Динамическое построение функций сравнения с идеальным образом в задаче адаптивного распознавания текстовых символов

О.А. Славин, Ю.В. Титов

Аннотация. В работе рассмотрено адаптивное распознавание документа, представляющее собой двухпроходный процесс с обучением на результатах предыдущего прохода. Поставлены две задачи распознавания с помощью идеальных образов символов: получение идеальных образов и сравнение с идеальным образом. Предложен способ определения идеальных образов. Описан алгоритм построения функции сравнения с идеальными образами, учитывающей существенные области в начертаниях похожих символов.

Введение

Традиционно [1] задачи распознавания документа подразделяют на сегментацию графических объектов и собственно распознавание, состоящее в оценке гипотез принадлежности к одному или нескольким классам с последующей классификацией на основе полученных оценок. Качество распознавания может быть оценено как точностью классификации, так и информативностью оценок, а также быстродействием [2].

Рассмотрим основные этапы распознавания документа.

Прежде всего, необходимо получить изображение в виде графического файла на компьютере, создав в результате растровое изображение, содержащее копию страницы. При сканировании в результате налипания механического мусора, оптической дефокусировки, электростатических помех и прочего в изображении появляются искажения [3]. Во-первых, буквы не бывают идеально черными, а бумага не бывает идеально белой. Во-вторых, стекло сканера не бывает идеально чистым – всегда присутствуют различные загрязнения. И, наконец, в-третьих, изображение при прохождении сквозь оптическую систему дефокусируется, и на сенсор сканера попадает уже искаженное отражение.

Вторым этапом является бинаризация, суть которой заключается в отделении символов текста (и иных объектов, например, иллюстраций, таблиц, линий разграфки) от фона. Бинаризацией называют преобразование полутонового изображения в черно-белое (бинарное). Общепринятая методология бинаризации состоит в разбиении изображения на области, в каждой из которых фон отсекается от содержания по значению порога, вычисляемого некоторым алгоритмом [4].

Третий этап включает в себя сегментацию страницы на области распознавания: выделение картинок, таблиц, математических формул и текстовых зон, подразделяющихся на строки, а также определение их взаимного расположения [5, 6]. Далее найденные текстовые строки распознаются. Вообще говоря, границы отдельных символов заранее являются неизвестными, поэтому распознавание чередуется с алгоритмами сегментации границ символов.

Описанные этапы распознавания могут повторяться. Например, после проведенной сегментации границ символов возможно формирование последовательности распознанных образов символов, обладающих различной оцененной надежностью классификации [7]. Эта последовательность служит основой для обучения и последующего повторного распознавания текстовых строк, в частности, при поиске альтернативных способов сегментации некоторых слов строки могут разрезаться заново иным образом [8].

Существует большое количество программ, хорошо распознающих четко напечатанные тексты, однако в более сложных случаях задача распознавания далека от решения. При этом существенно лучшие результаты можно получить, если известны «идеальные» образцы каждого символа (шаблоны, templates [9]) и распознавание осуществляется непосредственным сравнением (наложением) реального образа с образцом.

Чтобы воспользоваться этим методом, необходимо решить две задачи: получить идеальные образы символов и разработать алгоритм сравнения предъявляемого для распознавания символа с эталонами. Обе задачи нетривиальны. Рассмотрим их для начала очень бегло.

Идеальный образ символа непросто определить даже в том случае, когда есть его точное описание на каком-либо формализованном языке (например, Postscript) или существует ГОСТ на соответствующую гарнитуру и кегль, определяющий все литеры набора. Дело в том, что описательные эталоны могут дать каждый размер с любой степенью точности, тогда как эталон для сравнения при распознавании должен быть «нарисован» на пиксельной сетке, каждая клетка которой имеет определенный размер.

Пусть, например, мы хотим построить идеальный образ вертикального отрезка прямой шириной w=0,2 мм, ориентируясь на сканирование с разрешением 300 DPI (т.е. 300 точек на дюйм). При этом каждый пиксель отсканированного изображения соответствует на бумаге квадрату размером 85x85 микрон. Очевидно, что никакая фигура на сетке не соответствует точно теоретическому эталону.

