



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: Устранение шума в полутоновых и цветных изображениях.

Матричные фильтры

- ▶ 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

Как бороться с шумом аппаратуры?

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



Зашумленные изображения

Усреднение по 10
изображениям

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

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

Матричные фильтры и «свертка» (convolution)

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

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

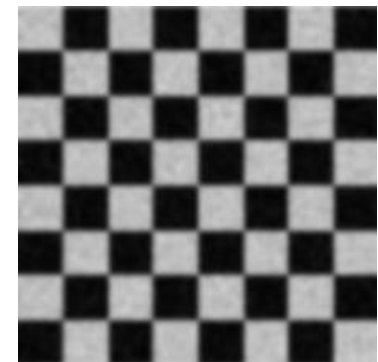
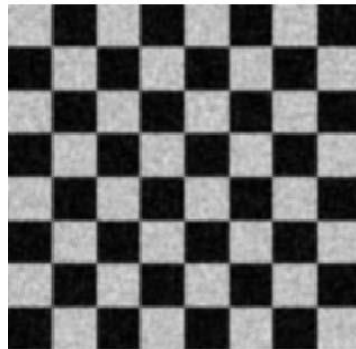
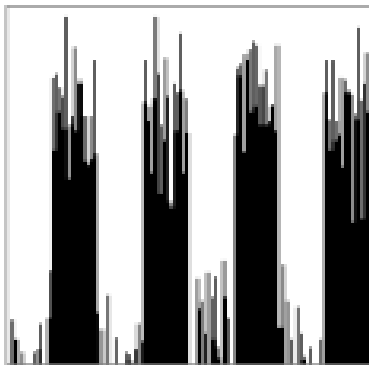
Часто свертка изображения по какой-либо функции называется применением фильтра к изображению.

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 $[-r, r]$ + текущий.
- ▶ Является ли медианный фильтр линейным?

Примеры подавления шума «соль и перец»



Исходное изображение



Испорченное
изображение

Примеры подавления шума «соль и перец»



Усреднение по 9
пикселям (3x3)



Медианный фильтр
(3x3)

Пример очистки изображения с помощью медианного фильтра

- ▶ Фильтр с окрестностью 3×3



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) с наследованием выборки и сортировки предыдущего шага
 - ▶ Не использовать сортировку вообще – считать медиану через гистограмму окрестности точки

Фильтр Гаусса (gaussian blurring)

- Двумерный фильтр Гаусса (свертка):

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

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

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

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

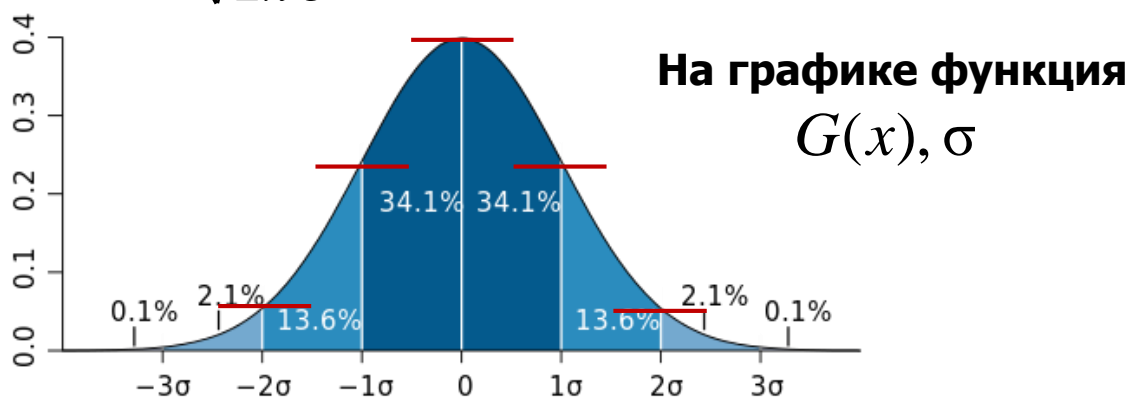
сепарабельность

Вопрос: Как оценить размер матрицы фильтра с величиной сигма = 1px, при точности = 1%?

Пример дискретизации и нормировки матрицы:

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

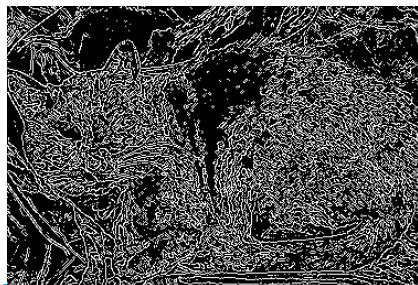


Фильтр Гаусса (gaussian blurring)

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

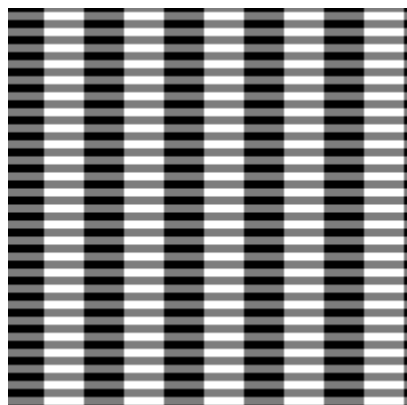
0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067
0.00002292	0.00078634	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00038771	0.01330373	0.11098164	0.22508352	0.11098164	0.01330373	0.00038771
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067

Как сглаживание влияет на детектирование границ: чем больше сглаживание, тем меньше детектировано границ (Wikipedia)

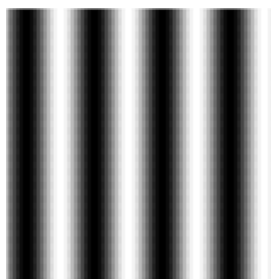


Фильтр Гаусса (gaussian blurring)

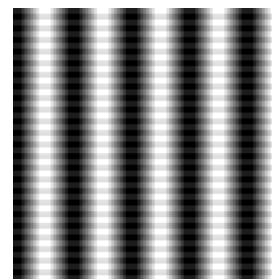
Результаты свертки по функции Гаусса и по константной функции (усреднения).



