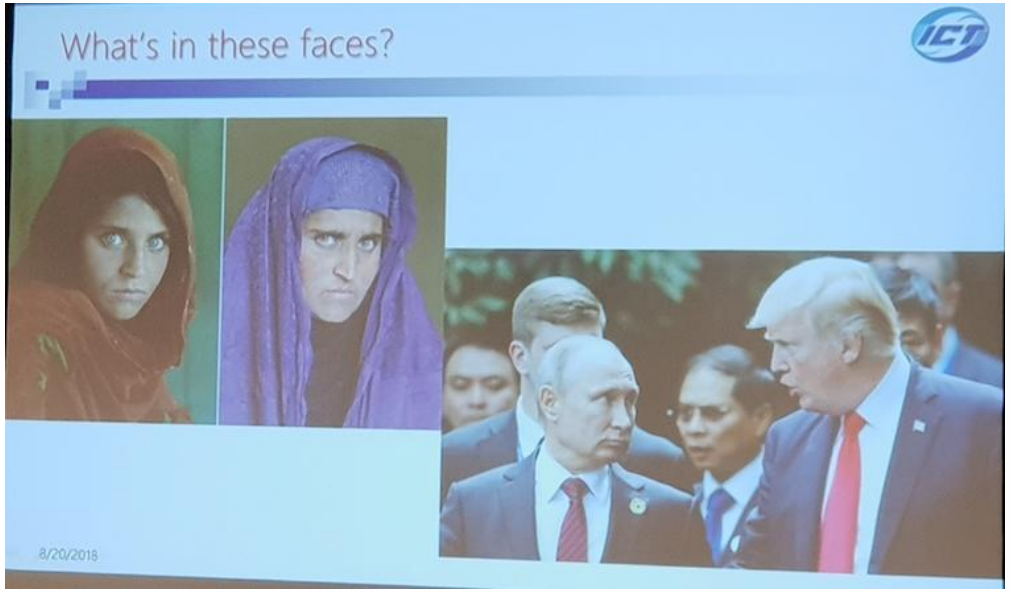
Luxand FaceSDK

Обработка изображений лиц (5). Распознавание эмоций



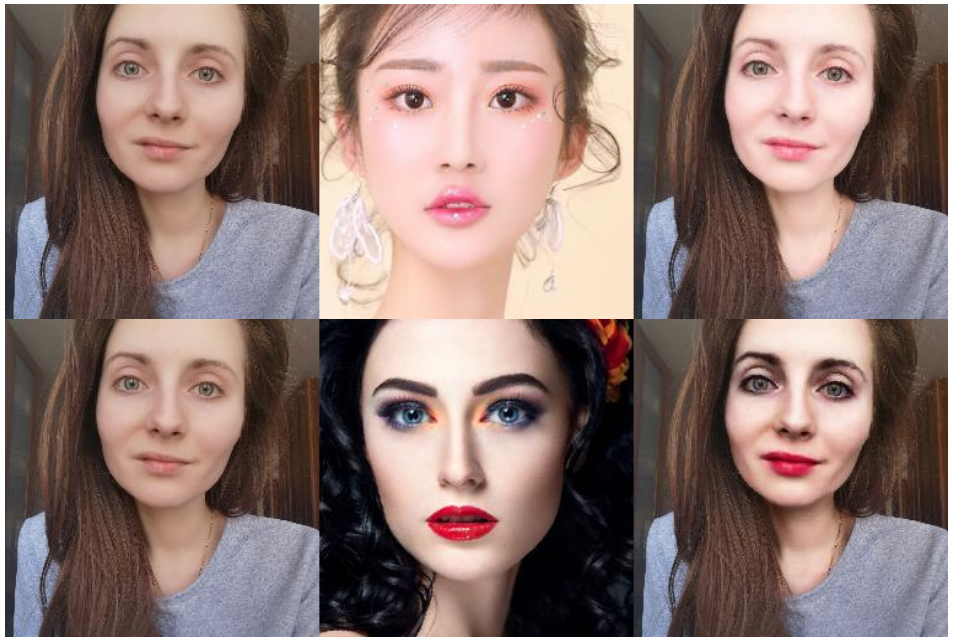
“positive” “neutral” “negative”

Xilin Chen: Challenges in facial expression understanding, ICPR18

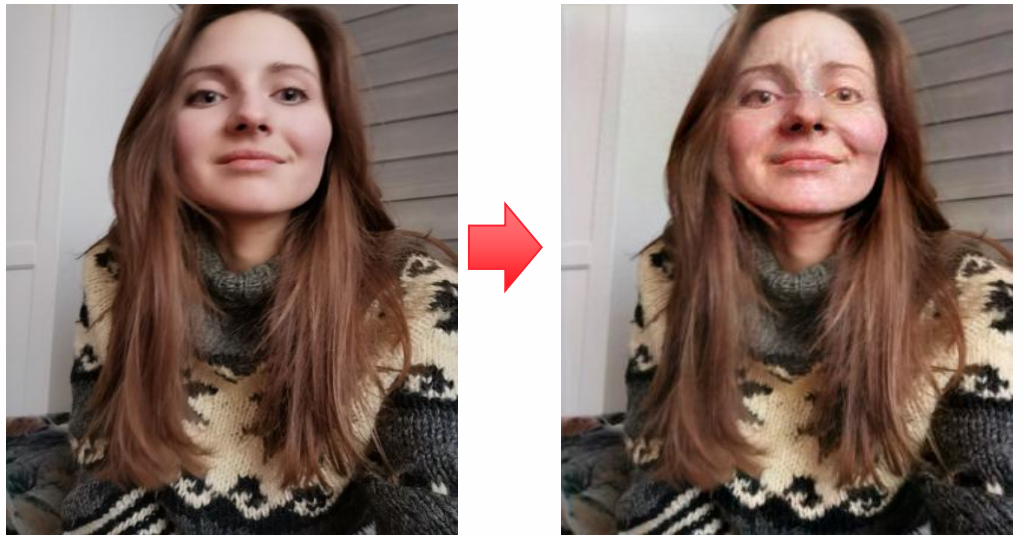


Обработка изображений лиц (6). Генерация

Перенос макияжа

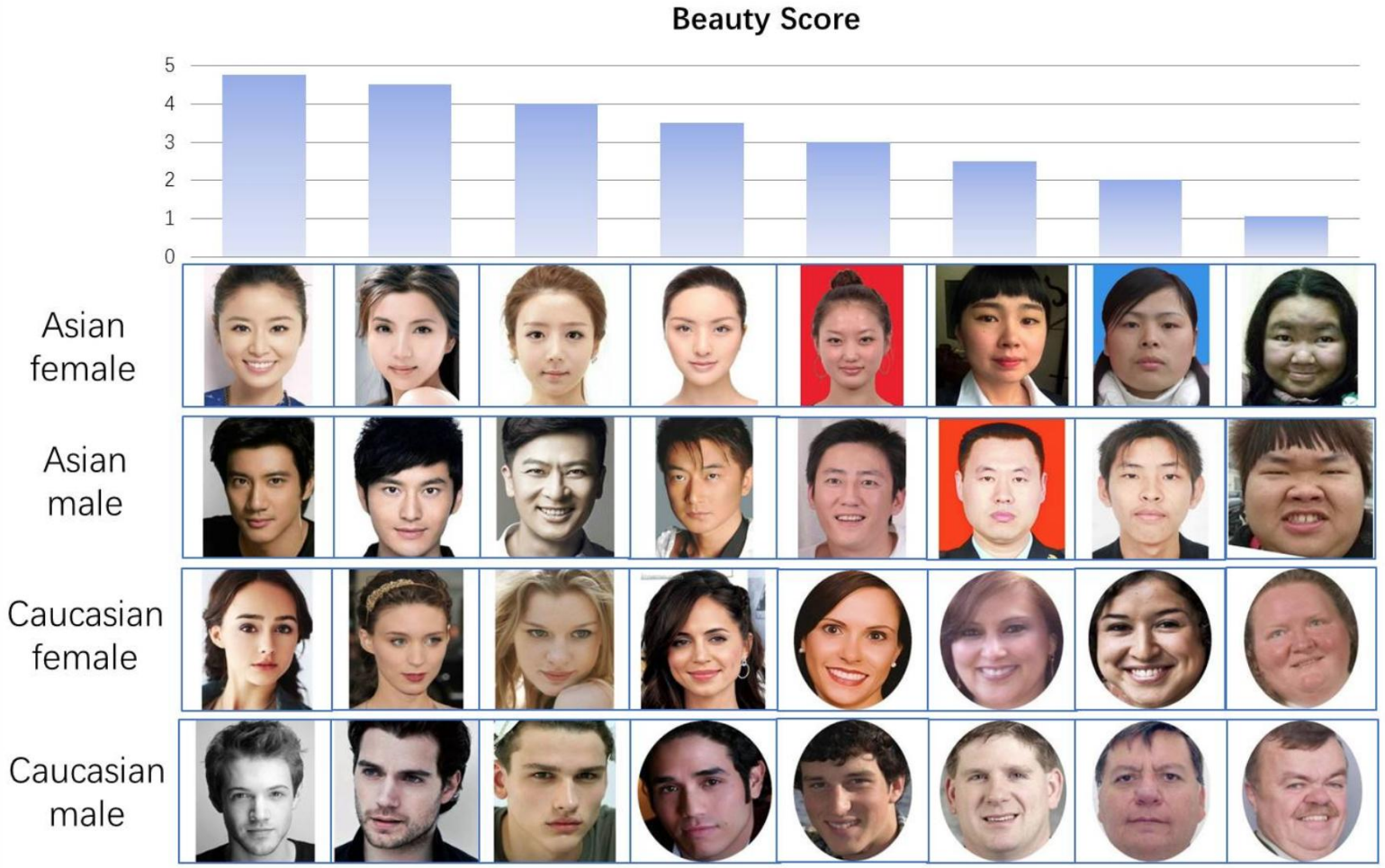


Старение



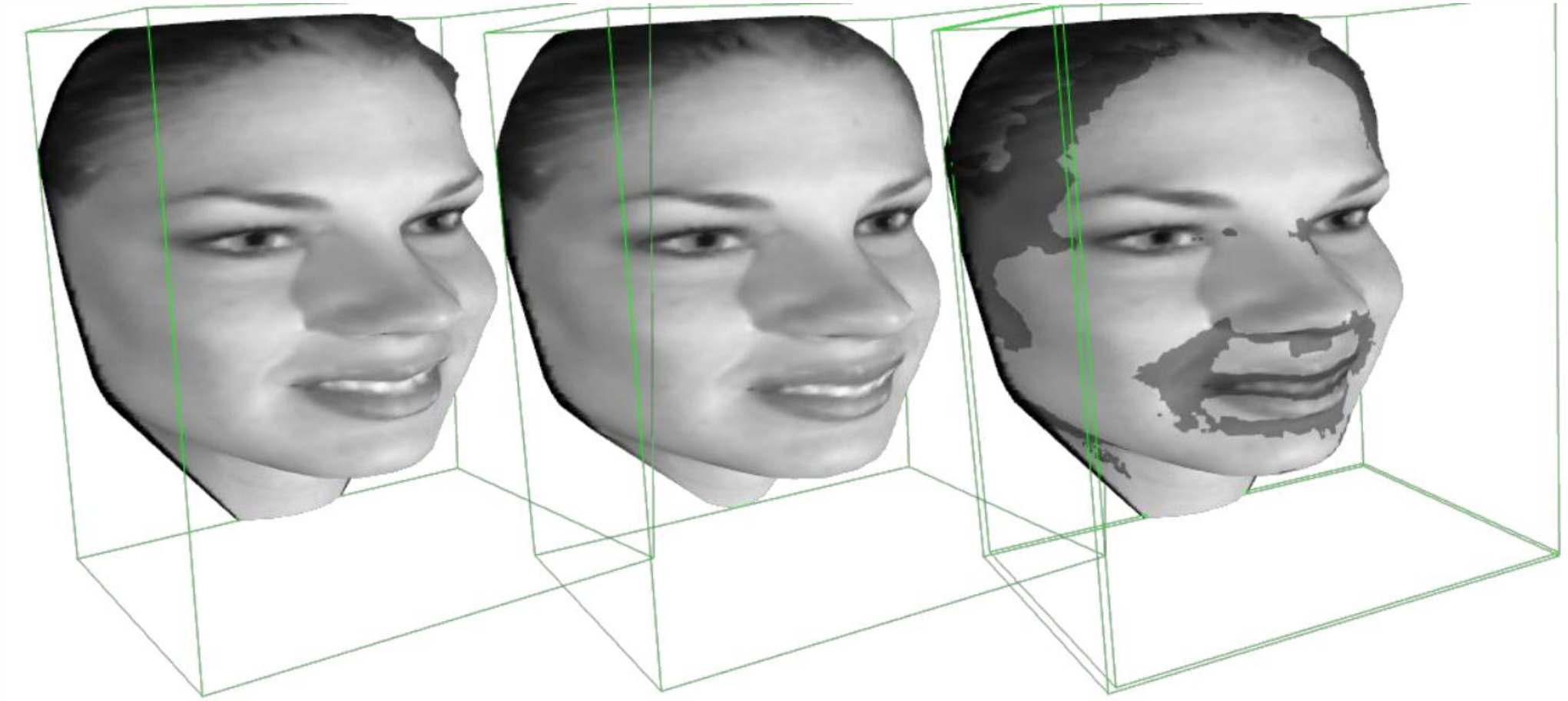
1. PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup, CVPR18
2. Соколова, 2020

Обработка изображений лиц (7). Оценка степени привлекательности



https://github.com/HCIILAB/SCUT-FBP5500-Database-Release

3D модели



[Рассадин, 2017]