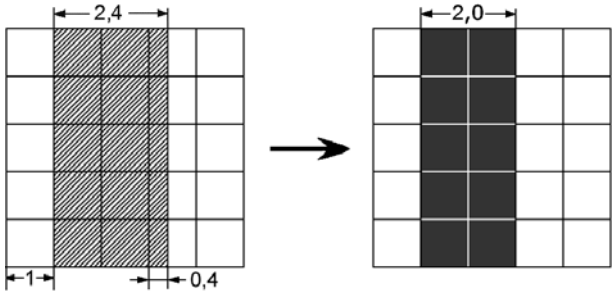


Рис. 1. Построение идеального образа

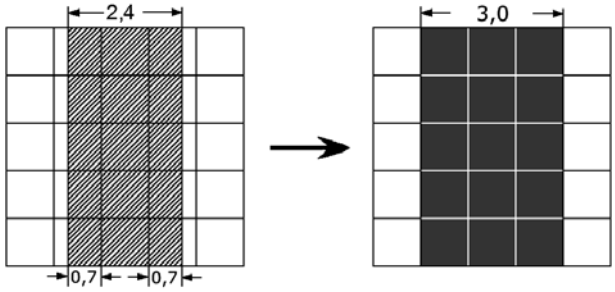


Рис. 2. Построение идеального образа при сдвиге сетки

Мы можем произвести округление и построить идеальный образ, как на Рис. 1, получив вертикальный отрезок шириной в два пикселя. Однако надо понимать, что даже идеально напечатанный символ при сканировании даст другой результат, если сетка будет наложена на него так, как это показано на Рис. 2, и в результате получается вертикальный отрезок шириной в три пикселя.

Таким образом, на границе символа никогда нельзя определить эталон абсолютно «правильно».

На самом деле положение еще хуже, т.к. все принтеры и наборные знаки немного отличаются, в чем легко убедиться, напечатав идентичный текст на разных принтерах и попытавшись совместить их на просвет.

Пусть теперь мы все же сформировали эталон для каждого символа и хотим построить алгоритм распознавания. Наиболее часто используемый прием состоит в наложении распознаваемого образа на каждый из эталонов и вычислении близости образа к эталону в некотором пространстве. Тот образ, для которого расстояние минимально, объявляется наиболее похожим, а соответствующий ему символ – распознанным значением.

Шрифтонезависимое распознавание печатных символов базируется на алгоритмах распознавания образов символов, которые были обучены на последовательности образов, соответствующих некоторому набору шрифтов. Предполагается, что набор шрифтов был достаточно представителен, то есть вероятность появления в тестовых документах незнакомого шрифта является малой величиной.

В реальной системе распознавания документов время, отпущенное на распознавание символов, существенно ограничено. Поэтому шрифтонезависимые алгоритмы ориентированы на ускорение процесса генерации гипотез как за счет уменьшения размерности пространства признаков, так и посредством упрощения механизма поиска. Например, многие нейронные сети используют представление символа в виде растра, отмасштабированного к стандартному размеру 16x16 [11].

Упрощение представления образа приводит к появлению ошибок в коллекции распознавания $X = \{(S_1,P_1), \dots, (S_n,P_n)\}$, где S_i – код распознанного символа, а P_i – его оценка, альтернативы (S_i,P_i) упорядочены. В наихудшем случае в коллекции X отсутствует правильная альтернатива, в наилучшем - нарушен порядок альтернатив, но правильный код символа S_i присутствует в одной из альтернатив. В последнем случае правильный порядок можно установить посредством пересмотра оценок.

Одним из способов пересмотра оценок является использование функций сравнения предъявляемого образа символа сопоставляемому идеальному образу. Возникает вопрос, возможно ли построение функции сравнения автоматически.

1. Построение идеальных образов в адаптивном распознавании

Опишем схему адаптивного распознавания, предлагаемую в [9], которая уже много лет успешно используется в программных продуктах Cuneiform и Cognitive FormReader.

Распознавание документа состоит из двух этапов (проходов). На первом проходе происходит первичная сегментация символов и распознавание одним из шрифтонезависимых алгоритмов. Затем осуществляется обучение (адаптация) шрифтозависимых алгоритмов на символах, показавших высокую достоверность при первом проходе. На втором проходе оценки достоверности для плохо распознанных символов первого прохода пересчитываются. При этом возможна частичная повторная сегментация: если все оценки и после второго прохода остаются очень низкими – это повод предположить, что на первом проходе сегментация была проведена некорректно.

В дальнейшем изложении возможность адаптации (на результатах предыдущего прохода распознавания) является существенным фактором для построения идеального образа для каждой группы. Мы покажем, как в процессе адаптивного распознавания могут быть решены обе перечисленные выше задачи:

1) формирование идеальных образов;

2) создание алгоритма сравнения образа с идеальными эталонами с помощью автоматически построенных функций сравнения.

Идеальные образы – это объекты, обладающие свойством наименьшей средней близости со всеми элементами рассматриваемого множества образов в смысле некоторой определенной функции близости.