Исходное изображение



Фильтр Гаусса с
Sigma = 4

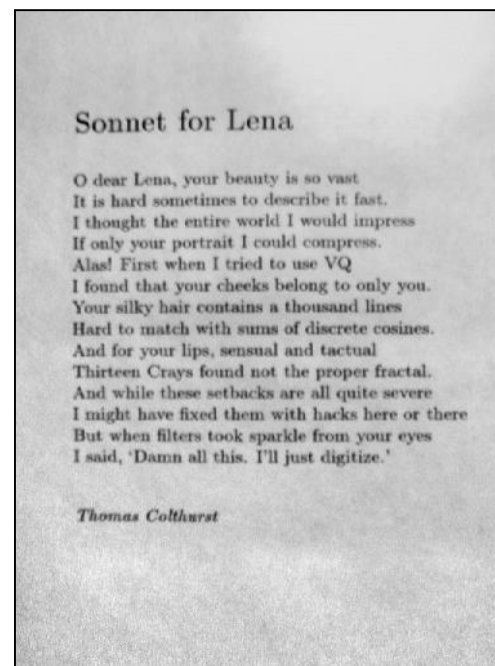
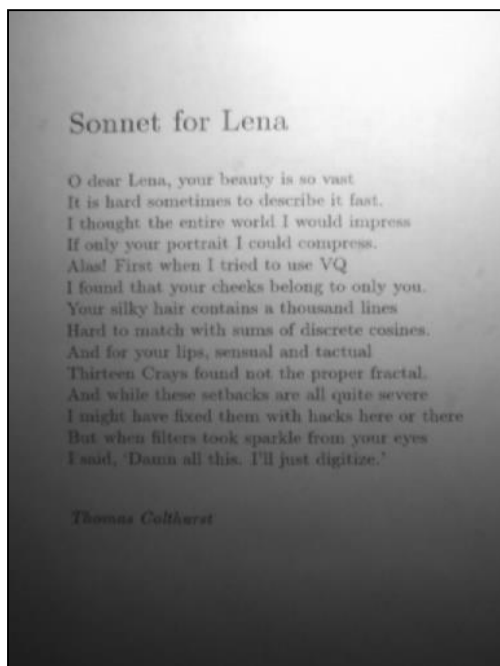


Усреднение по 49
пикселям (7x7)

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

Компенсация разности освещения

Пример



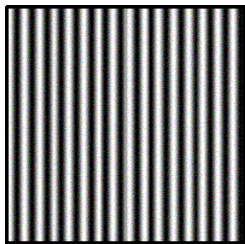
Компенсация разности освещения

Идея:

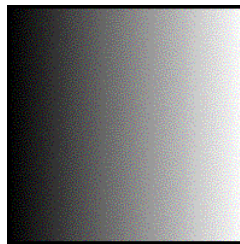
Формирование изображения:

$$I(i, j) = l(i, j) \cdot f(i, j)$$

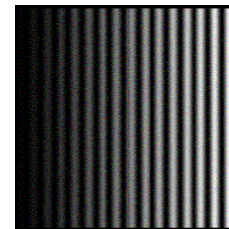
Плавные изменения яркости относятся к освещению, резкие - к объектам.



объект $f(i, j)$



освещение $l(i, j)$



Изображение
освещенного
объекта $I(i, j)$

Выравнивание освещения. Алгоритм

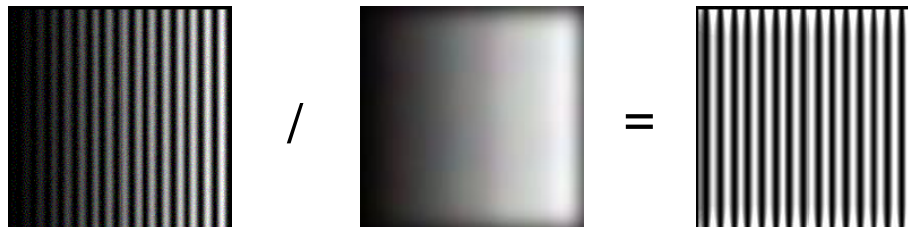
▶ Алгоритм

- ▶ Получить компонент освещения путем применения низкочастотного фильтра G к изображению $I(i,j)$