Цветовые модели

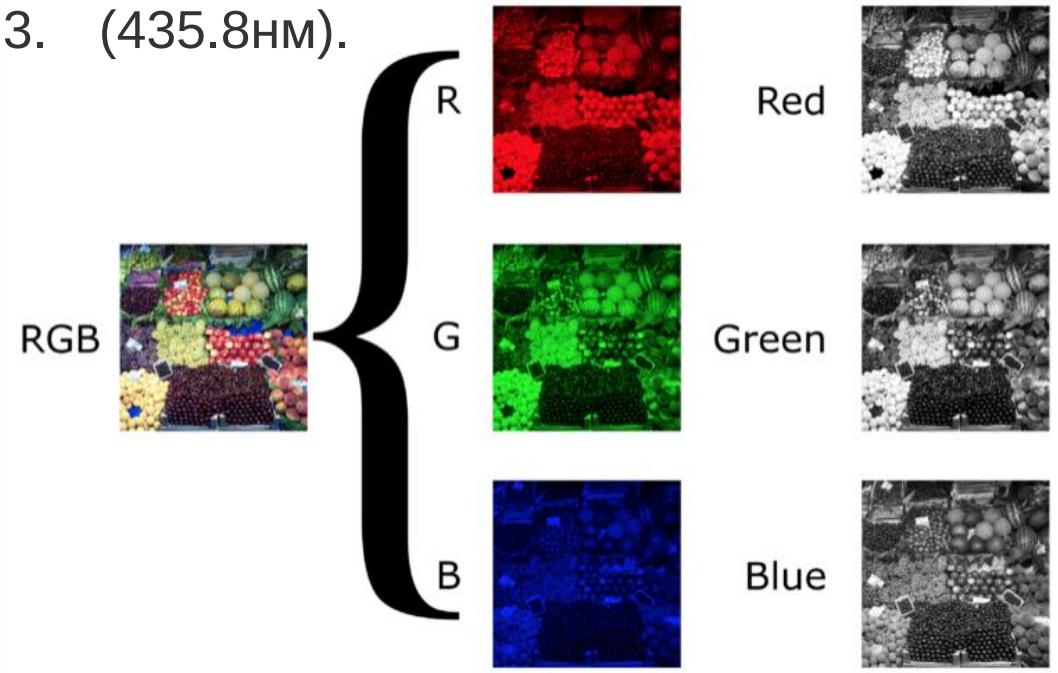
RGB (Red-Green-Blue): аддитивная модель

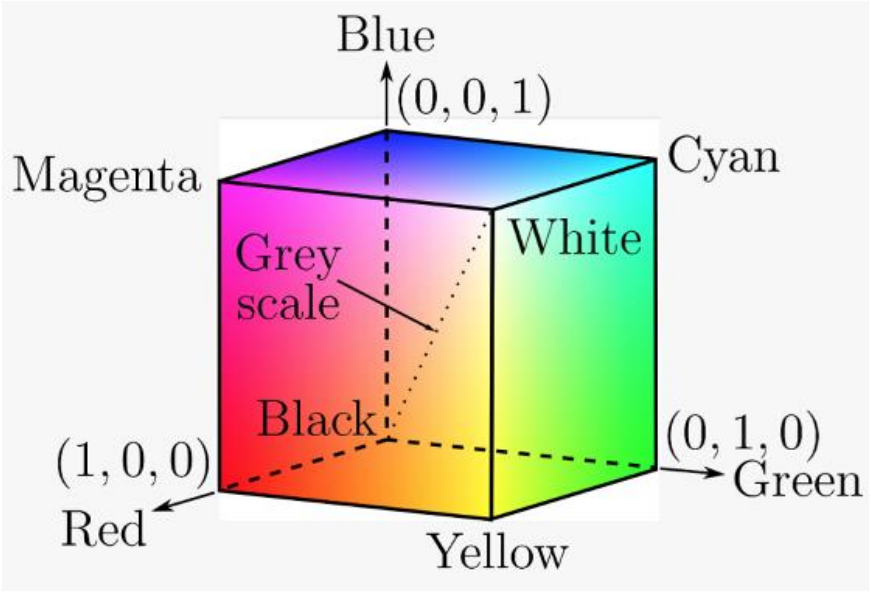
Трихроматическая теория:

Цвет=$\alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$

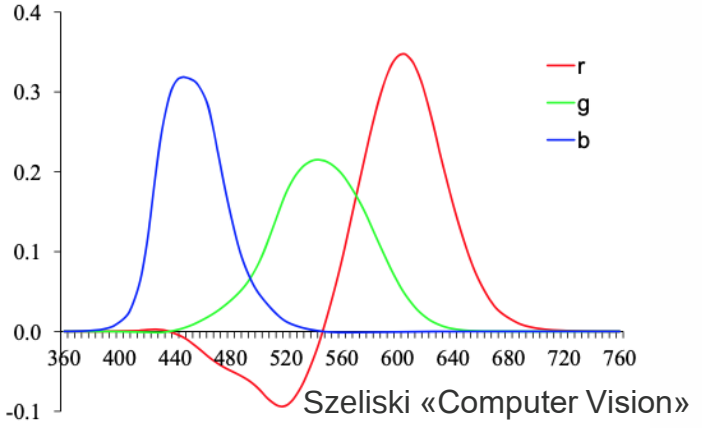
Commission Internationale d'Eclairage (CIE):

1. red (длина волны 700.0нм)
2. green (546.1нм)
3. (435.8нм).





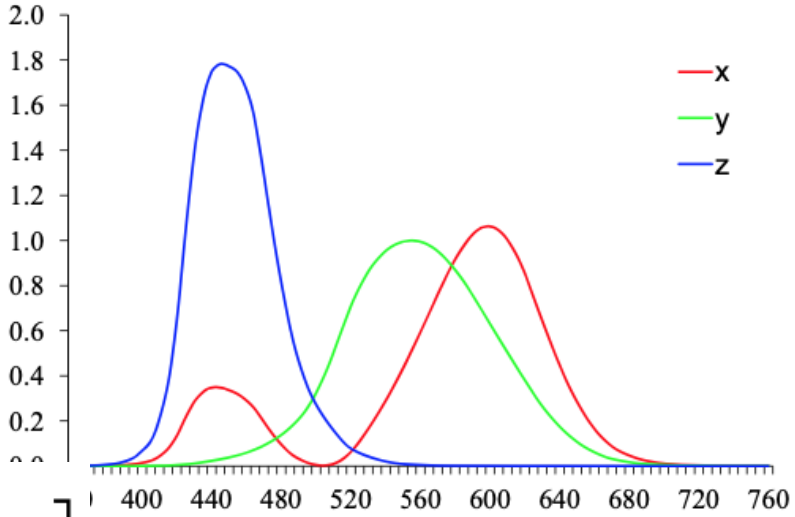
Функция цветового соответствия (усредненный наблюдатель)



Szeliski «Computer Vision»

XYZ

1. Аддитивная модель
2. Неотрицательная функция цветового соответствия
3. Покрывает весь видимый диапазон

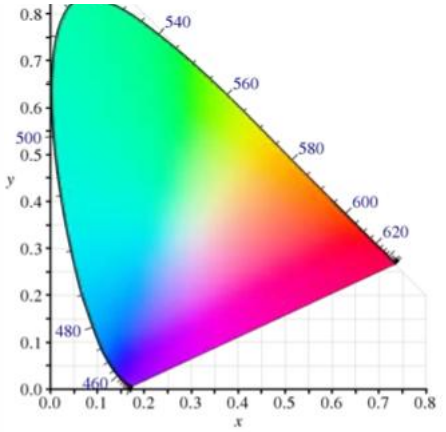


$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}$$

Luminance (яркость света)

Вариация - Yxy

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z},$$



Szeliski «Computer Vision»

CIELAB (L*a*b*)

Яркость воспринимается человеком логарифмически

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) \qquad a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \qquad b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

Y_n, Z_n – Y и Z компоненты белого цвета

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases}$$

(i) L*

(j) a*

(k) b*

Szeliski «Computer Vision»

YUV/YIQ

$$Y'_{601} = 0.299R' + 0.587G' + 0.114B'$$

$$Y'_{709} = 0.2125R' + 0.7154G' + 0.0721B'$$

YUV – для стандарта цветового ТВ PAL в Европе

YIQ – для стандарта цветового ТВ NTSC в США

$$U = 0.492111(B' - Y')$$

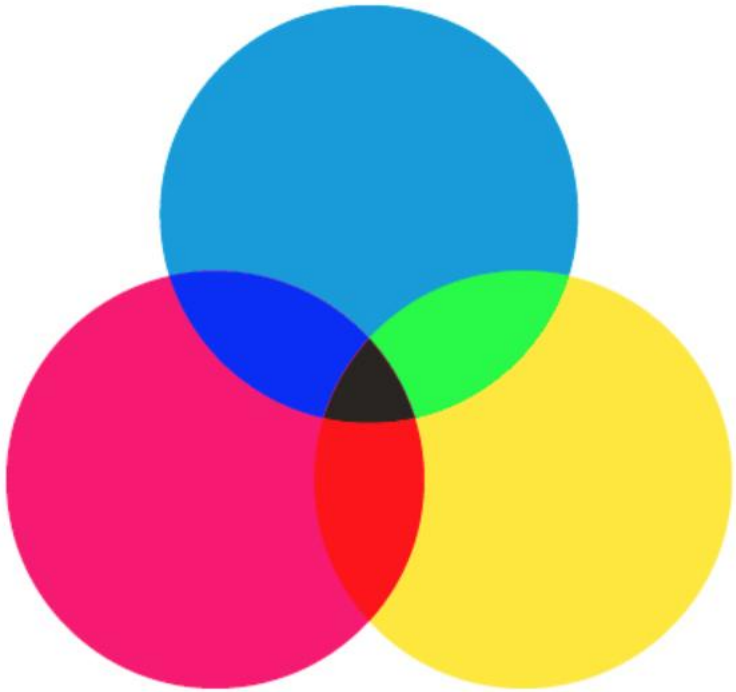
$$V = 0.877283(R' - Y')$$

поворот на 33 градуса

I = 0.596 R - 0.274 G - 0.322 B
Q = 0.211 R - 0.522 G + 0.311 B

CMY(K): субтрактивная модель

1. C=W(hite)-R
2. M=W-G
3. Y=W-B

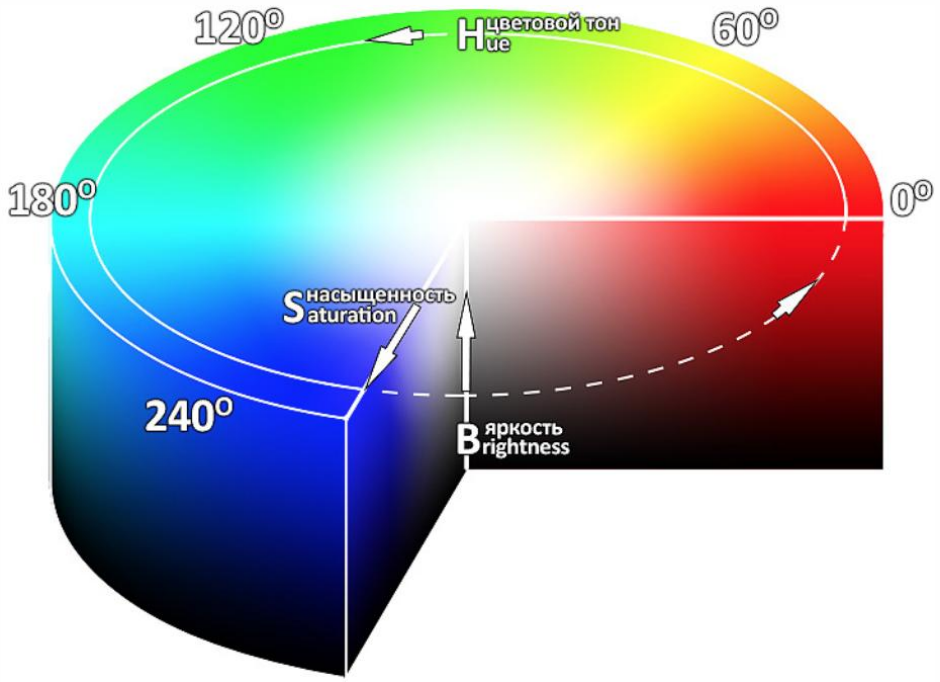


В теории при смешении трёх цветов должен получиться чёрный цвет. В реальности из-за примесей в красках получается грязно-коричневый.

Вводится чёрный цвет (К) для получения тёмных оттенков и непосредственно самого чёрного.

https://accent.su/blog/razlichiya-tsvetovyh-modelej-rgb-cmyk-hsb/

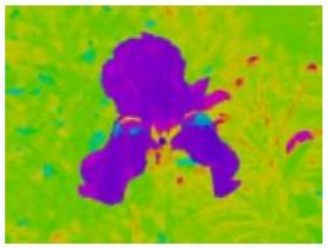
HSV (HSB): Hue-saturation-value(brightness)



$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$



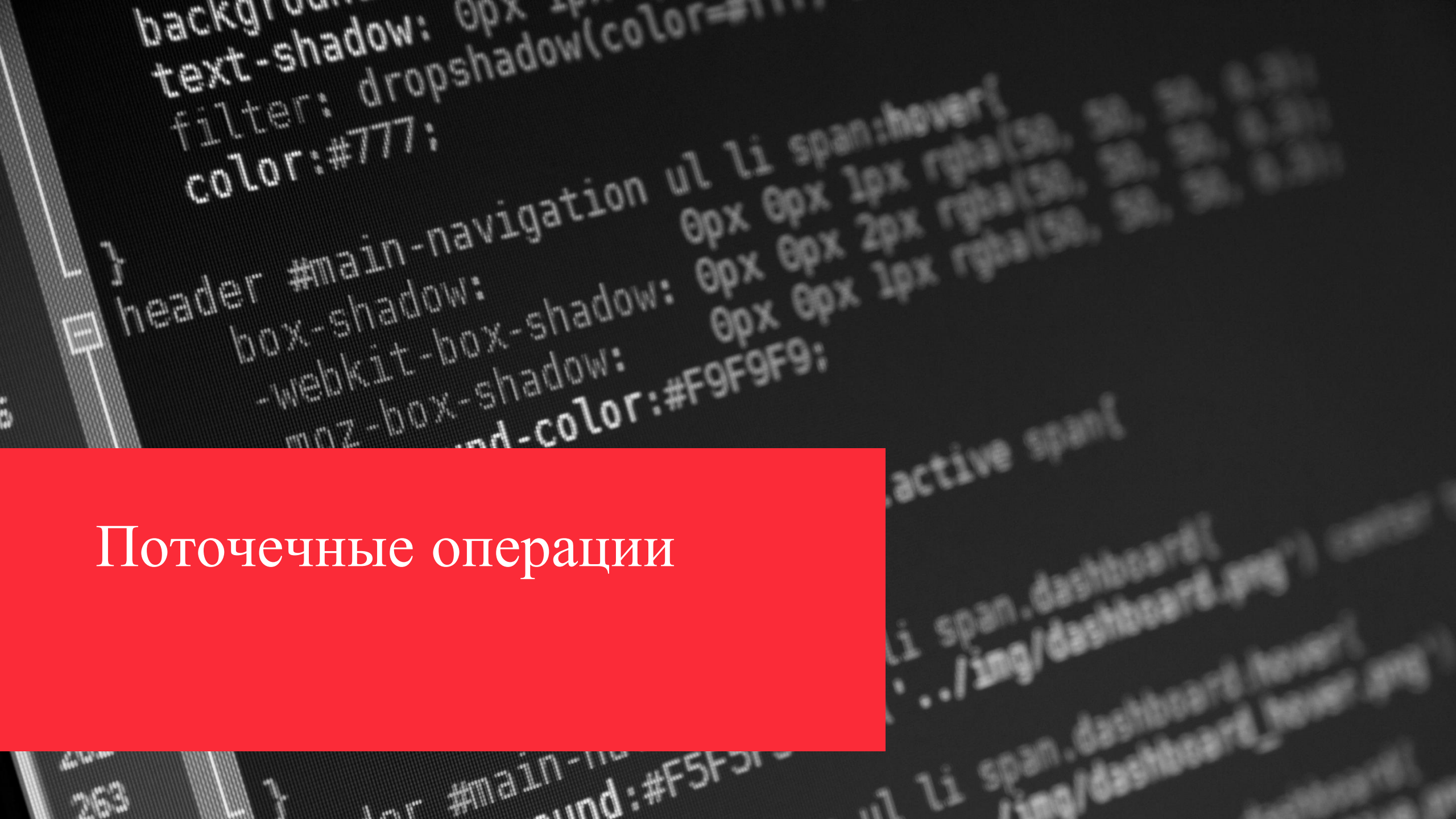
(l) H

(m) S

(n) V

https://accent.su/blog/razlichiya-tsvetovyh-modelej-rgb-cmyk-hsb/

Szeliski «Computer Vision»


Поточечные операции

OpenCV

- Open Source Computer Vision Library
- BSD license
- C++, C, Python, Java
- Windows, Linux, Mac OS, iOS and Android