Рассмотрим растры $R(m, n) = \| r_{ij} \|$, где ширина m и высота n – размеры растра, $i = \overline{1,m}$, $j = \overline{1,n}$, r_{ij} – значение растра в точке (i, j). Для бинарных изображений значение растра равно 0 или 255, а для полутоновых – принимает целочисленные значения от 0 до 255. Назовем точкой растра (или пикселем) R(m, n) пару чисел (i, j), где i и j – значения индексов элемента r_{ij}.

Для множества $S(\alpha)$ растров одноименных символов α идеальный образ определяется как растр $R_{ideal}(m,n)$, на котором достигается минимум выражения:

$$\frac{1}{N}\sum_{k=1}^{N} \mu\left(R_{ideal}, R_k\right) \rightarrow \min,$$

где N – количество элементов в множестве $S(\alpha)$, а $R_k \in S(\alpha)$. Минимум ищется по всевозможным растрам R(m, n), не обязательно из множества $S(\alpha)$. Заметим, что для бинарных растров идеальный образ не обязан являться бинарным растром. Более того, в нашем дальнейшем изложении идеальный образ всегда

является полутоновым. Функция близости μ зависит от конкретного выбранного алгоритма вычисления расстояния между двумя растрами.

На практике допускается наличие в множестве S(α) небольшого количества растров, отличных от α. Также предполагается, что множество растров S(α) является репрезентативной выборкой для символа α. При наличии в множестве S(α) большого количества символов, отличающихся от α, идеальный образ не определен.

Вследствие своих свойств идеальный образ может использоваться для идентификации еще не подвергнутого классификации символа по признаку принадлежности к тому или иному классу. В процессе распознавания его удобно применять для определения является ли данный образ тем же самым символом, что и растры, на основе которых построен идеальный образ.

В качестве идеального образа в алгоритме адаптивного распознавания [9] используют взвешенный растр. Два растра $R_1(m_1,n_1)$ и $R_2(m_2,n_2)$ будем называть равновеликими, если $m_1=m_2$ и $n_1=n_2$. Пусть M - произвольное множество N равновеликих растров $M = \{R_k(m,n)\} = \{\| r_{ij}^{(k)} \|\}$, где $k=\overline{1,N}$, $i=\overline{1,m}$ и $j=\overline{1,n}$. Тогда взвешенным растром для множества M назовем растр $R(m, n) = \|r_{ij}\|$ такой, что

$$r_{ij} = \frac{1}{N}\sum_{k=1}^{N} r_{ij}^{(k)} .$$

Взвешенный растр зависим от распознаваемых символов; в то же время зависимость от отдельно взятого представителя сведена к минимуму - эти свойства объясняются описанным способом построения взвешенного растра. Тем самым действительно можно утверждать, что взвешенный растр является вполне естественным идеальным образом для группы схожих символов.

На практике после сканирования даже одноименные символы одного шрифта имеют различные линейные размеры, поэтому для получения взвешенного растра необходимо приведение размеров всех растров коллекции S к единому шаблону. Поскольку для каждого растра сделать это можно не единственным образом, возникает задача выравнивания символа в расширенном растре непосредственно перед суммированием. В результате для произвольной группы неравновеликих растров взвешенный растр, вообще говоря, не является определенным однозначно. Отдельные аспекты построения взвешенного растра в этом случае подробно рассмотрены в [12, 13]. Способ укладки взвешенного растра может заметно повлиять на точность распознавания.

Пусть идеальные образы получены и используются в качестве эталонов - будем сравнивать с ними неуверенно распознанные на первом проходе символы (то есть символы, получившие низкие оценки). Сравнение осуществляется, как уже говорилось выше, путем наложения распознаваемого образа на каждый из эталонов и вычисления близости в некотором метрическом пространстве.

Функция сравнения может быть определена различными способами. Наиболее естественным кажется использование интегральной метрики – попиксельной суммы модулей разности. Однако у этой метрики есть явный недостаток: отличие символов от эталона на границе часто бывает несущественным за счет того, что случайные изменения символов именно на их границе при сканировании наиболее вероятны. В качестве примера достаточно рассмотреть полоску, которая попадает на сетку сканера таким образом, что закрывает по полпикселя на всем протяжении каждой из границ. При таком положении в процессе сканирования каждый пиксель на границе полоски с примерно равной вероятностью окажется либо черным, либо белым (пример на Рис. 3). Мы предположили независимость колебания в соседних пикселях – на самом деле это не так, но это все равно не влияет на итоговый вывод.

При сравнении такого образа с эталоном именно граница внесет наибольший вклад в интегральную сумму. Если наш символ является квадратной скобкой **[**, то из-за существенного вклада в интегральную сумму граничных пикселей влияние горизонтальных штрихов просто «затеряется», и этот символ будет отличаться от квадратной скобки не меньше, чем от вертикальной черты |. Аналогичные проблемы в той или иной степени появляются в случае необходимости различить буквы **Н** и **П**, **Б** и **В**, **О** и **С** и т.д.

