Usage of continuous skeletal image representation for document images de-warping.

Anton Masalovitch, Leonid Mestetskiy
Moscow State University, Moscow, Russia
anton_m@abbyy.com, l.mest@ru.net

Abstract

In this paper application of continuous skeletal image representation to documents' image de-warping is described. A novel technique is presented that allows to approximate deformation of interlinear spaces of image based on elements of image's skeleton that lie between the text lines. A method for approximation of whole image deformation as combination of single interlinear spaces deformations is proposed and representation of it in the form of 2-dimensional cubic Bezier patch is suggested. Experimental results for batch of deformed document images are given that compare recognition quality of images before and after de-warping process. These results prove efficiency of the proposed algorithm.

1. Introduction

All the modern OCR systems assume that text lines in a document are straight and horizontal while in real images they are not. Image can be deformed before recognition in various ways. For example, if a thick book is scanned, text lines on the scan may be wrapped near the spine of book. If a digital camera is used to retrieve the image instead of a scanner, the text lines may be still wrapped because of low-quality optics of digital cameras. One important example of such deformation is the rounding of an image on borders as result of barrel distortion. Moreover, several types of deformation could be applied to the same image, making it impossible to build a precise model of image deformation. This is how the task of image de-warping appears.

The approach proposed in this paper is based on the construction of outer skeletons of text images. The main idea of the proposed algorithm is based on the fact that it is easy to mark up long continuous branches that define interlinear spaces of the document in outer skeletons. We approximate such branches by cubic Bezier curves to find a specific deformation model of each interlinear space of the document. On the basis of a set of such interlinear spaces' approximations, the whole approximation of the document is built in the form of a 2-dimensional cubic Bezier patch. After all this work is completed, we can de-warp an image using obtained approximation of image deformation.

This work is an extension of the article [1]. In this paper new method of automatic search for interlinear branches of skeleton is described. Also iteration method of image deformation approximation adjustment is given.

To test our algorithm we compare recognition results for a batch of images before and after the de-warping process.

2. Existing solutions

Algorithm of automatic image de-warping is needed nowadays for automatic OCR systems. Plenty of algorithms for image deformation approximation appeared in the last several years (see for example [7-11]). Unfortunately, most of these algorithms have some disadvantages that make them unusable for commercial OCR systems.

Existing solutions can be divided to three approaches:

- First approach is to single out text lines by combining close black objects and then approximating each line shape using some characteristic points of line's black objects. For example, one can approximate text lines' shape by using middle points of black objects' bounding rectangles. Main disadvantage of this approach is that it is hard to define such characteristic points of black objects that can give a stable approximation of line shape.
- Second approach is to build a model of possible deformation of an image and then try to apply this model for a specific image. Main disadvantage of this method is that it is almost impossible to build a complete model of image deformation. And if such a model describes only one type of deformation, one

should make sure that the used model can be applied for processing the concrete image.

- Finally, the third approach is to describe some estimation of text lines' straightness and iteratively deform image to achieve a maximum possible straightness of text lines. Main disadvantage of this method is that it uses numerical computing, and therefore is time-consuming, while the results of the method are often unpredictable.

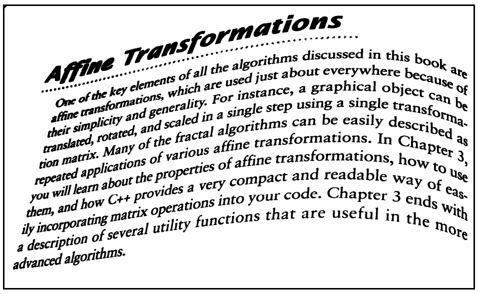
In our work we try to avoid described disadvantages. So our goal is to create image de-warping algorithm that does not depend on text symbols quality, is applicable to most of possible optic image deformations, with predictable results and not time-consuming.

3. Characteristics of images under consideration

It is necessary to describe some characteristics of images that our algorithm works with:

- Initial image should be black and white, with black text and white background. It should not contain inverted areas. It also should not contain noise or textures. In all modern OCR systems efficient add-ons exist that allow bringing almost every image to the marked model. And applied binarization and noise removal technique may be very rough because our algorithm does not depend on text symbols quality.
- Initial image should contain one big text block. This is an important assumption, because the proposed algorithm works with interlinear spaces rather than with text lines, and therefore initial image must contain a sufficient number of long text lines located one under another. All modern OCR systems can divide initial image into a set of text blocks with high precision, even when the images are deformed.
- Let us also assume that the deformation of text lines in an image can be approximated by continuous cubic curves and patches. This assumption is also not very restrictive, since most common deformations of images are created by cameras and scanners. Such deformations can be approximated even by quadratic patches and curves. As for more complicated cases, experiments have shown that cubic approximation is precise enough for them. In the case if additional experiments will show that cubic approximation is not sufficient after all, the degree of Bezier curves and patches can be easily increased without making considerable modifications to the proposed algorithm.

