

ПРИКЛАДНІ ДОМЕНИ І ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

УДК 681.3.

Мучник М.М.,

Київська державна академія водного  
транспорту ім. гетьмана П. Конашевича-  
Сагайдачного

# ПРЕДПРОЦЕССОРНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ

*Для обеспечения более качественного автоматического распознавания печатных и/или рукопечатных символов необходима их предварительная обработка, т.е. приведение к виду, более удобному для распознавания.*

*Для забезпечення більш якісного автоматичного розпізнавання друкованих та \ або рукопечатних символів необхідна їх попередня обробка, тобто приведення до вигляду, більш зручному для розпізнавання.*

*To provide better automatic recognition of printed and \ or hand-printed characters is necessary to pre-treatment, ie leading to a form more convenient for recognition.*

**Ключевые слова:** скелетизация символов, сглаживающие пространственные фильтры, фильтры повышения резкости.

## Введение

В последнее десятилетие разработаны ряд методов и технологий автоматического распознавания графических образов символов печатного или же «рукопечатного» текста. Графические образы текстовых документов обычно формируются посредством их сканирования. Однако 100% правильное распознавание отсканированных текстов не обеспечивает ни одна из известных в настоящее время систем автоматического распознавания. Это вызвано, в первую очередь, следующими причинами:

- написание одного и того же символа даже одним и тем же человеком может иметь существенные отличия (возможны разрывы линий, толщина линий символа может существенно отличаться от эталонной, могут отличаться цвет символов, размеры и тип шрифта и др.);

- символы, написанные разными людьми, могут отличаться еще более существенно.

Это делает актуальным поиск методов улучшения качества распознаваемых текстов до того, как производится собственно процесс распознавания.

В системах автоматического распознавания символов перед собственно распознаванием проводится их скелетизация (уточнения линий до ширины в один пиксель). Это накладывает на изображение некоторые требования, главное из которых – изображения должно быть черно-белым без градаций серого.

В данной статье предложен метод перевода черно-белого изображения с 256 оттенками серого в бинарное изображение. Результатом

такого перевода без предварительной фильтрации изображения, в общем случае, будет зашумлённое изображение с некоторым искажением символов. По этой причине вначале будет уместно рассмотреть **пространственные методы улучшения изображения** (фильтрации изображения), направленные на то, чтобы сделать отсканированное изображение максимально «пригодным» для скелетизации.

## 1. Особенности отсканированных изображений

Необходимо подчеркнуть, что методы обработки различных отсканированных изображений могут отличаться в силу различных настроек сканера. Изображение может быть тёмным или светлым, чётким или размытым, высоко или низко контрастным, заполненным только текстом или загрязненным не имеющими информационной ценности пятнами или точками («шум»). Именно по этой причине с одной стороны, перед началом распознавания необходимо провести фильтрацию изображения (т.е. удалить с изображения все мешающие распознаванию объекты). С другой стороны, достаточно сложно однозначно назвать метод фильтрации, который был бы идеальным для применения к любым изображениям. Можно лишь говорить о том, насколько использование того или другого метода не «испортит всю картину», если принять его за базовый.

## 2. Сглаживающие пространственные фильтры

Сглаживающие фильтры [1,2] применяются для расфокусировки изображения и подавления шума. Расфокусировка может применяться

как предварительный шаг обработки изображения, например, для удаления мелких деталей перед обнаружением больших объектов, или же для устранения разрывов в линиях или деталях. Для подавления шумов может использоваться расфокусировка с применением как линейной, так и нелинейной фильтрации.

### 3. Линейные сглаживающие фильтры

Выход (отклик) простейшего линейного сглаживающего пространственного фильтра есть среднее значение элементов по окрестности, покрытой маской фильтра. Такие фильтры иногда называют усредняющими фильтрами или сглаживающими фильтрами.

Идея применения сглаживающих фильтров достаточно ясна. Уменьшение «резких» переходов уровней яркости достигается заменой исходных значений элементов изображения на средние значения по маске фильтра. Поскольку случайный шум как раз характеризуется резкими скачками яркости, наиболее очевидным применением сглаживания является подавление шума. Для решения общих задач негативной стороной применения сглаживающих фильтров является расфокусировка контуров различных объектов (которые, также, характеризуются резкими перепадами яркостей), однако для задачи скелетизации это только плюс, т.к. чем более «гладким» будет контур символа, тем более ровным получится его скелет.

Самым простым сглаживающим фильтром является фильтр, который даёт среднее значение яркости пикселей в окрестности  $3 \times 3$ , что достигается суммированием значений всех элементов в этой окрестности, и делением результата на 9. Такой пространственный фильтр иногда называют однородным усредняющим фильтром.

