УДК 004.413:338

D.A.Dolotov

National aviation university

Software engineering department

Lecturer - Chebanyuk O.V., candidate of engineerings sciences, associate professor

HANDWRITTEN TEXT RECOGNIZTION BASED ON ANALYSIS OF MOTION VECTORS

У статті проаналізовано підходи до розпізнання тексту при використанні технологій рукописного вводу . Визначено особливості вимог до програмних модулів рукописного вводу, які використовуються у мобільних пристроях. Представлено опис розробленого методу та наведено аналіз ефективності його складових. За результатами експериментів робиться висновок про порівняльну ефективність запропонованого та існуючих методів для розпізнання тексту при рукописному вводі на мобільних пристроях

В статье проанализированы подходы к распознаванию текста при использовании технологий рукописного вводу. Определены особенности требований к программным модулям рукописного вводу текста, которые используются в мобильных устройствах. Представлено описание разработанного метода и приведен анализ эффективности его составляющих. По результатам экспериментов делается вывод сравнительную эффективность существующих методов распознания текста при рукописном вводе на мобильных устройствах.

Article reviewed approaches of using recognition text technologies of handwriting input. The particular requirements of handwriting text input software modules that are used in mobile devices are identified. The designed method and the analysis of the effectiveness of its constituents are presented in this article. According to the results of the experiments found the comparative effectiveness of existing methods to recognize text in handwritten input on mobile devices.

Keywords: handwriting recognition, motion analysis, mobile devises, motion vectors, user input.

Introduction

Touchscreen is an electronic visual display that can detect the presence and location of a touch within the display area. Nowadays touchscreens are widely used in devices such as all-in-one computers, tablet computers and smartphones. To input the text via touchscreen can be used virtual keyboard or handwriting recognition technology.

Handwriting recognition is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touchscreens and other devices [1]. So, user can input the characters by writing them on the touchscreen surface.

The task of developing methods for handwriting recognition especially important when we consider mobile devices (smartphones). To develop a method that allows you to efficiently handle the information entered on your mobile device; you need to take into account the specific requirements software for mobile devices. There are the speed of the text recognition, usage of minimum resources; database size should be as small as possible, openness (the easy way of making changes to the code), scalability, and ease of integration to existing systems ability to maintain individual peculiarities of handwritten style.

The analysis of existing systems of handwriting input

We make an analysis of existing systems that allow us to solve the problem of writing a text on compliance with the above requirements, which were formulated above

Authors in ref [1] proposed approach to the recognition of words by using multiple agent architecture for handwritten text recognition. They analyze ideas of text recognition in such a way.

Two main approaches are used to perform the recognition of handwritten cursive words. The first, called analytical, is a data-driven bottom-up approach in which letters are recognized before a lexical analysis is performed [2, 3]. To counteract the problem of letter segmentation before (without) recognition, several segmentation hypothesis must be managed, which makes in return the letter recognition module more complex since it must be therefore able to reject the bad segmentation hypothesis. However, the final decision can only

be taken by the lexical verification module. This scheme of recognition is also called segmentation/recognition. The second approach, called holistic, is a top-down approach with verification. In this approach, the segmentation into letters is counteracted by recognizing a word in its whole and by selecting word candidates in a lexicon. This approach leans either on the detection of holistic features in the word [4, 5].

Authors propose agent oriented architecture for handwritten text recognition.

Briefly speaking agents are entities that have the ability to communicate with other agents in the same environment, have an autonomous behavior that allows them to act according to their own goal and knowledge of their environment. Model of multiple agent architecture for handwritten text recognition allows distributed control, and is therefore better adapted to the implementation and the test of various strategies of control, e.g. various strategies of reading, when replaced in our context of constraints relaxation for handwritten text recognition. However, since cooperation between agents requires sharing the same common data about the problem, we have found that the major drawback of the multiple-agents architecture is the necessity for agents to incorporate the data into control messages. They have built an open platform called EMAC [6] that allows plugging expert treatments dedicated to handwriting analysis with the ability to share a common distributed workspace.

