Segments Graph-Based Approach for Document Capture in a Smartphone Video Stream

Alexander Zhukovsky[1], Dmitriy Nikolaev[2], Vladimir Arlazarov[3], Vasiliy Postnikov[3], Dmitriy Polevoy[3,4], Natalya Skoryukina[4], Timofey Chernov[4], Julia Shemiakina[3], Arseniy Mukovozov[4], Ivan Konovalenko[1,2], Mikhail Povolotsky[1,2]

[1]Moscow Institute for Physics and Technologies
Moscow, Russian Federation

[2]Institute for Information Transmission Problems of Russian Academy of Sciences
Moscow, Russian Federation

[3]Federal Research Centre "Informatics and Control Systems" The Institute for Systems Analysis of Russian Academy of Sciences
Moscow, Russian Federation

[4]National University of Science and Technology MISIS
Moscow, Russian Federation

***Abstract* — the paper is devoted to the analysis of the problem of document boundaries detection in images and in a video stream. The paper proposes an algorithm for obtaining the position of the document, consisting of very reliable segments of a document boundaries extraction and a construction of an intersection graph that satisfies the projective model of the rectangle. An online algorithm for selecting and integrating possible document positions in a video stream based on the Kalman filter is proposed. The analysis of possible modifications of the algorithm and their effect on the final result are provided. Evaluation of the quality of the document at ICDAR'15 Smartphone Document Capture competition's dataset [1] showed a mean result of 95.5% in Jaccard index of projectively corrected document quadrangles and a 3rd place in the competition.**

Keywords—Smartphone Document Capture Competition, SmartDoc, document detection, segments detection, projective transform, mobile cameras, Kalman filter

I. INTRODUCTION

Electronic document management is increasingly coming into our lives, replacing paperwork. However, paper documents are still very often present in our lives. Most of them can be converted to an electronic form: editable, more human-readable and searchable. Not so long ago, the standard de facto for the entry point was a stationary scanner, which allows digitizing a paper version with the proper quality. But, after the emergence of smartphones, their evolution to the current state and the trend towards global mobilization, more tasks for digitizing and recognizing documents have been transferred to smartphones or specialized mobile devices [2][3][4]. However, this process requires updating of existing algorithms and developing new ones for working with documents and creating image databases for their evaluation.

In 2015 ICDAR organized the world's first competition for the development of such algorithms (ICDAR'15 Smartphone Document Capture and OCR Competition [1]) and opened an access to the dataset for quality evaluation. It was aimed at evaluating two steps in the process of digitizing images of documents received from the camera in natural conditions: document capture and text recognition. In this paper we consider only the first part of the competition, dedicated to the detection and segmentation of a document page obtained from a mobile device.

Typically, such problems are solving with using the Canny edge detector and Hough transform [5] to detect straight lines (or detecting segments using LSD [6]) and sorting out all possible quadrangles from given lines, taking into account some planar geometric constraints. This scheme is often modified and expanded. For example, one can use trained filters to obtain an edges map [7], interpolation of contours with the Bezier lines, or use of color and contrast features for the selection of the final quadrant. Alternatively, one can use the geodesic object proposals [9] or the tree of shapes [10]. We in this paper investigate how to assemble common algorithms into a full-fledged algorithm, how they interact to solve the task of document capture from a smartphone, and which moments play a key role in the development of such algorithms.

The paper is structured as follows: in section II we describe the competition and dataset structure, in section III we present a method for frame-wise document capture, in section IV we propose an algorithm for integration of results in a video stream, and in section V we discuss the results obtained on the dataset.

II. COMPETITION AND DATASET DESCRIPTION

The input data for the algorithm was 150 10-second video clips containing the image of the document. The detection algorithm had to provide the coordinates of the quadrangle of the document page on each frame of the video clip. Total dataset contained about 25K frames. Video clips were shot on a different background (totally 5 different backgrounds). The documents set were divided into six types and there were totally five documents of each type. As a mobile device, Google Nexus 7 tablet was used in video mode with a resolution of 1920x1080 and variable bit rate.

