

ЛЕКЦИЯ 3. ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

Демидов Д.В.

Обработка аудиовизуальной информации.
Бакалавры 6 семестр. Магистры, 9 семестр

План лекции

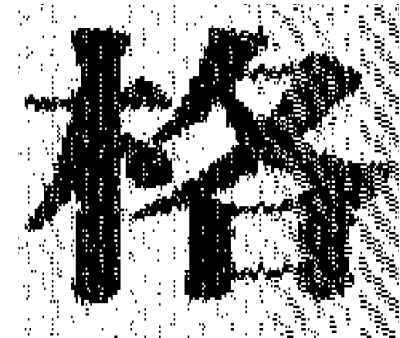
2

- Модели шума
- Фильтрация изображений. Линейные и нелинейные фильтры. Пороговая фильтрация.
- Оконная фильтрация бинарных изображений.
- Логический фильтр. Медианный фильтр. Ранговый фильтр.
- Фильтр «Стирание бахромы».
- Морфологические операции.

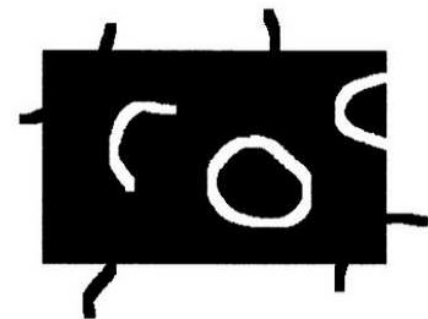
Модели шума бинарного изображения

3

- *Соль и перец.* Шум типа «соль и перец» может быть описан моделью, в которой замещение 1 на 0 происходит с вероятностью p и замещение 0 на 1 с вероятностью q .



- *Дырки и выступы.* Под шумом «дырки и выступы» на бинарном изображении понимают внутренние и внешние дефекты объектов, имеющие некоторую неслучайную форму.



Линейные и нелинейные фильтры

4

- Задача фильтрации состоит в восстановлении исходных значений яркости пикселей в ситуации, когда шумовая компонента является неизвестной случайной величиной.
 - Линейные фильтры принимают во внимание только яркость рассматриваемого пикселя.
 - Нелинейные фильтры учитывают яркость пикселей в окрестности рассматриваемого.

Оконная фильтрация

5

- Для оценки шумовой компоненты необходимо использовать статистическое оценивание, предусматривающее рассмотрение не одного пикселя, а целой выборки близлежащих пикселей (окна).
- Выборка близлежащих пикселей обычно осуществляется из прямоугольного окна с рассматриваемым пикселем в центре (базовый пиксель).
- Учитываемые при принятии решения пиксели образуют *апертуру окна*.
- В типовой процедуре оконной фильтрации окно последовательно движется (скользит) по всей площади изображения с расчетом интенсивности пикселя выходного изображения $Y[i,j]$.
- Расчёт осуществляется на основе интенсивностей пикселей входного изображения $X[i,j]$, включённых в апертуру, по некоторому *правилу принятия решения*.

6

Фильтрация бинарных изображений

Логический, медианный, ранговый фильтр

Взвешенный ранговый фильтр

Разреженная маска, стирание бахромы

Логическая фильтрация

7

- Апертурой логического фильтра является проколота окрестность базового пикселя 3x3, включающая 8 соседних пикселей:

$$\Omega_{ij} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- Решающее правило выглядит следующим образом:

$$y_{ij} = \begin{cases} 1, & \forall x \in \Omega_{ij} \rightarrow x = 1 \\ 0, & \forall x \in \Omega_{ij} \rightarrow x = 0 \\ x_{ij}, & otherwise \end{cases}$$

Логическая фильтрация. Примеры

8

- Такой фильтр справляется с изолированными пикселями-артефактами и практически бесполезен при более интенсивных шумах.



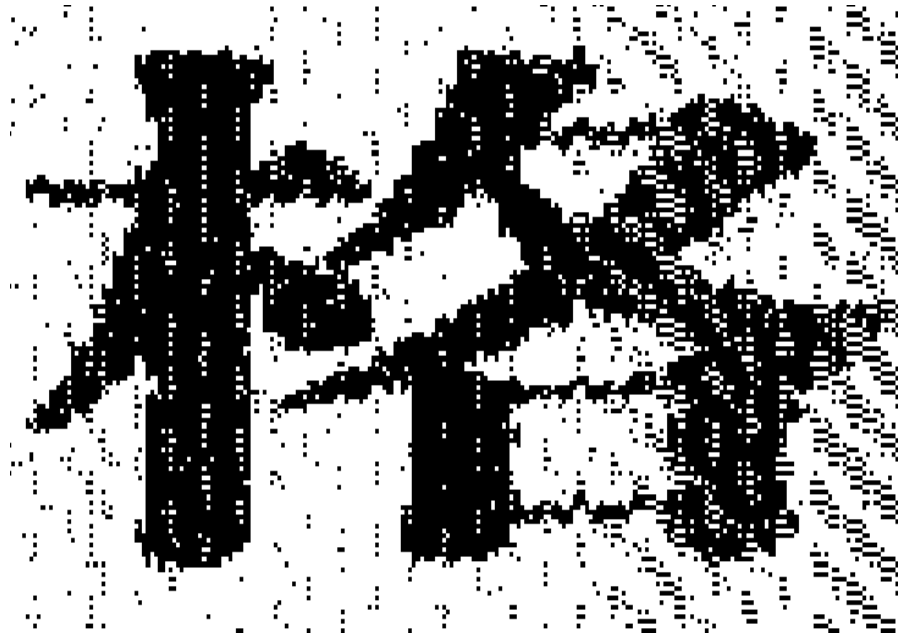
Медианная фильтрация

9

- Для некоторой апертуры с нечётным количеством элементов измеряются интенсивности входящих в неё пикселей.
- Решающее правило медианного фильтра: значение результирующего пикселя выбирается равным 1, если в апертуре больше единиц, чем нулей (число единиц больше половины апертуры), и равным 0 в противном случае.