$$l'(i, j) = I(i, j) * G$$

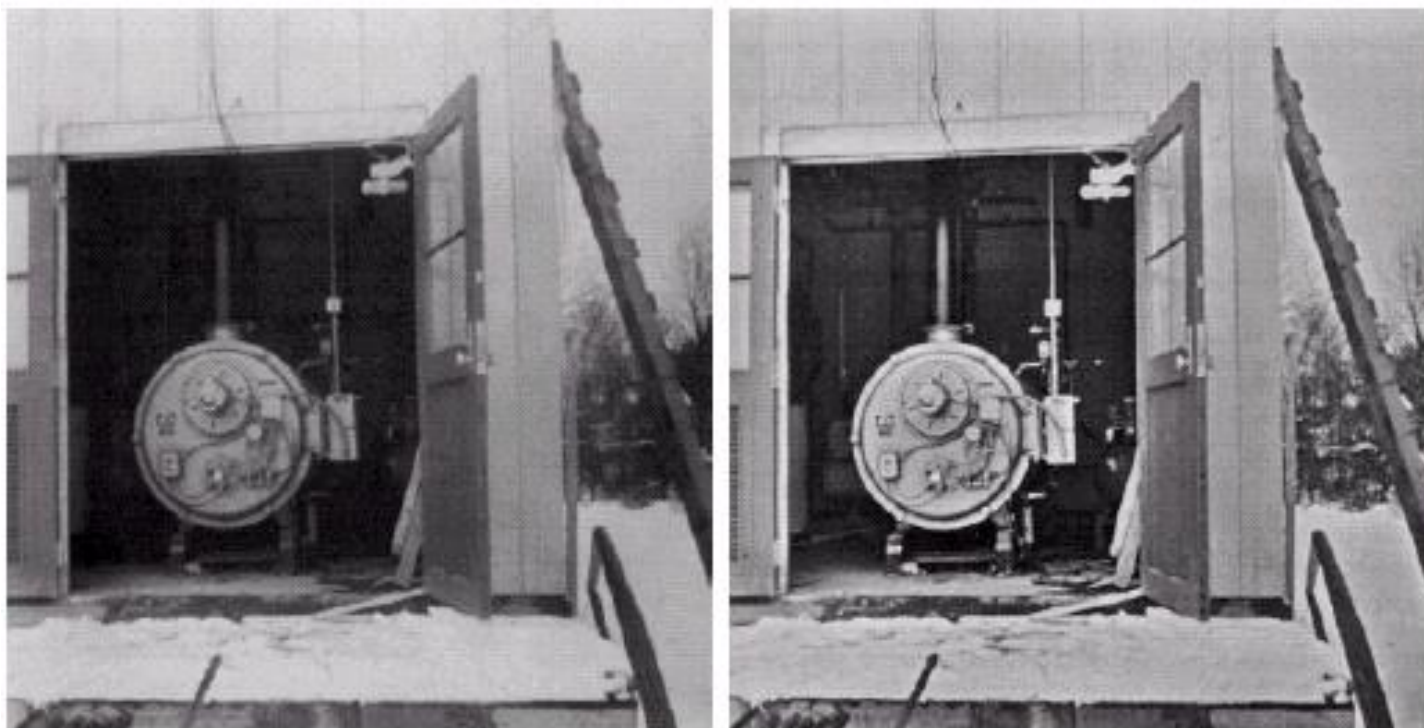
- ▶ Восстановить изображение по формуле

$$f'(i, j) = \frac{I(i, j)}{l'(i, j)}$$



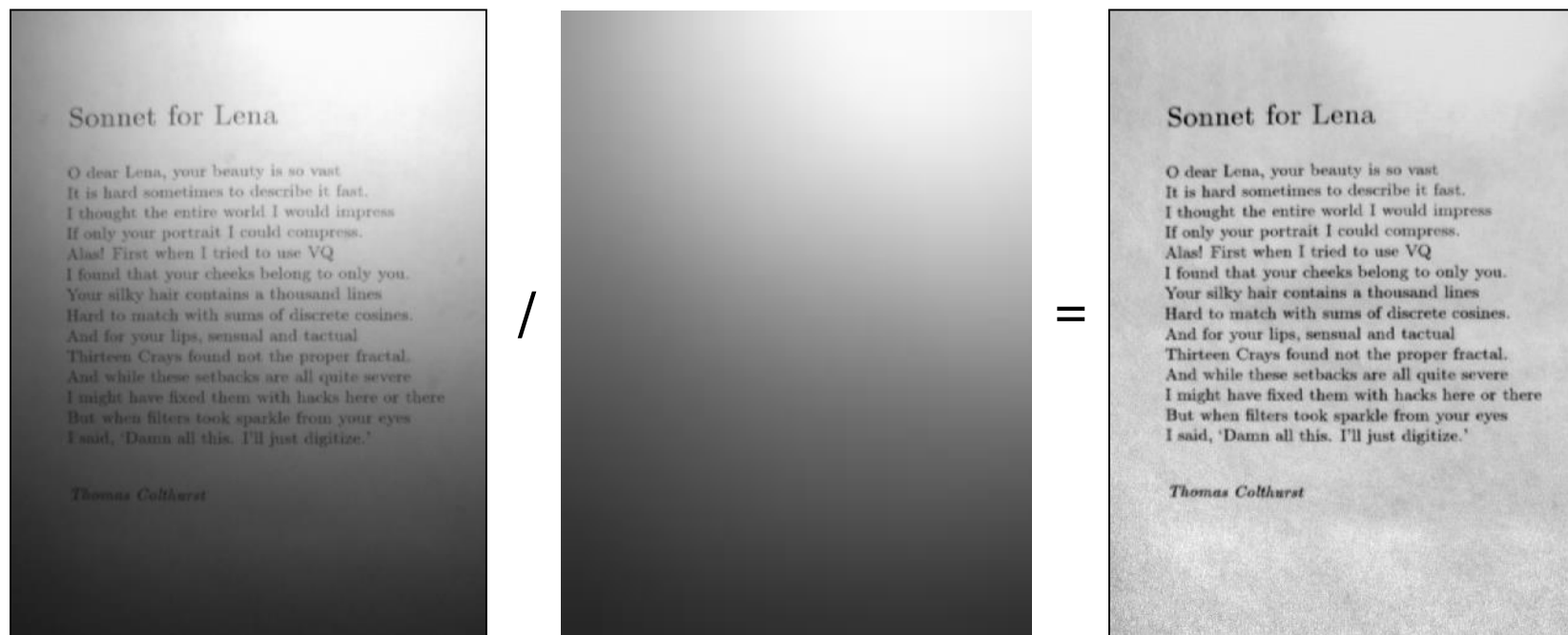
Выравнивание освещения

Пример 1



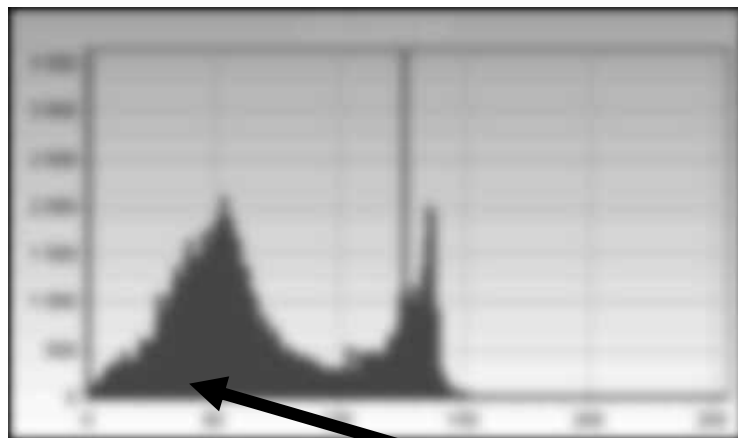
Компенсация разности освещения.

Пример 2



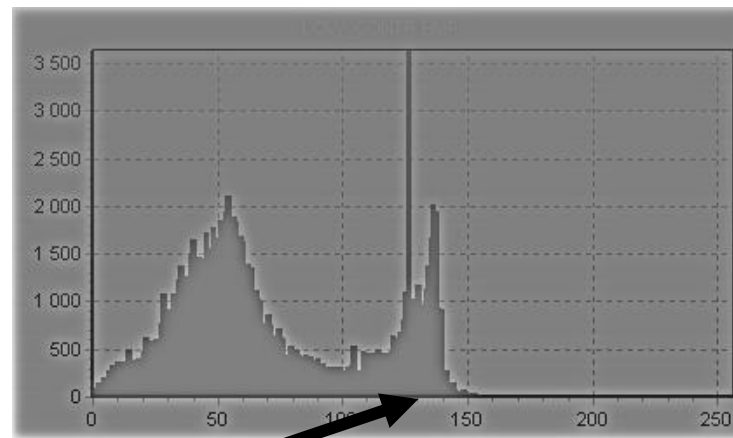
Gauss 14.7 пикселей

Преобразование Фурье

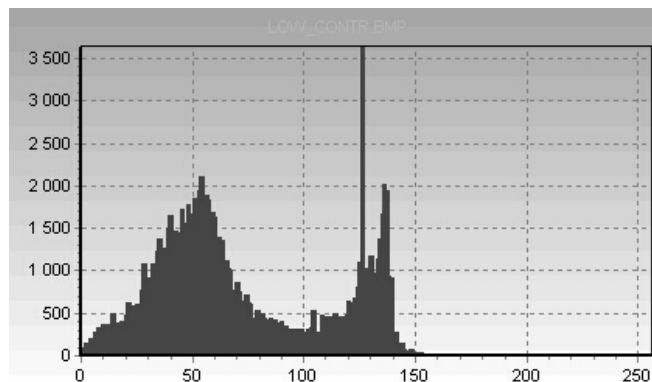


