

## Поиск размеров эталонов при распознавании текстовых изображений

*Савчинский Б.Д., Олефиренко С.А.\**

### Аннотация

Предложен подход к оценке эталонных изображений букв в задаче распознавания текста на основе обучающей выборки, состоящей из примеров входных изображений и соответствующих им текстовых строк. Предложенный подход требует решения двух задач: поиск размеров эталонов букв и оценка значений эталонов. В работе вторая задача формулируется и решается как задача самообучения, первая – как задача поиска оптимальной разметки на графе. Показывается, что полученная  $(\min,+)$  задача разметки является субмодулярной, а значит, разрешимой.

### Анотація

Запропоновано підхід до оцінки еталонних зображень літер в задачі розпізнавання тексту на основі навчальної вибірки, що складається із прикладів вхідних зображень та відповідних їм текстових рядків. Запропонований підхід потребує вирішення двох задач: пошук розмірів еталонів літер і оцінка значень еталонів. В роботі друга задача формулюється і розв'язується як задача самонавчання, перша – як задача пошуку оптимальної розмітки на графі. Показано, що отримана  $(\min,+)$  задача розмітки є субмодулярною, а, отже, розв'язною.

### Abstract

An approach to the estimation of character template images in the problem of text recognition is proposed where learning sample contains examples of images and text that corresponds to them. The proposed approach requires solving of two problems: estimation of templates sizes and estimation of template values. In the paper the second problem is formulated and solved as the problem of self-learning, and the first one – as the problem of finding the optimal labeling on a graph. It is shown that obtained  $(\min,+)$  labeling problem is submodular and, therefore, solvable.

### Введение

Работа посвящена обучению эталонов символов в задаче распознавания изображений печатных текстовых строк. Несмотря на почтенный возраст проблемы распознавания текстов, существует целый ряд практически важных задач обучения,

---

\* Работа выполнялась в рамках проекта EU INTAS PRINCESS 04-77-7347

формальные постановки которых либо не изучены, либо изучены недостаточно глубоко. Важность этих задач является несомненной, поскольку, несмотря на широкий спектр как методов распознавания текстовых изображений, так и соответствующего программного обеспечения, распознавание сильнозашумленных изображений или даже просто изображений, содержащих символы в нестандартном начертании, написанных на языках, отличных от европейских, представляет определённые трудности. Эти трудности можно было бы легко решить, если бы наряду с эффективными методами для распознавания текстов существовали эффективные методы обучения распознаванию.

Задача обучения распознаванию печатных текстов (для языков, имеющих алфавит) состоит преимущественно в построении эталонов символов этого текста. Существует целый ряд подходов к её решению (см. [1–5]), но большинство из них имеют одну особенность, которая затрудняет их использование на практике. Эта особенность содержится в формате обучающего множества, а именно, оно состоит из входных изображений, для каждого из которых указана его сегментация на изображения отдельных символов. Качество обучения во многом зависит от качества этой сегментации: для получения хороших результатов она должна быть сделана довольно тщательно, каждый символ должен быть строго центрирован, сегменты, отвечающие одному и тому-же символу, должны иметь одинаковые размеры. Получение такой сегментации является наиболее сложным этапом создания обучающего множества.

В нашей работе предложен иной подход к обучению, в котором не требуется точной сегментации изображения на части, содержащие отдельные символы. А именно, обучающая выборка состоит из изображения текстовой строки и соответствующей ей последовательности символов, т.е. собственно самой текстовой строки. Получение таких обучающих данных требует значительно меньших усилий, что, в свою очередь, обозначает значительное уменьшение труда по настройке алгоритма распознавания на новый тип текста.

Кроме этого, мы предлагаем подход к обучению размеров эталонов символов текста. Это избавляет оператора от ручного измерения этих размеров, что тоже составляет важную и довольно значительную часть ручной работы по подготовке обучающего множества.

Задачи подобного типа ставились в работах [6–9], но алгоритмы их решения ориентированы на двухуровневые, чёрно-белые изображения. Мы предлагаем подход, вообще говоря, не зависящий от количества цветов на изображении, но для определённости в выкладках ограничимся случаем изображений, заданных в оттенках серого цвета.

Наконец, отметим, что мы рассматриваем текстовые изображения как прекрасный полигон для отработки и тестирования новых методов в структурном распознавании. Такие качества эти изображения имеют благодаря своей чётко выраженной структурированности и, одновременно, сравнительной простоте по

сравнению с другими классами структурированных изображений. Поэтому целью данной работы является не только и не столько разработка определённых методов обучения в распознавании текстов, как разработка методов обучения распознаванию структурированных изображений вообще и тестирование этих методов на текстовых изображениях.

Предлагаемый нами метод решения задачи обучения состоит в итеративном решении двух подзадач: (i) оценивании раскраски эталонных изображений символов текста при условии фиксированных известных размеров этих эталонов и (ii) оценивании размеров эталонов символов текста. Для решения первой подзадачи мы предлагаем использовать метод самообучения (см. [10–12]), сходно с тем, как это делается в работах [6–9]. Существенной новизной обладает постановка второй подзадачи, которую мы формулируем, как задачу отыскания оптимальной разметки (см. [13–16]), в общем случае NP-полную, но в нашем частном случае, как это будет показано, принадлежащую полиномиально разрешимому классу супермодулярных задач максимизации (см. [17–21]).

Работа состоит из четырёх частей. В первой часть введены основные определения, а также кратко сформулирована и описан способ решения задачи обучения эталонных изображений символов текста при условии фиксированных известных размеров этих эталонов. Вторая часть посвящена задаче обучения в общем виде, оговорены возникающие трудности и предложен способ решения путем декомпозиции задачи на подзадачи. В третьей части показано, каким образом оценивание размеров эталонов символов текста сводится к полиномиально разрешимой задаче отыскания наилучшей разметки. Наконец, четвёртая часть посвящена экспериментальной проверке построенных методов.

## 1 Основные определения и постановки задач

### 1.1 Задачи распознавания и обучения

Введем основные определения и обозначения, которые будут использоваться в работе.

*Поле зрения*  $T$  назовем конечное множество пар координат пикселей:

$$T = \{(i, j) \mid i = \overline{0, W-1}, j = \overline{0, H-1}\}.$$

Величины  $W$  и  $H$  будем называть, соответственно, *шириной* и *высотой* поля зрения. Также на множестве пикселей поля зрения будет считаться определенной естественным образом операция векторного сложения:  $(i, j) + (i', j') = (i + i', j + j')$ .

*Изображением* назовем функцию  $x: T \rightarrow \mathbb{R}_+$ . Величину  $x(t), t \in T$  будем называть *цветом пикселя*. Мы будем различать два типа изображений: изображения, поданные на распознавание, и эталонные изображения букв, о которых речь пойдет ниже в этом разделе. Будем считать, что все изображения имеют одинаковую высоту

$H$ , но их ширины, вообще говоря, отличаются.

Определим модель изображения строки текста. Будем считать, что это изображение строится путем наложения аддитивного гауссовского шума на незашумленное изображение независимо во всех пикселах. В свою очередь, незашумленное изображение, соответствующее заданной текстовой строке, является горизонтальной последовательностью эталонных изображений букв строки, причем эти изображения не перекрываются, а возможные промежутки между ними заполняются определенным цветом (цветом фона).

Конечное множество  $A_0$  букв текста будем называть *алфавитом*. Последовательность элементов алфавита  $\bar{c} = (c_1, c_2, \dots, c_{N_{\bar{c}}})$ ,  $c_n \in A_0$ ,  $n = \overline{1, N_{\bar{c}}}$  назовем *текстовой строкой*. Для описания промежутков между изображениями букв введем дополнительный элемент алфавита — *вставку*  $\kappa$ . Новое множество  $A_0 \cup \{\kappa\}$  обозначим через  $A$ , а его элементы будем называть *символами*. Таким образом, символ — это либо буква алфавита, либо вставка.