Медианная фильтрация. Примеры

10



Медианная фильтрация.

Особенности

11

- С увеличением размера окна апертуры растёт способность медианного фильтра подавлять шумовые точки [Бондаренко, 2010, стр.117], но очертания объектов становятся сильно искажёнными.
- Использование медианы подходит для случаев, когда вероятности шума «соль и перец» удовлетворяют соотношению $p=q<0,5$.

Ранговый фильтр

12

- Для рангового (процентильного фильтра) выбирается некоторая константа (ранг) k , меньшая, чем число элементов n в апертуре.
- Решающее правило рангового фильтра: значение результирующего пикселя выбирается равным 1, если в апертуре количество единиц не меньше k ($k_1 \geq k$), и равным 0 в противном случае ($k_0 \geq n + 1 - k$).

Ранговый фильтр. Примеры

13

□ Ранг 2/9



□ Ранг 4/9



□ Ранг 5/9



□ Ранг 7/9



□ Ранг 9/9



Ранговый фильтр. Особенности

14

- Фильтрацию со значением ранга меньше медианы следует применять, если вероятность перехода $p(1 \rightarrow 0)$ существенно больше вероятности перехода $q(0 \rightarrow 1)$ (много соли, мало перца) и наоборот. Для рассматриваемого изображения эти вероятности примерно одинаковы.
- Видно, что фильтры с малыми рангами усиливают соль внутри глифов, практически убирают перец снаружи и сильно портят сами глифы, удаляя значимые черные точки. Фильтры с большими рангами сохраняют глифы, практически полностью удаляют соль, но резко усиливают черный шум (происходит сгущение).

Взвешенный ранговый фильтр

15

- Если более близким точкам окрестности придать больший вес, то они будут оказывать большее влияние на результат. Это повысит устойчивость фильтрации.
- Для этого вместо значений 0 и 1 в апертуре необходимо задавать веса элементов апертуры.
- Например, следующая апертура реализует принцип «штрафа за удаление»:

$$\Omega_{ij} = \begin{matrix} & 1 & 2 & 1 \\ 2 & 2 & 4 & 2 \\ & 1 & 2 & 1 \end{matrix}$$

Взвешенный ранговый фильтр (2)

16

- Модифицированные значения k и n для взвешенного рангового фильтра рассчитываются следующим образом:

$$k = \sum_{x_{ij} \in \Omega} \omega_{ij} x_{ij} \quad n = \sum_{x_{ij} \in \Omega} \omega_{ij}$$

- Для апертуры на предыдущем слайде $n = 16$
- Решающее правило взвешенного рангового фильтра аналогично ранговому с указанной трактовкой параметров k и n .

Взвешенный ранговый фильтр.

Примеры

17

□ Исходное



□ Ранг 10/16



□ Ранг 14/16



Взвешенный ранговый фильтр. Разреженная маска

18

□ Исходное:



□ Апертура:

$$\Omega_{ij} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

□ Ранг 3/5



Фильтр «стирание бахромы»

19

- ❑ Фильтр направлен на удаление неровностей границ объекта, которые мешают его распознаванию по контурному признаку. Неровности границ символа мешают правильно определить его размер, искажают его образ, что мешает распознаванию по контурному признаку.
- ❑ Алгоритм предусматривает последовательное удаление изолированных и крайних элементов, а остальные элементы переносятся в результирующее изображение.

Фильтр «стирание бахромы»

20

- Изолированный пиксель распознается матрицей:

0	0	0
0	1	0
0	0	0

- Крайний верхний пиксель – пиксель, не имеющий соседей сверху, слева и справа. Варианты апертуры:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0	1	1	1	0	0	1	1	1	1	1

- Поворачивая эти варианты на 90, 180 и 270 градусов получают апертуры для крайнего левого, крайнего нижнего и крайнего правого пикселя.

Фильтр «стирание бахромы».

Примеры

21

- ❑ фильтр бахромы из единиц, а также соль внутри глифов:
- ❑ фильтр бахромы из нулей, а также перец внутри глифов:
- ❑ смешанный фильтр белой и чёрной бахромы:



Фильтрация полутоновых изображений

Пространственное и консервативное сглаживание

Фильтр преобладающего оттенка

Фильтр уменьшения контрастности

Гауссово размытие и фильтр Габора

Пространственное сглаживание

23

- Суммируются элементы окна сглаживания, перемноженные на соответствующие коэффициенты маски фильтрации и результат делится на сумму коэффициентов маски фильтрации.
- При одинаковых весовых коэффициентах результат равен среднему арифметическому яркости окружающих пикселей.
- Формула регуляризации:
$$g_W(i, j) = \frac{1}{S_W} \sum_{(m, n) \in W} g(m, n)$$
 - ▣ Здесь S – площадь окна W вокруг точки (i, j) ,
 - ▣ $g(i, j)$ – дискретная функция интенсивности

