



LOBACHEVSKY  
UNIVERSITY

Information Technology, Mathematics & Mechanics (ITMM) institute  
Software & Supercomputing Technology department

## CS255. Computer Graphics Introduction Course

# Обработка изображений. Часть 2

Турлапов Вадим Евгеньевич  
проф. каф. МОСТ, ИТММ, ННГУ

*По материалам  
Владимира Вежневца (МГУ)*

## Раздел 4. Устранение шума в полутоновых и цветных изображениях

---

- ▶ Усреднение (box filter)
- ▶ Медианный фильтр
- ▶ Фильтр Гаусса (gaussian blurring)
- ▶ Адаптивные фильтры

# Причины и примеры шума изображения

- ▶ Причины возникновения шума:
  - ▶ Несовершенство регистрирующих приборов
  - ▶ Хранение и передача изображений с потерей данных



Шум фотоаппарата



Сильное сжатие JPEG

# Операция «свертка» (convolution)

---

- ▶ Свертка двумерной функции  $f$  по функции  $g$  в непрерывном и дискретном случае.

$$\langle f * g \rangle (i, j) = \sum_{l=n_0}^{n_1} \sum_{k=m_0}^{m_1} f(i-l, j-k) \cdot g(l, k)$$

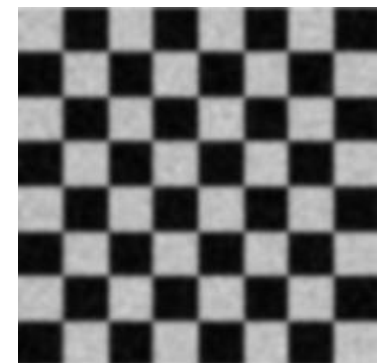
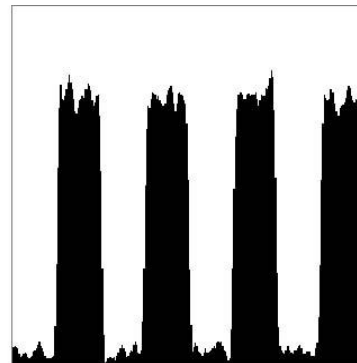
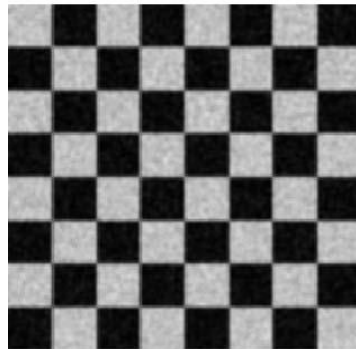
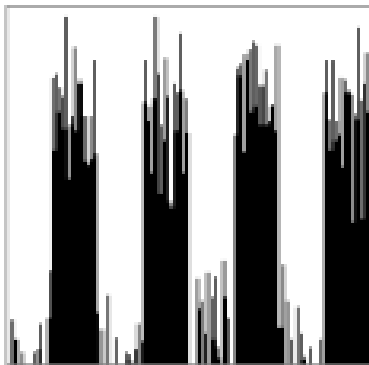
Часто, свертка изображения по какой-либо функции называется применением фильтра к изображению.

# Усреднение (box filter)

Операция усреднения значения каждого пикселя – свертка по константной функции:

$$I'(i, j) = \sum_{l=-n}^n \sum_{k=-m}^m I(i-l, j-k) \cdot \frac{1}{(2n+1)(2m+1)}$$

Результат применения:



# Подавление и устранение шума.

## Медианный фильтр.

---

- ▶ Устранение шума в полутонных, цветных и бинарных изображениях с помощью медианного фильтра - выбор медианы среди значений яркости пикселей в некоторой окрестности.
- ▶ Определение медианы:

$A_i, i = \overline{1, n}$ ; - отсортированный набор чисел,

$A_{[n/2]}$  – медиана набора.

- ▶ Медианный фильтр радиусом  $r$  – выбор медианы среди пикселей в окрестности  $[-r, r]$ .