С каждым символом  $a \in A$  свяжем его эталонное изображение  $e_a$  высоты  $H$  и ширины  $d(a)$ , вообще говоря, различной для разных символов. Эталон вставки имеет ширину  $d(\kappa) = 1$ . Множество эталонных изображений обозначим через  $E$ .

Назовем *сегментом*  $s = (i, a)$  поименованный прямоугольный фрагмент поля зрения, который определяется координатой левого столбца  $i = i(s)$ ,  $i \in \overline{0, W - d(a)}$  и именем  $a = a(s)$ ,  $a \in A$ , при этом имеет высоту  $H$  и ширину  $d(a(s))$ . Множество всех сегментов обозначим  $S$ . С каждым сегментом свяжем элемент  $t_s = (i, 0)$  поля зрения, соответствующий левому верхнему углу сегмента  $s$ .

*Сегментацией* изображения назовем последовательность сегментов  $\bar{s} = (s_1, \dots, s_N)$  произвольной длины  $N$ , которые покрывают все поле зрения и расположены вплотную друг к другу:

$$\begin{cases} i(s_1) = 0; \\ i(s_{n+1}) = i(s_n) + d(a(s_n)), \quad n = \overline{1, N-1}; \\ i(s_N) + d(a(s_N)) = W. \end{cases}$$

Множество всех сегментаций обозначим  $\bar{S}$ . Понятно, что при известных ширинах эталонов сегментация  $\bar{s}$  полностью определяется последовательностью символов  $(a(s_1), \dots, a(s_N))$ . Иногда мы будем называть сегментацией именно эту последовательность.

Для каждой сегментации  $\bar{s} = (a_1, \dots, a_N)$  можно получить соответствующую ей текстовую строку удалением всех символов-вставок. И в обратную сторону, каждой текстовой строке  $\bar{c}$  соответствует множество сегментаций, отличающихся друг от друга только вставками. Будем обозначать это множество  $\bar{S}(\bar{c})$ . Также для сегментации  $\bar{s} \in \bar{S}(\bar{c})$  будем обозначать ее сегменты, отвечающие буквам  $(c_1, c_2, \dots, c_{N_{\bar{c}}})$

строки  $\bar{c}$ , соответственно  $(s(c_1), s(c_2), \dots, s(c_{N_{\bar{c}}}))$ .

Сегментация  $\bar{s}$  определяет изображение, составленное путем стыковки по горизонтали изображений эталонов, соответствующих символам сегментации. Введем штраф за сегментацию  $\bar{s}$  как сумму квадратов отклонений цветов пикселей такого составного изображения от пикселей изображения  $x$  тех же размеров:

$$f(x, \bar{s}, E) = \sum_{n=1}^{N(\bar{s})} \sum_{t \in T(s_n)} (x(t_{s_n} + t) - e_{a(s_n)}(t))^2, \quad (1)$$

где при помощи  $N(\bar{s})$  обозначено количество сегментов в сегментации  $\bar{s}$ , а при помощи  $T(s)$  — прямоугольный фрагмент поля зрения, который отвечает сегменту  $s$ .

**Задача 1** (*распознавание изображения  $x$* ) Для заданных входного изображения  $x$  и множества эталонов  $E$  найти сегментацию, имеющую минимальный штраф:

$$\bar{s}^* = \arg \min_{\bar{s}} f(x, \bar{s}, E).$$

Известно, что эта задача решается при помощи динамического программирования [1, 11].

Параметром этой задачи является множество  $E$ , т.е. эталонные изображения символов. Целью данной работы является построение этих изображений на основании обучающего материала. Каждое изображение определяется своими размерами и цветами пикселей. В работе [22] указан способ вычисления цветов пикселей при условии фиксированных размеров эталонных изображений. Краткое описание этого метода приведено в разделе 1.2. Основная часть статьи (разделы 2, 3) посвящена определению размеров эталонов. В разделе 2 также описано совместное использование этих двух методов, т.е. одновременное определение размеров эталонов и цветов их пикселей.

## 1.2 Построение эталонов символов известных размеров

Как правило, построение эталонов происходит на основании обучающего множества, содержащего изображения и соответствующие им сегментации ([1, 11, 23, 24]). Эти сегментации содержат информацию как о расположении букв, так и символов вставки между ними. Построение таких сегментаций требует значительной ручной работы, так как необходимо указывать точное положение каждого символа на изображении. Значительно меньших усилий требует указание для данного изображения соответствующей ему строки текста.

Сформулируем эту задачу. Пусть имеется обучающее множество  $D$ , состоящее из входных изображений  $x^1, x^2, \dots, x^M$  и соответствующих им текстовых строк  $\bar{c}^1, \bar{c}^2, \dots, \bar{c}^M$ . Пусть  $p(D; E)$  — вероятность обучающего множества  $D$ ,

параметризованная известными эталонными изображениями  $E$ . Задача построения эталонов состоит в поиске таких их значений  $E^*$ , которые максимизируют вероятность обучающего множества  $D$ :

$$E^* = \arg \max_E p(D; E). \quad (2)$$

В работе [22] для ее решения использован итеративный алгоритм самообучения [10, 12], для случая, когда

$$p(D; E) = \prod_{m=1}^M \sum_{\bar{s} \in \bar{S}(\bar{c}^m)} p(x^m, \bar{s}; E), \quad p(x, \bar{s}; E) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ - \sum_{n=1}^{N(\bar{s})} \sum_{t \in T(s_n)} \frac{(x(t_{s_n} + t) - e_{a(s_n)}(t))^2}{2\sigma^2} \right\}.$$

Там же приведены практические примеры использования этого алгоритма. В этой работе мы продолжаем использовать модель шума, описываемую указанным распределением.

## 2 Поиск размеров эталонов

Использование задачи (2) на практике ограничивается необходимостью задавать размеры эталонов. В нашей постановке высота эталонов известна и равна высоте входных изображений, однако необходимо задавать ширины эталонов. Ручное определение этих размеров является неестественным и трудоемким. В этом разделе мы модифицируем задачу (2) для автоматической настройки ширин эталонов вместе с их расцветкой.

Пусть  $D$  — обучающее множество,  $\bar{w} = \{d(\alpha) | \alpha \in A\}$  — вектор, содержащий ширины всех символов;  $E(\bar{w})$  — эталоны символов, ширины которых определяются вектором  $\bar{w}$ . Необходимо найти такой набор эталонов, определяемых ширинами  $\bar{w}^*$  и расцветкой  $E^*(\bar{w}^*)$ , которые максимизируют  $p(D; \bar{w}, E(\bar{w}))$  — вероятность обучающего множества при условии заданных эталонов:

$$(\bar{w}^*, E^*(\bar{w}^*)) = \arg \max_{(\bar{w}, E(\bar{w}))} p(D; \bar{w}, E(\bar{w})). \quad (3)$$

Решение задачи в таком виде нам неизвестно. Мы предлагаем приближенный алгоритм ее решения. Он состоит в итеративном разбиении (3) на две подзадачи: поиска оптимальной раскраски эталонов при фиксированных их размерах (задача (2)) и поиска оптимальных размеров эталонов при фиксированной их раскраске.

Полученный итеративный алгоритм на каждой итерации выполняет следующие три действия:

1. Найти эталоны  $E^*(\bar{w}^0)$  с фиксированными начальными ширинами  $\bar{w}^0$  при помощи алгоритма, решающего задачу (2):

$$E^*(\bar{w}^0) = \arg \max_{E(\bar{w}^0)} p(D; \bar{w}^0, E(\bar{w}^0)). \quad (4)$$

2. Распознать обучающие изображения  $x^m$ ,  $m = \overline{1, M}$  с использованием эталонов  $E^*(\bar{w}^0)$ , построенных на предыдущем шаге; поиск оптимальной сегментации  $\bar{s}^m$  происходит среди тех сегментаций, которые соответствуют текстовой строке  $\bar{c}^m$ :

$$\bar{s}^m = \arg \min_{\bar{s} \in \bar{S}(\bar{c}^m)} f(x^m, \bar{s}, E^0(\bar{w}^0)), m = \overline{1, M}. \quad (5)$$

Эта задача решается при помощи динамического программирования.

