Запропонований новий метод стиску бітонального зображення тексту, де в якості основних елементів обробки розглядаються не зв'язні символи зображення тексту, а вертикальні елементи рядка. Представлена імовірнісна модель і алгоритм статистичного аналізу і класифікації вертикальних елементів рядка. Використання представленого методу дозволяє отримати перевагу в ступені стиску порівняно з алгоритмом JB2 формат DjVu близько 37 % при найбільш використовуваній роздільній здатності у 300 dpi

Ключові слова: стиск зображення тексту, вертикальні елементи рядка, статистичний аналіз, класифікація

Предложен новый метод сжатия битонального изображения текста, где в качестве основных элементов обработки рассматриваются не связные символы изображения текста, а вертикальные элементы строки. Представлена вероятностная модель и алгоритм статистического анализа и классификации вертикальных элементов строки. Применение представленного метода позволяет получить преимущество в степени сжатия в сравнении с алгоритмом JB2 формат DjVu около 37 % при наиболее используемом разрешении в 300 dpi

Ключевые слова: сжатие изображения текста, вертикальные элементы строки, статистический анализ, классификация

УДК 621.391:519.728

СЖАТИЕ ИЗОБРАЖЕНИЯ ТЕКСТА НА ОСНОВЕ СТАТИСТИЧЕСКОГО АНАЛИЗА И КЛАССИФИКАЦИИ ВЕРТИКАЛЬНЫХ ЭЛЕМЕНТОВ СТРОКИ

В. Г. Иванов
Доктор технических наук,
профессор, заведующий кафедрой*
E-mail: inform@nulau.edu.ua
Ю. В. Ломоносов
Кандидат технических наук, доцент*
М. Г. Любарский
Доктор физико-математических наук, профессор*
E-meil: inform@nulau.edu.ua
*Кафедра информатики и вычислительной техники
Национальный юридический университет
им. Ярослава Мудрого
ул. Пушкинская, 77, г. Харьков, Украина, 61024

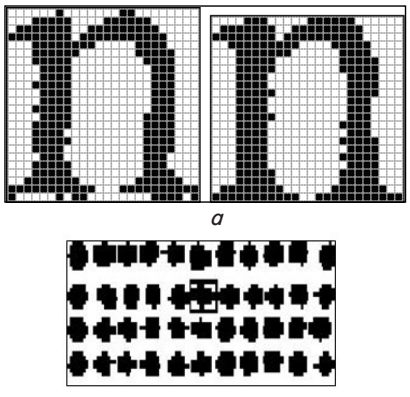
1. Введение

Методы сжатия, основанные на различных ортогональных преобразованиях, дают хороший результат при сжатии размытых изображений, но не эффективны для битональных изображений, тем более изображений текста, изобилующего множеством мелкими деталей – буквами, цифрами, знаками препинания. В настоящее время лучшие алгоритмы для сжатия битональных изображений текста основаны на выделении изображений символов и их классификации. Это – алгоритмы JB2 и JBIG2, используемые соответственно в широко распространённых форматах DjVu и PDF [1–4]. Степень сжатия информации с помощью методов классификации тем выше, чем меньше классов образуется при классификации и чем больше элементов в каждом классе [5, 6]. В идеале при сжатии изображения страницы текста изображения каждого символа должны находиться в одном и только одном классе. Однако ни один из известных алгоритмов этому условию не удовлетворяет. Дело в шумах (случайных искажениях), возникающих при печати страницы и ее последующем сканировании. На рис. 1, а, представлены два случайно выбранные изображения буквы «n» из различных 257, входящих в изображение страницы текста формата А4, при разрешении сканирования 300 dpi. Легко верится, и это действительно так, что на странице не найдется ни одной пары символов «n», полностью совпадающих друг с другом. То же относится и к другим символам, даже точкам, рис. 1, *б*.

И хотя человек без труда может правильно разбить изображения символов на классы, формализовать его действия пока не удалось. Имеющиеся алгоритмы классификации отводят несколько классов для изображений одного и того же символа, что уменьшает степень сжатия изображения. Кроме того в один класс иногда попадают изображения разных символов. Так, например, алгоритм JB2 иногда «путает» буквы «b» и «h».

Указанные недостатки алгоритмов, классифицирующих изображения символов, наводят на мысль о том, что хотя выбор изображений симво-

лов в качестве элементов изображения страницы является естественным, этот выбор не является оптимальным.



а

б

Рис. 1. Влияние шумов на изображения символов: *а* — искажения символа «n»; *б* — искажения символа «точка»

В настоящей работе в качестве классификации элементов изображения страницы рассматриваются вертикальные элементы ее строк. Результаты этой работы отображают новый подход авторов к сжатию графических текстовых данных на основе статистических методов анализа и классификации совокупности вертикальных элементов строки изображения.

2. Анализ литературных данных и постановка проблемы

Один из основных принципов сжатия информации методом классификации состоит в следующем. Пусть информацию можно разбить каким-то образом на элементы. В случае изображения текста естественными элементами являются изображения символов – букв, цифр, знаков препинания. Если эти элементы информации объединить в классы так, чтобы в каждом классе находились тождественные (или почти тождественные) элементы, то нет нужды хранить все элементы информации – достаточно хранить только по одному представителю каждого класса. Совокупность этих представителей называется *словарем*. Кроме того для восстановления информации нужно еще составить таблицу, называемую *картой размещения классов*, которая для каждого класса указывает, где в исходной информации находятся его элементы.

Решением актуальных задач обработки и классификации изображений занимались многие известные отечественные и зарубежные ученые. В работах [7, 8] представлены основные математические средства обработки и классификации изображений. При обработке битональных изображений основным элементом классификации рассматриваются связные символы (алгоритмы JB2 (DjVu) и JBIG2 (PDF)). В работе [9] рассматривается использование методов автоматической классификации изображений, но не рассмотрен критерий выбора оптимального количества классов классификации. Работа [10] представляет современный математический аппарат многомасштабного анализа, который широко используется в графических форматах JPEG2000 и DjVu для сжатия данных. Во многих модификациях этих алгоритмов применяются методы классификации изображений в плоскости вейвлет-коэффициентов, которые позволяют несколько повысить степень сжатия [11]. Много внимания уделяется вопросам классификации и сегментации смешанных изображений [12–14]. Данные методы позволяют понизить влияние шумов на битовую плоскость разделения, что повышает качество выходного изображения. Некоторые алгоритмы нацелены на сокращение временных затрат обработки, но при этом используются известные алгоритмы классификации и сжатия изображений [15]. Отдельный интерес вызывают работы реализующие полномасштабный поиск в сжатых изображениях печатных данных [16], однако размер выходных данных предлагаемых алгоритмов превышает 12 кБ для изображения одной страницы текста с разрешением 300 dpi, что значительно уступает показателям сжатия алгоритма JB2 формата DjVu.

Известен еще ряд алгоритмов [17, 18], основанных на том же принципе. Последние имеют несколько лучшие показатели сжатия, но пока не нашли своего практического применения.

Новый подход к сжатию графических текстовых данных заключается в следующем. Если представить себе прямоугольник, охватывающий какую-либо строку, то *вертикальным элементом* этой строки будем называть пересечение прямоугольника с любой вертикальной линией шириной в один пиксель. На рис. 2 показано разбиение изображения буквы «е» на вертикальные элементы строки.

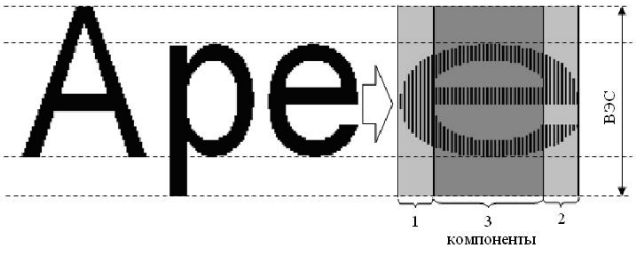


Рис. 2. Изображение буквы «е» и составляющие его вертикальные элементы строки с различным числом компонент

Таким образом, страницу текста можно рассматривать как упорядоченную совокупность вертикальных элементов. Такое разбиение удобно тем, что все вертикальные элементы имеют один и тот же размер и их можно представлять и как двоичные числа, и как векторы с координатами 0 (черный пиксель) и 1 (белый пиксель).