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.63

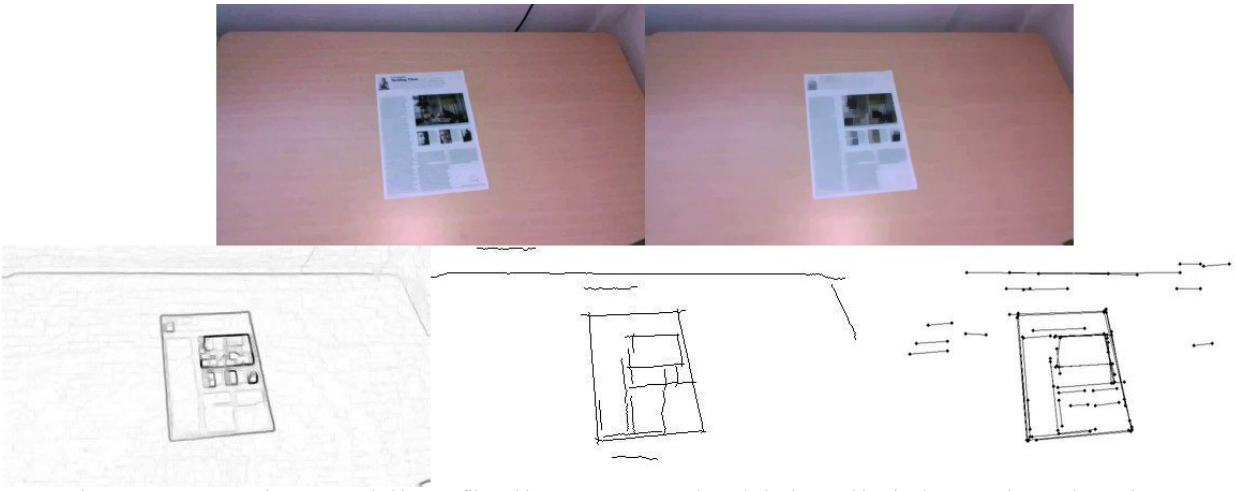

Figure 1. Image processing. Top: scaled image; filtered image. Bottom: gray boundaries image; binarized contours image; detected segments.

All 30 documents pages were a sheet of A4 paper, with straight borders and were never out of a frame. Frequent phenomena that arise when shooting in real scenarios and making it difficult to find the document were defocusing the camera and various lighting conditions, generating highlights and shadows. An important point was also the background on which the document was shot – it could differ only slightly from the document or be overloaded, which made it difficult to find the document.

To evaluate method's performance, the Jaccard index is used [10]. Using the document size and its coordinates in each frame, evaluation procedure started by transforming the coordinates of the quadrilaterals of the method *S* and of the ground-truth *G* to undo the perspective transform and to obtain the corrected quadrilaterals *S′* and *G′*. Then for each frame *f*, we compute the Jaccard index (JI) that measures the goodness of overlapping of the corrected quadrilaterals as follows:

$$JI(f) = area(G' \cap S') / area(G' \cup S') \qquad (1)$$

where $G' \cap S'$ defines the polygon resulting as the intersection of the detected and ground-truth document quadrilaterals and $G' \cup S'$ the polygon of their union. The overall score is the average of the frame score, for all the frames in the test dataset.

III. FRAME-WISE DOCUMENT DETECTION

The approach described is aimed to deal with a single frame and to do not use any information about document position (unlike [11]) or information from video. The algorithm is based on the document boundaries detection. It starts with the image preprocessing which consists of image scaling to 400x300 and morphological closing to remove small items, such as letters. Next, we simultaneously use several algorithms to detect segments on the image to increase the robustness of the entire algorithm. Then we construct the four-partite oriented graph of possible sides and use it to obtain all the possible document quadrangles. They are assigned with the weight according to the used segments and then sorted. To obtain the final quadrangle we apply projective transform restrictions to the quadrangles set and choose the number of quadrangles with largest weights. Finally, we refine the document boundaries on the original scale. Each part of the algorithm has several parameters and their proper tuning is very important.

