

Chapter 2

Graphological Analysis and Identification of Handwritten Texts

Leonid A. Mironovsky, Alexander V. Nikitin,
Nina N. Reshetnikova and Nikolay V. Soloviev

Abstract The problem of recognition of handwriting text is still far from its final solution. The existing systems of recognition of handwritten texts are usually developed for some special applications. The difficulties are caused by recognition of the conjoint writing because a variability of handwritings is the highest and often it is necessary to solve the problem of delimitation of the separate letters. In this chapter, along with to the known methods of the handwritten fragments' analysis, it is offered to use the developed methods of vectorization of raster images and vector dynamic parameterization. Also, a description of the automated information storage and retrieval system for the graphological analysis and identification of unintelligible fragments of handwritten texts is given. The system contains a database of handwriting samples with variants of the author's calligraphy from the Manuscript Department of the Institute of Russian Literature (Pushkin's House) of the Russian Academy of Sciences.

Keywords Graphological analysis • Handwritten text • Text segmentation
Comparison of words • Symbols and ligatures • Dynamic parameterization
Drafts autographs • Automated information retrieval system

L.A. Mironovsky (✉) · A.V. Nikitin · N.N. Reshetnikova · N.V. Soloviev
St. Petersburg State University of Aerospace Instrumentation,
67 Bol. Morskaya St, Saint Petersburg 190000, Russian Federation
e-mail: miron@aanet.ru

A.V. Nikitin
e-mail: nike51@mail.ru

N.N. Reshetnikova
e-mail: reni_07@list.ru

N.V. Soloviev
e-mail: famsol@yandex.ru

2.1 Introduction

The first attempts to solve the problem of handwritten or printed text recognition in bitmap images were made more than fifty years ago, almost immediately after appearance of the devices that could upload images into computer memory [1, 2]. In this context, the text recognition means an automated process for obtaining of ASCII codes of symbols (letters, numbers, and punctuation marks). Selection and partial recognition of image segments in the photos, diagrams, plots, mathematical formulas, and tables are also possible.

Today there are a lot of software products that successfully recognize the printed characters [3, 4]. For bitmap images with high resolution (300–600 dpi), contrast, and sharpness, the number of recognition mistakes does not exceed 0.5% [5]. Also, there are a lot of software products for processing of images to be recognized in order to increase their contrast and sharpness with removal possible noise and defects [6]. In this case, the successful recognition can be implemented easily thanks to the fact that a bitmap image of printed text could be easily segmented into images of separate characters and the images of similar characters are nearly identical.

The process of recognition of printed texts is commonly referred to as Optical Character Recognition (OCR) [7] even though the OCR means recognition of any text, printed or handwritten [8]. It is notable that apart from the OCR software, which was designed for image processing using the desktop image scanners, some devices were designed specifically for recognition of the printed texts [9]. The problem of recognition of the mixed printed and handwritten texts is also solved quite successfully [4]. The main difference of such text from printed ones lies in the fact that images of similar characters can significantly differ from each other. To solve this problem, some text recognition software tools include the learning algorithms. In this case, a software tool attempts to learn the specific handwriting style from an image, which significantly increases the chance for successful recognition [10].

The most difficult task is the recognition of joined-up handwriting. This process is known as Hand Writing Recognition (HWR) [11]. The main differences of handwritten texts from the printed ones are described below. First, the lines of handwritten texts are often not parallel, which is especially common for texts written on an unlined sheet. There also can be partial overlapping of the adjacent lines and words/letters in these lines. Second, even though the words in handwritten texts are separated with spaces, some words are not written jointly, which means that some letters or even letter fragments can be separated with spaces. Finally, even one person's handwriting depends on many factors. For example, a readability of a word in the official document can differ a lot from the same word written in a personal note book. Moreover, a way of the letter writing greatly depends on the writing of adjacent letters. The first two of the mentioned above reasons complicate the segmentation of images with handwritten texts into separate words, while the third one complicates the recognition of letters.

The chapter is organized as follows. The related works are reviewed in Sect. 2.2. Section 2.3 provides the analysis and identification of handwritten texts including a problem formulation. A formation of alphabets based on the handwriting samples is considered in Sect. 2.4. The methods of bitmap binary image based on the graphological analysis and identification are represented in Sect. 2.5, while the methods of the graphological analysis and identification based on vectorization of bitmap images are developed in Sect. 2.6. Section 2.7 includes a description of architecture of the designed software tool. Section 2.8 concluded the chapter.

2.2 Related Works

The interest to the problem of recognition of the joined-up handwritten texts is unabated, which is evident from the great amount of publications on this topic [12–33]. There are two main approaches that support two different ways to obtain the handwritten texts called as “on-line” recognition (when the characters are written with a pen in a special screen, for example a tablet PC screen) and “off-line” recognition (when a document with a handwritten text already exists). Generally, the on-line recognition is easier task because the problem of text segmentation into words does not appear due to the on-line mode of capturing a lot of clues that can be successfully used to learn the users’ handwritings. The existing on-line hardware and software recognition products are effectively demonstrate this proposition [34].

The off-line text recognition is a more complicated task. As it is listed in many reviews [13, 35–37], the classic methods for selection of the handwriting features, creation of a database with samples, and features identification do not work successfully in some cases [38]. It is known [39] that the artificial neural networks are able to identify the generalized features of recognizable images. In this regard, the application of artificial neural networks to the off-line recognition of handwritten texts seems to be a good idea [40–52]. The reasons for mistakes lie in the facts that the same characters can be written in different ways by different people and it is very hard to identify the separate characters in the handwritten texts.

While recognizing a handwritten text, people consider its context, which means that they take into account the information they obtained from the parts of the same text, which they read before. It is known [53] that a human being can easily read a word if it is a part of a discourse (normal text) but the same person will have problems while indentifying the same word in a text, consisting of random words. It is even more difficult for a human being to identify a separate letter even though that same letter in a text can be read easily. Presumably, this is the recognition systems’ ability to understand the meaning of handwritten texts that will allow for a breakthrough in the science but the existing intelligent systems are still far from solving this task.

Sometimes a person, who reads a handwritten text, faces a problem of recognizing a separate word that can be one of several possible similar-looking words, each of which does not violate the rules of grammar and retains the meaning of the

text. For example, this problem can be faced by a literary scholar, who works with a handwritten archive of a writer [54]. Such person visually compares an unidentified word fragment with similar-looking fragments of words that are already recognized in order to find the right one. The system described below in this chapter was designed to speed up this process and increase the correctness of recognition results.

It is obvious that the problem of handwritten text recognition is far from being completely solved. The existing handwritten text recognition systems are usually developed for some special cases or applications based on the specific features of the texts to be recognized. These specific features are the quality of handwritten texts and size of vocabulary. The main quality groups of handwritten texts are texts written with the block letters and those written with the cursive handwriting. The cursive handwriting, in its turn, falls into the separate letters handwriting and joined-up handwriting.

The most difficult problem is the recognition of joined-up handwriting because in this case a variability of handwriting samples for different letters is very high and the task of identifying the borders of each letter becomes very complicated. The volume of vocabulary, i.e. a number of different words in the text, is also very important. Vocabularies of authors' handwriting styles can be of great value, when literary texts are recognized. Such vocabularies are not comprised only of a large number of words used in a literary text but can also include sets of the author's calligraphy variants (ways of writing of different letters and links between them—ligatures) in order to define the correctness of this or that word. The use of computers to recognize the illegible handwritten texts can significantly speed up the process of finding the letters, ligatures, and words and increases an objectivity of comparison of the reference fragments of handwritten texts (taking into account the variability of handwriting samples) with ones under recognition.

2.3 Analysis and Identification of Handwritten Texts: Problem Formulation

Reading the handwritten drafts is often a very difficult task. Let us take draft manuscripts written by A.S. Pushkin and A.S. Griboedov as an example. These manuscripts are kept at the Manuscript Department of the Institute of Russian Literature (Pushkin's House) of the Russian Academy of Sciences. Just looking at the reproduction of Pushkin's draft manuscripts in the Big Academic Collection of his works (16 volumes printed in 1937–1949), one can see a huge number of notes and <illegible text> signs, that bespeak of the uncertainty of editors' interpretation of the author's handwriting [54].

There are longstanding academic disputes about some authors' handwritten drafts. Moreover, the samples of writing of some letters can greatly vary depending on the speed of writing and the writing tool used (goose quill, pen point, or pencil). A textual analyst has to create a vocabulary of various samples of the author's

calligraphy (ways of writing separate letters and links between them—ligatures) in order to read correctly this or that word. Figure 2.1 shows examples of A.S. Pushkin's handwritten drafts of the “Poltava” poem (courtesy provided by the Manuscript Department of the Institute of Russian Literature of the Russian Academy of Sciences).

