OMR form processing system

Tomasz Gąciarz
Faculty of Physics, Mathematics and Computer Science
Cracow University of Technology
Cracow, Poland
tga@pk.edu.pl

***Abstract*— OMR (Optical Mark Reader) is a technology used for recognizing the presence or absence of optical marks (checkmarks) [1]. Typical applications include the processing of educational tests and reporting, questionnaires, ballots, confidential data and ordering sheets. The documents to be processed are form-like and filled in manually by respondents. This paper presents the design of a system performing fast and accurate processing of OMR forms. The user has ability to work with developed software in batch mode or with so called "correction client". Simplicity of use and high accuracy of the system were main goals of the work.**

Keywords: Form processing; Automatic data acquisition; Image processing

I. INTRODUCTION

The fast and easy in use acquisition of the specific type of information in data entry systems is still barely to be obtained. The traditional way of entering the data is with the keyboard. However, this is not always the best nor the most efficient way. It has been proved that effectiveness of traditional, paper based opinion poll systems is higher in some cases where anonymity is required. In such cases automatic, document processing and identification system may be an alternative. Optical mark reader belongs to the this branch of techniques. It is the fastest and most accurate of the data collection technologies. Efficient OMR system should be simple in use (from hardware and software point of view) and be easily applicable in practice. Although it is well covered area with institutions using such systems for over 40 years but it lacks of the system which can be prepared (form designing) and applied (by using phone camera or webcam) by the user in a few minutes even without using the computer.

It is not easy to fulfill both of this requirements. In this article we describe briefly such a system which was particularly designed to assist the processing of educational tests and questionnaires.

II. OMR SYSTEM OVERALL

The using of the system can be generally divided into two phases. In the first of them appropriate OMR form must be defined. Then we can use such forms in the process of data acquisition. After filling it by respondent they must be entered to the system by means of some scanning device. Our solution is versatile so the user can use a wide range of scanning tools from scanners or high resolution camera to simple and popular phone cameras.

Thanks to the sophisticated image preprocessing module it is possible to use simple and cheap input devices delivering monochrome, grayscale or color images.

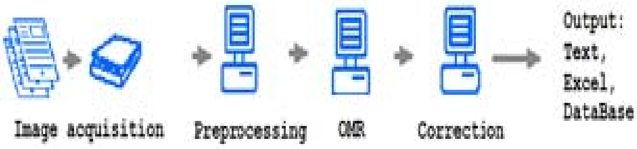


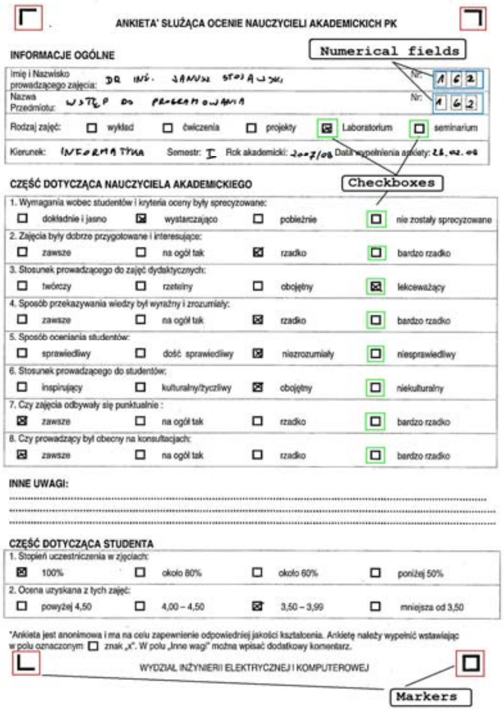
Figure 1. OMR System workflow

III. PREPARING OF OMR FORMS

A. Using word-processing editor

OMR form can be prepared in two ways. First (standard one) solution is to use design packages with pre-defined templates or any word-processing editor. Beside the description content they include rectangular checkboxes where we will detect the absence or presence of check marks. In our system there is facultative possibility to use numerical fields which will be recognized with OCR module designed for this purposes. All the fields containing data to be processed should be well separated from the rest to minimize the risk of misclassification or accidental filling. Usually some free space around checkboxes and other OCR fields is required, see figure 2.