3. Улучшить оценку ширин  $\bar{w}^*$  для заданных начального набора ширин и сегментации, полученной на предыдущем шаге:

$$\{\bar{w}^0, x^m, \bar{c}^m, \bar{s}^m \mid m = \overline{1, M}\} \rightarrow \bar{w}^*. \quad (6)$$

Сосредоточимся на третьей задаче, поскольку первые две имеют известное решение. Мы предложим метод решения данной задачи, состоящий в сведении ее к одной из решаемых задач оптимальной разметки [13–16]. Каждой букве алфавита будет соответствовать пара вершин графа, определяющего задачу разметки. Метки вершин будут задавать ширины эталонов букв, а наилучшая разметка — оптимальные в определенном смысле ширины эталонов.

Вначале вкратце неформально опишем постановку задачи (6). Будем считать, что сегментация обучающего изображения  $x$ , полученная на шаге 2, определяет положения букв на нем. Положение буквы будем отождествлять с положением ее центра. Для заданных ширин эталонов букв их положения на обучающем изображении определяют оценку максимального правдоподобия раскраски эталонов. (В нашем случае таковой является результат усреднения всех фрагментов обучающего изображения, соответствующих определенной букве.) Раскрашенные эталоны, в свою очередь, определяют штраф за соответствие построенной таким образом сегментации (определяемой центрами букв и их ширинами) обучающему изображению. Задача состоит в поиске таких ширин эталонов, которые минимизируют этот штраф.

Введем необходимые определения, опишем указанный алгоритм более детально и сформулируем задачу поиска оптимальных эталонов как задачу разметки.

### Множество наборов ширин эталонов

Пусть  $\bar{w}^0 = \{w_a \mid a \in A_0\} \cup \{w_x \equiv 1\}$  — набор начальных ширин эталонов, построенных на предыдущей итерации алгоритма 2. Разобьем поле зрения каждой буквы  $a \in A_0$  на две части: правую и левую, с ширинами, соответственно,  $\omega_a^L = \left\lfloor \frac{w_a}{2} \right\rfloor$  и  $\omega_a^R = w_a - \left\lfloor \frac{w_a}{2} \right\rfloor$ . Изменение ширины эталона буквы будет происходить за счет изменения ширины правой или левой его частей на величину, не превышающую определенное наперед заданное значение  $\Delta$ . Таким образом, ширина левой половины буквы  $a$  выбирается из множества  $\Omega_a^L = \{\omega_a^L \pm i, i = \overline{0, \Delta}\}$ , а правой — из множества  $\Omega_a^R = \{\omega_a^R \pm i, i = \overline{0, \Delta}\}$ . Множество векторов, содержащих ширины правых и левых половин всех букв алфавита, обозначим  $\Omega = \prod_{a \in A_0} \Omega_a^L \times \Omega_a^R$ . Обозначения  $\times$  и  $\prod$  определяют декартово произведение множеств.

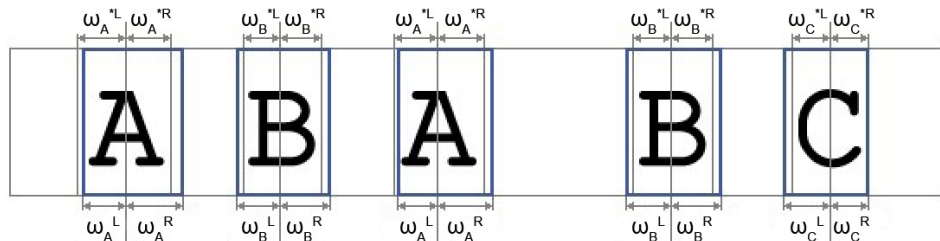


**Рис. 1: Вариация ширины эталона: фиксируется центральный столбец эталона, и изменение ширины левой и правой частей приводит к сдвигу соответственно левой и правой «границ» эталона**

### Сегментация, задаваемая измененными ширинами эталонов

Пусть  $\bar{s}^*$  — исходная сегментация, полученная на шаге 2 алгоритма 2. Пусть ей соответствуют набор ширин  $\bar{\omega}^* \in \Omega$  и текстовая строка  $\bar{c}$ . Пусть  $\bar{\omega}$  — новый набор ширин букв. Сегменты соответствующей ей сегментации  $\bar{s} = \bar{s}(\bar{\omega}, \bar{s}^*)$  строятся таким образом:

- для каждого значащего сегмента  $s' = s^*(c_n), n = \overline{1, N_{\bar{c}}}$  исходной сегментации строится сегмент  $s(c_n) = (i(s') + \omega_{a(s')}^L - \omega_{a(s')}^L, a(s'))$  в новой сегментации;
- ширина полученного сегмента  $s(c_n)$  принимается равной  $\omega_{a(s')}^L + \omega_{a(s')}^R$ ;
- новые ширины  $\bar{\omega}$  считаются недопустимыми, если значащие сегменты новой сегментации перекрываются, т.е.  $i(s(c_n)) + \omega_{a(s(c_n))}^L + \omega_{a(s(c_n))}^R > i(s(c_{n+1}))$ ,  $n = \overline{1, N_{\bar{c}} - 1}$ , или первый или последний значащие сегменты вылазят за границы изображения:  $i(s(c_1)) < 0$ ;  $i(s(c_{N_{\bar{c}}})) + \omega_{a(s(c_{N_{\bar{c}}}))}^L + \omega_{a(s(c_{N_{\bar{c}}}))}^R > W_x$ ;
- оставшиеся не распределенные между сегментами столбцы поля зрения изображения заполняются сегментами-вставками.



**Рис. 2: Построение сегментации  $\bar{s}(\bar{\omega}, \bar{s}^*)$ : от зафиксированных центральных столбцов сегментов букв откладываются новые значения ширин левых и правых их**



## частей, образуя новые сегменты букв, а оставшиеся столбцы поля зрения заполняются вставками

### Оценка расцветки эталонов для данного набора ширин

Пусть  $\bar{\omega}$  — набор ширин эталонов букв,  $\bar{s}$  — соответствующая ему сегментация входного изображения. Для каждой буквы  $a^* \in A_0$  ее эталон  $e(a^*)$  строится путем усреднения соответствующих ей фрагментов  $x(s), a(s) = a^*$  обучающих изображений  $x^1, \dots, x^M$ :

$$e_{a^*}^{r+1}(t) = \frac{1}{Z(a^*)} \sum_{m=1}^M \sum_{\substack{s \in \bar{s}^m: \\ a(s)=a^*}} x^m(t_s + t), \quad t \in T_{a^*},$$

где  $Z(a^*) = \sum_{m=1}^M |\{s \in \bar{s}^m : a(s) = a^*\}|$  — суммарное количество сегментов с именем  $a^*$  в сегментациях всех обучающих изображений. Множество построенных таким образом эталонов букв будем обозначать  $E_0(\bar{\omega}, \bar{s}) = \{e(a), a \in A_0\}$ .

### Задача поиска оптимальных ширин

Таким образом, мы определили, как по исходной сегментации  $\bar{s}^*$ , построенной на шаге 2 алгоритма 2, строится множество  $\Omega$  допустимых наборов ширин эталонов, как по вектору  $\bar{\omega} \in \Omega$  ширин эталонов строится новая сегментация  $\bar{s}(\bar{\omega}, \bar{s}^*)$  и как по этой сегментации оценивается раскраска эталонов букв  $E_0(\bar{\omega}, \bar{s}(\bar{\omega}, \bar{s}^*))$ .