Another solution of on-line handwriting recognition tasks researchers presently use classification methods which are based on a Bayesian *generative* approach: hidden Markov model is represent a model a sequence of class conditional densities based on (and thus restricted to) a certain function class. A discriminant function is obtained in a second step using Bayes' rule. Indeed HMMs have proven to deal very well with the complex on-line handwriting data structure. This is usually a variable-size *sequence* of feature vectors that may have been distorted in particular ways, each vector computed from sampled coordinates of the pen tip curve [7].

The authors in [8] proposed usage of template database. When user input letter the system compare it with set of templates and gives most appropriate template. The drawback of usage of proposed decision for mobile system is volumetric calculations for finding the proper pattern.

Another well-known approach is usage neural nets [9]. Neural Nets and Hidden Markov Models are the popular, amongst the techniques which have been investigated for handwriting recognition. It has been observed that NNs in general obtained best results than HMMs, when a similar feature set is applied [10]. The most widely studied and used neural network is the Multi-Layer Perceptron [8]. Such an architecture trained with back-propagation is among the most popular and versatile forms of neural network classifiers and is also among the most frequently used traditional classifiers for handwriting recognition.

Common property, which made difficult to use neural nets for text recognition in mobile devices, is necessity of large data base for exact recognition of inputted character. For example is necessary to avoid many operations while recognition, which can discharge battery more quickly. These requirements are reflected in "minimum resources", "the speed of recognition" and "small database" criteria. Also represented requirements reflect other peculiarities of usage of mobile devices.

Let's consider all three considered approaches to the text recognition according to the peculiarities requirements to mobile devises.

Table 1. Criteria effectiveness handwritten modules for mobile devices

The criterion	**Multiply agent architecture**	**Template method (base from raster images)**	**Template method (base from vectors)**	**Trained neural network**
the speed of recognition	+	-	+	+
the accuracy of recognition	+	+	+	+
small database	-	-	+	-
minimum resources	-	-	+	-
open code	-	-	-	-
scalability and easiest for portability	-	-	-	-

Analyzing the table 1 we can see that existed recognition systems recognize characters [1,7,8,9] don't meet all peculiarities of requirement towards modules witch recognize text in mobile systems.

The closest approach is template method. That why it was decided, to make the ideas of template method like ground and develop method of text recognition improving the rest characteristics.

The purpose of the work is to create method for handwritten recognition, which will meet all user requirements, and account peculiarities of handwritten modules in mobile devices.

Background

Each user has different handwriting style and sometimes the algorithms of handwriting recognition cannot recognize input character correctly. To recognize correctly an input character we need a knowledge base, which contains all possible combinations of character size, shape and position on the screen. It is a hundreds of combinations which required a large amount of memory. And still there always will be such character input which is not stored in knowledge base [11].

On the figure 1 you see several possible inputs of character "A".

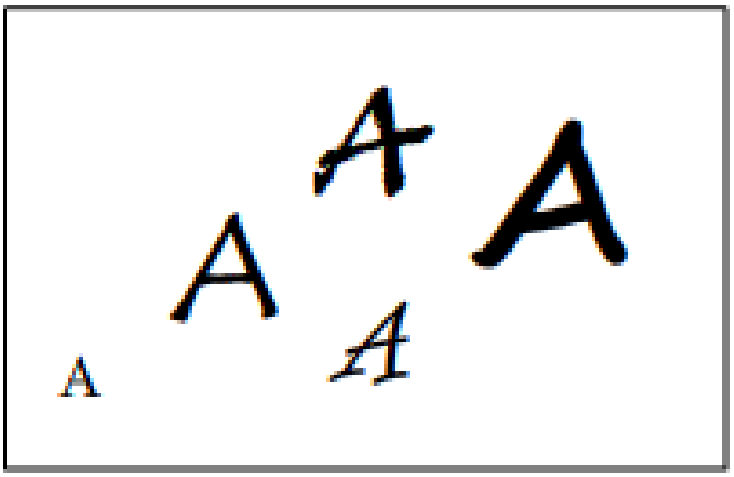

Fig.1 Different handwritten styles

Each time user makes the same 3 motions:

1) UP-RIGHT ↗

2) DOWN-RIGHT ↘

3) RIGHT →

But knowledge base should contain all of the possible images, to recognize the character correctly.

People can draw this character in different way, but amount of possible combination of motions will be much less than amount of possible input images. That's why database size of this system is much smaller that its analogs, which is extremely important for mobile devices. Ability of system to learn makes it flexible. System can be used by users with different handwriting style.