Intel 169,22
Компания

9 ноября 2016 в 14:52

Itseez, дважды Intel Company

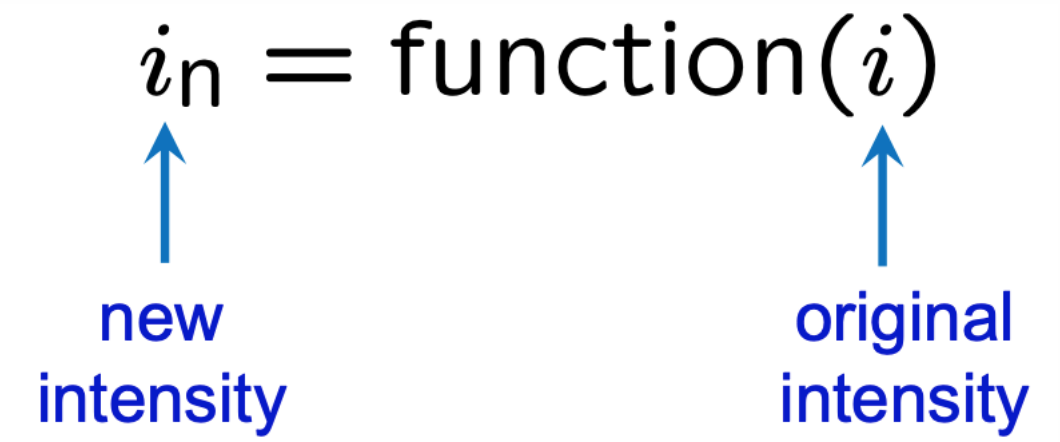
Программирование*, Обработка изображений*, Блог компании Intel

- Официальные релизы: https://opencv.org/releases/
- Nightly build (+contrib modules): https://pullrequest.opencv.org/buildbot/export/opencv_releases/master-contrib_pack-contrib-android/
- Вручную:

1. Clone opencv/opencv-contrib
2. Создать папку (build_android) внутри репозитория opencv и запустить компиляцию

```
python ..\platforms\android\build_sdk.py --sdk_path=D:/Android-Sdk --
    extra_modules_path=D:\src_code\opencv\opencv_contrib/modules --no_samples_build
```

Поточеченые (попиксельные) операции



http://www.robots.ox.ac.uk/~az/lectures/ia/lect1.pdf

Преобразование формата. Полутоновые изображения

OpenCV: cvtColor(), RGB2GRAY (CCIR601)

$$Y' = 0.299R' + 0.587G' + 0.114B'$$

https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor

Нормировка

Приведение всех пикселей к нулевому среднему и единичной дисперсии

$$\mu = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} p_{ij}}{IJ}$$

$$\sigma^2 = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} (p_{ij}-\mu)^2}{IJ}$$

$$x_{ij} = \frac{p_{ij}-\mu}{\sigma}$$

OpenCV: normalize()

normType=NORM_INF, NORM_L1, or NORM_L2

$$\|\mathtt{dst}\|_{L_p} = \mathtt{alpha}$$

normType=NORM_MINMAX

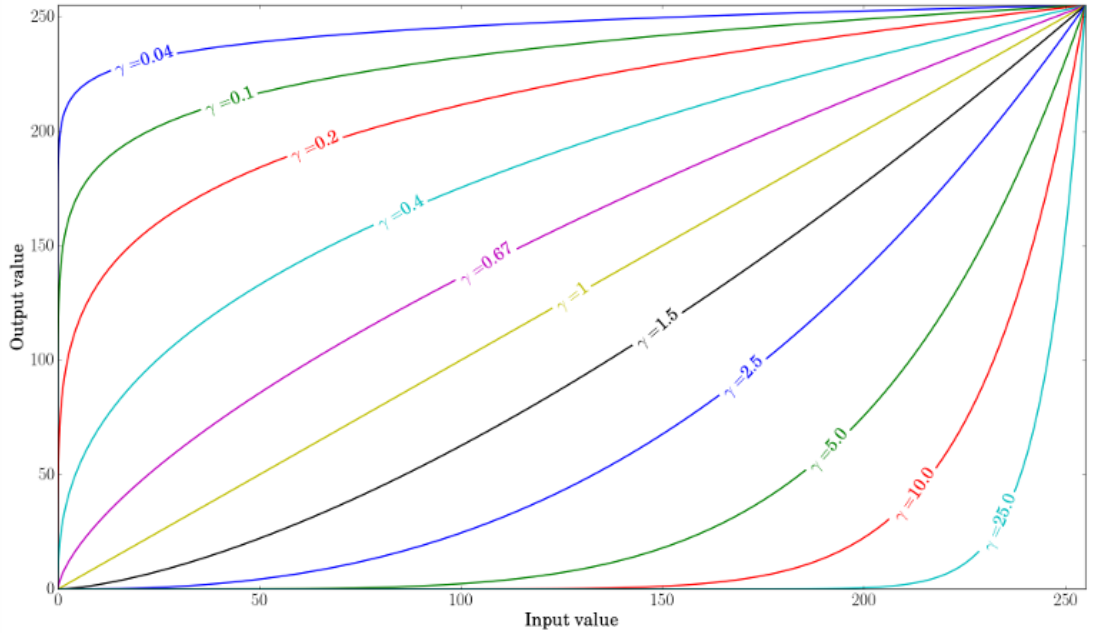
$$\min_I \mathtt{dst}(I) = \mathtt{alpha}, \ \max_I \mathtt{dst}(I) = \mathtt{beta}$$

Prince «Computer vision: models, learning and inference»

Гамма-коррекция

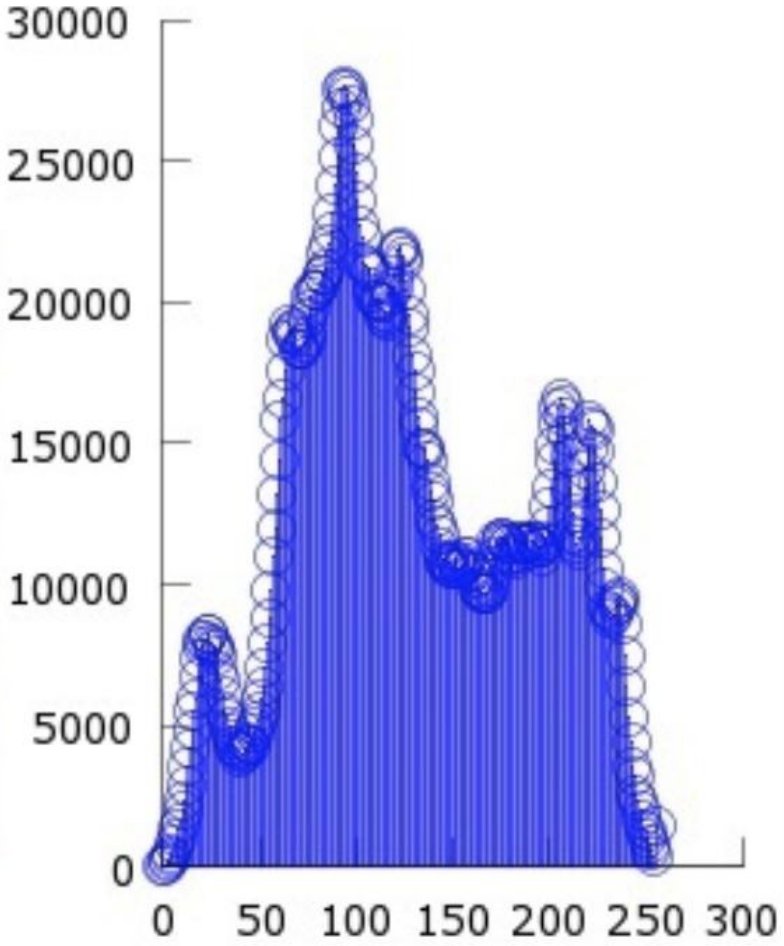
В ранних ЧБ ТВ сигнал нелинейно зависел от входного напряжения

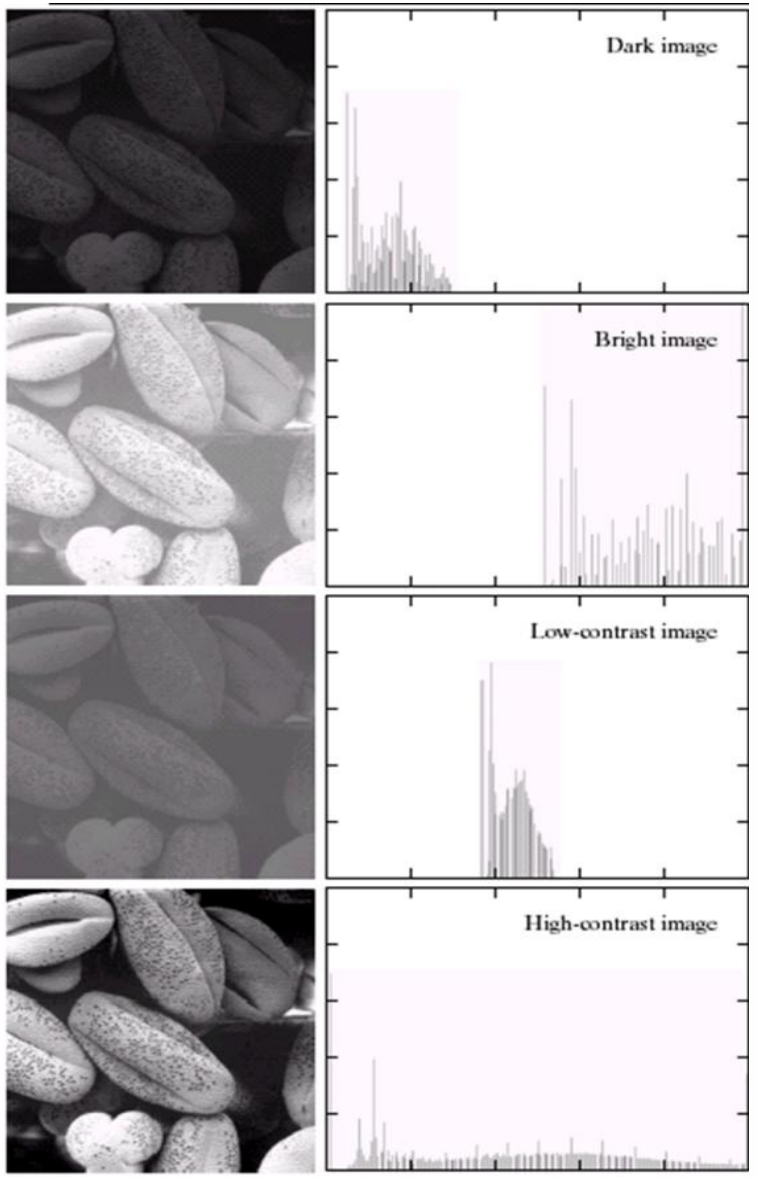
$$O = \left(\frac{I}{255}\right)^{\gamma} \times 255$$



https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html
https://www.pyimagesearch.com/2015/10/05/opencv-gamma-correction/

Гистограмма полутонового изображения





OpenCV: calcHist()

Muhammad «OpenCV Android Programming By Example»