# Пример очистки изображения с помощью медианного фильтра

---

- ▶ Фильтр с окрестностью  $3 \times 3$



# Быстрая реализация медианного фильтра

---

- ▶ Медианный фильтр считается дольше, чем операция свертки, поскольку требует частичной сортировки массива яркостей окрестных пикселей.
- ▶ Возможности ускорения:
  - ▶ Использовать алгоритмы быстрой сортировки
  - ▶ Конкретная реализация для каждого радиуса (3x3, 5x5)
  - ▶ Не использовать сортировку вообще – считать через гистограмму окрестности точки



# Фильтр Гаусса (gaussian blurring)

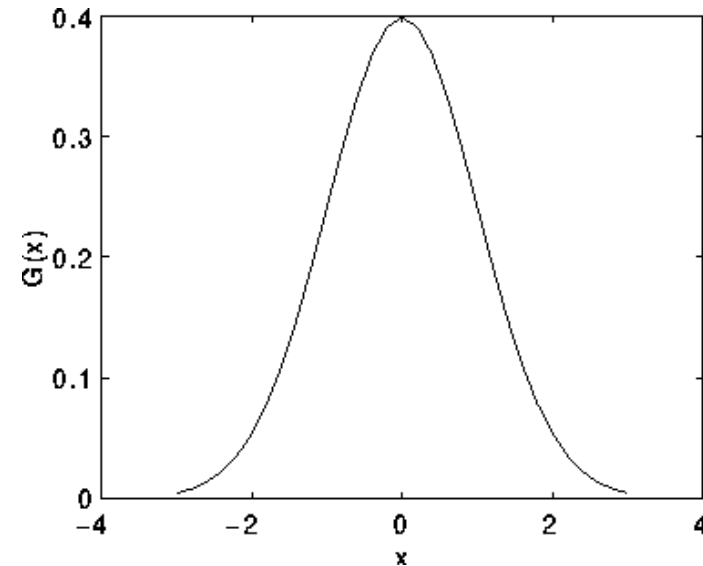
- Двумерный фильтр Гаусса (свертка):

$$I'(i, j) = \sum_{l=-n}^n \sum_{k=-m}^m I(i-l, j-k) \cdot \frac{1}{2\pi\sigma^2} e^{-\frac{d^2}{2\sigma^2}}$$

$$d = \sqrt{l^2 + k^2}$$

$$G(l, k) = \frac{1}{2\pi\sigma^2} e^{-\frac{l^2+k^2}{2\sigma^2}} = G(l) \cdot G(k),$$

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



- На графике функция  $G(x)$ ,  $\sigma = 1$

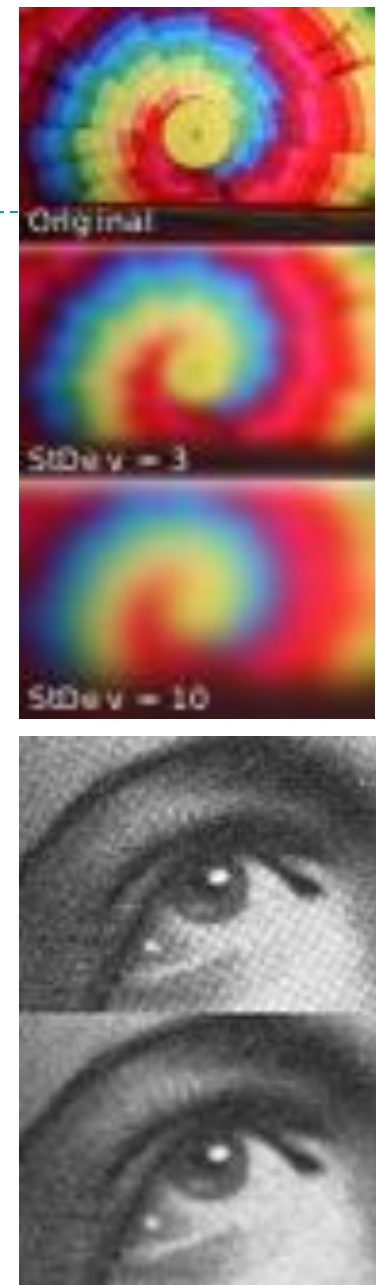
Задание: Оценить размер матрицы фильтра с величиной сигма = 1px, при точности = 0.01

# Фильтр Гаусса (gaussian blurring)

The Gaussian filter kernel with  $\sigma = 0.84089642$  (Wikipedia):