Usually, while the analyzed fragment of handwritten text cannot be interpreted unambiguously, a textual analyst performs the following steps:

- Step 1. Selection a possible combination of letters in accordance with the grammar rules, semantic content of the text, and known individual specifics of author's writing.
- Step 2. Searching the author's fair copies to pick text fragments that matches the selected combination of letters.
- Step 3. Finding a fragment that matches the selected example from the analyzed handwritten text.
- Step 4. Visually comparison the sample with the recognized/analyzed fragment of handwritten text.
- Step 5. If the fragments do not match, repeats Steps 2–4 or maybe even Steps 1–4.

The use of advanced computer means and technologies can significantly speed up the process of finding samples (Steps 1 and 2), facilitate in objective integral assessment of a degree of closeness of compared fragments, and reduce the subjective judgment due to a human factor.

The automatic analysis and identification of illegible fragments of handwritten text requires to solve a number of problems [55–60]:

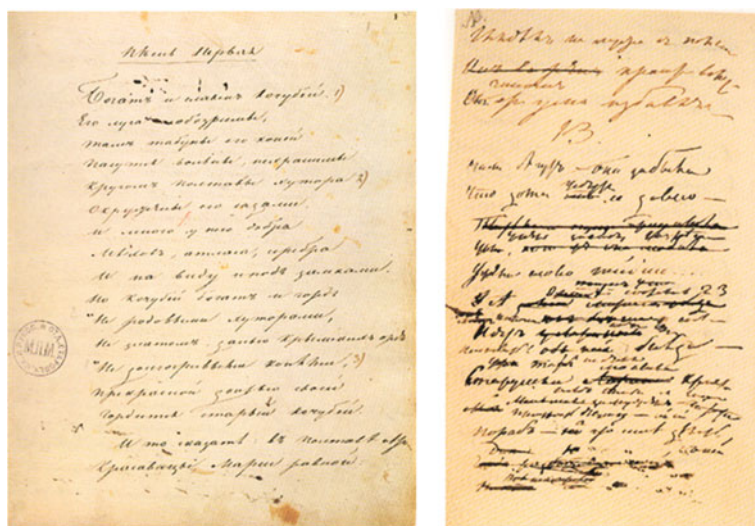


Fig. 2.1 Examples of A.S. Pushkin's handwriting

- Identification of the type of representation of the script fragment using the bitmap binary image with the suppressed brightness distortions and vector representation of an image.
- Analysis and selection of acceptable geometric transformations of an image in order to achieve the best possible matching.
- Development of methods for assessment of a degree of closeness of images during identification.
- Development of architecture for a computer system to store and retrieve information needed for the handwriting analysis and identification of illegible fragments of handwritten text.
- Creation of a database of alphabets based on samples of the author's handwriting taking into account the variability of author's calligraphy and difference of samples of the same letters found in the different author's texts.
- Creation of a bank of efficient methods for the recognition, handwriting analysis, and identification of handwritten texts.
- Design of a system of queries to work with the bank of methods and database of alphabets based on the author's handwriting samples and ensures access to them using prospective technologies of augmented reality through the intuitive interface.

Consider some of the mentioned above tasks in details. We will illustrate studies and experiments with examples of fair copies and drafts of handwritten scripts written by A.S. Pushkin, the classic of Russian literature, whose works always have a big number of non-text fragments and noises.

2.4 Formation of Alphabets Based on Handwriting Samples

Consider a formation of alphabets using samples of the author's handwritings. This is one of the most important and time-consuming tasks that requires direct involvement of specialists in textual analysis. The main procedures that need to be implemented to form an author's handwriting alphabet are the following:

- Input of handwritten text. Samples of handwritten text are represented on a computer screen as black-and-white or color bitmap images.
- Pre-processing of images. Elimination of image defects and noises on the background, and selection of text zones for analysis.
- Text segmentation that includes a breaking the text into the separate lines, lines into words, and words into letters and ligatures. Segmentation should be carried out more than once in cases of controversies in the interpretation of analysis results, since the intervals between letters in handwritten texts are often wider than those between words, whereas words can be connected with each other.

- Saving the letters, ligatures, and words as the author's handwriting samples in a database.

Selection of handwritten text fragments can be made using either widely known computer software (bitmap graphics editors like Adobe Photoshop or GIMP as an open source software) or specialized applications with a range of functions for textual analysis (horizontal line adjustment, selection of different combinations of segments, etc.) with the intuitive interface.

To address the task of alphabet formation, a technique for pre-processing of images of handwritten text fragments with elimination of image defects and background noises was developed and tested. The most commonly used tools for input of static visual information are the digital cameras and scanners (hand and flatbed).

Consider a still image on a flat media—rectangle sheet of paper (Figs. 2.1, 2.2, and 2.5). Our input device is a flatbed scanner that allows to obtain both color and grayscale images of various resolutions. The comparative analysis of the color, grayscale, and binary images of handwritten text showed that the color images do not have noticeable advantages over grayscale ones in terms of use for text recognition [57, 58]. Based on the conducted experiments, it was found that the most suitable resolution for analysis of images of handwritten texts is 200–300 pixels per inch (dpi). Further increase of resolution does not lead to noticeable increase in an image quality, while its decrease impairs image representation on computer screens and sometimes leads to the loss of small text elements.

Preliminary image processing, as a rule, includes the suppression of noises and image binarization. The latter operation is needed for segmentation, i.e. selection of

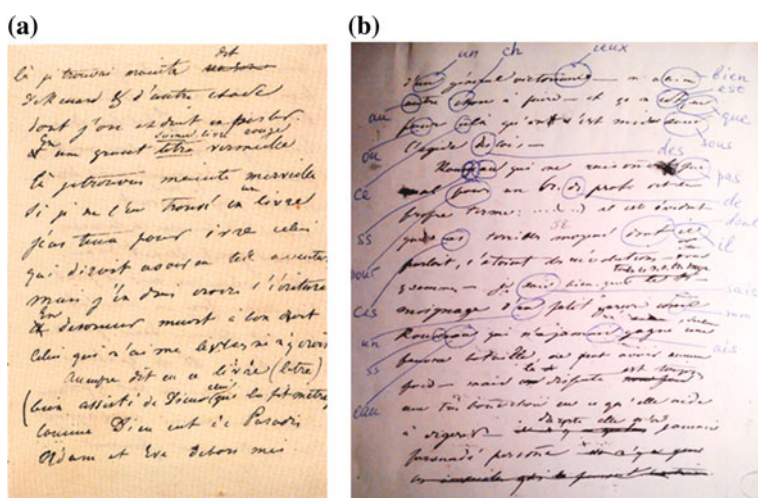


Fig. 2.2 Parts of “Roman de Renard”, a French literary classic of the 12–13th centuries: **a** original image, **b** marked image

the separate fragments consisting of adjacent pixels. Noises in the scanned bitmap images can be divided in two groups:

- Technical caused by errors in the operation of image acquisition, transfer, and storage equipment.
- Real ones in the form of sheet of paper defects (spots, fold lines) and author's notes (corrections, strikethroughs, and images).

The analysis of images obtained from a scanner showed virtually complete absence of noises of the first group. On the other hand, the problem of automated identification and elimination of noises in the second group is an extremely challenging task. In the first place, it is due to the need for formalization of attributes that can be used to identify image areas, which should be considered noises. At the stage of testing of a textual image pre-processing technology, the elimination of such noises should be carried out by a specialist in textual analysis in the interactive mode. Any graphics software that allows to change the brightness of pixels in bitmap images can be used for this operation.

Image binarization methods can be considered as non-linear pixel-by-pixel transformations with simple image transformation algorithms, where the main problem is the selection of binarization thresholds. Existing thresholding methods try to adapt to different brightness levels of separate image fragments. Most of these methods are based on the analysis of brightness histograms [61].

Brightness histogram of an image depicting several objects on a uniform background has two maximums, one of which corresponds to the brightness of object pixels and the another reflects the brightness of background pixels. There are many methods of thresholding using image brightness histograms [61]. For example, the histogram's global minimum found between two highest maximums can be used as a threshold. This method produces acceptable results if distinct maximums take place. The experiments show that a slight displacement of the threshold in any direction can lead to a significant change in the binarization results. This drawback can be compensated by a method that calculates the threshold as a weighted average of the brightness histogram [61]. This method does not require to search the histogram extremes that significantly reduces a processing time.

Image segmentation is the process of breaking an image into separate components that are valuable for analysis. In our case, this is the process of breaking lines into words and then words into letters and ligatures. The segmentation algorithm consists in identification of similarities between separate pixels of an image and finding the homogenous areas. The main difficulty in segmentation lies in the determination and formalization of the homogeneity. As a result of segmentation, each pixel should be attributed to a segment (number of segment), which it relates to (background is a zero segment), where the number of segments is unknown beforehand. The image is segmented into a number of homogenous areas by some attribute S , which qualifies the similarity of elements of each area. In our case, this attribute is the brightness. For a discrete image, the neighboring (adjacent) pixels are grouped in the homogenous areas with similar brightness.