The template holds also the information (markers) that allows the software to localize the areas of data on the form even in the case of rotation and deskewing. Markers are usually placed in corners of the form and one of them is different than the others in order to determinate the correct page orientation of the form.



ANKIETA' SŁUŻĄCA OCENIE NAUCZYCIELI AKADEMICKICH PK

Numerical fields

INFORMACJE OGÓLNE

Imię i Nazwisko prowadzącego zajęcia: DR INŻ. JANUSZ STOJAŁSKI Nr: 1 6 2

Nazwa Przedmiotu: WSTĘP DO PROGRAMOWANIA Nr: 1 6 2

Rodzaj zajęć: □ wykład □ ćwiczenia □ projekty ☒ Laboratorium □ seminarium

Kierunek: INFORMATYKA Semestr: I Rok akademicki: 2007/08 Data wypełnienia ankiety: 26.02.08

Checkboxes

CZĘŚĆ DOTYCZĄCA NAUCZYCIELA AKADEMICKIEGO

1. Wymagania wobec studentów i kryteria oceny były sprecyzowane:
□ dokładnie i jasno ☒ wystarczająco □ pobieżnie □ nie zostały sprecyzowane

2. Zajęcia były dobrze przygotowane i interesujące:
□ zawsze □ na ogół tak ☒ rzadko □ bardzo rzadko

3. Stosunek prowadzącego do zajęć dydaktycznych:
□ twórczy □ rzetelny □ obojętny ☒ lekceważący

4. Sposób przekazywania wiedzy był wyraźny i zrozumiały:
□ zawsze □ na ogół tak ☒ rzadko □ bardzo rzadko

5. Sposób oceniania studentów:
□ sprawiedliwy □ dość sprawiedliwy ☒ niezrozumiały □ niesprawiedliwy

6. Stosunek prowadzącego do studentów:
□ inspirujący □ kulturalny/życzliwy ☒ obojętny □ niekulturalny

7. Czy zajęcia odbywały się punktualnie :
☒ zawsze □ na ogół tak □ rzadko □ bardzo rzadko

8. Czy prowadzący był obecny na konsultacjach:
☒ zawsze □ na ogół tak □ rzadko □ bardzo rzadko

INNE UWAGI:

....

....

....

CZĘŚĆ DOTYCZĄCA STUDENTA

1. Stopień uczestniczenia w zjęciach:
☒ 100% □ około 80% □ około 60% □ poniżej 50%

2. Ocena uzyskana z tych zajęć:
□ powyżej 4,50 □ 4,00 – 4,50 ☒ 3,50 – 3,99 □ mniejsza od 3,50

*Ankieta jest anonimowa i ma na celu zapewnienie odpowiedniej jakości kształcenia. Ankietę należy wypełnić wstawiając w polu oznaczonym □ znak „x". W polu „Inne wagi" można wpisać dodatkowy komentarz.

WYDZIAŁ INŻYNIERII ELEKTRYCZNEJ I KOMPUTEROWEJ

Markers

Figure 2. Example of OMR form used in the process of evaluation of quality of teaching

B. *Fast Manual preparation of OMR form*

User can design OMR form from scratch even without using the computer and text editors. In this case he can design the form just using color pen and the paper. Areas marked with predefined color will define recognition fields connected with markers, checkboxes and numerical fields. System will automatically set up all necessary parameters needed to find this fields in scanned documents and additionally will automatically prepare correction client. All we need is to preserve the relation between dedicated color and field type. Figure 3. shows the required steps.

