COMPUTER ANALYSIS OF MULTIPLE-CHOICE EXAMINATION DOCUMENTS

Łukasz Karpowicz, Witold Malina, Maciej Smiatacz

Gdańsk University of Technology, ul. Narutowicza 11/12, 80-952 Gdańsk, Poland, e-mails:*{malwit, slowhand}@eti.pg.gda.pl*

Abstract. The paper describes a system called AATE, which contains a test editor that helps the user to prepare an examination document. Every test created with AATE is stored in the database together with its parameters and it can be printed when needed. After the examination the filled forms are imported to the system with the help of a scanner equipped with document feeder. Then the system analyzes the forms and compares the detected answers with a template stored in the database. When the document processing is completed, the final results are also added to the database, from which they can be retrieved and printed at any moment.

1. Introduction

At the present time we often have to deal with a problem of examining large groups of people, e.g. students or employees. Usually the examination has a written or oral form. The oral form is considered to be the better one although it is slower (takes more time) and not always objective. On the other hand we have several different types of written examinations. The simplest approach is to prepare a test, i.e. the printed form that is filled by the examined person. The preparation of the test document usually takes a lot of time but it allows us to perform the examination quickly and to judge all the test participants objectively. The answers read from the form are the basis of good or bad score. Continuously growing number of persons that must be examined requires the automation of the testing process, namely the computerization of acquisition, analysis and processing of the test forms [3, 4].

The AATE system [1] presented in this paper meets the requirements mentioned above. It is still in the development stage but the final version is supposed to be able to process several types of tests. However, the authors believe that the type of test described in the following sections is the most general one and that it would be most commonly used in practice. It represents a multiple-choice test, in which the number of the answer boxes can change from 2 to 5. The number of correct answers may also vary and the lack of answer or the wrong answer can be treated in several ways.

The general diagram of the system is shown on fig. 1.

In the following sections we shall describe the structure of the implemented test type and some of the algorithms contained in the AATE system.

2. The structure of the form

The test form can be created with the help of the built-in editor of the AATE system. The examples of the test editor windows and printed forms are shown on figs. 2 and 3.

From the analysis stage point of view the test form consists of several specific blocks. The first page, called the “main page”, can be divided into the following sections:

1) correction lines – the 3 lines that help to eliminate the skew;
2) the test identifier (bar code) – it contains the test number and the number of the current page;
3) the header field – includes the title of the test (name of the subject) and the space for the name of the participant;
4) the section of questions – every question consists of the two blocks, the question text and the field where the answer can be selected;
5) comment field (the footer) – optional.

If the test contains many pages than the subsequent pages must include the 1st, 2nd and 4th section.

In the case of the multiple-choice test described here we can allocate 2 ÷ 5 answer boxes for each question. It is also possible to individually set the number of correct answers and the way that the correct answer, wrong answer or the lack of answer is scored for every question. During the editing stage the system automatically records the correct answer template for each question in the database.

The dimensions of the 1st, 2nd and 4th section are precisely specified. The dimensions of the other sections can be adjusted to the examiner needs. On all the pages except the main one the sections 3 and 5 are optional.

