

УЛУЧШЕНИЕ ФОТОГРАФИЙ ТЕКСТА: ИСПРАВЛЕНИЕ ГЕОМЕТРИЧЕСКИХ ИСКАЖЕНИЙ

студ. Левашов А. Е.

Московский Государственный Университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, лаборатория математических методов обработки изображений (МГУ, ВМК)

1. Введение

В условиях широкого распространения сотовых телефонов со встроенной цифровой фотокамерой с автофокусом и высокого разрешения, востребованной задачей является их использование как «карманного ксерокса». Однако фотографии текста, получаемые таким образом, страдают рядом недостатков, два основных из которых это шумы, связанные с плохими условиями освещения (не всегда текст фотографируется в летний полдень), и геометрические искажения, в первую очередь связанные с неплоскостью фотографируемого листа бумаги (например, типичный изгиб страницы книги и, соответственно, строк текста). Текст на подобных фотографиях хотя и читаем, но неудобен для восприятия, программы распознавания текста также испытывают трудности при работе с такими текстами. Настоящая работа направлена на коррекцию второго из перечисленных недостатков – распрямление изогнутых строк печатного текста.

Предполагается, что на входе имеется изображение одной страницы с текстом, у которого строки в целом близки к горизонтальным. Строки детектированы с помощью многомасштабного детектора жирных линий [2] и оценена высота строк. Детектированные строки записаны в виде последовательности координат и направлений нормали, но имеют разрывы внутри одной строки (слово детектируется как жирная линия, а пробел между словами естественно нет). Целью является получить изображение, на котором строки текста прямолинейны и горизонтальны. Задача решается путем поиска параметрической аппроксимации линий строк, а также поиска преобразования, переводящего такие кривые в параллельные горизонтальные прямые, после чего это преобразование применяется к исходным фотографиям текста.

2. Устранение разрывов в строках.

Результатом работы детектора жирных линий [2] и последующей классификации найденных линий по ширине, генеральному направлению и цвету (светлый или темный) является набор осевых линий слов. Для поиска строк требуется увязать эти фрагменты строки между собой, т.е. разбить слова по отдельным строкам (рисунок 1). Предлагается решать эту задачу с помощью морфологических операций с предварительным преобразованием системы координат, в которой записаны детектируемые линии, а именно преобразовать строки (в векторном представлении) в горизонтальные. Ищется преобразование плоскости такое, чтобы среднее направление нормалей стало равно $(0,1)$. Найдем среднее направление нормалей $(\overline{n_x}, \overline{n_y})$ по всем детектированным точкам и преобразование R такое, чтобы $(\overline{n_x}, \overline{n_y}) \cdot R = (0,1)$. Перемножим теперь все точки в собранных списках на R , тем самым перейдем в новую систему координат, где строки горизонтальные.

После этого запишем преобразованные строки в промежуточное изображение B и сделаем следующую операцию.

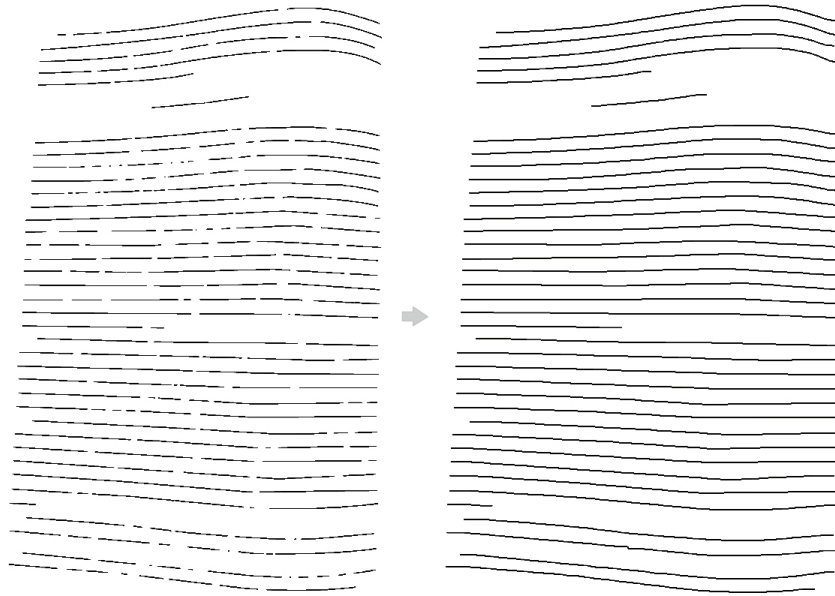


Рисунок 1. Детектированные строки: до и после устранения разрывов.