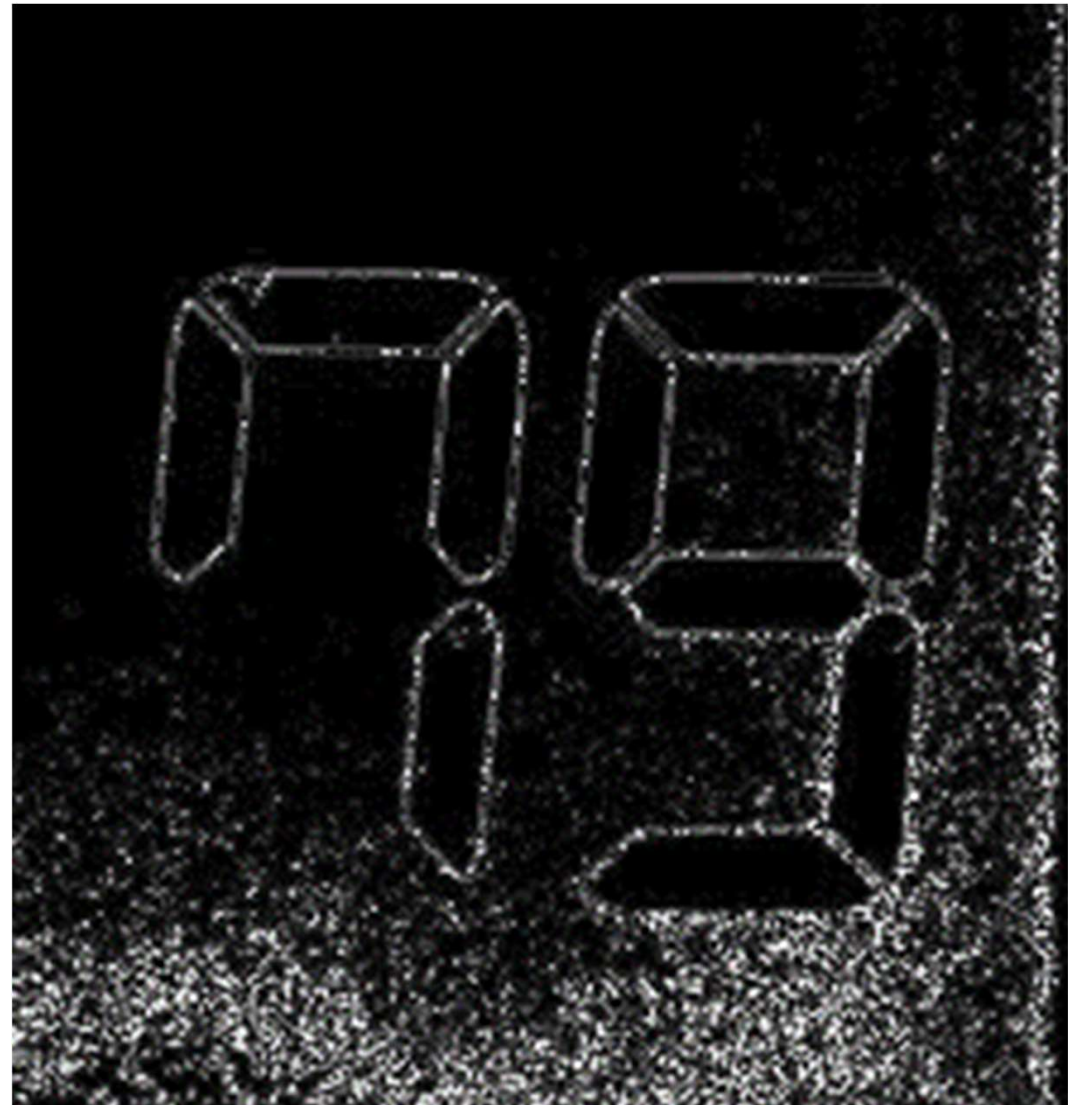
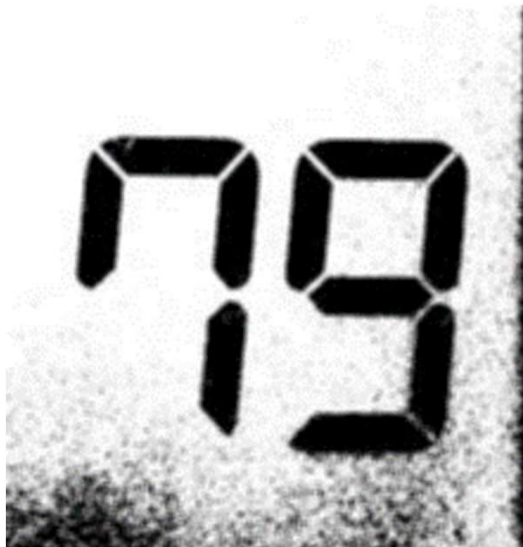
Примеры