Next we will describe all of the main parts of algorithm and the parameter tuning process.

A. Segments detection

First, we perform a step of segment detection. On a binarized contours image (obtained as in [11]) we detect a straight segments as follows. We will consider a successive set of points of a contour, where a hypothesis of contour's straightness has been admitted to be correct in each point's neighborhood. To test the hypothesis we will evaluate a covariation matrix for each point of a contour $\{(x_i, y_i)\}_{i=0}^{N}$ in the neighborhood of fixed size *l:*

$$X = \{x_i\}, Y = \{y_i\}, i = \max(0, k-l)..\min(N, k+l) \quad (2)$$

$$\Sigma(X,Y) = \begin{pmatrix} DX & cov(X,Y) \\ cov(X,Y) & DY \end{pmatrix} \qquad (3)$$

$$cov(X,Y) = M[XY] - MX * MY \qquad (4)$$

Eigenvectors and eigenvalues enable us to evaluate dimensions and a shape of distribution of points in each section of a contour by its approximation with an ellipse. Finally, we will accept the hypothesis in current section of a contour if the ratio of lengths of axes of ellipse is greater than a threshold.

Since the use of binary image alone may lead to missing some parts of the right boundary, we additionally use Hough transform for straight segment detection [5] on a gray boundaries image and Line Segment Detector [6]. To obtain a gray boundaries image, we calculate a maximum of gradient magnitude among all of image color channels, linearly stretched to [0, 255]. Segments, obtained from Hough transform and contours analysis, are used together. It is important to note that they are configured not to perform at its

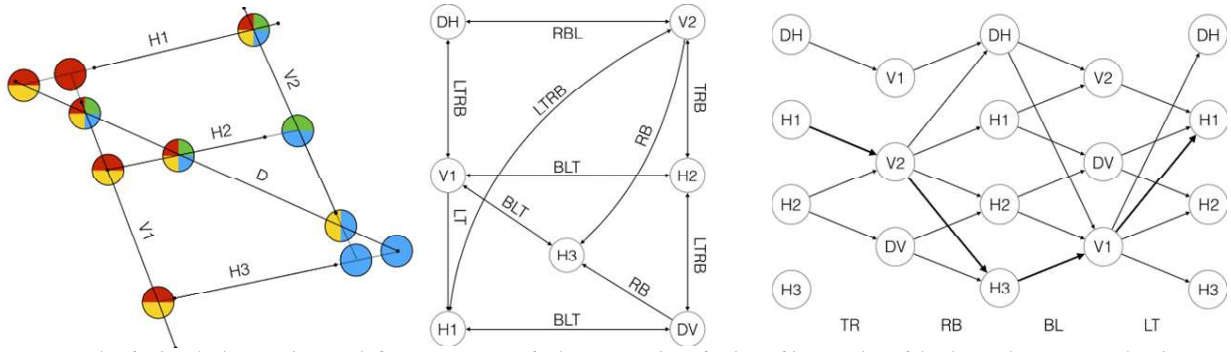

Figure 2. Example of using the intersection graph for 6 segments. Left: the construction of points of intersection of the detected segments and assignment of types; center: the intersection graph, each edge corresponds to one intersection point; right: an expanded intersection graph, each edge corresponds to the possible side of the quadrangle

best alone, but to be complementary. All the image processing is shown in the Figure 1.

Each segment is assigned with a weight that equals to its length and segments lying on the same line are merged. Two segments are considered to lie on the same line, if the maximum of the distances from the ends points of the one segment to the line containing the other segment is less than the threshold. Merged segment has its ends in the utmost ends of these segments and its weight equals to the sum of their weights. Weight sum usage enables us to reduce the merged segment weight when it is composed of two remote non-overlapping parts and to increase the weight of the segment obtained by using two algorithms simultaneously. Finally, we select a fixed number of segments largest by weight, which are used to assemble a quadrangle.

