MSC 90C35, 90C27 **DOI: 10.14529/mmp220407**

METHOD FOR ANALYZING THE STRUCTURE OF NOISY IMAGES OF ADMINISTRATIVE DOCUMENTS

O.A. Slavin[1]***, E.L. Pliskin***[1,2]
[1]Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russian Federation
[2]LLC "Smart Engines Service", Moscow, Russian Federation
E-mail: oslavin@isa.ru, epliskin@isa.ru

The problem of extracting content elements (fields) from the images of administrative documents via descriptions of anchoring elements is considered. Administrative documents contain static elements and content elements (filled information). The static objects of the document model are the lines of the document structure and the words. Sets of objects united by properties and relationships are described. The text descriptor can contain attributes that distinguish it from similar descriptors. We suggest using combined descriptors of line segments and words. We showed experimentally that the extraction of object sets improves the recognition accuracy of the document fields by 17% and the accuracy of information extraction by 16%. For optical character recognition, we employed SDK Smart Document Engine in the experiment.

Keywords: noisy image; document recognition; special text point; descriptor.

Introduction

The document image recognition is a relevant problem since the number of documents printed on paper is growing. For example, in large organizations, the volume of incoming and outgoing document flows can reach up to $O(10^6)$ pages per day. Administrative document recognition allows to automate organization document management processes [1].

We define a document as a set of fields and static information. This paper considers administrative documents. Administrative documents are characterized by a relatively simple structure and a limited vocabulary for static text. Static elements are, first of all, the words of a static text. Static words are grouped into lines, headings, and paragraphs. Fields can be defined as an object that

- differs from neighboring elements in some neighborhoods (bar code),
- is bounded by several static elements.

Extracting information from recognized administrative documents has several nuances. For example, the short dictionary of allowed keywords, and the fact that a significant number of recognition errors when comparing words should be taken into account. The recognition problem can be defined as follows. Based on the recognition of text objects and graphical primitives, the information from the areas corresponding to the boundaries of the fields should be extracted. It is required to extract information for the maximum number of fields with the smallest number of errors for each field. This work considers a two-pass method of document recognition using field descriptors. During the first pass after text recognition and extraction of graphic primitives, the boundaries of the fields are predicted. The boundaries of the fields are defined via the word boundaries of the static text. Static text words are grouped for reliable identification. The invariance to document distortions is ensured by a constellation model which takes into account a large number of possible recognition errors. During the second pass, the quality of the recognition of the fields in the detected boundaries is improved by parametrization of the recognition.

This work considers recognized noisy document images. Images of characters, line segments, and checkboxes can be obscured by noise. Currently, this topic is of great interest [2,3].

One of the document image recognition work flows includes the following steps:
- page processing and normalization;
- word recognition, extraction of graphic primitives and other objects;
- rectification of objects and search for local features;
- search for field boundaries using local feature boundaries;
- extraction of field contents in the detected boundaries via field attributes.

1. Background

For rigid templates, local features are specified as keypoints of various types (SURF, YAPE, YOLO, SIFT, ASIFT). The descriptors of modern keypoints can be quite complicated to detect [4, 5]. The field is bounded by a set of keypoints picked through several conditions. It is assumed that connections between pairs of points are preserved in the rigid form after digitizing the document. In a flexible document, the connections between pairs of points are not preserved at all, because the document design can be changed. The structure of a flexible document can be augmented as follows:
- changing fonts;
- changes in line spacing, margins of text columns;
- substitution of words in the static text;
- changing the positions of words and lines of document structure.

Similar to the keypoints for rigid templates of flexible documents, it is possible to use special text points. This work continues the study of a model of flexible administrative documents based on special text points [6,7]. A pair $T(W), B(W)$ defines a descriptor of a special text point W, where $T(W)$ is a core of a special text point W, i.e. the sequence of positions for characters corresponding to some alphabet, and $B(W)$ is a boundary which consists of coordinates of a quadrilateral which corresponds to an area of a special text point W. The detector of the special text point is the recognition procedure via OCR. The comparison of two special text points is performed using the modified Levenshtein distance [8]. The proposed modifications take into account the possibility of a large number of errors in the recognition of noisy document images. The proposed modifications also effectively distinguish words that are close according to the classical Levenshtein metric.

