Литература

1. *Кручинин В. В.* Генераторы в компьютерных учебных программах. – Томск: Изд-во Том. ун-та, 2003. – 200 с.
2. *Таранчук В. Б.* Введение в язык Wolfram: учебные материалы для студентов факультета прикладной математики и информатики специальности 1-31 03 04 «Информатика». – Минск: БГУ, 2015. – 51 с.
3. *Таранчук В. Б.* Основы программирования на языке Wolfram: учебные материалы для студентов факультета прикладной математики и информатики специальности 1-31 03 04 «Информатика». – Минск: БГУ, 2015. – 49 с.

РАСПОЗНАВАНИЕ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ НА ИЗОБРАЖЕНИИ

В. М. Гошко

ВВЕДЕНИЕ

С теоретической и практической точек зрения, распознавание образов – сложная задача. Для распознавания образов человек использует весь накопленный опыт и знания. Кроме того, человек не перестает обучаться и, обладая абстрактным мышлением, может очень точно выделять характерные признаки образов, что при высокой "вычислительной мощности" головного мозга значительно повышает шансы на успешное распознавание.

Часто для улучшения качества распознавания текста каждый символ слова рассматривают в его контексте и используют словари для уточнения результата, что не применимо при распознавании математических формул. Кроме того, структура математической формулы сложнее и более непредсказуема, чем структура текста [1]. По этой причине важность высокой точности распознавания отдельных символов возрастает.

Описание этапов подготовки изображения к распознаванию опущено. Дано описание реализованных алгоритмов. Предложен алгоритм распознавания символов с использованием персептрона и векторного скелета символа, а также комбинированный подход на основе персептрона с последующим уточнением результата, используя метод сравнения с шаблоном. Были проведены замеры производительности некоторых классических алгоритмов и предложенных модификаций на одинаковых входных изображениях.

АЛГОРИТМЫ РАСПОЗНАВАНИЯ СИМВОЛОВ

Алгоритм наилучшего совпадения с шаблоном

Пусть для каждого символа, который мы хотим уметь распознавать, имеется изображение-шаблон, тогда каждому найденному символу на входном изображении можно сопоставить число Δ_k, которое определяет схожесть найденного символа с k-м шаблоном. Индекс $j = \max_k(\Delta_k)$ является индексом строкового представления найденного символа.

В одной из реализаций предлагается вычисление Δ_k по формулам (1) и (2), где $I_c(x,y)$, $I_p(x,y)$ – функции яркости пикселя изображения символа шаблона, а $D(x,y)$ – расстояние от пикселя с координатами (x,y) до ближайшего черного пикселя.

$$\Delta_k = \sum_{i=0}^{w}\sum_{j=0}^{h} W(i,j) \tag{1}$$

$$W(i,j) = \begin{cases} 1, I_c(x,y) = I_p(x,y) \\ -D(x,y), \sim \end{cases} \tag{2}$$

Для подготовки шаблона можно использовать несколько шрифтов и считать веса $W(x,y)$ как среднее значение веса для всех шрифтов.

Алгоритм с использованием персептрона и характеристик скелета символа

На вход нейронной сети подается вектор характеристик скелета изображения символа. Построив непрерывный однопиксельный скелет символа, мы можем достаточно просто численно оценить его структуру. Более подробно подход описан в [2]. В данной работе используется алгоритм Гуо для построения однопиксельного растрового скелета символа, а также был разработан алгоритм векторизации растрового скелета. Сравнение растровых и векторных скелетов представлено на Рисунок 1.

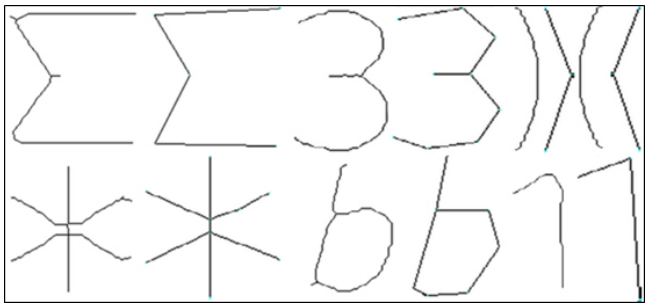

Рисунок 1. Сравнение растровых скелетов (слева) и векторных скелетов (справа)

Растровый скелет сохраняет все особенности изображения символа, что не всегда полезно. Кроме того, для большого размера изображения скелет будет содержать достаточно большое количество точек.

Убрать излишнюю точность скелета, а также уменьшить количество точек для обработки позволяет векторизация растрового скелета. Это достигается путем замены последовательности соседних точек скелета на отрезок. Также необходимо сохранить все ключевые точки скелета (точки самопересечения кривой, узлы), отсюда вытекают следующие правила:

1. Последовательности точек скелета $P_1,...,P_k$ можно заменить отрезком с концами P_1, P_k, если максимальное расстояние от отрезка (P_1, P_k) до точек последовательности $P_1,...,P_k$ не превосходит заранее заданного числа D.

2. Пусть имеется последовательность точек скелета $P_1,...,P_k$. Эту последовательность можно заменить отрезком с концами P_1, P_{k+1}, если P_{k+1} удовлетворяет одному из трех условий: P_{k+1} имеет одну точку соседа; P_{k+1} имеет три точки соседа; P_{k+1} имеет четыре точки соседа;

После этого можно перевести векторный скелет из неупорядоченного списка отрезков в граф, который строится путем замены каждого отрезка на ребро графа, а концов этого отрезка – на смежные вершины графа.