Шумы печати и сканирования случайным образом искажают вертикальные элементы. Так что среди них могут быть искаженные и неискаженные элементы. Однако бессмысленно разбивать совокупность вертикальных элементов, составляющих изображение страницы, на классы тождественных или почти тождественных элементов, поскольку многие из них могут быть искажениями сразу нескольких неискаженных элементов. Более того, встречаются пары неискаженных элементов, которые совпадают с искажениями друг друга.

Имеет смысл говорить только о нечеткой классификации вертикальных элементов, то есть о вероятности того, что данный элемент есть искажение того или иного неискаженного элемента. При этом вопрос о том, является ли какой-то элемент неискаженным, тоже имеет лишь вероятностный ответ.

Таким образом, основная задача статистического анализа совокупности вертикальных элементов, представляющих текстовую страницу, ставится так: *по имеющейся на странице совокупности* $\tilde{X}$ *вертикальных элементов указать минимальную наиболее правдоподобную совокупность* $C \subset \tilde{X}$ *неискаженных элементов, а также для каждой пары* $x \in \tilde{X}$ *и* $c \in C$ *найти вероятность того, что данный элемент* x *является искажением элемента* c.

Термины «минимальную» и «наиболее правдоподобную» будут уточнены позже.

После нахождения этих вероятностей легко получить правильную классификацию изображений символов, представив последние как упорядоченный набор вертикальных элементов. Грубо говоря, изображения двух символов можно отнести к одному классу, если у каждой пары вертикальных элементов, составляющих эти изображения и имеющих один и тот же порядковый номер, достаточно велика вероятность того, что они являются искажениями одного и того же вертикального элемента.

3. Вероятностная модель, статистический анализ и классификация вертикальных элементов строки

Рассматривая рис. 3, легко заметить, что шумы печати и сканирования носят контурный характер, то есть искажения возникают только на границе изображений символов и имеют глубину внутрь или наружу в один пиксель. Искажение, которому может подвергнуться вертикальный элемент $c \in C$ в результате наложения шумов печати и сканирования, состоит в том, что каждая его черная компонента связности на верхнем и нижнем концах может удлиниться или укоротиться на один пиксель (рис. 3).

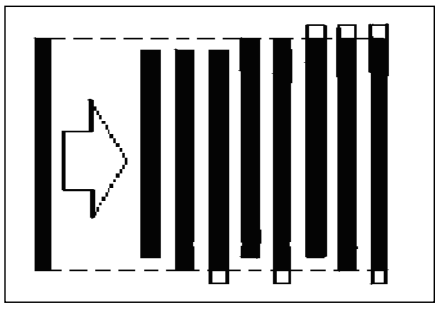


Рис. 3. Возможные искажения черной компоненты вертикального элемента строки

Шумы печати и сканирования носят случайный характер, который опишем следующей упрощенной вероятностной моделью:

Вероятности искажения конца черной компоненты, состоящие в приобретении или потере одного пикселя, одинаковы и равны q. (Число $q \le 1/2$ не известно, поскольку оно зависит и от качества печати, и от разрешения сканирования, и других обстоятельств, но экспериментально легко обнаружить, что оно достаточно мало: $q \approx 0.05$).

1. Искажения концов одной компоненты или концов разных компонент являются независимыми событиями.
2. Вероятность изменения числа компонент из-за искажений равна нулю. (То есть предполагается, что расстояние между черными компонентами не менее 3-х пикселей, что справедливо для подавляющего большинства вертикальных элементов.)

Строго говоря, все три принятые аксиомы неверны. Можно составить более адекватную вероятностную модель, учитывающую механизмы возникновения шумов печати и сканирования. Однако, как оказалась, это не нужно – приведенная простейшая модель достаточно хорошо описывает случайный характер этих шумов.

Из пункта 3 модели следует, что совокупность $\tilde{X}$ всех вертикальных элементов можно разделить на группы $\tilde{X}_s$, $s = 1,2,3,...$, содержащие только элементы с S компонентами, и проводить статистический анализ отдельно для каждой группы. Далее для краткости письма индекс s опускается.

Если элемент x является искажением элемента c, то они могут отличаться друг от друга на $r = 0,1,2,...,2s$ точек. Рассматривая вертикальные элементы строки как векторы евклидова пространства с координатами 0 и 1, число различий r можно вычислить по формуле

$$r = \|x - c\|^2 .$$

Пусть для некоторого неискаженного элемента $c \in C$ множество F_c состоит из него самого и всех возможных его искажений. Будем называть это множество семейством искажений вертикального элемента c. И пусть $P(x|\xi_c)$, – вероятность получить вертикальный элемент $x \in \tilde{X}$ при условии, что он является искажением вертикального элемента $c \in C$. Рассмотрим случайный вектор ξ_c, принимающий значения из семейства F_c с вероятностями $P(x|\xi_c)$.

Из п. 1 и п. 2 вероятностной модели следует, что условная вероятность $P(x|\xi_c)$, зависит только от числа r различий между x и c:

$$P(x|\xi_c) = q^r (1-2q)^{2s-r}, \quad r = \|x - c\|^2 . \qquad (1)$$

Число различных элементов x из семейства искажений F_c элемента $c \in C$, имеющих с ним ровно r различий, равно:

$$m(r) = 2^r C_{2s}^r , \quad r = 0,1,...,2s , \qquad (2)$$

где C_{2s}^r $\dfrac{(2s)!}{r!(2s-r)}$ – биномиальный коэффициент.

Дополним теперь п. п. 1–3 вероятностной модели, упрощенно описывающей случайный характер шумов печати и сканирования, следующим положением, позволяющим провести статистический анализ совокупности $\tilde{X}$ вертикальных элементов, составляющих изображение текстовой страницы:

4. Наблюдаемую совокупность $\tilde{X}$ вертикальных элементов, присутствующих на странице, будем рассматривать как совокупную выборку значений случайных векторов из множества $\{\xi_c : c \in C\}$, получаемую следующим образом. Из множества $\{\xi_c : c \in C\}$ с неизвестной нам вероятностью $P(\xi_c)$ выбирается случайный вектор ξ_c и берется его значение. Этот эксперимент повторяется столько раз, сколько элементов расположено на странице.

3. 1. Нахождение апостериорных вероятностей $P(\xi_c | x)$ при известной совокупности неискаженных вертикальных элементов C

Далее используются следующие обозначения:

X – совокупность *отличающихся друг от друга* вертикальных элементов строки, встречающихся на странице. (Иначе говоря, X – фактор-множество множества $\tilde{X}$ по отношению тождества.) Предполагается, что $C \subset X$, и $X \subset \bigcup_{c \in C} F_c$.

$n(x)$ – количество экземпляров вертикального элемента $x \in X$, имеющихся на странице. В частности, $n(c)$ – количество вертикальных элементов страницы, совпадающих с неискаженным вертикальным элементом $c \in C$.

$N = \sum_{x \in X} n(x)$ – число всех вертикальных элементов, расположенных на странице.

$\nu(x) = \frac{n(x)}{N}$ – частота появления элемента x на странице.

N_C – количества элементов в множестве C.

Вероятность $P(\xi_c)$, о которой идет речь в п. 4 вероятностной модели, – это априорная вероятность того, что очередной вертикальный элемент получен как значение случайного вектора ξ_c. Напомним, что основная задача состоит в нахождении для каждой пары вертикальных элементов $x \in X$ и $c \in C$ вероятности того, что данный элемент x появился в результате искажения вертикального элемента c. Иначе говоря, эта вероятность, обозначим ее через $P(\xi_c | x)$, является апостериорной вероятностью появления случайного вектора ξ_c при условии, что полученным значением является x.

