

# Тема 3

Работа с камерами — получение данных.  
Цифровые фильтры.

# Фильтры: классификация

- Линейные фильтры (свертка с базисной функцией)
- Фильтры на основе скользящего окна
- Нелинейные фильтры

# Матрица свертки

Матрица свертки- матрица, которая умножается на окно исходного изображения, чтобы получить значение результирующего пиксела умноженное на коэф. нормирования ( Сумма элементов матрицы нормирования).

Входное изображение

12	14	41
43	84	24
2	1	43

Матрица

0,5	0,75	0,5
0,75	1,0	0,75
0,5	0,75	0,5

Результат

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \left( \begin{array}{l} 12 * 0,5 + 14 * 0,75 + 41 * 0,5 + \\ 43 * 0,75 + 84 * 1,0 + 24 * 0,75 + \\ 2 * 0,5 + 1 * 0,75 + 43 * 0,5 \end{array} \right) \times \frac{1}{\text{div}} \begin{array}{c} \text{---} \\ \text{---} \end{array} 32,41667$$

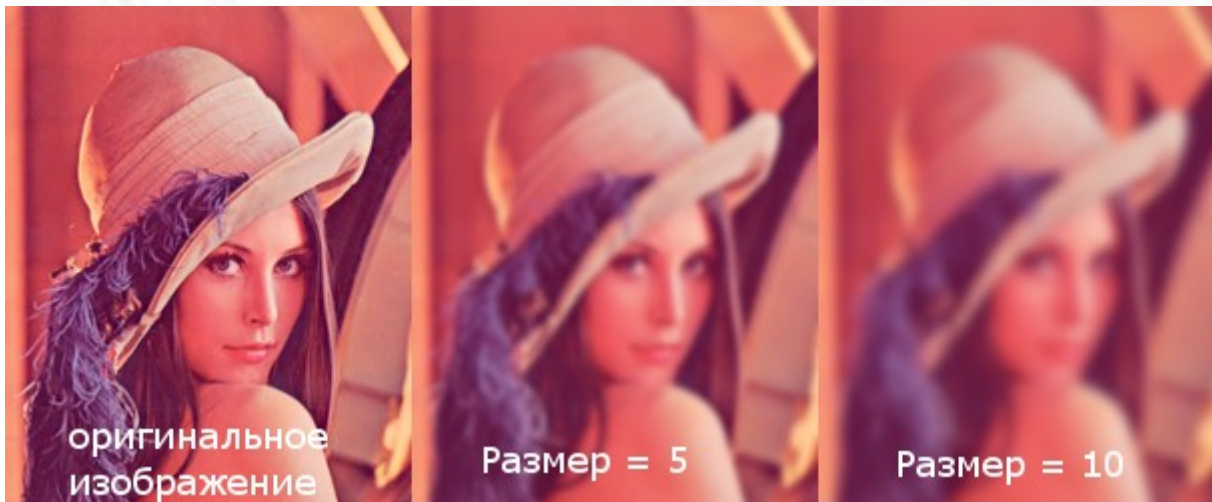
div = 6

# Размытие по Гауссу

- Фильтр размытия по Гауссу имеет сложность  $O(h_i * w_i * n * n)$ , где  $h_i$ ,  $w_i$  – размеры изображения,  $n$  – размер матрицы (ядра фильтра).

# Матрица размытия по гауссу

0,000789	0,006581	0,013347	0,006581	0,000789
0,006581	0,054901	0,111345	0,054901	0,006581
0,013347	0,111345	0,225821	0,111345	0,013347
0,006581	0,054901	0,111345	0,054901	0,006581
0,000789	0,006581	0,013347	0,006581	0,000789



# Фильтр Гаусса ОПТИМИЗИРОВАННЫЙ

- Два прохода: вертикальный и горизонтальный с матрицей одномерного распределения.
- Сложность данного алгоритма будет  $O(h_i * w_i * n) + O(h_i * w_i * n) = 2 * O(h_i * w_i * n)$ ,

0,028087	0,23431	0,475207	0,23431	0,028087
----------	---------	----------	---------	----------

