Материалы конференции
International Conference on Machine Vision

Fast localization and rectification of documents folded into thirds

A. Ershov [1,2], D. Tropin [1,3], D. Nikolaev [1,3]
[1] Smart Engines Service LLC, Prospekt 60-letiia Oktiabria 9, Moscow, 117312, Russia;
[2] Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Bolshoy Karetny per. 19, build.1, Moscow, 127051, Russia;
[3] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Prospekt 60-letiia Oktiabria 9, Moscow, 119333, Russia

Abstract

The ubiquitous usage of smartphones makes camera-captured document images as widely used as scanned ones as the input of a modern document recognition system. A document captured by a smartphone camera may appear mechanically distorted in the image creating the need for an image rectification step. The present paper considers a particular case of document image distortions. Specifically, if a business document is sent via postal service, it may need to be folded to fit the envelope. Once the document is taken out of the envelope and unfolded, its geometric shape is distorted in a very particular pattern. Since the most popular envelope formats in Europe and America require the document to be folded into thirds, this case is considered in this paper. We propose a novel content-independent model-based algorithm for the localization and geometrical rectification of documents folded into thirds. Our algorithm outperforms current SOTA rectification methods on the recently published dataset FDI by key rectification accuracy metrics (AD and CER) and is able to rectify documents held in hand. Moreover, it can be executed on a mobile CPU and has a reasonable execution time: it takes only about 17 ms to localize a document and about 110 ms to projectively rectify it. So it makes it possible to embed the proposed algorithm into document recognition systems designed for on-device acquisition.

Keywords: Folded documents, document rectification, document unwarping, on-device acquisition.

Citation: Ershov A, Tropin D, Nikolaev D. Fast localization and rectification of documents folded into thirds. Computer Optics 2025; 49(6): 1049-1060. DOI: 10.18287/COJ1755.

Introduction

Despite the fact that digital technologies are becoming more and more widespread and business tasks often become being solved on computers, the paper document flow is not decreasing. A lot of business documents are still printed on paper, folded into envelopes and sent with postal services. The problem of integrating such physical printed documents into electronic business systems is an important issue creating the need for printed document digitization systems. Unfortunately, when sending a paper document by postal service, one has to fold it so that it fits into an envelope, distorting the paper sheet.

Common OCR systems are often tuned to operate with scanned documents [1, 2]. However, flatbed scanners are bulky and inconvenient, whereas mobile phones equipped with cameras are compact and handy. These cameras are able to reproduce an image of a document that has enough quality for a human to be able to read the text on the document image. Unfortunately, unlike scanned images, photos can make the documents appear [3] distorted (see Fig. 1). To deal with such distortions one has to make use of a "rectification" or "unwarping" algorithm.

If a document is simply a flat rectangle, it will be captured by a camera as a quadrilateral, which is a projective distortion of the initial rectangle. Even such simple perspective deformation may cause trouble to a document recognition system. The problem of perspective quadrilaterals rectification is well-studied, a lot of methods were proposed to assess it and many of them show extremely good results [4, 7].

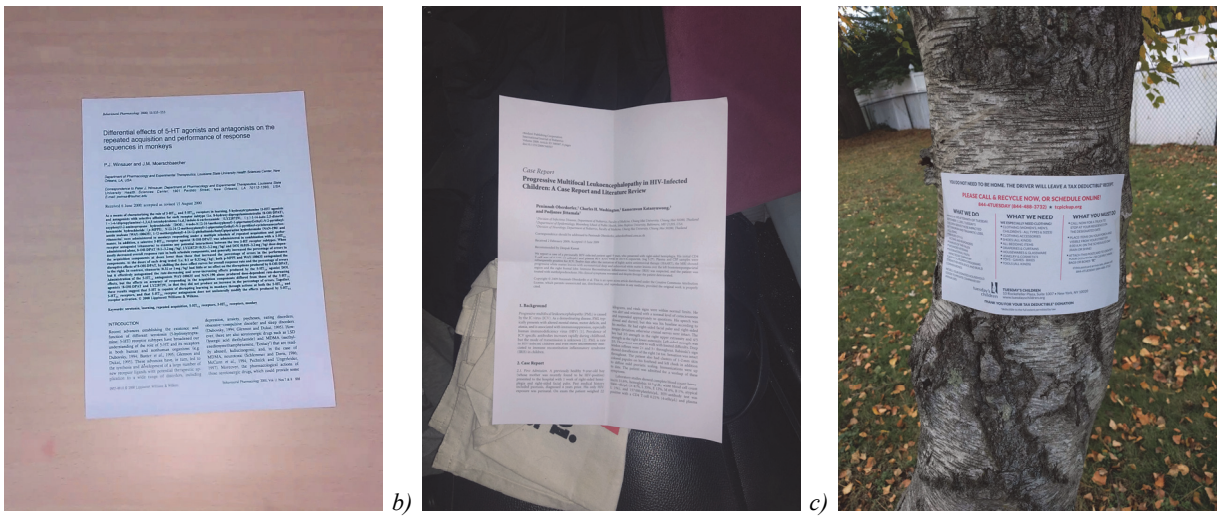
The rectification problem becomes a lot more complicated if the document has undergone some mechanical distortions. A paper sheet being a medium of the document can be easily obscured. For example, it may appear with creases, curls, or crumples. These changes complicate the rectification dramatically – both in terms of techniques used and computational cost. In recent years, this problem has been mostly addressed with the usage of neural networks, that are tuned to work with the most general obscurations. Despite being reported to gradually improve the rectification accuracy, the published algorithms are still far from being as good as the previously mentioned methods for the rectification of flat documents.

Of course, the development of such general methods is a task of the highest importance, but, until these methods are accurate and fast enough, their applications are only of academic interest. In real life, no business document is crumpled or contains multiple curls. However, neatly and carefully folded documents with creases parallel to the sides of the paper sheet

are commonly used everywhere. Thus, this specific case is of interest for document recognition systems being used in real business applications. Now the question is the following: what creases appear when one takes the document out of its envelope?

Several international standards regulate the most common envelope sizes. The most used and most popular standard envelope sizes in Europe, Russia, and the United States suppose the sent letter to be folded into halves, quarters, or thirds. Namely, in Europe, the most used envelope formats C6/C5 from the ISO 216 [8] and DL from DIN 678 [9] (and the corresponding Russian GOST standard GOST R 51506-99 [10]) are suitable for standard A4 paper folded into thirds. The formats C5 and C6 from the same ISO standard are suitable for an A4 folded into halves and quarters, respectively. Similarly, the most commonly used (though not being regulated by any standardizing organization) envelope size in the USA, No. 10 "business" envelope was designed to fit the US Letter (ANSI A as defined in ANSI/ASME Y14.1 [11]) standard paper folded into thirds.

So, summing up, documents withdrawn from an envelope are most typically folded into halves, quarters, or thirds. The first and the second classes of folds are assessed in the works [12] and [13] respectively. So, the rectification of documents folded into thirds is not thoroughly studied, thus we address this problem in the present paper. We provide a new content-independent model-based method for document rectification that can be executed on a mobile phone. Sample images and their expected rectifications are presented in Fig. 2.



a) b) c)

Fig. 1. Examples of document distortions: (a) projective distortion (source: SmartDoc dataset [4]), (b) creased document (source: WarpDoc dataset [5]), (c) curved document (source: DocUNet benchmark dataset [6])

1. Related work

Despite the obvious need for practical applications, the problem of envelope rectification is not deeply studied. A recently published work [14] is dedicated to the rectification of historical letters folded into envelopes in many different ways via computer tomography. Despite the significance of this work, the problem solved in it differs a lot from the one studied here. In [13] a paper folded into quarters is considered. It is proposed to detect each quadrilateral of the paper sheet, and then utilize Coons patches [15] to rectify each quarter separately. Unfortunately, this method may lead to content tearing along the lines of concatenation, and the authors report this issue in their paper. A recently published work [12] deals with the documents folded into halves. The authors provide a contour-based method for the localization of document outer border and rectify the image with a piecewise-projective transformation.