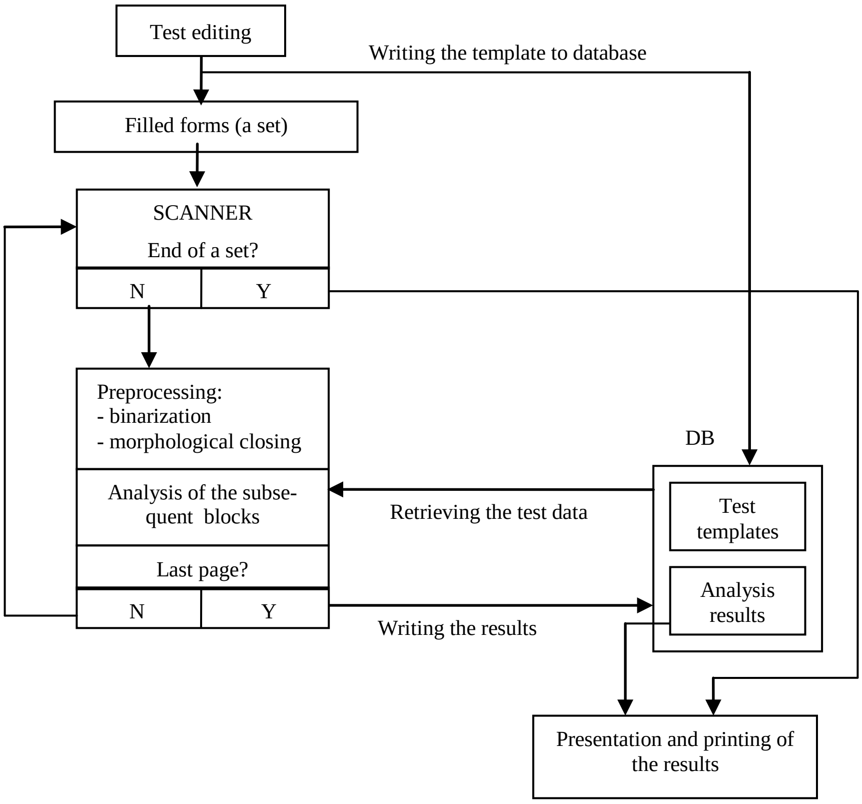


Fig. 1. The functional diagram of the AATE system

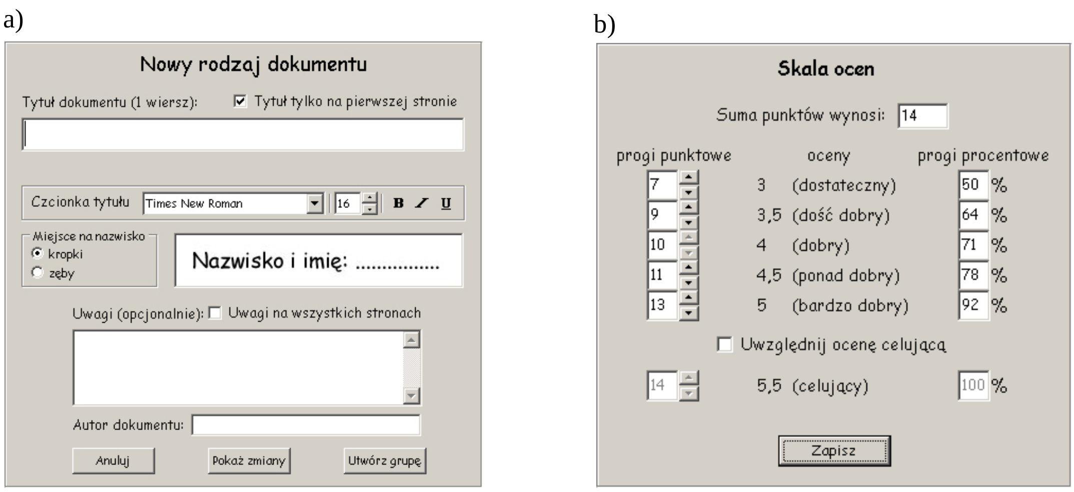
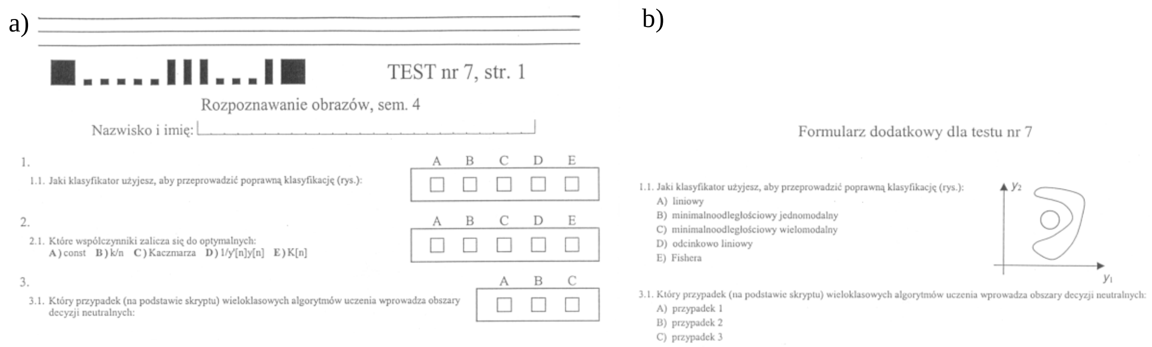


Fig. 2. Examples of AATE windows: a)one of the editor windows, b) window showing the grade thresholds


a)

TEST nr 7, str. 1

Rozpoznawanie obrazów, sem. 4

Nazwisko i imię:

1.

1.1. Jaki klasyfikator użyjesz, aby przeprowadzić poprawną klasyfikację (rys.): A B C D E

2.

2.1. Które współczynniki zalicza się do optymalnych: A B C D E
A) const B) k/n C) Kaczmarza D) 1/y'[n]y[n] E) K[n]

3.