http://www.robots.ox.ac.uk/~az/lectures/ia/lect1.pdf

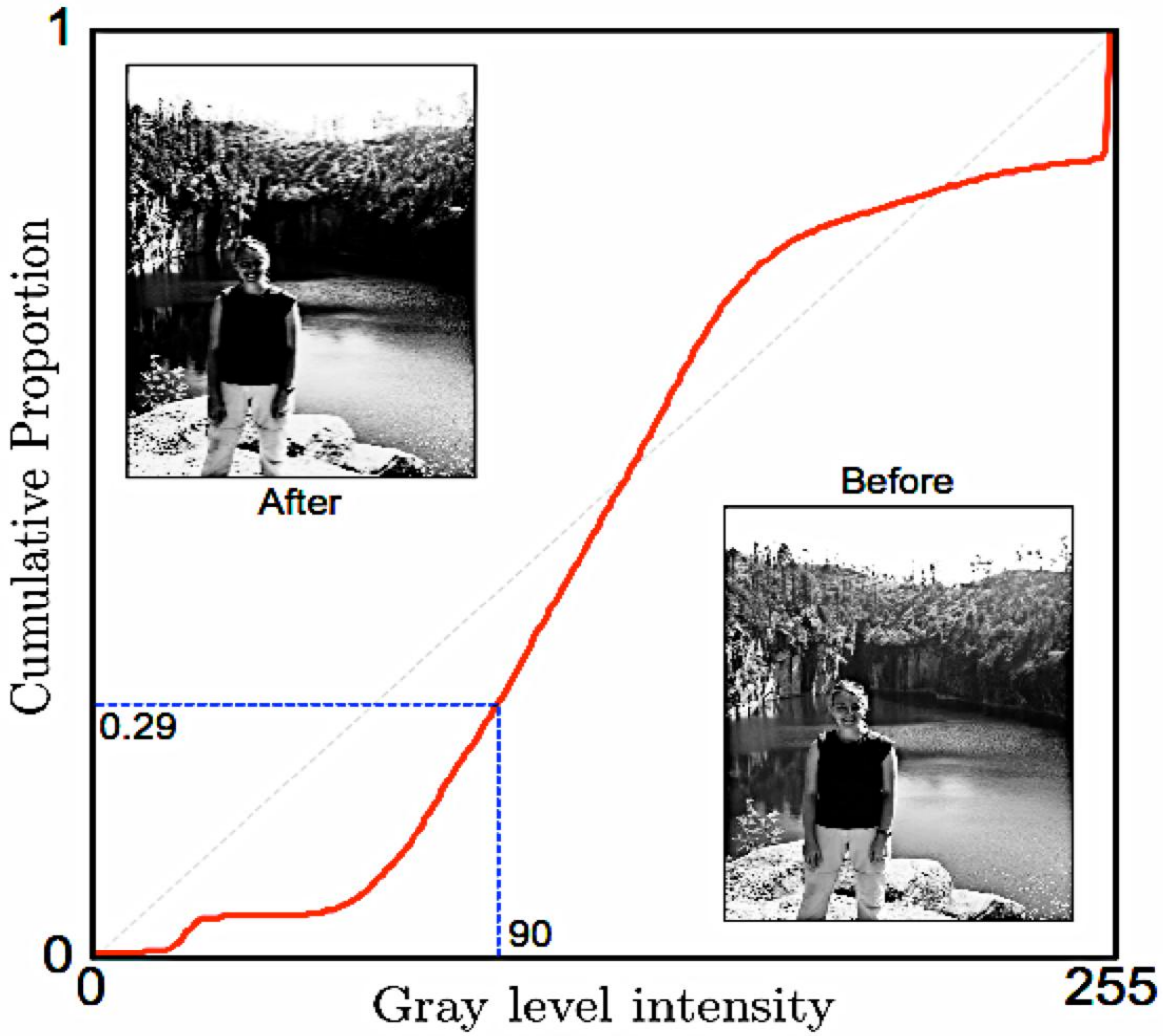
Выравнивание (эквализация) гистограммы

$$h_k = \sum_{i=1}^{I}\sum_{j=1}^{J}\delta[p_{ij} - k]$$

$$c_k = \frac{\sum_{l=1}^{k} h_l}{IJ}$$

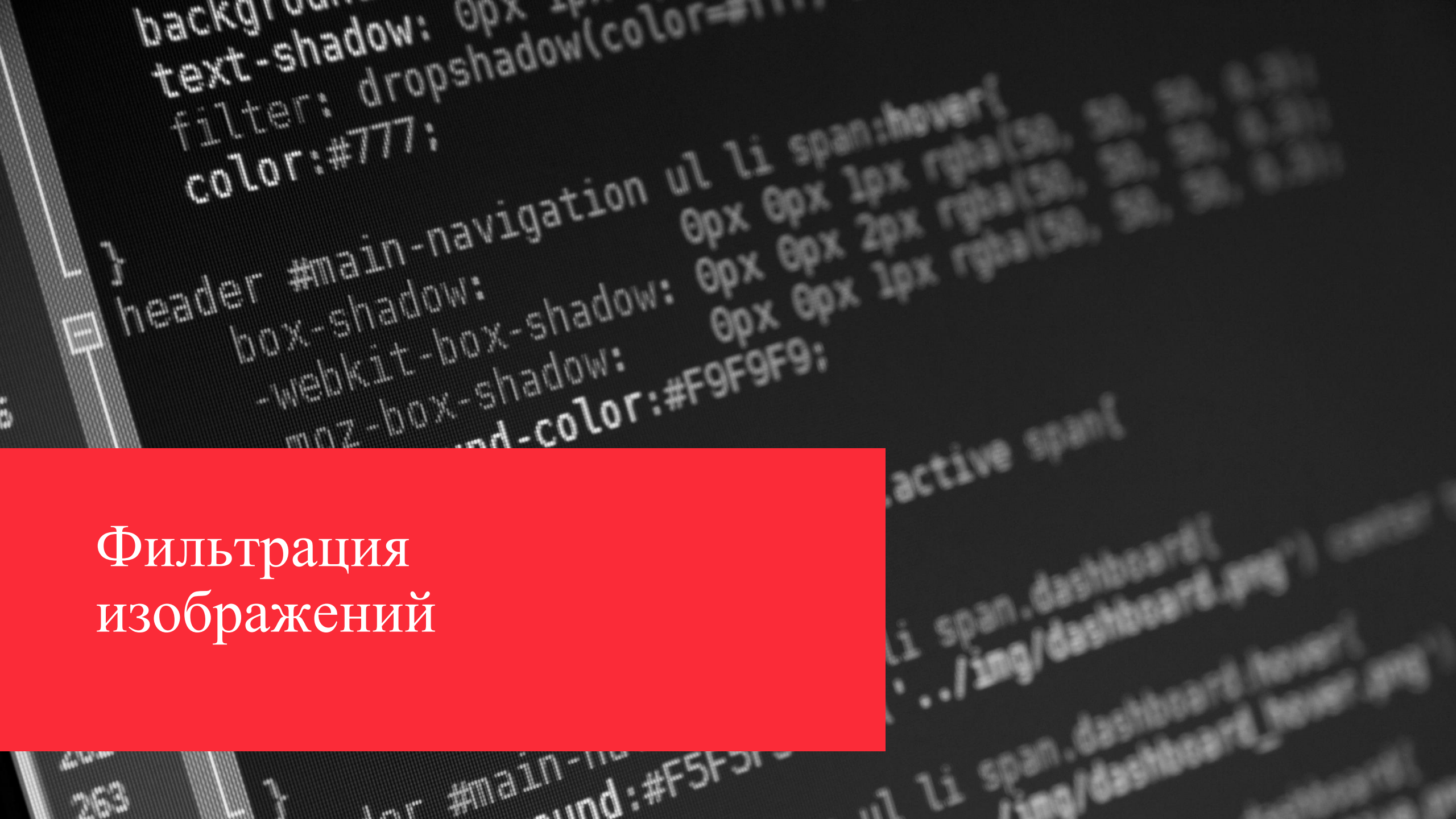
$$x_{ij} = K c_{p_{ij}}$$

K – максимальное значение интенсивности (обычно 256)



Prince «Computer vision: models, learning and inference»

OpenCV: equalizeHist()



Фильтрация изображений

Локальные (spatial/neighborhood) преобразования

$$i_n(x, y) = \text{function}(i \in \text{spatial neighbourhood}(x, y))$$

new intensity

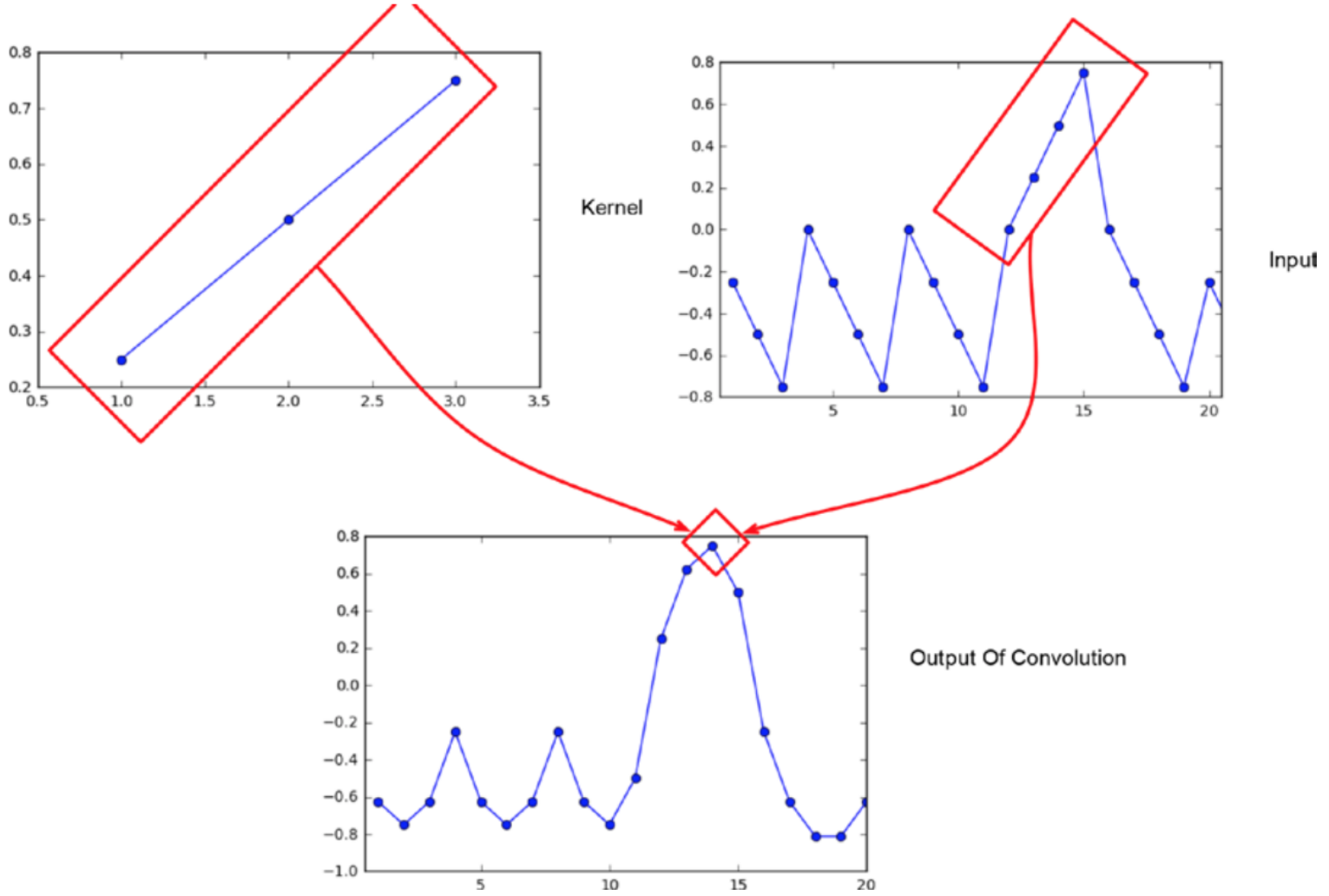
original intensity

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

Szeliski «Computer Vision»

http://www.robots.ox.ac.uk/~az/lectures/ia/lect1.pdf

Линейная фильтрация (1). Одномерный сигнал



Свертка
(convolution)

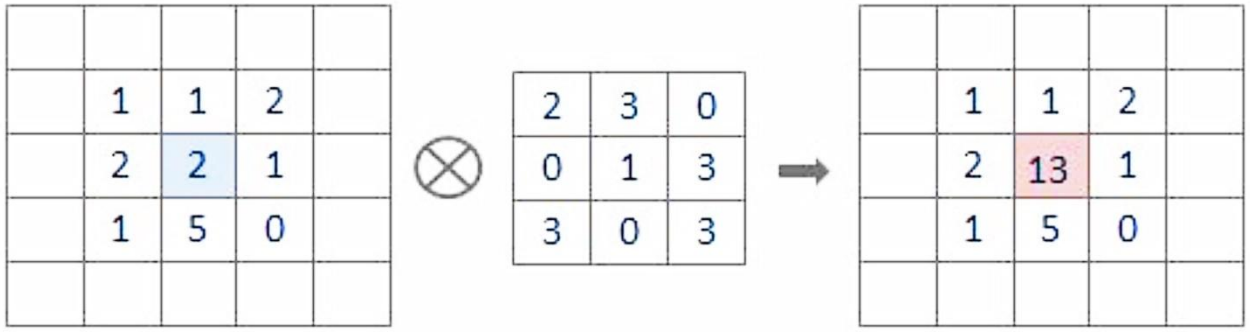
$$s(t)=\sum_a I(t-a)\cdot K(a)$$

Кросс-корреляция

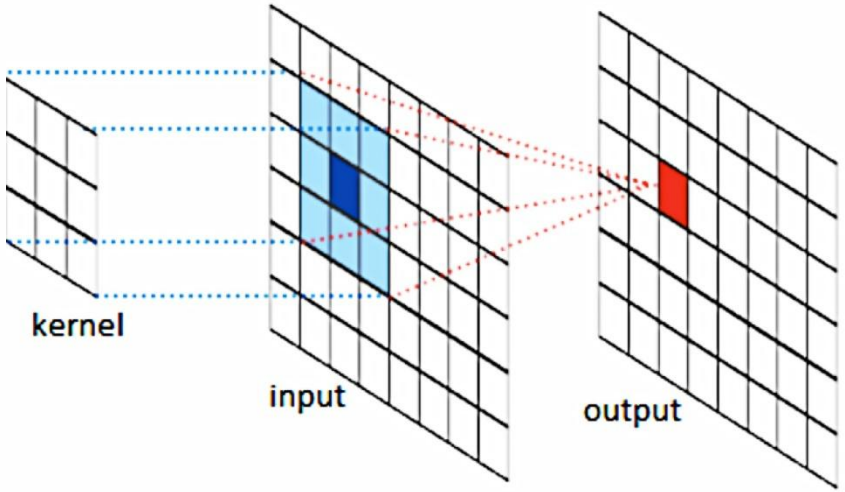
$$s(t)=\sum_a I(t+a)\cdot K(a)$$

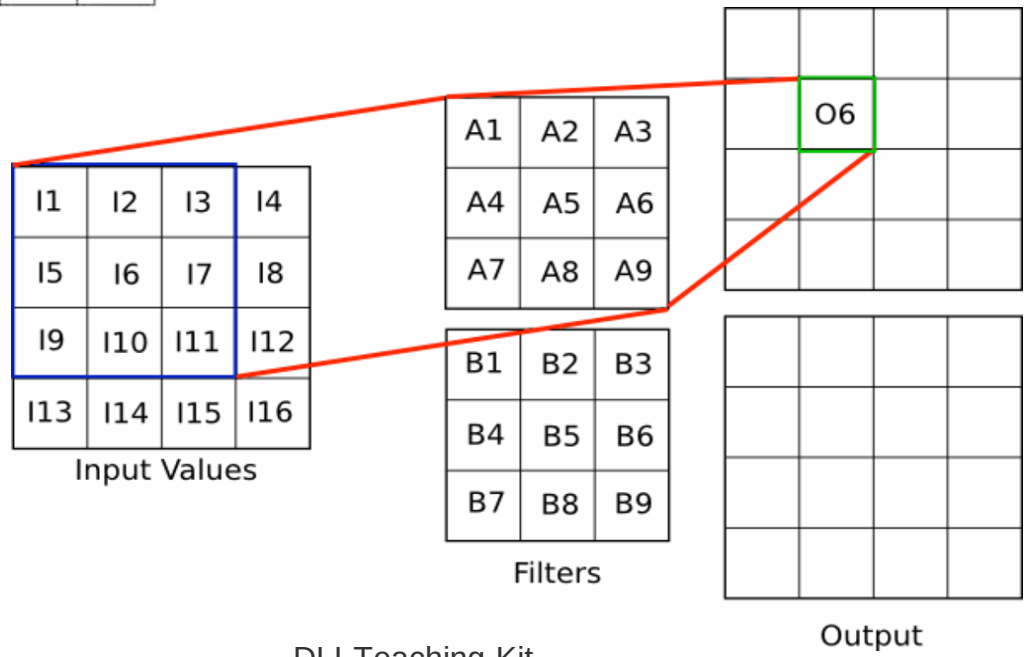
Ketkar “Deep Learning with Python”

Линейная фильтрация (2). Изображения



$$x_{ij} = \sum_{m=-M}^{M} \sum_{n=-N}^{N} p_{i-m,j-n} f_{m,n}$$





DLI-Teaching-Kit

http://intellabs.github.io/RiverTrail/tutorial/

OpenCV: filter2D()

Сглаживание (smoothing)

Box-фильтр: каждый пиксель заменяется на среднее значение яркости в окрестности

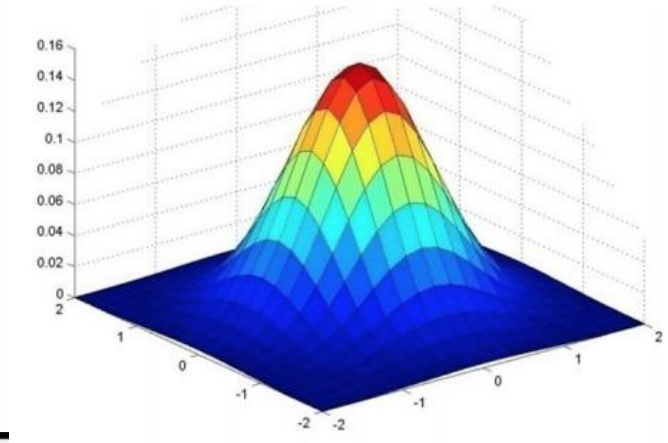
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

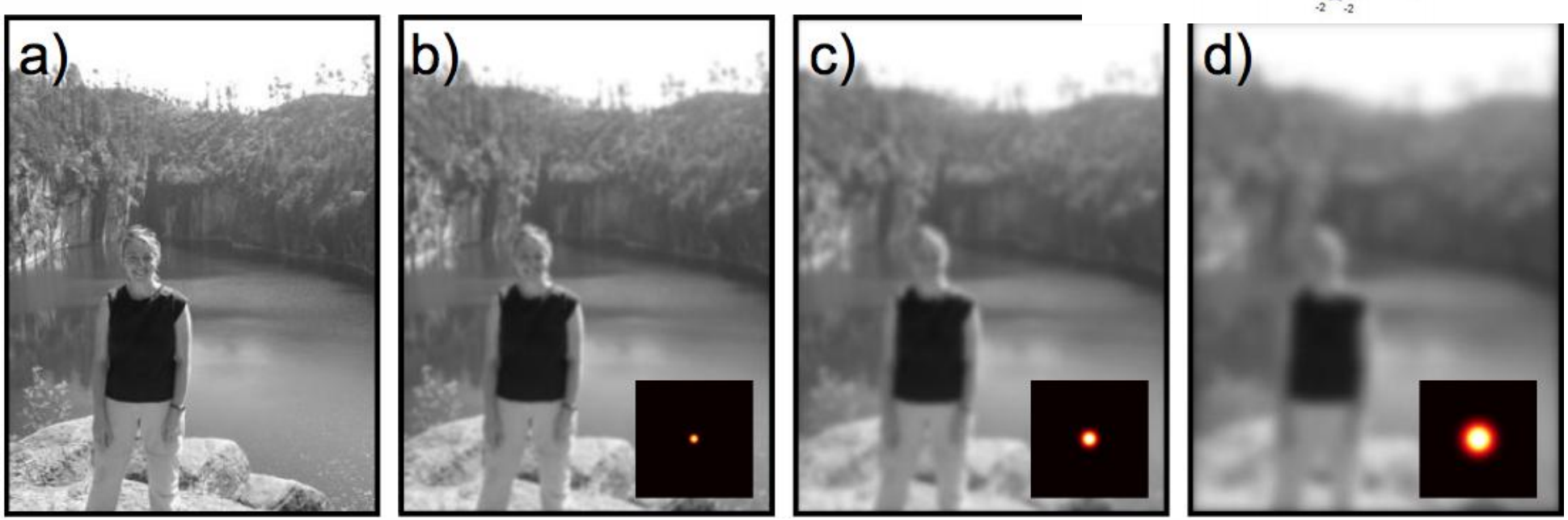
https://www.mathworks.com/help/images/ref/imboxfilt.html