Now let us review the methods addressing the most general problem of document image rectification. All the approaches for document image rectification can be separated into two main categories: model-based and data-driven.

1.1 Model-based approaches

All the model-based approaches can be categorized into several groups of methods based on the addressed problem. The first group utilizes the explicit 3D reconstruction of a document surface. Typically the authors obtain a dense 3D point cloud with the help of additional equipment [16 – 20], or methods of algorithmical 3D shape restoration, such as Shape-from-Motion (SfM) [21 – 23] or Shape-from-Shading (SfS) [24]. Once the spatial parameters of the document are figured out, the flattening process can be performed by physical or geometrical optimization. Since additional equipment is bulky and hard to acquire, SfM requires several images instead of one, and SfS depends heavily on illumination conditions, none of these methods appears practical in our case.

Another group of methods utilizes the outer borders of the document to approximate its deformation. Several papers [13, 25 – 27] make use of the so-called Coons patches [15] for rectification. This simple yet efficient method can come

very handy when document distortions are not severe. Unfortunately, when the depth variation of the document surface is significant, the rectified image can contain uneven distortions, which makes this class of methods limited in the considered matter.

The last subclass of model-based approaches deals with the content of the document as discussed further.

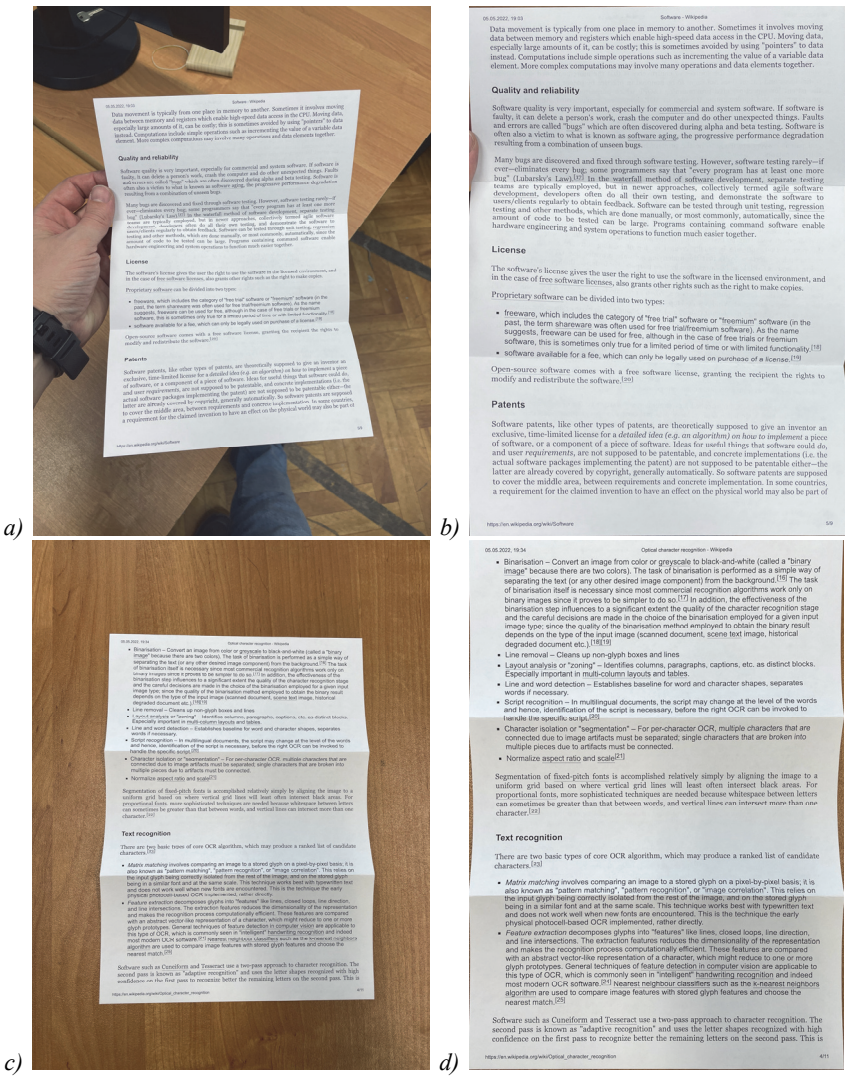


Fig. 2. Examples of (a, c) given mechanically distorted document images, (b, d) corresponding geometrical rectifications. Input images are taken from the open dataset FDI [12]

One way of considering it is to rectify the document content itself. For example, to rectify it word-by-word (or use other textline atomization) as proposed in [28, 29], or unwarp whole textlines as in [30 – 33]. Unfortunately, these methods heavily depend on the "textness" of the content and may fail on documents containing graphical information typical to business documents, such as graphs, images, tables, logos, stamps, etc.

Another way to use the document contents is to extract several text lines to deduce a global image transformation assuming that the document shape is cylindrical [34 – 36], developable [37] or ruled [38]. Unfortunately, these assumptions do not apply to the considered case. Documents with two folds do not produce a strict developable or even ruled surface as the creases are the points of non-smoothness. They can be considered as piecewise-developable, or piecewise-ruled, but unfortunately, none of the works mentioned in this class considers these surfaces.

1.2 Data-driven approaches

Deep learning methods in document image rectification problem were first introduced in 2018 by Ma et.al. [6]. The authors trained an end-to-end neural network consisting of two stacked UNets. They also proposed a DocUNet benchmark that has become a standard way to compare document rectification methods. The subsequent neural-network-based approaches were shown to increase the accuracy of rectification on this benchmark. For example, in [39 – 41] the authors propose to explicitly restore the 3D shape of the document. Others improve performance by introducing gated networks [42] or residual blocks in UNet architecture [43].

One of the latest improvements in data-based rectification algorithms is concerned with introducing transformer architectures. A Geometric Transformer DocTr was introduced in [44] and was shown to have a SOTA performance on the DocUNet benchmark. Future development of transformers led to the introduction of 3D data prediction [45] and template consideration [46].

Despite the fact that neural networks are widely represented as document image rectification algorithms, they are not fit for on-device rectification for several reasons. On one hand, execution of such algorithms on CPUs is very inefficient. Namely, in this paper, we conduct experiments showing that the SOTA geometric transformer DocTr, despite having a fairly small number of parameters (26.9×10^6) has a running time of 6 s on a mobile 6-core ARM Cortex-A78AE v8.2 (1.5 GHz) processor. This measurement correlates with the ones performed in [47].