Низкие частоты

+



Высокие частоты



Адаптивные фильтры

- ▶ Что нужно
 - ▶ Размывать шум, резкие границы – сохранять.

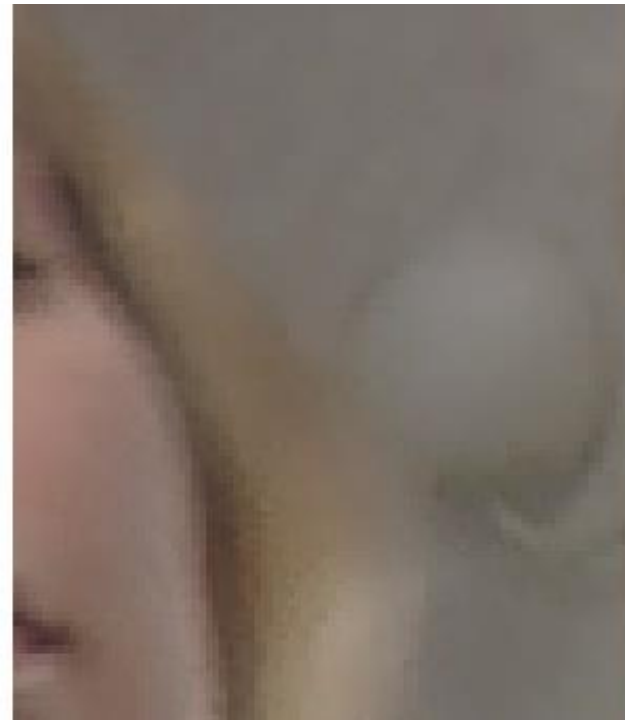
- ▶ Как этого добиться
 - ▶ Предположение: перепады яркости из-за шума относительно перепадов на резких границах невелики
 - ▶ Алгоритм: При расчете новой яркости усреднять только по тем пикселям из окрестности, которые не сильно отличаются по яркости от обрабатываемого