Пусть P – пиксель на изображении B . Считаем, что $P = 1$, если P принадлежит детектированной линии, 0 – иначе. Вокруг каждого пикселя P из промежуточного изображения берется окно 3×3 пикселя

$$\begin{Bmatrix} P_9 & P_2 & P_3 \\ P_8 & P_1 & P_4 \\ P_7 & P_6 & P_5 \end{Bmatrix}. \text{ При действии морфологической операции с окном } 3 \times 3 \text{ на изображении}$$

результатирующий пиксель обозначается как P^* . Производятся следующие морфологические операции:

1. *for* $iter = [1 \dots k_3]$ *do*
2. *for* $i = [1 \dots k_1]$ *do*
3. *for each* $P \in B : P^* = \max_{j=1..9}(P_j)$
4. *end do*
5. *for* $i = [1 \dots k_2]$ *do*
6. *for each* $P \in B : P^* = \max(P_1, P_4, P_8)$
7. *end do*
8. Утончение B алгоритмом [3] до линий шириной в 1 пиксель
9. *end do*

Для того чтобы устранить разрывы, вызванные пробелами в строках, достаточно взять $k_3 = 1$, однако, в зависимости от количества и величины разрывов, k_3 можно увеличить до нескольких раз. Считаем, что $k_1 + k_2 < \text{высота_строки}$, чтобы при отсутствии дефектов в детектируемых строках разные строки гарантировано не соединились. В нашей реализации мы брали $k_1 = k_2 = \frac{\text{высота_строки}}{2}$.

В п.3 происходит морфологическое расширение линий, это нужно, чтобы разорванные куски одной линии соединялись, т.к. из-за изогнутости они могут находиться на разных ординатах. Пункт 6 выполняет анизотропное морфологическое расширение вдоль строки, чтобы расширение в этом направлении происходило быстрее и приводило к слиянию слов. Т.к. алгоритм [3] в п.8 при утончении линий не отбрасывает пиксели с концов линии, то п.2–8 можно делать несколько раз, при этом «длина слов» будет каждый раз увеличиваться до их слияния. При наличии небольших дефектов в детектированных строках приведенный алгоритм показывает хорошую стабильность.

3. Параметрическая аппроксимация текстовой строки

После устранения разрыва в строках, можно собрать текстовые строки в списки связанных пикселей, в нашей реализации мы пользовались сборкой из алгоритма [4], т.е. тем самым мы отходим от изображения и переходим к списку массивов точек. При сборке точек в списки будем записывать только исходные детектированные точки, а остальные использовать для того, чтобы благодаря им все кусочки одной текстовой линии записывались в один массив.

Теперь задача аппроксимации точек каждой текстовой строки параметрической кривой упрощается, имеется выделенное направление и параметрическая кривая может быть задана в форме, где координаты (x, y) не равноправны: отклонение по y как функция от x . Параметризация кривой выбиралась в виде $y = ax^3 + bx^2 + cx + d$, так как оно соответствует физической модели изогнутого упругого стрелы. Оценка вектора параметров $\vec{p}^T = (a \ b \ c \ d)^T$ осуществлялось методом наименьших квадратов с использованием сингулярного разложения [1], что обеспечивает устойчивость решения за счет регуляризации. Решалась система уравнений

$$\begin{pmatrix} x_i^3 & x_i^2 & x_i & 1 \end{pmatrix} \vec{p} = (y_i), \quad (1)$$

где $(x_i \ y_i)$ – это набор точек j -ой линии. Для каждой текстовой строки $j = 1..N$ получаем аппроксимацию $y_j = a_j x^3 + b_j x^2 + c_j x + d_j$. Каждую такую кривую будем преобразовывать в прямую линию $y = \bar{y}_j$, где $\bar{y}_j$ – среднее среди всех ординат y_{ij} в точках, относящихся к данной кривой j . Получаем преобразование

$$\Delta y_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i - \bar{y}_i = a_i x^3 + b_i x^2 + c_i x + \bar{d}_i. \quad (2)$$

Этот сдвиг $\Delta y_i(x)$ переводит кривую в горизонтальную прямую.

4. Распрямление текста

Если исходное изображение имеет размеры $w \times h$, то для N строк преобразования $\Delta y_i(x)$ уже найдены. Остается найти $\Delta y(x, y)$ для любого $0 < y < h$. Предлагается коэффициенты $(a_j \ b_j \ c_j \ \bar{d}_j)$ также аппроксимировать кубической кривой, т.е. искать эти коэффициенты кривой как функцию y . Таким образом, будем искать матрицу M_p коэффициентов размерности 4×4 . Решая систему $4 \times N$ уравнений:

$$\begin{pmatrix} \bar{y}_j^3 & \bar{y}_j^2 & \bar{y}_j & 1 \end{pmatrix} M_p = (a_j \ b_j \ c_j \ \bar{d}_j) \quad (3)$$

относительно элементов матрицы M_p получим искомое преобразование страницы текста. Конечная формула смещений выглядит так:

$$\Delta y(x, y) = a(y)x^3 + b(y)x^2 + c(y)x + \bar{d}(y) = \begin{pmatrix} y^3 & y^2 & y & 1 \end{pmatrix} M_p \begin{pmatrix} x^3 \\ x^2 \\ x \\ 1 \end{pmatrix} \quad (4)$$

После того, как для любой точки найдена формула смещения, приведем алгоритм растеризации результирующего изображения. Исходное изображение обозначается I_{IN} , результирующее изображение – I_{OUT} . Пиксель в позиции (x, y) обозначается $I_{IN}(x, y)$, если (x, y) содержит не целые компоненты, то цвет пикселя выбирается с помощью билинейной интерполяции.