One example of an image with which our algorithm works is represented on figure 1.


Affine Transformations

One of the key elements of all the algorithms discussed in this book are affine transformations, which are used just about everywhere because of their simplicity and generality. For instance, a graphical object can be translated, rotated, and scaled in a single step using a single transformation matrix. Many of the fractal algorithms can be easily described as repeated applications of various affine transformations. In Chapter 3, you will learn about the properties of affine transformations, how to use them, and how C++ provides a very compact and readable way of easily incorporating matrix operations into your code. Chapter 3 ends with a description of several utility functions that are useful in the more advanced algorithms.

Figure 1. Processing image example

4. Problem definition

Let us assume that we have image $I(x, y)$, where I is the color of image pixel with coordinates (x, y). Let us also assume that this image contains text block with deformed lines. We further assume that we can rearrange pixels in this image without changing their colors to retrieve document image where initial lines become straight and horizontal. So, we want to develop a continuous vector function $\overline{D}(x, y)$ to obtain a de-warped image in the form: $I'(x, y) = I(\overline{D}_x(x, y); \overline{D}_y(x, y))$. This function $\overline{D}(x, y)$ will be an approximation of the whole image deformation.

To estimate the quality of our de-warping algorithm, we attempt to recognize the image before and after de-warping using one of the modern OCR systems. Recognition quality in modern OCR systems depends heavily on the straightness of text lines in images under consideration. Therefore, an improvement in recognition quality after image de-warping is a good evaluation of the quality of our de-warping algorithm.

5. Continuous border representation of binary image

In this work skeleton of polygonal figures is exploited. Before using such skeleton with binary images we must define representation of discrete binary image as a set of continuous polygonal figures.

Let us assume that a scanned document is stored in the form of a binary image represented as a Boolean matrix (one bit per pixel). A discrete model of the binary image is the integer lattice I in the Euclidean plane R^2 with 0 and 1 representing black and white elements. For elements of the lattice the relation of the 4-adjacent neighborhood is given. We designate $B \subset I$ as the set of black and $W \subset I$ as the set of white nodes of the lattice. Sets (B, W) serve as a model of the discrete binary image. In the same Euclidean plane R^2, we define the polygonal figure μ as the set of the points formed by association of a finite number of non-overlapping bounded closed domains. This figure is then a model of the continuous binary image. There is a problem consists in the construction of the figure μ that adequately describes properties of the discrete image B. In mathematical terms this problem is posed as an approximation of a discrete object with a continuous object. Natural criteria of good approximations should satisfy the following natural criteria:

1) $B \subset \mu$, $W \subset \left[R^2 \setminus \mu\right]$, where $[\ ]$ means closure of a set;

2) Let $x, y \in I$ be a pair of adjacent nodes of the lattice and s_{xy} be a segment connecting these nodes. Then if $x, y \in \mu$, then $s_{xy} \in \mu$, and if $x, y \notin \mu$ then $s_{xy} \cap \mu = \varnothing$.

The first condition means that the figure covers all black points of a discrete image and all white points lie either outside of or on the boundary of the figure. The second condition can be reduced to the condition that the boundary of μ lies in the interface between white and black boundary points of the discrete image.

Let M be the set of all figures satisfying conditions 1 and 2. Any of them can be considered a continuous model of a binary image with acceptable accuracy. As we are going to build a skeleton of this figure, the most convenient representation for us is the figure with a piecewise linear boundary, since for such figures there are effective algorithms for construction of a skeleton. In this situation it is natural to choose from M a polygonal figure (PF) with minimal perimeter (see fig. 2). First, such PF exists and it is unique. Second, the number of its vertices is close to minimal among all PF satisfying conditions 1 and 2.