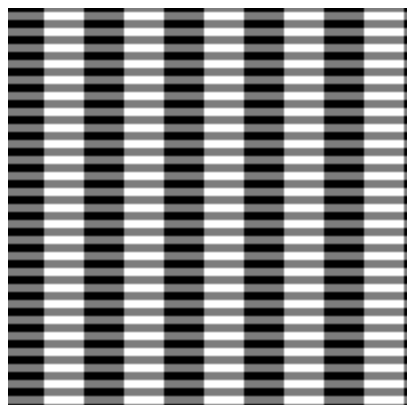
0.00000067	0.00002292	<b>0.00019117</b>	0.00038771	<b>0.00019117</b>	0.00002292	0.00000067
0.00002292	0.00078634	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
<b>0.00019117</b>	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	<b>0.00019117</b>
0.00038771	0.01330373	0.11098164	<b>0.22508352</b>	0.11098164	0.01330373	0.00038771
<b>0.00019117</b>	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	<b>0.00019117</b>
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00000067	0.00002292	<b>0.00019117</b>	0.00038771	<b>0.00019117</b>	0.00002292	0.00000067

This shows how smoothing affects edge detection. With more smoothing, fewer edges are detected (Wikipedia)

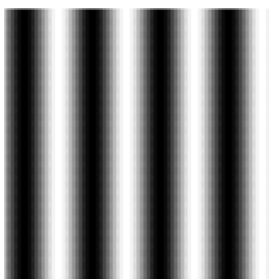


# Фильтр Гаусса (gaussian blurring)

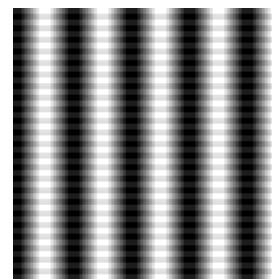
Результаты свертки по функции Гаусса и по константной функции (усреднения).



Исходное изображение



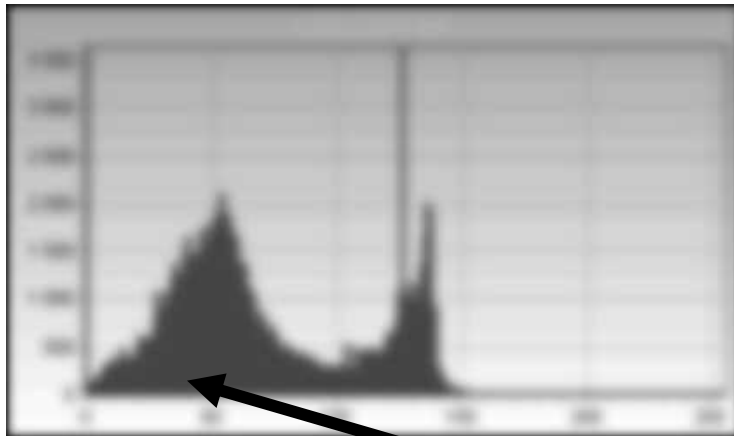
Фильтр Гаусса с  
Sigma = 4



Усреднение по 49  
пикселям (7x7)

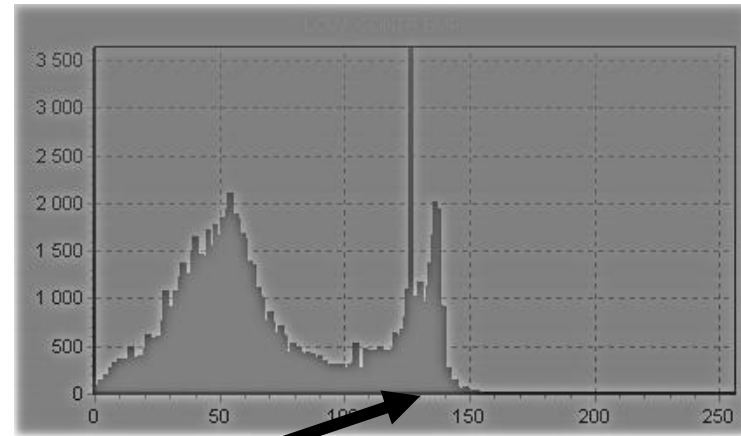
Важное свойство фильтра Гаусса – он по сути является низкочастотным фильтром!