The set of characters different from neighboring characters in same vicinity is called a *local text feature*. For example, a recognized word that is identical to some local text feature allows the identification of neighboring recognized words. Some words bounding the field are not local features. We suggest using composite descriptors (*chains*). A chain C consists of a sequence of *terms*

$$C = \{\mathrm{Tm}_1(C), \mathrm{Tm}_2(C), \dots \mathrm{Tm}_n(C)\}.$$

The term $\mathrm{Tm}_i(C)$ is defined as a set of descriptor alternatives of special text points $W_1(\mathrm{Tm}_i(C)), W_2(\mathrm{Tm}_i(C)), \dots$ and several thresholds $d_1(\mathrm{Tm}_i(C)), d_2(\mathrm{Tm}_i(C)) \dots$ between a given term Tm_i and a previous term $\mathrm{Tm}\, i - 1$ when $i > 1$. The connections between terms are not required. Each of the thresholds $d_k(\mathrm{Tm}_i(C))$ is a parameters within a condition

$$\rho_T^k(\mathrm{Tm}_{i-1}(C), \mathrm{Tm}_i(C)) < d_k(\mathrm{Tm}_i(C)),$$

where ρ_T^k is one of the possible metrics between two terms. Examples of metrics are: the number of words between two terms or the distance calculated using the boundaries of special text points within terms. Terms may have attributes to distinguish them from other terms when comparing them. Term *linking* is defined as selecting words similar to a term from the set of words in a recognized document. Linking is performed using the Levenshtein distance in combination with a set of connections to other terms.

The simplest chain consists of a single term. The definition of a chain allows a description of a unique sequence of terms that differs from other chains in the context of a document or part of a document (lines, paragraphs, fragments).

During linking of the chain C and a recognized document, the descriptor of a chain and recognized words are employed. The candidates W_q^{REC} to be the term $\mathrm{Tm}_i(C)$ are verified in terms of correspondence to the sequence of the terms in the chain C. Specifically, from all candidates to be a term, the selected candidates should minimize the score of terms sequence of the chain C:

$$\delta(C) = \max(\rho_{\mathrm{LEV}}(T(\mathrm{Tm}_i(C)), W_q^{\mathrm{REC}})) \to \min\,. \tag{1}$$

A single-line or a multi-line field F is described by a field descriptor $\{C_1; F; C_2\}$. The left boundary of the field F is determined via boundaries of the rightmost term of the chain C_1, the right boundary is determined via the leftmost term of the chain C_2. The chains C_1 and C_2 are anchoring elements for the field F. If the boundary terms are not bounded, other terms of the chains C_1 and C_2 can be used for linking. In this case, the field linking error may increase (the difference between the predicted and the real field boundaries in the document image). One of the chains C_1 or C_2 may be missing in the descriptor field. In such cases, the boundary corresponding to the field is set by the boundary of the fragment or document. The upper and lower boundaries of one of the field F lines are set by the boundaries of text lines within chains or positioned between the lower boundary of the chain C_1 and upper boundary of the chain C_2. For a line or a paragraph containing several fields, the descriptor can be defined as follows:

$$D_{\mathrm{TEXT}}(F_1, F_2, \ldots, F_{p2}) = \begin{cases} C_1^{\mathrm{top}}; C_2^{\mathrm{top}}; \ldots C_{p1}^{\mathrm{top}}; \\ C_1; F_1;\ C_2; F_2; \ldots C_{p2}; F_{p2};\ C_{p2+1}; \\ C_1^{\mathrm{bottom}}; C_2^{\mathrm{bottom}}; \ldots C_{p3}^{\mathrm{bottom}}. \end{cases} \tag{2}$$

In descriptor (2), the chains $C_1^{\mathrm{top}}, C_2^{\mathrm{top}}, \ldots C_{p1}^{\mathrm{top}}$ limit from above the areas of search for the anchoring chains $C_1, C_2, \ldots C_{p2+1}$. Similarly, the chains $C_1^{\mathrm{bottom}}, C_2^{\mathrm{bottom}}, \ldots C_{p3}^{\mathrm{bottom}}$ limit from below the area of search for anchoring chains. Linking scoring of descriptor for several fields is based on linking scores for chains within this group

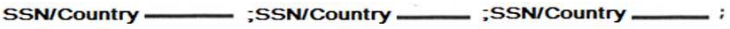
$$\delta(C_1, C_2, \ldots C_{p2}) = \min(\delta(C_k)).$$

Note that the effective linking is based on the search for special for some neighborhood text points and chains. In the context of the entire administrative document, a hierarchical division into fragments – areas that are separated from each other by dividing lines or large gaps – is possible. The context of a fragment has substantially fewer words than the context of an entire document. This division simplifies the use of brute-force algorithms.