24



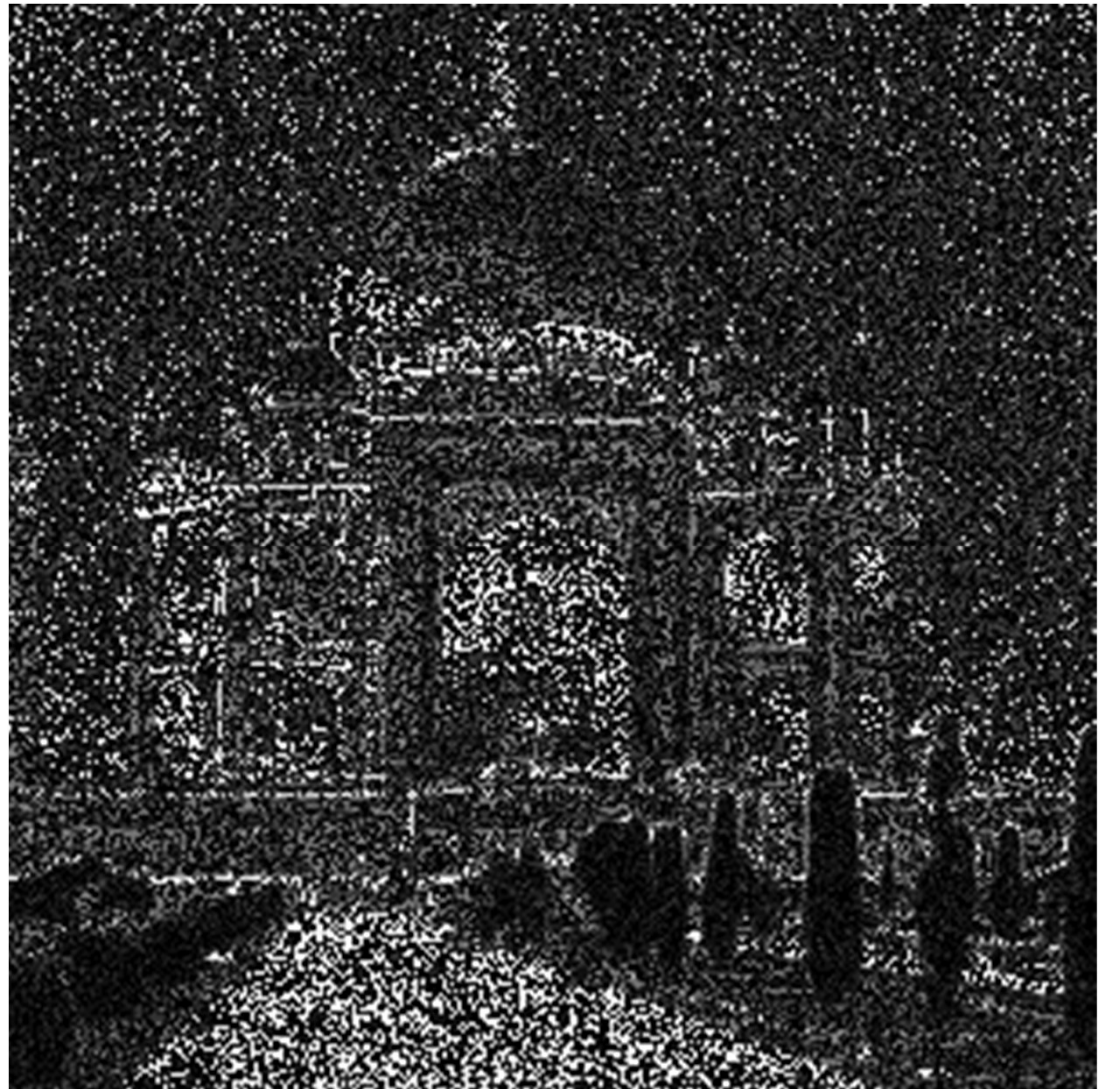
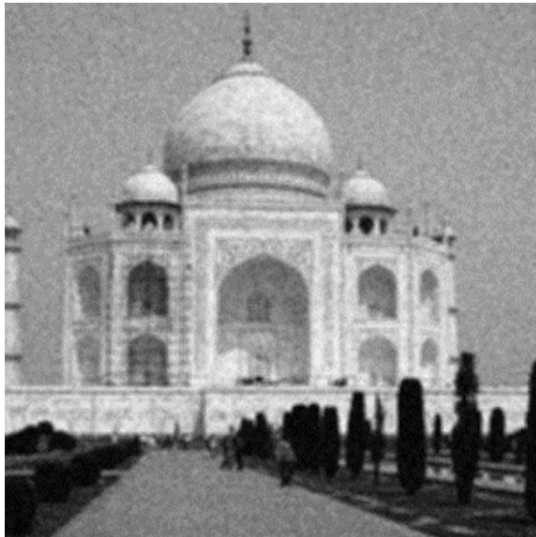
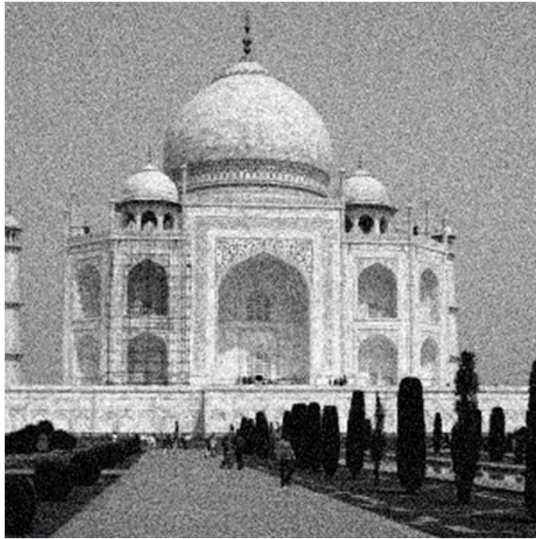
Примеры

25



Примеры

26



Консервативное сглаживание

27

- Среди элементов окна фильтрации отыскивается максимальный и минимальный элементы по яркости.
- Если центральный элемент окна фильтрации
 - не является ни максимальным, ни минимальным элементом, то никаких действий не выполняется.
 - является максимальным по яркости, то среди остальных элементов отыскивается максимальный и помещается на место центрального элемента.
 - является минимальным по яркости, то среди остальных элементов отыскивается минимальный и помещается на место центрального элемента.