Задача поиска оптимальных ширин состоит в нахождении таких ширин букв  $\bar{\omega}^*$ , которые минимизируют суммарный штраф за сегментацию  $\bar{s}(\bar{\omega}, \bar{s}^*)$  для обучающего изображения  $x$ :

$$\bar{\omega}^* = \arg \min_{\bar{\omega} \in \Omega} f(x, \bar{s}(\bar{\omega}, \bar{s}^*), E_0(\bar{\omega}, \bar{s}(\bar{\omega}, \bar{s}^*))). \quad (7)$$

Прежде чем перейти к задаче (7), отметим, что в процессе ее решения эталон вставки не изменяется: его ширина постоянна и равна единице, а расцветка определяется на шаге 2 алгоритма 2.

## 3 Формулировка задачи поиска оптимальных ширин как задачи разметки

В этом разделе мы сформулируем задачу разметки [13–16], соответствующую задаче поиска оптимальных ширин эталонов (7). Вершинам графа этой задачи поставим в соответствие левые и правые половины букв алфавита, а меткам — возможные значения ширин эталонов этих букв. Каждой разметке будет соответствовать сегментация, построенная на основании выбранных ширин букв. Штрафы на метках и дужках определим так, чтобы штраф за произвольную разметку совпадал со штрафом за соответствующую ей сегментацию. В конце раздела мы

покажем, что полученная задача разметки является субмодулярной, а значит, имеет полиномиальное решение.

Дадим определение задачи разметки. Рассмотрим граф  $(V, E)$ , задаваемый конечным множеством вершин  $V$  и множеством  $E \subseteq V \times V$  ребер, определяющим структуру соседства. Вершины  $(v, v') \in E$ , соединенные ребром, будут называться соседними. Каждому объекту  $v \in V$  соответствует конечное упорядоченное множество меток  $K_v$ . Отображение  $\bar{k} : V \rightarrow \otimes_{v \in V} K_v$ , ставящее в соответствие каждой вершине  $v \in V$  метку  $k \in K_v$ , будем называть разметкой графа  $(V, E)$ . Будем считать заданными отображения  $q_v : K_v \rightarrow \mathbb{R}, v \in V$  и  $g_{vv'} : K_v \times K_{v'} \rightarrow \mathbb{R} \cup \{\infty\}, (v, v') \in E$ , определяющие штраф  $q_v(k)$  за выбор метки  $k$  в вершине  $v$  и штраф  $g_{vv'}(k, k')$  за выбор пары меток  $k$  и  $k'$  в вершинах  $v$  и  $v'$  соответственно. Равенство величины  $g_{vv'}(k, k')$  бесконечности обозначает, что метки  $k$  и  $k'$  не могут быть выбраны одновременно в вершинах  $v$  и  $v'$  соответственно. Задача разметки состоит в нахождении разметки  $\bar{k}^*$ , имеющей минимальный штраф:

$$\bar{k}^* = \arg \min_{\bar{k}} G(\bar{k}) = \arg \min_{\bar{k}} \left[ \sum_{v \in V} q_v(k_v) + \sum_{(v, v') \in E} g_{vv'}(k_v, k_{v'}) \right]. \quad (8)$$

Задача разметки называется субмодулярной, если для произвольных двух пар меток  $k_i, k_j \in K_v, i < j$  и  $k_{i'}, k_{j'} \in K_{v'}, i' < j'$  в соседних вершинах  $(v, v') \in E$  выполняется следующее неравенство

$$g_{vv'}(k_i, k_{i'}) + g_{vv'}(k_j, k_{j'}) \leq g_{vv'}(k_i, k_{j'}) + g_{vv'}(k_j, k_{i'}). \quad (9)$$

Таким образом, суммарный штраф на непересекающихся (будем называть их параллельными) дужках не должен превосходить суммарный штраф на перекрещивающихся дужках (см. рис. 4(b)). Для случая бесконечных штрафов условие (9) принимает следующий вид: если левая часть неравенства равна бесконечности (одна из параллельных дужек имеет бесконечный штраф), то и правая часть неравенства должна быть равна бесконечности (хотя бы одна из перекрещивающихся дужек должна иметь бесконечный штраф).

Субмодулярные  $(\min, +)$  задачи разметки полиномиально разрешимы, так как при изменении значений  $q_v(k)$  и  $g_{vv'}(k, k')$  на равные по величине и противоположные по знаку, они превращаются в супермодулярные задачи максимизации [19, 20], для которых существует эффективный алгоритм [17, 18, 21].

Вернемся к нашей задаче. Множество  $V$  вершин графа содержит в два раза больше элементов, чем содержится букв в алфавите  $A_0$ . Каждой букве  $a \in A_0$  соответствуют две вершины  $a^L \in V$  и  $a^R \in V$ , отвечающие за левую и правую части эталона буквы  $a$ :

$$V = \{a^L, a^R \mid a \in A_0\}.$$

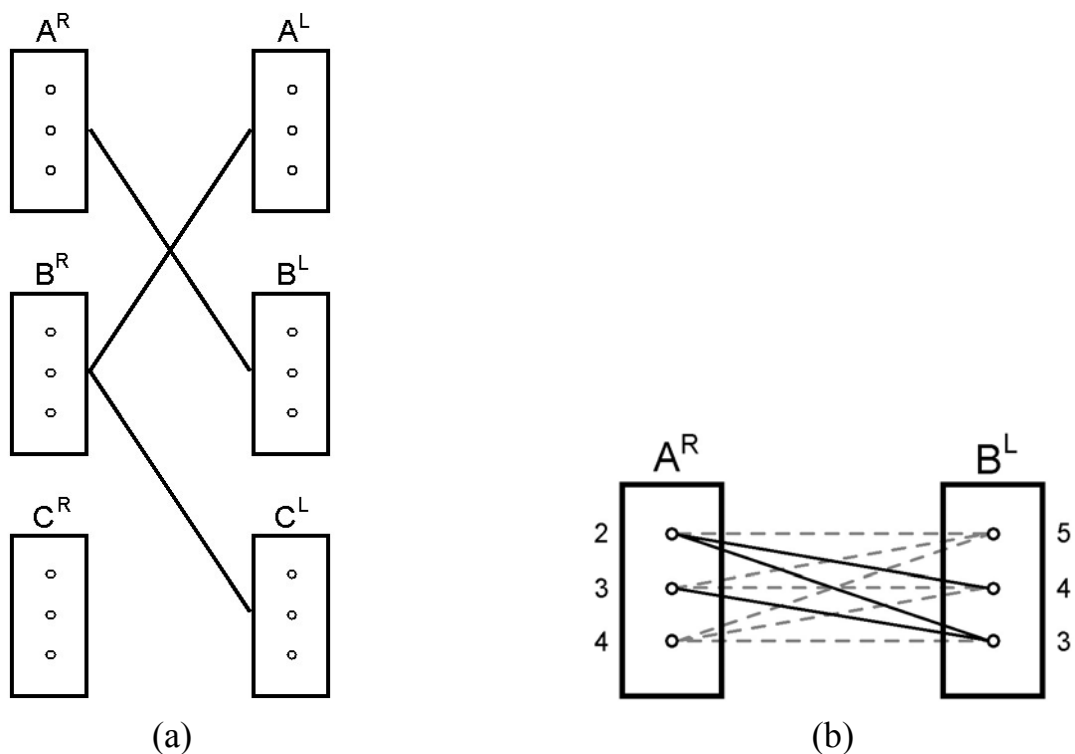
Вершинам  $a^L$  и  $a^R$  приписаны множества меток  $K_a^L$  и  $K_a^R$ , задающие возможные

ширины соответствующих половин эталона буквы  $a$  :

$$\begin{aligned} K_a^L = W_a^L &= \{w_a^L + \Delta, \dots, w_a^L, \dots, w_a^L - \Delta\}, \quad v = a^L, a \in A_0, \\ K_a^R = W_a^R &= \{w_a^R - \Delta, \dots, w_a^R, \dots, w_a^R + \Delta\}, \quad v = a^R, a \in A_0. \end{aligned} \quad (10)$$

Здесь по-прежнему  $\Delta$  — максимальная величина вариации ширины за один шаг. Таким образом, каждой вершине соответствует  $|K_v| = 2\Delta + 1$  упорядоченных меток. Отметим, что для «правых» вершин метки упорядочены по возрастанию ширины (см. (10)), а для «левых» — по убыванию. Это в дальнейшем обеспечит нам субмодулярность, а, значит, и разрешимость задачи.