General overview of the proposed method for handwritten recognition

The main features of "Handwriting Recognizer" are:

1. Recognize input characters based on motion vectors
2. Increase quality of input character:

2.1 Remove redundant information

2.2 Smooth input image

2.3 Set recognition parameters to eliminate the effect of "shaking hand"

3. Ability to learn

Description of handwriting recognition algorithm

The process of recognition consists of the following stages:

1. Analyzing of user input.
2. Removing redundant information (noises).
3. Smoothing of the input image.
4. Analyzing of motion vectors.
5. Generation of character schema.
6. Searching of the character in knowledgebase.

Activity diagram is represented below (fig 2):

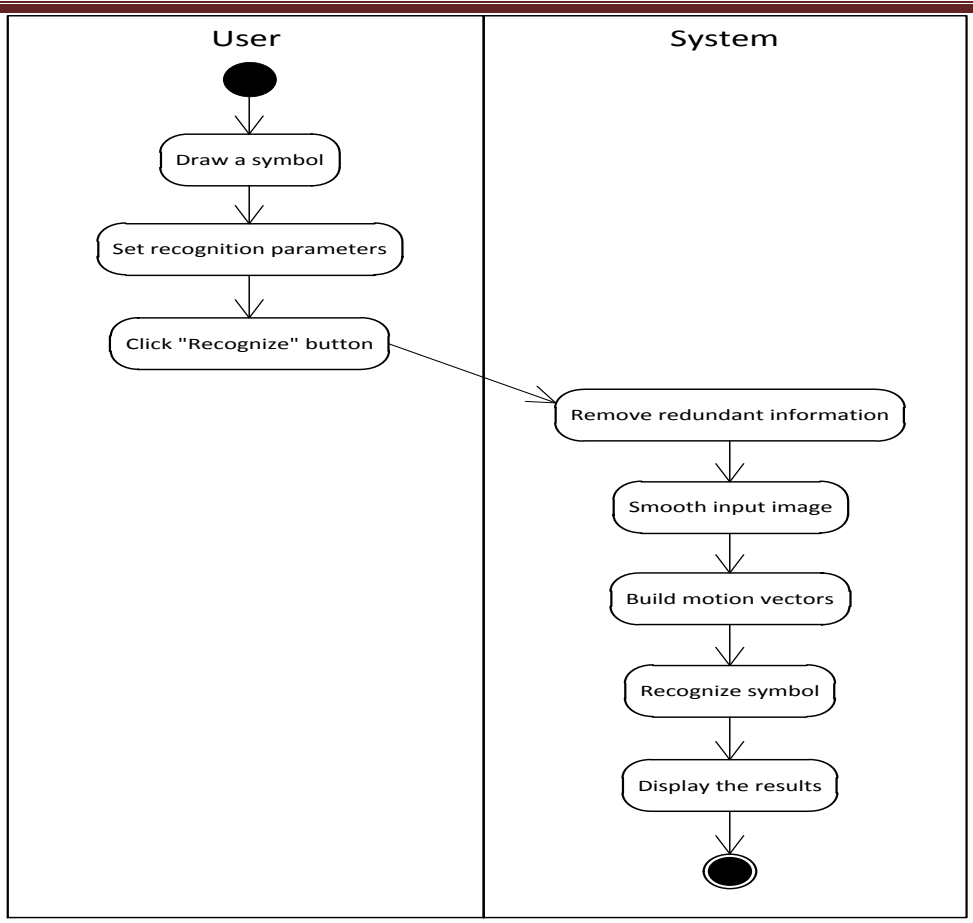

Fig.2 Activity diagram of recognition process

Let us consider these stages in more details. First, user draws a character in the input field (figure 3). System stores coordinates (X, Y) of each point of input character in the array. System also count amount of strokes, that user made, during drawing.