2. Line Segments Descriptor

In the example shown in Fig. 1, linking can be difficult both because of the large number of possible character recognition errors and because of the loss of some words.

In the case of poor character recognition as illustrated in Fig. 1, field linking can be improved by using line segments. In the cases discussed below, detecting document objects bounded by lines makes it easier to link fields, table cells, or more complex areas. The attributes of the line segment S are



SSN/Country ;SSN/Country ;SSN/Country ;

Fig. 1. Example of a document line that lacks local text features

- orientation (vertical, horizontal);
- type (solid, dashed, dotted);
- boundary of the line segment S;
- length $h(S)$ and thickness $w(S)$ of the line segment S;
- approximate position in the image of a document or fragment.

There are several algorithms for the extraction of line segments from the document images [9–11]. Some algorithms form a table or a complex shape from an array of selected line segments. More often than not, a single line segment does not differ from a set of similar line segments and cannot be considered as an anchoring element. Let us consider objects consisting of text objects and line segments.

We suggest to use combined objects consisting of chains and line segments. The combined object is the following set of chains and line segments:

$$D(F_1, F_2, \ldots, F_{p2}) = \begin{cases} C_1^{\text{top}}; C_2^{\text{top}}; \ldots C_{p1}^{\text{top}}; \\ C_1; \frac{F_1}{S_1}; C_2; \frac{F_2}{S_2}; \ldots C_{p2}; \frac{F_{p2}}{S_{p2}}; C_{p2+1}; \\ C_1^{\text{bottom}}; C_2^{\text{bottom}}; \ldots C_{p3}^{\text{bottom}}. \end{cases} \tag{3}$$

In definition (3), the line segments $S_1, S_2, \ldots S_{p2}$ should be a part of the single line S. The distance between the line S and each term boundary of the chains $C_1, C_2, \ldots C_{p2+1}$ is limited. Elements in (3) are selected and adjusted during the model training so that the descriptor is locally special. Some of the chains and segments (3) may be optional in linking. Linking scoring is based on the linking scores of each of the chains and the linking scores of the set of line segments. A set of line segments is scored by comparing the number and size ratios of the line segments of the combined descriptor with the line segments extracted from the document image. The linking score of the multi-fields descriptor is based on the linking scores of chains within the group and the linking score of the line segments set

$$\delta(C_1, C_2, \ldots C_{p2}) = \min(\delta(C_k)) - \delta(S_1, S_2, \ldots, S_{p2+1}).$$

Let us consider simplified representation of descriptor (3):

$$D(F_1, F_2, \ldots, F_{p2}) = \begin{cases} C_1^{\text{top}}; C_2^{\text{top}}; \ldots C_{p1}^{\text{top}}; \\ \frac{F_1}{S_1}; \frac{F_2}{S_2}; \ldots \frac{F_{p2}}{S_{p2}}; \\ C_1^{\text{bottom}}; C_2^{\text{bottom}}; \ldots C_{p3}^{\text{bottom}}. \end{cases} \tag{4}$$

In descriptor (4), the anchoring elements are upper and lower chains and the sequence of line segments $S_1, S_2, \ldots, S_{p2+1}$. It is assumed that the projections on the vertical axis of all line segments are the same or insignificantly different. During the linking, we consider all extracted line segment candidates $s_1, s_2, \ldots s_r$ located within the area bounded by the chains $C_1^{\text{top}}, C_2^{\text{top}}, \ldots C_{p1+1}^{\text{top}}$ and $C_1^{\text{bottom}}, C_2^{\text{bottom}}, \ldots C_{p3+1}^{\text{bottom}}$. The coordinates of candidate line segments are normalized to the width of the fragment or the page width of the document. The cluster analysis of the candidate line segments $\{s_1, s_2, \ldots\}$ is performed. The clustering is performed via analysis of the proximity of candidate line segments projections onto the vertical axis. Then we score the correspondence between each of the resulting line segments cluster with $\mathrm{Cl}_t = \{s_1^t, s_2^t, \ldots\}$ and the line segments descriptors $S_1, S_2, \ldots, S_{p2+1}$. If the number of line segments in the cluster coincides with the number of line segments within the descriptor, the cluster linking score is calculated as the sum of the length discrepancies within the pairs of line segments:

$$\delta(\mathrm{Cl}_t, D(F_1, F_2, \ldots, F_{p2+1})) = \sum_j |h(S_j) - h(s_j)|. \tag{5}$$