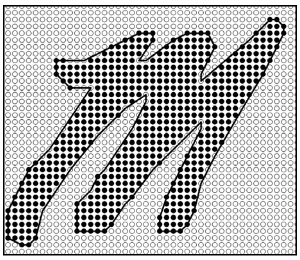


Figure 2. Representation of raster object with polygonal figure with minimal perimeter

The algorithm for solving this problem which requires a single pass over a raster image, has been described in [4].

6. Continuous skeletal representation of an image

The choice of the polygonal figure as a continuous model of the binary image reduces the problem of construction of a skeleton of the image to the well-known medial axis transform [5]. Contrary to discrete images for which the skeleton is determined ambiguously, the concept of a skeleton of a continuous figure has a strict mathematical formulation. The skeleton of a figure is the locus of points of centers of maximal empty circles. An empty circle does not contain any boundary points of the figure. The maximal empty circle is a circle which is not contained in any other empty circle, and which is not congruent to another. Note that empty circles can be thus either internal or external for the domains comprising the figure. Accordingly their centers form internal and external skeletons of the figure (see fig. 3).

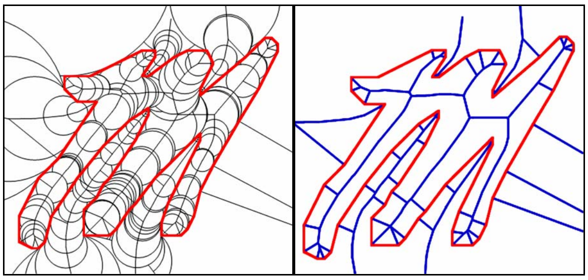


Figure 3. Empty circles for polygonal figure and skeleton of polygonal figure.

This definition applies to any type of shape, not just a polygon. However there exist effective algorithms for construction of polygonal figures [4,6]. The algorithm used [2,3] is based on a generalization of Delauney triangulation for a system of sites of two types (points and segments) that comprise a PF boundary. It builds a skeleton in time O(n log n) where n is the number of PF vertices.

Skeleton of polygonal figure can represented as a planar graph, where nodes are points on a plane and bones are straight lines that connect the nodes. In such representation of a skeleton all nodes have no less than three nearest points on the border of the area and all bones lie between two linear fragments of the area border. Later in this article we will use only graph representation of a skeleton.

Let us also define a knot in skeleton as a node with more then two connected bones and final node as a node with only one connected node. And let us define a branch of skeleton as a consistent set of bones that has final node or knot node on each end and does not have knots in the middle of the branch. Later in this article we will operate only with branches of the skeleton and not with single bones.

7. Main idea of the algorithm

Main idea of the proposed algorithm is that in outer skeleton of text document image, one can easily find branches that lie between adjacent text lines. Then, one can use this separation branches to approximate deformation of interlinear spaces in an image.

The proposed algorithm consists of the following steps:

- Continuous skeletal representation of an image is built.
- Skeleton is filtered (useless bones are deleted).
- Long near-horizontal branches of the skeleton are singled out.
- List of singled out branches is filtered to leave only branches that lie between different text lines.
- Cubic Bezier approximation is built for each branch.
- Bezier patch is built based on the obtained curves.

8. Image and skeleton preprocessing

As was mentioned before, one of the steps of our algorithm is the preprocessing step, on which we try to delete all small garbage branches and branches that can be obviously determined as non-interlinear from the skeleton. Let us describe this step in more detail.

First of all, before building a skeleton, we flood all white horizontal strokes with length smaller than some predefined threshold. By doing so, we glue symbols in words in one line, so we erase from image skeleton a lot if intersymbol branches that are useless for our algorithm. We set the value of the flooding parameter equal to 0.1 inches or 30 pixels for 300 dpi images (this value determined empirically). That value is sufficient to glue most adjacent symbols and not to glue adjacent curved lines.

Then we build outer skeleton of the expanded image.

The next step is to delete branches of the skeleton that divide different parts of the same object. Such branches describe borders of one symbol and are not relevant for the whole text line. We also delete branches of skeleton that divide objects in image and border of an image. Figure 4 shows an example of such image preprocessing.