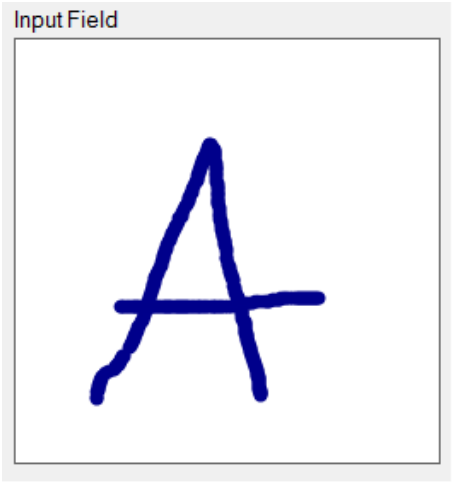

Fig.3 Initial symbol

Next step is removing of redundant information. For this purpose system uses parameter digitalizing step. Digitizing step – is the parameter, which specifies the interval between analyzed points. Step = 20 means that we will use only every 20^{th} point of input symbol (figure 4).

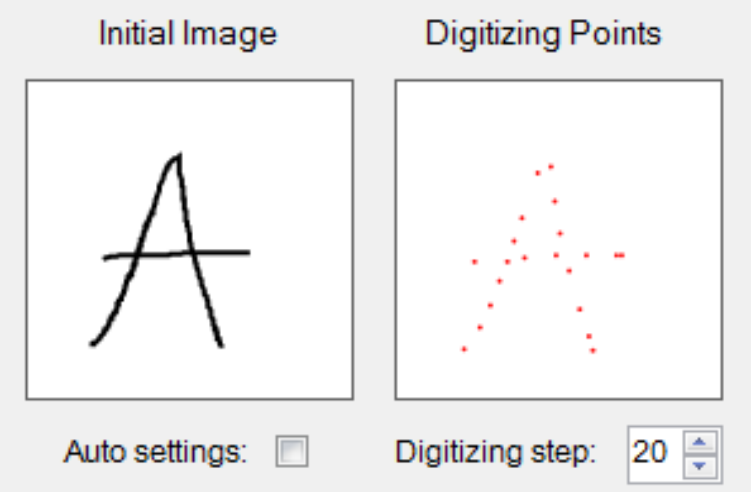

Fig.4 Removing redundant information

After redundant information was removed, system will smooth the input image. Smoothing increases quality of input character and eliminate the effect of "shaking hand". User can select one of two algorithms: linear three points or linear five point.

Linear three point algorithm [12] uses following formula:

$$\begin{aligned} X_1 &= (5X_1 + 2X_2 - X_3) / 6 \quad Y_1 = (5Y_1 + 2Y_2 - Y_3) / 6 \\ X_2 &= (X_1 + X_2 + X_3) / 3 \quad Y_2 = (Y_1 + Y_2 + Y_3) / 3 \\ X_3 &= (5X_3 + 2X_2 - X_1) / 6 \quad Y_3 = (5Y_3 + 2Y_2 - Y_1) / 6 \end{aligned} \tag{1}$$

Linear five point algorithm [12] uses following formula:

$$\begin{aligned} X_1 &= (3X_1 + 2X_2 + X_3 - X_5) / 5 \quad Y_1 = (3Y_1 + 2Y_2 + Y_3 - Y_5) / 5; \\ X_2 &= (4X_1 + 3X_2 + 2X_3 + X_4) / 10; \quad Y_2 = (4Y_1 + 3Y_2 + 2Y_3 + Y_4) / 10; \\ X_3 &= (X_1 + X_2 + X_3 + X_4 + X_5) / 5; \quad Y_3 = (Y_1 + Y_2 + Y_3 + Y_4 + Y_4) / 5; \\ X_4 &= (X_2 + 2X_3 + 3X_4 + 4X_5) / 10; \quad Y_4 = (Y_2 + 2Y_3 + 3Y_4 + 4Y_5) / 10; \\ X_5 &= (3X_5 + 2X_4 + X_3 - X_1) / 5; \quad Y_5 = (3Y_5 + 2Y_4 + Y_3 - Y_1) / 5; \end{aligned} \tag{2}$$

Example of image smoothing is presented on the figure 5.

