

ПРИКЛАДНІ ДОМЕНИ І ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

УДК 681.3

Мучник М.М.

**Киевская государственная академия
водного транспорта им. гетьмана П.
Конашевича-Сагайдачного**

СКЕЛЕТЕЗАЦИЯ ГРАФИЧЕСКИХ ОБРАЗОВ ПЕЧАТНЫХ И РУКОПИСНЫХ СИМВОЛОВ

В статье рассмотрены подходы к предварительной обработке печатных и рукопечатных текстов, подготавливаемых к автоматическому распознаванию, с помощью метода скелетизации. Предложен новый – спиральный - метод скелетизации, обеспечивающий ускорение и улучшение качества распознавания текстов. Метод спиральной скелетизации был реализован в системе автоматического распознавания "Cunning Eye".

У статті розглянуті підходи до попередньої обробки друкованих та рукодрукованих текстів, що готуються до автоматичного розпізнавання, за допомогою методу скелетизації. Запропонований новий (спіральний) метод скелетизації, який забезпечує прискорення й поліпшення якості розпізнавання текстів. Метод спіральної скелетизації був реалізований у системі автоматичного розпізнавання "Cunning Eye".

The article describes the approach to pre-processing of printed and hand-printed text, prepared for automatic detection, the method skeletonization. A new (spiral) skeletonization method that provides faster and improve the quality of OCR. With spiral skeletonization was realized in the system of automatic recognition of "Cunning Eye".

Ключевые слова: скелетизация символов, метод эрозии, спиральный метод скелетизации

Введение

В настоящее время для обработки бумажных документов, заполненных печатными и/или рукопечатными символами, широко используются системы автоматического распознавания текстов. Однако ни одна из них не дает 100% гарантии качества при распознавании, поскольку пока не созданы алгоритмы распознавания, инвариантные к почерку, яркости и контрастности изображения, толщине линий в изображениях символов и др., что приводит к многочисленным ошибкам распознавания.

В настоящей работе рассмотрен метод скелетизации графических образов рукопечатных символов, т.е. метод утоньшения линий образа до ширины в один пиксель, который позволяет существенно улучшить качество дальнейшего автоматического распознавания символов.

1. Особенности метода скелетизации

Скелетизацию часто применяют для уменьшения объема данных для дальнейшей обработки. В частности, она широко применяется при распознавании образов, т.к. именно графическая информация имеет наибольшую избыточность. Основная идея этого метода - получение упрощенных и более

экономичных по количеству точек изображения моделей всех обрабатываемых объектов. Одной из основных сложностей таких операций является оценка их эффективности. Оценить ее довольно сложно, т.к. скелетизация - это только промежуточный этап, хотя от него зависит весь последующий процесс. Ошибочные предположения о виде символа в большинстве случаев можно заметить только на следующем этапе, что приведет к значительным потерям времени и сил. И, в тоже время, удачный метод обработки входных данных значительно облегчит дальнейшую классификацию и устойчивость к искажениям. Данный метод можно рассматривать как своеобразное сжатие данных с потерями. К операции скелетизации предъявляются, как правило три основных требования [1]:

- Связность объектов изображения и фона должна быть сохранена (не должно быть разрывов);
- Концы средних линий должны располагаться как можно ближе к их истинному положению;
- центральные линии объектов должны быть выделены достаточно точно.

Самая большая группа алгоритмов скелетизации основана на идее итеративного удаления внешних слоев или контурных точек объектов до получения скелета. Итеративные алгоритмы используют маску (как правило, размером 3x3), которая перемещается по всему изображению, чтобы определить новое значение центрального пикселя.

Алгоритмы данной группы можно разделить на два класса: параллельные и последовательные. В параллельных алгоритмах окно располагается одновременно во всех пикселях, и при его обработке не используют новые полученные на данной итерации значения пикселей. При последовательных алгоритмах пиксели обрабатываются последовательно.

```

        0 0 0 0 ? 0
      0 1 1 1   ?
    0 1 1 1     ?
    0 1 1       1 1 0
    0 1 1 1 1 1 1 1 0
    0 0 0 0 0 0 0 0 0
    
```

Рис.1. Исходное изображение

где 1 – это черная точка; 0- это контур; ? – точка, которую нельзя удалить.

Основной особенностью этого алгоритма является то, что удаление точек происходит не сразу. Сначала точки помечаются как претенденты на удаление. Это делается для того, чтобы можно было определить, когда дальнейшие итерации уже не нужны, т.е. достигнут скелет. Если мы проверили весь контур и в этом контуре нет точек на удаление, то скелет достигнут, иначе надо повторить прохождение по контуру шаблонов.

