Cost Effective Real-Time Image Processing Based Optical Mark Reader

Amit Kumar, Himanshu Singal, Arnav Bhavsar

Abstract—In this modern era of automation, most of the academic exams and competitive exams are Multiple Choice Questions (MCQ). The responses of these MCQ based exams are recorded in the Optical Mark Reader (OMR) sheet. Evaluation of the OMR sheet requires separate specialized machines for scanning and marking. The sheets used by these machines are special and costs more than a normal sheet. Available process is non-economical and dependent on paper thickness, scanning quality, paper orientation, special hardware and customized software. This study tries to tackle the problem of evaluating the OMR sheet without any special hardware and making the whole process economical. We propose an image processing based algorithm which can be used to read and evaluate the scanned OMR sheets with no special hardware required. It will eliminate the use of special OMR sheet. Responses recorded in normal sheet is enough for evaluation. The proposed system takes care of color, brightness, rotation, little imperfections in the OMR sheet images.

Keywords—OMR, image processing, hough circle transform, interpolation, detection, Binary Thresholding.

I. INTRODUCTION

OPTICAL MARK RECOGNITION (OMR) is technique widely used in education sector. Most of the schools, colleges and exams use it to record answers. The answer sheet checking system has become much easier, powerful, and automatic using OMR [1], [2]. Currently the authorities have to use dedicated scanner, special OMR software and sheets to evaluate the responses. The small organizations or institutes, authorities cannot use OMR due to the limitation that a dedicated system is required to evaluate the OMR sheet, which is not an economically viable option. In this paper we are presenting an Image Processing based approach to evaluate the OMR sheet which works pretty well detecting the OMR responses. We want to automate the work of OMR correction to be smart, intelligent and economical. Using the approach we proposed a totally free solution can be developed where the image of the OMR sheet can be fed into the algorithm which will do the evaluation the OMR sheet and provide the scores. It can take care of different orientation of the images, detect responses and will provide the scores in real time.

We take input the image of the OMR, this image can be a normal image of OMR sheet, there is no special need of the dedicated scanner and special OMR paper. We take into account that the image can be little distorted or misaligned. Our algorithm runs on this image and tries to evaluate it. We try to find out all the regions of interest(ROI), i.e. circles in OMR sheet. We pass this input image through Hough circle

Amit Kumar, Himanshu Singal and Arnav Bhavsar is with the IIT Mandi, Mandi, India (e-mail: amityadavkv2@gmail.com, singalh2@gmail.com, arnav@iitmandi.ac.in).

transformation [3], which runs in solution domain and tries to find out all the circles in the image. We consider that the Hough transform may not give accurate results. After Hough transform, there might be a few circles that hough transformation was not be able to detect and there might be some false circles detected by the algorithm. To rectify the errors from circle detection, we use the special fact that the OMR sheet circles are in grid, the circles are vertically and horizontally parallel. We leverage this characteristic of the OMR sheet and do regression by defining a error function and minimizing it. At the end of the regression process we end up finding out vertically parallel lines which fits into most of our detected circle. Once all the circles of the sheet are detected with their centres using hough transform and parallel lines interpolation, we detect the options marked by the student. For that we perform binary thresholding on the image followed by a voting scheme to figure out the student responses and evaluate the sheet comparing it with the correct responses.

II. RELATED WORK

The OMR machines expects a fixed dimension OMR sheet on a constant dots per inch (dpi). So the ROI(Region of interest) is determined and fixed. Most of the current systems and algorithms leverage the special scanning for finding the ROI, they do not try to find out the circles but rely heavily on the scanning and OMR sheet characteristics. There are few attempts by the people to simplify this process and develop image processing based algorithm. There are efforts to do the tilt correction in the scanning but it also leverage proper scanning for finding ROI [4]. For mitigating the minor scanning errors the softwares leverage the existence of pointers on both side of the papers and with the help of those it matches the rows and does the tilt correction. OMR sheet with the special marking is shown in the Fig. 1 The paper [1] talks about a finding ROI and applying template based matching algorithm for the evaluation. This work [5] tries to register the image based on the two squares, they find the angle of the rotation based on the top two square and then rotate the whole image in opposite direction to counterfeit the rotation. This work [6] does the tilt correction using the top left and top right boxes finding the angle of rotation of the scanned image. They do flag points searching which is the marker at the beginning of each answers.