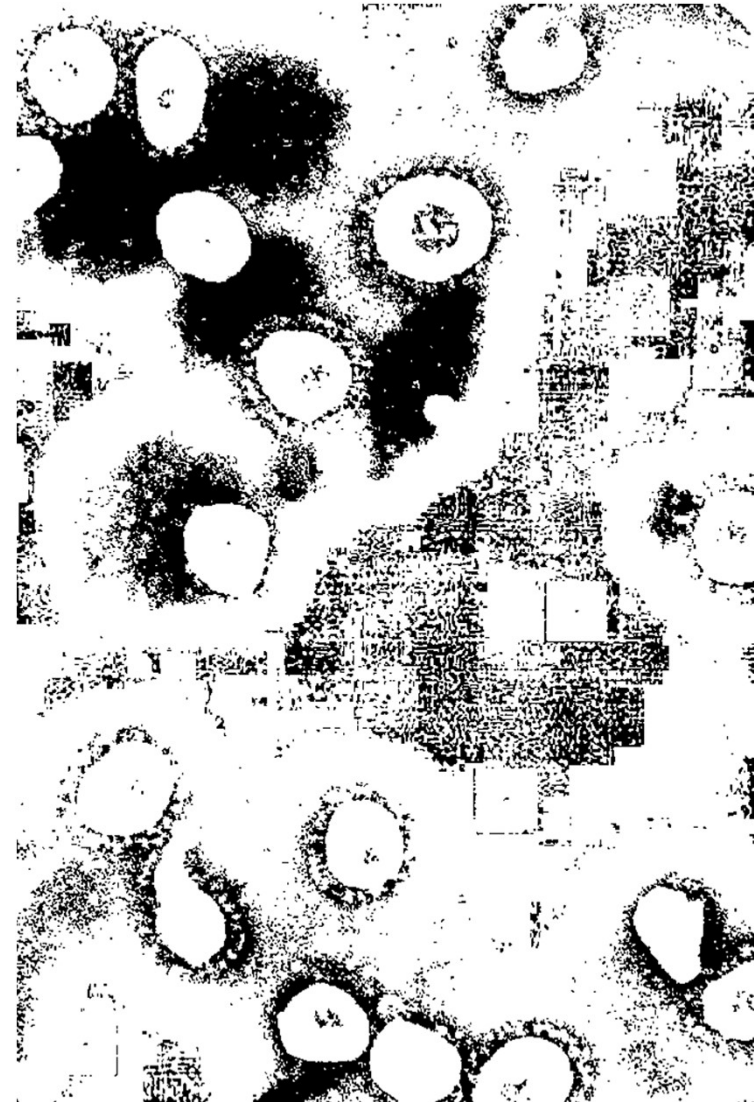
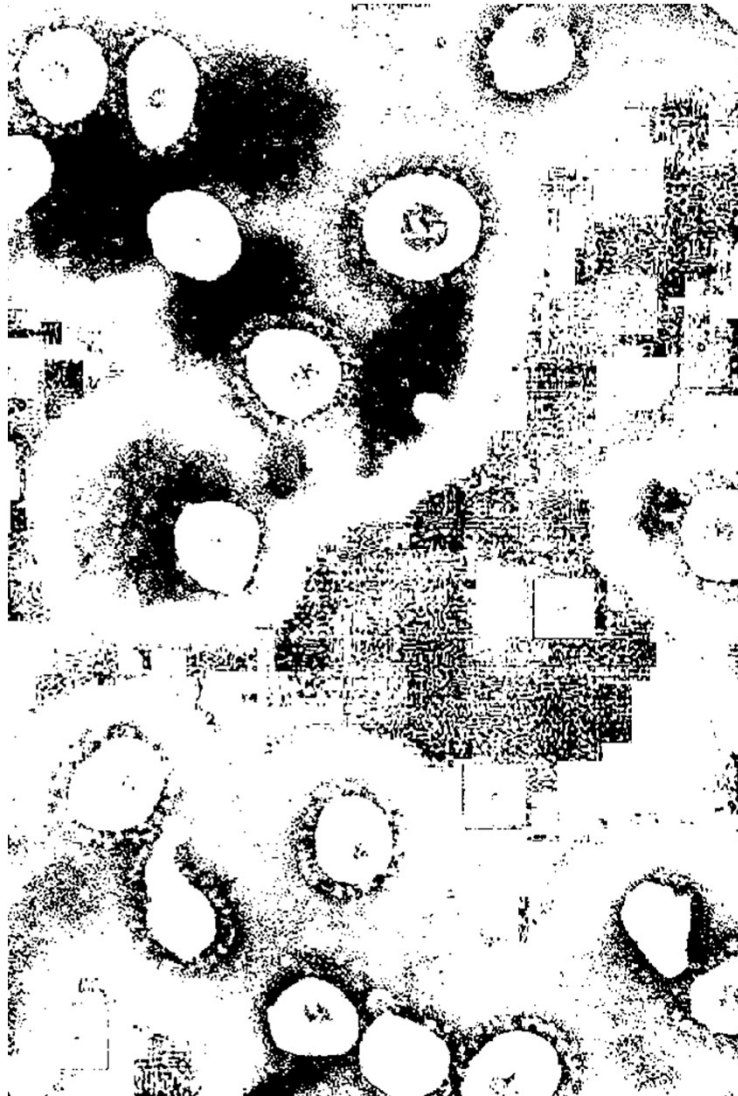
Примеры

28



Примеры

29



Фильтр преобладающего оттенка

30

- Окно фильтрации 3x3 разбивается на 4 квадрата 2x2 элемента.
- Для каждого из этих квадратов вычисляется среднее значение и дисперсия.
 - ▣ $M[x] = \frac{1}{4} \sum x$
 - ▣ $D[x] = \sum (x - M[x])^2$
- Выбирается квадрат, для которого дисперсия минимальна, и среднее значение элементов этого квадрата заносится в центр окна фильтрации

Примеры

Фильтр уменьшения контрастности

32

- Контрастность Майкельсона:
$$\gamma = \frac{B_{\max} - B_{\min}}{B_{\max} + B_{\min}}$$
- B_{\max} , B_{\min} — максимальное и минимальное значение яркости изображения.
- Уменьшение контрастности должно приводить к сближению B_{\max} и B_{\min} .
- Увеличение контрастности должно приводить к расхождению B_{\max} и B_{\min} .