3.1. Który przypadek (na podstawie skryptu) wieloklasowych algorytmów uczenia wprowadza obszary decyzji neutralnych: A B C

b)

Formularz dodatkowy dla testu nr 7

1.1. Jaki klasyfikator użyjesz, aby przeprowadzić poprawną klasyfikację (rys.):
A) liniowy
B) minimalnoodległościowy jednomodalny
C) minimalnoodległościowy wielomodalny
D) odcinkowo liniowy
E) Fishera

3.1. Który przypadek (na podstawie skryptu) wieloklasowych algorytmów uczenia wprowadza obszary decyzji neutralnych:
A) przypadek 1
B) przypadek 2
C) przypadek 3

Fig. 3. A part of a test form with the additional page: a) main page, b) additional page

3. Description of the selected algorithms

Once the scanning of the test form is completed, the system runs the preprocessing methods, i.e. the binarization of the image followed by the morphological closing that improves the continuity of object boundaries.

3.1. Skew estimation and correction algorithm

One of the most common difficulties that we can encounter during the document processing is the incorrect positioning of the pages in the scanning device. As a result incorrect scans are produced and the tilted image of the document is forwarded for processing and recognition.

Manual positioning of the page as well as using the document feeder can cause the skew of about 1-3°. When the skew reaches 2-3° the recognition gets much more difficult and for 5° the results are uncertain and completely useless. The significant skew hinders the recognition of markers and other elements. Moreover, it can make some information disappear from the regions close to the boundary of the scanned document.

There are many known algorithms for skew correction. In our case of the document with the fixed structure, the following simple algorithm of skew angle detection was implemented.

Making use of the fact that there are three horizontal lines at the top of each page we draw one or more (if we want to average the measurements) supporting vertical lines at unequal distances and then we look for the points where they cross the skew correction lines (fig. 4). For each of the crossing points we store its coordinates and calculate the lengths of the *A* and *B* segments (fig. 4). Then the skew angle can be calculated with the help of the following formula

$$\alpha = \tan^{-1}\left(\frac{B}{A}\right) \tag{1}$$

where: $B = (B_1 + B_2 + B_3) / 3$, $B_i = y_{i+3} - y_i$, $i = 1, 2, 3$;
A – the distance between the supporting lines (fig. 4).

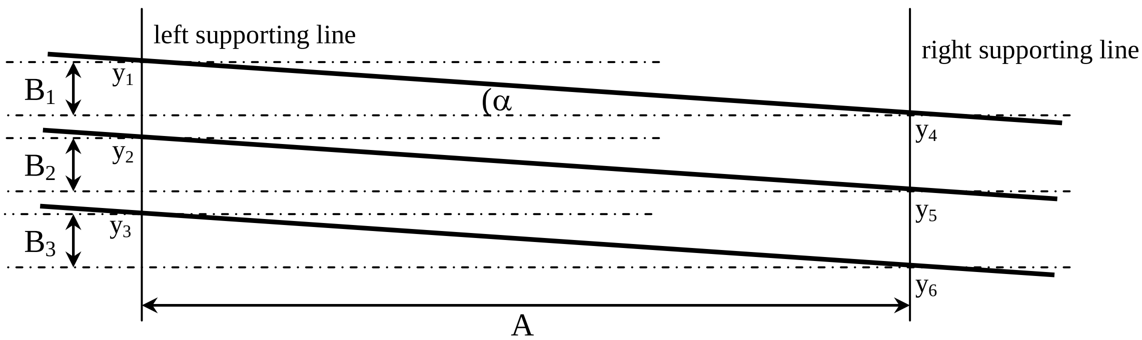


Fig. 4. Graphical illustration of the skew angle analysis method; A – a constant value for a given measurement

Having calculated the α angle we can correct the skew. We chose the hybrid method [2, 5] that looks for a corresponding pixel of the original image for every pixel of the output image.

3.2. Identifier localization

The purpose of the identifier is to encode the test number (that determines its structure) and the number of the test page. An example test identifier and the description of its configuration are shown on fig. 5.

The bar code analysis is based on a histogram, i.e. the projection of pixels onto the horizontal line. Properly selected values of the two local thresholds (p_1 to separate the information from noise, p_2 to distinguish between 0 and 1 bits) allow us to ignore the small distortions and help to properly determine the subsequent values of the analyzed bits. The beginning and the end of the identifier is indicated by the delimiting mark that has the width of 3*d* where *d* is the assumed bar thickness. Other bars and spaces have the width of *d* and the bit with the value of ‘1’ is black while the bit representing ‘0’ is blackened only to the $^1/_4$ of the height of ‘1’. The test number is encoded on 8 bits (it allows to create 256 different tests) and the page number is encoded on 3 bits so the maximum number of pages in one test is 8 (the page numbering starts with 1, not with 0 – see fig. 5).