III. ALGORITHM

Most of the work trying to make OMR using image processing uses a predefined grid-based system in which prior

Fig. 1 Traditional OMR sheet showing the black marks(used for tilt correction) on the sheet boundaries

Fig. 3 Input OMR sheet with responses clicked with a normal phone camera

knowledge about the distance between different circles need to be known in advanced. They use dots on the top of the grid to detect any tilt and measure the length of the image. Based upon that, they correct the image for any inclination or tilt and try to fit the template to the image. They already know the position of the circles of non-tilted, particular size image. Thus, they will fail if the position of circles is changed, e.g. the spacing between the rows or columns might increase. If the scanning method changes or the OMR sheet changes the approaches will fail to evaluate the sheet.

We tested a completely different approach in which we focused on detecting all the circles we could in the image and from that find the circles marked in the image.

One of the challenges in detecting circles is that we need to detect them all so that given a circle, we could know which question number and option that circle belongs to. We implemented the code in python and used opencv library. The outline of the algorithm is described below

Fig. 2 Block diagram of the proposed algorithm

A. Input Image

We tested the algorithm on the image of an OMR sheet. Multiple scanned images of the OMR sheet were taken for testing. Fig. 3 shows the image on which the approach was tested. We also tilted the image to different inclinations and tested our algorithm. The tilted images are shown in Fig. 9. Current algorithm works on OMR sheets with circles, it will not work with OMR sheets of other shapes. Though a variant of our algorithm will work fine on it, where the first step will be replaced with the respective transformation for detecting the new shapes.

B. Hough Circle Transform

Hough Circle Transform is a technique used to identify circles in an image [7], [8]. It first detects edges in the image. It is a kind of voting based algorithm where each edge votes for a circle at center x,y and radius r if such a circle passes through the edge point. If a circle receives more than a particular number of votes, it is chosen to be a valid circle. A list of all such circles is returned. We have used opencv implementation of the Hough circle transform. It requires single channel image, so the image is first converted to gray scale. Then we fed that gray scale image to Hough circle transform algorithm. The minimum vote was kept to 11, min distance between two circles to 25, max radius to 20. We assumed that the circles radius will be less than 20. For radius less than 20 pixels, if even 11 pixels fit the circle equation, it will work fine because of oval nature of the options in OMR sheet. As the radius can be up to 20 pixels, we can assume that the center of any other option is at least 20+5 pixels away from a detected option (giving 5 pixels as minimum spacing one may expect).
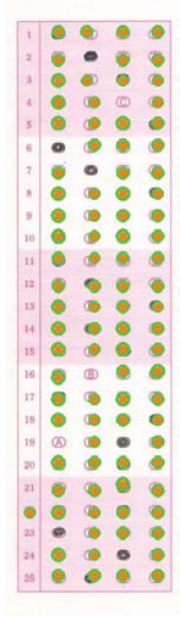
Fig. 4 Input Image overlapped with the circles detected from Hough circle transformation algorithm
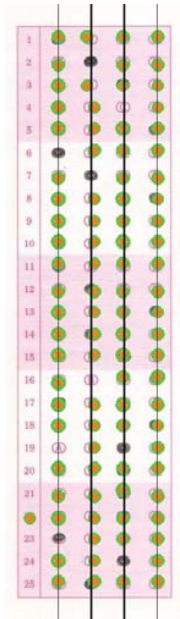

Fig. 5 Input Image overlapped with the interpolated parallel lines derived from error minimization after hough transform

The problem with Hough circle Transform is that it is not able to detect all the circles and also gives false circles as shown in Fig. 4. Some circles are always left undetected. To tackle this situation for detecting all the circles, we used parallel line interpolation as explained in the next section.

C. Parallel Line Interpolation

As explained above, all the circles are not detected by the Hough Circle Transform. We use a special property of the OMR sheet to find all the circles. The responses are one after the other and there are 4 options available. This means that there exists four parallel lines passing through all the real circles. Though Hough circles is not able to find all the circles, it is able to find majority of them. This means, that if we interpolate four parallel lines passing through detected circles centers, it will pass through missing circles too. Based on the idea above, we come up with an error minimization method to find the four parallel lines passing through detected circles. We define the equation of the first parallel line as

$$my = x + c \qquad (1)$$

where y is the Y coordinate, x is the X coordinate, m is the inverse of the slope of the parallel lines, c is the y intercept for line.