B. Quadrangle assembly

In order to construct a quadrangle on the found segments, a graph of possible sides is used. The vertices of this given graph correspond to the found segments, and the edges to the points of intersection of lines that contain these segments.

To construct a graph, all segments are divided into predominantly vertical and predominantly horizontal, depending on the angles that they make with the axes of the image. If a segment can be assigned to both classes (diagonal), then two segments are added instead: horizontal and vertical (for example, in Figure 2 on the left segment D corresponds to two vertices DH and DV on the center one). Also, each vertex is assigned with a weight w_i^{vertex} equal to the weight of the corresponding segment.

The edges of the graph are constructed as the intersections of all horizontal segments with all vertical ones. If we assume that the graph is constructed on N segments, then the entire graph contains $O(N^2)$ edges. Each edge is classified according to the type of the document angle, which can be the corresponding intersection point (left top LT, top right TR, right bottom RB, bottom left BL). The edges are classified relative to the position of the point of intersection with respect to the segments that produced it. When the intersection point of lines is the intersection point of the segments themselves, and not just the lines on which they lie, it can correspond to an arbitrary angle (LTRB), or for instance when it lies to the left of the horizontal segment and on the vertical segment, it may be the left upper or lower left (BLT). Each edge is assigned with a weight w_i^{edge} equal to the minimum distance between the ends of the segments that produced the corresponding intersection point.

The graph constructed in this way can be expanded into a four-partite directed graph, so that each edge has a definite and unique type (Figure 2, right). In such a graph every cycle consisting of four edges and not passing through the same vertices will correspond to a possible quadrangle in the image. Detection of cycles can be organized, for example, by radial detours with a maximum depth of bypass equal to four, which begin in one of the parts of the graph and can repeatedly enter one vertex. The complexity of this bypass will be $O(N^4)$, which imposes additional restrictions on the number of initial segments described at the end of section III.A.

Each quadrangle is assigned a weight equal to the difference of the sums of the weights of all edges and the sums of the weights of all vertices:

$$w^{quad} = A*\sum_{i=1}^{4} w_i^{vertex} + B*\sum_{i=1}^{4} w_i^{edge} \qquad (5)$$

Where A and B are the specified parameters.

At the end of algorithm, the required number of cycles in the graph, the largest in weight, are selected, and a quadrangle of a possible document is constructed for each of them.

C. Rectangular projective model

However, not all of quadrangles obtained by algorithm used in section III.B may occur to be a projection of a rectangle (with a fixed aspect ratio, if it is specified). Assuming that the document is rectangular, we can evaluate the possibility of a quadrangle to be its projection. To do this we consider a centrally symmetric imaging model [12] — tetrahedral angle with vertex O in the center of the camera lens and the section $ABCD$, that is formed by the image plane and crosses the angle legs in the corners of the obtained quadrangle (Figure 3). It is known that any tetrahedral angle has only one (up to a parallel transfer) section $A'B'C'D'$ which has the form of a parallelogram [13]. Assuming that $\overrightarrow{OD} = \overrightarrow{OD'}$ we calculate the points of this section as follows:

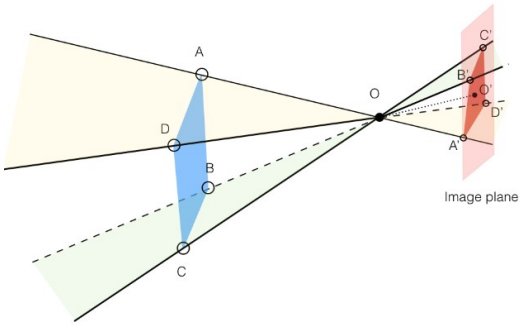

Figure 4. Central-symmetric model of image construction

$$\overrightarrow{OA} = \overrightarrow{OA'} \cdot \left(\vec{b} \wedge \vec{c} + \vec{b} \wedge \vec{d} + \vec{c} \wedge \vec{d} \right) / k \quad (6)$$