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

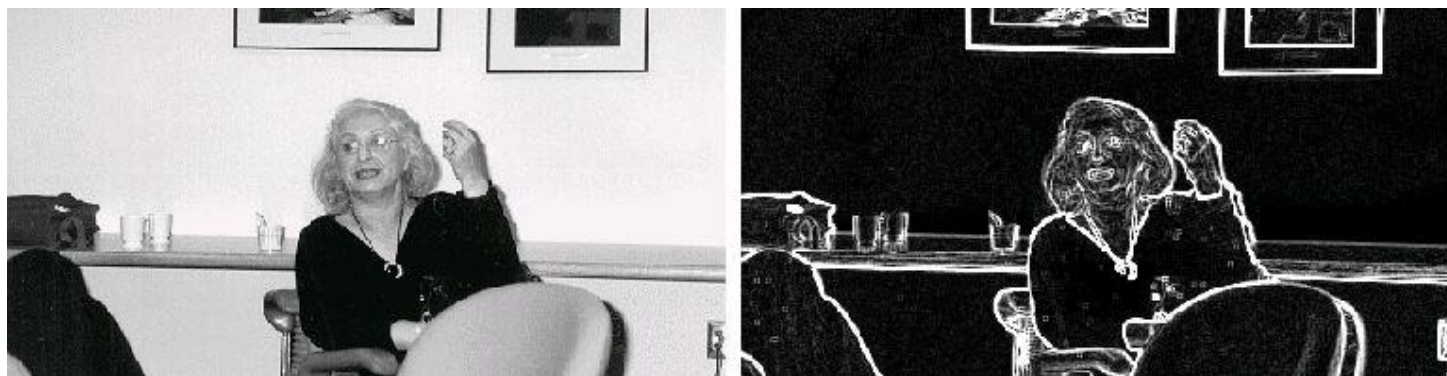
Разд.5. Выделение контуров

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

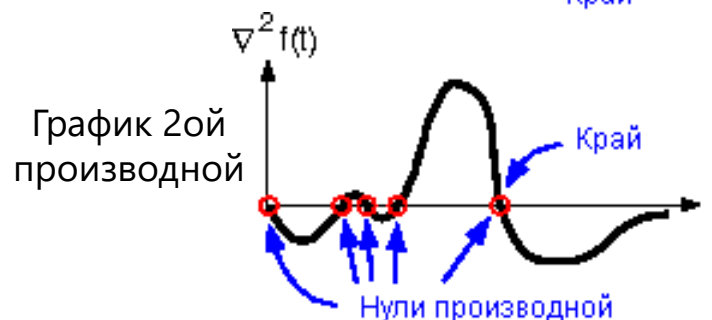
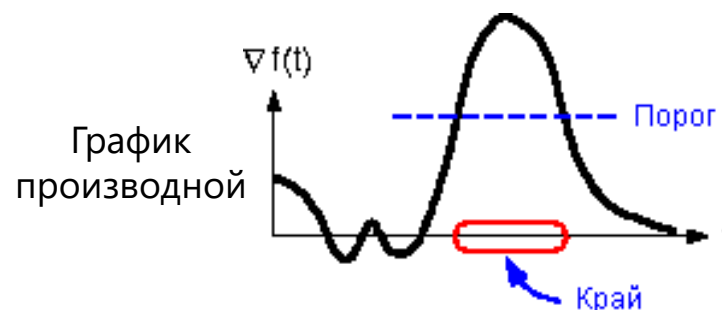
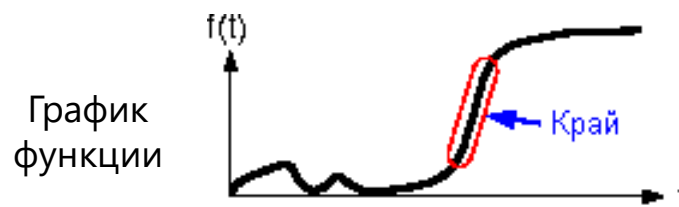
Нас интересуют области резкого изменения яркости – нахождение таких областей можно организовать на основе анализа первой и второй производной изображения.

Матричный фильтр

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

В чем отличие результата?

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



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

Выделение точек контура. Примеры

Примеры применения операторов подчеркивания краев:



Робертса



Превитта



Собеля

Как выполнить фильтрацию на краях изображения?

Ситуация, когда окно фильтра выходит за границы изображения.
Необходимо экстраполировать изображение.

Возможные варианты экстраполяции:

- 1) copy edge –
- 2) clip filter –
- 3) wrap around –
- 4) reflect across edge –
- 5) линейная экстраполяция за границу –

Какой вариант лучше?