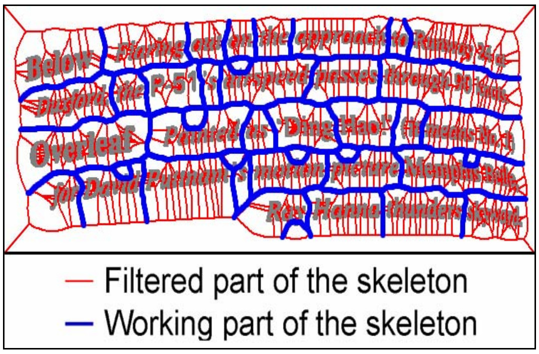


Figure 4. Image skeleton after preprocessing

9. Skeleton bones clusterization

After outer skeleton of a document image was built, we could divide branches of the skeleton into two groups: branches that lie between objects in one text line and branches that lie between adjacent text lines.

The main idea of the proposed algorithm is that such clusterization can be performed automatically for any document image.

First we sort out all skeleton branches that are shorter then some predefined threshold. Such branches appear when several long branches connected in one point. Such short branches works only for connectivity propose, the angle of such branches is unpredictable, so they are not used during clusterization process (see fig. 5).

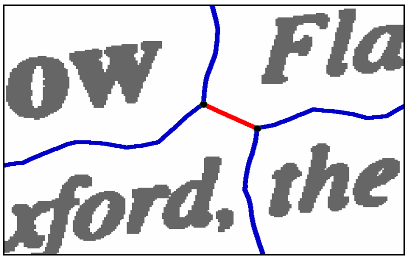


Figure 5. Short branch that connects several long brances.

As a threshold value for short branches we use empirical value of 0.05 inches or 15 pixels for 300 dpi images (determined empirically). It is about half of small letters height for standard font size, so we don't treat any of intersymbol branches as short.

To clusterize long branches we define parameter $A_{\max}$ - maximal absolute value of angle of interlinear branch (as angle of skeleton branch we use angle of linear approximation of that branch). Experiments show that it is possible for each image skeleton to define this parameter in such a way that all long vertical branches with |angle| > $A_{\max}$ will be only intersymbol branches.

This idea can be confirmed by graphic representation, if we draw all linear approximation of skeleton branches on one plane, so that they all begin in one point. For a document image the obtained figure will look like a cross, the horizontal part of which is created by interlinear branches, while the vertical part is created by intersymbol branches (see fig. 6).

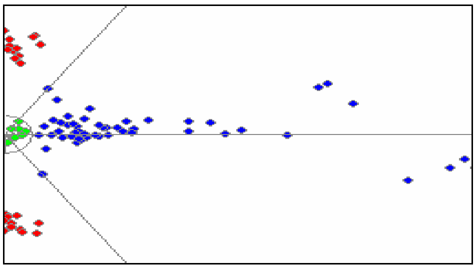


Figure 6. Branches of skeleton from figure 4 marked on one plane.

To define parameters $A_{\max}$ we use simple automatic clusterization mechanism. Each possible value of angle divides all branches into two classes with angle greater and less then the given threshold. For each class we define μ as the mean value of the angle in this class and σ as the standard deviation of the angle from the mean value. Using these two values we can define separation factor of two classes $J(t)$ in the form:

$$J(t) = \frac{\sigma_R + \sigma_L}{\mu_R - \mu_L}.$$

Then we iterate among the angles, looking for that with the minimum separation factor, using one degree as the size of the step (see fig. 7).

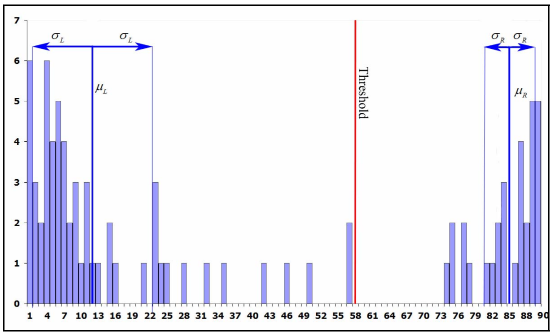


Figure 7. Histogramm of branches' angles from figure 6 with detected threshold

After clusterization we delete all long vertical branches from the skeleton (see fig. 8).

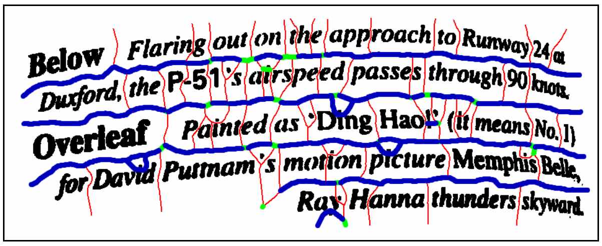


Figure 8. Skeleton of document image after clusterization of branches.