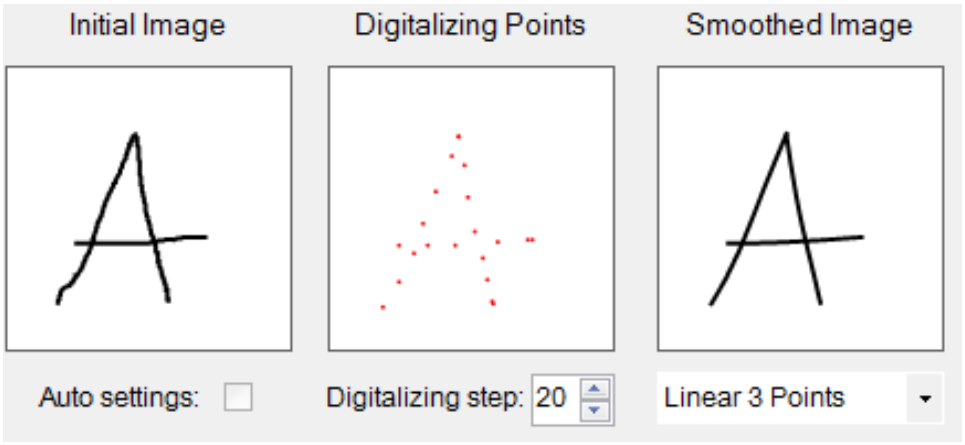

Fig. 5 Smoothed image

The fourth stage is the analysis of motion vectors. Vector - is the directed segment, which has a start point, end point and direction.

System recognizes 8 possible directions of motions and the Point motion. Each direction has corresponding index which is stored in memory:

UP ↑ = 1; UP-RIGHT ↗ = 2; RIGHT → = 3; DOWN-RIGHT ↘ = 4; DOWN ↓ = 5; DOWN-LEFT ↙ = 6; LEFT ← = 7; UP-LEFT ↖ = 8; POINT = 0.

Process of building of motion vectors consists of two stages:

1. Determination of vector direction
2. Joining of vectors with same direction

To define vector direction, system use Min Rotation Angle parameter that can be specified on UI.

Min Rotation Angle – is minimum angle between the vector and the axis needed to change the direction of the vector. This parameter increases quality of input symbol and accuracy of recognition.

For example, we have vector with angle α and Min Rotation Angle will be equal β.

If α > β, direction of the vector will be determined as *up-right* and vector will remain unchanged (Fig. 6.1).

If α < β, direction of vector will be determined as *right* and angle α will be set to 0. (Fig. 6.2).

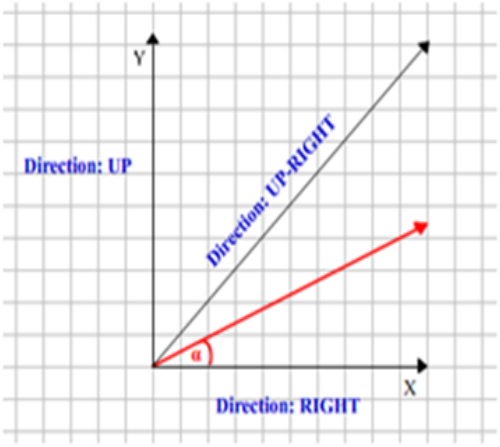

Fig.6.1

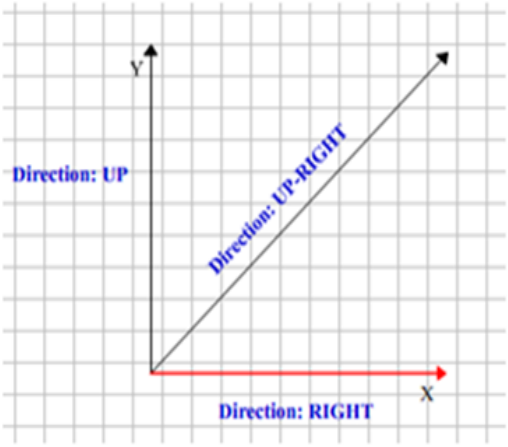

Fig. 6.2

Fig 6 The foundation of choosing minimum rotation angle

For example, user wants to draw symbol “I”, but he could not draw it accurately. In this case, without Min Rotation Angle specified, system will recognize two motion vectors: DOWN and DOWN-RIGHT (figure 7.1). But actually user wanted to make just one motion DOWN to draw letter “I”. With Min Rotation Angle parameter specified, system will recognize this motion correctly (figure 7.2).