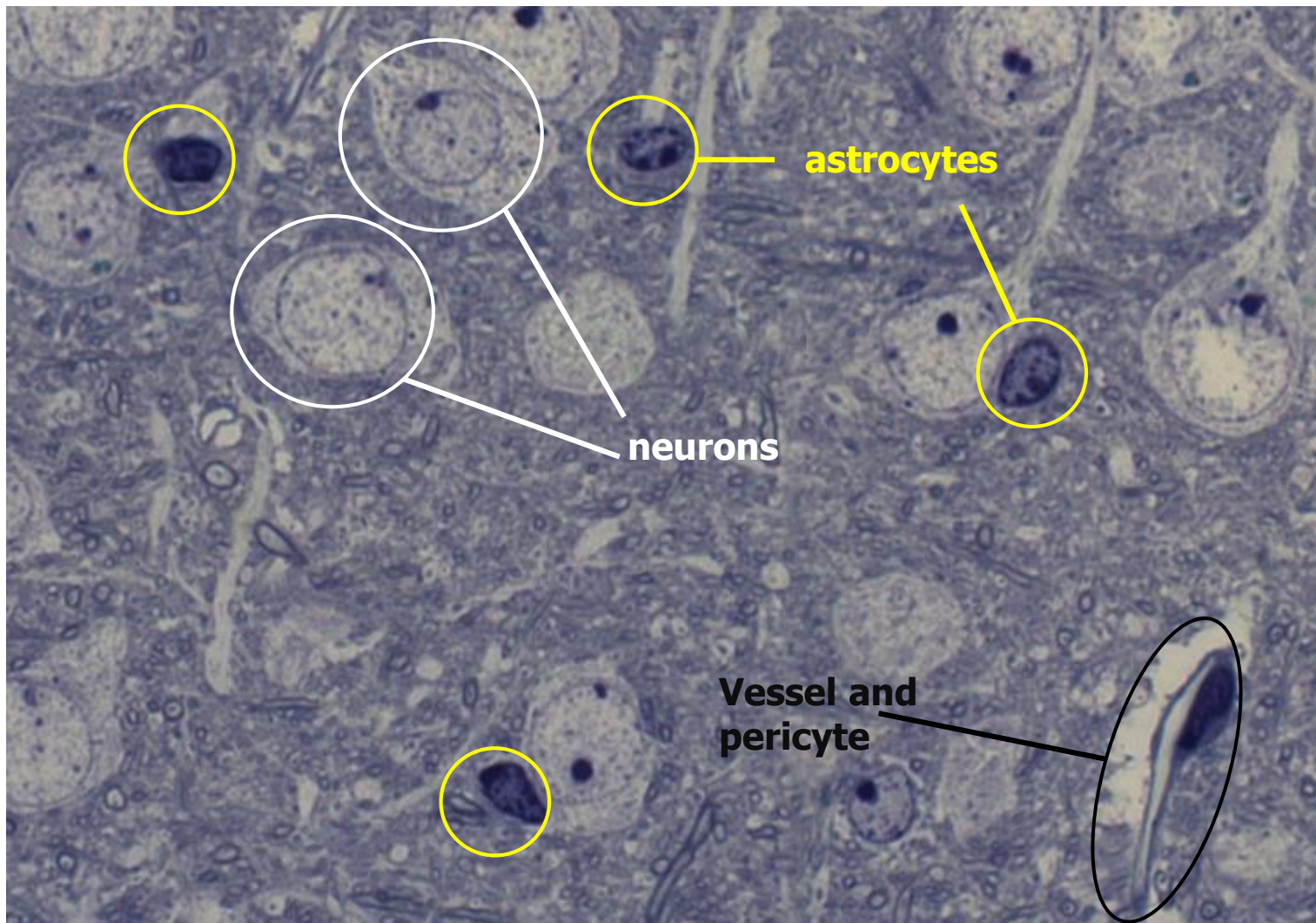
Как выполнить фильтрацию на краях изображения?

Ситуация, когда окно фильтра выходит за границы изображения. Необходимо экстраполировать изображение.

Возможные варианты экстраполяции:

- 1) clip filter – прекращает действие фильтра за границей (black), возможны существенные искажения
- 2) **copy edge – наиболее популярен, но точность оценки производной - 1-го порядка малости (хуже 2-го порядка)**
- 3) reflect across edge – в среднем хуже, чем 1)
- 4) wrap around – круговое замыкание изображения – дает искажения, если не соответствует реальности
- 5) **линейная экстраполяция за границу – наименьшие искажения, но практически не применяется из-за необходимости дополнительных вычислений**

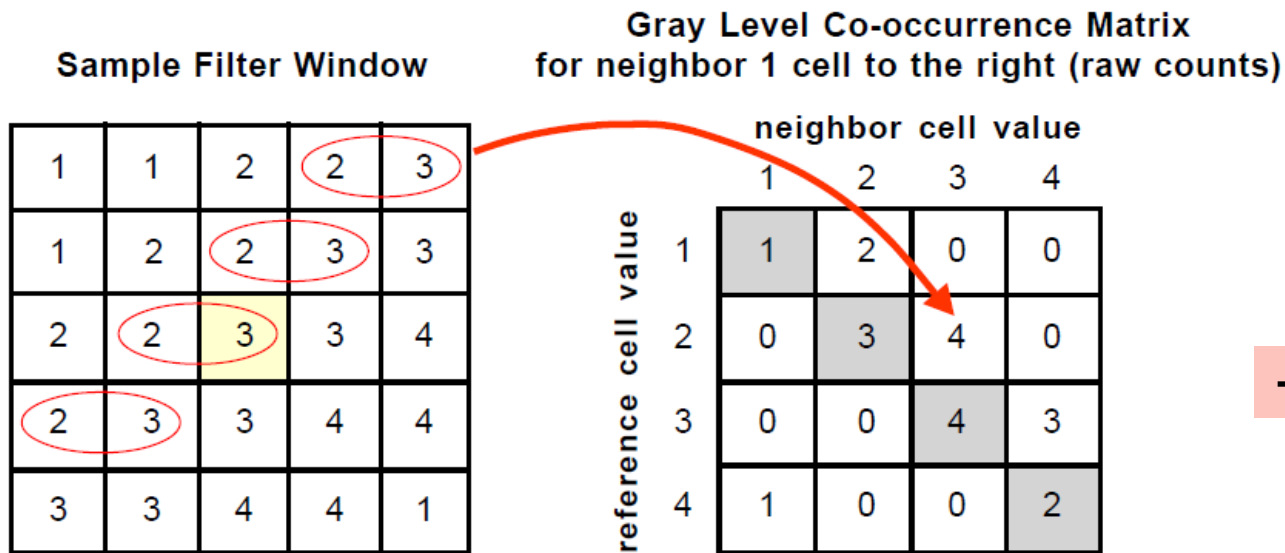
Сегментация и классификация объектов



Пиксельные (текстурные) методы сегментации.