10. Building interlinear branches.

After all vertical branches are deleted, the remaining branches are processed in cycle according to the following rules:

- If two nodes are connected by two non-intersected branches (such a problem appears when text language includes diacritics and additional branch goes between diacritic and symbol (see fig. 9)), we delete most curved branch of these two.

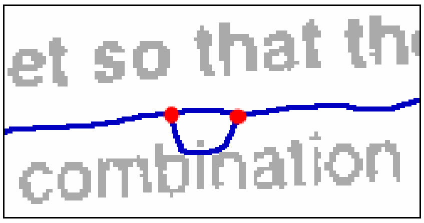


Figure 9. Two skeleton branches around a diacritic mark.

- If three branches are connected in one point (such a problem appears because some short branches remain after all vertical branches were deleted (see fig. 10)), we delete the shortest of the these branches.

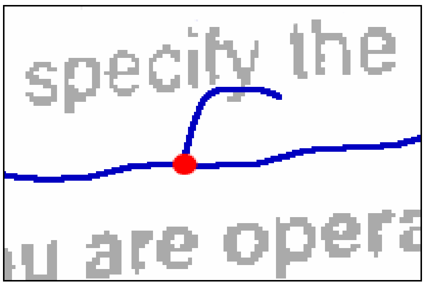


Figure 10. Remaining of vertical branches.

- If two long horizontal branches are connected near the border of an image (such a problem appears when two interline branches merge together outside the borders of a text block (see fig. 11)), we separate connection node of these branches into two independent nodes.

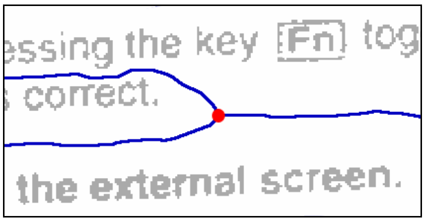


Figure 11. Two branches connected on the end of text line.

After all these rules were applied, only long horizontal branches that lie between adjacent text lines remain in the skeleton. We approximate them with cubic Bezier curves using method of least-square approximation.

11. Approximation of image deformation

After we get approximation of each interlinear space in the image, we must approximate deformation of whole image.

Define control points of interlinear curves as I_{ki}, where k is the index of a curve and i is the index of a control point on this curve.

For each set of points $\left\{\bigcup_{k=0}^{n} I_{ki}\right\}$ (control points from all interline curves with same index) we build approximation with vertical Bezier curve. Let us define control points of obtained curves as P_{ij}, where i is the index of initial control points and j is the index of new control points on created curve (see fig. 12).

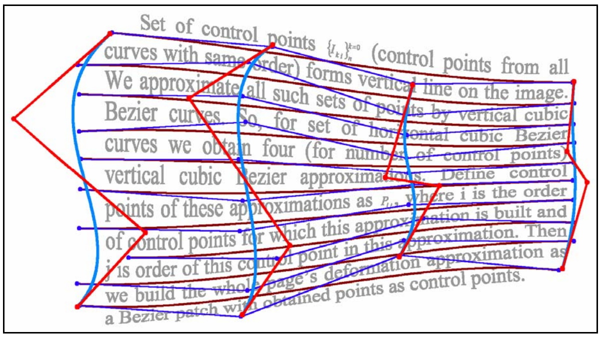


Figure 12. Definition of control points of Bezier patch

After we get the set of points P_{ij}, we can build whole image deformation using Bezier patch. In other words, our approximation may be described by the following formula:

$$\overline{D}(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}P_{ij}*b_{i,3}(x)*b_{j,3}(y)$$

$b_{r,3}(t)$ - cubic Bernstein polynomial.

12. Bezier patch adjustment

Unfortunately, when we approximate interline spaces we cannot define clearly where each text line begins. Because of this, vertical points of the patch might be very randomly curved.

To avoid such an effect we use the following adjustment procedure:

For each interline curve $C_i(x)=\sum_{j=0}^{3}A_j^i*b_{j,3}(x)$ we search for nearest curve in Bezier patch. Define obtained curve as $\overline{C}_i(x)=\overline{D}(x,y_i)=\sum_{i=0}^{3}\sum_{j=0}^{3}P_{ij}*b_{i,3}(x)*b_{j,3}(y_i)$.