The above said is true, when the image fragments corresponding to the separate characters are separated with background color and do not linked with each other. For handwritten texts, only words are separated with background color and even then not in every case. On the other hand, even a way of writing of some letters in a script can sometimes look like separate letters. For example, parts of Cyrillic letters “й” or “ф”, which in A.S. Pushkin’s scripts looks like letters “c” and “p” written one after another, are separated with background, and, therefore, can be recognized as separate symbols if the segmentation is executed in completely automatic mode.

Taking into account the interactive work of textual analysts (in the process of alphabet formation), we propose the following sequence of actions on the example of a small portion of French text (Fig. 2.2). After capturing an image of handwritten text with a flatbed scanner in black and white mode and resolution of 300 dpi, a text analyst uses the graphics editing software in order to select the separate line or word, removes noises (underlines, corrections, etc.), and puts dividers of letters and ligatures in the image changing the brightness of pixels that relate to the text (Fig. 2.2).

The edited text fragment is loaded into the segmentation software that automatically converts the image into binary format using the weighted average threshold of the brightness histogram or random threshold set by the operator, determines the relevance of pixels to separate segments using a two-pass algorithm, and calculates the number of segments. The textual analyst uses the segmentation software to look through the selected segments or groups of segments and saves them as separate images into a database specifying all necessary attributes (uppercase or lowercase letter, number of the page and line in the text, etc.).

The proposed segmentation algorithm includes two passes of each image and minimized linking table. The algorithm is used in the information storage and retrieval system that allows to form and view the script segments, merge segments into groups, and save the letters, combinations of letters (ligatures), and fonts of scripts in the database. The structure of this information storage and retrieval system is described in Sect. 2.6.

2.5 Methods of Bitmap Binary Image Based Graphological Analysis and Identification

Two approaches, representing in Sects. 2.5.1–2.5.2, were applied for analysis and identification of the bitmap binary images. Consider them in details.

2.5.1 Method of Skeleton Transformations of Letters, Ligatures, and Words

Two main approaches are used to construct an image skeleton—the skeletal transformation itself and the so-called “thinning” algorithm. Both of these approaches are applied if the relative position of strokes is the most important thing in an image. Usually, the skeleton transformation is a sequential removal of the maximum number of image pixels that allows preserving the general outline of that image. The process of highlighting the skeleton points is often associated with burning. Let us suppose that an object’s outline was set on fire and the flames spread along the normal towards its border. Then the points, where the flame fronts meet, will be considered as parts of the skeleton. This approach is called “thinning”.

Another possible definition of the skeleton is the locus of the centers of circles covering the object. The task of processing a handwritten text imposes certain specificity on the skeleton transformation algorithms applied. The experiments conducted to produce skeletons through different methods showed that the “thinning” algorithm is the most accurate way of solving the problem.

The image edge detection algorithms are also of certain interest for textual analysts. There are many of those algorithms but their basic idea is about the same. An original image is considered as a function of the $f(x, y)$ coordinates. If we cut an image in an arbitrary direction, then the edge of the image fragment will match the brightness jump with respect to the function $f(x, y)$ defined on a straight line. With respect to the gradient and Laplacian, this will correspond to maximum and zero values. Most of the existing edge detection methods are reduced to the discrete approximation of gradient or Laplacian. The Prewitt, Sobel, and Canny filters and the direct Laplace approximation were used to obtain the object contour images. The studies have shown that these algorithms produce the similar results. Therefore, the algorithms that use the scale selection, morphological or wavelet transformations do not seem appropriate.

2.5.2 Method Based on Calculation of the Hamming Distance Between Two Binary Images

The software algorithm was implemented to study the method based on the calculation of the Hamming distance. This algorithm allows to compare any two segments, which in the given software implementation should be presented in black and white colors [57, 60]. One black and white image of a handwritten fragment is added into the software application from a file, and another image is taken from the database. When an image is uploaded, the parameters that will later be used to for analysis are calculated. Those are the segment area and shape center. The segment shape center is marked by the intersection of two red lines; its coordinates and area are displayed in a popup window, when a popup window is hovered over the image.

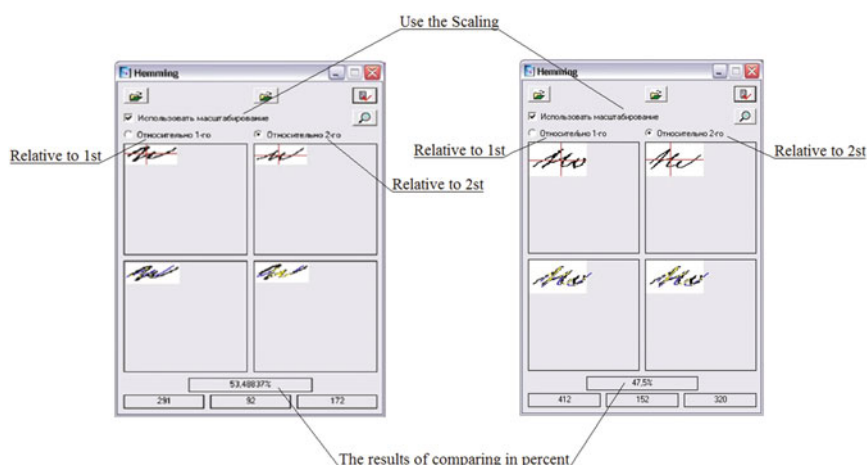


Fig. 2.3 Results of the comparisons of individual symbols and symbol combinations

The main application window, which allows to compare two manuscript fragment, is shown in Fig. 2.3. These fragments can be seen in the autograph by A.S. Pushkin illustrated in Fig. 2.1.

The comparison allows to create two combined images and show their matching degree. The first combined image shows the initial position of the segments. The initial position corresponds to the position with two shape centers combined. The second image (on the right part in Fig. 2.3) shows the result generated by the application. The segment mutual offset providing the maximum overlap area is obtained. The text box below these images displays their matching degree. Of course, it only makes sense to compare the segments that have the similar shapes, otherwise, the result of such comparison can be very poor and a feasibility of such comparison becomes highly questionable.

The algorithm can be divided into three stages:

- Calculation of the image parameters (area and shape center).
- Identification of a mutual image offset value, wherein the Hamming distance between the images is minimal.
- Calculation of the matching degree of the two images.

Two moments of the zero-order and first-order are calculated during the first stage. The zero-order moment corresponds to the area. The shape center is calculated using the OX and OY axes. An offset is calculated during the second stage as a combination of two shape centers. The Hamming distance is calculated based on this offset. Single shifts in different directions on both axes are performed in a cycle afterwards. The Hamming distance is defined for each shift. All distances are compared and the offset corresponding to the shortest distance is chosen. The distance should be less than the one obtained in the previous stage. This operation is

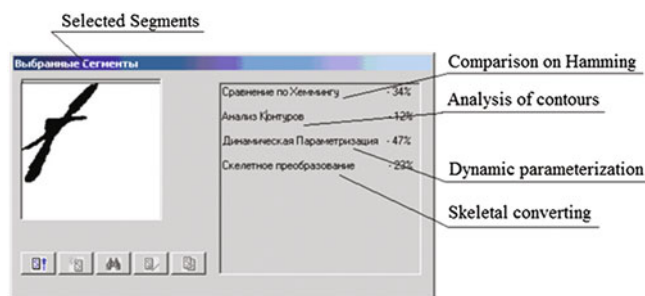


Fig. 2.4 Results of the analysis of the “F” symbol

repeated until the offset with the minimum possible distance is obtained. This offset is the result of the algorithm.

During the third stage, two images are combined using the resulting offset, and their matching degree is determined. The testing showed that the algorithm is fast enough and can be applied to compare image arrays. Further modification of the algorithm will make it capable of comparing not only black and white but any type of images. It should be noted that the segments ought to have a number of matching parameters (scale, color, orientation, etc.) for the algorithm to run properly. Figure 2.4 shows the result of the analysis made in the application specifically designed for the study of the analysis and identification methods. A textual analyst can check the given parameter combination using a special-purpose menu.

Consider the methods of the graphological analysis and identification representing in Sect. 2.6.

2.6 Methods of Graphological Analysis and Identification Based on Vectorization of Bitmap Images

Along with the methods and algorithms of bitmap transformation described above, we propose the following methods of analysis and identification of handwritten fragments [58, 62, 63]:

- Vectorization of bitmap images of handwritten texts.
- Vector dynamic parameterization.

Three groups of methods can be distinguished at the level of symbol identification. Conventionally, they can be called the reference, structural, and feature methods. The first group is formed by the reference methods. They are based on the comparison of a recognized character with a set of prototypes—reference symbol images. The second group of methods uses the information about the structural interrelations of graphemes or elementary parts of a symbol: vertical and horizontal strokes, loops, and their parts. In the third case, the recognition process is based on

the analysis of a set of invariant features that characterize the symbol, such as its size, number of loops, position relating to the line, etc.