On the other hand, the execution of a neural network on a GPU (Graphics Processing Unit) may be quite untrivial. Even if a smartphone is equipped with a GPU and it is available for thirdparty applications (which is not always), their deployment into an end-user application, such as a document recognition system, can suffer from some untrivial difficulties. While iOS-based mobile devices (such as iPhones and iPads) all utilize a single API (Metal API developed by Apple Inc.) to access a GPU, Android-based devices may use variable hardware acceleration libraries for neural network inference [48]. If one is willing to create a multiplatform document recognition system, the developed algorithms have to be ready to work with a wide range of different APIs, complicating the deployment of mobile versions. Moreover, as shown in [48], network inference time may vary on different Android-driven smartphone models, creating the need to optimize the recognition pipeline for every model separately.

Thus, no method allows for the rectification of documents folded into thirds, that could be executed on a CPU of a mobile device and have reasonable running time while delivering high accuracy. One way of overcoming this problem is the improvement of neural networks making it possible to run them on a mobile CPU. One can improve the inference time of a neural network by reducing the number of parameters [49] or itroducing neuron approximations [50]. Another way is to develop a model-based method for a popular real business case. In the present paper, we choose the second way and propose a novel algorithm for the rectification of documents folded into thirds. We use a recently published content-independent contour-based method Unfolder [12]. We perform several simplifications and generalizations of Unfolder to make this algorithm fit for the addressed problem.

So, the main contributions of the present work are the following:

1. We propose a simple, yet efficient algorithm for the localization and rectification of documents folded into thirds;
2. The proposed algorithm shows superior performance by key rectification metrics (AD and CER) compared to the current SOTA neural network-based approach DocTr [44];
3. The proposed algorithm can rectify a document held in hands;
4. The proposed algorithm self-assessment can detect inaccurate localizations and reject them, making it possible to combine our algorithm with the rectification network to produce a cascade self-sufficient document rectification system;
5. The proposed algorithm can be executed on the CPU of a mobile device, e.g. a smartphone, and has reasonable running time: the execution time of the proposed method on an iPhone XR is only 127 ms (17 ms for the localization step and 110 ms for the projective rectification).

2. Proposed algorithm

We follow a pipeline proposed in [12] for the rectification of documents with a single crease. Its generalization is schematically illustrated in Fig. 3. It can be easily adapted for any number of horizontal creases. The present study is concentrated only on the considered case – documents with two horizontal creases. We utilize a simple contour-based method to localize the outer border of a folded document and then apply a piecewise-projective image transformation to rectify it. Due to the usage of the continuity criterion for a piecewise-projective transformation, the proposed algorithm does not produce any content tearing on image concatenation lines.

2.1 Notation

In the text below we use the following notation. All the images, sets, and vertices are denoted with capital Latin letters. All the geometrical primitives (except for points) such as lines, segments, and polygons, are denoted with lowercase Latin letters. Subscript indices t, b, m refer to the considered object position relative to the image or other objects: subscript index t denotes "top" position, m denotes "middle" position, b denotes "bottom" position. Superscript indices h, v denote the object orientation: superscript index h denotes "horizontal" direction, v denotes "vertical" direction.

A document folded into thirds is considered to consist of three flat concatenated rectangles, forming an octagon having two "horizontal" segments and six "vertical" segments when it is captured by a camera. An example of the octagon in question is demonstrated in Fig. 3 (center image). The input image is denoted with I, the localized octagon is denoted with p, and the rectified image is denoted with R.

2.2. Localization

The document localization step consists of the following stages. At first (see Sec. 2.2.1) the document top and bottom parts are roughly approximated by quadrilaterals with the usage of quadrilateral search proposed in [51]. After that, a pair of quadrilaterals is considered to generate an octagon corresponding to a document with two crease lines (see Sec. 2.2.2). Then each octagon is ranked based on its corresponding to the image contours (see Sec. 2.2.3). In the end, the best octagon is corrected to fit the continuity criterion so that it is guaranteed not to produce content tearing (see Sec. 2.2.4). This corrected octagon is the output of the localization step.

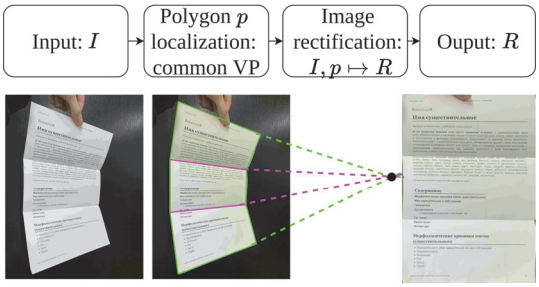


Fig. 3. The pipeline of the proposed method is presented at the top of the figure. The bottom of the figure contains the following: input image I (left), the polygon in question p having horizontal segments sharing a common vanishing point (VP) (center), and the rectification result R (right)

2.2.1. Rough quadrilateral approximation

The rough quadrilateral search begins with the extraction of image edge maps E^h, E^v the same way as proposed in [51]. The resulting edge map for a sample image is illustrated in Fig. 4*a*. After that, Fast Hough Transform [52] is employed to extract sets of straight lines. The set L^h contains lines obtained from E^h and set L^v contains lines obtained from E^v. The resulting sets of lines are illustrated in Fig. 4*b*.

Once the sets of lines are collected, we obtain a rough quadrilateral combining 4 lines: two vertical lines from L^v, one horizontal line from L^h, and the line l_m – the horizontal line splitting the image into two equal parts horizontally. We get two sets of quadrilaterals: the ones lying in the upper half of the image $Q_t = \{q_t \in I_t\}$ and the ones lying in the bottom half of the image $Q_b = \{q_b \in I_b\}$. These sets are sorted by the contour score as described in [51]. The best resulting quadrangles are depicted in Fig. 4*c*.

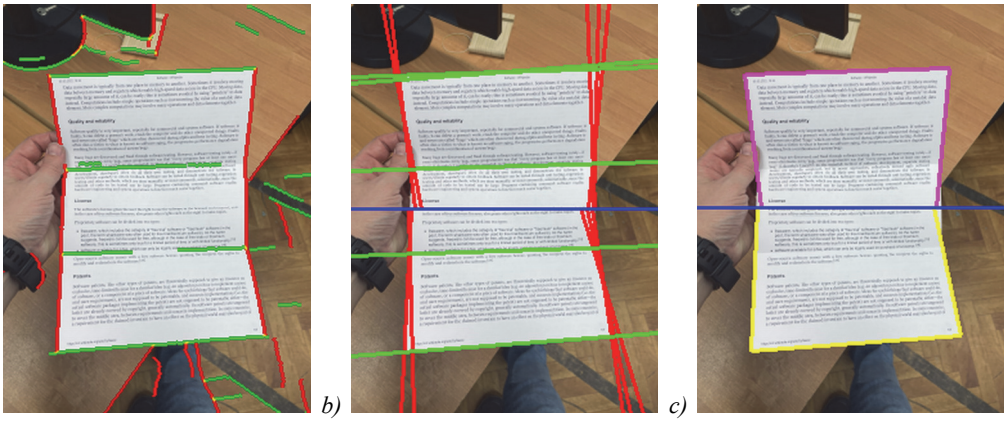


Fig. 4. Folded document localization steps: (a) merged smoothed image edge map E, green represents horizontal edges (E^h), red represents vertical edges (E^v), (b) the sets of lines L_v, L_h and a line l_m dividing image I into two equal parts, (c) top-ranked quadrilaterals, roughly approximating the top and bottom parts of the document

2.2.2. Octagon formation

Let us consider a pair of quadrilaterals q_t and q_b. The 8 points of the octagon in question are defined the following way.

