Webcam Based Real-Time Robust Optical Mark Recognition

Huseyin Atasoy(✉), Esen Yildirim, Yakup Kutlu, and Kadir Tohma

Department of Computer Engineering, Iskenderun Technical University, Hatay, Turkey
{hatasoy,eyildirim,ykutlu,ktohma}@mku.edu.tr

Abstract. This study proposes a robust, low cost, real-time optical mark recognition (OMR) system that uses a webcam and a small OMR form to read hand-marked data from plain paper. The system is designed to read data from any user-designed OMR form which can be customized for any purpose. It was implemented and tested on examination papers to read students' numbers and their examination results automatically. Results and numbers on 87 out of 88 papers were correctly identified. It was tested under different lighting conditions and with different mark colors. The results indicate that the system is robust and reliable.

Keywords: Optical mark recognition · Clustering · Mark detection

1 Introduction

Optical mark recognition is a generic name for techniques of data extraction from hand-filled forms. This technology provides a solution for reading and processing large number of forms such as questionnaires or multiple-choice tests. It is widely used, especially for grading students in schools.

There are many solutions that use devices specialized for holding large number of forms and reading them as they are being fed into the devices. But these solutions require scanner devices and a special kind of paper which is more expensive than a regular sheet and has a fixed layout. Therefore, cheaper solutions are needed for simpler purposes that do not need a whole OMR sheet. For instance to read student number from a regular sheet that can contain anything beside the OMR form.

Several studies have been proposed for optical mark recognition so far. Perez-Benedito et al. have tested and compared two softwares to decrease the time consumed for the process of student assessment, as a part of their educational innovation project [1]. Nguyen et al. have proposed a camera based multiple choice test grading method [2]. In another work, Xiu et al. have proposed a style-based ballot recognition approach and have compared their approach with the others [3]. Deng et al. have presented an OMR solution that uses scanners and supports plain sheets and low printing quality [4]. Kubo et al. have developed a web based integrated OMR system that needs scanner devices [5].

S. Arik et al. (Eds.): ICONIP 2015, Part II, LNCS 9490, pp. 449–456, 2015.
DOI: 10.1007/978-3-319-26535-3_51

In this paper we propose a webcam-based, real-time OMR system that can reads data from user-designed OMR forms. The system is focused on reading user-designed small forms rather than special OMR sheets and it can perform the task using a simple webcam, under bad lighting conditions, different mark colors and intensities.

2 Methods

The proposed system consists of two steps, namely, adaptive binarization and mark detection. In the first step, some well-known basic image processing techniques such as histogram equalization, erosion, dilation are applied to the image to obtain the binary form of the image without being affected by lighting conditions or image quality. In the second step, binary image is subjected to some processes to detect traces and to determine corners of the OMR form and locations of hand-drawn marks. The proposed system is shown in Fig. 1.

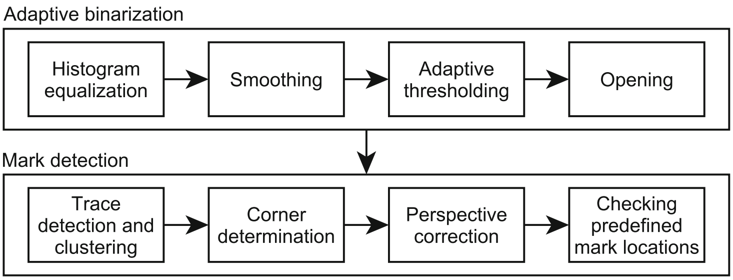

Fig. 1. Proposed mark recognition system

2.1 Adaptive Binarization

Binarization is a simple process of converting images from grayscale to binary format by using a pixel value threshold. A constant threshold is not applicable to images captured under different lighting conditions. For this reason, adaptive threshold determination algorithms such as Otsu's method [6] are preferred. But in optical mark recognition, marks have not always the same color or intensity and the system is desired to be invariant to lighting conditions, mark color and intensity. Therefore Otsu's method was used along with histogram equalization, smoothing and opening in the order shown in Fig. 1. Noise is reduced on these steps gradually.