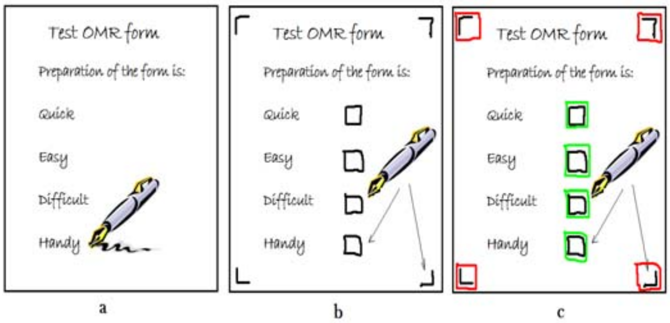


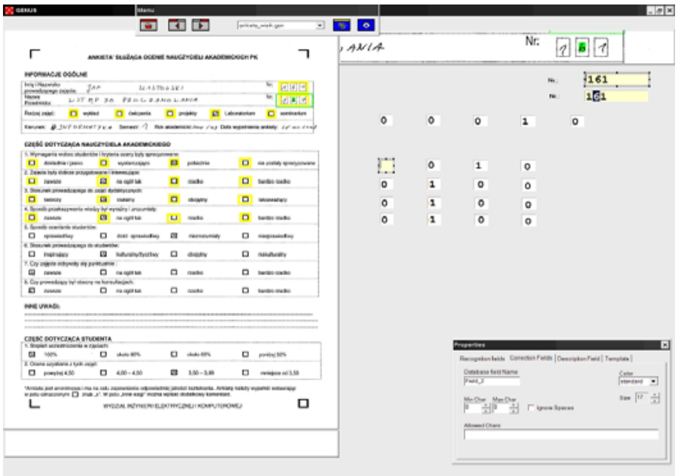
Figure 3. Manual preparation of the OMR form: a) – writing the text content using pen or text editor, b) – placing markers and checkboxes, c) – coloring recognition areas

IV. SETTING UP THE SYSTEM

Having OMR form prepared we can easily set up the system to process it. The whole workflow (excluding scanning) is done by one program. The software is simple and intuitive, built as one dialog box window. It works in two modes: so called "the design mode" and "the working mode". Basically the same interface is used in both of them. In the design mode user defines positions of recognition fields and markers after loading the image of scanned form and layout of control widgets displaying images and correction fields linked with appropriate recognition fields. All this actions are made easily by resizing existing controls or by drawing rectangles indicating recognition fields and markers (figure 4).

However this step can be omitted if we choose "manual" designing of the form as shown in figure 4. In this case the system will automatically detect all recognition fields based on their color and propose correction fields corresponding to them.

The same window is then used in working mode when processing the filled and scanned forms.



Nr: 1 6 1

Nr: 161

Nr: 161

0 0 0 1 0

0 1 0

0 1 0 0

0 1 0 0

0 1 0 0

Properties

Recognition fields | Correction Fields | Description Field | Template

Database field Name: Field_2

Color: standard

Size: 17

Min Char: 0 Max Char: 0 □ Ignore Spaces

Allowed Chars:

Figure 4. Defining positions of recognition fields and markers in design mode

V. IMAGE ACQUIRING AND PREPROCESSING

A. *Adaptive binarization*

After the forms have been filled out, we get the data into the computer. Our solution is versatile so the user can use a wide range of scanning tools from scanners or high resolution camera to simple and popular phone cameras. In the case when the color or gray scale image is acquired there is a need to binarize it. According to Brauel [3] we have developed fast, adaptive binarization method. It combines the statistical constraints of Sauvola's technique [2, 5, 6] with integral images [4].

Proposed method can conceptually be divided into the two following steps:

1) labeling of areas containing foreground at low-resolution

2) performing threshold at the labeled areas containing foreground at high-resolution

The initial segmentation between areas containing foreground and background is based on common assumption that the variance of grey levels in small neighborhood of pixels belonging to the text areas is larger than in background areas.

At this stage only a raw estimation and labeling of areas containing foreground is required. Using suitable threshold level of variance we can distinguish between areas containing foreground and background. At the next stage, the original image is binarized. For every pixel, laying in labeled foreground areas, suitable threshold level - according to Sauvola's technique - is applied and final decision is made.

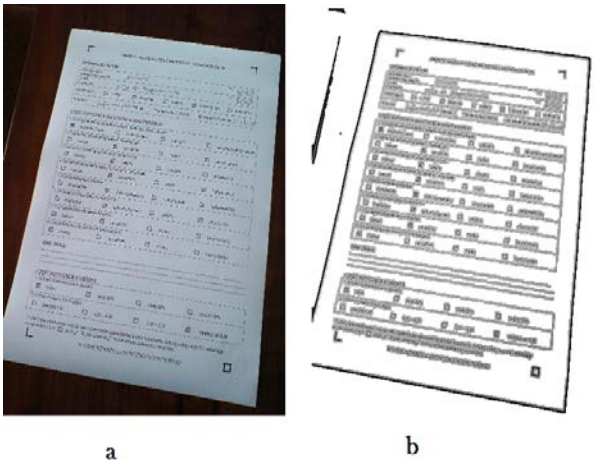


Figure 5. Example of adaptive binarization method (step 1)