Определим структуру соседства  $E$  на графе. Правая половина  $a^R$  одной буквы и левая половина  $b^L$  другой являются соседними вершинами тогда, когда в исходной текстовой строке  $\bar{c}$  встречается хотя бы одно буквосочетание "ab". Так как у «правой» вершины могут быть только левые соседи, и наоборот, у «левой» — только правые, то граф является двудольным. На рис. 3(a) приведен пример графа для строки «АВАВС».



**Рис. 3: Структура графа: (a) — пример структуры соседства графа для строки «АВАВС»; (b) — пример структуры дужек между метками двух вершин (черной линией обозначены допустимые дужки, серой пунктирной — не допустимые, т.е. соединяющие недопустимую комбинацию ширин)**

Определим функции штрафов на метках и дужках. Пусть  $\bar{w}$  — набор ширин эталонов букв,  $\bar{s}_{\bar{w}}$  — соответствующая ему сегментация входного изображения.

Пусть также набору  $\bar{w}$  соответствует разметка  $\bar{k}$ . Штраф за набор ширин  $\bar{w}$  равен

$$f(x, \bar{s}_{\bar{w}}, E(\bar{w}, \bar{s}_{\bar{w}})) = \sum_{s \in \bar{s}_{\bar{w}}} \sum_{t \in T(s)} (x(t_s + t) - e_{a(s)}(t))^2. \quad (11)$$

Распределим составляющие этого штрафа по меткам и ребрам так, чтобы штраф за разметку  $\bar{k}$  равнялся штрафу (11). Для этого перепишем его в другом виде, разбив на три составляющих: сумму штрафов за левые половины сегментов букв, сумму штрафов за правые половины сегментов букв и сумму штрафов за сегменты-вставки:

$$\begin{aligned} f(x, \bar{s}_{\bar{w}}, E(\bar{w}, \bar{s}_{\bar{w}})) &= \sum_{s \in \bar{s}_{\bar{w}}} \sum_{t \in T(s)} (x(t_s + t) - e_{a(s)}(t))^2 = \\ &= \sum_{a \in A_0} \sum_{s \in \bar{s}_{\bar{w}}:} \sum_{\substack{t \in T_L(s) \\ a(s)=a}} (x(t_s + t) - e_{a(s)}(t))^2 + \sum_{a \in A_0} \sum_{s \in \bar{s}_{\bar{w}}:} \sum_{\substack{t \in T_R(s) \\ a(s)=a}} (x(t_s + t) - e_{a(s)}(t))^2 + \\ &+ \left( \sum_{a_1 \in A_0} \sum_{a_2 \in A_0} \sum_{l=1}^{N_{\bar{c}}-1} \mathbf{1}_{\{c_l=a_1 \wedge c_{l+1}=a_2\}} \sum_{s \in \bar{s}_{\bar{w}}:} \sum_{\substack{t \in T(s) \\ s(c_l) < s < s(c_{l+1})}} (x(t_s + t) - e_{\kappa}(t))^2 + \right. \\ &\left. + \sum_{s \in \bar{s}_{\bar{w}}:} \sum_{\substack{t \in T(s) \\ s < s(c_l)}} (x(t_s + t) - e_{\kappa}(t))^2 + \sum_{s \in \bar{s}_{\bar{w}}:} \sum_{\substack{t \in T(s) \\ s > s(c_{N_{\bar{c}}})}} (x(t_s + t) - e_{\kappa}(t))^2 \right). \quad (12) \end{aligned}$$

Здесь  $T_L(s)$  и  $T_R(s)$  — прямоугольные фрагменты поля зрения, отвечающие левой и правой половинам сегмента  $s$ ,  $\mathbf{1}_{\{x\}}$  — индикатор истинности выражения  $x$ , а выражение  $s(c_l) < s < s(c_{l+1})$  обозначает перебор по всем сегментам сегментации  $\bar{s}_{\bar{w}}$ , находящимся между сегментами  $n$ -й и  $(n+1)$ -й буквы строки  $\bar{c}$ . Очевидно, это будут сегменты-вставки. Заметим, что штраф за сегменты-вставки (сумма в скобках) в свою очередь состоит из трех частей: штраф за вставки между парами букв, а также штрафы за вставки в начале и конце строки.

Метке  $k = w_a^L$  вершины  $a^L$  поставим в соответствие штраф  $q_{a^L}(k)$  за отличие всех встречающихся в сегментации  $\bar{s}_{\bar{w}}$  левых половин изображения буквы  $a$  ширины  $k$  от построенного эталона:

$$q_{a^L}(k) = \sum_{s \in \bar{s}_{\bar{w}}:} \sum_{\substack{t \in T_L(s) \\ a(s)=a}} (x(t_s + t) - e_{a(s)}(t))^2.$$

Отметим, что  $k$  входит в выражение для штрафа не явно, но посредством сегментации  $\bar{s}_{\bar{w}}$ , в которой левая половина буквы  $a$  имеет ширину  $k$ . Несложно убедиться в том, что данное определение штрафа корректно, т.е. что значение  $q_{a^L}(k)$  будет одинаковым для любой разметки, в которой вершина  $a^L$  имеет ширину  $k$ .

Таким образом, штраф  $q_{a^L}(k)$  является частью суммы (12), соответствующей

левой половине элемента  $a$ . Полностью аналогично вводится штраф  $q_{a^R}(k)$  метки «правой» вершины:

$$q_{a^R}(k) = \sum_{\substack{s \in \bar{s}_{\bar{a}}: \\ a(s)=a}} \sum_{t \in T_R(s)} (x(t_s + t) - e_{a(s)}(t))^2.$$

Дужке между метками  $k_1$  и  $k_2$  вершин  $a^R$  и  $b^L$  поставим в соответствие штраф  $g_{a^R b^L}(k_1, k_2)$  за отличие промежутков между всеми встречающимися в сегментации парами соседствующих букв  $a$  и  $b$  (соответствующие половины которых имеют ширины  $k_1$  и  $k_2$ ) от эталона вставки, построенного на шаге 2 алгоритма 2:

$$g_{a^R b^L}(k_1, k_2) = \sum_{l=1}^{N_{\bar{c}}-1} \mathbf{1}_{\{c_l = a \wedge c_{l+1} = b\}} \sum_{\substack{s \in \bar{s}_{\bar{a}}: \\ s(c_l) < s < s(c_{l+1})}} \sum_{t \in T(s)} (x(t_s + t) - e_{a(s)}(t))^2.$$

Таким образом, штраф  $g_{a^R b^L}(k_1, k_2)$  является частью суммы (12), соответствующей промежуткам между парами букв  $a$  и  $b$ . Если для данной пары ширин существуют перекрывающиеся сегменты, то штраф на соответствующем ребре полагается равным бесконечности.