Histogram Equalization: Histogram equalization enhances contrast in an image by stretching intensity range. The main idea of the histogram equalization is to build a lookup table (Eq. 2) for mapping intensity values to new values that make distribution of them wider and more uniform. Calculation of cumulative histogram, constructing a lookup table and mapping intensities are given in Eqs. 1, 2 and 3, respectively, where H is the histogram, H' is the cumulative histogram, L is the lookup table, M is the maximum intensity and N is the number of pixels in the image.

$$H'(i) = \sum_{j=0}^{i} H(j) \tag{1}$$

$$L(i) = \frac{M \times H'(i)}{N} \tag{2}$$

$$dst(x, y) = L(src(x, y)). \tag{3}$$

Smoothing: Beside increasing the image contrast, histogram equalization may increase noise contrast too. So it is important to reduce the noise after this process. Smoothing is one of the methods that reduce noise in an image. Since the system is wanted to run in real-time, the simplest and the fastest smoothing approach that is based on changing each pixel value to sum of pixel values in its NxN neighbourhood, was used [7]. Default value of N is set to 4 and user is allowed to change it from the user interface. Pixels' final values are not scaled after the summation to whiten the image.

Adaptive Thresholding: Grayscale images can simply be converted to binary images by using constant threshold values. But constant values are not useful for images that are captured under different environmental conditions. Otsu's method [6] is used to determine an adaptive threshold value in this study.

Otsu's method searches for the threshold that minimizes the within-class variance where classes are assumed to be background and foreground. Since the between-class variance is faster to calculate and it takes its maximum value when the within-class variance is minimum, the threshold that makes the between-class variance maximum is considered as the best threshold. The between-class variance is defined as

$$W_b W_f \left(\mu_b - \mu_f\right)^2 \tag{4}$$

where W_b and W_f are weights of background and foreground pixels (ratios of background and foreground pixels to total number of pixels) and μ represents the means.

Opening: If two or more marks are too close in the image, they may seem like a single mark. To reduce the possibility of such confusion, opening which is an erosion (shrinking) followed by a dilation (expanding) [8] is applied to the image. This also helps to eliminate small traces and this is the final step to remove the noise.

Results that were obtained after each step of the first step are shown in Fig. 2.

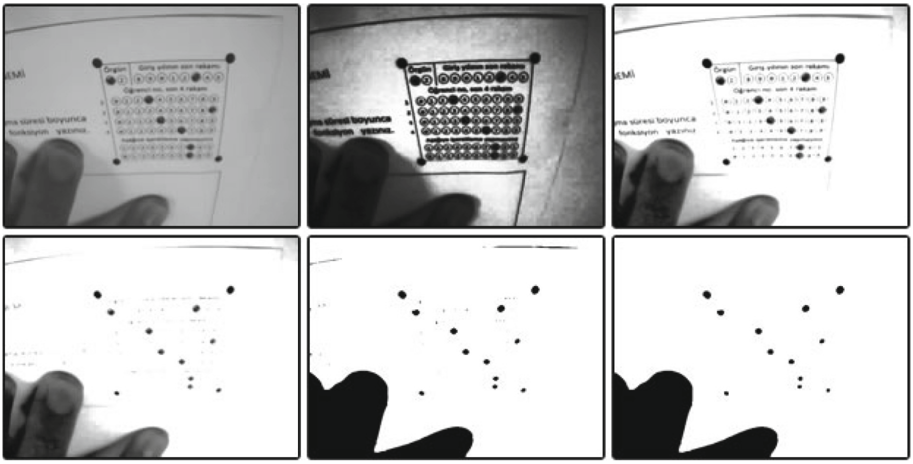

Fig. 2. Results of each step in step 1, from left to right, from top to bottom: original gray image, histogram equalization, smoothing, erosion, adaptive thresholding and opening.

2.2 Mark Detection

Trace Detection and Clustering: After obtaining the binary image, gravitational center of each element in the binary image is determined by calculating means of coordinates of pixels that belongs to the same element. These centers are considered as centers of mark candidates. Then Euclidean distances between the center points and a predefined clustering point are calculated. $M + 4$ points that has the shortest distance values are considered as marks where M is the number of marks to be drawn by hand and 4 is the number of points that will be used to determine the corners.

Blue circles with fixed radii are drawn on the points in the cluster to be used on the next steps. The other points that are not included in the cluster are marked with circles filled with red color.

Corner Determination: Corners of the form are assumed to be the nearest points to the corners of the image. Therefore, 4 points that has the shortest distances to the corner points of the image are selected and considered as the corners of the OMR form.