OpenCV: boxFilter()

Гауссовский фильтр (smoothing, blurring)

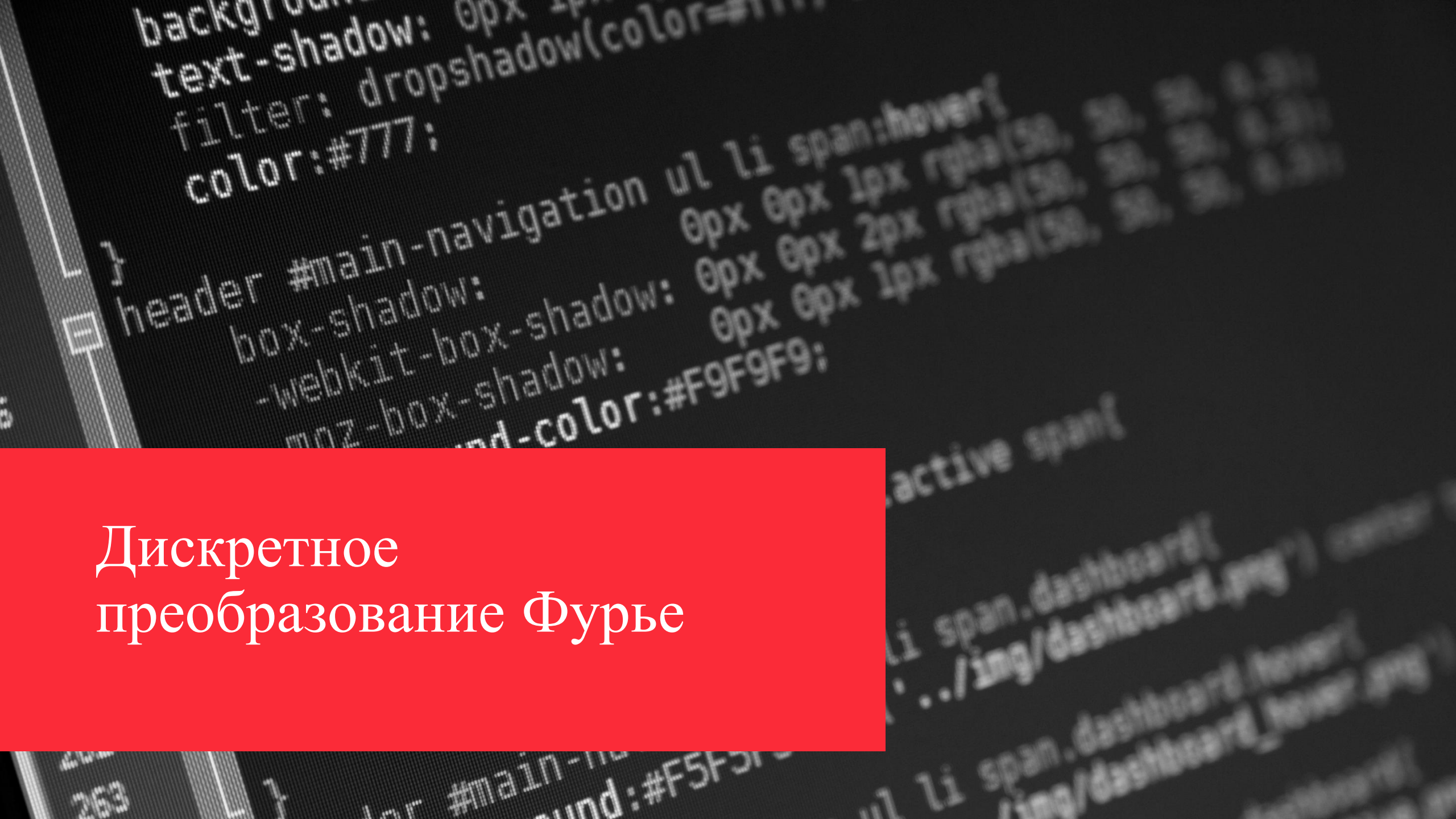
$$f(m,n) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{m^2+n^2}{2\sigma^2}\right]$$





OpenCV: GaussianBlur()

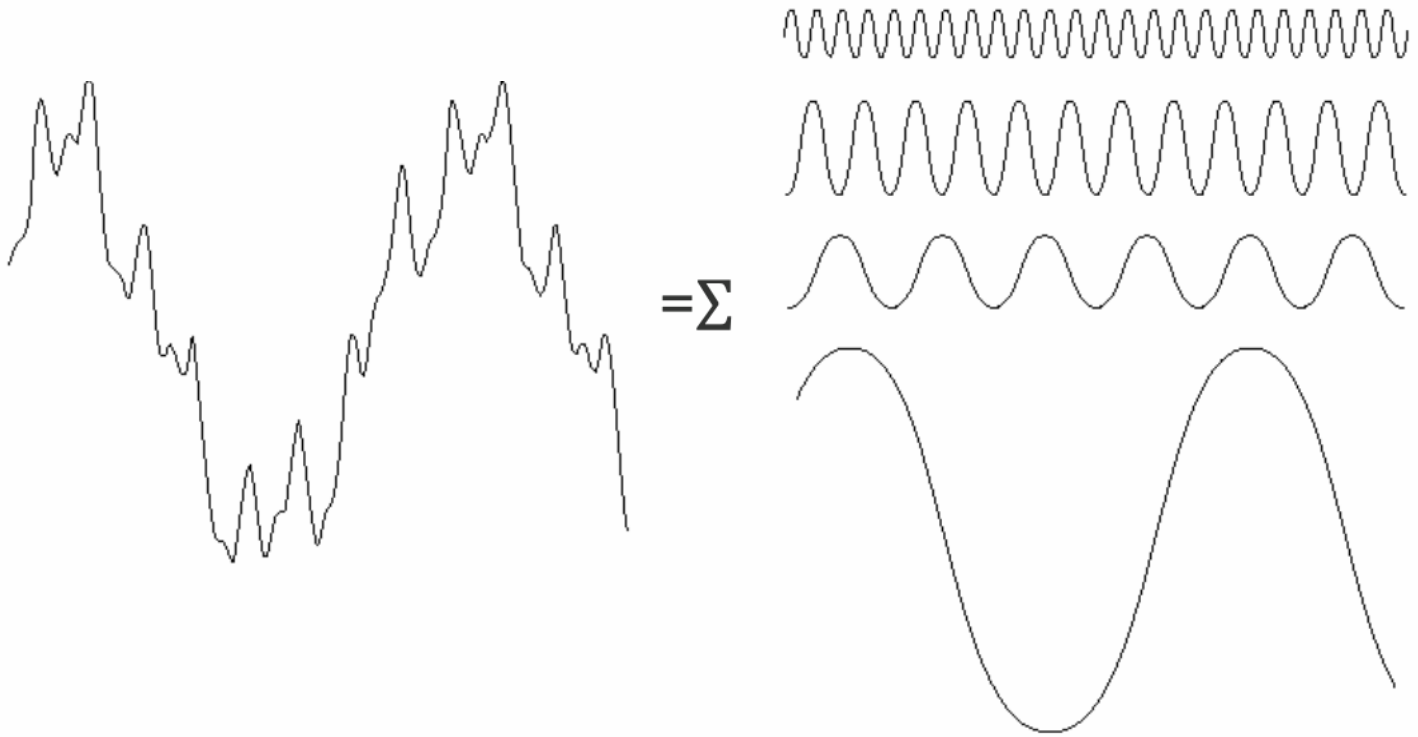
Prince «Computer vision: models, learning and inference»



Дискретное преобразование Фурье

Ряд Фурье

Сигнал можно представить как взвешенную сумму синусов и косинусов



https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid346761.pdf

Преобразование Фурье

Удобнее работать с комплексным сигналом:

$$r\,e^{j\alpha} = r\left(\cos\alpha + j\sin\alpha\right)$$

Интеграл (непрерывное преобразование) Фурье

$$F(\omega) = \int_{t} f(t)e^{-j\omega t}\,dt \qquad \text{analysis}$$

$$f(t) = \frac{1}{2\pi}\int_{\omega} F(\omega)e^{j\omega t}\,d\omega \qquad \text{synthesis}$$

Пространственная частота $\omega=2\pi u$

Дискретное преобразование Фурье (ДПФ)

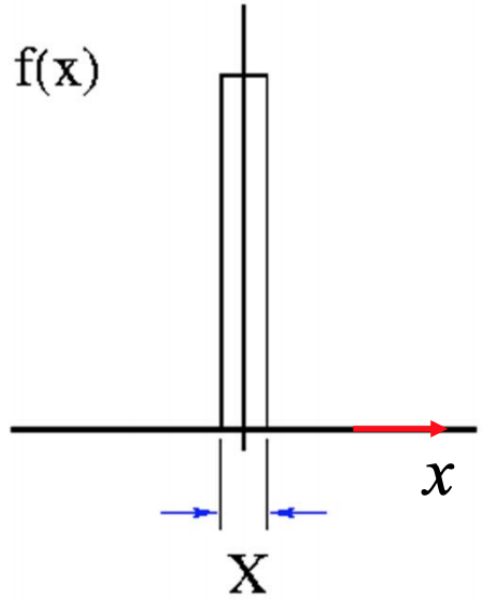
$$F[k] = \frac{1}{N}\sum_{n=0}^{N-1} f[n]e^{-j2\pi kn/T}$$

$$f[n] = \sum_{n=0}^{N-1} F[k]e^{j2\pi kn/T}$$

https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid346761.pdf

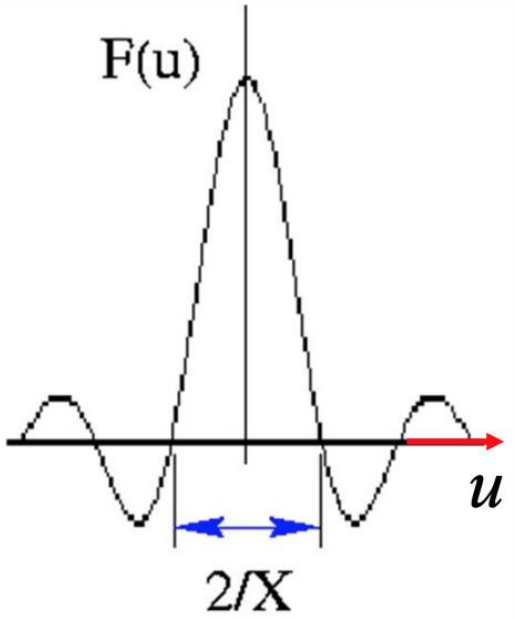
Пример

$$f(x) = \begin{cases} 1, & |x| < \frac{X}{2}, \\ 0, & |x| \geq \frac{X}{2}. \end{cases}$$



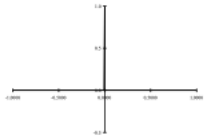
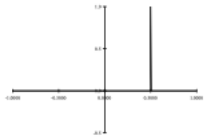
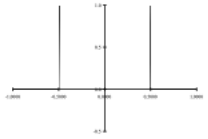
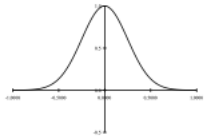
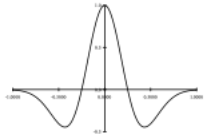
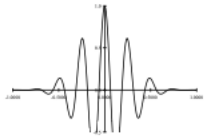
$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx = \int_{-X/2}^{X/2} e^{-j2\pi ux} dx$$

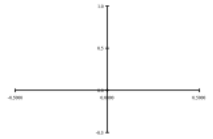
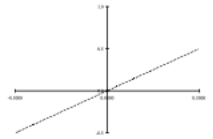
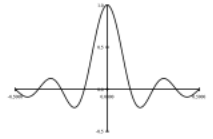
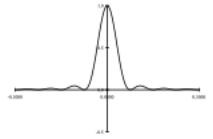
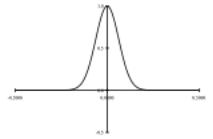
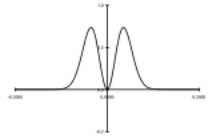
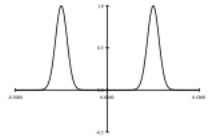
$$= \frac{1}{-j2\pi u}[e^{-j2\pi uX/2} - e^{j2\pi uX/2}] = X\frac{\sin(\pi Xu)}{(\pi Xu)} = X\mathrm{sinc}(\pi \mathrm{Xu}).$$



http://www.robots.ox.ac.uk/~az/lectures/ia/lect2.pdf

Известные Фурье-пары

impulse		$\delta(x)$	$\Leftrightarrow$	1
shifted impulse		$\delta(x-u)$	$\Leftrightarrow$	$e^{-j\omega u}$
box filter		$\text{box}(x/a)$	$\Leftrightarrow$	$a\text{sinc}(a\omega)$
tent		$\text{tent}(x/a)$	$\Leftrightarrow$	$a\text{sinc}^2(a\omega)$
Gaussian		$G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$
Laplacian of Gaussian		$(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$\Leftrightarrow$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1})$
Gabor		$\cos(\omega_0 x)G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega\pm\omega_0;\sigma^{-1})$

Szeliski «Computer Vision»

Теорема о свертке

$$\mathcal{F}[f * g] = \sum_n f * g \, e^{-i\omega n} = \sum_n \sum_m f(m)\, g(n-m)\, e^{-i\omega n}$$

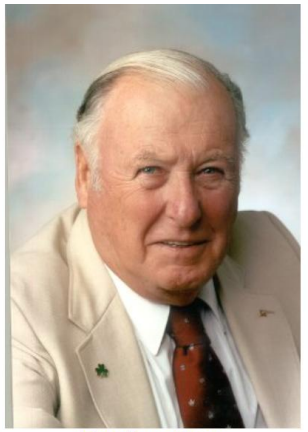
$$= \sum_m f(m) \sum_n g(n-m) e^{-i\omega n}$$

$$= \sum_m f(m)\, \hat{g}(\omega)\, e^{-i\omega m}$$

ДПФ от свертки – произведение ДПФ!

http://www.cs.toronto.edu/~jepson/csc320/notes/linearFilters2.pdf

Быстрое преобразование Фурье (БПФ)



Джеймс Кули

Джон Тьюки

Сложность ДПФ: $O(N^2)$.
Пусть N - степень двойки. «Разделяй и властвуй»:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i\,2\pi\,k\,n\,/\,N} = \sum_{m=0}^{N/2-1} x_{2m} \cdot e^{-i\,2\pi\,k\,(2m)\,/\,N} + \sum_{m=0}^{N/2-1} x_{2m+1} \cdot e^{-i\,2\pi\,k\,(2m+1)\,/\,N}$$

$$= \sum_{m=0}^{N/2-1} x_{2m} \cdot e^{-i\,2\pi\,k\,m\,/\,(N/2)} + e^{-i\,2\pi\,k\,/\,N} \sum_{m=0}^{N/2-1} x_{2m+1} \cdot e^{-i\,2\pi\,k\,m\,/\,(N/2)}$$