Следующий фильтр является усовершенствованием предыдущего и результат его применения - так называемое взвешенное среднее. Этот термин применяется, чтобы показать, что значения элементов умножаются на разные коэффициенты, что позволяет присвоить им как бы разные «важности» (веса) по сравнению с другими. Основная стратегия присвоения центральному пикселю наибольшего веса, а остальным – обратно пропорционально их расстоянию, имеет целью уменьшение расфокусировки при сглаживании.

1	2	1
2	4	2
1	2	1

В приведенной маске показаны коэффициенты, которые используются для вычисления в таком фильтре. Можно было бы выбрать и другие значения коэффициентов, но сумма показанных в этой маске равна 16, что удобно при компьютерной реализации, поскольку это степень двойки.

На практике результаты сглаживания первым или другим фильтром выглядят достаточно похожими, однако применение маски с коэффициентами даёт лучшие результаты при последующей скелетизации.

Также следует отметить, что маска фильтра может быть не только  $3 \times 3$ . Если применить фильтр по маске  $5 \times 5$ , то чёткость символов падает настолько (сливаются линии символа), что скелетизация символов становится проблематичной.

### 4. Фильтры, основанные на порядковых статистиках

Фильтры, основанные на порядковых статистиках, относятся к классу нелинейных пространственных фильтров. Отклик такого фильтра определяется предварительным упорядочиванием (ранжированием) значений пикселей, покрываемых маской фильтра, и последующим выбором значения, находящегося на определённой позиции упорядоченной последовательности (т.е. имеющего определённый ранг). Собственно фильтрация сводится к замещению исходного значения пикселя (в центре маски) на полученное значение отклика фильтра. Наиболее известен **медианный фильтр**, который, как следует из названия, заменяет значение пикселя на значение медианы распределения яркостей всех пикселей в окрестности (включая и исходный). Медианные фильтры весьма популярны потому, что для определённых типов случайных шумов они демонстрируют отличные возможности подавления шума при значительно меньшем эффекте расфокусировки, чем у линейных сглаживающих фильтрах с аналогичными размерами.

Медиана набора чисел есть такое число  $\xi$ , что половина чисел из набора меньше или равны  $\xi$ , а другая половина – больше или равны  $\xi$ . Чтобы выполнить медианную фильтрацию для элемента изображения, необходимо сначала упорядочить по возрастанию значения пикселей внутри окрестности, затем найти значение медианы, и, наконец, присвоить полученное значение обрабатываемому элементу. (Для окрестности  $3 \times 3$  элементов медианой будет

пятое значение по величине, для окрестности  $5 \times 5$  – тринадцатое, и так далее).

В контексте рассматриваемой задачи улучшения изображения перед скелетизацией медианный фильтр может быть применён как к 256-цветному изображению, так и (после перевода) к чёрно-белому. Во втором случае происходит зачастую полезное сглаживание краёв символов (хотя на изображениях с плохим качеством текста эта процедура приводит к сливанию воедино соседних символов).

Также следует отметить, что на практике используется не только медианный фильтр, а также фильтр максимума и фильтр минимума, откликами которых являются последний и первый элементы отсортированного массива соответственно. Но такие фильтры могут удалять и мелкие детали символов, что впоследствии приведет к ошибкам распознавания.

### 5. Пространственные фильтры повышения резкости

Рассмотренные выше фильтры ориентированы на понижение резкости символов. Несмотря на тот факт, что изначально кажется неразумным использование одновременно сглаживающих фильтров и фильтров повышения резкости, практика показывает, что такая комбинация заслуживает внимания. Главная цель использования фильтров повышения резкости заключается в том, чтобы подчеркнуть мелкие детали изображения (линии символов). Но использование этих фильтров самостоятельно приводит к искажениям букв и разрывам в линиях.

Рассматриваемый в данной статье фильтр основан на использовании вторых производных: лапласиан. Это обусловлено тем, что повышение резкости достигается пространственным дифференцированием. Величина отклика оператора производной в точке изображения пропорциональна степени разрывности изображения в данной точке. Таким образом, дифференцирование изображения позволяет усилить перепады и не подчёркивать области с медленными изменениями уровней яркости.