Define α and β as parameters of points on the curve $\overline{C}_i$ nearest to begin and end points of curve C_i.

$$\begin{cases}\alpha=\arg\min\limits_t \rho\left(\overline{C}_i(t),C_i(0)\right)\\ \beta=\arg\min\limits_t \rho\left(\overline{C}_i(t),C_i(1)\right)\end{cases}$$

Then we build curve C_i' that identical to C_i, but differs in parameterization (has shifted parameters), so that $C_i'(\alpha)=C_i(0)$ and $C_i'(\beta)=C_i(1)$. In other words,

$$C_i'(t)=C_i\left({}^{t-\alpha}\!\big/_{\beta-\alpha}\right)=$$

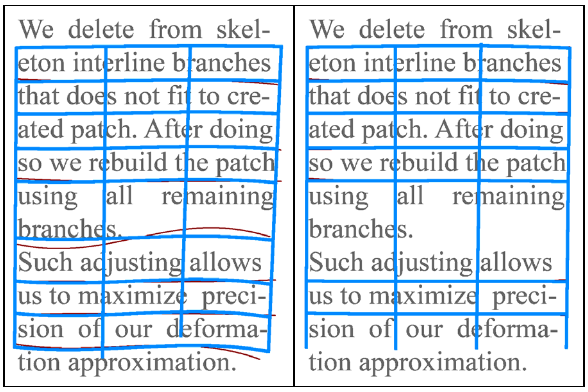
$$\sum_{j=0}^{3}A_j^i*b_{j,3}\left({}^{t-\alpha}\!\big/_{\beta-\alpha}\right)=\sum_{j=0}^{3}B_j^i*b_{j,3}(t)$$

Then we calculate mean deviation d between curves C_i' and $\overline{C}_i$. If this deviation is greater than some predefined threshold, the original curve C_i must be excluded from patch creation, otherwise original curve C_i must be replaced with C_i'.

After the processing of all initial curves is completed we build a new Bezier patch using updated set of curves.

We repeat this procedure until deviations of all initial curves from curves from Bezier patch reach some predefined threshold.

This adjustment procedure allows to approximate vertical borders of text block and improves deformation approximation of whole page because of exclusion of erroneously created curves (see fig. 13).


We delete from skel-
eton interline branches
that does not fit to cre-
ated patch. After doing
so we rebuild the patch
using all remaining
branches.
Such adjusting allows
us to maximize preci-
sion of our deforma-
tion approximation.

We delete from skel-
eton interline branches
that does not fit to cre-
ated patch. After doing
so we rebuild the patch
using all remaining
branches.
Such adjusting allows
us to maximize preci-
sion of our deforma-
tion approximation.

Figure 13. Image deformation approximation before and after Bezier patch adjustment.

13. Experimental results

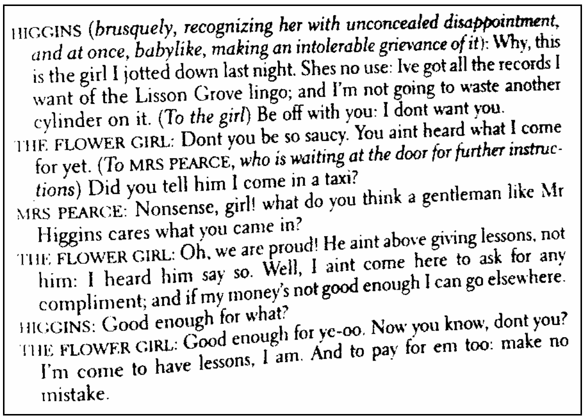
To test efficiency of our algorithm we take a set of 31 images. All images from this set satisfy the conditions described in section 3 – they are black-and-white images without noise, which contain one big text blocks with deformed lines. We recognize all these images with one modern OCR system before and after the de-warping process.

For deformed images there were 2721 recognition errors on all pages (4.92% of all). For de-warped images there were 830 recognition errors on all pages (1.50% of all symbols). Therefore, after the de-warping process 1891 errors were corrected (69.5% of original errors). In addition, 14 lines were not found on initial images, because of their high deformation, and after de-warping all text lines were defined correctly.

The attained results show high efficiency of the proposed algorithm, but its quality is not maximal yet. Recognition quality for straight images is higher than 99,5% in modern OCR systems. And for de-warped images we obtain the quality of only 98,5%. The main reason for this gap is that our algorithm deforms symbols a little during de-warping and that in turn causes errors in symbols' recognition.