Поскольку $\sum_{c \in C} P(\xi_c | x) = 1$ для всех вертикальных элементов $x \in X$, то вероятность $P(\xi_c | x)$, рассматриваемая как функция двух переменных на декартовом произведении $C \times X$, представляет собой нечеткую классификацию совокупности X вертикальных элементов страницы.

Апостериорная вероятность $P(\xi_c | x)$ связана с априорной вероятностью $P(c)$, формулой Байеса:

$$P(\xi_c | x) = \frac{P(x | \xi_c) P(\xi_c)}{P(x)}, \quad c \in C, x \in X, \tag{3}$$

где

$$P(x) = \sum_{c' \in C \cap F_x} P(x | \xi_{c'}) P(\xi_{c'}). \tag{4}$$

Здесь $P(x)$ полная вероятность появления вертикального элемента x при однократном проведении эксперимента, описанного в п. 4 вероятностной модели, а $P(x | \xi_c)$ – известная (1) с точностью до параметра q вероятность появления вертикального элемента x при условии, что он является значением случайного вектора ξ_c.

Формула Байеса позволяет найти апостериорные вероятности $P(\xi_c | x)$, если известны вероятность искажения q и априорные вероятности $P(\xi_c)$. Рассмотрим зависящий от этих вероятностей функционал

$$\Phi = \frac{1}{2} \sum_{x \in X} \left[P(x) - \nu(x) \right]^2, \tag{5}$$

и будем считать, что чем он меньше, тем правдоподобнее выбранные значения искомых вероятностей $P(\xi_c), c \in C$, и q. Поэтому будем искать эти вероятности как доставляющие минимум функционалу Φ при дополнительном условии $\sum_{c \in C} P(\xi_c) = 1$, которое позволит трактовать найденные значения как априорные вероятности.

Итак, дополнительно используя соотношения (1) и (4), получим следующую задачу на условный экстремум:

$$\Phi = \frac{1}{2} \sum_{x \in X} \left(\sum_{c \in C \cap F_x} q^{\|x-c\|^2} (1-2q)^{2s-\|x-c\|^2} P(\xi_c) - \nu(x) \right)^2 \to \min, \tag{6}$$

$$\sum_{c \in C} P(\xi_c) - 1 = 0. \tag{7}$$

Следующие уравнения получены методом множителей Лагранжа для нахождения условного экстремума (например, [19]).

$$\frac{\partial \Phi}{\partial P(\xi_c)} + \frac{h}{N_C} \equiv \sum_{x \in F_c} \left(\sum_{c' \in C \cap F_x} q^{\|x-c'\|^2} (1-2q)^{2s-\|x-c'\|^2} P(\xi_{c'}) - \nu(x) \right) \times$$
$$\times q^{\|x-c\|^2} (1-2q)^{2s-\|x-c\|^2} + \frac{h}{N_C} = 0,$$
$$c \in C, \tag{8}$$

$$\frac{\partial \Phi}{\partial q} \equiv \sum_{x \in X} \left(\sum_{c \in C \cap F_x} q^{\|x-c\|^2} (1-2q)^{2s-\|x-c\|^2} P(\xi_c) - \nu(x) \right) \times$$
$$\times \left(\sum_{c \in C \cap F_x} q^{\|x-c\|^2-1} (1-2q)^{2s-\|x-c\|^2-1} \left(\|x-c\|^2 - 4sq \right) P(\xi_c) \right) = 0. \tag{9}$$

Здесь h – множитель Лагранжа, нормированный для удобства на N_C – число уравнений в системе (8). Соответствующее множителю Лагранжа уравнение $\frac{\partial \Phi}{\partial h} = 0$ представляет собой уравнение (7). Система уравнений (7)–(9) является необходимым условием экстремума задачи (6), (7). В этом пункте будут найдено ее приближенное аналитическое решение, а также приведен алгоритм численного решения методом последовательных приближений.

Рассмотрим параметр $\mathbf{v} = (\nu(x) : x \notin C)$, представляющий собой вектор, составленный из частот всех искаженных вертикальных элементов. Если $\mathbf{v} = \mathbf{0}$, то

есть искаженные элементы отсутствуют, то задача (6), (7) имеет тривиальное решение. А именно, вероятность искажений q равна 0, и $P(\xi_c)=\nu(c)$ для всех неискаженных элементов $c\in C$. Соответственно система уравнений (7)–(9) имеет в качестве решения точку $\mathbf{u}_0$ с координатами $P(\xi_c)=\nu(c)$, $c\in C$, $q=0$ и $h=0$.

Как было отмечено в п. 1 вероятностной модели, шумы печати и сканирования характеризуются малыми значениями вероятности искажений q и, следовательно, малыми значениями параметра $\mathbf{v}$. Можно ожидать, что в этом случае у системы (7)–(9) существует решение $\mathbf{u}$, близкое к решению $\mathbf{u}_0$. Строго это вытекает из теоремы о неявной функции (например, [20]), если линеаризованная в точке $\mathbf{u}_0$ система (7)–(9) имеет невырожденную матрицу.

Несложные вычисления позволяют получить эту линеаризованную систему:

$$\sum_{c\in C}\left[P(\xi_c)-\nu(c)\right]=\sum_{x\notin C}\nu(x)$$

$$\left[P(\xi_c)-\nu(c)\right]+q\left[-4s\nu(c)+\sum_{c'\in C,\,\|c'-c\|=1}\nu(c')\right]+\frac{h}{N_C}=0$$

$$\sum_{c\in C}\left\{\left[P(\xi_c)-\nu(c)\right]+q\left[-4s\nu(c)+\sum_{c'\in C,\,\|c'-c\|=1}\nu(c')\right]\right\}\times$$

$$\times\left[-4s\nu(c)+\sum_{c'\in C,\,\|c'-c\|=1}\nu(c')\right]+q\sum_{x\notin C}\left[\sum_{c'\in C,\,\|c'-x\|=1}\nu(c')\right]^2=$$

$$=\sum_{x\notin C}\left[\nu(x)\sum_{c'\in C,\,\|c'-x\|=1}\nu(c')\right]. \qquad (10)$$

Чтобы упростить систему уравнений (10) и облегчить вычисление ее определителя, приведем матрицу системы к виду, близкому к верхнетреугольному. Для этого второе из уравнений (10) запишем первым и исключим с его помощью переменные $\left[P(\xi_c)-\nu(c)\right]$, $c\in C$, из двух оставшихся уравнений. Для краткости письма предварительно введем обозначения:

$$\alpha(c)=4s\nu(c)-\sum_{c'\in C,\,\|c'-c\|=1}\nu(c'),\ c\in C;$$

$$\beta=\sum_{c\in C}\alpha(c)=4s-\sum_{c\in C}\sum_{c'\in C,\,\|c'-c\|=1}\nu(c');$$

$$\gamma=N_C\sum_{x\notin C}\left[\sum_{c'\in C,\,\|c'-x\|=1}\nu(c)\right]^2. \qquad (11)$$

Система линейных уравнений (10) после указанных преобразований принимает следующий вид:

$$\left[P(\xi_c)-\nu(c)\right]-\alpha(c)q+\frac{h}{N_C}=0$$

$$\beta q-h=\sum_{x\notin C}\nu(x)$$

$$\gamma q+\beta h=N_C\sum_{x\notin C}\left[\nu(x)\sum_{c'\in C,\,\|c'-x\|=1}\nu(c')\right]. \qquad (12)$$

Матрицей полученной системы уравнений (12) является матрица

$$T=\begin{pmatrix}T_{11} & T_{12}\\ T_{21} & T_{22}\end{pmatrix},$$

где T_{11} – единичная матрица размерности $N_C\times N_C$, T_{12} – матрица размерности $N_C\times 2$, каждая строка которой имеет вид: $\left(-\alpha(c)\ \ \frac{1}{N_C}\right)$, $c\in C$, T_{21} – нулевая матрица размерности $2\times N_C$, и $T_{22}=\begin{pmatrix}\beta & -1\\ \gamma & \beta\end{pmatrix}$.