Сегментация по статистическим текстурным признакам.

Gray Level Co-occurrence Matrix. Формирование GLCM



+ Нормировка

Sample 5 x 5 filter window showing image gray level values ranging from 1 to 4. Red ellipses outline four instances where value 3 occurs one cell to the right of value 2. The filter value will be written to the center cell position (yellow background) in the filtered image.

Gray Level Co-occurrence Matrix for the filter window at left showing counts of neighbor relations for different reference cell values (rows) and right-neighbor cell values (columns). Note count of 4 for matrix location [2,3] and different count (0) for location [3,2]; there are no instances where value 2 occurs to the right of value 3. Matrix cells on main diagonal (shaded) register counts where reference and neighbor gray level values are the same.

Haralick R.M., Shanmugam L., Dinstein I. Textural Features for Image Classification // IEEE Trans. on Systems, Man and Cybernetics, 1973. Vol. SMC-3, PP. 610–621. <http://haralick.org/journals/TexturalFeatures.pdf>



Правила формирования GLCM. Симметрия

<http://haralick.org/journals/TexturalFeatures.pdf>

Выбор числа уровней серого? Нужна ли предобработка?

GLCM Filters

Selecting Gray Level Co-occurrence Matrix from the Type menu on the Spatial Filter window provides a choice of eight filters on the Filter menu. Results of each of the filter types for a 3 by 3 filter window with invariant direction and distance = 1 are shown below for the grayscale image illustrated to the right, which has both smooth and rough areas. The GLCM filters fall into three conceptual subsets: Contrast, Order-Disorder, and Statistics.



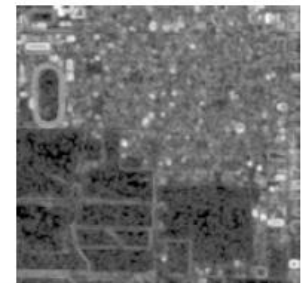
Sample Image



Angular
second
moment



Contrast



Dissimilarity



GLCM Mean



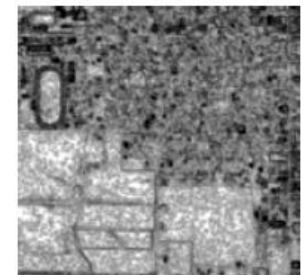
GLCM Variance



GLCM Correlation



Entropy



Homogeneity



Текстурные и объектные признаки

Центральные моменты. Ориентация главной оси инерции объекта

$$M_{ij} = \sum_{(x,y) \in \text{Reg}} x^i y^j I(x, y); \quad M_{00} = \sum_{(x,y) \in \text{Reg}} I(x, y);$$

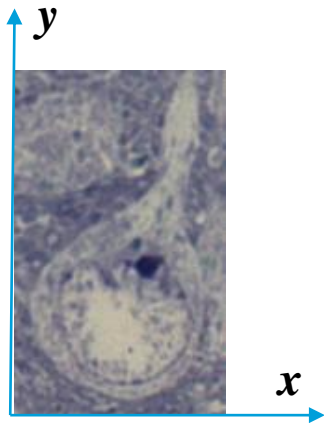
- сырые моменты,
 I – изображение в оттенках серого

$$m_{ij} = \sum_{(x,y) \in \text{Reg}} (x - \bar{x})^i (y - \bar{y})^j I(x, y);$$

- центральные моменты

$$\bar{x} = M_{10}/M_{00} \quad \bar{y} = M_{01}/M_{00}$$

- координаты центра изображения I
или центра анализируемого объекта



$$\theta = \frac{1}{2} \arctan \left(\frac{2m_{11}}{m_{20} - m_{02}} \right)$$

- ориентация главной
оси инерции объекта

θ - не является инвариантом к повороту, но предоставляет полезную в ряде случаев информацию об ориентации объекта.

В чем различие между текстурными и объектными признаками?



Инвариантные характеристики области интереса (объекта)

Для распознавания объектов нас чаще интересуют характеристики инвариантные по отношению к аффинным преобразованиям (масштабированию, переносу, повороту):

- **Удлиненность, нецентрированность (эксцентриситет):**

$$Elongation = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}$$

- **Компактность:**

$$C = \frac{P^2}{A} \quad - \text{отношение квадрата периметра к площади}$$

- **Scale invariants:**

$$\mu_{ij} = \frac{m_{ij}}{m_{00}^{(1+(i+j)/2)}}$$

Invariants μ_{ij} with respect to both translation and scale can be constructed from central moments by dividing through a properly scaled zero-th central moment:

where $i + j \geq 2$. **Note that translational invariance directly follows by only using central moments.**