Симметрия: $$X_{N+k} = \sum_{n=0}^{N-1} x_n \cdot e^{-i\,2\pi\,(N+k)\,n\,/\,N} = \sum_{n=0}^{N-1} x_n \cdot e^{-i\,2\pi\,n} \cdot e^{-i\,2\pi\,k\,n\,/\,N} = X_k$$

Сложность БПФ: O(NlogN)

https://habr.com/ru/company/otus/blog/449996/

Двумерное преобразование Фурье

НПФ

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy,$$
$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}\,du\,dv$$

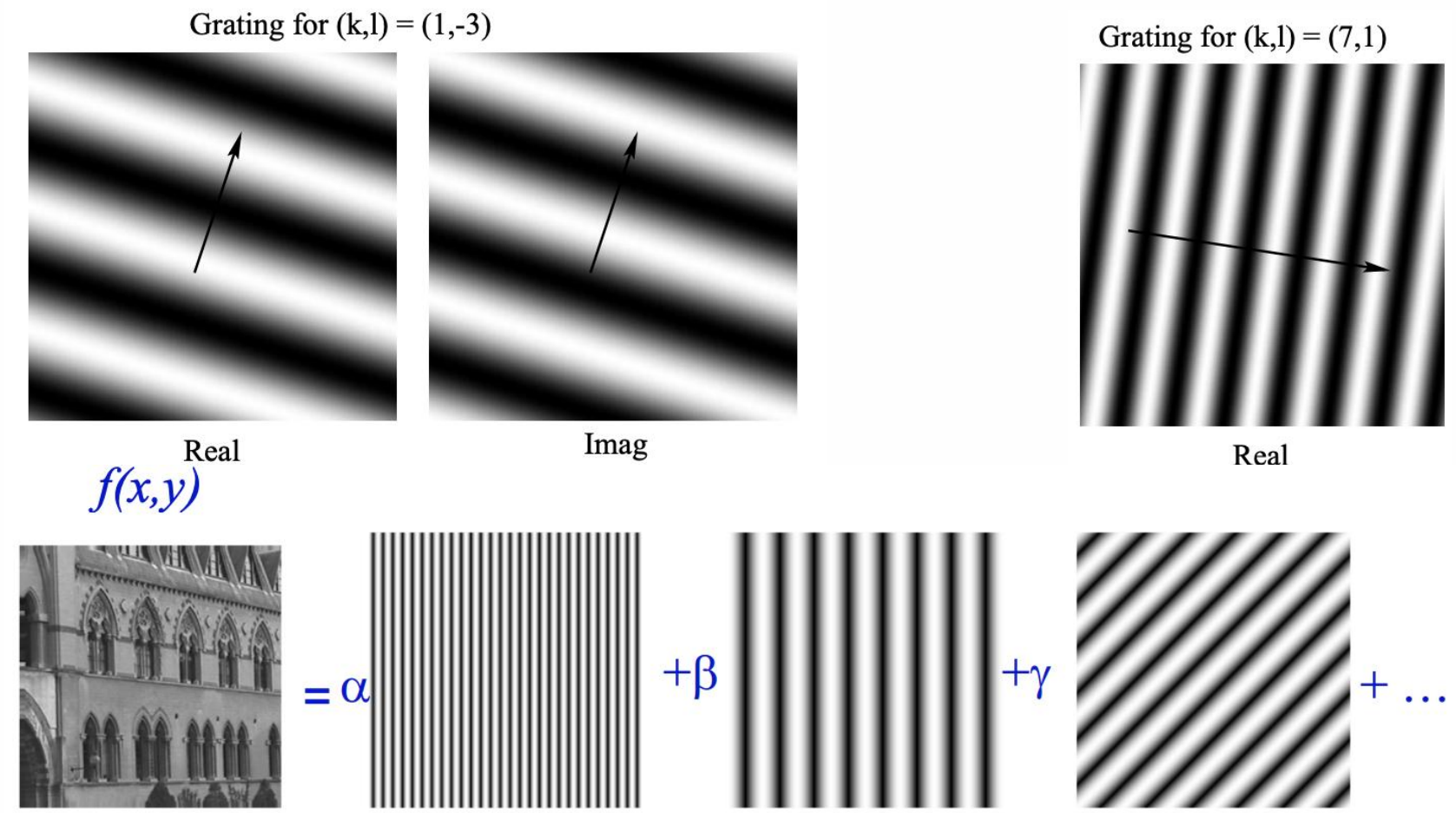
ДПФ

$$\hat{h}(k,l) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} e^{-i(\omega_k n+\omega_l m)}\,h(n,m)$$
$$h(n,m) = \frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} e^{i(\omega_k n+\omega_l m)}\,\hat{h}(k,l)$$

|F(u,v)| - амплитудный (magnitude) спектр
atan(Im(F)/Re(F)) – фазовый (phase angle) спектр

http://www.robots.ox.ac.uk/~az/lectures/ia/lect2.pdf

Базисные функции



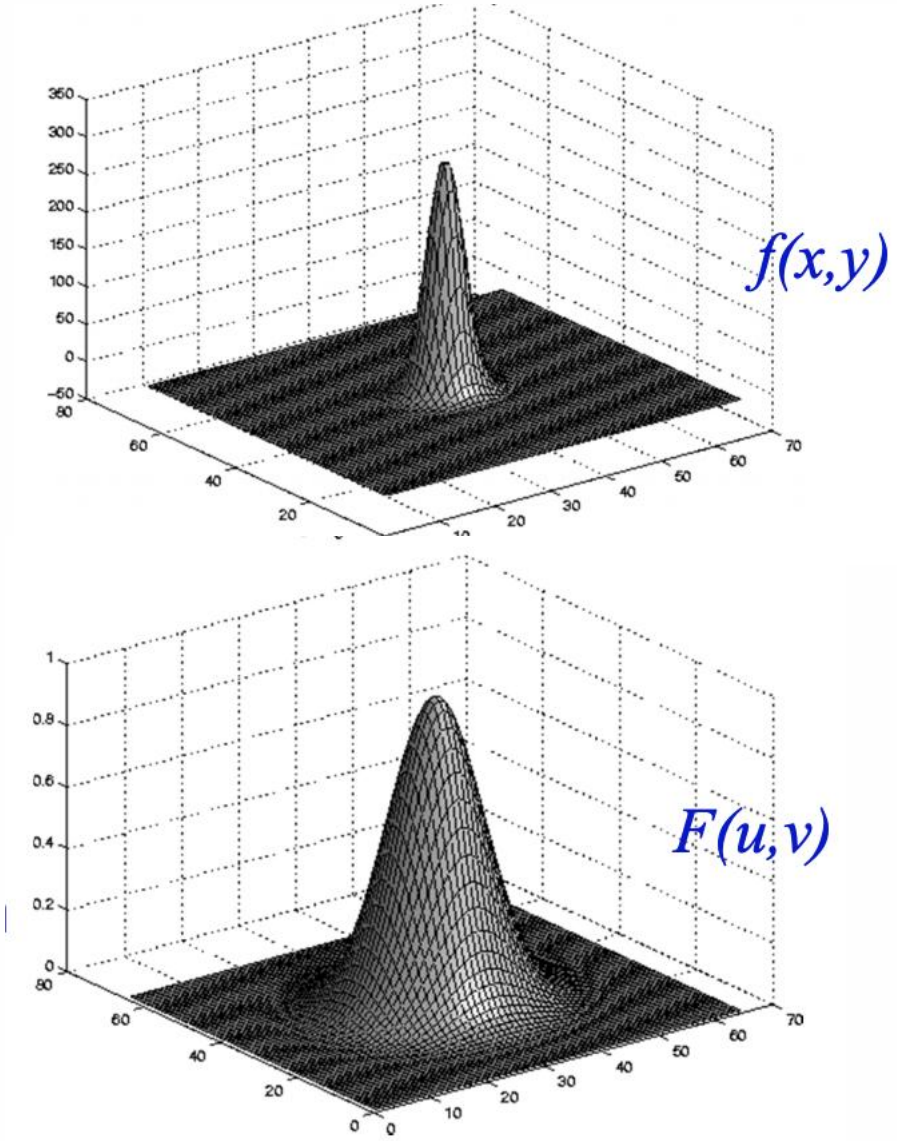
https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid346761.pdf

http://www.robots.ox.ac.uk/~az/lectures/ia/lect2.pdf

Функция Гаусса

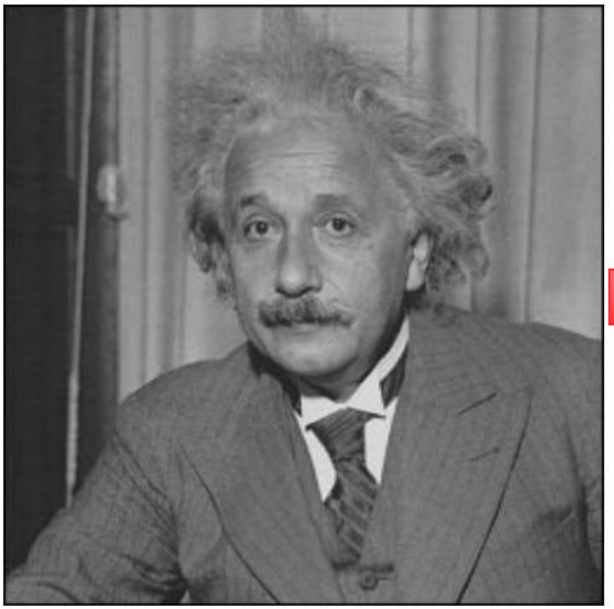
$$f(r) = \frac{1}{2\pi\sigma^2} e^{-r^2/2\sigma^2} \qquad r^2 = x^2 + y^2.$$

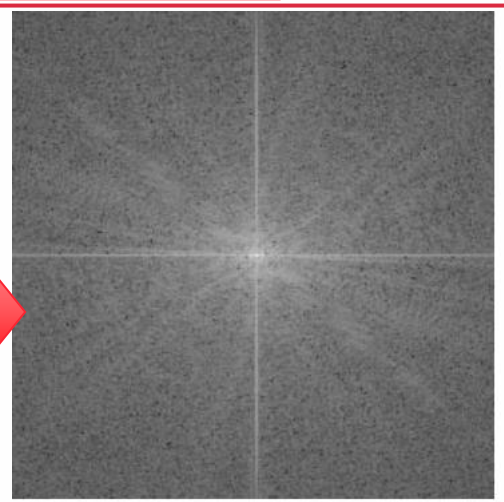
$$F(u,v) = F(\rho) = e^{-2\pi^2\rho^2\sigma^2} \qquad \rho^2 = u^2 + v^2.$$



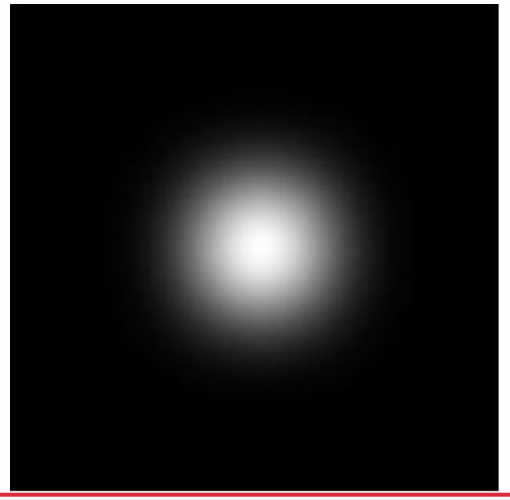
http://www.robots.ox.ac.uk/~az/lectures/ia/lect2.pdf

Низкочастотный (low-pass) фильтр

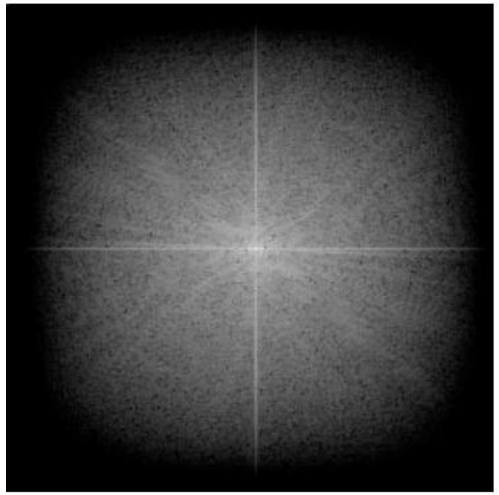


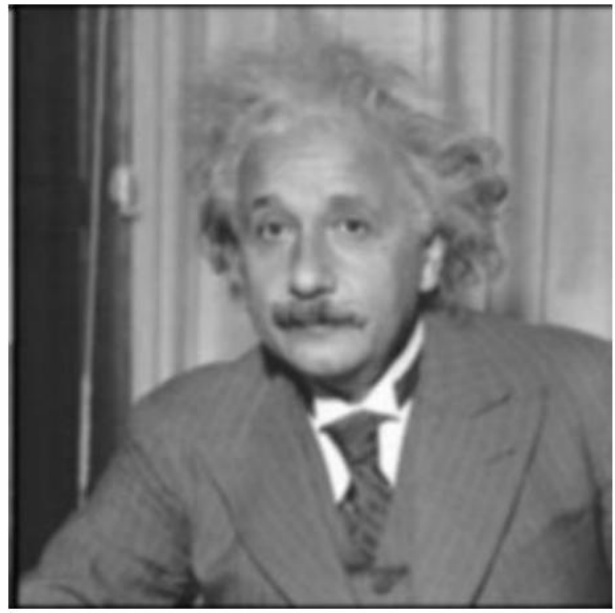


x



=





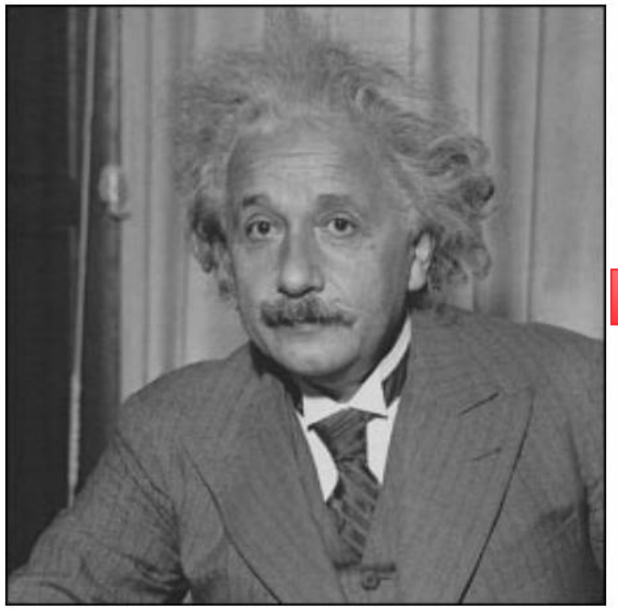
Space domain

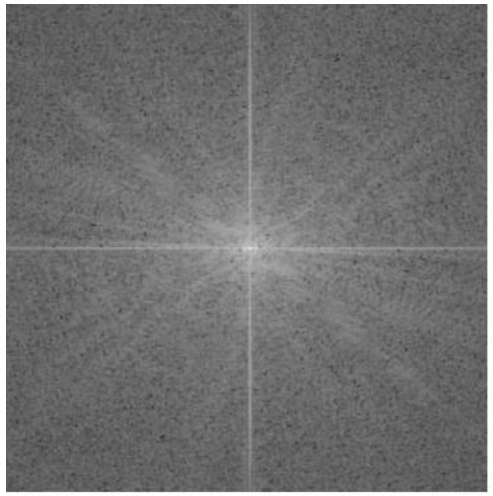
Frequency domain

Space domain

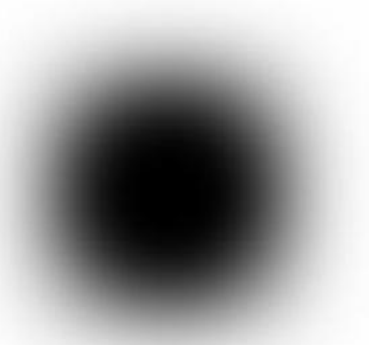
https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid346761.pdf