Теперь легко найти определитель этой матрицы:

$$\det T=\det T_{22}=\beta^2+\gamma. \qquad (13)$$

Хотя сразу видно, что благодаря второму слагаемому γ определитель не равен нулю, однако для достоверности приближенных решений, которые будут получены ниже, и сходимости итераций при численном решении исходной нелинейной системы требуется, чтобы определитель не был малой величиной.

Рассмотрим функцию $k(c)=\sum_{\|c'-c\|=1}1$, равную количеству элементов из C, находящихся на расстоянии 1 от элемента $c\in C$. Эта функция характеризует плотность расположения неискаженных элементов. Поскольку по формуле (2) количество возможных вертикальных элементов, находящихся на расстоянии 1 от элемента c, в точности равно 4s, то $k(c)\le 4s$. Поэтому

$$\sum_{c\in C}\sum_{c'\in C,\,\|c'-c\|=1}\nu(c')=\sum_{c\in C}k(c)\nu(c)\le 4s\sum_{c\in C}\nu(c)=4s.$$

Таким образом, если элементы из C расположены настолько плотно, что почти каждый элемент из C полностью окружен элементами из этого же множества, то почти для всех неискаженных элементов $k(c)=4s$. Поэтому коэффициент β хотя и строго положителен, но может быть малым. Однако, на самом деле, множество неискаженных вертикальных элементов достаточно разрежено – лишь незначительная доля элементов из C имеет хотя бы одного соседа на расстоянии 1. Это объясняется тем, что кириллические, латинские и другие символы состоят из линий. На рис. 4, а показаны изображения отрезков прямых под разными углами наклона. Легко заметить, что наиболее близкие вертикальные элементы имеют два отличия, то есть находятся на расстоянии $\sqrt{2}$. Однако встречаются пары вертикальных элементов, находящиеся и на расстоянии 1. В основном они, как это видно на рис. 4, *б*, встречаются на изгибах линий, что особенно характерно для шрифтов з засечками.

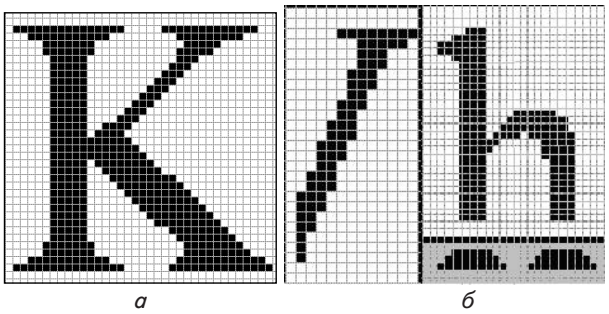
Таким образом, для большинства неискаженных элементов $c\in C$ выполнено $k(c)=0$, и поэтому $\beta^2\approx(4s)^2$.

Оценим теперь второе слагаемое γ, снова используя тот факт, что для большинства элементов $c\in C$ выполняется $k(c)=0$:

$$\gamma=N_C\sum_{x\notin C}\left[\sum_{c'\in C,\,\|c'-x\|=1}\nu(c')\right]^2\ge$$

$$\ge N_C\sum_{x\notin C}\sum_{c'\in C,\,\|c'-x\|=1}\nu^2(c')=N_C\sum_{c\in C}\left[4s-k(c)\right]\nu^2(c)\approx$$

$$\approx 4sN_C\sum_{c\in C}\nu^2(c).$$



а *б*

Рис. 4. Вертикальные элементы строки, образующие:
а — отрезки прямых под разными углами наклона;
б — криволинейные элементы символов

Из известного неравенства между средним квадратичным и средним арифметическим:

$$\sqrt{\frac{\sum_{c\in C}\nu^2(c)}{N_C}} \ge \frac{\sum_{c\in C}\nu(c)}{N_C} = \frac{1}{N_C},$$

вытекает неравенство

$$N_C\sum_{c\in C}\nu^2(c) \ge 1.$$

Таким образом, второе слагаемое определителя можно оценить величиной $\gamma \approx 4s$, а сам определитель – величиной $\det T \approx 4s(4s+1)$.

Вычислим далее матрицу T^{-1}, обратную матрице T. Легко проверяется, что

$$T^{-1} = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix}, \qquad (14)$$

где A_{11} – единичная матрица размерности $N_C \times N_C$, A_{21} – матрица размерности $N_C \times 2$, каждая строка которой имеет вид:

$$\frac{1}{\det T}\left(\alpha(c)\beta + \frac{\gamma}{N_C} \qquad -\frac{\beta}{N_C} + \alpha(c)\right),\ c\in C,$$

A_{12} – нулевая матрица размерности $2\times N_C$, и $A_{22} = \frac{1}{\det T}\begin{pmatrix} \beta & 1 \\ -\gamma & \beta \end{pmatrix}$.

Применяя матрицу T^{-1} к правой части системы линейных уравнений (12), получим приближенное аналитическое решение задачи (6), (7):

$$P(\xi_c) = \nu(c) + \frac{1}{\det T}\times$$
$$\times\left\{\left[\alpha(c)\beta + \frac{\gamma}{N_C}\right]\sum_{x\notin C}\nu(x) + \left[-\frac{\beta}{N_C} + \alpha(c)\right]N_C\sum_{x\notin C}\left[\nu(x)\sum_{\|c-x\|=1}\nu(c)\right]\right\},$$
$c\in C$,

$$q = \frac{1}{\det T}\left\{\beta\sum_{x\notin C}\nu(x) + N_C\sum_{x\notin C}\left[\nu(x)\sum_{\|c-x\|=1}\nu(c)\right]\right\}.$$

Численное решение системы нелинейных уравнений (7)–(9) проводится стандартным методом последовательных приближений, который используется, например, в [20] для доказательства теоремы о неявной функции.

Предварительно над уравнениями исходной системы (7)–(9) нужно провести те же действия, какие были проведены при переходе от линеаризованной системы уравнений (10) к линейной системе (12). Полученная таким образом система уравнений (15) эквивалентна системе (7)–(9), а линейная система (12) является ее линеаризацией.

$$W_{P(\xi_c)} \equiv \frac{\partial\Phi}{\partial P(\xi_c)} + \frac{h}{N_C} = 0,\ c\in C$$
$$W_q \equiv \sum_{c\in C}P(\xi_c) - 1 - \sum_{c\in C}\left[\frac{\partial\Phi}{\partial P(\xi_c)} - \frac{h}{N_C}\right] = 0$$
$$W_h \equiv N_C\left\{\frac{\partial\Phi}{\partial q} + \sum_{c\in C}\alpha(c)\left[\frac{\partial\Phi}{\partial P(\xi_c)} + \frac{h}{N_C}\right]\right\} = 0, \qquad (15)$$

где частные производные функции Φ заданы соотношениями (8) и (9).

Пусть, как и ранее, $\mathbf{v} = (\nu(x): x\notin C)$ – вектор, составленный из частот всех искаженных вертикальных элементов, и $\mathbf{u}_0 = (P(\xi_c) = \nu(c): c\in C, q = 0, h = 0)$ – тривиальное решение системы уравнений (15) в случае $\mathbf{v} = 0$. Тогда для достаточно малых значений параметра $\mathbf{v}$ решение этой системы уравнений, а значит, и экстремальной задачи (6), (7), можно найти, итерируя формулу $\mathbf{u}_{k+1} = \mathbf{u}_k - \mathbf{T}^{-1}\mathbf{W}(\mathbf{u}_k, \mathbf{v})$, $k = 0,1,...$ Здесь $\mathbf{W} = (W_{P(\xi_c)}: c\in C, W_q, W_h)$ – преобразование, стоящее в левой части системы (15), $\mathbf{T}^{-1}$ – отображение, обратное производной $\mathbf{W}_u'$, взятой в точке $(\mathbf{u}_0, \mathbf{0})$. Матрица этого отображения уже найдена – это матрица (14). Таким образом получаем следующий алгоритм.

3. 2. Алгоритм нахождения априорных вероятностей $P(\xi_c), c\in C$, и вероятности искажения q при заданном множестве C неискаженных элементов.