Note that the feature detection phase is crucial for the effective operation of identification algorithms. First, it includes the significant data compression (matrix, describing the symbol image, is converted into a much smaller set of features) and, second, identification of relations between elements of object images that are characteristic to each class (intra-class invariant search).

Some of the features used in existing detection and identification systems are shown below. Simple geometric feature is the ratio of vertical and horizontal size of an object. Other geometric signs describe the integral properties of top, bottom, left, and right symbol profiles. Statistical features are associated with the average number of intersections of the symbol with vertical and horizontal lines. The symbol image is divided into equal horizontal areas. In each zone, the feature value is calculated as the average number of intersections of the symbol with all horizontal lines in this area. Topological features reflect the presence of jumps (sudden changes) of symbol profiles, as well as the number of internal topological areas (holes) on the object image. In classical handwriting, this number equals 0 for Cyrillic letters “к”, “л”, “м”, “н”, 1 for “а”, “б”, “о”, “я”, and 2 for “в” and “ф”. In practice, these relations are not as strict; however, the statistical dependencies are maintained. Many other features can be offered besides the ones mentioned above, for example, the center of gravity of letters, presence of the vertical, horizontal or another symmetry, direction of the letter bypass, etc. The number of potential features increases if the bitmap letter representation is replaced with vector images and the static representation is substituted by dynamic representation. The proposed methods and algorithms are discussed in Sects. 2.6.1–2.6.2.

2.6.1 Method of Vectorization of Bitmap Handwritten Text Images

Identification of handwritten symbols requires the ability to use the symbol contour bitmap image to create its vector representation, i.e. the polygonal chain or spline that accurately approximates the image. In this regard, the orientation field (direction field)-based method of vectorization of bitmap handwritten text images was designed. Orientation fields, that is, vector fields describing the orientation of image contours, are widely used in the recognition of bitmap images mainly formed by the shape contours. In particular, this technique is used for fingerprint recognition.

Description of the implemented algorithm. Bitmap black and white handwritten text images are the input data for the algorithm. It is assumed that the image resolution is at least as thick as the line or stroke. Besides, it is deemed that a line is darker than the background; otherwise, an inverse (negative) image can be used. A fragment of the manuscript by A.S. Pushkin shown in Fig. 2.5 can be considered example of such image. This is the beginning of the entry to a poem “The Fountain

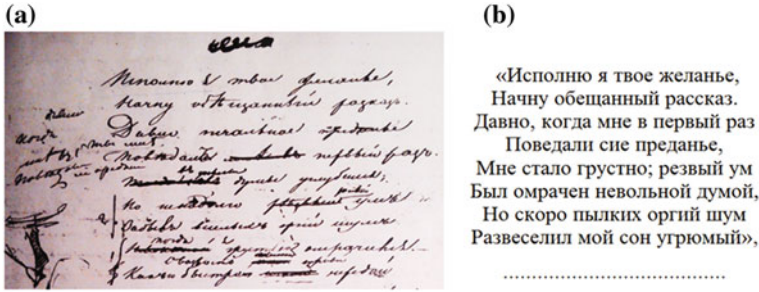
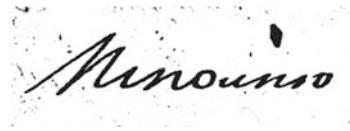


Fig. 2.5 Beginning of the entry to a poem “The Fountain of Bakhchisarai”: **a** fragment of the manuscript by A.S. Pushkin, **b** printing analogue

Fig. 2.6 Bitmap image of the word from the manuscript



of Bakhchisarai”, which was not included in the final text. Figure 2.6 shows a bitmap image of the first words of this piece “cut out” from the text for further processing and analysis.

The implemented processing algorithm includes the following stages:

- Pre-filtering of the bitmap image.
- Construction of the orientation field for this bitmap image.
- Filtering and extrapolation of the direction field.
- Searching for and tracking of image contours using the created orientation field.

The core idea of the algorithm is to trace the contours in the original (or pre-filtered) image with regard to the orientation field that provides information about the direction of a line or a stroke near a given point.

Image pre-filtering. A standard initial step for all image tracing algorithms includes image normalization and application of blur filter with small blur radius (of about one pixel).

Creation and representation of the orientation field. The filtering and extrapolation require the orientation field in any given point to be a smooth function of the image brightness values near this point.

Mathematically, the orientation field may be described by Eq. 2.1, where $\{A(x, y)\}$ are the tangent angles to level lines, $I(x, y)$ is a brightness function, $angle$ is an angle between the vector and the OX axis, mod is a calculation of the remainder of division.

$$A(x, y) = \left(\text{angle}(\text{grad}(I(x, y))) + \frac{\pi}{2} \right) \bmod \pi \quad (2.1)$$

Obviously, the direct representation of the field in this form is not continuous at each point. Therefore, the following complex (vector) function $D(x, y)$ is proposed to represent the orientation field:

$$D(x, y) = \left(-\frac{\partial}{\partial x} I(x, y) + i \frac{\partial}{\partial y} I(x, y) \right)^2 \quad (2.2)$$

Here, the squared expression is a perpendicular to the gradient written in the form of a complex number. Squaring is performed in order to eliminate the differences between mutually opposite directions of the gradient, which correspond to the complex gradient values opposite in their signs. The direction field values must be the same.

The level curves angle $A(x, y)$ is associated with the $D(x, y)$ function through Eq. 2.3.

$$A(x, y) = \frac{\arg(D(x, y))}{2} \quad (2.3)$$

When working with bitmap images the partial derivatives of brightness functions are replaced by finite brightness differences of the horizontally and vertically adjacent pixels.

Orientation field extrapolation and filtering. If the direction field is created for handwritten text images similar to monochromatic ones, the vector function $D(x, y)$ is noticeably different from zero only in stroke borders. Therefore, it is necessary to extrapolate the non-zero field values in adjacent areas in order to use the orientation field to track image lines and strokes. Image defects, such as ragged edges, add a noise to the direction field. The linear filtering algorithm with subsequent normalization is used to eliminate such noise.

Contour tracking. The following algorithm is proposed to track the contours:

- Step 1. A point lying on the contour is located.
- Step 2. A polygonal chain lying inside the contour (vectorized line), moving along the direction field in predetermined increments (about one pixel), is drawn from this point. The chain is completed, when it goes outside the contour.
- Step 3. Repeat Steps 1 and 2 until all points of the contour of a given bitmap image are used.

To improve the tracking accuracy along the field line, an additional offset proportional to the negative field gradient is added:

Fig. 2.7 Image of the orientation field and the result of image vectorization



$$r_{n+1} = r_n - \beta \text{grad}(I(r_n)) \quad (2.4)$$

where β is a small positive coefficient, r_n is radius vector of vectorized contour points.

This offset provides a slow drift of the point towards the lowest brightness level, i.e. to the middle of the contour line (assuming that the contour is darker than the background, that is, the brightness function value inside the contour is smaller than it is outside). An example of the results of the algorithm for a scanned fragment with $\beta = 0.01$ is shown in Fig. 2.7.

Thin lines here show the direction of the orientation field, bold lines are based on the image tracing results. The figure shows how the vectorized contours move along the lines of an extrapolated direction field. The direction field away from the contours looks chaotic, as the figure does not display the orientation field vector amplitude. If the Gaussian smoothing is applied to the direction field, the amplitude of randomly oriented areas decreases due to the mutual suppression. The described algorithm was implemented as a Java application that processes the image files of various graphic formats.

Based on the results of the computer experiment, the following properties of the proposed method of image vectorization using the orientation field can be noted:

- The method is insensitive to the additive white noise and small contour irregularities, since such defects correspond to chaotic orientation fields smoothed during the filtering process.
- The method allows to ignore the short contour breaks smaller than the extrapolation radius.
- The method allows to recognize successfully the handwritten lines with the “acute angle” elements, which are often found in handwritten text images. Unlike the skeleton methods generating the Y-shaped contours (Sect. 2.4) for acute angles, the orientation field-based method restores a V-shaped contour that is closer to the original path of a writing tool.

These properties allow to expect a more effective solution to future recognition problems.

2.6.2 *Method of Vector Dynamic Parameterization*

If an image is stored in a bitmap format, such important information as the order of writing of letters, pen speed, and its direction, is lost. The image vectorization method based on the orientation fields shown above provides some possibilities in this direction but it solves the problem only partially. The following is an alternative approach, which may be called the method of vector dynamic parameterization. This method allows to obtain more information about letter images using their vector dynamic representations. It creates additional possibilities for the formation of new diagnostic features, as well as the ways to use the thoroughly developed means of 1D analysis and statistics for processing and analysis of 2D letter images.