# Преобразование Фурье

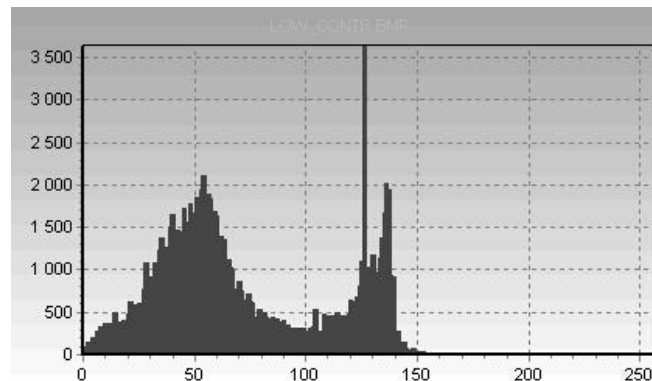


Низкие частоты

+



Высокие частоты



# Адаптивные фильтры

---

- ▶ Что нужно
  - ▶ Размывать шум, резкие границы – сохранять.
  
- ▶ Как этого добиться
  - ▶ Предположение: перепады яркости из-за шума относительно перепадов на резких границах невелики
  - ▶ Алгоритм: При расчете новой яркости усреднять только по тем пикселям из окрестности, которые не сильно отличаются по яркости от обрабатываемого

# Адаптивный фильтр. Программа

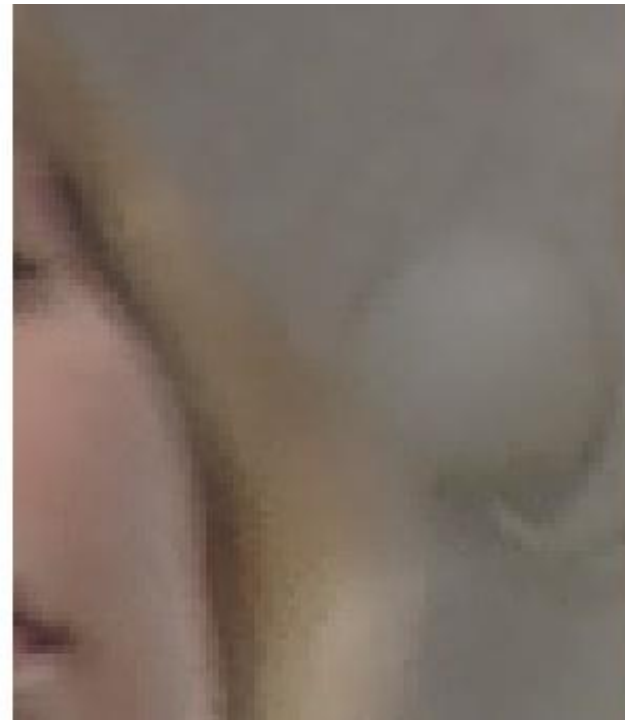
---

```
for (each pixel of the current video frame)
{
    GetRGB (source_pixel, r, g, b);
    tot_red = tot_green = tot_blue = 0;
    count_red = count_green = count_blue = 0;
    for (each pixel in the specified radius)
    {
        GetRGB (neighbour_pixel, r1, g1, b1);
        if (abs(r1-r) < Threshold)
            {tot_red += r1; count_red ++;}
        if (abs(g1-g) < Threshold)
            {tot_green += g1; count_green ++;}
        if (abs(b1-b) < Threshold)
            {tot_blue += b1; count_blue ++;}
    }
    destination_pixel = RGB (tot_red / count_red,
                             tot_green / count_green ,
                             tot_blue / count_blue );
}
```

# Адаптивные фильтры - пример

---

- ▶ Примеры таких фильтров:  
<http://www.compression.ru/video/denoising/denoising.pdf>



# «Продвинутые» фильтры

---

- ▶ Примеры таких фильтров (см. лек. А. Лукина, G&MLab ВМК МГУ)





# В чем отличие разных фильтров?

---

- Box filter (простое размытие) – помимо подавления шума портит резкие границы и размывает мелкие детали изображения
- Gaussian filter – меньше размывает мелкие детали, лучше убирает шум
- Median filter – резких границ не портит, убирает мелкие детали, изображение становится менее естественным
- Адаптивные фильтры – меньше портят детали, зависят от большего числа параметров. Иногда изображение становится менее естественным.
- «Продвинутые» фильтры – лучшее сохранение деталей, меньше размытие. Часто сложны в реализации и очень медленные.