Perspective Correction: Since the corners may not form a rectangle depending on the view angle, a perspective transform [7] is applied to the corners of the form to correct the perspective. The area that contains the marks is scaled to a fixed size with a correct perspective in this manner.

Checking Predefined Mark Locations: Predefined locations where the user can draw marks are checked if they include blue pixels. Because if these areas are marked by user, blue dots are drawn on them on the "Trace Detection and Clustering" step. Locations that include blue pixels are considered as marked fields.

Figure 3 shows the image after applying the mark detection step. On the left the red point (in this example there is only 1 red point) are the traces which are not included in the cluster. The blue points are the points that are considered as marks and the blue points rounded by red circles represent the corners of the OMR form. The area rounded by a red rectangle is the view of the form after perspective correction (on the right bottom corner of Fig. 3a). On Fig. 3b, the red lines are the distances between gravitational centers of the elements in the binary image and the clustering point. In this example, M = 8 and the nearest 8 + 4 points to the clustering point are selected. The blue lines represent the distances between the corners of the image and the closest points in the cluster to the corners. The red circle was drawn to show approximate location of the clustering center.

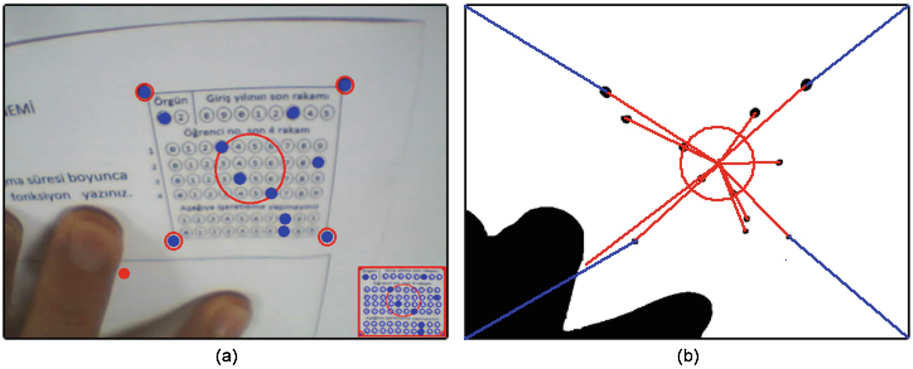

Fig. 3. (a) Image after the mark detection step. (b) The cluster and the corners on the same image (Color figure online).

3 Implementation

The proposed system was implemented as a customizable software in Visual Basic.NET and implementations of OpenCV [9] were used for basic image processing methods. The software is customized to read students' numbers and their examination results from a webcam in real-time and writes them into a file. It can be adapted for different purposes by changing its configurations using the provided interface (Fig. 4).

3.1 User-Designed OMR Forms

The implementation allows users to design their own OMR forms. But some requirements need to be satisfied while designing the form:

- Corners of the forms must be pre-marked to help the system to determine the corners.
- A clustering point must be predefined for the proposed mark determination method.
- The number of marks to be drawn by users must be fixed, optional fields are not allowed.

A screen-shot of the program and the form that was designed to test the implementation are given in Fig. 4.

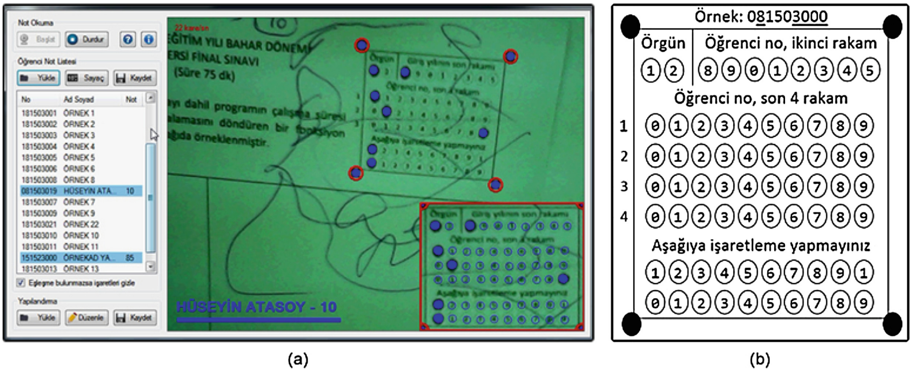

Fig. 4. (a) A screenshot of the program. (b) The OMR form that was used to test the implementation.