We also consider cases when several line segment candidates $s_z^t, s_{z+1}^t, \ldots, s_{z+l(j)}^t$ correspond to a single line segment S_j. And we consider cases when several line segments $S_V, S_{V+1}, \ldots, S_{V+r(z)}$ correspond to a single line segment candidate s_z. For these cases, combining several line segments or candidate line segments is provided as follows. Combining is based on a brute-force algorithm over descriptor line segments and candidate line segments. In the case of the line segment size invariance (up to the scaling accuracy), when documents via minimizing penalty (5), we select sets of candidate line segments Cl_t, boundaries of which best correspond to the boundaries

$$S_1, S_2, \ldots$$

Linking using combined descriptor (3) is improved compared to linking using text descriptor(2). This is due to the following reasons:

- images of massive line segments obscured by signatures and seals are extracted more reliably than characters and words;
- multiple chains are used to limit the search area of line segments set.

The success of the linking via descriptors (3) and (4) is guaranteed in cases when the line segment extraction method robustly detects line segments within noisy document images. Employment of descriptors with dotted line segments within noisy images can be less effective.

3. Checkboxes Descriptor

The checkbox is one of the most common graphical elements in document design. The checkbox area is bounded by four line segments. The checkbox line segments can be either solid or dotted. Examples of checkboxes are shown in Fig. 2.

An image of an empty or filled checkbox is not difficult to detect using contour selection [12], CNN [13], or Viola and Jones method [15]. Let us assume that at the first stage of document processing, along with word and character recognition, checkbox frames were detected. During recognition, checkboxes can be not detected and false detection is possible. Combined descriptors can also be used to correct redundant detection of candidate checkboxes.

Fig. 2. Examples of checkboxes

In descriptor (2), each of the fields F can be interpreted as a predicted checkbox frame. Assume that descriptor (2) is bounded. The linking of the checkboxes included in a descriptor is performed as follows. As a predicted boundary of the field F let us select a candidate checkbox, the area of intersection of which with the frame F is maximal. The checkbox boundary may be obscured due to handwritten filling. In this case, a predicted boundary is used to detect the checkbox.

With a large number of errors in character recognition and line segment detection, the chains in descriptors (2), (3), (4) may not be bounded. In some cases, for example, for a group of vertical checkboxes, the following descriptors can be applied:

$$D_{CB}(F_1, F_2, \ldots, F_{p2}) = \begin{cases} C_1^{\text{top}}; C_2^{\text{top}}; \ldots C_{p1}^{\text{top}}; \\ C_1; F_1; C_1'; \\ C_2; F_2; C_2'; \\ \qquad \ldots \\ C_{p2+1}; F_{p2}; C'_{p2+1}; \\ C_1^{\text{bottom}}; C_1^{\text{bottom}}; \ldots C_{p3}^{\text{bottom}}. \end{cases} \tag{6}$$

In descriptor (6), the anchoring elements are the upper and lower bounding chains $C_1^{\text{top}}, C_2^{\text{top}}, \ldots C_{p1+1}^{\text{top}}$ and $C_1^{\text{bottom}}, C_2^{\text{bottom}}, \ldots C_{p3+1}^{\text{bottom}}$ and a group of vertical checkboxes. An example of such a group is illustrated in Fig. 3. The chains $C_1, C_1', C_2, C_2', \ldots C_{p2+1}, C'_{p2+1}$ do not require elements when linking a group of checkboxes. However, linking of some chain from the chains $C_1, C_1', C_2, C_2', \ldots C_{p2+1}, C'_{p2+1}$ makes linking easier when not all of the checkboxes were detected.