Это самый простой метод, позволяющий определить, когда нужно прекратить скелетизацию. К его недостаткам можно отнести то, что один и тот же пиксель (точку) мы должны просматривать дважды.

Получение скелетного представления символа - не самоцель, это изображение используется затем для выделения характеристик символа. Главная задача скелетного представления символа - предоставить возможность для получения ряда характеристик исходного изображения.

2. Метод эрозии

Большинство алгоритмов скелетизации используют так называемый метод эрозии, основная идея которого состоит в том, чтобы найти контур символа и применить к нему шаблоны, а затем удалить часть контура, в соответствии с полученными результатами. Для наглядности приведем пример

```

          ? ?
        1 1 1   ?
    1 1 1     ?
    1 1       1 1
    1 1 1 1 1 1 1
    
```

Рис.2. Изображение после первой итерации скелетизации

Анализ скорости скелетизации с помощью разработанной по данному алгоритму программы на языке программирования С# показал её довольно низкую скорость. Для ускорения работы алгоритма было решено использовать временный буфер, в который записываются координаты всех подлежащих удалению пикселей изображения. Благодаря этому отпадает необходимость повторного поиска пикселей контура и их сравнения. Очевидно, что этот метод требует дополнительного объема памяти, придется также использовать динамический массив, а время для его создания и изменения размера достаточно велико даже при использовании такого языка, как С++, обеспечивающего генерацию достаточно оптимальных программ.

3. Спиральный метод скелетизации

Для улучшения характеристик процесса скелетизации был разработан и реализован метод спиральной скелетизации, не содержащий отрицательных моментов двух предыдущих методов. Суть его состоит в том, что точки, которые надо удалить, удаляются сразу (в данном случае контур буквы

удаляется не кольцами, а по спирали). Метод нахождения следующей точки аналогичен поиску контура.

Рассмотрим более подробно расположение точки. В данном случае предполагается, что точка имеет 8 соседей (т.н. 8-связность), см. рис.3, хотя может быть и 4 (4-связность), см. рис.4.

0	1	2
7	8	3
6	5	4

Рис. 3

	0	
3	4	1
	2	

Рис. 4

Для удобства пронумеруем их. Предположим, что предыдущая точка имела номер «0», а анализируемая в данный момент - номер «8». Тогда поиск следующей точки будет осуществляться таким образом: анализируем точку «1» на предмет того, является ли она черной. Если да, то переходим к ней. В противном случае анализируем точку «2» и т.д. до нуля.

В качестве точки останова алгоритма используется следующее условие: если мы одну за другой встречаем те две точки, которые уже были ранее просмотрены, и после первого просмотра не было удалено не одной из этих точек, то скелет найден. Вместо динамического буфера мы используем только две переменные и повторно просматриваем контур только один раз, в конце, когда он является скелетом. Использовать надо обязательно две точки и важно их следование. Продемонстрируем это следующим примером: Будем обходить скелет по часовой стрелке, начиная с верхнего левого угла.

Как мы видим на рис.5, два выделенных пикселя мы проходим два раза, сначала сверху вниз, а затем снизу вверх. Именно поэтому важно учитывать порядок следования точек символа.

Чтобы определить, нужно удалять точку или нет, будем анализировать соседние точки в соответствии с шаблонами.

В зависимости от шаблонов, мы будем получать различные скелеты. Следует упомянуть два подхода к созданию шаблонов. При первом подходе мы описываем невозможность удаления точки, а при втором описываем, когда она должна быть удалена. Ниже представлены два вида шаблонов, второй из которых приведен в [2]. Хотя при использовании первого шаблона необходимо проделать 10 сравнений, а при использовании второго только 6, первый метод можно оптимизировать, и четыре последних правила заменить проверкой окружения, а именно, если вокруг центральной точки только одна черная, то центральную точку удалять нельзя.