В качестве дополнительной обработки построенного графа можно применить процедуру *стягивания вершин*. Она подразумевает замену двух вершин одной новой вершиной в точке одной из рассматриваемых вершин, удаление ребра, инцидентного этим вершинам, и сохранения всех остальных связей. Операция стягивания применяется для каждой пары смежных вершин, если расстояние между ними меньше заранее заданного D_1.

В качестве характеристик символа берутся следующие характеристики: центр тяжести по осям X, Y; среднеквадратичное отклонение от центра тяжести по осям X, Y; количество скелетных точек, связанных с соседними по горизонтали, вертикали, главной диагонали, обратной диагонали, в отношении к общему количеству скелетных точек; количество скелетных точек, имеющий одного, трех, четырех соседей.

Векторизация скелета, помимо упрощения вида скелета, позволяет:

1. Убрать незначительные особенности символа.
2. Уменьшить количество точек скелета: вместо всех точек скелета рассматриваются только ключевые точки и связи между ними.
3. Добавить новые характеристики: например, количество циклов в графе, отношение числа правых поворотов к числу левых поворотов.

Алгоритм с использованием комбинации персептрона и алгоритма наилучшего совпадения с шаблоном

На выходе персептрона мы имеем вектор, элементы которого принимают значение в промежутке [-1;1]. В идеальном варианте, на одном выходе сети должна быть 1, а на всех остальных – -1. Тогда считается, что сеть дала однозначный ответ. На практике чаще всего на выходе сети мы имеем несколько значений больших 0, среди которых находим максимальное. Индекс максимального значения и будет считаться индексом символа в алфавите. Однако можно попробовать воспользоваться этим недостатком.

Итак, пусть есть множество индексов $R = \{i : Output(i) > 0\}$. Тогда можно провести еще одно распознавание на основе сравнения с шаблоном, но только для шаблонов с индексами из множества R.

Оценка результатов работы реализованных алгоритмов

Анализируя результаты, полученные на проверке реализованных алгоритмов и записанных в Таблице 1, можно сделать следующие выводы:

1. Комбинирование алгоритмов дает заметный прирост точности распознавания.
2. Алгоритм на основе векторного скелета в текущей реализации имеет точность распознавания ниже, чем алгоритм на основе растрового скелета. Среди выходов сети есть выход с индексом соответствующим верному символу алфавита, но выход с этим индексом примерно в 30% случаев не является максимальным.
3. Сравнение с шаблоном является самым медленным алгоритмом в чистом виде (без комбинаций с другими), однако его использование в комбинации с другими алгоритмами уменьшает время работы части сравнения с шаблоном примерно втрое.
4. Алгоритм, основанный на векторном скелете, чаще не имеет выхода с правильным символом (примерно на 17% меньше алгоритма на основе растрового скелета).

Таблица 1

Сравнительная таблица реализованных алгоритмов

	Растр. скелет + Шаблон	Вект. скелет + Шаблон	Комб. скелет + Шаблон	Шаблон	Растр. скелет	Вект. скелет	Комб. скелет
Точность распознавания	87.2%	70.5%	75.7%	12.8%	75.6%	73.1%	66.7%
Полное время распознавания (сек)	202.31	177	172.785	152.1	136.47	135.81	142.79

Литература

1. *Nazemi, A.* Mathematical Formula Recognition and Transformation to a Linear Format Suitable for Vocalization / A. Nazemi, I. Murray // International Journal on Computer Science and Engineering. – 2013. – P. 847–855.
2. *Варламов, А.Д.* Распознавание символов по скелетному изображению на основе нейронной сети. / А.Д. Варламов, М.Ю. Арефьева, Ю.А. Севитова, Ю.С. Степанова. [Электрон. ресурс]: www.обработка-изображений.рф/публикации/статья-30.pdf.

АЛГОРИТМЫ ВЫДЕЛЕНИЯ И АНАЛИЗА КРОВЕНОСНЫХ СОСУДОВ НА ТРЕХМЕРНЫХ ТОМОГРАФИЧЕСКИХ ИЗОБРАЖЕНИЯХ

А. И. Жилка

1. ВВЕДЕНИЕ

Для диагностики различных заболеваний кровеносной системы активно используются изображения магнитно-резонансной ангиографии (МРА), получаемые различными методами в зависимости от диагностируемой области организма.

В данной статье описываются алгоритмы выделения кровеносного дерева мозга и приводится некоторое их сравнение. Также представлены заключения по исследованию возможности вычисления скорости крови по времяпролетным (Time-of-Flight, ToF) МРА-изображениям.

2. ПРОЦЕДУРА АНАЛИЗА ИЗОБРАЖЕНИЙ

Для выделения и анализа кровеносных сосудов используется процедура, разделенная на несколько фаз. Каждая фаза реализует некоторый алгоритм, результаты которого передаются на вход следующей. Перечислим и дадим пояснения этим фазам.

1. Автоматический поиск маркеров: под маркерами сосудов понимается множество вокселов (трехмерный аналог пикселов) изображения, о которых мы точно знаем или, в случае автоматического их определения, в какой-то степени уверены в том, что они лежат в областях изображения, соответствующих сосудам.

2. Сегментация: на данном этапе выделяются все вокселы, относящиеся к областям сосудов. Все остальные вокселы перекрашиваются в цвет фона. Важно учитывать, что выделенное кровеносное дерево должно быть связным. Также, ввиду недостаточного разрешения МРА-изображений границы сосудов не всегда достаточно контрастны. Упо-