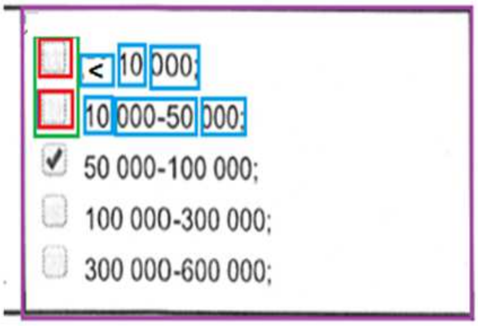


Fig. 3. Example of a vertical group of checkboxes

4. Using Seal Images

Currently, there are several groups of methods for the detection of seals in the document images:

- methods that are based on the search of geometric primitives [16], including methods based on generalized Hough transform and the detection and analysis of keypoints in the image;
- methods that are based on spectral analysis of image pixels [17]. The problem of seals detection in document images is complicated by a number of factors:
- violation of the integrity of the seal;
- a large number of noise elements;
- physical fading over time;
- presence of various security elements in the area of search for a seal (watermarks, guilloche patterns).

The case of black and white copies of document images is especially difficult. It makes spectral methods for seals detection useless.

We use the modified Viola and Jones method to detect seals in a document image. The modifications include the simultaneous use of brightness features and Haar edge features and the processing of seal area features [14, 18].

The authentication of administrative documents with seals must be taken into account for several reasons. First, the detection of a seal in a particular area is similar to the extraction of other attributes. The presence or absence of a seal is one of the elements of document verification. Second, a detected seal is a special object. It can be used as the same anchoring element as bounded words, line segment groups, and checkboxes.

In some documents, the seal can be arbitrarily located.

5. In Some Documents, the Seal Can Be Arbitrarily Located. Creating Descriptors

The descriptors proposed in this paper as well as similar descriptors were created using the developed linking language within SDK Smart Document Engine[1] for customization of flexible document linking.

Descriptors are created based on the following objects:

- keyword;
- word from a dictionary;
- field;
- line segment;
- line;
- checkbox;
- fragment.

Real objects are as follows:

- recognized word, i.e. the sequence of characters and the boundary of the word;
- detected line segment, i.e. the boundary and attributes of line segment;
- detected checkbox, i.e. the boundary of the checkbox;
- detected fragment, i.e. the boundary of the fragment and its type.

Procedures for processing bounded objects are possible. Tags containing arbitrary parameters can be assigned to each object, descriptor, or procedure.

Below, we provide an example of a combined descriptor.

$$D(F_{295}, F_{300}, F_{296}, F_{306}) = \begin{cases} \{VZ_{0397}; VZ_{0314}\}; \\ \frac{F_{295}}{S_{295}}; \frac{F_{300}}{S_{300}}; \frac{F_{296}}{S_{296}}; \frac{F_{301}}{S_{301}}; \end{cases}$$

where the elements of the descriptor $D(F_{295}, F_{300}, F_{296}, F_{301})$ are defined as follows:

- VZ_{0397} and VZ_{0314} are keywords;
- $S_{295}, S_{300}, S_{296}$ and S_{301} are segments;
- $F_{295}, F_{300}, F_{296}$ and F_{301} are fields;

The description of each element includes:

- identifier and a unique number;
- core of the keyword;
- boundary;
- element type;
- comment;
- links to other elements of the descriptor.

The proposed language for descriptors develops the ideas proposed in [19]. In these papers, a model of a flexible document consisting of structural elements (a static text or a field enclosed by quadrangles) and geometric links between them is considered.

[1] Smart Document Engine – Automatic Analysis and Data Extraction from Business Documents for Desktop, Server and Mobile Platforms. Available at: https://smartengines.com/ocr-engines/document-scanner (accessed April 22, 2022)

6. Results of Experiments

The proposed method of document structure analysis was tested on private test dataset which includes images of “universal transfer document” type documents scanned at an optical density from 100 to 300 dpi with varying quality of digitization. The accuracy of recognition and the accuracy of field linking were investigated. The cases with and without the use of line segments as elements of the combined descriptors were considered. SDK Smart Document Engine was used for recognition. The results are summarized in Table, which illustrates the efficiency of the proposed method.

Table

Linking accuracy for the fields of universal transfer documents in the test dataset

Document	Number of documents	Number of fields	Fields recognition accuracy	Fields linking accuracy
with employment of line segments	1545	55648	66,77%	82,53%
without employment of line segments	1545	55648	49,15%	66,26%

The accuracy of seals detection calculated for the private dataset of “KYC” documents is 97,5%.