Denote the upper horizontal segment of q_t as s_{tt} and the lower horizontal segment of q_b as s_{bb}. Now let us find two brightest (in terms of Hough space) distinct horizontal lines l_{tm} and l_{mb}, having l_{tm} closer to s_{tt} and l_{mb} closer to s_{bb}. All these designations are depicted in Fig. 5*a* for clarity.

Now the only thing left is to find where the actual octangle vertices lying on the lines l_{tm} and l_{mb} are. Since the vertical segments of q_t and q_b do not necessarily correspond to the actual document borders, for each of them the longest path graphs from the edge map lying along them are retrieved (red path on Fig. 5*b*) as described in [12]. These path graphs are intersected with the corresponding crease lines to obtain the crease vertices. If there is no such an intersection, the lines are intersected with the vertical segments to obtain the vertices in question.

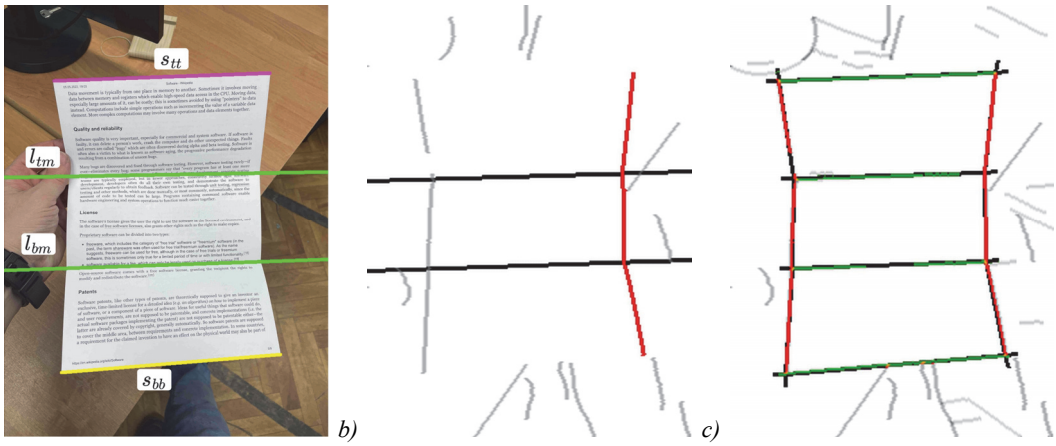


Fig. 5. Folded document localization steps: (a) the crease lines between s_{tt} and s_{bb} (b) the crease lines intersection with the left document edge, (c) edges lying along the detected octagon p

2.2.3. Octagon scoring and ranking

After the octagon p was constructed, its "brightness" is used to compare it with the other octagons to choose the one with the highest score. It is computed the following way. The edge maps are subjected to the Gaussian smoothing. The score considers ten segments: the sides of the octagon and two crease lines. For every such a segment, the following values are calculated: the sum of pixels under the segment p_k, the number of zero-pixels under the segment r_k, and the penalty q_k, where k is some arbitrary index for a segment. The penalty q_k is calculated as the sum of pixels under the segment extension by 10 px in each direction for horizontal segments and in the direction opposite to the crease lines for upper and lower vertical segments. The number of zero-pixels below each segment is summed and divided by the total length of all segments l, resulting in the ratio $r = \sum r_k/l$. Then the contour score of the octagon is the value

$$S(p) = \frac{\sum p_k}{r+1} - \sum q_k. \quad (1)$$

The scoring procedure is depicted in Fig. 5*c*. The figure represents the edge map E with the highlighted regions considered in $S(p)$. The color coding on this illustration is the following. White- and gray-colored pixels are out of consideration. Black-colored pixels represent the absence of an edge, the colored pixels represent its presence.

2.2.4. Octagon correction to fit the continuity criterion

Since we are willing for the final rectification to produce no content tearing, we correct an octagon corresponding to the document outer borders so that it fits the continuity criterion. As it was stated in [12], if one supposes that a document in the image consists of several flat concatenated rectangles with each of them undergoing a perspective transformation during the image capture, then for two quadrilaterals to produce a continuous piecewise-projective transformation, one has to follow the continuity criterion proposed in that paper. We generalize this criterion for the documents consisting of three parts. Turns out that a piecewise-projective transform in our case is continuous if and only if all the horizontal segments of our octagon share a common vanishing point. Due to the inaccuracies in the localization process, the horizontal segments might not satisfy this condition. To make the octagon fit for it, we utilize the algorithm described in [53]. It finds a common vanishing point for the 4 considered horizontal segments. Once the new vanishing point has been established, these segments are rotated and intersected with the vertical ones to create a corrected octagon $\tilde{p}$. We shall drop the tilde for the sake of simplicity of further narration and shall write simply p instead of $\tilde{p}$.

2.2.5. Octagon filtration

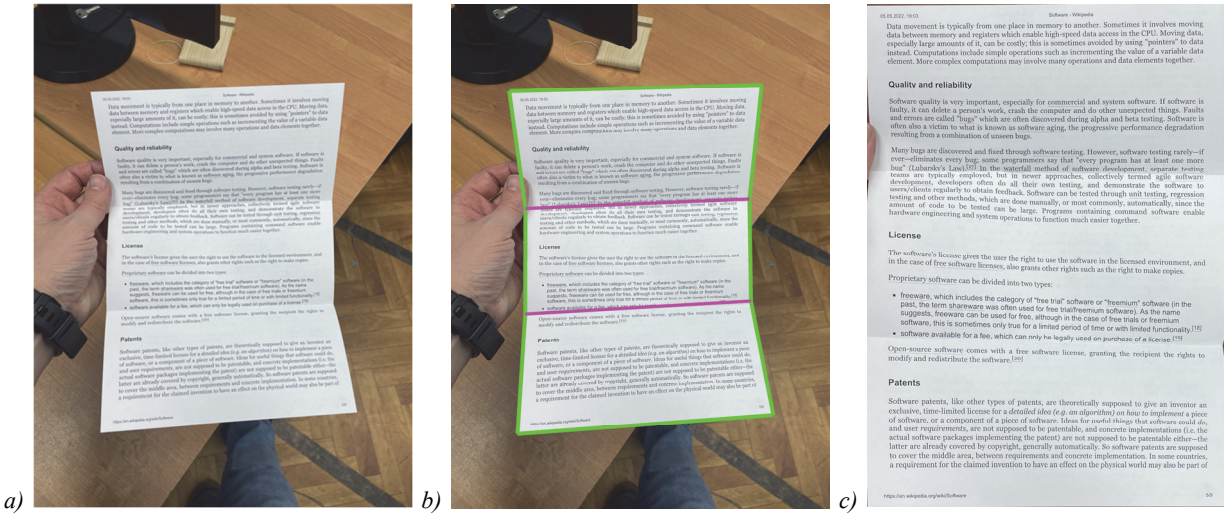
The final stage of document localization is the validation of the detected octagon p. Namely, we compute the aspect ratios for each of the three concatenated quadrilaterals with the usage of an algorithm proposed in [7] and if one of them exceeds the true value of 99/210 (A4 folded into thirds) by 30 %, then the localization is rejected and the output of the localization step is empty.

2.3. Rectification

If the localization algorithm returned no octangle, then we simply return the input image as the rectification result. Otherwise, we do the following.