The purpose of this method is to restore the dynamic information about the movement of a pen, when writing a letter, including the indication of the start and end points, movement direction, and number of cyclic letter outlines. This method includes two stages [62, 63]. First, the image vectorization is done, during which an operator or a textual analyst uses a light pen or a mouse to outline the letter image as they had been written originally. This approach allows the substantial use of important but poorly formalized information about the ways the letters were written based on the given manuscript and personal and on the personal experience of an expert. Note that a vector image obtained through the orientation field method can be considered as the initial image. Such vectorization creates 2D array of Cartesian coordinates of image points (x, y) arranged in the order they were written. For convenience of further processing, it can be converted to a file, containing ASCII codes of coordinates. In a more complete version, such array can contain a third coordinate representing time. Second, the resulting 2D array can be visualized not only as the letter image but also as two separate curves $x(t)$ and $y(t)$. This allows to create the diagnostic features, such as the number of minima and maxima on the curves, inflection points, and spectral ratios. The possibility of calculating the correlation ratios between the curves $x(t)$ and $y(t)$ should be highlighted. In this case, thoroughly developed means of 1D analysis and statistics can be used to process and analyze 2D letter images. The procedure of performing letter scaling, inclination correction, and affine and projective transformations is greatly simplified.

The MATLAB software was used for practical implementation of the method. 2D numeric arrays created through the dynamic factorization and containing the letter coordinates are processed as standard package files. At the stage of letters' pre-processing, these arrays were standardized, i.e. the differences in the size and inclination of such letters were removed. Letters were rotated and compressed through affine transformations.

Before converting a letter, it is reasonable to calculate the numerical features, such as its width, height, the width to height ratio, as well as the center of gravity and the number of extrema of functions $x(t)$ and $y(t)$. Additionally, it is useful to count the number of internal topological areas ("holes") in the letter. For this purpose, five horizontal lines equidistant from one another were allocated in functions $x(t)$ and $y(t)$ and the number of intersections between the $x(t)$ and $y(t)$ curves and these lines

were counted. For convenience of further comparison of letters, they must have the same number of points. The experiments showed that a number of points required to adequately describe a letter ranges from 40 to 240. Accordingly, the standard length for $x(t)$ and $y(t)$ arrays was selected as 150 points. Furthermore, the array points should be evenly distributed throughout the period of writing a letter. The interpolation procedure providing an array of points equidistant from each other was used for this purpose.

The informative diagnostic features of the letter recognition include the correlation ratio between the $x(t)$ and $y(t)$ curves of compared letters. The application that compares a given letter with each letter in the matrix of reference letters was designed to calculate this ratio and identify the best matches. Afterwards, the correlation ratios between the $x(t)$ and $y(t)$ functions of compared letters are identified and added to a special-purpose matrix. This data, among with other diagnostic features, will later be used for statistical analysis.

One of the advantages of the vector dynamic representation is the ability to visualize each letter in the form of a 3D image with x , y , and t coordinates. The corresponding letter images look like a spatial spiral, where the first point corresponds to the beginning of a letter and last point corresponds to its end. An important geometric property of this representation is the absence of self-intersection points even if they were present in the original image. As an example, Fig. 2.8 shows the spatial representation of Cyrillic letter “a”.

The spatial representation makes it possible to compare two letters by the average distance between the spirals. The distance between the analyzed letter and the current reference letter is then calculated. When identifying the letters, the distances between the analyzed letter and each of the reference letters are calculated and recorded in a separate array. The position of the minimum element of this array is taken as the number of a most similar reference letter. The calculation of the

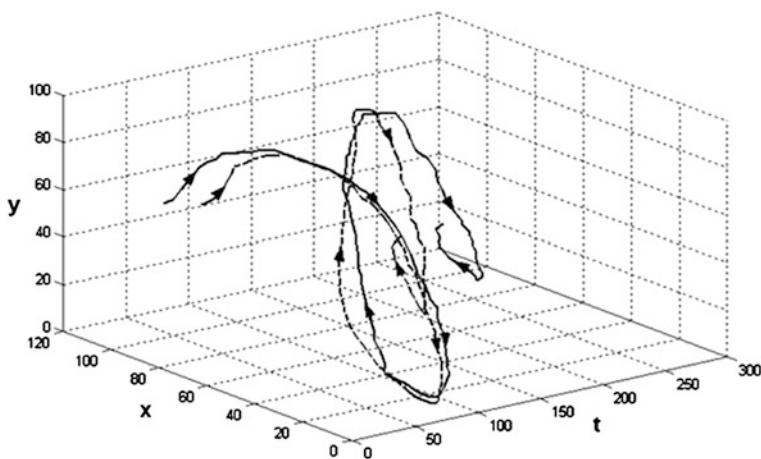


Fig. 2.8 3D image of two examples of Cyrillic letter “a”

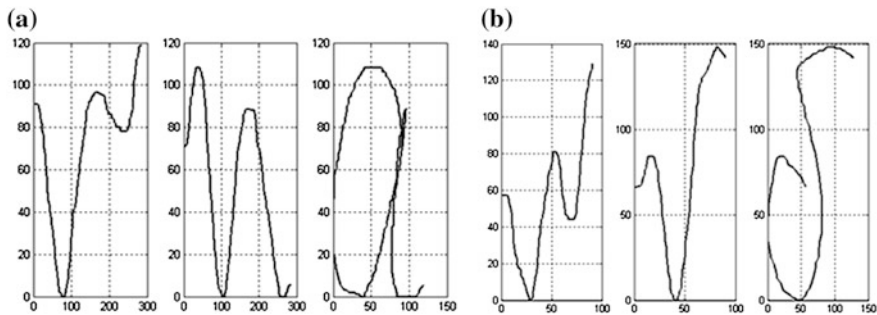


Fig. 2.9 Reference curves: **a** for Cyrillic letter “a”, **b** for Cyrillic letter “o”

correlation ratios and the geometric characteristics of 3D letter images essentially extends the standard set of diagnostic features used to analyze and identify the letters. The calculated diagnostic features are used as a basis for the table, into which the results of the comparison of the given letter with the reference letters are added. The feature table is a matrix, each row of which contains the results of the calculation of different characteristics for each letter. During the computing experiments, this table included the following features: the distance between letter images in the three dimensions, ratio of letter’s height to its length, correlation ratios, the number of extrema of $x(t)$ and $y(t)$ functions, and number of intersections between $x(t)$ and $y(t)$ functions and five fixed horizontal levels.

The features of an analyzed letter are identified and stored as separate variables in order to analyze the data contained in the table. They are later compared with similar features of the reference letters. The comparative results are recorded in the feature table consisting of 16 columns and 33 rows (the number of letters in the Cyrillic alphabet). The Euclidean norm is calculated to determine the quality of matching. As a result, the algorithm finds the number of a letter from the reference array that appears as the most similar to the analyzed one.

The reference letter images were created for the purposes of computer experiments (3–4 files for each letter). Figure 2.9 shows the reference curves $x(t)$, $y(t)$, and $y(x)$ for Cyrillic letters “a” and “o”.

The reference array contains references of all letters and different dynamic versions of their writing. The recognition quality during the computer experiments amounted to 85%. It can be increased by the way of more careful selection of diagnostic features and improvement of letter pre-processing, in particular, using the matrixes of affinity, projective, or other types of transformations.

2.7 Architecture of Information Storage and Retrieval System for Graphoanalysis and Identification of Handwritten Texts

The obtained results formed a theoretical basis for an automated information storage and retrieval system that is capable to recognize, decode, and identify the handwritten texts [55–57, 59, 60]. The system's architecture uses the classic client-server model for on-line applications. The system's structure consists of a number of components:

- Database.
- Server side.
- Client side.
- A set of plug-ins.

The database contains the author's handwritings, such as the alphabets, ligatures, lines of handwritten texts, and manuscript codes. The system of queries to database is divided into three groups of stored procedures providing both data access and storage security:

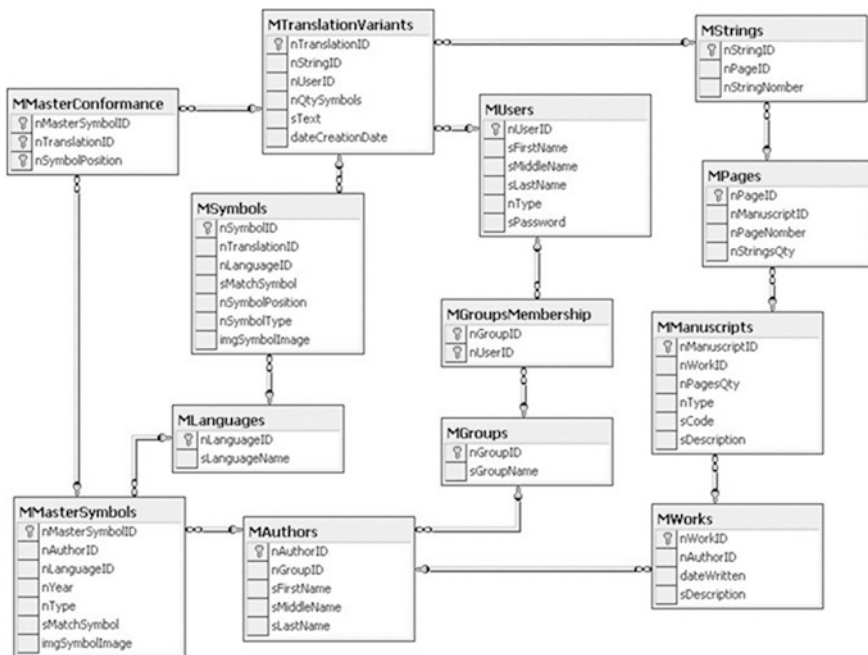


Fig. 2.10 Database structure

- The security group contains the user authentication procedures and interaction dialogs.
- The manuscript codes' group stores the procedures related to processing of manuscript data.
- The handwriting alphabets' group contains the procedures that work with samples of the letters, ligatures, words, and lines of manuscripts.