$$\overrightarrow{OB} = \overrightarrow{OB'} \cdot \left(\vec{a} \wedge \vec{c} - \vec{a} \wedge \vec{d} + \vec{c} \wedge \vec{d} \right) / k \quad (7)$$

$$\overrightarrow{OC} = \overrightarrow{OC'} \cdot \left(\vec{a} \wedge \vec{b} - \vec{a} \wedge \vec{d} + \vec{b} \wedge \vec{d} \right) / k \quad (8)$$

$$k = \vec{a} \wedge \vec{b} - \vec{a} \wedge \vec{c} + \vec{b} \wedge \vec{c} \quad (9)$$

where $\wedge$ is the operation of the skew product of vectors [14]; $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ is a two-dimensional vectors $\overrightarrow{O'A}, \overrightarrow{O'B}, \overrightarrow{O'C}, \overrightarrow{O'D}$; O' — is the projection of O on the image plane. Another point we could easily evaluate is the angle between the camera optical axis and the normal to the document plane:

$$\vec{n} = \overrightarrow{AB} \times \overrightarrow{AC} \quad (10)$$

$$\cos(\alpha) = \left(\vec{n}, \vec{e} \right) \Big/ \left| \vec{n} \right| \cdot \left| \vec{e} \right| \quad (11)$$

Imposing certain restriction on geometric properties (e.g. angles and aspect ratio) of obtained parallelogram enables us to filter out unsatisfactory quadrangles.

D. Boundaries refining on the original scale image

Since segment detection algorithms are performed on a scaled image to reduce operating time, there ought to be inevitable inaccuracies when finding the position of the document boundaries on the original scale image. Therefore, in the final stage of the algorithm we refine the exact position of the boundaries. For this, we consider a narrow neighborhood of each side of obtained quadrangle on original image as a region of interest (ROI), which should contain the right border. To detect it we put the ROI in the rectangular form, preprocess it with a non-local gradient filter (Figure 4, $p_{4..6}$ corresponds to some light line on dark background):

$$p'_5 = \min(p_{4..6}) - \max(p_{1..3}, p_{7..9}) \quad (12)$$

and then detect a single segment with a Hough transform. We also compare the Hough transform accumulator score of found segment to the threshold and if it is less than threshold, we filter out obtained quadrangle.

E. Parameter tuning

All of the algorithms described above have a number of parameters that significantly affect the final algorithm quality. The total number of parameters of all of algorithms is as high as 50 (like A and B in Formula 5, or thresholds for geometric constraints). Proper parameter tuning on a large dataset is as important for the development of the algorithm as the choice of the basic components. At the same time manual tuning of 50 parameters on a large dataset is quite a long and fastidious process, which, moreover, will not necessarily result in finding the optimal solution.

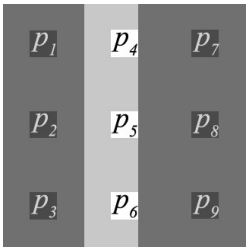

Figure 3. Non-local gradient filter for boundaries refinement

For parameter tuning we use MADS [15] algorithm, implemented in the NOMAD package [16]. This algorithm solves the problem of optimization of an arbitrary function (blackbox optimization). Algorithm takes into account that function to be optimized is typically a simulation, which possesses no exploitable properties such as derivatives, and they may fail to be evaluated at some trial points, even feasible ones. In addition, the evaluation may take a long time and the function may be noisy.

We carry out optimization in two stages. Stage 1 is aimed on segment detection parameter tuning. Therefore, we exclude the projective model filtering and the boundaries refining stages and run the algorithm to obtain all the possible quadrangles. This allows us to ignore errors at the stage of quadrangle assembly and consider only the possibility to obtain the right quadrangle from given set of segments. At the second stage, we fix parameters obtained at the stage 1, and tune other parameters to optimize the final statistics. As an optimized function, we use the Jaccard index as in section II.