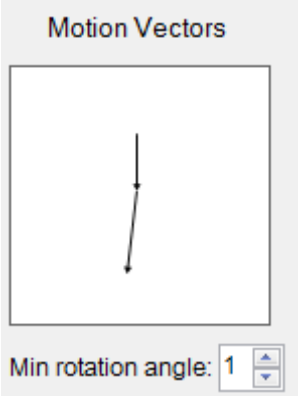

Fig. 7.1

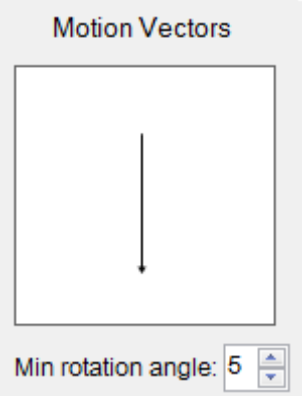

Fig.7.2

Fig 7 The influence of minimum rotation angle on appearance of resulting symbol

After direction for all vectors will be determined, related vectors with same direction will be joined. Joining operation works in the following way. Consider we have two related vectors: AB and BC. If they have the same direction, end point of AB will be equal to end point of BC, B = C.

For example, user has drawn a character “I”. System has smoothed the image and removed redundant information. After this we obtained and

array of points. If we will build the motion vectors based on these points, we will get a several vectors with the same direction (figure 8.1).

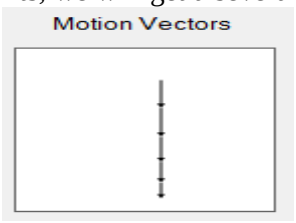

Fig 8.1

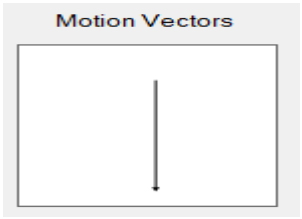

Fig 8.2

Fig. 8 Joining the motion vectors

But user actually made only one motion, so we need to join vectors with same direction (figure 8.2).

The result of these operations is a set of motion vectors, which represent the motion user made, when was drawing the image.

The final stage in recognition process is generation of character schema. Schema – is a set of motions represented in string format. As you remember, each direction of vector has its own index form 0 to 8. If user took away his hand, when was drawing, system uses "&" character. For example, user has drawn the character "A". He made three motions: up-right, down-right, right. Also he took away his hand before the last motion. The scheme of this character will be "24&3".

System searches this schema in the knowledge base and returns the symbol that matches the most (figure 9).

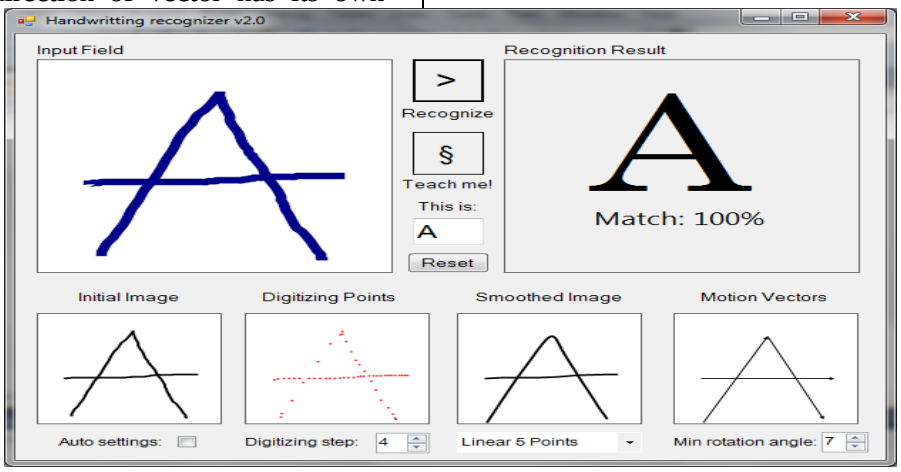

Fig. 9 The interface of program

Knowledge base - is a database which contains the possible variants of writing of a character. One character can have unlimited amount of writing variants. Database contains the symbols and theirs schemas. Storing of schemas will take much less memory than store the images (figure 10).