7. Runtime of Method

The C++ implementation of the proposed method for analyzing the structure of noisy documents was tested on a computer with Intel Core i9-9900 3.60 GHz, DDR-2666 MHz. Microsoft Visual Studio Community 2019 environment with */O2* parameter optimization was used to implement the method in C++ programming language.

Linking takes from 50 to 200 milliseconds, depending on the document type. Most of the time (up to 95%) is spent comparing terms and recognized words using a modified Levenshtein distance. The average time to compute word comparisons using a modified Levenshtein distance was $1,8 \cdot 10^{-4}$ milliseconds.

It should be noted that the first pass, which includes searching for graphical primitives (words, segments, checkboxes, seals) and recognizing characters, takes from 300 milliseconds up to several seconds.

Conclusion

The proposed method of document structure analysis is applicable for the recognition of highly noisy and distorted document images. In particular, the method makes it possible to detect field boundaries in black and white copies of color documents. Using several types of primitives for linking guarantees efficiency of detection and predicting single- and multiline field borders. Efficiency is achieved through the use of combined descriptors.

Combined descriptors are based on simple descriptors of static text words, line segments, checkboxes, and seals images.

The proposed method was tested on several types of administrative documents, such as tax declaration, financial statements, trade documents, bank documents.

The accuracy of the proposed method was evaluated on private datasets. The linking accuracy when using combined descriptors ranges from 80 to 99% for different fields. The worst results correspond to complex cases of seals and signatures which overlap with words in a black and white image of a color administrative document. For such fields the recognition accuracy is low. However, field boundaries prediction is necessary for navigating through the document during user analysis.

Acknowledgment.*The research was supported by the Russian Foundation for Basic Research (Project 20-07-00934 "Development, properties study and justification of anytime algorithms for computed tomography").*

References

1. Rusinol M., Frinken V., Karatzas D., Bagdanov A.D., Llados J. Multimodal Page Classification Inadministrative Document Image Streams. *International Journal on Document Analysis and Recognition*, 2014, vol. 17, no. 4, pp. 331–341. DOI: 10.1007/s10032-014-0225-8
2. Jain R., Wigington C. Multimodal Document Image Classification. *Document Analysis and Recognition*, 2019, vol. 2019, pp.71–77. DOI: 10.1109/ICDAR.2019.00021
3. Qasim S.R., Mahmood H., Shafait F. Rethinking Table Recognition Using Graph Neural Networks. *Computer Vision and Pattern Recognition*, 2019, vol. 1, pp. 142–147. DOI: 10.1109/ICDAR.2019.00031
4. Bellavia F. SIFT Matching by Context Exposed. *Transactions on Pattern Analysis and Machine Intelligence*, 2022, vol. 2022, pp. 1–17. DOI: 10.1109/TPAMI.2022.3161853
5. Bay H., Tuytelaars T., Luc Van Goolab. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2006, vol. 110, no. 3, pp. 404–417. DOI: 10.1016/j.cviu.2007.09.014
6. Slavin O., Andreeva E., Paramonov N. Matching Digital Copies of Documents Based on OCR. *Control and Modeling Problems*, 2019, vol. 2019, pp. 177–181. DOI: 10.1109/CSCMP45713.2019.8976570
7. Slavin O., Arlazarov V., Tarkhanov I. Models and Methods Flexible Documents Matching Based on the Recognized Words. *Cyber-Physical Systems: Advances in Design and Modelling*, 2021, vol. 350, pp. 173–184. DOI: 10.1007/978-3-030-67892-0_15
8. Deza M.M., Deza E. *Encyclopedia of Distances.* Berlin, Springer-Verlag, 2009.
9. Matas J., Galambos C., Kittler J. Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. *Computer Vision and Image Understanding*, 2000, vol. 78, issue 1, pp. 119–137. DOI: 10.1006/cviu.1999.0831
10. Grompone von Gioi R., Jakubowicz J., Morel J.M. On Straight Line Segment Detection. *Journal of Mathematical Imaging and Vision*, 2008, vol. 32, pp. 313–347. DOI: 10.1007/s10851-008-0102-5
11. Grompone von Gioi R., Jakubowicz J., Morel J.M., Randall G. LSD: A Fast Line Segment Detector with a False Detection Control. *Transactions on Pattern Analysis and Machine Intelligence*, 2010, vol. 32, issue 4, pp. 722–732. DOI: 10.1109/TPAMI.2008.300
12. Emaletdinova L., Nazarov M. Construction of a Fuzzy Model for Contour Selection. *Studies in Systems, Decision and Control*, 2022, vol. 417, pp. 243–246. DOI: 10.1007/978-3-030-95116-0_20
13. Zlobin P., Chernyshova Y., Sheshkus A., Arlazarov V.V. Character Sequence Prediction Method for Training Data Creation in the Task of Text Recognition. *Machine Vision*, 2021, vol. 2021, article ID: 120840, 10 p. DOI: 10.1117/12.2623773
14. Matalov D., Usilin S., Arlazarov V.V. About Viola–Jones Image Classifier Structure in the Problem of Stamp Detection in Document Images. *Machine Vision*, 2021, vol. 2021, article ID: 11605, 16 p. DOI: 10.1117/12.2586842