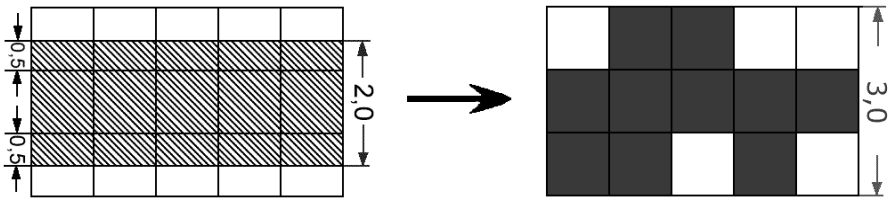


Рис. 3. Большая величина отличия от эталона при сканировании полоски

Указанный недостаток отсутствует у альтернативной функции-сравнения растров H(A,B). Функция отличия H(A,B) между двумя растрами A и B определяется как сумма двух величин. Первая величина показывает, на сколько пикселей надо расширить A′, чтобы в него поместился B. Вторая величина – на сколько пикселей надо расширить B′, чтобы в него поместился A, где растры A′ и B′ суть расширенные на 1 пиксель соответственно растры A и B. Более формально: определим для каждого растра R функцию расширения I_{ij} (R)=1, если $r_{i,j}=1$, или $r_{i-1,j}=1$, или $r_{i+1,j}=1$, или $r_{i,j+1}=1$, или $r_{i,j+1}=1$. В противном случае I_{ij} (R)=0. Для растров A и B альтернативная функция сравнения равна:

$$H(A,B)=\sum_{i,j}(S(I_{ij}(A),b_{ij})+S(I_{ij}(B),a_{ij})),$$

где

$$S(a,b)=\begin{cases}0, a>b\\ 1, a\le b\end{cases}.$$

Чем меньше значение неотрицательной функции H(A,B), тем лучше накладываются друг на друга растры A и B. Однако равенство H(A,B) нулю не означает полного совпадения растров.

На тестах с символами, размеры которых более 15x10 (размеры соответствуют разрешению 150dpi для шрифта 12pt), эта мера показывает лучшие результаты, чем простая интегральная (в большей мере это отличие выражено для черно-белых растров).

Однако при небольших размерах символов их отличие друг от друга становится сравнимым с отличием одноименных символов между собой. Причем для альтернативной функции H(A,B) этот порог при уменьшении символов достигается раньше, чем для интегральной метрики, поскольку отличие мелкоразмерных символов друг от друга часто как раз и находится в пределах расширения.

Вновь возникает вопрос о необходимости «смотреть» на отдельные участки символов для выявления отличия похожих символов типа «**Н И П**», «**О С**», «**a s e**» и т.п.

Перед тем как предложить алгоритм автоматического построения функций сравнения, мы дадим оценку применимости произвольной интегральной метрики для отличия двух растров в контексте распознавания.

2. Отличие от идеальных образов

Пусть имеется идеальный образ. Заметим, что в случае с взвешенным растром мы имеем дело с полутоновым изображением.

Будем сравнивать бинаризованый после сканирования символ и бинаризованый идеальный образ. При этом в силу случайных искажений неминуемо возникнет некоторая ошибка (количество не совпадающих пикселей). Поставим вопрос – какова вероятность P(T) того, что ошибка равна T?

Будем считать, что положение сканируемого символа на сетке сканера жестко определено, т.к. используется привязка эталона к пиксельной сетке. Возможные сдвиги уже учтены во взвешенном растре. Тем самым можно говорить о некотором среднем положении символа внутри растра.

Проведенные нами эксперименты подтверждают, что при известной интенсивности λ в некоторой ограниченной области сканирования вероятность того, что эта область реализуется в пиксель интенсивности k, приближается случайной величиной с распределением Пуассона [10]. Функция вероятности соответственно равна:

$$p_k=\frac{\lambda^k e^{-\lambda}}{k!}.$$

Различные модели сканеров могут искажать эту зависимость, незначительно нарушая ее, но

мы не будем сейчас на это отвлекаться. Можно считать, что:

$$p_{255} = \sum_{k=255}^{\infty} p_k = 1 - \sum_{k=0}^{254} p_k ,$$

так как в случае k > 255 происходит переполнение сенсора сканера и пиксель получается абсолютно белым.