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$

Стандартное откл. В формуле: 1

# Матрица резкость (sharpen)

-1	-1	-1
-1	9	-1
-1	-1	-1





# Простые операции

NEGATIVE

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

div = 1 offset = 256

BLUR

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

div = 9 offset = 0

SHARPEN

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

div = 1 offset = 0

EMBOSS

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

div = 1 offset = 0

LIGHT-BLUR

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

div = 4 offset = 0

LIGHT-SHARPEN

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

div = 1 offset = 0

LIGHT-EMBOSS

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

div = 1 offset = 0



# Реализация в GIMP и PS

- Все современные редакторы изображений предоставляют инструмент работы с матрицами сверток.
- Gimp->

# Общая проблема всех алгоритмов со скользящим ОКНОМ

- Что делать с границами?
  - а) ничего не делать- размер  $(n-k/2;n-k/2)$
  - б) заполнить крайними элементами (выход из положения)

# Медианный фильтр

- Уникальный по своим характеристикам фильтр, сохраняющий положение фронтов.
- Значения отсчетов внутри окна фильтра сортируются в порядке возрастания (убывания); и значение, находящееся в середине упорядоченного списка, поступает на выход фильтра. В случае четного числа отсчетов в окне выходное значение фильтра равно среднему значению двух отсчетов в середине упорядоченного списка. Окно перемещается вдоль фильтруемого сигнала и вычисления повторяются.

# Медианный фильтр

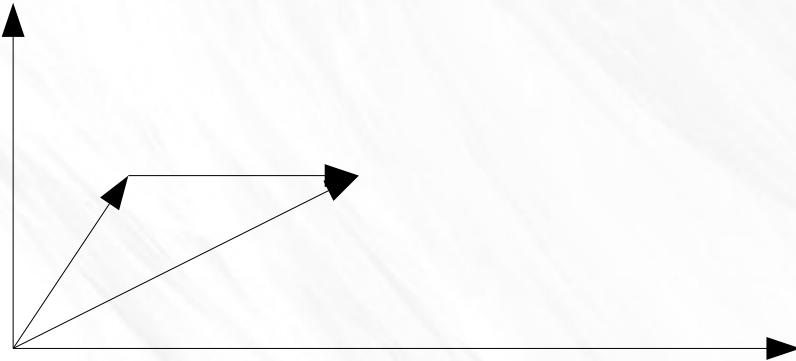
- Не изменяет фазу и амплитуду фронтов сигнала
- Может работать в индексированных цветовых пространствах
- Классическая формула работает быстро
- Может работать в многомерных векторных пространствах

# Медианный фильтр: математическое описание

$$y_{m,n} = \left\{ x_k : \sum_{j=1}^N |x_j - x_i| \geq \sum_{j=1}^N |x_j - x_k| \text{ для всех } i \in [1, \dots, N] \right\}.$$

# Векторная фильтрация

- В данном случае цветовые компоненты рассматриваются как векторы и работают законы векторной арифметики.



# Нелинейные фильтры

- Нелинейные фильтры невозможно описать ни матрицей ни оператором свертки.
- Нелинейные фильтры обычно предполагают некий алгоритмический подход, который можно описать в терминах условных операторов.

$$B(x, y) = \begin{cases} 1, & \text{если } A(x, y) \\ 0, & \text{иначе} \end{cases}$$



# Нелинейные фильтры

- Минимальное, максимальное значение окна

$$B_{\min}(x, y) = \min \{N(x, y)\}$$

$$B_{\max}(x, y) = \max \{N(x, y)\}$$

# Яркостные преобразования, преобразования гистограммы

- Яркими преобразованиями изображения называются преобразования двумерных функций яркости, описываемые простой формулой:

$$I'(x,y)=f(I(x,y))$$

причем данная формула никак не зависит от положения пиксела.