1. Полагаем в качестве нулевого приближения:

$P_0(\xi_c) = \nu(c)$ для всех $c\in C$, $q_0 = 0$, $h_0 = 0$.

2. Последующие приближения находим по формулам

$$P_{k+1}(\xi_c) = P_k(\xi_c) + W_{P(\xi_c)} +$$
$$+\frac{\left[\alpha(c)\beta + \frac{\gamma}{N_C}\right]W_q + \left[\frac{\beta}{N_C} - \alpha(c)\right]W_h}{\beta^2 + \gamma},\ c\in C,$$

$$q_{k+1} = q_k + \frac{\beta W_q + W_h}{\beta^2 + \gamma},\ h_{k+1} = h_k - \frac{\gamma W_q - \beta W_h}{\beta^2 + \gamma},\ k = 1,2,...,$$

где $W_{P(\xi_c)}$, W_q и W_h – функции, определенные равенствами (15) и вычисленные для переменных предыдущего, то есть k-того, приближения, а коэффициенты $\alpha(c), c\in C$, β и заданы соотношениями (11).

3. Алгоритм заканчивает работу, когда последовательные приближения при заданной точности перестают изменяться.

3. 3. Нахождение совокупности C неискажённых вертикальных элементов строки

Предположим, что совокупность неискаженных вертикальных элементов точно не известна, но предполагается, что ею является некоторое множество C. Проверку правдоподобности этого предположения можно провести следующим образом.

Прежде всего множество C должно обладать тем свойством, что оно, дополненное всеми возможными искажениями его элементов, должно содержать множество всех вертикальных элементов страницы, то есть

$$\bigcup_{c\in C} F_c \supset X. \qquad (16)$$

Далее, предположив, что C – совокупность неискаженных элементов, вычислим с помощью алгоритма предыдущего пункта априорные вероятности $P(\xi_c), c\in C$, и вероятность искажения q. После чего найдем полные вероятности

$$P(x)=\sum_{c\in C} q^{\|x-c\|^2}(1-2q)^{2s-\|x-c\|^2}P(\xi_c),\ x\in X, \qquad (17)$$

и рассмотрим функционал

$$\Gamma=\sum_{x\in X}\nu(x)\left|P(x)-\nu(x)\right|, \qquad (18)$$

где $\nu(x)$ – частота появления элемента x.

Будем считать, чем меньше значение функционала Г, тем правдоподобнее, что рассматриваемое множество C является совокупностью неискаженных элементов.

Приведенный ниже алгоритм строит *локально наиболее правдоподобную* совокупность C в том смысле, что добавление или изъятие любого элемента из этой совокупности не уменьшает значение функционала Г. Не в строгом смысле совокупность C можно назвать *минимальной*, так ее построение начинается с множества, которое имеет минимальное или близкое к минимальному число элементов среди множеств, удовлетворяющих обязательному условию (16). Отметим еще, что наличие весовых множителей $\nu(x)$ в сумме (18) объясняется тем, что предпочтительным является хорошее совпадение значений $P(x)$ и $\nu(x)$ на вертикальных элементах x с наиболее полной статистикой.

Алгоритм основан на том, что полная вероятность $P(x)$ появления вертикального элемента $x\in X$ в условиях малой вероятности искажений q существенно зависит от того, считается ли этот элемент искаженным или неискаженным. Действительно, согласно формуле (17) во втором случае по сравнению с первым вероятность $P(x)$ имеет дополнительное слагаемое $(1-2q)^{2s}P(\xi_x)$. Причем в отличие от остальных слагаемых, имеющих первый или больший порядок малости по q, дополнительное слагаемое имеет нулевой порядок малости. Из этого можно сделать два вывода.

Во-первых, наибольшие значения частоты $\nu(x)$ с большой степенью правдоподобия приходятся на неискаженные элементы $c\in C$. На этом основана первая часть приведенного ниже алгоритма.

Во-вторых, если наблюдается большая по модулю невязка

$$\Delta(x)=P(x)-\nu(x), \qquad (19)$$

то ее в некоторых случаях можно уменьшить, объявив элемент x принадлежащим множеству C, если он до этого не принадлежал C, или, наоборот, исключив элемент x из множества C, если он ему принадлежал. Это свойство используется во второй части алгоритма, минимизирующей функционал Г путем добавления к множеству или изъятия из него вертикальных элементов, дающих наибольшую по модулю невязку (19).

3. 4. Алгоритм построения совокупности неискаженных вертикальных элементов строки

I часть

1. Полагаем $C=\varnothing$ (пустое множество).
2. Присоединяем к множеству C любую точку в которой достигается максимум функции $\nu(x)$ на множестве $X\setminus\bigcup_{c\in C}F_c$;
3. Повторяем действия п. 2, пока $X\setminus\bigcup_{c\in C}F_c\neq\varnothing$.

II часть

1. Полагаем $R=\varnothing$.
2. Применяем к текущему множеству C алгоритм предыдущего пункта для нахождения вероятностей $P(\xi_c), c\in C$, и вероятности искажения q. Вычисляем полные вероятности $P(x)$, $x\in X$, по формуле (17).
3. Находим любую точку x, в которой модуль функции $\nu(x)\Delta(x)$, вычисляемой с помощью формулы (19), достигает максимума на множестве $X\setminus R$ и помещаем x в множество R. Если
   i) $x\in C$ и $\Delta(x)>0$, то исключаем x из множества C и выполняем действия, описанные в п.п. 2 и 3 части I;
   ii) $x\notin C$ и $\Delta(x)<0$, то включаем x в множество C.

Если множество C изменилось после этих действий, снова выполняем п 2. Если функционал Г, вычисляемая по формуле (18), не уменьшился, то в случае i) возвращаем x в множество C, а в случае ii) исключаем x из этого множества.

4. Повторяем действия п. 3, пока $X\setminus R\neq\varnothing$.
5. Повторяем действия п. п. 1–4 до момента, когда функционал Г перестанет изменяться.

4. Классификация связных символов с помощью нечеткой классификации вертикальных элементов строки

Окончательным этапом обработки изображения текста является классификация связных символов в изображении текста. Для этого потребуется сформировать связные символы изображения текста, которые состоят из вертикальных элементов строки. В данном случае каждый связный символ представлен в двух формах. Изображение связного символа представлено набором вертикальных элементов строки, а структура связного символа последовательностью чисел, каждое

из которых соответствует номеру вертикального элемента строки.

Для проведения классификации связных символов необходима таблица распределения вероятности вертикальных элементов строки по найденному множеству неискаженных вертикальных элементах строки, как показано в алгоритме построения неискаженных вертикальных элементов строки. Данная таблица получена для всех вертикальных элементов $x \in X$ с равным числом компонент, и вероятность $P(\xi_c | x)$, представлена как функция двух переменных на декартовом произведении $C \times X$, и представляет собой нечеткую классификацию совокупности X всех вертикальных элементов строки в изображении текста.

Перед классификацией связных символов необходимо выполнить ряд действий по предварительной обработке всего изображения, которое представлено всей совокупностью вертикальных элементов строки $x \in X$. Все предварительные операции можно разделить на несколько этапов:

1) Выделить все связные символы изображения текста использую пробелы между ними. Каждый связный символ представлен совокупностью смежных вертикальных элементов строки (рис. 2).
2) Определить ширину каждого связного символа, как количество вертикальных элементов строки из которых он состоит. Сформировать последовательность связных символов по мере увеличения ширины символа. В исследуемом изображении этот диапазон колеблется в пределах от 5 до 76. Причем наиболее широкие символы могут состоять, как правило, из нескольких символов.
3) Сформировать последовательность связных символов, состоящую из чисел, которые показывают количество компонент в соответствующем вертикальном элементе строки. Представить каждый связный символ набором числа компонент в вертикальных элементах строки.