The local *mean* and *variance* is computed using integral images. An integral image *i*, denoted as i(x,y), at location (x,y) contains the sum of pixel intensities above and to the left of (x,y) (see figure 6a) and can be written as:

$$i(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} g(i,j) \quad (1)$$

It can be efficiently computed in one single pass over the entire grayscale image - *g* using the following recurrence relation:

$$i(x,y) = g(x,y) + i(x-1,y) + i(x,y-1) - i(x-1,y-1) \quad (2)$$

where *i*(*x*, -1)=*i*(-1,*y*)=0;

Given the integral image, the sum of pixels intensities within a rectangular region of the image aligned with the coordinate can be easily computed in constant time with only four array references (see figure 3b).

$$\sum_{A(x)<x\le C(x),A(y)<y\le C(y)} g(x,y) = i(C) + i(A) - i(B) - i(D)$$

The local mean for any window size *w* can be computed as:

$$m_{ABCD} = (i(C) + i(A) - i(B) - i(D))/n \quad (3)$$

Similarly the local variance can be computed by using an integral image of the squared pixel intensities:

$$ii(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} g^2(i,j) \quad (4)$$

$$v_{ABCD} = \left(1/n \sum_{A(x)<x\le C(x),A(y)<y\le C(y)} g^2(x,y)\right) - m_{ABCD}^2 \quad (5)$$

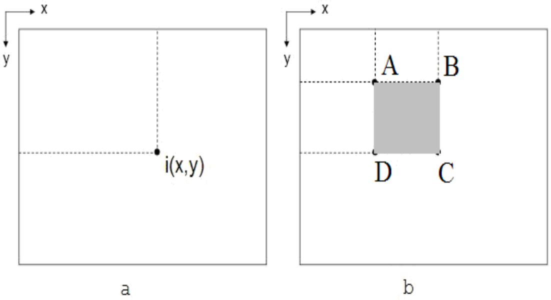


Figure 6. Integral image calculation

For grayscale images in which pixels can have intensity values *g*(*x*,*y*) from 0 to 255 the output value for each pixel in binary image is computed as:

$$o(x,y) = \begin{cases} 0 & \text{for } g(x,y) \le t(x,y) \\ 255 & \text{otherwise} \end{cases} \quad (6)$$

Where the threshold *t*(*x*,*y*) according to Sauvola's technique is computed using the mean value of pixel intensities and variation *v*(*x*,*y*) in a *w* × *w* window around the pixel (*x*,*y*):

$$t(x,y) = m(x,y)\left(1 - k\left(1 - \frac{\sqrt{v(x,y)}}{R}\right)\right) \quad (7)$$

where parameter *R* refers to dynamic range of the standard deviation (*R* = 128 for grayscale images) and *k* takes positive value between 0 and 1 (in described in this paper method *k*=0.15 and *w*=13).

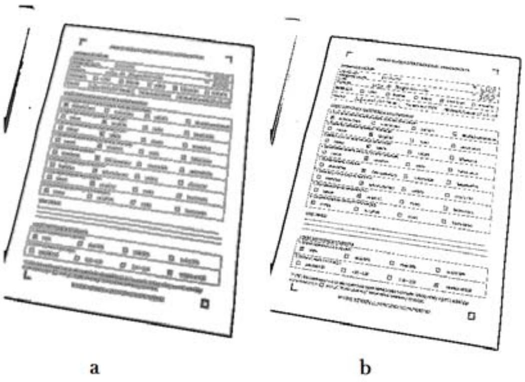


Figure 7. Example of adaptive binarization method (step 2)

B. Deskewing

During this step of preprocessing we remove the skew from images based on previously defined markers. This step is not essential in order to find and evaluate the data fields but can help during the correction phase when the user inspect the image visually.

The distortion parameters are computed based on located markers points on scanned documents (hypothesis points) and their corresponding points (reference points) from registered pattern form. They are estimated using LSQ method and account for rotation, translation, and scale [7]. Linear equations (with 3 unknowns) are defined for x and y directions as follow:

$$\begin{aligned} x_h &= \Delta x + m_{x_x} x_r + m_{x_y} y_r \\ y_h &= \Delta y + m_{y_y} y_r + m_{y_x} x_r \end{aligned} \qquad (8), (9)$$

The hypothesized x_h-coordinate is linearly dependent on the reference x_r and y_r. The same is true for the y_h in the second equation.