Примеры

33



Фильтр «удаление контрастной точки»

34

- Мера локального контраста

$$C(i, j) = 1 - \frac{1}{1 + k \cdot \sigma^2(L)}$$

- Если центральный пиксель является наиболее контрастным в окне, то заменить его яркость на яркость предыдущего наиболее контрастного пикселя в окне.
- ?

Примеры

35



Фильтр Гаусса (gaussian blur)

36

- В качестве импульсной переходной функции использует функцию Гаусса:
- Фильтр является сепарабельным – раскладывается на произведение двух одномерных фильтров:
- Фильтрует низкие частоты. При широких окнах приводит к размытию.
- Среднеквадратическое отклонение настраивается. Пример ядра фильтра 5*5 при СКО=1:

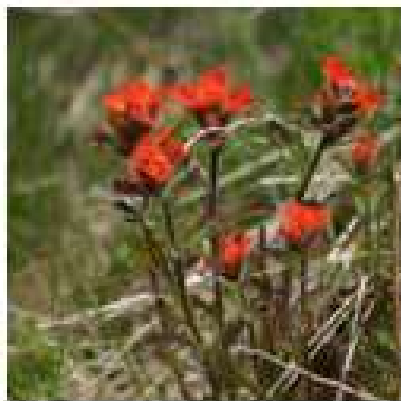
$$I(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

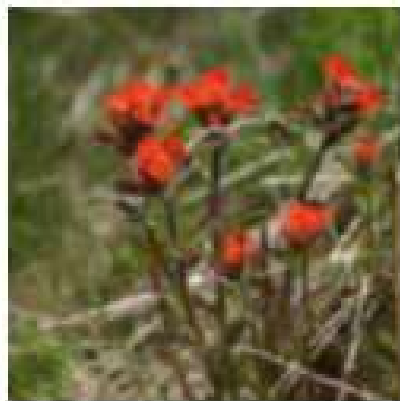
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

Фильтр Гаусса. Пример

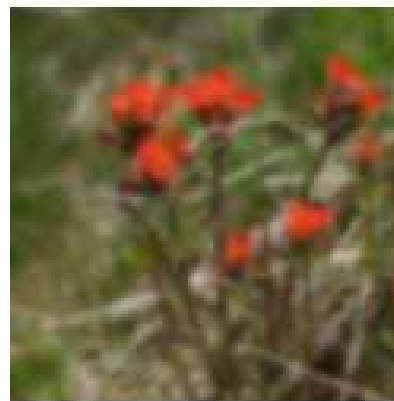
37



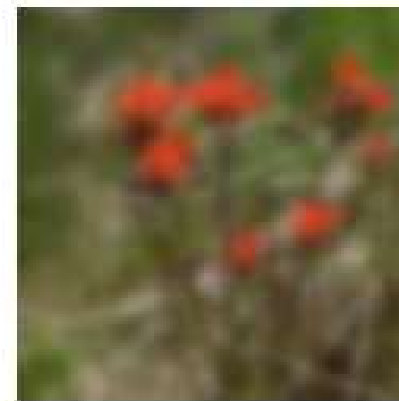
Original Image



Blurred Image ($\sigma = 10$)



Blurred Image ($\sigma = 20$)



Blurred Image ($\sigma = 40$)

- Результат обработки изображения при различных значениях стандартного отклонения в распределении Гаусса
- <https://software.intel.com/en-us/articles/iir-gaussian-blur-filter-implementation-using-intel-advanced-vector-extensions/>

Фильтр Габора

38

- Ядро фильтра (функция Габора) определяется в виде гармонической функции, помноженной на гауссиан:

$$G(x, y) = \cos(2\pi\theta x_\phi) e^{-\frac{1}{2}\left[\frac{x_\phi^2}{\sigma_x^2} + \frac{y_\phi^2}{\sigma_y^2}\right]}$$

$$x_\phi = x \cos(\phi) + y \sin(\phi)$$

$$y_\phi = -x \sin(\phi) + y \cos(\phi)$$

- Где
 - σ_x, σ_y – стандартные отклонения гауссова ядра по осям x, y , определяющие растянутость фильтра по осям
 - θ – частотная модуляция фильтра;
 - ϕ – пространственная направленность фильтра, определяющая его ориентацию относительно главных осей.
- Фильтр Габора применяется для распознавания границ объектов при обработке изображений.