The detected polygon being the result of the localization step 2.2 defines three concatenated quadrilaterals, each representing a flat third of the physical paper sheet. Each of these thirds is projectively transformed to one third of the rectangle with the same aspect ratio as the printed document (i.e. 210/297 mms for A4 paper or 8.5/11 inches for US

Letter). The rectified image is the output of the proposed algorithm. The rectification result for a localized octangle is presented in Fig. 6.



a) *b)* *c)*

Fig. 6. Proposed algorithm results: (a) input image; (b) localization step, the resulting octagon $\mathbf{p}$ is depicted with green, crease lines are depicted with magenta; (c) rectification result

In the present study, we suppose the document aspect ratio to be known, however, it is also possible to figure out the aspect ratio using estimation based on a pinhole camera model [7].

3. Experiments

3.1. Dataset

Document image rectification problem was addressed in many publications, and there are plenty of datasets available. The DocUNet benchmark [6] has been widely used to compare the approaches to solving the task of arbitrary distortions. WarpDoc dataset presented in [5] contains about a thousand images with different document distortions, and it is suitable to assess a similar problem under various conditions. Both datasets combined contain only three images of the target case. Such datasets as DIR300 [45], DIW [54], Inv3d [46] and FDI-AC [47] either do not contain enough images of documents folded for envelopes or do not contain such images at all. So, the only dataset known to the authors containing enough images of documents folded into thirds is the subset *3fold* of the dataset FDI [12].

In this study, we make use of it and measure the accuracy of our algorithm on it. The dataset contains two types of scenes: when the document is held in hand (see Fig. 7*a*, *b* and Fig. 8*a*) and when the document is placed on the table (see Fig. 7*c*, *d* and Fig. 8*e*). We test our algorithm on both of these subsets.

3.2. Metrics

In the present paper we follow the proposed in [6] way to measure and compare the accuracy of image rectification algorithms. Namely, a rectified image is compared with the original grayscaled image that was printed out with the following standard metrics: multiscale structural similarity index (SS = 1 – MS SSIM, [55]), local distortion (LD, [22]), absolute Levenstein distance between a string extracted from the input image and reference image, also called edit distance (ED, [56]) and relative levenstein distance, or character error rate (CER), which is ED normalized by the recognized string length. We also make use of the recently published aligned distortion (AD, [54]) metric. It was proposed to overcome the disadvantages of the LD metric. Namely, the LD metric sometimes tends to produce high values in the areas where the document has no content. Thus, AD reweights the LD so that the content distortion is more significant than the distortion of blank areas.

In the present paper, all these metrics are evaluated to compare the rectification algorithms. However, in our opinion, one should prefer AD and CER to the other metrics, as the former is shown [54] to be more robust and representative than MS SSIM and LD, and the latter can provide insights on the whole document recognition pipeline.

3.3 Rectification accuracy

We conduct 4 experiments on the dataset FDI/*3fold*. At first, we measure the accuracy across all the metrics for the raw (original) FDI images. These results are presented in the first row of the Tab. 1. Secondly, we measure the accuracy of the proposed algorithm and the baseline SOTA network GeoTr [44], the results are shown in the second and the third rows respectively. The illustration of the rectified images from FDI/*3fold*/*hand* and FDI/*3fold*/*table* are presented in Fig. 8.

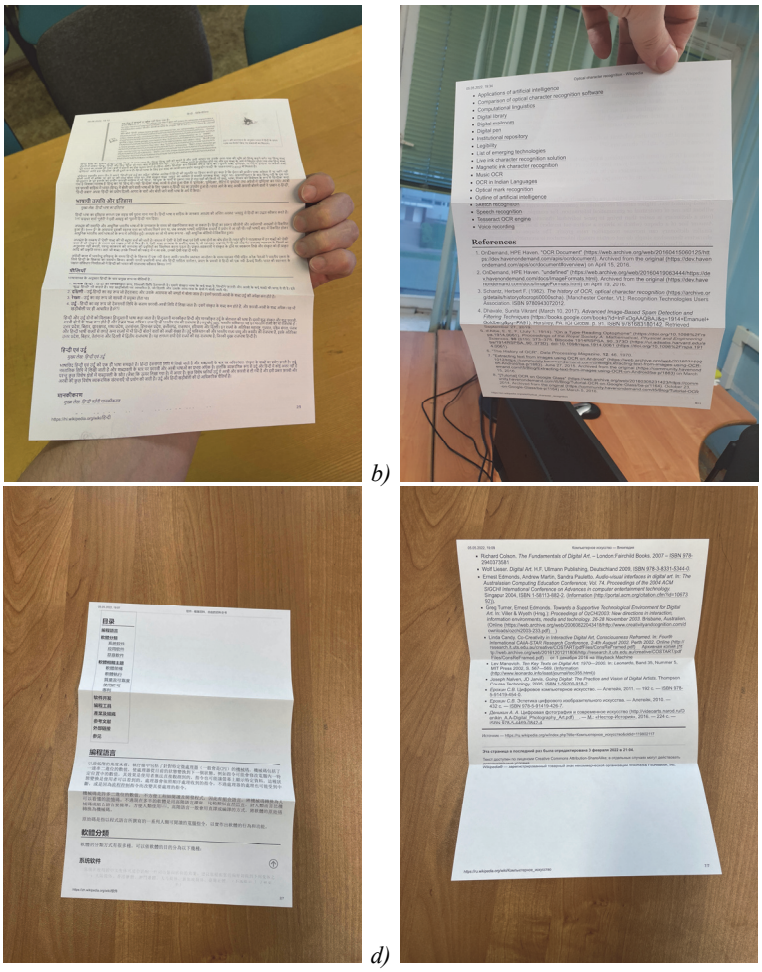


Fig. 7. Sample images from FDI [12]: (a, b) hand subset, (b, d) table subset

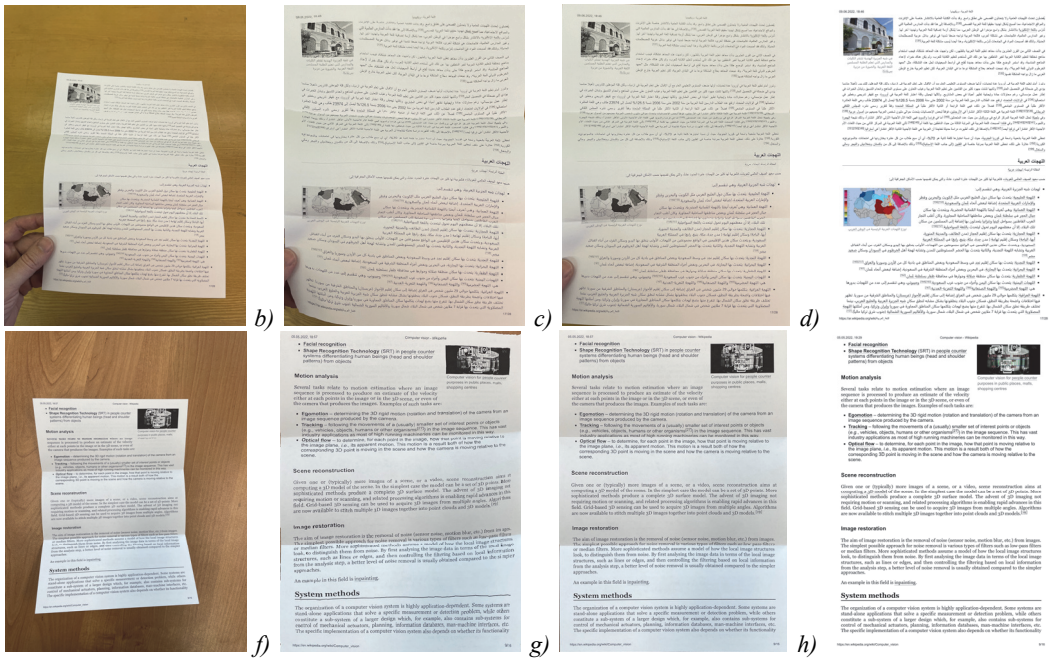


Fig. 8. Rectification results on FDI dataset, hand (a-d) and table (e-h): (a, e) input image, (b, f) result by GeoTr, (c, g) proposed method result, (d, h) reference image. Note that the accuracy measurements considered the grayscaled versions of all the images