The handwriting database is worked under Microsoft SQL Server MDBS and configured to store images of the separate letters, ligatures, words, and lines of manuscripts and related information (author, year, name of work, line number, etc.). The database structure is shown in Fig. 2.10. Interaction with the database is carried out through the stored procedures.

Server part is the main component of the system. On the one hand, it works with the database directly through the stored procedures; while on the other hand, it interacts with users. Also, the server part allows users to run various analysis and identification procedures in the form of plug-ins. The user interface allows the registration of remote users and separate storage of the results of their work, as well as a simultaneous access to the general database to assist in reading of illegible manuscript fragments.

The development of a friendly and intuitive interface for specialists in textual analysis is a very important task. Two approaches for creation of the client side of the information system for graphoanalysis and identification of handwritten texts were considered:

- Standard Windows application [57].
- Windows application with the use of the augmented reality technology.

The software interface for textual analyst realized as standard Windows application [57] includes the algorithms for segmentation of handwritten texts into the



Fig. 2.11 Main window of the system's Windows application

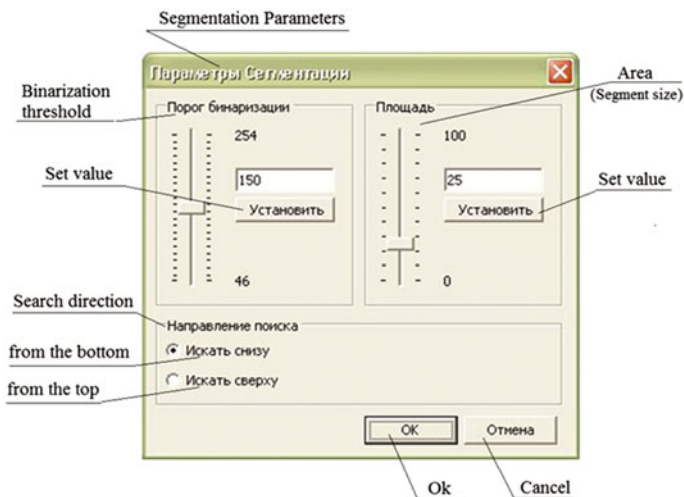


Fig. 2.12 Segmentation parameters

lines, words, ligatures, and letters and also involves an algorithm for preliminary processing that helps to remove the noises and heterogeneity of background of the scanned handwritten texts.

Consider the algorithm of interactive work of a textual analyst, while forming an alphabet based on a real example. The textual analyst uses the segmentation application to look through the selected text segments or groups of segments one by one and saves them as separate images into the database indicating all necessary attributes (uppercase or lowercase letter, line number, page number, etc.). The application's main window for segmentation includes the menu, toolbar, status bar, and area that shows the image that is being segmented (Fig. 2.11). Figure 2.11 shows a part of A.S. Griboedov's "Forgive us Fatherland" poem.

The toolbar repeats some menu commands. The status bar reflects information about the image that is being segmented, such as the image path and its size in pixels. The "Segmentation Parameters" window (Fig. 2.12) permits to change the binarization threshold, set the size of valuable segments, and choose the search direction. Segment size sets a minimum segment size. All segments that are smaller than the value of this parameter will not be analyzed. The search direction defines a way of the segmentation algorithm work. For example, for handwritten texts with a set to the right, it is recommended to choose the search direction from the bottom. The segmentation program automatically binarizes the text fragment using the average weighted threshold of the brightness histogram, determines the relation of pixels to separate segments using a two-pass algorithm, and calculates the resulting number of segments.

The results of a work of the segmentation program are shown in the Segments dialog window (Fig. 2.13). The toolbar includes the navigation buttons that switch from one segment to another, remove segments, and save segments in the database

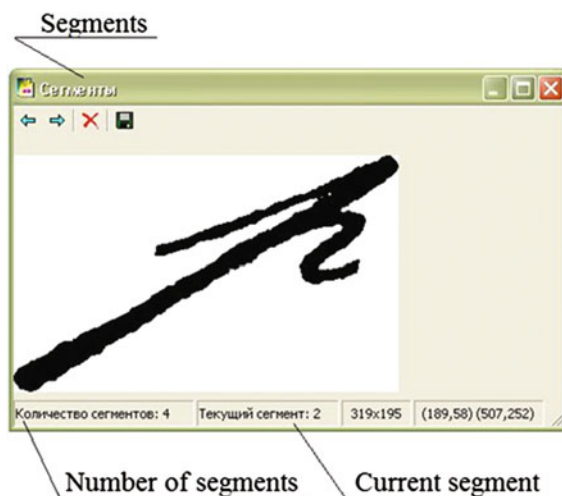


Fig. 2.13 Segments of a text fragment



Fig. 2.14 Merging of segments

or file in a hard drive. The Status Bar reflects the information about the number of segments, number of the currently viewed segment, its size and position in the source image.

To merge segments, just select the segments to merge and press the Merge button (Fig. 2.14). The selected segments and the currently viewed segment are highlighted.

As a result, the samples of handwritings of the classics of Russian literature of the nineteenth century A.S. Pushkin and A.S. Griboedov formed using the fair and draft copies of their manuscripts were created and saved in the database. The manuscripts that were used for this research were A.S. Pushkin's "Eugene Onegin" novel, "The Gypsies" poem, early drafts of the "The Fountain of Bakhchisaray" poem, as well as letters and poems by A.S. Griboedov (courtesy provided by the

Manuscript Department of the Institute of Russian Literature of the Russian Academy of Sciences) as a part of joint research studies [RFBR, grants 04-01-00464 and 04-07-90354].

The set of plug-ins includes the software algorithms and methods for analysis and identification of handwritten manuscript fragments. Due to the use of a unified set of interfaces by every module, it is easy to add new modules to the system, which makes the system flexible and scalable. The existing software product comprises a bank of methods for analysis and identification of illegible fragments of manuscripts including:

- Algorithms for skeleton transformation and selection of contours of separate letters and letter combinations.
- Vectorization of bitmap images of handwritten texts.
- Vector dynamic parametrization.

The realization of a bank of methods allows to use various types of skeleton algorithms and algorithms for selection of contour images as COM objects. The augmented reality technology can be applied as a prospective approach to the organization of textual analyst's interface.

While conducting the automated study of handwriting, even highly experienced textual analysts constantly refer to different parts of computer screen and printed materials. This leads to the fact that the users constantly switch their focus between the studied area and various information materials related to the studied handwriting that they need to correctly interpret. This extends the overall study time and increases the potential number of mistakes, let alone the sensorimotor and cognitive load on the users. One of the possible solutions of this problem is to place both images of illegible writing and its samples in the viewing area using the augmented reality technology, which in our case is visual [64, 65].

Vision-based Augmented Reality (AR) is a real-time technology for overlaying of digital objects (text, images, and audio) over the picture of the real environment displayed on a computer screen. The AR technology allows to obtain additional information contextually related to the objects in the environment. The AR interaction is carried out from the point of view of each separate user and in accordance with each user's individual settings. Examples of use of the AR in text recognition are given in [65–67].

The potential advantages of application of the augmented reality technology to handwriting recognition are mentioned below:

- Reduced time consumption and error rate.
- Reduced sensorimotor and cognitive load on the users (movements of head, eyes and hands, data interpretation).
- Possibility of real time cooperation of researchers in the events, where the expert advices are required (multi-user mode).
- Individual and cooperative practical training.
- Possibility to lower the level of skill requirements for experts.

Thus, the problem of handwritten text recognition is reduced to a visual comparison of background images representing a handwritten fragment (letters, ligatures, words, and lines) with overlaid sample images downloaded from the generated database.