On figures 14-16 an example of image de-warping for one of the images from our test set is given.

Also our algorithm was tested during Document Image De-warping Contest that was held in CBDAR 2007 [12]. On the contest de-warping algorithms were applied to test base of 100 images (test set available for download here - http://www.iupr.org/downloads/data). Experiments shown that mean edit distance for images de-warped by our algorithm was less then 1% on contest data set. Those results are statistically the same for the other two participants of the contest. And on quarter of test images our algorithm shown lowest edit distance.


HIGGINS (*brusquely, recognizing her with unconcealed disappointment, and at once, babylike, making an intolerable grievance of it*): Why, this is the girl I jotted down last night. Shes no use: Ive got all the records I want of the Lisson Grove lingo; and I'm not going to waste another cylinder on it. (*To the girl*) Be off with you: I dont want you.

THE FLOWER GIRL: Dont you be so saucy. You aint heard what I come for yet. (*To* MRS PEARCE, *who is waiting at the door for further instructions*) Did you tell him I come in a taxi?

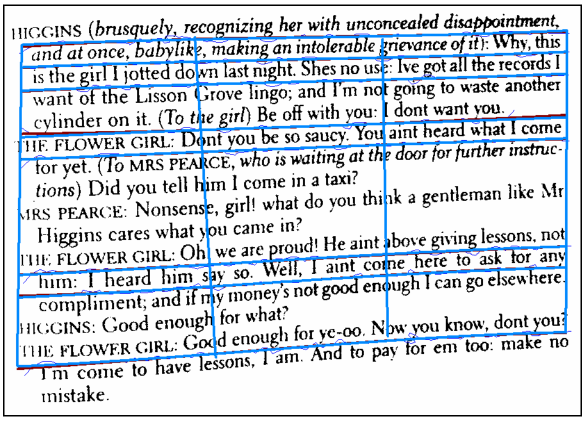
MRS PEARCE: Nonsense, girl! what do you think a gentleman like Mr Higgins cares what you came in?

THE FLOWER GIRL: Oh, we are proud! He aint above giving lessons, not him: I heard him say so. Well, I aint come here to ask for any compliment; and if my money's not good enough I can go elsewhere.

HIGGINS: Good enough for what?

THE FLOWER GIRL: Good enough for ye-oo. Now you know, dont you? I'm come to have lessons, I am. And to pay for em too: make no mistake.

Figure 14. Initial deformed image


HIGGINS (*brusquely, recognizing her with unconcealed disappointment, and at once, babylike, making an intolerable grievance of it*): Why, this is the girl I jotted down last night. Shes no use: Ive got all the records I want of the Lisson Grove lingo; and I'm not going to waste another cylinder on it. (*To the girl*) Be off with you: I dont want you.

THE FLOWER GIRL: Dont you be so saucy. You aint heard what I come for yet. (*To* MRS PEARCE, *who is waiting at the door for further instructions*) Did you tell him I come in a taxi?

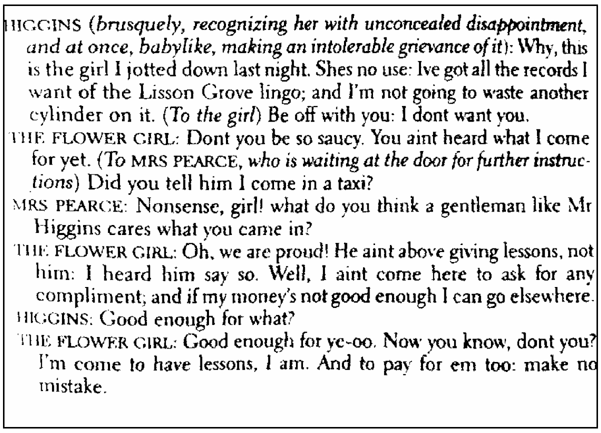
MRS PEARCE: Nonsense, girl! what do you think a gentleman like Mr Higgins cares what you came in?

THE FLOWER GIRL: Oh, we are proud! He aint above giving lessons, not him: I heard him say so. Well, I aint come here to ask for any compliment; and if my money's not good enough I can go elsewhere.

HIGGINS: Good enough for what?

THE FLOWER GIRL: Good enough for ye-oo. Now you know, dont you? I'm come to have lessons, I am. And to pay for em too: make no mistake.

Figure 15. Image deformation approximated with Bezier patch.