# Гистограмма

- Строим гистограмму изображения (look-Up-table)

$$Im[i, j] = LUT[Im[i, j]],$$

- Работаем с гистограммой

# Популярные операции

- Гамма коррекция
- Нормализация яркости
- 
- 
- Эквализация

$$\text{LUT}[i] = r(i)^\gamma,$$

$$\text{LUT}[i] = 255 \cdot \frac{i - I_{\min}}{I_{\max} - I_{\min}}$$

$$\text{LUT}[i] = 255 \cdot \frac{\sum_{j=1}^i \text{Hist}[j]}{\sum_{j=1}^{255} \text{Hist}[j]}$$

# Линейные фильтры,

Дискретное преобразование Фурье  
Wavelet-transform

1. Нет проблемы границы
2. Есть проблема эффекта Гиббса

# Идея преобразования Фурье

Вещественную функцию  $f(x)$  можно разложить по ортогональной системе тригонометрических функций, то есть представить в виде

$$f(x) = \int_0^{\infty} A(\omega) \cos(2\pi\omega x) d\omega - \int_0^{\infty} B(\omega) \sin(2\pi\omega x) d\omega,$$

где  $A(\omega)$  и  $B(\omega)$  называются интегральными косинус- и синус-преобразованиями:

$$A(\omega) = 2 \int_{-\infty}^{+\infty} f(x) \cos(2\pi\omega x) dx; \quad B(\omega) = 2 \int_{-\infty}^{+\infty} f(x) \sin(2\pi\omega x) dx.$$

# ДПФ

Дискретное преобразование Фурье переводит конечную последовательность вещественных чисел в конечную последовательность коэффициентов Фурье.

Пусть  $\{x_i\}, i = 0, \dots, N - 1$  - последовательность вещественных чисел - например, отсчеты яркости пикселей по строке изображения. Эту последовательность можно представить в виде комбинации конечных сумм вида

$$x_i = a_0 + \sum_{n=1}^{N/2} a_n \cos\left(\frac{2\pi ni}{N}\right) + \sum_{n=1}^{N/2} b_n \sin\left(\frac{2\pi ni}{N}\right),$$

где

$$a_0 = \frac{1}{N} \sum_{i=0}^{N-1} x_i, \quad a_{N/2} = \frac{1}{N} \sum_{i=0}^{N-1} x_i (-1)^i, \quad a_k = \frac{2}{N} \sum_{i=0}^{N-1} x_i \cos\left(\frac{2\pi ik}{N}\right),$$

$$b_k = \frac{2}{N} \sum_{i=0}^{N-1} x_i \sin\left(\frac{2\pi ik}{N}\right), \quad i \leq k < N/2.$$



# Вейвлет преобразование

- Логично было бы поговорить про него тут, но мы оставим его к теме «распознавание лиц» через преобразование Хаара

# Оценка качества изображений

- Субъективные критерии (анализ человеком)
- СКО (нужно знать эталонное изображение)
- Обеспечение одинаковых условий проведения экспертиз в разных лабораториях рекомендация союза электросвязи – ITU-R BT.500-11.

# DSIS (Double-stimulus impairment scale – метод двойного воздействия со шкалой искажений)

Метод предложен Мерцем Фаулером и Кристофером

n- k-во категорий,

Шкала погрешностей:

1. Незаметные
2. Еле заметные
3. Вполне заметные, но слабо ухудшающие изображение
4. Ухудшающие изображение, но допустимые
5. Несколько нежелательные
6. Определенно нежелательные
7. Крайне нежелательные

Результаты экспертных оценок обычно выражают с помощью среднего балла, определяемого как

$$\bar{C} = \frac{\sum_{k=1}^K n_k C_k}{\sum_{k=1}^K n_k},$$

где  $K$  – количество категорий,  $n_k$  – число изображений, отнесенных к  $k$ -й категории, а  $C_k$  – соответствующий ей балл. Считается, что для получения надежной оценки качества изображения необходимо опросить не менее двадцати наблюдателей. Одной из трудностей, связанных с оценками в баллах, является возможная нелинейность шкалы.

# Объективный критерий (нет абсолютного значения)

$$ПОСШ = 20 \log_{10} \frac{255}{\sqrt{СКО}},$$

$$СКО = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2,$$

где  $N$  – число пикселей в изображении,  $x_i$  и  $y_i$  – значения пикселей исходного и восстановленного изображений соответственно, а СКО – среднеквадратичная ошибка. Высокое значение ПОСШ означает определенную схожесть восстановленного и исходного изображений, но оно не дает гарантии того, что зрителю понравится восстановленный образ.