We also make use of the self-assessment of the proposed system in the following way. Our algorithm can decide if the localized polygon does not correspond to the document outer borders or if it does not fit the model of a document folded into thirds, and the output image in this case is simply the input image (see Sec. 2.2.5, 2.3). So, this self-assessment

makes it possible to introduce a cascade-like scheme: if the proposed algorithm returned a valid localization, we use the proposed rectification (see Sec. 2.3), otherwise, we use the rectification of GeoTr. This cascade system is self-sufficient since it returns a rectified image no matter what. We test it together with other algorithms, and its results are presented in the fourth row in Tab. 1.

*Tab. 1. Rectification accuracy on FDI/3fold dataset. The best value in each column is highlighted with underline, and the best value in the first 3 rows is highlighted with **bold***

#	System	Subset *hand*				
		SS↓	LD↓	AD↓	ED↓	CER↓
1	Raw (no algo)	0.72	52.57	1.12	1595	0.60
2	Proposed	**0.61**	**24.30**	**0.80**	**1422**	**0.53**
3	GeoTr	0.63	24.82	0.85	1653	0.62
4	Proposed+GeoTr	0.57	15.35	0.58	1434	0.54
#	System	Subset *table*				
		SS↓	LD↓	AD↓	ED↓	CER↓
1	Raw (no algo)	0.75	49.33	1.30	1187	0.46
2	Proposed	**0.53**	16.34	**0.34**	**1046**	**0.41**
3	GeoTr	0.54	**11.48**	0.36	1209	0.48
4	Proposed+GeoTr	0.50	7.81	0.20	1036	0.41

Let us consider Tab. 1 in detail. At first, one can notice that the proposed algorithm shows performance superior to GeoTr across all measurements except for LD metric on *table* subset. It is also of interest that judging by OCR metrics GeoTr shows results that are worse than the ones for raw FDI images even though its values for SSIM, LD, and AD are quite good, especially on *table* subset. It means that embedding this system into a document recognition system "as-is" will not improve the accuracy of the whole recognition pipeline: it will only consume time to corrupt the input image for the upcoming OCR system. Nevertheless, the combined cascade system involving GeoTr as its part shows performance comparable with the ones of the proposed algorithm, particularly reaching its peak accuracy in measurements by AD metric. This fact allows us to make a conclusion that the errors of the proposed method and GeoTr are uncorrelated, and such a cascade can hypothetically be utilized in applications.

3.4. Proposed algorithm AD values distribution

Let us perform some further analysis of the proposed algorithm in terms of AD metric. Based on the data from Tab. 1 one can estimate the average AD value for the proposed algorithm on full FDI/*3fold*: $\mathrm{AD}_{ave} = (0.8 + 0.34)/2 = 0.57$. Since the proposed algorithm sometimes rejects the localization result and returns the input image as its result, it is of interest to analyze, what is the distribution of the values across all the images in the FDI dataset depending on whether the localization result is rejected or not. Such a distribution is presented as a histogram in Fig. 9. The blue subgraph shows the distribution for the non-rejected localizations, and the orange subgraph illustrates the rejected ones. The histogram depicts the number of images having an AD value in a certain range: 1 range – one histogram bin and the rightmost bin for each color represent the range $[1.35; \infty)$. A blue dashed line shows the mean value for all non-rejected localizations (0.26), a red dashed line shows the mean value for all rejected localization values (1.31) and a magenta dashed line shows the mean value across all the images ($\mathrm{AD}_{ave} = 0.57$). It can be clearly seen that the non-rejected values tend to have a low value of AD, and most of these images have AD values less than 0.2, while the distribution of the rejected values is concentrated in the right part of the graph.

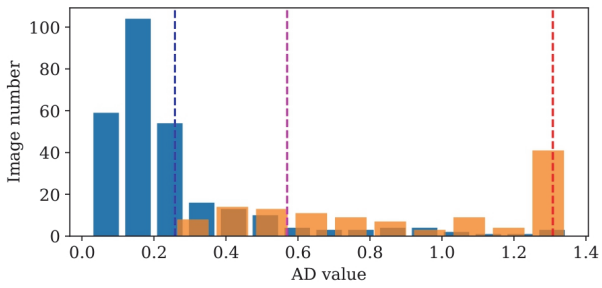


Fig. 9. The histogram of AD values for rejected (orange) and non-rejected (blue) rectifications. The blue dashed line is the mean value for non-rejected answers, the red one for rejected answers, and the magenta one is the mean for the whole dataset

3.5. Runtime measurements

We conduct runtime measurements of the two algorithms: the proposed one and GeoTr. Two devices were employed for this task: Jetson Orin Nano equipped with a 6-core ARM Cortex-A78AE v8.2 (1.5 GHz) processor and iPhone XR

with Apple A12 Bionic, which is a 6-core ARM-based processor (max. 2.49 GHz). The algorithms were allowed to use only a CPU with all its cores available at execution time. Note that our implementation of the proposed algorithm used parallelization only in standard image processing operations, such as edge extraction, Hough transform, projective transform, etc. Mean execution time in milliseconds is illustrated in Tab. 2. As it can be seen, the proposed method allows for the better execution time than DocTr, with the localization step 60 times faster, and the normalization time 3 times faster than DocTr. Moreover, our approach takes only 137 ms when being run on iPhone XR.

Tab. 2: Mean execution time per image in milliseconds. The asterisk (*) indicates the average time on a subset of the dataset with a non-rejected localization result

	Device	Localization	Normalization
Proposed	Jetson	**87**	**199***
GeoTr	Jetson	5415	592
Proposed	iPhone XR	17	110*

Conclusion

The paper presents a novel model-based method for the localization and rectification of documents folded into thirds. This folding type is very common in business cases since it appears when a document is taken out of a standard-sized envelope. The proposed method allows to localize a document with two creases in a camera-captured image. This localization is used to projectively rectify the document to flatten its geometrical shape. Our method outperforms the current SOTA rectification algorithm DocTr by key rectification metrics (AD, CER) on the FDI/*3fold* dataset. The proposed algorithm can rectify a document even if it is held in hand. The proposed algorithm self-assessment can detect bad localizations and reject them, making it possible to combine our algorithm with the rectification network DocTr to produce a cascade-like self-sufficient document rectification system. It is also shown that the algorithm can be run on the CPU of a mobile device: the execution time on an iPhone XR is only 127 ms (17 ms for localization and 110 ms for projective rectification), making it possible to utilize it in an on-device recognition system.