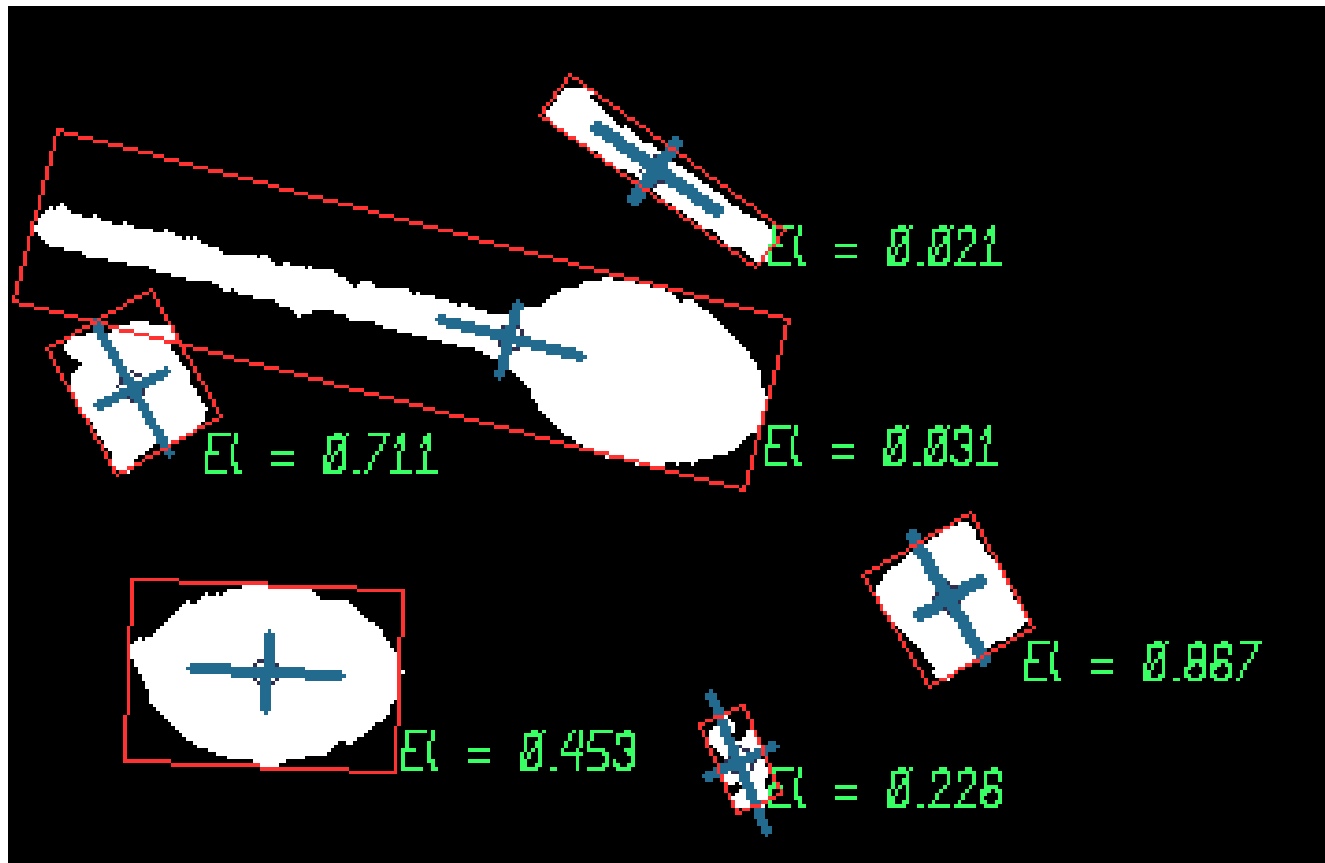
$$M_{ij} = \sum_{(x,y) \in Reg} x^i y^j I(x, y); \quad M_{00} = \sum_{(x,y) \in Reg} I(x, y);$$

$$m_{ij} = \sum_{(x,y) \in Reg} (x - \bar{x})^i (y - \bar{y})^j I(x, y);$$

$$\bar{x} = M_{10}/M_{00} \quad \bar{y} = M_{01}/M_{00}$$



Пример изображения с подсчитанными характеристиками областей



Инварианты вращения (Hu moment invariants)

$$M_1 = m_{20} + m_{02}$$

$$M_2 = (m_{20} - m_{02})^2 + 4m_{11}^2$$

$$M_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

$$M_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$$

$$M_5 = (m_{30} - 3m_{12})(m_{30} + m_{12})[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2]$$

$$+ (3m_{21} - m_{03})(m_{21} + m_{03})[3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2]$$

$$M_6 = (m_{20} + m_{02})[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] +$$

$$+ 4m_{11}(m_{30} + m_{12})(m_{03} + m_{21})$$

$$M_7 = (3m_{21} - m_{03})(m_{12} + m_{30})[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2]$$

$$- (m_{30} - 3m_{12})(m_{21} + m_{03})[3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2]$$

$$M_{ij} = \sum_{(x,y) \in Reg} x^i y^j I(x, y); \quad M_{00} = \sum_{(x,y) \in Reg} I(x, y);$$

$$m_{ij} = \sum_{(x,y) \in Reg} (x - \bar{x})^i (y - \bar{y})^j I(x, y);$$

$$\bar{x} = M_{10}/M_{00} \quad \bar{y} = M_{01}/M_{00}$$

Hu, M.-K. Visual Pattern Recognition by Moment Invariants" // IRE Trans. on Information Theory. 1962. V. IT-8, pp.179-187
<http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Hu.pdf>

Wikipedia. Image moment (https://en.wikipedia.org/wiki/Image_moment) – еще больше инвариантов.



Источники

- ▶ Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений/ Уч.пособие ИТМО. -2008. -192с. (<http://books.ifmo.ru/file/pdf/398.pdf>)
- ▶ А.Бовырин, П.Дружков, В.Ерухимов, Н.Золотых, В.Кустикова, И.Лысенков, И.Мееров, В.Писаревский, А.Половинкин, А.Сысоев. Академия Intel: Разработка мультимедийных приложений с использованием библиотек OpenCV и IPP. (<http://www.intuit.ru/studies/courses/10622/1106/info>)
- ▶ 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.ru>)
- ▶ 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)
- ▶ 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>