Высокочастотный (high-pass) фильтр



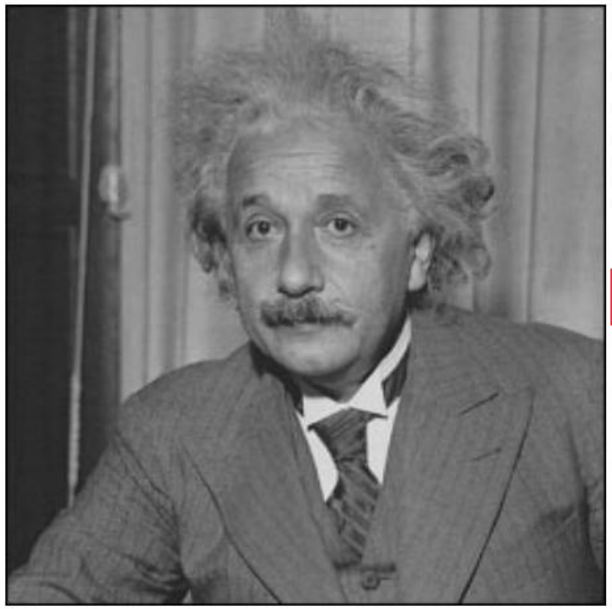


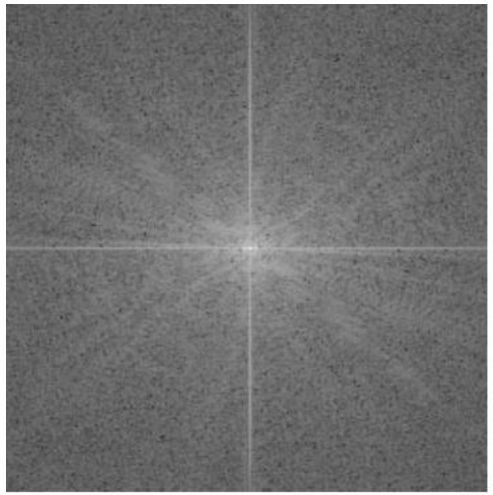
x



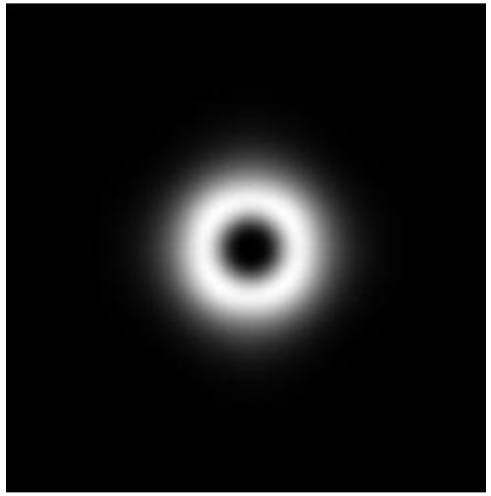
https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid346761.pdf

Полосовой (band-pass) фильтр

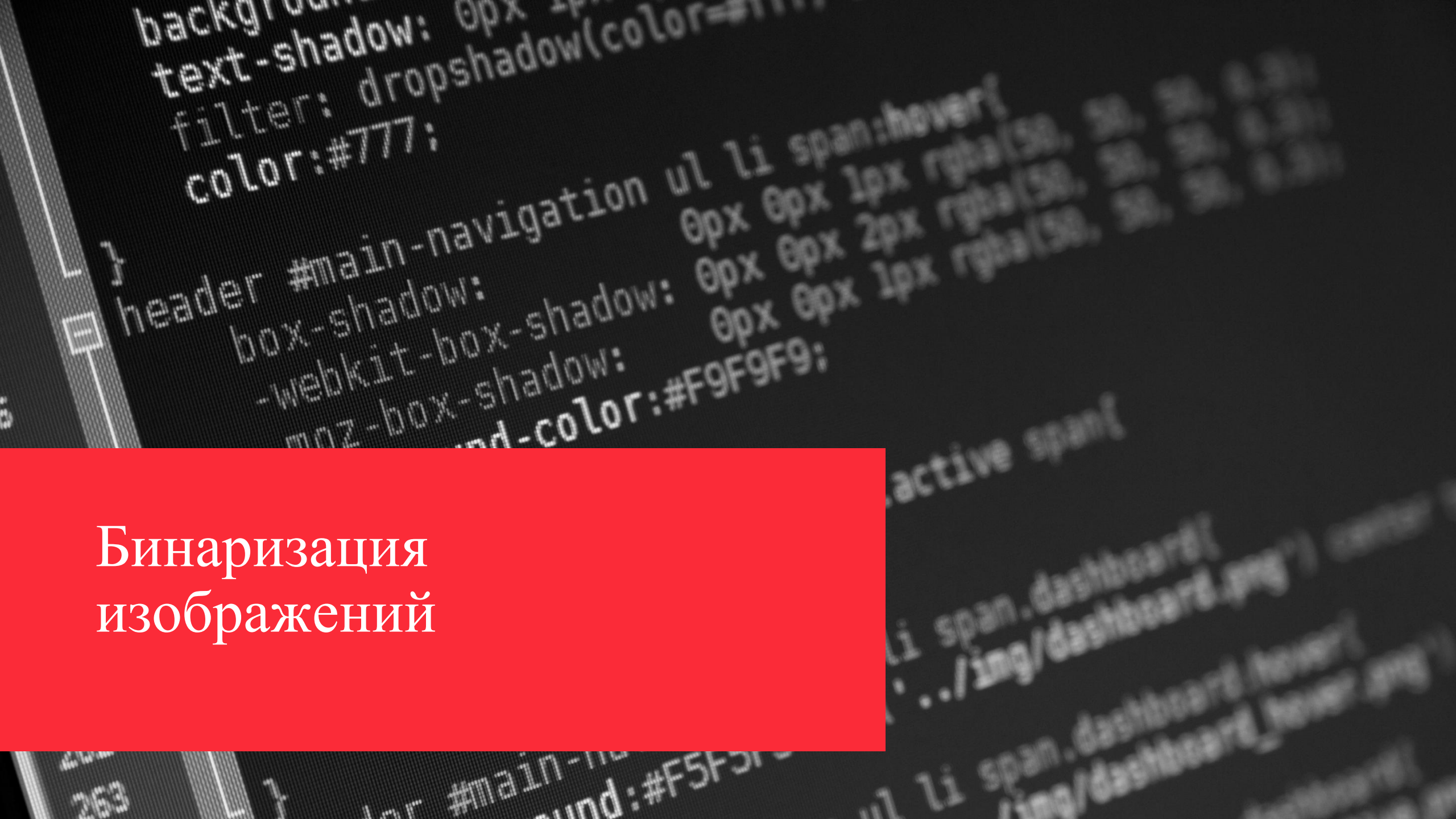




x



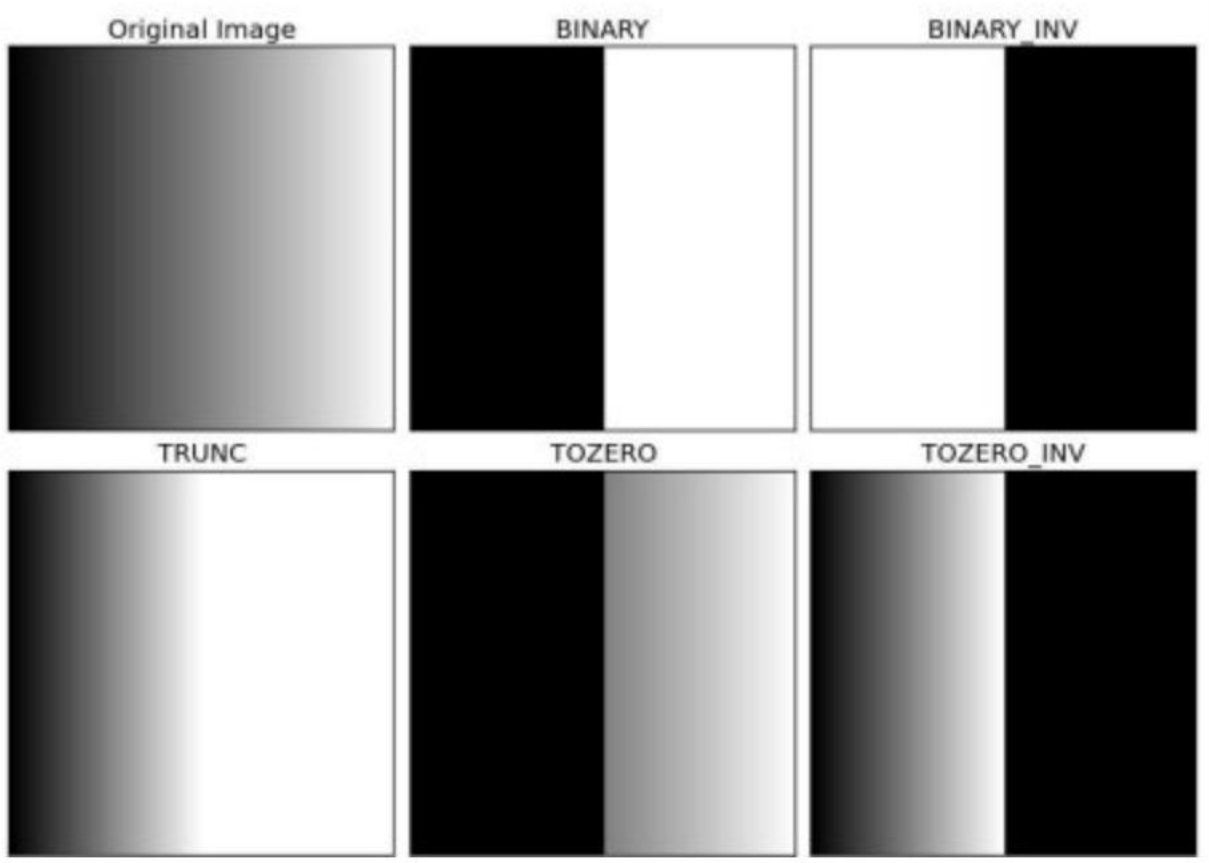
https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid346761.pdf



Бинаризация изображений

Пороговая бинаризация (global thresholding)

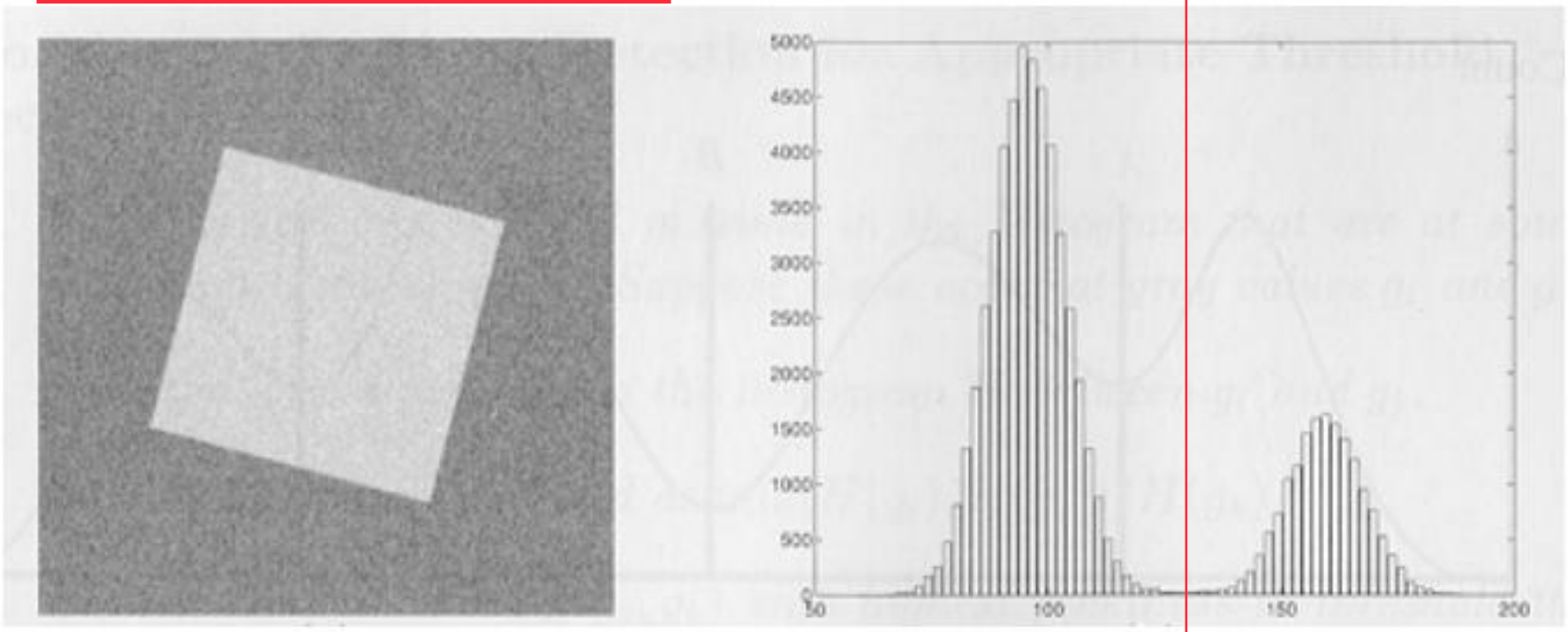
If $f(x, y) > T$ then $f(x, y) = 0$ else $f(x, y) = 255$



OpenCV: threshold()

https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

Автоматический выбор порога (1)