Фильтр Габора. Пример

39

- Обработка изображения фильтром Габора достигается путём усреднения значений обрабатываемого изображения по некоторой области в каждой точке.
- Наложение фильтра Габора на изображение имеет вид:

$$I'(x, y) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n I\left(x - \frac{n}{2} + i, y - \frac{n}{2} + j\right) \cdot G(i, j)$$

- Где
 - $I(x, y)$ – интенсивность исходного изображения
 - $I'(x, y)$ – интенсивность нового изображения
 - $G(i, j)$ – значение функции Габора



Морфологические операции на бинарных изображениях

Понятие структурирующего элемента

Морфологическое сжатие и расширение

Открытие и закрытие

Базовые морфологические операции

41

- К исходному изображению X применяется структурирующий элемент V .
- Элемент V выбирается в виде прямоугольника, диска или кольца, что задаётся матрицами:

$$B_{rect} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad B_{disc} = \begin{pmatrix} & 1 & 1 & \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ & 1 & 1 & \end{pmatrix} \quad B_{ring} = \begin{pmatrix} & & 1 & 1 & \\ & 1 & & & 1 \\ 1 & & & & \\ & 1 & & & 1 \\ & & & 1 & 1 \end{pmatrix}$$

- Базовыми морфологическими операциями являются
 - ▣ Сжатие (Erosion, эрозия)
 - ▣ Расширение (Dilation, дилация)

Алгоритмы расширения и сжатия

42

□ Для бинарного изображения:

■ Расширение (*Dilation*):

- Если в окне операции находится хотя бы одна единица,
- То в центральную позицию окна заносится 1.

■ Сжатие (*Erosion*):

- Если в окне операции находится хотя бы один ноль,
- То в центральную позицию окна заносится 0.

Алгоритмы расширения и сжатия

43

□ *Для полутонового изображения:*

▣ Расширение (Dilation):

■ в окне операции отыскивается элемент с максимальным значением и помещается в центральную позицию.

■
$$I(x_c, y_c) = \max_{(x,y) \in \Omega} I(x, y)$$

▣ Сжатие (Erosion):

■ в окне операции отыскивается элемент с минимальным значением и помещается в центральную позицию

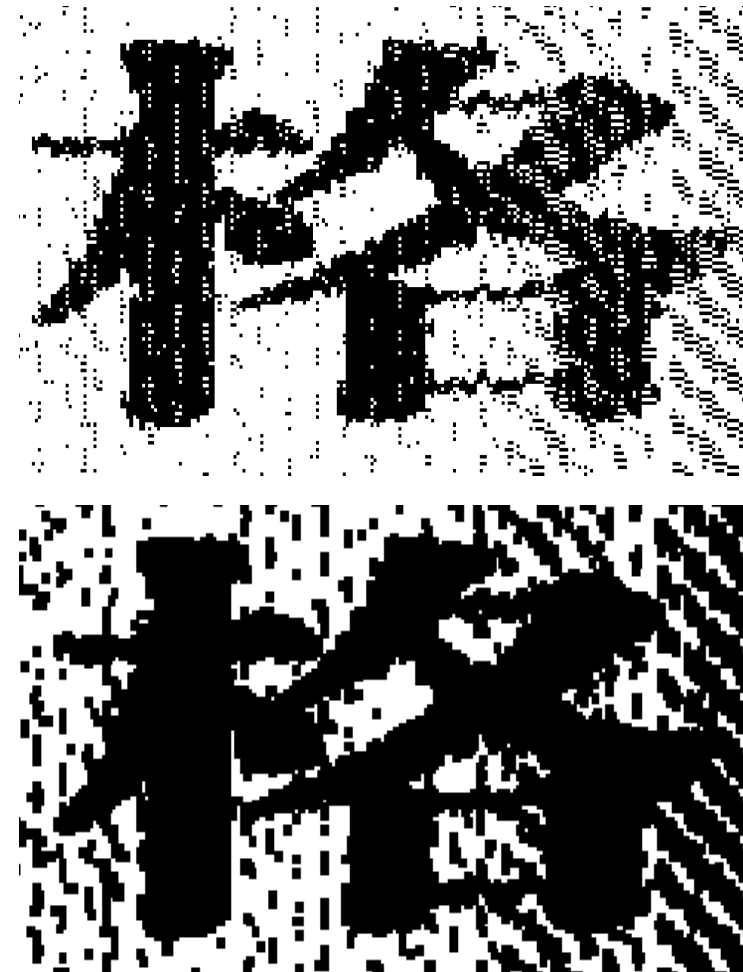
■
$$I(x_c, y_c) = \min_{(x,y) \in \Omega} I(x, y)$$

Расширение

44

- ❑ Расширение удаляет внутренние дыры и каналы, увеличивая при этом размер объекта и внешние дефекты.
- ❑ Расширение чёрного (сжатие белого):

$$B_{rect} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Сжатие

45

- Сжатие «съедает» внешние выступы формы, уменьшая размер объекта и увеличивая внутренние дефекты.



- Сжатие чёрного (расширение белого):

$$B_{rect} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Восстановление размера объекта

46

- Для того чтобы восстановить размер объекта, нужно применить двойственную операцию: расширить после сжатия или сжать после расширения.
- Операция **открытия** (сжатие + расширение) используется для удаления внешних дефектов.
- Операция **закрытия** (расширение + сжатие) используется для удаления внутренних дефектов.
- При этом расширение и сжатие выполняются с одинаковой апертурой и одинаковым числом итераций.