The structure of the implemented desktop AR system includes:

- The computer, monitor and web-camera located perpendicularly to a table at a height of 50 cm. The operating zone of the camera is fitted with markers.
- Software includes MS Windows 10, Unity3D [68] for development of scenes to place and manipulate images, and Vuforia [69] to create augmented reality.
- A set of markers is used to study images of fragments of handwritten texts (letters, ligatures, words, and lines) both digitized and processed using analysis and identification methods.
- A set of samples (letters, ligatures, words, and lines) on a transparent background selected by a textual analyst from the database for comparison with the studied fragments of the manuscript.

Two modes called as the preparatory and operating modes are used during a working with the system. The preparatory mode includes:

- Digitizing of handwritten fragments (letters, ligatures, words, and lines) and processing of their images applying the developed methods, which are described in details in Sects. 2.4–2.6.
- Selection of images of handwritten fragments for markers.
- Recognition of markers using the Vuforia platform [69].
- Binding sample images from the database to the markers.

The operating mode involves:

- Placement of the marker under the web-cam.
- After the marker is recognized, it is displayed on the monitor under the ribbon of the sample images from the database.
- Selection of an image from the ribbon that (visually) overlays the analyzed sample.
- Moving, rotating or zooming of the sample images in order to visually fit the marker.
- If we find a matching, the symbol or fragment is considered as recognized; otherwise, another sample is selected from the database.

The experiments that included the analysis of draft autographs by A.S. Pushkin showed that the developed automated information search and retrieval system can be successfully used for the detection, decoding, and identification of handwritten texts given a highly variable author's calligraphy. In particular, it can be used by textual analysts or can serve as an evidence base for the correct recognition of draft autographs. During design the prototype handwriting recognition system based on the AR, the experience of the authors in development of applications in healthcare and cultural heritage was considered [70].

2.8 Conclusions

This chapter provides an overview of known handwriting recognition methods. The tasks of the computer analysis of handwriting and identification of illegible fragments of manuscripts are identified on the base of textual analysts' work with draft autographs of famous writers. The procedures of formation of alphabets composed of handwriting samples of the writers are considered using the developed methods of vectorization bitmap image and vector dynamic parameterization. The prepared alphabets are saved in the database taking into account a variability of calligraphy of the author's. The results of the identification methods for bitmap image and their program implementation on the examples of the literary manuscripts are given.

Also the description of the automated information retrieval system graphological analysis and identification illegible fragments of manuscripts is given. The system interacts with the database, which is pre-filled with a collection of different versions of the author's calligraphy (images of individual letters and their ligaments/ligatures). In the database, the fragments of text from a variety of literary manuscripts are saved. The database size is about 400 Mb. The system allows to use the methods described in the chapter for a quantitative assessment of the degree of coincidence recognizable fragment of manuscript with fragments of handwriting from the database. As a result a textual analyst receives the objective data that help him/her to read illegible handwritten fragment or conclude about affiliation of fragment of a specific author.

Researches were carried out on the example of manuscripts and draft autographs analysis of A.S. Pushkin and A.S. Griboyedov with the participation of the textual analysts from the Institute of Russian Literature (The Pushkin House) of the Russian Academy of Sciences. The use of advanced computer means and technology can significantly speed up the process of finding the handwritten fragments, facilitate the objective integral assessment of a closeness degree of the compared fragments, and reduce the subjective judgment due to a human factor.

Acknowledgements The reported study was funded by the Russian Fund for Basic Researches according to the research projects No. 04-01-00464 and No. 04-07-90354.

References

1. Avrukh, M.L., Birman, N.Y.: Automatic reading of typewritten texts. Proceeding/Automatic Reading of the Text. VINITI (1967) (in Russian)
2. Weinstein, V.S., Grigoryan, A.A.: The input device of graphics information into a computer. Automatic Reading of the Text. VINITI (1967). (in Russian)
3. Text Recognition. Available from: <http://mrtranslate.ru/download/textrecognition.html>. Accessed 20 June 2016
4. ABBYY Fine Reader Engine 8.1 for Windows. Available from: <http://www.interface.ru/home.asp?artId=6911>. Accessed 1 July 2017 (in Russian)

5. Lisin, S.: OCR-systems. Available from: http://old.ci.ru/inform16_02/p_22text.htm. Accessed 1 July 2017 (in Russian)
6. Edit of the Scanned Document in Photoshop. Available from: <http://photolessons.org/edit-scan>. Accessed 1 July 2017 (in Russian)
7. Andrianov, A.I.: Comparison of OCR-systems based on the accuracy of image analysis. *Bus. Inform.* **4**:44–50 (2009) (in Russian)
8. Bashmakov, A.I., Bashmakov, I.A.: *Intelligent Information Technologies: Textbook*. MSTU N.E. Bauman Publ. (2005) (in Russian)
9. Pint, E.M., Romanenko, I.I., Elichev, K.A., Semov, I.N.: Designing device for recognition of printed information. *Young Sci.* **4**, 235–238 (2015)
10. Misyurev, A.V.: Using artificial neural networks for characters recognition handwritten. In: *Proceedings/ISA RAS intelligent technology of input and information processing*, pp. 122–127 (1998) (in Russian)
11. Edwards, J.A.: Easily adaptable handwriting recognition in historical manuscripts. Technical Report No. UCB/EECS-2007-76, University of California at Berkeley (2007)
12. Dedgaonkar, S.G., Chandavale, A.A., Sapkal, A.M.: Survey of methods for character recognition. *Int. J. Eng. Innov. Technol.* **1**(5), 180–189 (2012)
13. Gaidukov, N.P., Savkova, E.O.: Review of handwriting recognition methods. In: *International Scientific-technical Conference of Students, Graduate Students and Young Scientists (Information Control Systems And Computer Monitoring'2012)* (2012) (in Russian)
14. Aggarwal, A., Rani, R., Dhir, R.: Handwritten devanagari character recognition using gradient features. *Int. J. Adv. Res. Comput. Sci. Software Eng.* **2**(5), 85–90 (2012)
15. Deshpande, P.S., Malik, L., Arora, S.: Handwritten Devanagari Character Recognition Using Connected Segments and Minimum Edit Distance. *IEEE TENCON'2007*, pp. 1–4 (2007)
16. Dholakia, K.: A survey on handwritten character recognition techniques for various indian languages. *Int. J. Comput. Appl.* **115**(1), 17–21 (2015)
17. Impedovo, D., Pirlo, G.: Zoning methods for handwritten character recognition: a survey. *Pattern Recogn.* **47**(3), 969–981 (2014)
18. Jieun, K., Yoon, H.: Graph matching method for character recognition in natural scene images: a study of character recognition in natural scene image considering visual impairments. In: *15th IEEE International Conference on Intelligent Engineering Systems (INES'2011)*, pp. 347–350 (2011)
19. John, J., Pramod, K.V., Balakrishnan, K.: Unconstrained handwritten malayalam character recognition using wavelet transform and support vector machine classifier. *Proc. Eng.* **30**, 598–605 (2012)
20. Kato, N., Suzuki, M., Omachi, S., Aso, H., Nemoto, Y.: A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(3), 258–262 (1999)
21. Khobragade, R.N., Nitin, A.K., Mahendra, S.M.: A survey on recognition of devnagari script. *Int. J. Comput. Appl. Inf. Technol.* **2**(1), 22–16 (2013)
22. Kim, H.C., Kim, D., Bang, S.Y.: A numeral character recognition using the PCA mixture model. *Pattern Recogn. Lett.* **23**(1–3), 103–111 (2002)
23. Kimura, F., Wakabayashi, T., Tsuruoka, S., Miyake, Y.: Improvement of handwritten Japanese character recognition using weighted direction code histogram. *Pattern Recogn.* **30**(8), 1329–1337 (1997)
24. Kressel, U., Schürmann, J.: Pattern classification techniques based on function approximation. In: Bunke, H., Wang, P.S.P. (eds.) *Handbook of Character Recognition and Document Image Analysis*. World Scientific, pp. 49–78 (1997)
25. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient based learning applied to document recognition. *IEEE Proc.* **86**(11), 2278–2324 (1998)
26. Mohamed, M., Gader, P.: Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(5), 548–554 (1996)