Пусть порог бинаризации равен B. Если λ ≥ B, то вероятность возникновения ошибки в этом пикселе после бинаризации равна:

$$P_\lambda = P(\lambda, \mathbf{B}) = \sum_{k=0}^{\mathbf{B}-1} p_k , \qquad (1.1)$$

то есть, отклонение будет реализовываться, если значение пикселя после сканирования будет меньше B. Аналогично при λ < B имеем:

$$P_\lambda = P(\lambda, \mathbf{B}) = \sum_{k=\mathbf{B}}^{255} p_k . \qquad (1.2)$$

Заметим, что рассматривать P_0 не имеет смысла, так как при нулевой интенсивности λ ничего, кроме пикселя нулевой яркости, получиться не может (на практике значение λ=0 не встречается – это абсолютно черная поверхность). Рассмотрим n пикселей с одинаковой интенсивностью λ.. По известной формуле получаем вероятность реализации отклонения ровно на s единиц (0 ≤ s ≤ n):

$$C_n^s P_\lambda^s (1 - P_\lambda)^{n-s} ,$$

где $P_\lambda = P(\lambda, B)$ и равно (1.1) или (1.2) в зависимости от соотношения λ и B, а C_n^s – биномиальный коэффициент.

После дополнительных выкладок получаем итоговую формулу вероятности P(T):

$$\sum_{\{b_1, b_2 \ldots b_{255}\} \in \mathbf{W}} \prod_{i=0}^{255} C_{\alpha_i}^{b_i} P_i^{b_i} (1 - P_i)^{\alpha_i - b_i} \qquad (2)$$

где $b_1, \ldots, b_{255}$ – решение в неотрицательных целых числах уравнения $b_1 + b_2 + \ldots + b_{255} = T$, W – множество этих решений, α_i - количество пикселей интенсивности λ в растре.

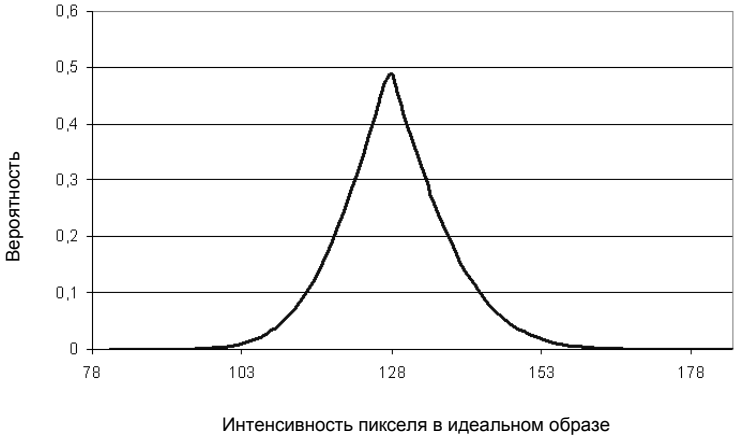


Рис. 4. Функция вероятности ошибки в одном пикселе

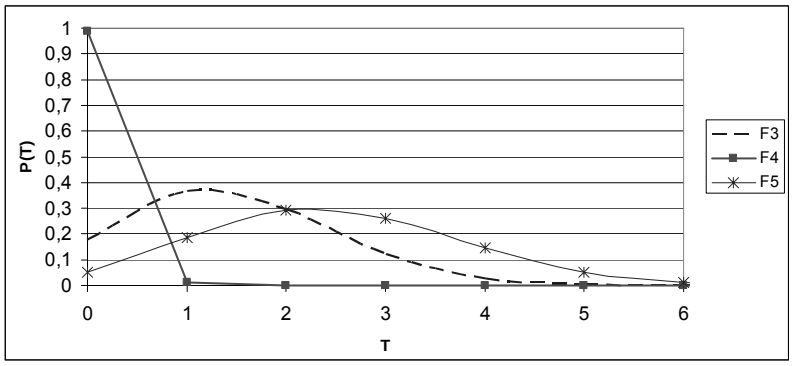


Рис. 5. Примеры функций вероятности ошибки

На Рис. 4 изображен график P_λ при уровне бинаризации B, равном 128.

Как видно из графика, наибольшая вероятность наблюдается в окрестности уровня бинаризации. Именно этим объясняется повышенная вероятность отклонения на границе символа.

В рассуждениях выше мы считали, что идеальный образ полностью определяет положение символа на сетке сканера. На Рис. 5 приведен график (2) для нескольких реальных символов F (шрифт Arial, 12pt, разрешение сканирования – 150 dpi).