Алгоритмы открытия и закрытия

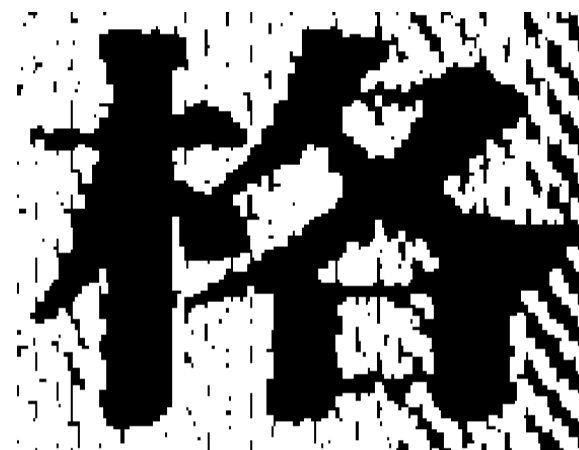
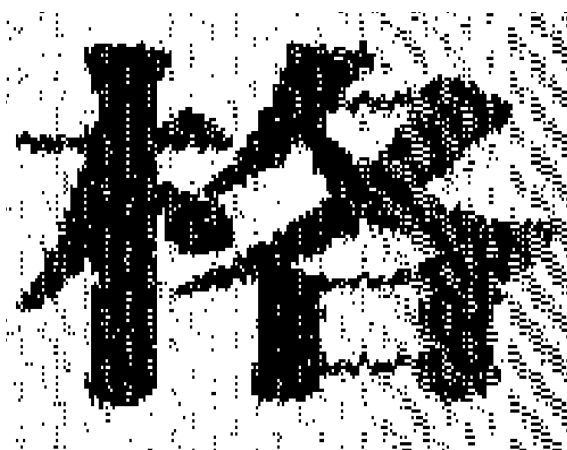
47

- Операция открытия *Opening*
 - На обрабатываемом изображении отыскиваются те участки, которые, будучи покрытые структурирующим элементом, содержат одни единицы. Эти “единичные” области сохраняются в результирующем изображении. Остальные пиксеты заполняются нулями.
- Операция закрытия *Closing*
 - На обрабатываемом изображении отыскиваются те участки, которые, будучи покрытые структурирующим элементом, содержат одни нули. Эти “нулевые” области сохраняются в результирующем изображении. Остальные пиксеты заполняются единицами.

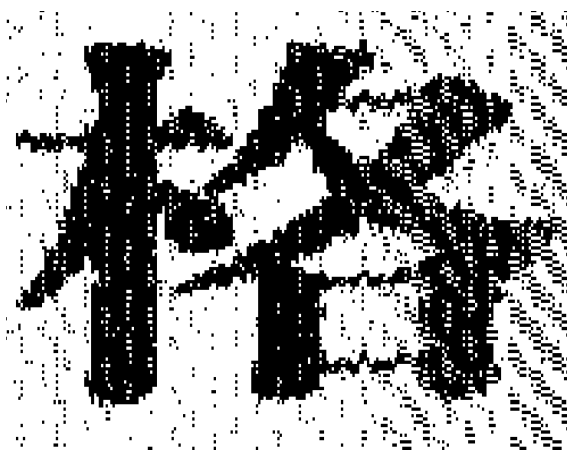
Открытие и закрытие

48

□ Открытие (opening) = Сжатие-расширение белого:



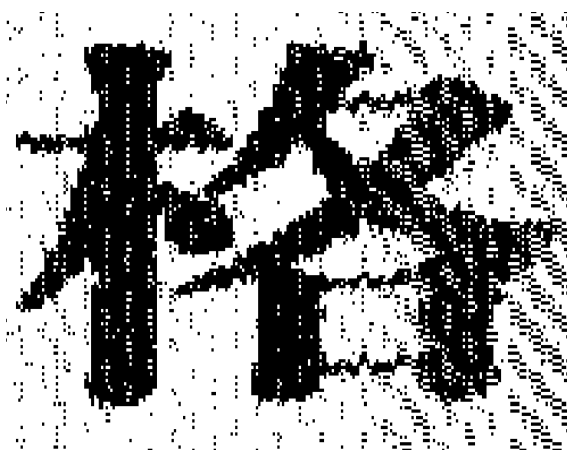
□ Закрытие (closing) = Расширение-сжатие белого:



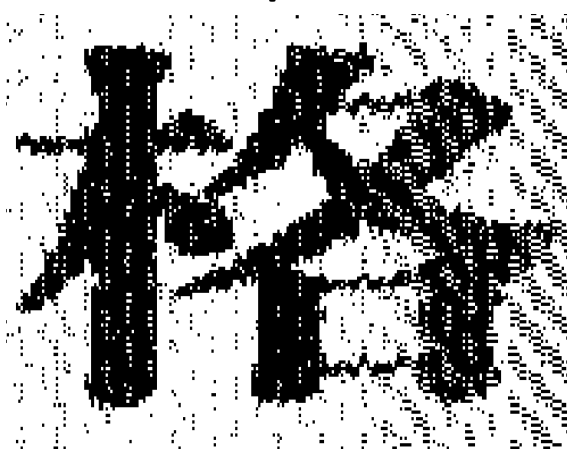
Пары открытия и закрытия

49

□ Открытие-закрытие белого:



□ Закрытие-открытие белого:



Что почитать

50

- Бондаренко А.В., Галактионов В.А., Горемычкин В.И., Ермаков А.В., Желтов С.Ю. Исследование подходов к построению систем автоматического считывания символьной информации. Препринт ИПМ им. М.В.Келдыша РАН. Москва, 2003
- Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. — М.: Вильямс, 2004. — 928 с.
- Шапиро Л., Стокман Дж. Компьютерное зрение. — М.: БИНОМ. Лаборатория знаний, 2006. — 752 с.
- http://wiki.technicalvision.ru/index.php/%D0%9D%D0%B5%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%84%D0%B8%D0%BB%D1%8C%D1%82%D1%80%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D0%BE%D0%BB%D1%83%D1%82%D0%BE%D0%BD%D0%BE%D0%B2%D1%8B%D1%85_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9