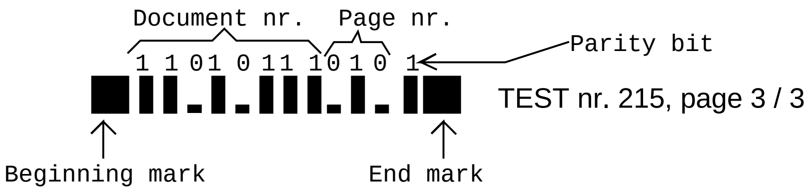


Fig. 5. The structure of the test identifier

The set of forms retrieved from the test participant must begin with the title page; the other pages can be supplied in any order. The additional pages are not used.

3.2. Name extraction

The name of the test participant can be entered to the form in two ways (fig. 6):

1) over the dotted line,
2) in the “teeth”, i.e. every letter in the separate field.

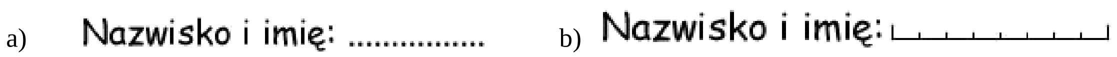


Fig. 6. A field for the participant name: a) over the dotted line, b) in the “teeth”

The AATE system removes the baseline and extracts the whole signature, which is later on included in the list of final test results.

The name extraction algorithm is as follows.

1) Construct the histogram of the name image along the y-axis. Find the maximum value and the delimiting values in the histogram. These values represent the position of the baseline (over which the signature was written) and the upper and lower boundary of the name field. Once the baseline is found, the algorithm removes it from the image.
2) Construct the histogram along x-axis and determine the boundaries of the signature.
3) On the basis of the delimiting values found in step 1 and 2 determine the rectangle containing the signature.
4) Once the whole test has been properly analyzed, save the signature rectangle as a bitmap and add the name of the image to the database.

The section containing the test title is skipped in the analysis stage. This is because the name of the subject shown on the screen and on the final printout is entered directly to the document database during the test preparation.

Further analysis concerns the section of questions that also contains the answers marked in the appropriate boxes. Because during the analysis we are interested only in answers, the subsequent actions performed by the system may be simplified and limited only to the block containing the answer boxes.

4. Presentation of the results

When the analysis of the whole set of documents collected after an examination is completed we can go on to the presentation of the results. First we choose the test from the database using the date of the examination and the test title. Then we can accept or modify the threshold for each grade (fig. 2b) and next the final results are graphically presented. While the grade thresholds are being modified the charts showing the distribution of results are also updated. When the thresholds are finally accepted we are able to print the list of results in the two versions: students receive the simplified list and the examiner can keep the extended one.

5. Conclusion

The AATE system was tested on more than 50 two-page tests filled by students. All the properly filled forms were analyzed correctly. The processing of the filled forms can be considered as the critical operation because the scanning and the analysis are the most time-consuming procedures performed by the system.

Our ability to reduce the time of the scanning process is very limited. The scanning time depends mainly on the chosen resolution. Our experiments show that the minimal resolution of 200 dpi is sufficient for faultless recognition.

We are able, however, to modify the process of analysis of the scanned document. This is why we tried to prepare effective image processing algorithms. Our goal was to develop the methods that work in a "continuously forward" manner, which means that there are as little loops and returns as possible so that the repeated image transfers and analyses are almost eliminated. The algorithms utilize the knowledge about the structure of the document so their operations are limited to the necessary minimum. The analysis is much faster when the page is properly positioned in the scanning device – otherwise the time-consuming process of the skew correction is required.

References

[1]. Ł. Karpowicz, "Komputerowy system analizy arkuszy testów egzaminacyjnych", *Praca dyplomowa, op. W. Malina. Politechnika Gdańska, Wydział ETI* (2002).

[2]. H. K. Kwag et al, "Efficient skew estimation and correction algorithm for document images", *Image and Vision Computing*, **20** (2002), 25-35.

[3]. W. Malina, W. Pawlak, "Modelowanie systemu syntezy i analizy dokumentów", *IX Ogólnopolskie Konwers. "Sztuczna Inteligencja i Systemy Hybrydowe", Warszawa-Siedlce* (1996), 325-333.

[4]. A. Wojtysiak, "Projekt systemu automatycznej syntezy i analizy ankiet", *Praca dyplomowa, op. W. Malina. Politechnika Gdańska, Wydział ETI* (1995).

[5]. P.Y. Yin, "Skew detection and block classification of printed documents", *Image and Vision Computing* **19** (2001), 567-579.