4 Experimental Results

The application was tested with 88 examination papers to read students' numbers and their examination results. A small OMR form which was designed specifically for this purpose was placed on top right of the first page of each examination paper and students were asked to fill out the forms with required information to construct the student number. Exam result was marked in the form by filling out two bubbles on the bottom of the form.

87 of the 88 forms were correctly read by the application with an average processing time of 23.8 ms which means that the theoretical processing speed is 42 fps (frames per second). But the webcam that was used in this experiment was not able to provide images more than 25 in a second. So the speed is limited to 20 fps and thus, possible excessive processor usage is also prevented. The form that could not be read was filled incorrectly (Fig. 5a).

A form was scratched and shown to two webcams that provide different image qualities under two different lighting and background to test robustness of the system under different conditions. They were successfully read. The results are shown in Fig. 5.

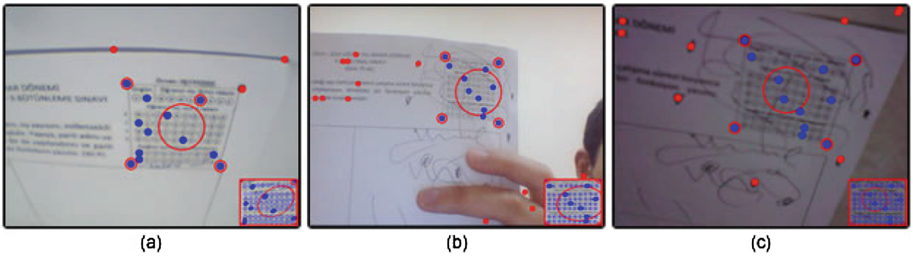

Fig. 5. (a) The form that was filled incorrectly and could not be read properly. Because two marks were drawn in the same row. (b) A scratched form that was read successfully over an unconstrained background. (c) The same scratched form that was read successfully again, under low lighting.

5 Conclusion

In this paper a webcam-based, fast and robust OMR system was presented. The system was implemented and tested on real examination papers to read student number and grade and to save them as an Excel file. All the forms, except one that was filled incorrectly, were read successfully in a very short average processing time.

The experiments show that the system is fast and robust enough to be used in real-time applications even with low-quality webcams. It can be adapted for different purposes using different OMR forms that are designed to be filled by hand with more specific data.

But it can not read forms that have optional fields because of the clustering method which was proposed to determine marks. In future works, different clustering methods can be investigated to make the system able to read OMR forms that has optional fields.

References

1. Perez-Benedito, J.L., Querol Aragon, E., Alriols, J.A., Medic, L.: Optical mark recognition in student continuous assessment. Tecnologias del Aprendizaje, IEEE Rev. Iberoamericana de **9**(4), 133–138 (2014)
2. Nguyen, T.D., Manh, Q.H., Minh, P.B., Thanh, L.N., Hoang, T.M.: Efficient and reliable camera based multiple-choice test grading system. In: 2011 International Conference on Advanced Technologies for Communications (ATC), pp. 268–271. IEEE (2011)

3. Xiu, P., Lopresti, D., Baird, H., Nagy, G., Smith, E.B.: Style-based ballot mark recognition. In: 10th International Conference on Document Analysis and Recognition, ICDAR 2009, pp. 216–220. IEEE (2009)
4. Deng, H., Wang, F., Liang, B.: A low-cost OMR solution for educational applications. In: International Symposium on Parallel and Distributed Processing with Applications ISPA 2008, pp. 967–970. IEEE (2008)
5. Kubo, H., Ohashi, H., Tamamura, M., Kowata, T.: Shared questionnaire system for school community management. In: International Symposium on Applications and the Internet Workshops, pp. 439–445. IEEE (2004)
6. Otsu, N.: A threshold selection method from gray-level histograms. Automatica **11**(285–296), 23–27 (1975)
7. Forsyth, D.A., Ponce, J.: A Modern Approach. Computer Vision: A Modern Approach (2003)
8. Jain, R., Kasturi, R., Schunck, B.G.: Machine Vision. McGraw-Hill, New York (1995)
9. Bradski, G.: The opencv library. Doct. Dobbs J. **25**(11), 120–126 (2000)