References

[1] Liang J, Doermann D, Li H. Camera-based analysis of text and documents: a survey. International Journal of Document Analysis and Recognition (IJDAR) 2005; 7: 84-104. DOI: 10.1007/s10032-004-0138-z.
[2] Slavin OA, Arlazarov VL. Method for classifying recognized pages of administrative documents on the basis of text key points. Trudy ISA RAN (Proceedings of ISA RAS) 2018; 68 (S1): 32-42. DOI: 10.14357/20790279180504.
[3] Doermann D, Liang J, Li H. Progress in camera-based document image analysis. In: Seventh International Conference on Document Analysis and Recognition (ICDAR2003): 606-616. IEEE(2003). DOI: 10.1109/ICDAR.2003.1227735.
[4] Burie J, Chazalon J, Coustaty M, Eskenazi S, Luqman MM, Mehri M, Nayef N, Ogier J, Prum S, Rusiñol M. ICDAR2015 competition on smartphone document capture and OCR (SmartDoc). In: 13th International Conference on Document Analysis and Recognition (ICDAR2015): 1161-1165. IEEE(2015). DOI: 10.1109/ICDAR.2015.7333943.
[5] Xue C, Tian Z, Zhan F, Lu S, Bai S. Fourier Document Restoration for Robust Document Dewarping and Recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2022): 4573–4582. IEEE(2022). DOI: 10.1109/CVPR52688.2022.00453.
[6] Ma K, Shu Z, Bai X, Wang J, Samaras D. Docunet: Document image unwarping via a stacked u-net. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR2018): 4700–4709. IEEE(2018). DOI: 10.1109/CVPR.2018.00494.
[7] Zhang Z, He L. Whiteboard scanning and image enhancement. Digital signal processing 2007; 17 (2): 414–432. DOI: 10.1016/j.dsp.2006.05.006.
[8] ISO 216:2007. Writing paper and certain classes of printed matter — Trimmed sizes — A and B series, and indication of machine direction. Geneva, Switzerland: ISO; 2007
[9] DIN 678-1-1998, Envelopes - Part 1: Sizes. Berlin, Germany: German institute for standardization; 1998
[10] GOST R 51506-99, Post envelopes. Technical requirements. Control methods [In Russian]. Moscow, Russia: Federal Agency for Technical Regulation and Metrology of the Russian Federation; 1999
[11] ASME Y14.1-2020, Drawing Sheet Size and Format. New York City, U.S.: American Society of Mechanical Engineers; 2020
[12] Ershov AM, Tropin DV, Limonova EE, Nikolaev DP, Arlazarov VV. Unfolder: Fast localization and image rectification of a document with a crease from folding in half. Computer Optics 2024; 48 (4): 542–553. DOI: 10.18287/2412-6179-CO-1406.
[13] Das S, Mishra G, Sudharshana A, Shilkrot R. The common fold: utilizing the four-fold to dewarp printed documents from a single image. In: Proceedings of the 2017 ACM Symposium on Document Engineering: 125–128. ACM(2017). DOI: 10.1145/3103010.3121030.
[14] Dambrogio J, Ghassaei A, Smith DS, Jackson H, Demaine ML, Davis G, Mills D, Ahrendt R, Akkerman N, Van der Linden D and others. Unlocking history through automated virtual unfolding of sealed documents imaged by X-ray microtomography. Nature communications 2021; 12 (1): 1184. DOI: 10.1038/s41467-021-21326-w.
[15] Coons SA. Surfaces for computer-aided design of space forms. 1967; (MIT-LCS-TR-041MAC-TR-041).
[16] Brown MS, Seales WB. Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents. In: Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV2001): 367–374. IEEE(2001). DOI: 10.1109/ICCV.2001.937649.

[17] Brown MS, Seales WB. Image restoration of arbitrarily warped documents. IEEE Transactions on pattern analysis and machine intelligence 2004; 26 (10): 1295–1306. DOI: 10.1109/TPAMI.2004.87.
[18] Chua KB, Zhang L, Zhang Y, Tan CL. A fast and stable approach for restoration of warped document images. In: Eighth International Conference on Document Analysis and Recognition (ICDAR2005): 384–388. IEEE(2005). DOI: 10.1109/ICDAR.2005.8.
[19] Sun M, Yang R, Yun L, Landon G, Seales B, Brown MS. Geometric and photometric restoration of distorted documents. In: Tenth IEEE International Conference on Computer Vision (ICCV2005): 1117–1123. IEEE(2005). DOI: 10.1109/ICCV.2005.106.
[20] Meng G, Wang Y, Qu S, Xiang S, Pan C. Active flattening of curved document images via two structured beams. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2014): 3890–3897. IEEE(2014). DOI: 10.1109/CVPR.2014.497.
[21] Perriollat M, Bartoli A. A computational model of bounded developable surfaces with application to image-based three-dimensional reconstruction. Computer Animation and Virtual Worlds 2013; 24 (5): 459–476. DOI: 10.1002/cav.1478.
[22] You S, Matsushita Y, Sinha S, Bou Y, Ikeuchi K. Multiview Rectification of Folded Documents. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018; 40 (2): 505-511. DOI: 10.1109/TPAMI.2017.2675980.
[23] Luo D, Bo P. Geometric Rectification of Creased Document Images based on Isometric Mapping. arXiv preprint arXiv:2212.08365 (2022).
[24] Zhang L, Yip AM, Brown MS, Tan CL. A unified framework for document restoration using inpainting and shape-from-shading. Pattern Recognition 2009; 42 (11): 2961–2978. DOI: 10.1016/j.patcog.2009.03.025.
[25] Brown MS, Tsoi Y. Geometric and shading correction for images of printed materials using boundary. IEEE Transactions on Image Processing 2006; 15 (6): 1544–1554. DOI: 10.1109/tip.2006.871082.
[26] Tsoi Y, Brown MS. Multi-view document rectification using boundary. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR2007): 1–8. IEEE(2007). DOI: 10.1109/CVPR.2007.383251.
[27] Koo HI, Cho NI. Rectification of figures and photos in document images using bounding box interface. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2010): 3121–3128. IEEE(2010). DOI: 10.1109/CVPR.2010.5540071.
[28] Gatos B, Pratikakis I, Ntirogiannis K. Segmentation based recovery of arbitrarily warped document images. In: Ninth international conference on document analysis and recognition (ICDAR2007): 989–993. IEEE(2007). DOI: 10.1109/ICDAR.2007.4377063.
[29] Lu S, Tan CL. The restoration of camera documents through image segmentation. In: Document Analysis Systems VII: 7th International Workshop (DAS2006): 484–495. Springer(2006). DOI: 10.1007/11669487_43.
[30] Ezaki H, Uchida S, Asano A, Sakoe H. Dewarping of document image by global optimization. In: Eighth International Conference on Document Analysis and Recognition (ICDAR2005): 302–306. IEEE(2005). DOI: 10.1109/ICDAR.2005.87.
[31] Stamatopoulos N, Gatos B, Pratikakis I, Perantonis SJ. A two-step dewarping of camera document images. In: 2008 The Eighth IAPR International Workshop on Document Analysis Systems: 209–216. IEEE(2008). DOI: 10.1109/DAS.2008.40.
[32] Zhang Z, Tan CL. Correcting document image warping based on regression of curved text lines. In: Seventh International Conference on Document Analysis and Recognition (ICDAR2003): 589–593. IEEE(2003). DOI: 10.1109/ICDAR.2003.1227732.
[33] Ulges A, Lampert CH, Breuel TM. Document image dewarping using robust estimation of curled text lines. In: Eighth International Conference on Document Analysis and Recognition (ICDAR2005): 1001–1005. IEEE(2005). DOI: 10.1109/ICDAR.2005.90.
[34] Gaofeng M, Chunhong P, Shiming X, Jiangyong D, Nanning Z. Metric Rectification of Curved Document Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 2012; 34 (4): 707-722. DOI: 10.1109/TPAMI.2011.151.
[35] Liang J, DeMenthon D, Doermann D. Geometric rectification of camera-captured document images. IEEE transactions on pattern analysis and machine intelligence 2008; 30 (4): 591–605. DOI: 10.1109/TPAMI.2007.70724.
[36] Meng G, Su Y, Wu Y, Xiang S, Pan C. Exploiting vector fields for geometric rectification of distorted document images. In: Proceedings of the European Conference on Computer Vision (ECCV2018): 172–187. Springer(2018). DOI: 10.1007/978-3-030-01270-0_11 .
[37] Liang J, DeMenthon D, Doermann D. Flattening curved documents in images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2005): 338–345. IEEE(2005). DOI: 10.1109/CVPR.2005.163.
[38] Fujimoto K, Sun J, Takebe H, Suwa M, Naoi S. Curved paper rectification for digital camera document images by shape from parallel geodesics using continuous dynamic programming. In: Ninth International Conference on Document Analysis and Recognition (ICDAR2007): 267–271. IEEE(2007). DOI: 10.1109/ICDAR.2007.4378717.
[39] Das S, Ma K, Shu Z, Samaras D, Shilkrot R. DewarpNet: Single-Image Document Unwarping With Stacked 3D and 2D Regression Networks. In: Proceedings of International Conference on Computer Vision (ICCV2019): 131–140. (2019). DOI: 10.1109/ICCV.2019.00022.
[40] Markovitz A, Lavi I, Perel O, Mazor S, Litman R. Can you read me now? content aware rectification using angle supervision. In: Proceedings of the European Conference on Computer Vision (ECCV2020): 208–223. Springer(2020). DOI: 10.1007/978-3-030-58610-2_13.
[41] Xu Z, Yin F, Yang P, Liu C. Document Image Rectification in Complex Scene Using Stacked Siamese Networks. In: 26th International Conference on Pattern Recognition (ICPR2022): 1550–1556. IEEE(2022). DOI: 10.1109/ICPR56361.2022.9956331.
[42] Bandyopadhyay H, Dasgupta T, Das N, Nasipuri M. A gated and bifurcated stacked u-net module for document image dewarping. In: 25th International Conference on Pattern Recognition (ICPR2020): 10548–10554. IEEE(2021). DOI: 10.1109/ICPR48806.2021.9413001.
[43] Li X, Zhang B, Liao J, Sander PV. Document rectification and illumination correction using a patch-based CNN. ACM Transactions on Graphics (TOG) 2019; 38 (6): 1–11. DOI: 10.1145/3355089.3356563.
[44] Feng H, Wang Y, Zhou W, Deng J, Li H. DocTr: Document Image Transformer for Geometric Unwarping and Illumination Correction. In: Proceedings of the 29th ACM International Conference on Multimedia: 273–281. ACM(2021). DOI: 10.1145/3474085.3475388.