«Неразделенными» в сумме (12) остались два последних слагаемых, соответствующие сегментам-вставкам в самом начале и самом конце строки:

$$q_S(k_{c_1^L}) = \sum_{\substack{s \in \bar{s}_{\bar{a}}: \\ s < s(c_1)}} \sum_{t \in T(s)} (x(t_s + t) - e_{a(s)}(t))^2;$$

$$q_E(k_{c_{N_{\bar{c}}}^R}) = \sum_{\substack{s \in \bar{s}_{\bar{a}}: \\ s > s(c_{N_{\bar{c}}})}} \sum_{t \in T(s)} (x(t_s + t) - e_{a(s)}(t))^2.$$

Эти штрафы зависят от ширин  $k_{c_1^L}$  и  $k_{c_{N_{\bar{c}}}^R}$  первой и последней буквы строки соответственно, поэтому мы добавим их к штрафам на метках вершин  $c_1^L$  и  $c_{N_{\bar{c}}}^R$ :

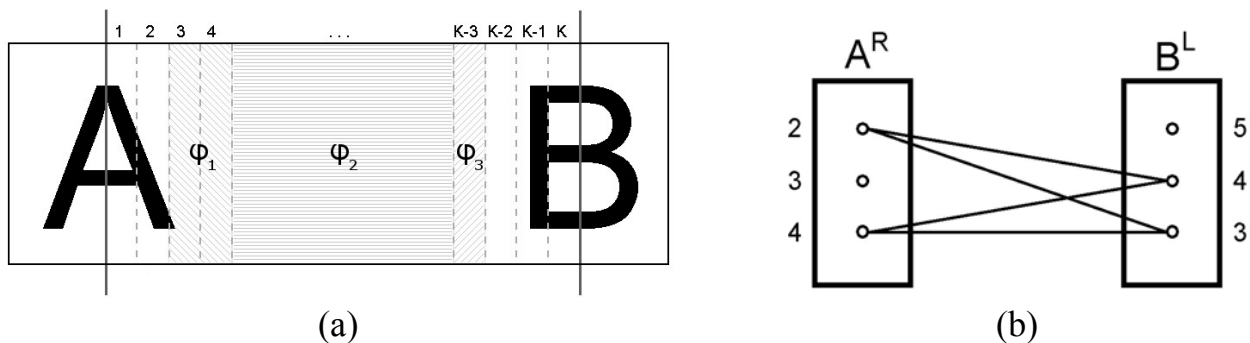
$$q_{c_1^L}(k) := q_{c_1^L}(k) + q_S(k), k \in K_{c_1^L}^L; \quad q_{c_{N_{\bar{c}}}^R}(k) := q_{c_{N_{\bar{c}}}^R}(k) + q_E(k), k \in K_{c_{N_{\bar{c}}}^R}^R.$$

Это, в свою очередь, означает, что в построенной задаче разметки штраф (8) за разметку  $\bar{k}$  совпадает со штрафом (12) за набор ширин  $\bar{a}$ .

Таким образом, задача поиска оптимальных ширины эталонов (7) эквивалентна следующей задаче разметки:

$$\bar{k}^* = \arg \min_{\bar{k}} \left[ \sum_{a^L \in V} q_{a^L}(k_{a^L}) + \sum_{a^R \in V} q_{a^R}(k_{a^R}) + \sum_{(a^R, b^L) \in E} g_{a^R b^L}(k_{a^R}, k_{b^L}) \right], \quad (13)$$

определяемой описанным выше графом  $(V, E)$ , множествами меток  $K_v$  на вершинах и штрафами  $q_{a^L}(k)$ ,  $q_{a^R}(k)$  и  $g_{a^R b^L}(k, k')$ .



**Рис. 4:  $K$  субмодулярности задачи (13): произвольно выбранная четверка дужек между метками (b) и соответствующие им столбцы поля зрения изображения пары букв (a); дужки (2;4) и (4;3) являются параллельными, дужки (2;3) и (4;4) — перекрестными**

Покажем теперь, что полученная задача субмодулярна, т.е. что для произвольных двух вершин и произвольных пар меток на каждой из них выполняется условие (9). Рассмотрим пару соседних вершин  $a^R$  и  $b^L$  (рис. 4(a)). Так как штрафы за вставки между разными парами букв  $ab$ , встречающимися на изображении, входят в общий штраф на дужке аддитивно, достаточно показать субмодулярность только на одной паре букв  $ab$ . Выберем произвольным образом четверку дужек (см. рис. 4(b)) и покажем, что для них выполняется условие субмодулярности (9). Для этого необходимо рассмотреть два отдельных случая: случай, когда все штрафы конечны, и случай, когда по крайней мере одна из дужек имеет бесконечный штраф.

В первом случае необходимо показать выполнение неравенства

$$g(2,4) + g(4,3) \leq g(2,3) + g(4,4), \quad (14)$$

где для краткости введено обозначение  $g(k,k') = g_{a^R b^L}(k,k')$ . Обозначим штраф за отличие столбцов  $\langle 3,4 \rangle$  от эталона вставки через  $\phi_1$ , столбцов  $\langle 5,6,\dots,K-4 \rangle$  — через  $\phi_2$ , столбца  $\langle K-3 \rangle$  — через  $\phi_3$  (см. рис. 4(a)). Тогда:

$$\begin{aligned} g(2,4) &= \phi_1 + \phi_2, \\ g(4,3) &= \phi_2 + \phi_3, \\ g(2,3) &= \phi_1 + \phi_2 + \phi_3, \\ g(4,4) &= \phi_2, \end{aligned}$$

из чего следует, что в (14) достигается равенство для всех четверок дужек.

Пусть теперь один из штрафов равен бесконечности. Напомним, что дужка имеет бесконечный штраф, если сумма ширин, соответствующих этой дужке, недопустима, то есть превышает расстояние между фиксированными центральными столбцами сегментов. Но в таком случае дужка  $g(4,4)$  (в нашем примере) обязательно имеет бесконечный штраф, так как для нее сумма ширин наибольшая. Отметим, что дужка, соответствующая наибольшей сумме ширин, всегда является

перекрестной. Таким образом, если в выбранной четверке дужек есть дужка с бесконечным штрафом, то одна из перекрестных дужек будет иметь бесконечный штраф. Субмодулярность задачи доказана.

Мы свели задачу 7 к субмодулярной  $(\min,+)$  задаче разметки (13), для решения которой может быть использован один из известных эффективных алгоритмов. В частности, субмодулярная  $(\min,+)$  задача может быть сведена к задаче поиска минимального сечения графа с неотрицательными весами на дужках. Последняя задача имеет известное эффективное решение.

## 4 Эксперименты

### 4.1 Как проводились тесты

На вход программы подавалось два множества: обучающее, содержащее изображения строк текста и соответствующие им «правильные» текстовые строки, и тестовое, содержащее изображения из того же источника. На каждом шаге работы программы выполнялся сначала поиск оптимальных ширин при текущих эталонах согласно задаче (13), а затем — раскраску эталонов при заданных ширинах согласно задаче (4). Для решения задачи (13) был применен алгоритм поиска минимального сечения методом проталкивания предпотока [25, 26]. Для поиска оптимальных эталонов при известных ширинах (задача (4)) применялся алгоритм, описанный в статье [22].

Алгоритм самообучения, решающий задачу (4), итеративный, и на практике выяснилось, что достаточно трех итераций, чтобы он сошелся к определенному результату. Все эти алгоритмы работали на основании обучающего множества. В конце шага происходило распознавание тестового множества с текущими эталонами, и результат этого распознавания выводился на экран.

Начальной шириной эталонов всех букв выбиралось значение, кратное высоте изображения. Оператор мог изменять это значение, но в качестве стандартного использовалась величина  $H/4$ . Это связано с тем, что алгоритм самообучения гораздо хуже реагирует на ширины, большие истинных, чем на ширины, меньше истинных. Причина этого кроется в постановке задачи, где не разрешается наложение эталонов.

Величина максимальной вариации ширины  $\Delta$  за один шаг эмпирическим путем была выбрана равной 3. Большие значения этой величины приводят к тому, что на первых итерациях алгоритм может остановиться в локальном минимуме. С другой стороны, слишком малые значения вариации приводят к тому, что алгоритм не в состоянии найти оптимальную ширину для букв, имеющих «периодическую» структуру, таких как латинские «m» и «n». Более подробно мы остановимся на этом ниже.

В [22] можно найти подробное описание особенностей алгоритма самообучения, решающего задачу (2). Мы их вкратце здесь перечислим, и далее не будем на них останавливаться. Алгоритм сходится довольно быстро — как правило, необходимо от 3 до 5 итераций для того, чтобы получить результат, который уже не улучшается, в том смысле, что функция правдоподобия мало изменяется. Тем не менее, полученный результат может оказаться локальным максимумом функции правдоподобия, а не глобальным, что отображается в неправильном построении нескольких эталонов и, соответственно, неправильном распознавании обучающей выборки. В таком случае алгоритм можно вытолкнуть из локального максимума установкой начальных значений («сбрасыванием») для неправильно построенных эталонов. Это приводит к довольно эффективному полуавтоматическому способу обучения — после нескольких итераций оператор оценивает качество построенных эталонов и «сбрасывает» те из них, которые построены неправильно.