Классификация связных символов выполнялась при условии совпадения классифицируемых символов по ширине (количество вертикальных элементов строки) и составу компонент в каждом вертикальном элементе строки. Допустимая погрешность по ширине связного символа выбиралась ±2 вертикальных элемента строки, что соответствует принятой вероятностной модели. Погрешность классифицируемых символов по количеству компонент в вертикальных элементах строки составляющих связные символы составляла максимум два смежных вертикальных элемента строки, что так же вписывается в вероятностную модель. Предварительная обработка позволяет исключить из процесса классификации заведомо разные связные символы в изображении текста.

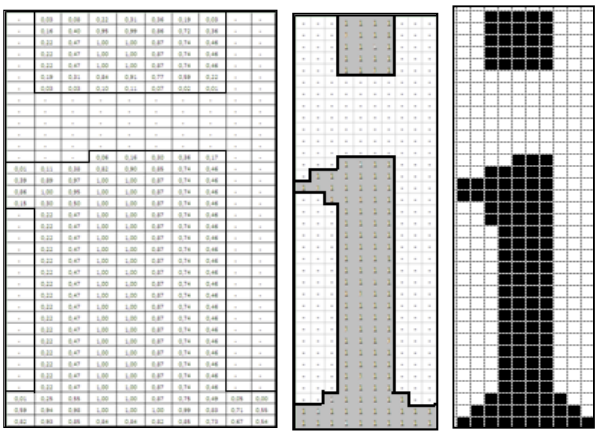
Если классифицируемые связные символы удовлетворяют принятым условиям предварительной обработки, то осуществляется формирование отдельного класса связных символов. Результатом классификации будет изображение связного символа. Каждая точка этого изображения получена, как результат суммы поэлементного произведения двух бинарных векторов представляющих вертикальные элементы строки, которые входят в состав сравниваемых связных символов. Для вероятностной оценки изображения связного символа каждую точку вертикального элемента строки $x \in X$ необходимо умножить на соответствующие вероятности $P(\xi_c | x)$ (выражение 3). Общее выражение можно представить следующим образом.

$$Let = \sum_{i,j=I;c \in C}^{K} \left[P(\xi_c | x_i) * x_i \right] * \left[P(\xi_c | x_j) * x_j \right], \qquad (20)$$

где Let – изображение связного символа, каждая точка которого имеет свою вероятностную оценку; K – количество вертикальных элементов в связном символе; $c \in C$ – неискаженный элемент строки из множества неискаженных элементов; $x_{i,j} \in X$ – классифицируемые вертикальные элементы строки.

Результат предложенной обработки представлен на рис. 5. На рис. 5, *а* показано общее изображение символа «i». Точки общего изображения показывают значения суммы произведений апостериорной вероятности $P(\xi_c | x)$ соответствующих вертикальных элементов строки, выражение 20. После нормировки и удаления из общего изображения точек с малой долей вероятности (Let <0,5), формируется выходное изображение связного символа «i», рис. 5, *б*. Данное изображение является представителем класса для всех изображений связного символа «i». Для сравнения полученного результата классификации связного символа приведено идеальное изображения символа «i» на рис. 5, *в*.

В результате проведенной классификации по всем связным символам, состоящих из вертикальных элементов строки формируется словарь связных символов. Словарь связных символов состоит из изображений представителей классов связных символов, аналогичных изображению символа «i», рис. 5, *б*. В рассматриваемом случае сформированный словарь состоит из 101-го представителя класса изображения. Необходимо отметить, что некоторые классы изображений связных символов могут иметь по одному представителю в своем классе. Есть две причины подобной ситуации: 1) изображение символа действительно присутствует в единственном представлении; 2) изображение символа состоит из нескольких смежных связных символов. Причиной подобной ситуации является отсутствие разделяющего вертикального элемента строки, который не содержит ни одной черной точки в своей структуре. Подобные случаи представлены на рис. 6, *а*. Решение данной задачи довольно простое. Если в словаре связных символов присутствуют отдельные изображения символов представленных на рис. 6, *а*, то необходимо используя выражение 20 найти максимум отношения числа совпадений с каждым представителем класса к общему числу представителей в каждом классе. В результате такой процедуры все символы, состоящие из нескольких связных символов, будут разделены между собой. Каждый разделенный связный символ будет помещен в соответствующий ему класс в качестве еще одного члена данного класса. Общее число классов связных символов в словаре связных символов уменьшилось со 101 до 67 классов. Фрагмент итогового словаря связных символов состоящего из 67 классов приведен на рис. 6, *б*.



а *б* *в*

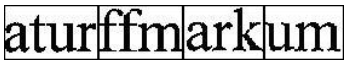
Рис. 5. Изображение символа «i», полученное в результате классификации связных символов с помощью нечеткой классификации вертикальных элементов строки: *а* — общее изображение символа «i»; *б* — изображение представителя класса символа «i» состоящее из наиболее вероятных точек в общем изображении; *в* — неискаженное изображение символа «i»

В результате приведенной классификации связных символов в изображении текста создаются:

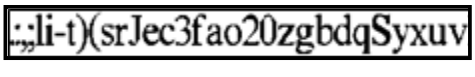
– словарь связных символов, рис. 6, *б*;

– карта размещения связных символов на изображении текста.

К полученным данным теперь можно применить методы компрессии без потерь и оценить количественные показатели сжатия изображения текста.



а



б

Рис. 6. Изображения представителей классов связных символов: *а* — некоторые классы, состоящие из одного представителя; *б* — фрагмент итогового словаря связных символов изображения текста

5. Апробация результатов исследований

Для проведения качественного и количественного анализа работы предложенного алгоритма была выбрана страница отсканированного текста формата А4 в формате – *.bmp со следующими параметрами: тип шрифт – Times New Roman, кегль – 12, межстрочный интервал – 1, разрешение – 300 dpi, глубина цвета – 1 бит. После разложения связных (связных) символов на вертикальные элементы строки их общее число составило 4671 вертикальных элемента, количество строк в изображении текста равно 48, высота вертикального элемента строки равна 51-й точке изображения. Общий размер данных изображения текста составляет – 1080,2 килобайта

Согласно п. 2 и 3 вероятностной модели все множество вертикальных элементов строки (ВЭС), присутствующее на тестовой странице, разбивается на подмножества с равным числом компонент. Каждое подмножество вертикальных элементов строки обрабатывается отдельно согласно алгоритму построения неискаженных вертикальных элементов строки. В таблице 1 показано:

– количество компонент, по которым распределено все множество вертикальных элементов строки;

– число вертикальных элементов строки, которое принадлежит каждому подмножеству;

– число итераций, которые необходимо выполнить согласно II-ой части предложенного алгоритма;

– начальное и конечное число центров, значений вероятности искажений вертикальных элементов строки-q и усредненного значения $\Delta(x)$, выражение 19.

Согласно I-ой части алгоритма построения совокупности неискаженных вертикальных элементов строки начальное число центров $c \in C$ и начальное значение вероятности искажения (q) находится согласно п. п. 1–3 для каждого подмножества с равным числом компонент. Рассмотрим основные этапы работы предложенного алгоритма. При практическом рассмотрении в анализируемом изображении текста максимальное число компонент всего множества вертикальных элементов строки равно 4 (табл. 1). Для краткости рассмотрим пример обработки вертикальных элементов строки с одной компонентой. Распределение невязки (выражение 19) полной вероятности появления вертикального элемента строки P(x) и частоты появления элемента x–v(x), а также начальное значение вероятности искажения – q, при минимально возможном количестве центров $c \in C$ после выполнения I-ой части алгоритма представлено на рис. 7, *а*.

На рис. 7, б представлен результат выполнения II-ой части предложенного алгоритма. Как видно из табл. 1 и рис. 7 для однокомпонентных ВЭС для остановки работы алгоритма потребовалось провести 10 итераций II-ой части алгоритма. Число неискаженных центров вертикальных элементов строки увеличилось со 124 до 343. Вероятность искажения вертикальных элементов строки (q) при этом уменьшилась с 0,17 до 0,049. Нужно отметить, что модуль среднего значения невязки (выражение 19) уменьшился с 2,4 (рис. 7, *а*) до 0,53 (рис. 7, *б*). По достижению указанных показателей функционал Г (выражение 18) перестает изменяться, достигая своего минимального значения. Это означает, что рассматриваемое множество конечных центров ВЭС (в приведенном случае это 343) является наиболее правдоподобной совокупностью неискаженных элементов строки состоящих из одной компоненты.