15. Arlazarov V., Voysyat Ju.S., Matalov D., Nikolaev D., Usilin S.A. Evolution of the Viola-Jones Object Detection Method: A Survey. *Bulletin of the South Ural State University. Mathematical Modelling, Programming and Computer Software*, 2021, vol. 14, no. 4, pp. 5–23. DOI: 10.14529/mmp210401
16. Roy P.P., Pal U., Llados J. Seal Detection and Recognition: An Approach for Document Indexing. *Document Analysis and Recognition*, 2015, vol. 2015, article ID: 367879, 15 p. DOI: 10.1109/ICDAR.2009.128
17. Katsuhiko U. Extraction of Signature ad Seal Imprint from Bankchecks by Using Color Information. *Document Analysis and Recognition*, 1995, vol. 1995, pp. 665–668. DOI: 10.1109/ICDAR.1995.601983
18. Matalov D., Usilin S., Arlazarov V.V. Modification of the Viola-Jones Approach for the Detection of the Government Seal Stamp of the Russian Federation. *Machine Vision*, 2019, vol. 2019, article ID: 10411, 11 p. DOI: 10.1117/12.2522793
19. Marchenko A.E., Ershov E.I., Gladilin S.A. The System for Parsing a Document Specified by Attributes of Structural Elements and the Rrelations between Structural Elements. *Trudy ISA RAN*, 2017, vol. 67, no. 4, pp. 87–97. (in Russian)

Received September 15, 2022

УДК 004.932.72'1 **DOI: 10.14529/mmp220407**

МЕТОД АНАЛИЗА СТРУКТУРЫ ЗАШУМЛЕННЫХ ОБРАЗОВ ДЕЛОВЫХ ДОКУМЕНТОВ

***О.А. Славин*[1], *Е.Л. Плискин*[1,2]**

[1]Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Российская Федерация

[2]ООО «Смарт Энджинс Сервис», г. Москва, Российская Федерация

Рассматривается задача извлечения из образа делового документа элементов заполнения (полей) с помощью описаний опорных элементов. Деловые документы содержат статические и переменные элементы (заполнение). Статичными объектами модели являются линии разграфки и слова текста. Описываются наборы объектов, объединенные свойствами и отношениями. Текстовый дескриптор может содержать атрибуты, позволяющие отличать его от сходных дескрипторов. Мы предлагаем применять комбинированные дескрипторы, состоящие из отрезков линий и слов. Экспериментально показано, что извлечение наборов объектов повышает точность распознавания полей документа на 17%, а точность извлечения информации из образа документа – на 16%. В качестве оптического распознавания символов в эксперименте использовалась система SDK Smart Document Engine.

Ключевые слова: зашумленный образ; распознавание документа; текстовая особая точка; дескриптор.

Олег Анатольевич Славин, доктор технических наук, главный научный сотрудник, Федеральный исследовательский центр «Информатика и управление» РАН (г. Москва, Российская Федерация), oslavin@isa.ru.

Евгений Львович Плискин, кандидат технических наук, старший научный сотрудник, Федеральный исследовательский центр «Информатика и управление» РАН (г. Москва, Российская Федерация); старший научный сотрудник-программист, ООО «Смарт Энджинс Сервис» (г. Москва, Российская Федерация), epliskin@isa.ru.

Поступила в редакцию 15 сентября 2022 г.