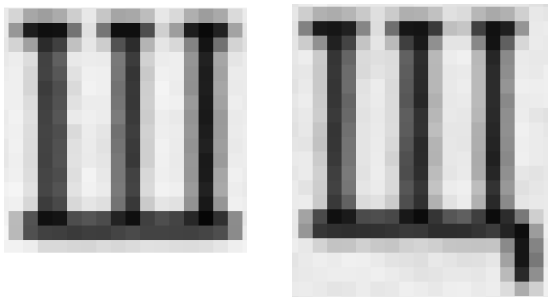
Символ, соответствующий графику F4, хорошо ложился на пиксельную сетку, и все пиксели были либо почти черными, либо почти белыми. В итоге вероятность возникновения нулевого отклонения практически равна 1, чего нельзя сказать про другие символы. Наоборот,

в случаях F3 и F5 вероятность того, что ошибки не будет, в 2-3 раза меньше вероятности наличия некоторой ошибки.

При увеличении размеров символов (например, эти же символы сканируются при разрешении 300 dpi) нулевая величина ошибки (как в случае с F4) практически не встречается. Одновременно с этим наиболее частое значение случайной величины P(T) сдвигается вправо все дальше от начала координат.

Из изложенного выше видно, что каков бы ни был символ, найдется такой его отсканированный образ, что вероятность P случайной ошибки некоторой ненулевой величины T будет заметно больше нуля. Этот факт очевиден, т.к. найдется такое положение символа на пиксельной сетке, что часть пикселей будет накрыта примерно наполовину. Более того, эта вероятность ошибки P(T) будет принимать существенное для практических задач распознавания текста значение.

Для иллюстрации сказанного рассмотрим примеры реальных отсканированных символов Ш и Щ (шрифт Courier New, 12pt, 150 dpi). На Рис. 6 мы привели символы сразу после сканирования (а) и их бинаризации (б).



а)

б)

Рис. 6. Пример близких образов

Сравнив бинаризованые символы Ш и Щ, видим, что они отличаются только «хвостиком» у буквы Щ, который состоит всего лишь из четырех пикселей.

По формуле (2) вычислим оценку ошибки на величину T. Величина ошибки в данном случае показывает, каково могло бы быть отличие после бинаризации вновь отсканированной буквы Ш при условии идентичного положения прообраза относительно пиксельной сетки. Формула (2) дает вероятность ошибки P(T) для каждого T.

Из графика на Рис. 7 видно, что вероятность случайной ошибки для буквы Ш ровно на 4 единицы превышает 1/5. Суммарная же вероятность ошибки на 4 или более единиц – превышает ½. На Рис. 7 приведен график для еще одного реального образа символа Ш (Ш_2), чтобы показать, что величина ошибки может быть еще больше.

Этот эксперимент показывает, что программы распознавания текста, использующие любую интегральную меру, непременно будут ошибаться на некоторых входных данных. В то же время любой носитель языка легко различит эти символы. Тем самым, невозможно обойтись без использования анализа существенных областей при сравнении символов в процессе распознавания.

В следующей главе мы приведем алгоритм, позволяющий в рамках адаптивного распознавания производить автоматическое обнаружение существенных областей у плохо различимых пар символов и производить сравнение символов с учетом этих особенностей.

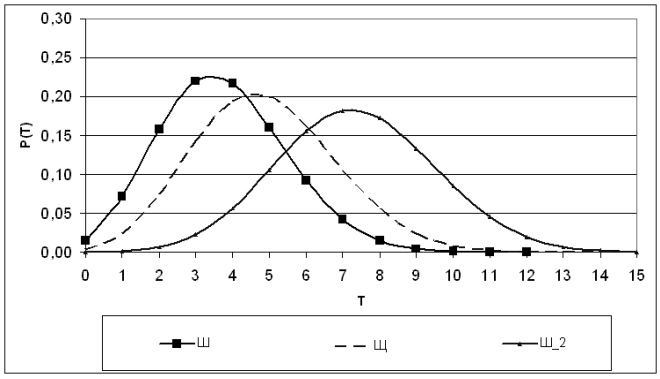


Рис. 7. Вероятность отклонения ровно на T единиц (буквы Ш и Щ, 12pt, 150 dpi)

3. Алгоритм построения функции сравнения

Предложим алгоритм построения функции для сравнения распознаваемого символа с идеальными образами $\overline{S_i}$ символов S_i из коллекции альтернатив $X = \{(S_1,P_1), \dots , (S_n,P_n)\}$. Данное сравнение на втором проходе адаптивного распознавания применяется при пересчете оценок P_i только для плохо распознанных на первом проходе символов, другими словами, для символов, которые получили низкие оценки P_i для всех S_i или наибольшие оценки которых P_1 и P_2 примерно равны.

Как правило, реальное отличие символов сосредоточено в небольшой области или объединении нескольких областей, относительный вклад которых в общую площадь, занимаемую символами, невелик. Например, горизонтальные перемычки букв **Н** и **П** достаточно тонкие по сравнению с остальными элементами символов.