Since it was not possible to get the ground truth for the whole dataset, we manually created a ground truth for a part of a dataset (less than a half) and used it to tune parameters. Final quality on a part with our ground truth almost coincide with results and ground truth provided by organizers for the whole dataset, so we could conclude that there was no overfitting. In addition, we should note that if we change a problem fairly enough we need to reevaluate parameters to achieve the best quality. Nevertheless, parameters and final quality does not differ significantly between not much different datasets.

IV. VIDEO INTEGRATION

There are two different approaches to integrate document position among the frames of the video clip. The first and the most qualitative is to assume that we first obtain the whole clip and after it we should provide information for all the frames. Therefore, we are able to detect document positions for all the frames first and then use dynamic programming technics to enhance predictions. However, in spite of this assumption completely correlate with challenge conditions, in the real life this approach lead us to the great time-consuming overhead. Another approach is to use online-manner video integration algorithm. Here we assume that we know only the

Figure 5. Frame sample from fifth background

previous video frames and we could not modify the positions, which we already committed. In this case we have less information about document positions but the algorithm became able to perform in the real time, for instance for predictions highlighting.

Our online-manner video integration algorithm is based on the Kalman filtering [17]. After obtaining each frame, we detect a number of alternatives of positions of the document on the image with the algorithm described in section III. We also fill the set of alternatives with additional alternatives obtained from homography estimation and corner detection.

A. Additional sources for alternative quadrangles

As only we deal with planar objects, we could tie every two consecutive frames with a homography. To do so we could detect feature points on both frames, describe their neighborhoods with some descriptors and evaluate homography with RANSAC. More specifically, we detected 256 key points with SURF [18], use BRIEF [19] to describe them, match descriptors with standard brute force matching and then run classic RANSAC [20] to evaluate homography matrix and number of inliers. Then we transform the best quadrangle from previous frame and add it to alternative set if we have enough confidence computed by the inliers count.

Another way to generate alternative quadrangles is by detecting corner points. That especially useful on the fifth background (Figure 5), where lines are not such strong and document's borders run into underlying document. We detect corners with a Harris corner detector [21] on the gray boundaries image described in the section III.A. Then we try to construct from them a quadrangle, which satisfied rectangular projective model described in section III.C.

B. Result integration with Kalman filter

To further improve the performance across the temporal axis we also implemented a video integration algorithm. As said before, the main algorithm of video integration is based on the Kalman filtering [17]. We use it to filter each corners movements, speed and acceleration, 24 measurements to be filtered in total. We evaluated covariance of noise of process Q and measurements R based on the marked clips.

We use the simple accelerated motion model as the state transition model F:

$$x_k = x_{k-1} + v_{k-1}t + \frac{a_{k-1}t^2}{2} \tag{13}$$

In the terms of Kalman filtering the predicted state estimate and predicted estimate covariance will be:

$$\hat{x}_k^- = F\hat{x}_{k-1} \tag{14}$$

$$P_k^- = FP_{k-1}F^T + Q \tag{15}$$

Then, given a set of alternatives of quadrangles, we choose the one with the largest score scaled by the probability of this quadrangle (assumed we have a normal distribution). Innovation covariance and the best-observed quadrangle will be:

$$S = HP_k^- H^T \tag{16}$$

$$z_k = \arg\max_{z_k^i} \left(s_i \exp\left(-\left(z_k^i - \hat{x}_k^- \right)^T S^{-1} \left(z_k^i - \hat{x}_k^- \right) \Big/ 2 \right) \right) \tag{17}$$

Where $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ – observation matrix, z_k^i – set of alternatives and s_i – their scores.