Аналогичные вычисления проводятся по всем подмножествам вертикальных элементов строки состоящих из равного числа компонент, в соответствии с рассматриваемым алгоритмом.

Следует подчеркнуть, что разбиение всего изображения текста на вертикальные элементы строки и дальнейшее вычисление минимальной наиболее правдоподобной совокупности неискаженных элементов с точки зрения сжатия всего изображения текста не дает позитивного результата. Если всю плоскость анализируемого изображение текста, состоящую из связных

символов (букв, цифр, знаков препинания) просто заменить полученной наиболее вероятной совокупностью неискаженных вертикальных элементов, то исследуемое изображение необходимо представить как словарь центров (наиболее правдоподобных вертикальных элементов строки) и карту размещения этих центров на плоскости изображения. Очевидно, что общее число центров ВЭС, состоящее из всех компонент, равно сумме конечных центров (табл. 1), в данном случае это 2296. При подобном количестве числа центров наибольшей проблемой при сжатии становится карта размещения из-за большого диапазона индексов центров ВЭС, которые составляют изображения отдельных связных символов. Это легко представить, если для указанного изображения текста минимальная ширина символа (. – точка) равна 5 неискаженным ВЭС, максимальная ширина связного символа (W) равна 45 центрам.

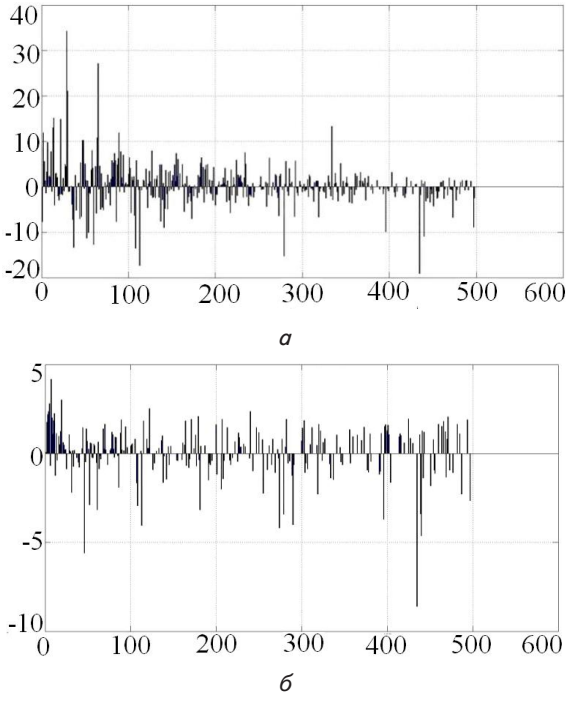


Рис. 7. Начальное и конечное распределение значений невязки (выражение 19): *а* – начальное распределение невязки при начальном количестве центров c∈C. 500 - ВЭС с одной компонентой. 124 – исходных центра. Вероятность искажений q=0,17; *б* – конечное распределение невязки при конечном количестве центров c∈C. 500 – ВЭС с одной компонентой. 343 – конечных центра. Вероятность искажений q=0,049. Число итераций алгоритма – 10

Таким образом, проводить процедуру компрессии на плоскости изображения текста, где связные символы представлены набором наиболее правдоподобных неискаженных ВЭС не имеет практического смысла из-за значительного числа представителей словаря. Известно, что чем больше число представителей словаря, тем меньше степень компрессии. Поэтому, имея в словаре 2296 представителя, прогнозировать достаточные количественные показатели в сжатии не стоит. В табл. 2 приведены данные, которые наглядно характеризуют данное утверждение. Здесь приведены:

– все изображение текста разделено на указанное количество строк;

– для соответствующего числа строк показано число центров ВЭС;

– представлен словарь центров ВЭС в сжатом виде в килобайтах;

– представлена карта размещения индексов центров ВЭС в килобайтах;

– показано общее сжатие словаря и карты размещения;

– для сравнения приведены показатели алгоритма JB2 входящего в формат DjVu.

Таблица 1

Начальные и конечные значения основных параметров при построении неискаженной совокупности вертикальных элементов строки

N компонент	Число ВЭС	Число итераций	$\sum\|\Delta(x)\|/Nx$		Число центров		Вероятность искажений – q	
	Nx		начало	конец	начало	конец	начало	конец
1	500	10	2,4	0,53	124	343	0,17	0,049
2	2150	8	1,99	0,39	303	1021	0,145	0,029
3	1779	5	3,42	0,28	161	850	0,11	0,017
4	242	3	4,63	0,24	15	82	0,124	0,031

Объем данных приведен в килобайтах (kb). Исходный размер изображения текста – 1080,2 (kb). Для всех данных, в качестве алгоритма сжатия без потерь, использовался алгоритм 7-zip, который является модификацией словарного метода компрессии LZ77 [9] – LZMA (Lempel-Ziv-Markov chain-Algorithm). Как видно из приведенных характеристик, невзирая на значительное число представителей словаря – 2296 центра ВЭС на 48 строках изображения текста, остаточное количество данных после сжатия сравнительно невелико – 6,97 (kb). При этом выходной размер карты размещения центров ВЭС почти на порядок выше – 52,9 (kb). Для сравнения приведены характеристики результатов сжатия алгоритма JB2 формата DjVu.

Таблица 2

Результаты сжатия словаря и карты размещения неискаженных элементов строки на плоскости изображения текста

Количество строк	2	3	4	6	12	24	36	48
Число центров – ВЭС	360	499	579	730	1099	1662	1936	2296
Словарь (kb)	1,37	1,9	2,13	2,63	3,37	5,07	5,92	6,97
Карта размещения (kb)	2,48	3,52	4,6	6,93	13,6	26,3	38	52,9
Сжатие общее (kb)	3,81	5,37	6,69	9,53	17	31,2	43,8	59,7
JB2 (kb)	1,14	1,59	1,89	2,32	3,31	5,01	7	8,7

Из анализа табл. 2 видно, что суммарная степень компрессии (Сжатие общее) значительно уступает показателям алгоритма JB2. Эти результаты определяют необходимость перехода от использования центров ВЭС к использованию связных символов. Для этого

требуется создать словарь связных символов и соответственно карту размещения связных символов, как показано в пункте 6 данной статьи.

В табл. 3 приведены данные в килобайтах, которые характеризуют результат компрессии:

– словаря связных символов, рис. 6, *б*;

– карты размещения связных символов на плоскости изображения текста;

– общего сжатия словаря связных символов и карты размещения связных символов – Alg (kb);

– число классов связных символов в словаре – N классов.

На рис. 8 приведены кривые, которые отображают остаток данных после сжатия изображения текста алгоритмом JB2 формата DjVu и предложенным алгоритмом сжатия. Из сравнительного анализа результатов обработки изображения текста, очевидно, что предложенный метод практически совпадает с алгоритмом JB2 при малых объемах связных символов, на обрабатываемых изображениях текста (2–6 строк). И имеет значительное преимущество в сжатии всей страницы изображения текста (48 строк). Остаток данных после сжатия изображения страницы текста алгоритмом JB2 составляет 8,7 кб, результат использования предложенного алгоритма можно поместить в объем 5,48 кб, что на 37 % превышает показатели формата DjVu при сжатии бинарного изображения текста.

Таблица 3

Результаты сжатия словаря и карты размещения связных символов на плоскости изображения текста

Количество строк	2	3	4	6	12	24	36	48
JB2 (kb)	1,14	1,59	1,89	2,32	3,31	5,01	7	8,7
N классов	31	39	40	44	47	54	66	67
Словарь символов (kb)	1,34	1,68	1,75	1,98	2,04	2,35	3,06	3,25
Карта размещения символов (kb)	0,379	0,398	0,438	0,548	0,81	1,32	1,81	2,28
Alg (kb)	1,55	1,89	2,11	2,4	2,77	3,52	4,75	5,48

Рис. 8. Результаты компрессии изображения текста алгоритмом JB2 формата DjVu и предложенным алгоритмом сжатия

На рис. 9, *а* представлен фрагмент изображения исходной тестовой страницы текста. На рис. 9, *б* показан аналогичный фрагмент выходного изображения.