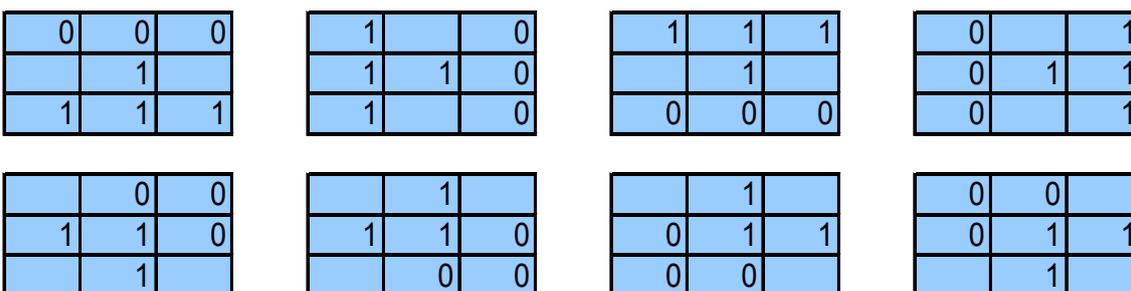
	1		1
	1		1
	1		1
	1	1	
	1		1
	1		1

Рис. 5



Если шаблон совпал то центральная точка (8) не может быть удалена

Рис. 6. Шаблон невозможности удаления центральной точки



Если шаблон совпал то центральная точка стирается

Рис. 7. Шаблон удаления центральной точки

Если реализовать алгоритм скелетизации на языке Ассемблера, используя процессорные команды MMX и SSE, которые поддерживаются современными процессорами (начиная с Pentium III, Athlon), то второй вариант будет предпочтительнее, поскольку скорость выполнения кода программы существенно повысится.

4. Анализ результатов

Как уже упоминалось выше, скелетизация нужна для выделения ряда характеристик символа и, возможно, не стоит добиваться 100% соответствия скелета реальным средним

линиям, а стоит учитывать те артефакты, которые дала скелетизация.

Например, при скелетизации буквы «е» (см. Рис. 8) проявились такие артефакты, как ненужная линия. Заметны значительные отклонения при соединении линий разной толщины в буквах. К сожалению, такие линии используются во многих печатных шрифтах, например в Times New Roman (см. Рис.9, 10).

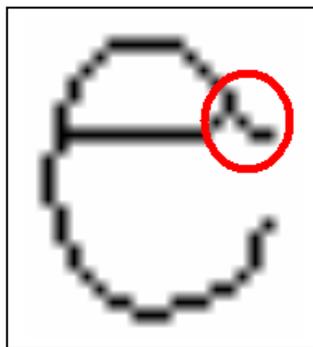


Рис.8

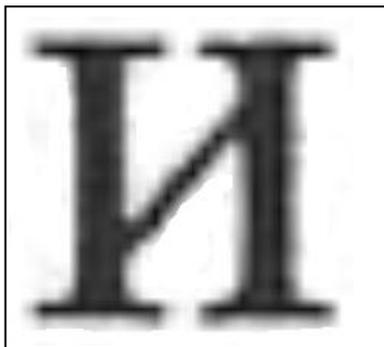


Рис. 9

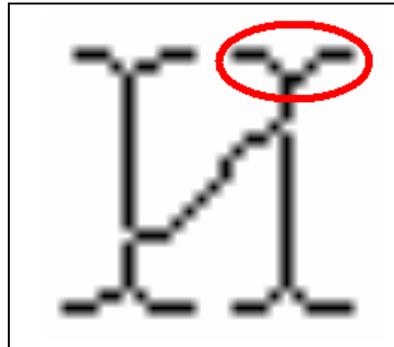


Рис. 10

Следует также отметить, что данный метод очень чувствителен к качеству исходного изображения. Даже неровный край

ведет к значительному искажению скелета, что требует специфической фильтрации изображений.(см. Рис.11, 12, 13).

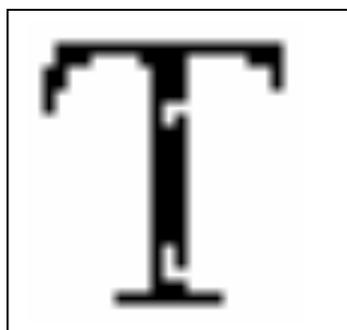


Рис. 11



Рис. 12

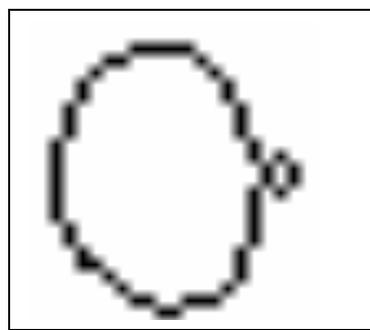


Рис. 13

Спиральный метод скелетизации был применен для создания скелетов в ICR-системе “Cunning Eye” [3] и доказал свою работоспособность.

Выводы

1. В статье проведен анализ различных методов скелетизации графических образов печатных и рукопечатных символов, отмечены положительные и отрицательные особенности этих методов;

2. Описан разработанный авторами спиральный метод скелетизации, обеспечивающий, по сравнению с известными, более быструю и качественную скелетизацию;

3. Реализация спирального метода скелетизации в рамках ICR – системы “Cunning Eye” позволила обеспечить более качественное распознавание текстов.

Литература

1. Абламейко С. В., Лагуновский Д. М. Обработка изображений: технология, методы, применения. Учебное пособие. – Мн.: Амалфея, 2000. – 304 с.

2. Р. Гонсалес, Р. Вудс “Цифровая обработка изображений”, Техносфера, Москва, 2005- 1072 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