Практической реализацией лапласиана (оператора Лапласа) является вычитание из исходного изображения, изображения отфильтрованного по следующей маске:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

или же, соединив операцию вычитания с операцией фильтрации, с помощью единой маски:

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \end{matrix} \quad (1)$$

Также, иногда используется маска:

$$\begin{matrix} 0 & -1 & 0 \\ -1 & -1 & -1 \\ -1 & 9 & -1 \end{matrix} \quad (2)$$

Применение маски (2) является более мощным средством повышения резкости, но, именно по этому, эта маска неприемлема для сохранения символов в исходном виде. Также маска (2) сильно увеличивает шум, который становится видим после перевода изображения в два цвета.

### 6. Перевод изображения в градациях серого в бинарное

Предположим, что некоторое изображение, заданное функцией  $f(x,y)$ , где  $f(x,y)$  – значение цвета в точке  $(x,y)$ , содержит тёмные объекты на светлом фоне. Очевидный способ выделения объектов из окружающего фона состоит в вычислении порогового значения  $T$ , такого что любая точка  $(x,y)$  в которой  $f(x,y) < T$  будет считаться точкой объекта, а в противном случае – точкой фона. Таким образом, получение бинарного изображения достигается путём поэлементного сканирования исходного изображения и присваивания каждому пикселю нового цвета (белого или чёрного).

Но зачастую случается, что у изображения, полученного в результате сканирования, одна часть затемнена, другая же – более светлая.

Для решения этой проблемы может быть предложен следующий подход:

- все изображение разбивается на квадраты;
- для каждого квадрата отдельно вычисляется пороговое значение.

Практические исследования показывают, что универсальным можно считать разбиение изображения на  $32 (2 \uparrow 5)$  квадрата по вертикали и горизонтали. Хотя, естественно, это число ничего не означает, и можно экспериментировать и с другими.

Пороговое значение вычисляется суммированием умножений каждой градации серого цвета (0..255) на количество ее «представителей» (пикселей) в квадрате и делением результата на общее количество пикселей (напомним, что цвет состоит из трёх составляющих (RGB), но у изображений в градациях серого все три составляющие равны, поэтому, для

определения цвета пикселя можно брать любую):

$$B = \frac{\sum_i c_i \cdot i}{\sum_i c_i}, \text{ где}$$

$B$  – искомое пороговое значение,  $c_i$  – количество пикселей  $i$ -того цвета.

В исследованиях, целью которых ставилось уменьшение шума на изображении, была найдена такая полезная операция: после вычисления порогового значения сильно его понизить (например, на 50) и посмотреть, останется ли после присвоения каждому пикселю чёрного или белого цвета хоть один чёрный пиксель в квадрате. Если останется, это будет означать, что в этом квадрате с вероятностью 95% находится текст (т.к. пиксели текста – самые тёмные на изображении), если не останется, то можно говорить о том, что в данном квадрате текста нет и его необходимо «забелять».

Метод перевода изображения в градациях серого в бинарное изображение, а также методы фильтрации изображения были реализованы для подготовки изображения к скелетизации, используемой в системе автоматического распознавания текстов “Cunning Eye” [3].

#### Выводы

1. Используемые в настоящее время методы фильтрации графических изображений текстовых символов не обеспечивает высококачественной скелетизации символов.

2. Целесообразность использования тех или иных методов фильтрации существенно зависит от качества изображения, поэтому поиск наиболее оптимального из них сводится к комбинированию этих методов и экспериментированию с различными коэффициентами.
3. При изменении каких-либо коэффициентов в методах, стартующих в программной реализации раньше остальных, для получения качественного результата необходимо корректировать и коэффициенты последующих методов.
4. Для обеспечения удобства и повышения качества скелетизации в статье предложен метод перевода черно-белого изображения с 256 оттенками серого в бинарное изображение.
5. Описанные в статье методы были реализованы в системе автоматического распознавания текстов “Cunning Eye” для подготовки изображения к скелетизации.

#### Список литературы:

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – Техносфера, Москва, 2005 – 1072 с..
2. Форсайт Д., Понс Д. Компьютерное зрение. Современный подход. – Вильямс, Москва, 2004- 928 с.
3. Мучник М.М., Прохоров В.Г. Об одном методе представления графических образов текстовых символов в виде характеристических векторов. – Киев, Водний транспорт, №3\15, 2012, с.188 – 193.

#### Сведения об авторе:



**Мучник Михаил Манусович**, кандидат технических наук, доцент кафедры информационных технологий, факультет экономики транспорта Киевской государственной академии водного транспорта им. гетьмана П.Конашевича-Сагайдачного. Научные интересы: программная инженерия, жизненный цикл обработки документов, жизненный цикл разработки программных продуктов, системы автоматизированного документооборота, системы автоматизированного ввода данных с бумажных документов.

**e-mail:** [muchnik@voliacable.com](mailto:muchnik@voliacable.com)