HIGGINS (*brusquely, recognizing her with unconcealed disappointment, and at once, babylike, making an intolerable grievance of it*): Why, this is the girl I jotted down last night. Shes no use: Ive got all the records I want of the Lisson Grove lingo; and I'm not going to waste another cylinder on it. (*To the girl*) Be off with you: I dont want you.

THE FLOWER GIRL: Dont you be so saucy. You aint heard what I come for yet. (*To* MRS PEARCE, *who is waiting at the door for further instructions*) Did you tell him I come in a taxi?

MRS PEARCE: Nonsense, girl! what do you think a gentleman like Mr Higgins cares what you came in?

THE FLOWER GIRL: Oh, we are proud! He aint above giving lessons, not him: I heard him say so. Well, I aint come here to ask for any compliment; and if my money's not good enough I can go elsewhere.

HIGGINS: Good enough for what?

THE FLOWER GIRL: Good enough for ye-oo. Now you know, dont you? I'm come to have lessons, I am. And to pay for em too: make no mistake.

Figure 16. De-warped image.

14. Future works

The main direction of feature work is to develop a better de-warping algorithm based on obtained image deformation approximation. De-warping algorithm that we use is very naïve, which heads to some additional recognition mistakes on the de-warped images.

More accurate approximation of deformation of vertical borders of text blocks is also one of our prior tasks.

15. Conclusion

This article describes a novel technique for approximation of text document image deformation based on continuous skeletal representation of an image.

In our work we try to avoid main disadvantages of existing de-warping solutions: use of separate symbol characteristics, use of specific deformation model, use of unpredictable numerical methods.

Main advantage of proposed algorithm is that it does not rely on quality of the initial text. Initial characters can be broken, flooded or erroneously binarized – proposed algorithm does not depend on it.

This paper describes all main steps of the proposed algorithm: construction of skeletal representation of an image, preprocessing of image's skeleton, detection of interlinear branches of the skeleton, approximation of such branches, final approximation of image deformation.

Based on the proposed algorithm a prototype of fully automatic system of image de-warping was built.

Experimental results that prove efficiency of the proposed algorithm and its importance for recognition of deformed images are given.

Acknowledgments

This research was supported by a grant from Russian foundation for basic research (grant 05-01-00542).

Bibliography

[1] A.A. Masalovitch, L.M. Mestetskiy, "Document Image Deformation Approximated by the Means of Continuous Skeletal Representation of the Image", Proceedings of international conference PRIP (Pattern Recognition and Information Processing), 2007, pp. 279-284.

[2] L.M. Mestetskiy, "Skeletonization of polygonal figures based on the generalized Delaunay triangulation", Programming and computer software, 25(3), 1999, pp. 131-142.

[3] L.M. Mestetskiy, "Skeleton of multiply connected polygonal figure", Proceedings of international conference "Graphicon", 2005.

[4] S. Fortune, "A sweepline algorithm for Voronoi diagrams", Algorithmica, 2, 1987, pp. 153-174.

[5] D.T. Lee, "Medial axes transform of planar shape", IEEE Trans. Patt. Anal. Mach. Intell. PAMI-4, 1982, pp.363-369.

[6] C.K. Yap, "An O(n log n) algorithm for the Voronoi diagram of the set of simple curve segments", Discrete Comput. Geom., 2, 1987, pp.365-393.

[7] Hironori Ezaki, Seiichi Uchida, Akira Asano, Hiroaki Sakoe, "Dewarping of document images by global optimization", Proceedings of international conference ICDAR, 2005, pp. 302-306.

[8] Udrian Ulges, Christoph H. Lampert, Thomas M. Breuel, "A Fast and Stable Approach for Restoration of Warped Document Images", Proceedings of international conference ICDAR, 2005, pp. 384-388.

[9] Li Zhang, Chew Lim Tan, "Warped Image Restoration with Application to Digital Libraries", Proceedings of international conference ICDAR, 2005, pp. 192- 196.

[10] A. Yamashita, A. Kawarago, T. Kaneko, K.T. Muira, "Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system", Proceedings of international conference ICPR, 2004, pp 482-485.

[11] M.S. Brown, W.B. Seales, "Image Restoration of Arbitrarily Warped Documents". IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26 , Issue 10, 2004, pp. 1295 - 1306.

[12] Faisal Shafait, Thomas M. Breuel, "Document Image Dewarping Contest". Proceedings of 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, Sep. 2007.