Что лучше – зависит от конкретной задачи

# Как бороться с шумом аппаратуры?

- ▶ Предположим, камера, которой производится съемка заметно «шумит». Обычно шум измерительной аппаратуры моделируется как нормально распределенная случайная величина с нулевым средним -  $Err(i,j)$  ;

$$I(i, j) = g_r(i, j) + Err(i, j);$$

$$\bar{I}(i, j) = \frac{1}{N} \sum_{k=1}^N I_k(i, j);$$

$$E(\bar{I}(i, j)) = g_r(i, j);$$

# Примеры шумоподавления



Зашумленные изображения

Усреднение по 10  
изображениям

Так работают камеры в некоторых сотовых телефонах

# Примеры шумоподавления

---



Исходное изображение



Испорченное  
изображение

# Примеры шумоподавления

---



Усреднение по 9  
пикселям (3x3)



Медианный фильтр  
(3x3)

## Разд.5. Выделение контуров

---

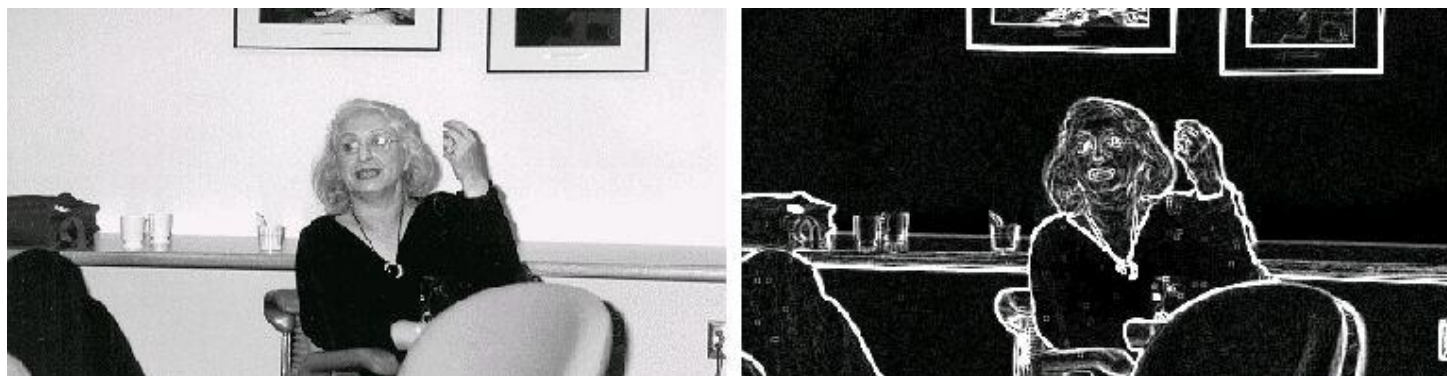
1. Введение
2. Оконтуривание объектов
3. Пример оконтуривания
4. Подчеркивание краев на основе анализа градиента и второй производной яркости

# Выделение контуров объекта

---

Рассмотрим выделение краев (границ)

Край (edge) – резкое изменение яркости на изображении, часто соответствует границам объектов на изображении.



# Операция оконтуривания объекта

При работе с бинарными изображениями контуры объекта можно получить с помощью операций математической морфологии