27. Niranjana, S.K., Kumar, V., Kumar, H.G., Aradhya, M.V.N.: FLD based unconstrained handwritten Kannada character recognition. *Int. J. Database Theory Appl.* **2**(3), 21–26 (2009)
28. Siddhartha, K.S., Dhir, R., Rani, R.: Handwritten Gurumukhi character recognition using zoning density and background directional distribution features. *Int. J. Comput. Sci. Inf. Technol.* **2**(3), 1036–1041 (2011)
29. Stamer, T., Makhoul, J., Schwartz, R., Chou, G.: On-line cursive handwriting recognition using speech recognition methods. *BBN Systems and Technologies* (1995)
30. Suen, C.Y., Kiu, K., Strathy, N.W.: Sorting and recognizing cheques and financial documents. In: Lee, S.W., Nakano, Y. (eds.) *Document Analysis Systems: Theory and Practice*, vol. 1655, pp. 173–187. Springer, LNCS (1999)
31. Suen, C.Y., Lam, L.: Multiple classifier combination methodologies for different output levels. In: Kittler, J., Roli, F. (eds.) *Multiple Classifier Systems*, vol. 1857, pp. 52–66. Springer, LNCS (2000)
32. Erosh, I.L., Sergeyev, M.B., Soloviev, N.V.: Methods for rapid character recognition are suitable for hardware implementation. *Inform. Control Sys.* **4**: 2–6, “Politehnica” Publication, St. Petersburg (2004) (in Russian)
33. Wakabayashi, T., Pal, U., Kimura, F., Miyake, Y.: F-ratio based weighted feature extraction for similar shape character recognition. In: 10th International Conference on Document Analysis and Recognition (ICDAR’2009), pp. 196–200 (2009)
34. Writepad Stylus. Available from: <http://androplanet.com/2804-writepad-stylus.html>. Accessed 1 July 2017 (in Russian)
35. Mozgovoth, A.A.: Problems of existing methods optical handwriting recognition. *Herald Voronezh State Techn. Univ.* **8**(7-1), 22–25 (2012) (in Russian)
36. Mori, S., Suen, C.Y., Yamamoto, K.: Historical review of OCR research and development. *IEEE Proc.* **80**(7), 1029–1058 (1992)
37. Rahman, A.F.R., Fairhurst, M.C.: Multiple classifier decision combination strategies for character recognition: a review. *Int. J. Doc. Anal. Recogn.* **5**(4), 166–194 (2003)
38. Buchkina, E., Soloviev, S.: Review of research on handwriting recognition problems. Available from: <http://www.textolog-rgali.ru/?view=articles&t=article1>. Accessed 1 July 2017 (in Russian)
39. Haykin, S.: *Neural Networks. A Comprehensive Foundation*, 2nd edn. Prentice Hall (1998)
40. Feng, Y., Fan, Y.: Character recognition using parallel BP neural network. In: International Conference on Audio, Language and Image Processing (ICALIP’2008), pp. 1595–1599 (2008)
41. Sahu, V.L., Kubde, B.: Offline handwritten character recognition techniques using neural network: a review. *Int. J. Sci. Res.* **2**(1), 87–94 (2013)
42. George, A., Gafoor, F.: Contourlet transform based feature extraction for handwritten malayalam character recognition using neural network. *Int. J. Ind. Electron. Electr. Eng.* **2**(4), 19–22 (2014)
43. Desai, A.A.: Gujarati handwritten numeral optical character reorganization through neural network. *Pattern Recogn.* **43**(7), 2582–2589 (2010)
44. Liu, C.L., Nakagawa, M.: Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition. *Pattern Recogn.* **34**(3), 601–615 (2001)
45. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recogn.* **36**(10), 2271–2285 (2003)
46. Liu, C.L., Sako, H., Fujisawa, H.: Performance evaluation of pattern classifiers for handwritten character recognition. *Int. J. Doc. Anal. Recogn.* **4**(3), 191–204 (2002)
47. Liu, C.L., Sako, H., Fujisawa, H.: Discriminative learning quadratic discriminant function for handwriting recognition. *IEEE Trans. Neural Networks* **15**(2), 430–444 (2004)

48. Liu, C.L., Fujisawa, H.: Classification and learning for character recognition: comparison of methods and remaining problems. Available from: <http://www.dsi.unifi.it/NNLDAR/Papers/01-NNLDAR05-Liu.pdf> (2006). Accessed 1 July 2017
49. Marinai, S., Gori, M., Soda, G.: Artificial neural networks for document analysis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(1), 23–35 (2005)
50. Oh, I.S., Suen, C.Y.: A class-modular feedforward neural network for handwriting recognition. *Pattern Recogn.* **35**(1), 229–244 (2002)
51. Pradeep, J., Srinivasan, E., Himavathi, S.: Diagonal based feature extraction for handwritten alphabets recognition system using neural network. *Int. J. Comput. Sci. Inf. Technol.* **3**(1), 27–38 (2011)
52. Arnold, R., Póth Miklós, P.: Character recognition using neural networks. In: *IEEE 11th International Symposium Computational Intelligence and Informatics (CINTI'2010)*, pp. 311–314. (2010)
53. Potapov, A.S. Artificial intelligence and universal thinking. St. Petersburg. « Politehnica » Publ. (2012) (in Russian)
54. Fomichev, S.A.: “The eyes above the letters glide” (To the problem of Pushkin’s textology). In: *VII International Conference on Pushkin’s “Pushkin and World Culture”: Proceedings/Institute of Russian Literature—NSU named after Yaroslav the Wise*, pp. 280–287. St. Petersburg—Velikiy Novgorod (2005). (in Russian)
55. Artemov, I.V., Reshetnikova, N.N., Soloviev, N.V.: The system architecture for the identification of literary handwritten texts. *SUAI Scientific session: Proceedings/Saint-Petersburg State University of Aerospace Instrumentation (SUAI)*. vol 2, pp. 414–415. St. Petersburg (2005) (in Russian)
56. Artemov, I.V., Reshetnikova, N.N., Soloviev, N.V.: On the tasks of the automated system of graphological research handwritten texts. In: *VII International Conference Pushkin’s “Pushkin and World Culture”: Proceedings/Institute of Russian Literature—NSU named after Yaroslav the Wise*. St. Petersburg-Velikiy Novgorod, pp. 287–294 (2005) (in Russian)
57. Artemov, I.V., Volkov, D.M., Kondakova, I.A., Nikitina, A.A., Reshetnikova, N.N., Soloviev, N.V.: Information retrieval system of graphological analysis and identification of handwritten texts. The certificate of registration of the industry development. *FSSI “State Coordinating Center for Information Technologies, an industry fund of algorithms and programs”*, pp. 1–10. Moscow (2006) (in Russian)
58. Mironovsky, L.A., Petrova, K.Y., Reshetnikova, N.N.: Prototyping of system analysis the handwritten documents by means MATLAB. In: *XIV International Scientific and Technical Workshop “Modern Technologies in Problems of Control, Automation and Information Processing”*, p. 299 (2005) (in Russian)
59. Volkov DM, Mironovsky LA, Reshetnikova, N.N.: Automated system of graphological research of manuscripts. In: *XIII International Scientific and Engineering Workshop “Modern technologies in problems of control, automation and information processing”*, pp. 482–483 (2004) (in Russian)
60. Artemov, I.V., Soloviev, N.V.: Analysis of coincidence the fragments handwriting text on the basis calculation of the Hamming distance. *SUAI Scientific Session: Proceedings/Saint-Petersburg State University of Aerospace Instrumentation (SUAI)*, vol 2, pp. 22–23 St. Petersburg (2006) (in Russian)
61. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 3rd edn. Prentice Hall (2008)
62. Mironovsky, L.A.: Redundancy, the invariance and symmetry in technical cybernetics. *Cybernetics and Informatics (50th anniversary of the section of cybernetics House of Scientists named A.M. Gorky RAS): Proceeding/Polytechnic University*, pp. 193–208 (2006) (in Russian)
63. Mironovsky, L.A., Shintyakov, D.V.: Interrelation Hankel singular values and frequency characteristics of linear systems. In: *Abstracts 14th International Scientific and Technical Workshop “Modern Technologies in Problems of Control, Automation and Information Processing”*, p. 300 (2005) (in Russian)

64. van Krevelen, D.W.F., Poelman, R.: A survey of augmented reality: technologies, applications and limitations. *Int. J. Virtual Real.* **9**(2), 1–20 (2010)
65. Rose, F., Bhuvaneswari, G.: Word recognition incorporating augmented reality for linguistic e-conversion. In: *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT'2016)* (2016)
66. Google Translate Adds 20 Languages to Augmented Reality Application. Available from: <http://www.Popski.Com/Google-Translate-Adds-Augmented-Reality-Translation-App>. Accessed 1 July 2017
67. Fragoso, V., Gauglitz, S., Kleban, J., Zamora, S., Turk, M.: Translat AR: A mobile augmented reality translator on the Nokia N900. In: *IEEE Workshop Applications of Computer Vision (WACV'2011)*, pp. 497–502 (2011)
68. Website Unity. Available from: <http://unity3d.com/>. Accessed 1 July 2017
69. Website Vuforia. Available from: <https://developer.vuforia.com/>. Accessed 1 July 2017
70. Website of Laboratory Computer Graphics and Virtual Reality SUAI. Available from: <http://guap.ru/labvr/projects/>. Accessed 1 July 2017

<http://www.springer.com/978-3-319-67993-8>

Computer Vision in Control Systems-4

Real Life Applications

Favorskaya, M.N.; Jain, L. (Eds.)

2018, XIV, 317 p. 137 illus., 86 illus. in color., Hardcover

ISBN: 978-3-319-67993-8