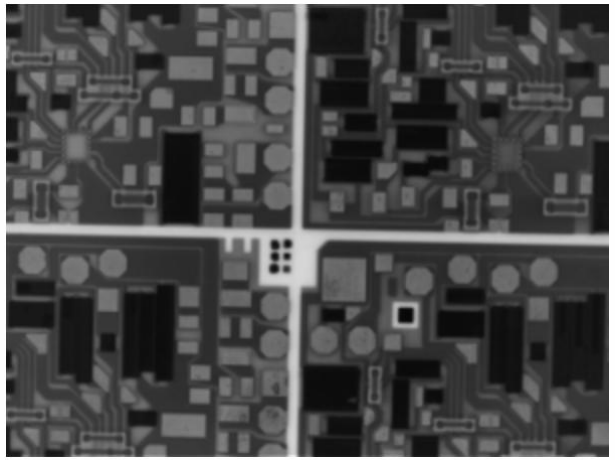


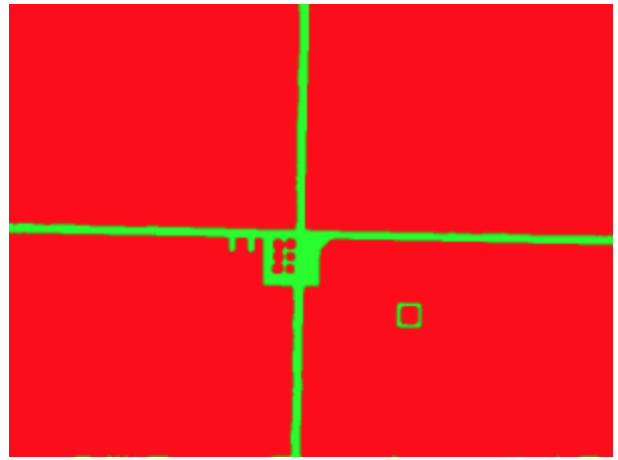
Порог можно подобрать по гистограмме

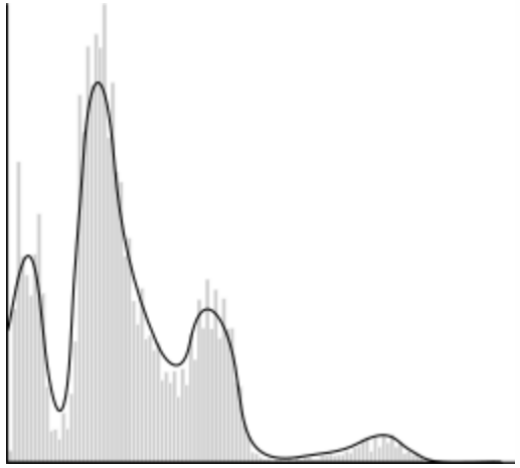
https://www.cse.unr.edu/~bebis/CS791E/Notes/Thresholding.pdf

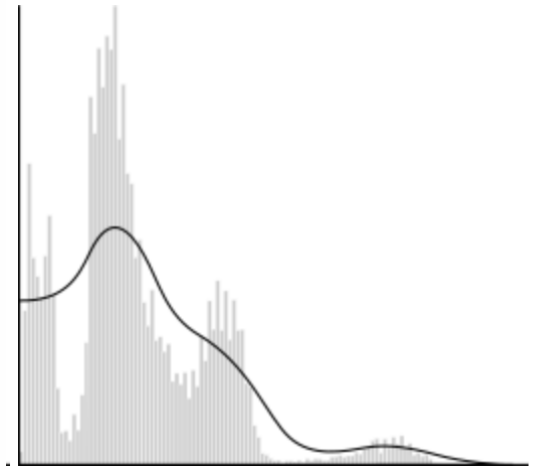
Автоматический выбор порога (2)

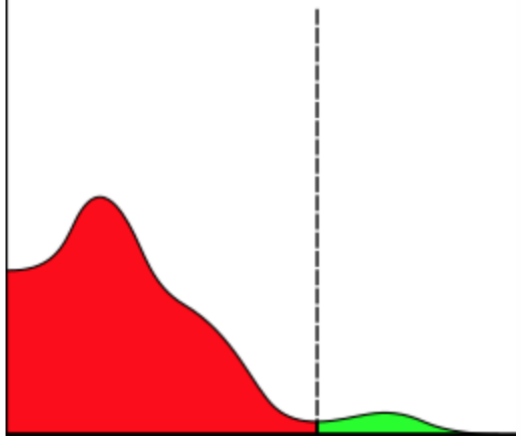
Гистограмма сглаживается с помощью одномерного фильтра Гаусса, пока не будет только один локальный минимум





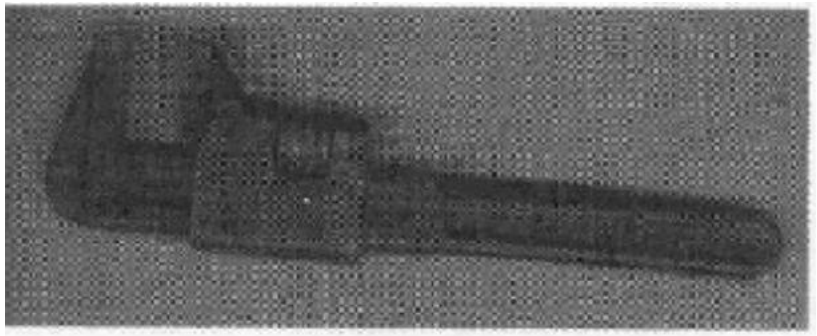


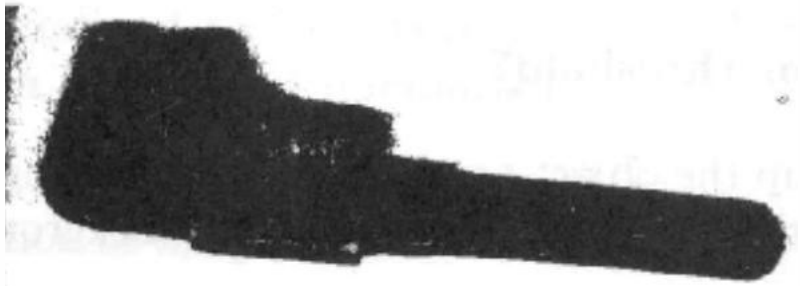


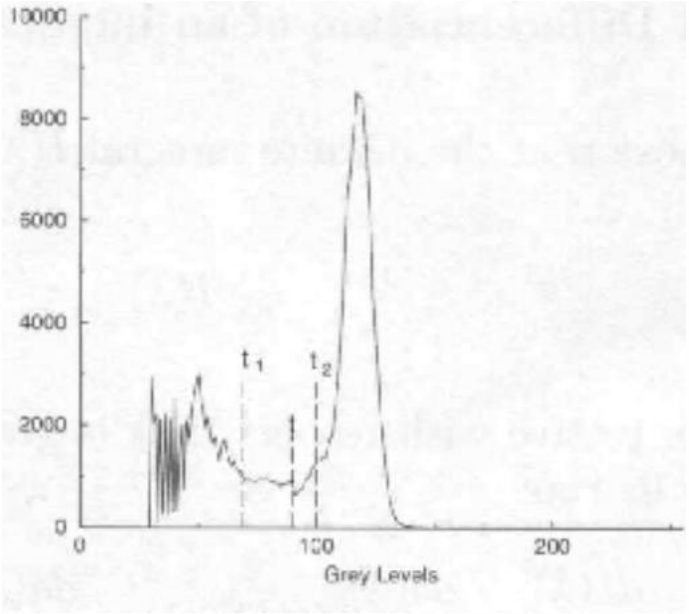


https://www.mvtec.com/doc/halcon/1805/en/toc_segmentation_threshold.html

Гистерезис



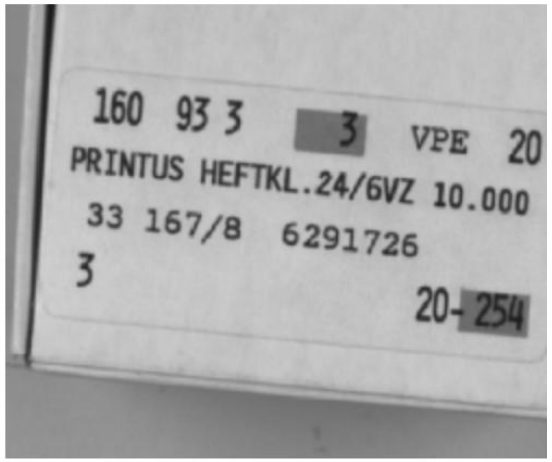


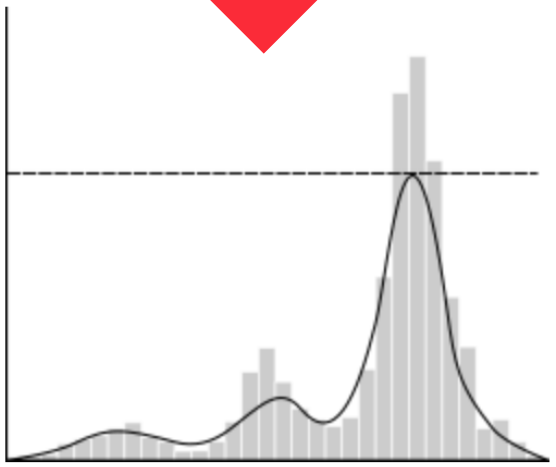


1. Два порога (точно фон и точно объект)
2. Остальные точки классифицируются как часть объекта, только если они касаются других точек объекта

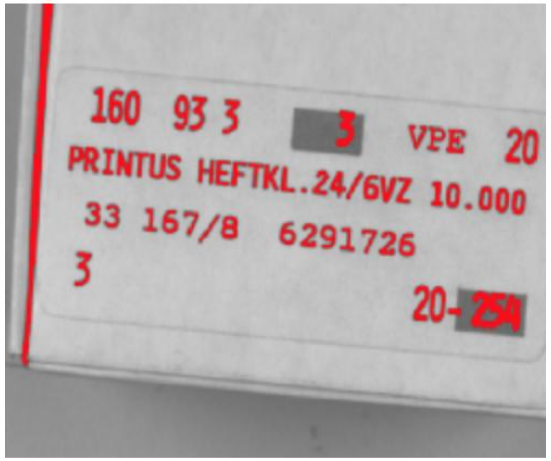
https://www.cse.unr.edu/~bebis/CS791E/Notes/Thresholding.pdf

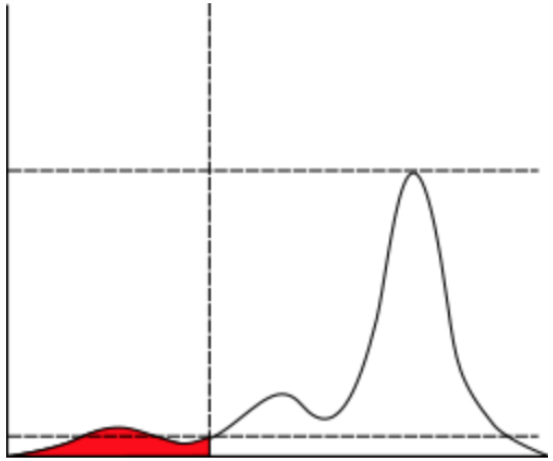
Сегментация текста





histogram[threshold] * 100.0 <
histogram[maximum] * (100.0 - Percent)

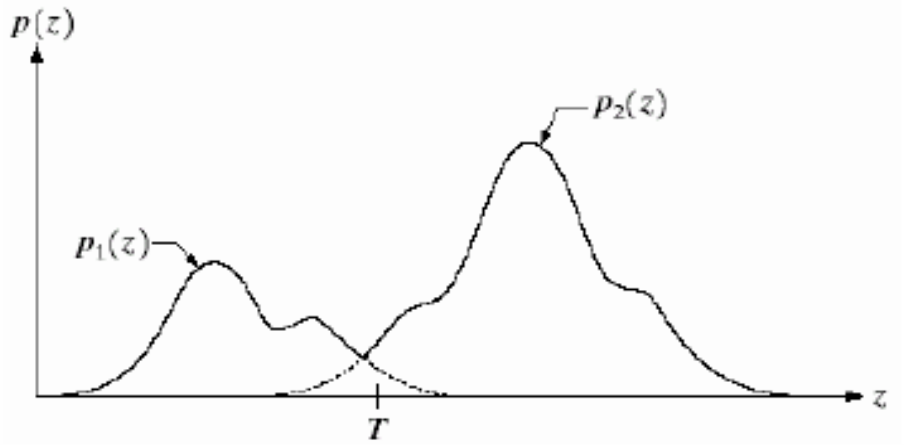




https://www.mvtec.com/doc/halcon/1805/en/toc_segmentation_threshold.html

Оптимальный выбор порога

Распределение интенсивностей объекта (1) и фона (2)



Вероятность ошибки ложной тревоги (фон как объект)

$$E_1(T) = \int_{-\infty}^{T} p_2(z)\,dz$$

Вероятность пропуска объекта (объект как фон)

$$E_2(T) = \int_{T}^{\infty} p_1(z)\,dz$$

Цель – выбрать порог T, минимизирующий среднюю ошибку

$$E(T) = P_2\ E_1(T) + P_1\ E_2(T)$$

Решение для нормальных распределений

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2}\ \ln\left(\frac{P_2}{P_1}\right)$$

https://slideplayer.com/slide/13407919/

Метод Оцу

P(i),i=1,2,…,L - гистограмма $q_b(T) = \sum_{i=1}^{T} P(i), \quad q_o(T) = \sum_{i=T+1}^{L} P(i) \quad (q_b(T) + q_o(T) = 1)$

$$\sigma^2 = \sum_{i=1}^{L} (i-\mu)^2 P(i) = q_b(T)\sigma_b^2(T) + q_o(T)\sigma_o^2(T) + q_b(T)(\mu_b(T)-\mu)^2 + q_o(T)(\mu_o(T)-\mu)^2$$

Средняя внутриклассовая дисперсия

Средняя межклассовая дисперсия

Минимум внутриклассовой дисперсии – в том же *T*, что и максимум межклассовой дисперсии

$$\sigma_B^2 = \frac{[\mu(T) - \mu q_B(T)]^2}{q_B(T) q_o(T)} \qquad \mu(T) = \sum_{i=1}^{T} iP(i) \qquad \mu = \frac{\sum_{i=1}^{L} iP(i)}{\sum_{i=1}^{L} P(i)} = \sum_{i=1}^{L} iP(i)$$

OpenCV:
threshold(THRESH_OTSU)

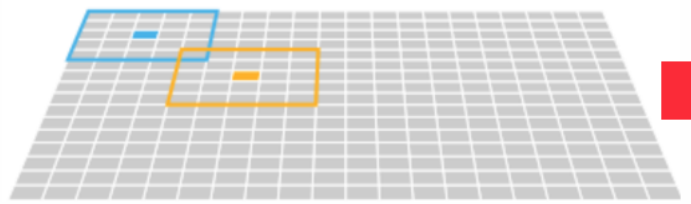
https://www.cse.unr.edu/~bebis/CS791E/Notes/Thresholding.pdf

https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

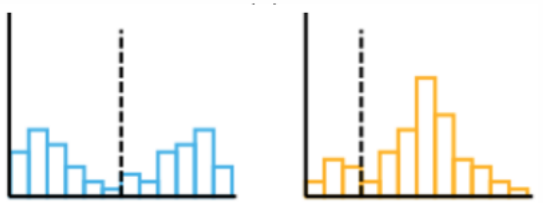
Локальные методы (local thresholding)

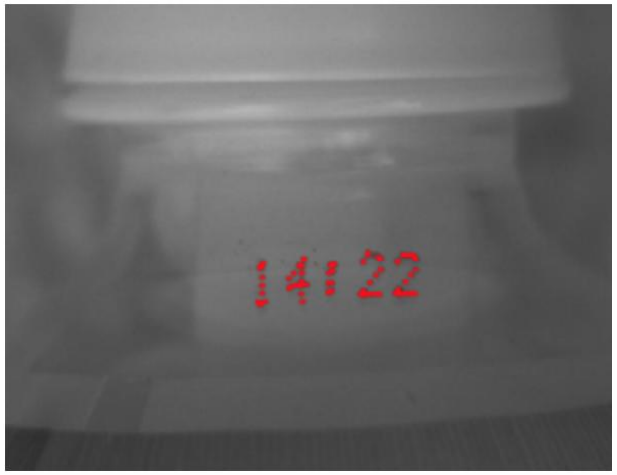
Глобальный порог

Окрестность каждой точки анализируется отдельно



Локальные гистограммы





OpenCV: adaptiveThreshold()

https://www.mvtec.com/doc/halcon/1805/en/toc_segmentation_threshold.html

Перейдем к примерам

https://github.com/HSE-asavchenko/MADE-mobile-image-processing/tree/master/lesson2/src