Knowledgebase

ID	Symbol	Vectors
3	A	2&4&3
27	A	2&4&7
30	A	6&4&3
4	B	5&346467
26	B	5&34634567
45	B	1246467
54	B	5&28286
58	C	8765434
59	C	765432
6	D	5&46
81	D	6&346
7	E	5&3&3&3
82	E	6&4&4&3

Fig. 10 Knowledge base

If system did not recognize correctly the input character, user can teach the system by specifying

correct character in corresponding textbox and pressing the "Teach" button. The character and schema will be added to knowledgebase.

Conclusions

In this paper, the method of recognizing handwritten text during user input was proposed. The described method takes into account all peculiarities of requirements for software handwriting text for mobile systems – notably the speed of recognition, the accuracy of recognition, small database, minimum resources, open code and scalability and easiest for portability Also the analysis of the existing system's possibilities to execute the task of handwriting text successfully was done.

The proposed method differs from multyagent systems[1] in the following capabilities: small database, minimum resources, open code and scalability and easiest for portability The method of text recognition using neural networks[9] has no such a requirements like minimum resources, open code and scalability and easiest for portability.

The method of text recognition using templates [7] do not meet the following requirements open code and scalability and easiest for portability.

Method of handwriting recognition, proposed by authors meets all requirements for this class of software for mobile devices can It can be used effectively like for improving existing software, handwriting, and while developing new applications that support this feature. The proposed method is implemented in software, using C # language. The created application tested under Windows Phone 7. When adjusting a small, considering that the language C # and Java - are related, you can convert the code in Java and use the design for the operating system Android.

References

1. Heutte L., NosaryA., Paquet T. A multiple agent architecture for handwritten text recognition Pattern Recognition 37 (2004) P.665-674

2. Tang Y.Y., Lee S.W., Suen C.Y., Automatic document processing: a survey, Pattern Recognition 29 (12) (1996) P1931–1952.

3. KimG., Govindaraju V., A lexicon driven approachto handwriting word recognition for real-time application, IEEE Trans. PAMI 18 (4) (1997) 366–379.

4. Shridhar M., Houle G., Kimura F., Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms, Proceedings of the ICDAR'97, Ulm, Germany, 1997, pp. 861–865.

5. Bramal P.E., Higgins C.A. A cursive recognition system based on human reading models, Mach. Vision Appl. 8 (1995) 224–231.

6. Hernoux C., EMAC, Un environnement Multi-Agents WamXemoire Collective, MXemoire d'ingXenieur, CNAM, CRA deRouen, Juin 999.

7. Claus Bahlmann, Bernard Haasdonk and Hans Burkhardt Computer Science Department

8. On-line Handwriting Recognition with Support Vector Machines—

9. A Kernel Approach Proc. of the 8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), pp. 49–54, 2002.

10. DeCoste D. and Schölkopf B.. Training invariant support vector machines. *Machine Learning*, 46(1/3):161, 2002. 1, 3

11. Muhammad Faisal Zafar, Dzulkifli Mohamad, and Razib M. Othman On-line Handwritten Character Recognition: An Implementation of Counterpropagation Neural Net

12. Tappert C.. Cursive script recognition by elastic matching. *IBM J.Res. Develop.*, 26(6):765–771, Nov. 1982. 2.1

13. Dolotov D.A., Chebanyuk O.V."Handwritten text recognition based on the analysis of motion vectors" Тези доповідей Міжнародної науково-практичної конференції аспірантів та студентів «Інженерія програмного забезпечення 2011». ст. 57-58.

14. Девид А. Форсайт, Джин Понс Компьютерное зрение. Современный подход. 928 стр., с ил.; ISBN 5-8459-0542-7, 0-13-085198-1; 2004 Вильямс

Відомості про авторів

Долотов Дмитро Анатолійович – студент 5 курсу факультету комп'ютерних наук Національного авіаційного університету, кафедра «Інженерії програмного забезпечення». Наукові інтереси – методи розпізнання образів, штучний інтелект.
E-mail: dolotov_dmytro@ukr.net

Чебанюк Олена Вікторівна - доцент кафедри «Інженерія програмного забезпечення» факультету комп'ютерних наук Національного авіаційного університету. Наукові інтереси – повторне використання коду, інженерія програмного забезпечення, доменний аналіз, методи розпізнання образів, технології програмного забезпечення.
E-mail: Elena.chebanyuk@livenau.net