Optimal Kalman gain, updated state estimate and updated estimate covariance will be:

$$K_k = P_k^- H^T \left(S + R \right)^{-1} \tag{18}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \tag{19}$$

$$P_k = \left(I - K_k H \right) P_k^- \tag{20}$$

where I – identity matrix. Using such a filter on each step, we choose the best appropriate quadrangle from a set of alternatives, based on the current state and update the state by this quadrangle. In addition, we should note we do not use the modified quadrangle outside the Kalman filter.

V. RESULTS

Let us now consider how the use of certain base algorithms affects the final quality of the method. For start the frame-by-frame scheme will be considered, without integration and without boundaries refinement (Table 1). It can be seen that the segments from the Hough transform (HT) are ill suited for this task, and the combined use of contour analysis (CA) and LSD works better than both algorithms separately. Further, the boundaries refinement stably improves the result by about 1.5% (Table 2).

Integration of the results with a Kalman filter for a single quadrangle gives an insignificant quality improvement. Integration of the results and the choice of eight alternatives gives an additional about 0.4%, reaching a result of 95.39%. Usage of additional quadrangles from different sources and post-processing of rejected frames allows us to improve the result up to 95.48%.

All the competition results are given in the Table 3. By dividing the results by backgrounds (Table 4), one can see that the algorithm shows an average result of 98.5% on the first four backgrounds, which is comparable to the result in 98.7% calculated only on quadrangles whose corners are in the 20-pixel neighborhood of the ground-truth corners. This allows one to evaluate the errors and limits of using this quality

TABLE 1. FRAME-BY-FRAME QUALITY WITHOUT REFINEMENT

Method	CA	LSD	HT	CA+LSD
JI	92.46	90.73	74.75	**93.45**

TABLE 2. FRAME-BY-FRAME QUALITY WITH REFINEMENT

Method	CA	LSD	HT	CA+LSD	CA+LSD+HT
JI	93.87	92.39	76.48	**95.02**	92.08

TABLE 3. COMPETITION RESULTS

Ranking	Method	Jaccard	Confidence Index Interval
1	LRDE	0.9716	[0.9710, 0.9721]
2	ISPL-CVML	0.9658	[0.9649, 0.9667]
3	SmartEngines	0.9548	[0.9533, 0.9562]
4	NetEase	0.8820	[0.8790, 0.8850]
5	A2iA run 2	0.8090	[0.8049, 0.8132]
6	A2iA run 1	0.7788	[0.7745, 0.7831]
7	RPPDI-UPE	0.7408	[0.7359, 0.7456]
7	SEECS-NUST	0.7393	[0.7353, 0.7432]

TABLE 4. QUALITY BY BACKGROUND

Method\background	1	2	3	4	5
A2iA-1	0.9724	0.8006	0.9117	0.6352	0.1890
A2iA-2	0.9597	0.8063	0.9118	0.8264	0.1892
ISPL-CVML	0.9870	0.9652	0.9846	0.9766	0.8555
LRDE	0.9869	0.9775	0.9889	**0.9837**	**0.8613**
NetEase	0.9624	0.9552	0.9621	0.9511	0.2218
SEECS-NUST	0.8875	0.8264	0.7832	0.7811	0.0113
RPPDI-UPE	0.8274	0.9104	0.9697	0.3649	0.2163
SmartEngines	**0.9885**	**0.9833**	**0.9897**	0.9785	0.6884
All	0.9465	0.9031	0.9377	0.8122	0.4041

measurement procedure. The fifth background, whose frames are about 10% of the stand, shows the congestion and the absence of precise boundaries of the document - the target sheet lies on the others (Figure 5). In such conditions, the algorithm shows only 68.84%.