1. *for each* $I_{OUT}(x, y) \in I_{OUT}$ *do*

$$2. \quad (x_1, y_1) = (x, y + \Delta y(x, y)), \quad \Delta y(x, y) \text{ из (4)}$$

$$3. \quad (x_2, y_2) = (x_1, y_1) \cdot R^{-1}$$

$$4. \quad I_{OUT}(x, y) = I_{IN}(x_2, y_2)$$

5. Результаты

Приведенный алгоритм показал хорошие результаты по выпрямлению текстовых линий, на рисунке 2 можно увидеть выпрямленные текстовые строки на правых изображениях. Стоит отметить, что на изображении зачастую встречаются короткие линии, например, последние строки абзацев. Аппроксимируя их можно получить большую ошибку в тех местах, где линии нет. Поэтому выбиралось 80% самых длинных кривых, по которым и оценивались параметры преобразования страницы. В случае если не все строки участвовали в оценке параметров M_p , это дает незначительные ошибки. В будущем планируется заменить часть алгоритма, описанную в разделе 2 на анализ кривых непосредственно в векторном виде с помощью RANSAC – подобных геометрических методов, что значительно сложнее логически, но быстрее и устойчивее с вычислительной точки зрения.

6. Благодарности

Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы и гранта РФФИ 09-07-92000-ННС_а.

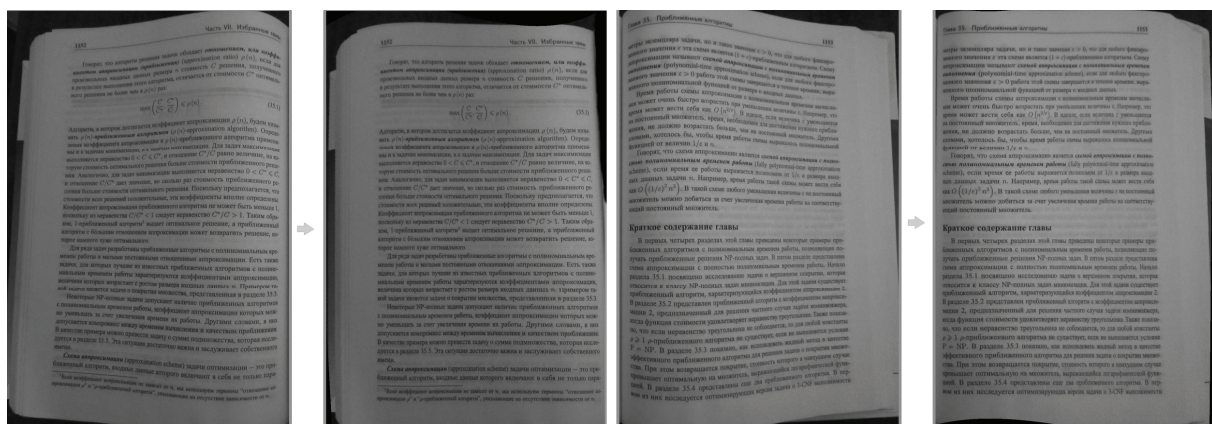


Рисунок 2. Результаты работы алгоритма.

Литература

- [1] G. E. Forsythe, M. A. Malcolm, C. B. Moler, Computer Methods for Mathematical Computations. P. 210, Prentice Hall, Inc., 1977.
- [2] N. A. Khanina, E. V. Semeikina, D. V. Yurin. Color Blob and Line Detection in Scale-Space. Pattern Recognition and Image Analysis, 21(2): 267–269, Pleiades Publishing, Ltd., 2011.
- [3] Zhang T.Y., Suen C.Y. A fast parallel algorithm for thinning digital patterns // Communications of the ACM, V. 27, No. 3, P. 236–239, 1984.
- [4] А. Левашов, Д. Юрин. «Система быстрого обнаружения параметрических кривых на серых и цветных изображениях с контролем достоверности» // в: 21-я международная конференция по компьютерной графике и зрению GraphiCon'2011. Москва, Россия, 2011, с. 212–215.

Лаборатория математических методов обработки изображений

Левашов А. Е.

<http://imaging.cs.msu.ru>

Email: alexeylevashov89@gmail.com

TEXT IMAGES ENHANCEMENT: CORRECTING OF GEOMETRIC DISTORTION

stud. Levashov A. E.

Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics

This work describes the method of straightening of curved lines in printed text. It is assumed that the input of our algorithm is an image of one text page. Text strings are detected using multiscale thick lines detector and height of strings is estimated. The goal is to get the image on which the text strings are straight and horizontal. The problem is solved by searching the parametric approximation of the text lines, and by finding the transformation that converts such curves into parallel horizontal lines. Then the transformation is applied to the original photos of the text.