При небольших размерах символов (или при малых разрешениях сканирования) подобные тонкие перемычки могут разрываться, оставляя на своем месте лишь отдельные пиксели. И наоборот - засечки могут слипаться, образуя непрерывную группу пикселей. Не используя контекста, отличить такие символы друг от друга можно только по отдельным пикселям. Тем самым, вклад этих пикселей в значение функции сравнения должен быть увеличен. Наоборот, вклад тех пикселей, которые совпадают у альтернатив-кандидатов, необходимо свести к минимуму или вообще обнулить.

Имея лишь по одному экземпляру образов двух символов, которые надо различить, не возможно без априорных знаний предполагать что-либо о ценности тех или иных элементов символов. Поскольку взвешенный растр составлен из нескольких образов символа, будем использовать устойчивость взвешенного растра к искажениям отдельных символов, из которых он составлен. Действительно, если у малой части символов, составляющих взвешенный растр, слипнутся засечки (Н П), произойдет разрыв перемычки, исчезнет «хвостик» (буква Щ) и т.п., то взвешенный растр в этом случае изменится несущественно, сохранив и «хвостик», и засечки, и перемычку.

Легко понять, что сами образы символов S_i из коллекции альтернатив X будут похожи друг на друга в том смысле, что при сравнении соответствующих им идеальных образов $\overline{S_i}$ последние будут отличаться незначительно.

Предполагается, что существует контроль за тем, чтобы взвешенные растры формировались из хорошо распознанных символов. Это предположение существенно, т.к. если взвешенный растр $\overline{S_i}$ составлен из кластера, наполовину разбавленного символами, отличными от S_i, то $\overline{S_i}$ уже не будет идеальным образом одного символа.

Итак, для распознаваемого растра R имеется коллекция альтернатив X, для каждого символа S_i которой построен идеальный образ $\overline{S_i}$. Без ограничения общности можно считать, что взвешенные растры $\overline{S_i}$ равновелики с R.

Приведем общую схему алгоритма:

1). Идеальные образы $\overline{S_i}$ накладываются друг на друга, и ищется наилучшее их взаимное расположение. Для двух растров наилучшим называется такое положение, при котором расстояние между ними по интегральной метрике минимально. Вначале на первый растр накладывается второй, затем на первый растр накладывается третий, четвертый и т.д.

2). В найденном наилучшем положении, если это необходимо, растры $\overline{S_i}$ приводятся к одинаковому (расширенному) размеру.

3). Строится матрица $M' = \|m'_{jk}\|$, каждый элемент которой равен максимальной разности соответствующих элементов растров $\overline{S_i}$:

$$m'_{jk} = \max_i(\overline{S}_i(j,k)) - \min_i(\overline{S}_i(j,k)),$$

где $\overline{S_i}(j,k)$ - значение растра $\overline{S_i}$ в точке (j,k).

4). Вторая матрица $M=\|m_{jk}\|$ заполняется штрафными коэффициентами. В каждом пикселе штраф вычисляется как неубывающая функция f(x) от значения соответствующего элемента M′, т.е. $m_{jk} = f(m'_{jk})$.

5). Ищется наилучшее положение растра R на каждом $\overline{S_i}$, при этом используется интегральная метрика.

6). В найденном в пункте (5) положении для каждого $\overline{S_i}$ вычисляется штраф за несовпадение с растром R. В пикселе с координатами (j,k) штраф определяется как произведение $m_{jk} \times \left|\overline{S_i}(j,k) - r_{jk}\right|$ соответствующего элемента штрафной матрицы M и модуля разности значений растра R и идеального образа $\overline{S_i}$ (M жестко зафиксирована по отношению к $\overline{S_i}$). Считается сумма штрафов по всем пикселям растра R(m, n):

$$\sum_{j=1}^{m}\sum_{k=1}^{n} m_{jk} \times \left|\overline{S_i}(j,k) - r_{jk}\right|.$$

7). Из множества альтернатив S_i результатом распознавания считается тот символ, для идеального образа $\overline{S_i}$ которого в пункте (5) получен минимальный штраф.

Как можно заметить, результат наложения последующих растров в пункте (1) зависит от выбора первого. Несложно придумать ситуацию, при которой изменение порядка наложения изменит в результате матрицу M′. Для большей постоянности результата в этом случае предлагается реализовать «второй проход» при накладывании растров, т.е. первым растром считать среднее арифметическое идеальных образов $\overline{S_i}$.

Выбор в пункте (4) неубывающей функции f(x) является существенным, ее окончательный вид приходится выбирать опытным путем на представительной выборке документов.