Внутреннее оконтуривание

▶  $C_i = A - (A (-) B)$

Внешнее оконтуривание

▶  $C_o = (A (+) B) - A$



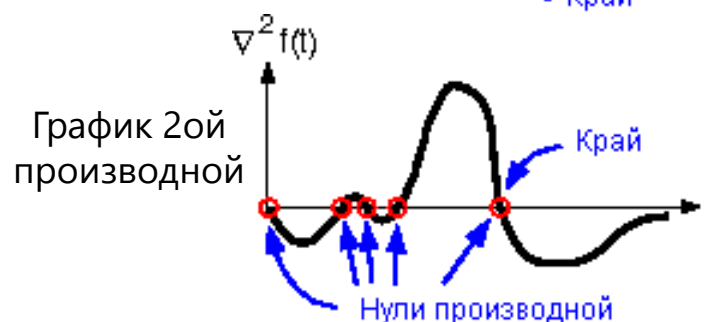
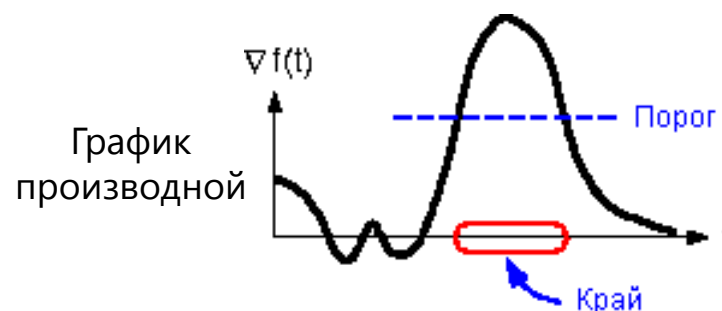
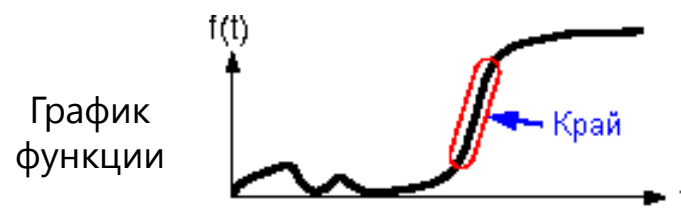


# Выделение точек контура

Нас интересуют области резкого изменения яркости –  
нахождение таких областей можно организовать на основе анализа первой и второй производной изображения.

Матричный  
фильтр

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



# Выделение точек контура. Градиент

- ▶ Наибольшее изменение функции происходит в направлении ее градиента. Величина изменения измеряется абсолютной величиной градиента.

$$\nabla I(x, y) = \left( \frac{\partial I}{\partial x}(x, y), \frac{\partial I}{\partial y}(x, y) \right);$$

$$|\nabla I(x, y)| = \sqrt{\left( \frac{\partial I}{\partial x}(x, y) \right)^2 + \left( \frac{\partial I}{\partial y}(x, y) \right)^2}$$

- ▶ Часто используется приближенное вычисление градиента:

$$|\nabla I(x, y)| \cong \left| \frac{\partial I}{\partial x}(x, y) \right| + \left| \frac{\partial I}{\partial y}(x, y) \right|$$

# Выделение контура. Приближения (маски) Робертса, Превитта и Собеля

- Семейство методов основано на приближенном вычислении градиента, анализе его направления и абсолютной величины. Свертка по функциям:

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

Робертса

**Roberts** cross convolution masks. **Gradient Slope = 45°**

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

Превитта

**Prewitte**

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

Собеля

**Sobel**

- Математический смысл – приближенное вычисление производных по направлению + подавление шума.
- [Дополнительный материал \(eng\)](#)

# Выделение точек контура. Примеры

Примеры применения операторов подчеркивания краев:



Робертса



Превитта



Собеля

# Источники

---

- ▶ Gary Bradski and Adrian Kaehler. Learning OpenCV/ Published by O'Reilly Media, Inc., 2008. - 577pp.
- ▶ Курсы и материалы лаборатории Graphics & Media Lab при ВМиК МГУ (<http://graphics.cs.msu.su/courses/cg/>)
- ▶ Ватолин Д.С. Сжатие изображений. :Изд. МГУ, 1999. -76с. ([local](#))
- ▶ Open Source Computer Vision Library. *Reference Manual*. Copyright © 1999-2001 Intel Corporation. Issued in U.S.A. Order Number: 123456-001 (<http://developer.intel.com>)
- ▶ Рекомендации для начинающих пользователей: PROGRAMMING WITH INTEL IPP (INTEGRATED PERFORMANCE PRIMITIVES) AND INTEL OPENCV (OPEN COMPUTER VISION) UNDER GNU LINUX: A BEGINNER'S TUTORIAL.( [j.Landre@iutlecreusot.u-bourgogne.fr](mailto:j.Landre@iutlecreusot.u-bourgogne.fr))
- ▶ Intel® OPEN SOURCE COMPUTER VISION LIBRARY
- ▶ Image Analysis Cookbook 6.0. <http://www.reindeergraphics.com/foveaprotutorial.html>

## OpenCV

- ▶ \_ Open Computer Vision (OpenCV) sources, download site - <http://sourceforge.net/projects/opencvlibrary>
- ▶ \_ Open Computer Vision (OpenCV) mailing list and group - <http://groups.yahoo.com/group/OpenCV>