К особенностям распознавания текста согласно принятой в работе модели можно отнести перепутывание похожих букв, а также чувствительность к ширине эталонов, большей, чем истинная. Последнее приводит к значительному ухудшению качества распознавания, так как из-за недопустимости наложения эталонов друг на друга большие эталоны «выталкивают» соседние символы из их оптимальных положений.

Ниже мы рассмотрим три примера. Все они были получены путем сканирования текста различного качества. Таким образом, условия тестирования были максимально приближены к реальному применению программы. После сканирования изображение обрабатывалось программой, которая автоматически находила текстовые строки, выравнивала и разрезала на отдельные изображения. Таким образом, для подготовки обучающего множества необходимо лишь набрать текст, соответствующий строкам обучающего множества, и скорректировать результаты работы программы, если это необходимо.

В таблице сведены данные по примерам: объемы обучающей и тестовой выборок, а также ошибка распознавания при их распознавании построенными эталонами.

**Таблица 1: Экспериментальная проверка (ОМ – обучающее множество, ТМ – тестовое множество)**

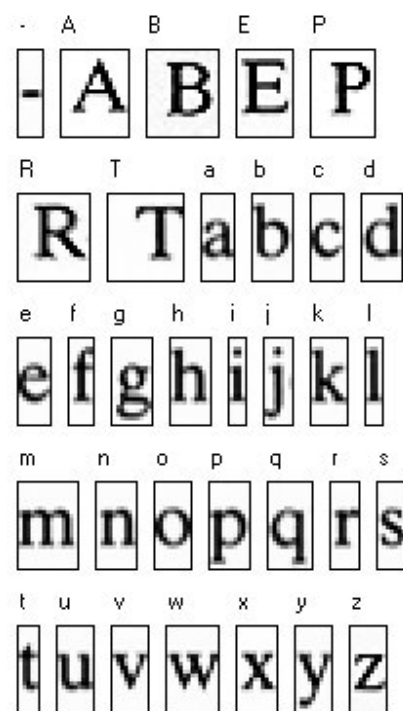
Пример	Объем ОМ	Ошибка, ОМ	Объем ТМ	Ошибка, ТМ
«Skeletonization»	1200	1.6%	360	0.8%
«Physiology»	860	0.2%	635	0.8%
«Big Brother»	1250	4.2%	650	2.5%

Здесь и далее для каждого примера приведены два рисунка. Рис. 5(а) содержит исходные тестовые изображения и результаты их распознавания в текстовом виде и



в виде изображения, составленного из оптимальной последовательности эталонов. Рис. 5(b) содержит построенные эталоны.

circular disks in the picture; disks that fit inside the picture  
 circular disks in the picture disks that fit inside the picture  
 circular disks in the picture disks that fit inside the picture  
 objects but in no other disk within picture ones. The set of  
 objects but in no other disk within picture ones The set of  
 objects but in no other disk within picture ones The set of  
 centers and radii of maximal disks in the picture is termed its  
 centers and radii of maximal disks in the picture is termed its  
 centers and radii of maximal disks in the picture is termed its  
 skeleton, representing a convenient description of the form  
 skeleton representing a convenient description of the form  
 skeleton representing a convenient description of the form  
 and thickness of the elongated objects in the picture. The  
 and thickness of the elongated objects in the picture The  
 and thickness of the elongated objects in the picture The



(a)

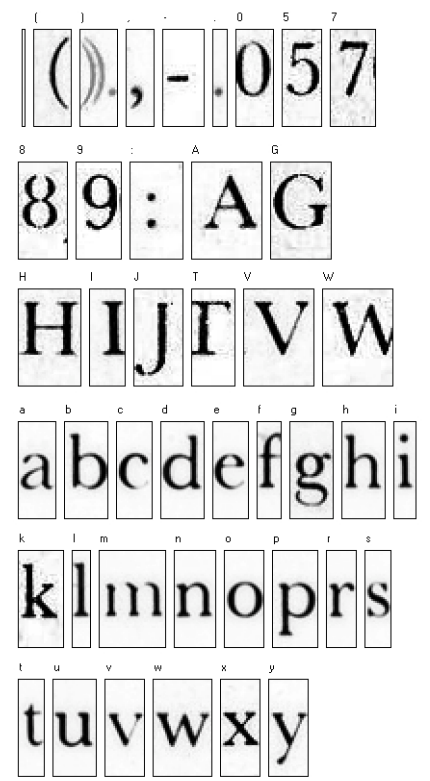
(b)

**Рис. 5: Пример «Skeletonization»**

При рассмотрении полученных эталонов видно, что некоторые эталоны заглавных букв имеют широкий пробел в левой части. Это можно объяснить тем, что в обучающей выборке заглавные буквы встречаются только в начале слова, и варианты эталона с различным количеством вставок перед буквой равнозначны с точки зрения алгоритма. До тех пор, пока в тестовой выборке не встречаются заглавные буквы в середине слов, это не влияет на качество распознавания.

В целом алгоритм демонстрирует уверенное определение размеров эталонов для часто встречающихся букв, но в случае сильнозашумленных текстов у него могут возникать проблемы с эталонами букв, которые встречаются один-два раза. Результаты подобного поведения можно заметить на примере «Physiology» — эталоны букв «Т», «W» и закрывающей скобки построены не очень точно. В таких случаях часто помогает сбрасывание таких эталонов до начальных значений. Так как большинство эталонов к этому моменту уже построены удовлетворительно, степени свободы алгоритма в определении сброшенных эталонов будут ограничены.

In addition to the central excitatory  
 In addition to the central excitatory  
 and inhibitory effects induced by the  
 and inhibitory effects induced by the  
 somatic afferent activity itself, other  
 somatic afferent activity itself, other  
 excitatory descending influences have been  
 excitatory descending influences have been  
 found to modify considerably the  
 found to modify considerably the



(a)

(b)

**Рис. 6: Пример «Physiology»**

Отдельной проблемой алгоритма являются буквы, имеющие «периодическую» структуру, например латинские буквы «m» и «n». Часто алгоритм строит эталон для таких букв, равный только одному элементу периодической структуры. Например, для для букв «m» и «n» — один вертикальный столбец. Иногда в таких случаях простое сбрасывание не помогает, и необходимо вместе со сбрасыванием увеличивать размер эталона. Именно такое поведение наблюдалось в примере «Skeletonization» — после 4–5 итераций все эталоны были построены правильно, кроме букв «m» и «n» (см. рис. 7). После сбрасывания эталонов этих букв с одновременным увеличением их ширины все эталоны были построены правильно.



**Рис. 7: Пример «Skeletonization»: автоматически настроенные эталоны букв «m» и «n»**

Третий пример существенно отличается от предыдущих шумом, который не описывается принятой в работе гауссовой моделью шума. Это проявляется, в первую очередь, в неравномерности яркости фона, что сказалось на результатах распознавания: ошибка составляет здесь около 2,5%. Тем не менее, эталоны, по крайней мере для строчных букв, были построены успешно.

С более детальным описанием приведенных здесь примеров, а также результатами работы на других данных, можно ознакомиться в работе [27].