CONCLUSION

In this paper, we consider the problem of camera-based document capture. The work was based on the video-dataset, proposed in the ICDAR'15 Smartphone Document Capture and OCR Competition. A set of algorithms that solved the problem of document capture was implemented and tested. The main components were the algorithms for frame-by-frame detection of the document on the image and the boundaries refinement. Further improvement was carried out by integrating the results in the video stream with Kalman filter and by attracting additional sources to obtain quadrangle alternatives. As a result, the quality of the method worked was 95.5% of the Jaccard index. The main problem for the algorithm is the overload of the background and the unobvious position of the boundaries shown on the fifth background. The average quality of the algorithm for the first four backgrounds was 98.5%.

ACKNOWLEDGMENT

The reported study was funded by RFBR according to the research projects 15-07-06520, 16-07-00616, 17-29-03263.

REFERENCES

[1] JC. Burie, J. Chazalon, M. Coustaty et al., "ICDAR2015 Competition on Smartphone Document Capture and OCR", in proc. of 13th Int. Conf. on Document Analysis and Recognition (ICDAR), 2015

[2] V. Arlazarov, A. Zhukovsky, V. Krivtsov, D. Nikolaev, D. Polevoy, "Analysis of specific character of usage fixed and mobile small-size video cameras for document recognition", Information Technologies and computing systems, Vol. 3, 2014, pp. 71-81

[3] J. Liang, D. Doermann, H. Li, "Camera-based analysis of text and documents: a survey", Int. J. of Document Analysis and Recognition, vol. 7, Issue 2, 2005, pp. 84-104

[4] D. Doermann, J. Liang, H. Li, "Progress in Camera-Based Document Image Analysis", IEEE Proc. 7th Int. Conf. on Document Analysis and Recognition, Vol.1, 2003, pp. 606-616

[5] H. Duda, "Use of the hough transformation to detect lines and curves in pictures", Mag. Comm. ACM 15, 1972

[6] R. von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," IEEE Trans. PAMI, vol. 32, no. 4, 2010, pp. 722–732

[7] P. Dollár, C. L. Zitnick, "Structured forests for fast edge detection", Proc. of the IEEE Int. Conf. on Computer Vision, 2013, pp. 1841-1848

[8] L.R.S. Leal, B.L.D. Bezerra, "Smartphone Camera Document Detection Via Geodesic Object Proposals", IEEE Latin American Conf. on Computational Intelligence (LA-CCI), 2016

[9] E. Carlinet, T. Géraud, "MToS: A Tree of Shapes for Multivariate Images", IEEE Trans. On Image Processing, 24 (12), 2015, pp.5330-5442

[10] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge" IJCV, vol. 88, no. 2, 2010, pp. 303–338

[11] N. Skoryukina, D. Nikolaev, A. Sheshkus, D. Polevoy, "Real Time Rectangular Document Detection on Mobile Devices", SPIE Proc. 9445, ICMV, 2014

[12] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, New York, 2003

[13] K. Shakhno, Collection of problems in elementary mathematics of increased difficulty, Ripol Classic, Moscow 2013

[14] V. Prasolov, Problems in plane and solid geometry, MCNMO, Moscow, 2001

[15] C. Audet, J. E. Dennis, Jr., "Mesh Adaptive Direct Search Algorithms for Constrained Optimization", SIAM J. Optim. 17, 2006, pp. 188–217

[16] S. Le Digabel, "Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm", ACM Trans. Math. Softw. 37, 4, 2011

[17] Kalman R. E., "A New Approach to Linear Filtering and Prediction Problems", J. of Basic Engineering 82, 35, 1960

[18] H. Bay, T. Tuytelaars, L. V. Gool, "Surf: Speeded up robust features", European Conf. on Computer Vision (ECCV), 2006, pp. 404-417

[19] M. Calonder, V. Lepetit, C. Strecha, P. Fua, "BRIEF: Binary Robust Independent Elementary Features", 11th European Conf. on Computer Vision (ECCV), 2010

[20] Fischler M.A., Bolles R.C., "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, 24(6), 1981, pp. 381-395.

[21] C. Harris, M. Stephens, "A combined corner and edge detector", in proc. of the 4th Alvey Vision Conference, 1988, pp. 147–151