# www.imagemagick.org



- ImageMagick мощный инструментарий обработки изображений.
- The current release is ImageMagick 6.8.7-0 available from <http://www.imagemagick.org/download>. It runs on Linux, Windows, Mac Os X, iOS, Android OS, and others.
- The official ImageMagick web site is <http://www.imagemagick.org>.



# Инструментарий получения изображений из камеры (Linux, без OpenCV)

- FireWire камера: libdc1394
- Ethernet камера: нет стандартного способа, вероятно будет либо библиотека, либо встроенный WEB сервер, либо ffmpeg
- USB камера: подсистема fideo4linux



# Прием работы с камерой через устройство /dev/video

- `rm ./*.mp4`
- `rm ./*.avi`
- `ffmpeg -t 10 -f video4linux2 -s 1280x1024 -r 5 -i /dev/video0 webcam.mp4`
- `ffmpeg -f video4linux2 -s 1280x1024 -i /dev/video0 -f image2 snapshot.jpg 2>/dev/null`
- `ffmpeg -f video4linux2 -s 1280x1024 -i /dev/video0 -f image2 snapshot.jpg 2>/dev/null`
- `ffmpeg -f video4linux2 -s 1280x1024 -i /dev/video0 -f image2 snapshot.jpg 2>/dev/null`
- `ffmpeg -f video4linux2 -s 1280x1024 -i /dev/video0 -f image2 snapshot.jpg 2>/dev/null`

# Особенности

- Сдернуть snapshot нормального качества с первого кадра в условиях плохого освещения не получится
- Будет работать совершенно с любыми камерами, которые взаимодействуют с V4L
- Можно работать через стандартное устройство Linux.

# Заставить фотокамеру сделать СНИМОК

- Gphoto2 — библиотека(+консольная утилита, позволяющая через libusb libp2psam управлять фотокамерами). Поддерживается около 500 моделей (модель должна поддерживать протокол P2P).
- Sony, Kodak, Fuji не поддерживаются совсем, остальные камеры нужно смотреть лист совместимости
- Ничто не заставит камеру саму перейти в положение ON
- Вполне неплохой способ для научных стендов и установок там, где нужна хорошая камера, но все готовы мириться с неудобствами

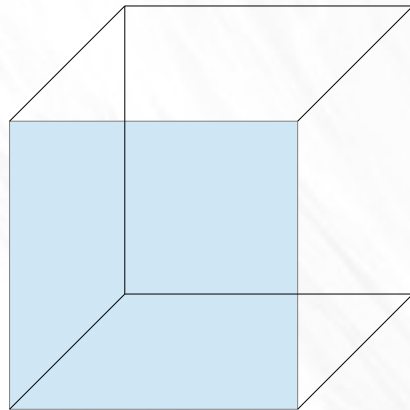
# Доступ к v4l

Video4Linux (v4l) — интерфейс прикладного программирования (API) захвата видео для Linux. Video4Linux тесно интегрируется с ядром Linux. Поддержка v4l появилась в конце цикла развития ядра Linux 2.1.X. Поддерживается большое количество веб-камер и видео устройств — плат видеозахвата, ТВ-тюнеров, плат приема DVB.

- V4L был назван по аналогии с Video For Windows (который иногда сокращенно «V4W»), но технически не связан с ним.
- <http://linuxtv.org/downloads/v4l-dvb-apis/>

# Практическое задание(2014)

- Создать средствами ImageMagick «прозрачный» куб, на гранях которого была бы нанесена ваша фотография, причем три из них разложить как RGB, а «дно» - Gray
- Создать анимацию, чтобы «границы» менялись местами



# Практическое задание 2015

-

# Что почитать

- <http://habrahabr.ru/post/142818/>
- [http://scholar.google.com/scholar\\_host?q=info:wh](http://scholar.google.com/scholar_host?q=info:wh)  
(Воскобойников Колкер 2002 Автометрия)
- <http://www.gphoto.org/doc/remote/>
- <http://habrahabr.ru/post/168517/>