[45] Feng H, Zhou W, Deng J, Wang Y, Li H. Geometric Representation Learning for Document Image Rectification. In: European Conference on Computer Vision (ECCV2022): 475–492. Springer(2022). DOI: 10.1007/978-3-031-19836-6_27.
[46] Hertlein F, Naumann A, Philipp P. Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping. International Journal on Document Analysis and Recognition (IJDAR) 2023; 26 (3): 175–186. DOI: 10.1007/s10032-023-00434-x.
[47] Ershov A, Tropin D, Kazimirov D, Bulatov K, Nikolaev D. Utilizing a Two Planes Model to Rectify Documents With a Single Arbitrary Crease. IEEE Access 2024; 12 (): 147073–147086. DOI: 10.1109/ACCESS.2024.3474099.
[48] Ignatov A, Timofte R, Kulik A, Yang S, Wang K, Baum F, Wu M, Xu L, Van Gool L. Ai benchmark: All about deep learning on smartphones in 2019. In: IEEE/CVF International Conference on Computer Vision Workshop (ICCVW2019): 3617–3635. IEEE(2019). DOI: 10.1109/ICCVW.2019.00447.
[49] Limonova E, Sheshkus A, Ivanova A, Nikolaev D. Convolutional neural network structure transformations for complexity reduction and speed improvement. In: : 24–33. Springer(2018). DOI: 10.1134/S105466181801011X.
[50] Limonova EE. Fast and gate-efficient approximated activations for bipolar morphological neural networks. Information Technologies and Computation Systems 2022; (2): 3–10. DOI: 10.14357/20718632220201.
[51] Tropin DV, Ershov AM, Nikolaev DP, Arlazarov VV. Advanced Hough-based method for on-device document localization. Computer Optics 2021; 45 (5): 702–712. DOI: 10.18287/2412-6179-CO-895.
[52] Brady ML. A fast discrete approximation algorithm for the Radon transform. SIAM Journal on Computing 1998; 27 (1): 107–119. DOI: 10.1137/S0097539793256673.
[53] Shemiakina J, Konovalenko I, Tropin D, Faradjev I. Fast projective image rectification for planar objects with Manhattan structure. In: Twelfth International Conference on Machine Vision (ICMV 2019): 450–458. SPIE(2020). DOI: 10.1117/12.2559630.
[54] Ma K, Das S, Shu Z, Samaras D. Learning from documents in the wild to improve document unwarping. In: ACM SIGGRAPH 2022 Conference Proceedings (SIGGRAPH2022): 1–9. ACM(2022). DOI: 10.1145/3528233.3530756.
[55] Wang Z, Simoncelli EP, Bovik AC. Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003: 1398–1402. IEEE(2003). DOI: 10.1109/ACSSC.2003.1292216.
[56] Levenshtein VI and others. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady: 707–710. (1966).

Authors' information

Aleksandr Ershov (b. 1999) is a Ph.D student in the Institute for Information Transmission Problems of RAS. He graduated from Lomonosov Moscow State University in 2023 specializing in fundamental mathematics and mechanics. He has been a developer at Smart Engines Service LLC since 2020. Since 2025 he has been working in the Institute for Information Transmission Problems of RAS as a junior researcher in a research group in the field of computer vision and X-ray tomography. His research interests are image processing and computer vision.

Daniil Tropin (b. 1995) received the Bachelor degree in applied mathematics from the National University of Science and Technology ''MISiS'', Moscow, Russia, in 2017, the Master degree in applied mathematics and informatics from Moscow Institute of Physics and Technology (National Research University), Moscow, Russia, in 2019, and the Ph.D. degree in Computer Science in 2022. Since 2019, he has been with the Federal Research Center ''Computer Science and Control,'' Russian Academy of Sciences. Since 2016, he has been with Smart Engines Service LLC, Moscow. He has authored more than 20 scientific publications. His research interests are computer vision, image processing, and document recognition systems.

Dmitry Nikolaev (b. 1978) was awarded a Doctor of Sciences in Computer Science in 2023. Since 2007, he has been the Head of the Vision Systems Laboratory, Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, and he has been the CTO of Smart Engines Service LLC, Moscow, since 2016. Since 2016, he has been an Associate Professor with the Moscow Institute of Physics and Technology (State University), Moscow, teaching the Image Processing and Analysis Course. Now he is the Head of the Vision Systems Laboratory at Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences. He has authored over 150 Scopus-indexed publications and 15 patents. His research interests encompass computer vision and color image understanding.

Received June 16, 2025. The final version – September 01, 2025.