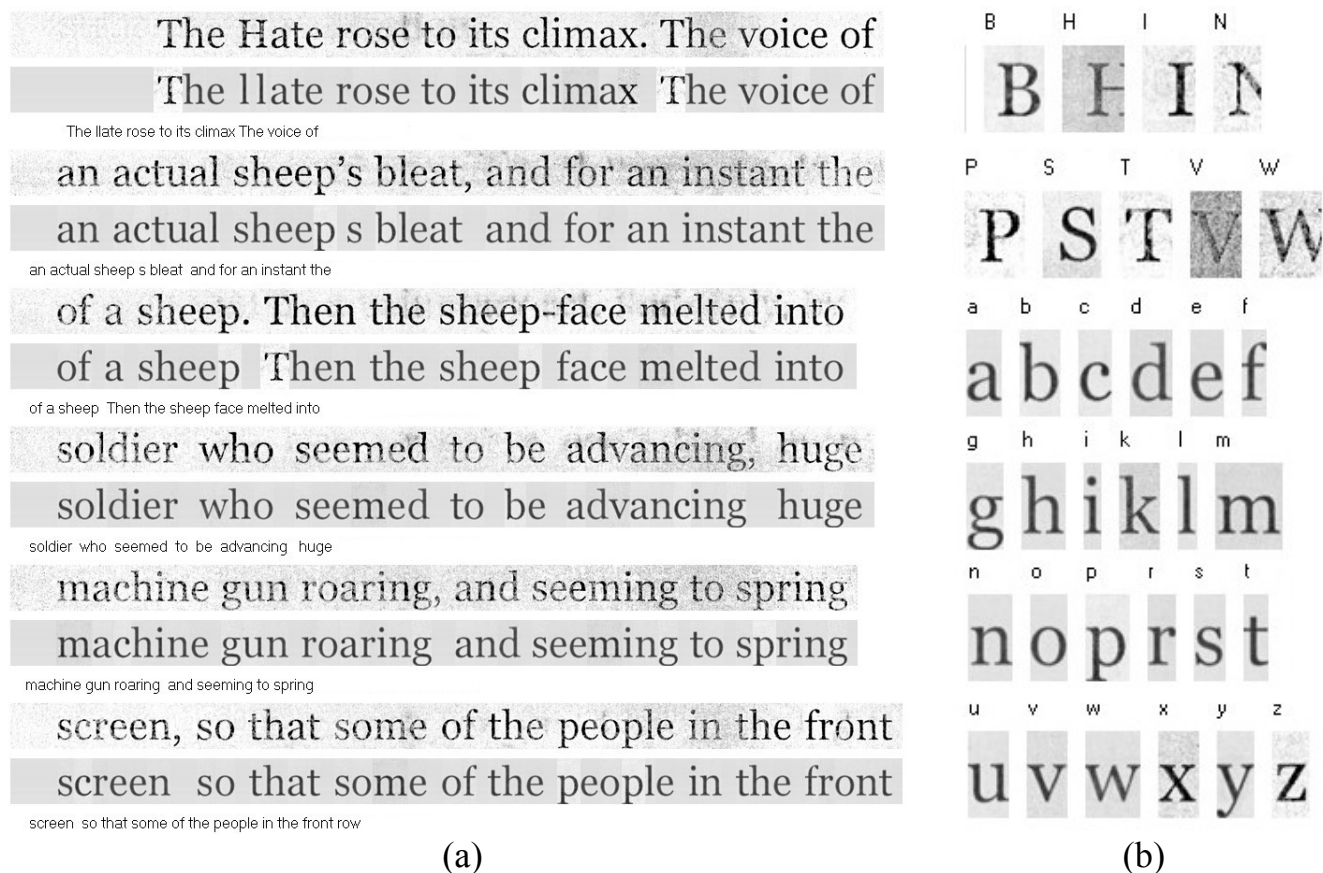


Рис. 8: Пример «Big brother»

## Список литературы

- [1] Ковалевский, В. А. Оптимальный алгоритм распознавания некоторых последовательностей изображений. *Кибернетика*, (4):75-80, 1967.
- [2] K. Aas and L. Eikvil and T. Andersen. Text Recognition from Grey Level Images Using Hidden Markov Models. *Lecture Notes in Computer Science*, 970:503--508, 1995.
- [3] Altun, Y. and Hofmann, T. Large Margin Methods for Label Sequence Learning. *Proc. of 8-th European Conference on Speech Communication and Technology (EuroSpeech--2003), Switzerland, Geneva, New York, 2003*. Springer-Verlag.
- [4] Yihong Xu and George Nagy. Prototype Extraction and Adaptive OCR. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(12):1280--1296, 1999.
- [5] Bogdan Savchynskyy and Olexander Kamotsky. Character templates learning for textual images recognition as an example of learning in structural recognition. *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 88--95, Lyon, France, 2006.

- [6] Kopec, G. E. and Lomelin, M. Document-Specific Character Template Estimation. *IS&T/SPIE 1996 Intl. Symposium on Electronic Imaging: Science & Technology*, San Jose, CA, 1996.
- [7] Kopec, G. E. Multilevel Character Templates for Document Image Decoding. *IS&T/SPIE 1997 Intl. Symposium on Electronic Imaging: Science & Technology*, San Jose, CA, 1997.
- [8] Kopec, G. E. and Lomelin, M. Supervised Template Estimation for Document Image Decoding.
- [9] Prateek Sarkar and Henry S. Baird and Xiaohu Zhang. Training on Severely Degraded Text-Line Images. *IAPR 7th International Conference on Document Analysis and Recognition (ICDAR03)*, pages 38--43, Edinburgh, Scotland, 2003.
- [10] Шлезингер, М. И. Взаимосвязь обучения и самообучения в распознавании образов. *Кибернетика*, (2):81-88, 1968.
- [11] В. А. Ковалевский. *Методы оптимальных решений в распознавании изображений*. Наука, Москва, 1976.
- [12] Шлезингер, М. and Главач, В. *Десять лекций по статистическому и структурному распознаванию*. Наукова думка, Киев, 2004.
- [13] М. И. Шлезингер. Синтаксический анализ двумерных зрительных сигналов в условиях помех. *Кибернетика, К.*, 4:113-130, 1976.
- [14] М. И. Шлезингер and В. К. Коваль. Двумерное программирование в задачах анализа изображений. *Автоматика и телемеханика, М.*, 2:149-168, 1976.
- [15] М.И. Шлезингер. *Математические методы обработки изображений*. Наукова Думка, Киев, 1989.
- [16] Werner, Tomáš. A Linear Programming Approach to Max-sum Problem: A Review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(7):1165--1179, 2007.
- [17] D.M. Greig and B.T. Porteous and A.H. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. *J. Royal Statistical Soc., Series B*, 51(2):271-279, 1989.
- [18] Hiroshi Ishikawa and Davi Geiger. Segmentation by grouping junctions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.
- [19] M. I. Schlesinger and B. Flach. Some solvable subclass of structural recognition problems. In Tomáš Svoboda, editors, *Czech Pattern Recognition Workshop 2000*, pages 55-62, Praha, 2000. Czech Pattern Recognition Society.
- [20] M.I. Schlesinger and B. Flach. Analyzis of optimal labelling problems and their applications to image segmentation and binocular stereovision. In Franz Leberl and Andrej Ferko, editors, *Proceedings East-West-Vision 2002 (EWW'02)*, pages 55-60, 2002. International Workshop and Project Festival on Computer Vision, Computer Graphics, New Media.
- [21] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts. In A. Heyden et al., editors, *ECCV 2002*, number 2352 in LNCS, pages 65-81, Berlin Heidelberg, 2002. Springer-Verlag.
- [22] Б. Д. Савчинский and С. А. Олефиренко. Обучение с неполной

информацией от учителя для распознавания текстовых изображений. *Управляющие системы и машины*, (1):19--29, 2007.

[23] Bokser, M. Omnidocument technologies. *Proceedings of the IEEE*, 80(7):1066-1078, 1992.

[24] Casey, R.G. and Lecolinet, E. A survey of methods and strategies in character segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):690-706, 1996.

[25] Hiroshi Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333--1336, 2003.

[26] Ravindra K. Ahuja and Thomas L. Magnanti and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, 1993.

[27] Олефиренко С.А. Поиск размеров эталонов при обучении алгоритма распознавания. Master's thesis, НТУУ «КПИ», 2008.