Мы использовали следующую штрафную функцию:

$$f(x) = \begin{cases} 0, x \le a \\ (x-a)/(B-a), x > a \end{cases},$$

где a – порог отказа от фона, B – порог бинаризации. Цель функции f(x) – усилить влияние существенных точек на величину штрафа при сравнении распознаваемого символа R с идеальными образами $\overline{S_i}$.

Итак, среди аргументов итоговой функции сравнения кроме самого растра R будут идеальные образы $\overline{S_i}$ для всех символов S_i из коллекции альтернатив X, а значениями будут пересчитанные оценки P_i для R.

Заключение

В работе поставлены и решены две задачи - получить идеальные образы символов и разработать алгоритм сравнения символа с эталонами.

В качестве идеальных образов предложено использовать взвешенные растры.

Обоснована необходимость использования специальной функции сравнения для схожих по начертанию символов, т.е. показано, что невозможно обойтись без анализа существенных областей при сравнении похожих символов в процессе распознавания. В итоге предложен алгоритм для автоматического построения специальных функций сравнения для произвольного (но достаточно представительного) множества символов – вне зависимости от языка распознавания и конкретных особенностей отдельных символов.

С помощью приведенного алгоритма была существенно улучшена схема адаптивного распознавания с дообучением. Например, в классе печатных документов среднего качества точность распознавания символов (без различия на стоящие отдельно и сегментированные) возрастает с 99.6-99.7% до 99.7-99.8%. При этом увеличение точности произошло именно за счёт улучшения классификации образов символов, обладающих сходным начертанием.

Авторы выражают признательность В.Л. Арлазарову и А.Я. Подрабиновичу за поддержку исследований и плодотворное обсуждение.

Литература

1. Арлазаров В.Л., Славин О.А. «Алгоритмы распознавания и технологии ввода текстов в ЭВМ». Информационные технологии и вычислительные системы, 1996. № 1. С. 48-54
2. Арлазаров В.Л., Логинов А.С., Славин О.А. Характеристики программ оптического распознавания текста // Программирование №3, 2002. С. 45-63

3. Титов Ю.В. Об искажении символов при сканировании // Сб. тр. ИСА РАН «Системный подход к управлению информацией», 2006. С. 260-288.
4. Trier Ø. D., Taxt T. Evaluation of Binarization Methods for Document Images // IEEE Transactions on pattern analysis and machine intelligence, vol. 17, No 3, March 1995. P. 312-315
5. Breuel T. M. An Algorithm for Finding Maximal Whitespace Rectangles at Arbitrary Orientations for Document Layout Analysis. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03), NJ, USA, 2003. P. 66-71
6. Kise K., Yanagida O., Takamatsu S. Page Segmentation Base on Thinning of Background // In Proc. of the 13th International Conference on Pattern Recognition, page 788-792, Vienna, Austria, August 1996. P. 788-792
7. Savachynskyy B, Kamotskyy O. Character templates learning for textual images recognition as an example of learning in structural recognition,// Proceeding of the Second International Conference on Document Image Analysis for Libraries (DIAL'06), 1997. P. 88-95.
8. Lebourgeois F., Henry J. L. An Evolutive OCR System Based on Continuous Learning // Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), December 1996. P. 272-277
9. Арлазаров В.Л., Котович Н.В., Славин О.А. Адаптивное распознавание // Информационные технологии и вычислительные системы № 4, 2002. С. 11-22
10. Постнов К.А. Лекции по Общей астрофизике для физиков, курс лекций 2001 г., физический факультет МГУ http://www.astronet.ru/db/msg/1170612/3lec/node5.html
11. Мисюрёв А.В. Использование искусственных нейронных сетей для распознавания рукопечатных символов. // Сб. трудов ИСА РАН "Интеллектуальные технологии ввода и обработки информации", 1998, С.122-127
12. Титов Ю.В. О восстановлении идеального прообраза по коллекции образов // Сб. тр. ИСА РАН «Системный подход к управлению информацией», 2006. С. 252-259.
13. Sawaki M., Hagita N., Ishii K. Robust Character Recognition of Gray-Scaled Images with Graphical Designs and Noise // Proceedings of Fourth International Conference Document Analysis and Recognition (ICDAR'97), 1997. P. 491-494.

Славин Олег Анатольевич. Окончил МИРЭА в 1988 году. Кандидат технических наук с 2000 года. Автор 50 научных работ и изобретений. Область научных интересов - распознавание образов, искусственный интеллект, моделирование электромагнитных процессов. Заведующий лабораторией в ИСА РАН.

Титов Юрий Васильевич. Окончил МГУ им. М.В.Ломоносова в 2003 году. Автор 4 научных работ и изобретений. Область научных интересов - распознавание образов, автоматическая классификация текстов, искусственный интеллект. Аспирант ИСА РАН.