After applying the LSQ on first equation it expands into the following system of three linear equations:

$$\begin{aligned} \sum_{i=1}^{n} x_h &= n\Delta x + m_{x_x} \sum_{i=1}^{n} x_r + m_{x_y} \sum_{i=1}^{n} y_r \\ \sum_{i=1}^{n} x_h x_r &= \Delta x \sum_{i=1}^{n} x_r + m_{x_x} \sum_{i=1}^{n} x_r^2 + m_{x_y} \sum_{i=1}^{n} x_r y_r \\ \sum_{i=1}^{n} x_h y_r &= \Delta x \sum_{i=1}^{n} y_r + m_{x_x} \sum_{i=1}^{n} x_r y_r + m_{x_y} \sum_{i=1}^{n} y_r^2 \end{aligned} \qquad (10)$$

Which allow us to estimate parameters $\Delta x, m_{x_x}, m_{x_y}$.

It is also applicable to the second equation.

Described method computes a linear mapping that minimizes the total discrepancy between all reference and hypothesized points. The error is decreased as the number of points used in Leas Squares calculation increases which improve registration quality.

Having estimated parameters $\Delta x, m_{x_x}, m_{x_y}, \Delta y, m_{y_x}, m_{y_y}$ they are substituted back into equations (8) and (9) and the pixels in the input form are transformed by computing $(x_{h,} y_h)$.

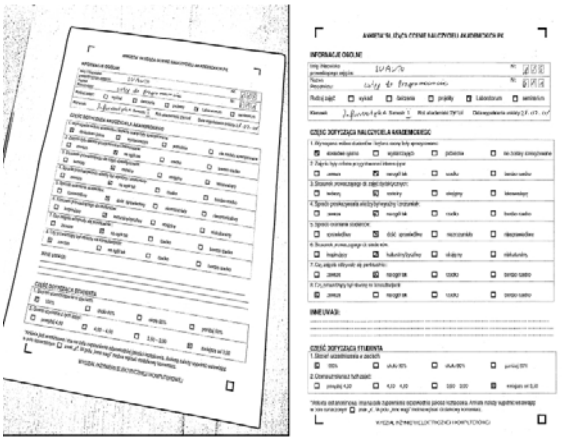


Figure 8. Example of deskewing method

VI. OMR PROCESS AND CORRECTION CLIENT

OMR software can process images in batch mode or in single image processing mode with correction. It can also process images continuously and stop for correction only when the error occurs or there is suspicious result. Navigation and all undertaken actions during correction should be possible using only the keyboard. The output data are stored in plain text file containing fields numbers and recognition results.

VII. CONCLUSIONS

Reading checkboxes on a paper form may seem like the easiest of all types of recognition, but accurate, versatile and simple in use OMR has many nuances both in image preprocessing and recognition. Presented solution was tested in practice and can achieve accuracy rate up to 99.99%. This result place it among the fastest and most accurate data collection technologies, especially in situations where the person entering the data does not have (or should not have) access to the sophisticated data recording equipment (e.g., students, poll-takers).

REFERENCES

[1] H. S. Cummings, M. McCubbrey, D. Pinsonnault, A. Donovan "Management Information Systems for the Information Age" (3rd ed.). Canada: McGraw-Hill Ryerson, 2006.

[2] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," Pattern Recognition 33(2), pp. 225–236, 2000.

[3] F. Shafait, D. Keysers, T. M. Breuel, "Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images", Document Recognition and Retrieval XV, San Jose, CA, 26-31 January 2008

[4] P. Viola and M. J. Jones, "Robust real-time face detection", Int. Journal of Computer Vision 57(2), pp. 137–154, 2004.

[5] N. Otsu, "A threshold selection method from gray-level histograms" in: *IEEE Trans. Systems, Man, and Cybernetics* 9(1), pp. 62–66, 1979.

[6] W. Niblack, "An Introduction to Image Processing", Prentice-Hall, Englewood Cliffs, NJ, 1986.

[7] M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, and C. L. Wilson, "NIST Form-Based Handprint Recognition System", NISTIR 5469, 1994