For truly effective compression, both general purpose and data specific data processing techniques should be readily composable together based on certain consideration of the data at hand. For example, consider a log file of network events. As the states of each event are encoded in fields, records of events of the same type would form a table. Since many tables may be mixed together in a log file, some invertible transformation of the data is needed to group like records into separate tables so that some table compression algorithm can be applied. The Vcodex data transformation platform provides a framework to develop and use such data transforms.

а

For truly effective compression, both general purpose and data specific data processing techniques should be readily composable together based on certain consideration of the data at hand. For example, consider a log file of network events. As the states of each event are encoded in fields, records of events of the same type would form a table. Since many tables may be mixed together in a log file, some invertible transformation of the data is needed to group like records into separate tables so that some table compression algorithm can be applied. The Vcodex data transformation platform provides a framework to develop and use such data transforms.

б

Рис. 9. Фрагменты изображения страницы текста с параметрами: тип шрифт – Times New Roman, кегль 12, межстрочный интервал – 1, разрешение – 300 dpi, глубина цвета – 1 бит: *а* – исходное изображение; *б* – выходное изображение

Качественный анализ предполагал отсутствие ошибок в изображении символов текста. Из приведенных фрагментов видно, что общий вид и структура текста исходного и полученного изображения практически идентичны (рис. 9, *а, б*).

6. Выводы

Используя отдельный этап представления связных символов изображения текста, в виде вертикальных элементов строки и применив их нечеткую классификацию, была получена минимальная наиболее правдоподобная совокупность неискаженных элементов строки. Приняв во внимание ограничения вероятностной модели, для каждого вертикального элемента строки была получена вероятность того, что он является искажением найденного неискаженного элемента строки.

Исследовав возможности компрессии данных представленных в виде словаря неискаженных вертикальных элементов строки и их карты размещения, была проведена классификация связных символов на основе нечеткой классификации вертикальных элементов. Формирование словаря связных символов основывалось не на сравнительном анализе геометрических форм сравниваемых символов [17], а на вероятностной оценке соответствующих вертикальных элементов строки, которые представляют состав классифицируемых связных символов.

Предложенный алгоритм представления и обработки изображения текста позволил получить достаточно высокую степень сжатия при хорошем качестве восстановленного изображения.

Сравнение с лучшим в настоящее время специальным алгоритмом сжатия изображений текста – JB2, входящим в формат DjVu, показало, что предлагаемый алгоритм сжатия изображения текста имеет преимущество в степени компрессии данных порядка 37 % при обработке страницы текста изображения. Для наиболее часто используемого на практике разрешения изображения текста 300 dpi авторами

были получены следующие сравнительные количественные показатели сжатия:

– в работе [17] преимущество над JB2 – 8 %;
– в работе [18] преимущество над JB2 – 25 %;
– в данной работе преимущество над JB2 – 37 %.

Это является основной характеристикой представленного метода и раскрывает новые возможности повышения информативности представления текстовых графических данных в инженерных реализациях.

Литература

1. Technical Papers from AT&T Labs [Electronic Resource] / Available at: http://djvuzone.org/techpapers/index.html
2. DjVu.org [Electronic Resource] / Available at: http://www.djvu.org/
3. Haffner, P. DjVu: Analyzing and Compressing Scanned Documents for Internet Distribution [Text] / P. Haffner, L. Bottou, P. G. Howard, Y. LeCun // Fifth International Conference on Document Analysis and Recognition (ICDAR'99), 1999. – P. 625. doi:10.1109/icdar.1999.791865
4. JBIG2.com : An Introduction to JBIG2 [Electronic Resource] / available at : URL : http://jbig2.com/index.html
5. Айвазян, С. А. Прикладная статистика: Классификация и снижение размерности [Текст] / С. А. Айвазян, В. М. Бухштабер, И. С. Енюков и др. – М.: Финансы и статистика, 1989. – 607с.
6. Иванов, В. Г. Сжатие изображений на основе автоматической и нечеткой классификации фрагментов [Текст] / В. Г. Иванов, Ю. В. Ломоносов, М. Г. Любарский // Проблемы управления и информатики. – 2009. – № 1 – С. 52–63. doi:10.1615/jautomatinfscien.v41.i1.40
7. Шлезингер, М. И. Математические средства обработки изображений [Текст] / М. И. Шлезингер. – Киев: Наукова думка, 1983. – 200 с.
8. Гонсалес, Р. Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с.
9. Земсков, В. Н. Сжатие изображений на основе автоматической классификации [Текст] / В. Н. Земсков, И. С. Ким // Известия вузов. Электроника. – 2003. – № 2. – С. 50–56.
10. Mallat, S. Multiresolution Approximation and Wavelet Orthonormal Bases L2(R) [Text] / S. Mallat // Trans of the American Mathematical Society. – 1989. – Vol. 315 (1). – P. 68-87. doi:10.1090/s0002-9947-1989-1008470-5
11. Gupta, M. R. Segmenting for wavelet compression [Text] / M. R. Gupta, A. Stroilov. – Data Compression Conference (DCC' 05), 2005. – 462 p. doi:10.1109/dcc.2005.80
12. Montiel, E. Texture classification via conditional histograms [Text] / E. Montiel, A. S. Aquado, M. S. Nixon // Pattern Recognition Letters – 2005. – Vol. 26 (11). – P. 1740–1751. doi:10.1016/j.patrec.2005.02.004
13. Lakhani, G. Improving Image Decomposition Method of the 3-MRC Coding of Scanned Compound Document Images [Text] : Sixth Indian Conference / G. Lakhani // Computer Vision, Graphics & Image Processing, 2008. – P. 289–296. doi:10.1109/icvgip.2008.94
14. Ding, W. Block-based Fast Compression for Compound Images [Text] / W. Ding, D. Liu, Y. He, F. Wu // IEEE International Conference on Multimedia and Expo. – 2006. – P. 809–812. doi:10.1109/icme.2006.262624
15. Malvar, H. S. Fast Adaptive Encoder for Bi-Level Images [Text] / H. S. Malvar // Data Compression Conference (DCC '01), 2001. – 253 p. doi:10.1109/dcc.2001.917156
16. Imura, H. Compression and String Matching Method for Printed Document Images [Text] : 10th Intern. Conference / H. Imura, Y. Tanaka // Document Analysis and Recognition, 2009. – P. 291–295. doi:10.1109/icdar.2009.182
17. Иванов, В. Г. Сжатие изображения текста на основе выделения символов и их классификации [Текст] / В. Г. Иванов, М. Г. Любарский, Ю. В. Ломоносов // Проблемы управления и информатики. – 2010. – № 6. – С. 111–122. doi:10.1615/jautomatinfscien.v42.i11.50
18. Иванов, В. Г. Сжатие изображения текста на основе формирования и классификации вертикальных элементов строки в графическом словаре символьных данных [Текст] / В. Г. Иванов, М. Г. Любарский, Ю. В. Ломоносов // Проблемы управления и информатики. – 2011. – № 5. – С. 98–109. doi:10.1615/jautomatinfscien.v43.i10.40
19. Канатников, А. Н. Функции нескольких переменных [Электронный ресурс] / А. Н. Канатников, А. П. Крищенко. – Режим доступа: http://mathmod.bmstu.ru/Docs/Eduwork/la_fnp/FNP-14.pdf
20. Некоторые материалы из лекций по анализу. Теорема о неявной функции [Электронний ресурс] / Режим доступа: http://new.math.msu.su/Sites/demosite/Uploads/Neyavnaya%20funktsiya.B6EF5654E2C8486FBA1EF9F186B32F1A.pdf