Slope for lines parallel to it will be same, only their y intercept will change, as the distance between two adjacent parallel lines will be same,

So the other lines will have the respective equations:

$$my = x + c + \alpha \qquad (2)$$

$$my = x + c + 2\alpha \qquad (3)$$

$$my = x + c + 3\alpha \qquad (4)$$

where α is the difference in y intercept of two adjacent lines. The joint equation of four parallel lines is

$$z = (my - x - c) * (my - x - c - \alpha) * \\ (my - x - c - 2\alpha) * (my - x - c - 3\alpha) \qquad (5)$$

We measure the error, E as

$$E = \sum_i [(my_i - x_i - c)* \\ (my_i - x_i - c - \alpha)* \\ (my_i - x_i - c - 2\alpha)* \\ (my_i - x_i - c - 3\alpha)] \qquad (6)$$

where, (x_i, y_i) are the x and y coordinates of ith circle centre.

For the line to best fit the circles data, E should be minimum. We optimize the parameters of the line m, c and α to find our set of parallel lines. Fig. 5 shows the lines passing through the circles. One can use a gradient descent [9] to minimize the Loss function and find out optimal set of m, c and α. But since there were only 3 parameters to optimize upon we find it more convenient to iterate toward all the possible values of the parameters with small increments and choose the parameters which gives the minimum loss. After we have the equations of the parallel lines, we know that actual circles should also pass through these parallel lines, so we will detect all of them now.
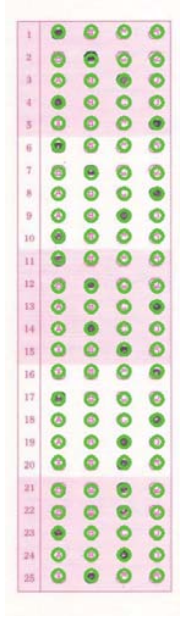
Fig. 6 Input Image overlapped with all the circles after full detection

D. Estimating Circles

We have got a set of four parallel lines passing through all the circles. Each line passes though one option, e.g. Line 1 passes through option 1 circle of all the questions. We have some missing circles which we need to find. We consider only those circles whose centers are less than their radius away from the line. Lets call the circles with radius distance away be circles Included.

For circle C_i in circles Included, we can find all the other 3 circles for that particular question as they exist on a line L which is perpendicular to our 4 parallel lines and passes through the center of circle C_i. Thus at the points of intersection of line L with our parallel lines are the centers of all the 4 options of a particular question.

The slope of parallel lines is $1/m$. Thus the slop of line L will be $-m$ by using this property of perpendicular lines $m * m_{perpendicular = -1}$ Thus equation of line L will be

$$y = -mx + c_{new} \tag{7}$$

This line passes through circle C_i, let p,q be the x,y coordinates of the center of circle C_i. Thus,
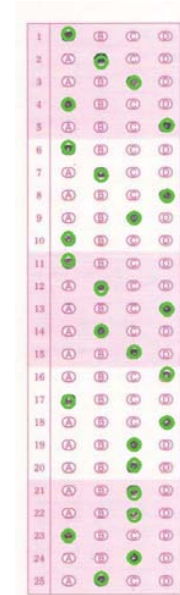
$$q = -mp + c_{new} \tag{8}$$

$$c_{new} = q + mp \tag{9}$$

Now we know the equation for line L. The other options for the question can be found by solving equation of line L with each of the parallel lines. Solving (7) with (1)-(4) simultaneously gives the x,y coordinates of the four options.

$$X = ((-cc + c_{new} * m)/(m * m + 1)) \tag{10}$$

(a) Binary image of the input image

(b) Input Image overlapped with detected responses

Fig. 7 Image illustrating black circle detection for tracking the student responses

$$Y = c_{new} - m * x \tag{11}$$

where $cc = c$ for line 1, $c + \alpha$ for line 2, $c + 2\alpha$ for line 3 and $c + 3\alpha$ for line 4.

By this technique, even if there is only one circle out of four for a particular question in circles Included, we are able to estimate the remaining circle. We have also tried to find the grid by finding the horizontal lines perpendicular to the vertical lines and find the ROI. The approach of finding grid also gives the same kind of results and can be used interchangeably. As hough transform works fairly well leaving very few circles undetected, there is atleast one circle detected for every question. Thus we are able to estimate the correct centers for all the circles using current approach. The approach of finding grid has advantage and will be independent of the above assumptions. Fig. 6 shows all the circles detected.

E. Black Circle Detection and Marking

After we have successfully located the centers of all the circles, we can now check for each circle which one is marked and evaluate the sheet. To check whether the circle is blackened or not, first we convert the input image into a binary image using binary thresholding [10], [11]. The converted binary image is shown in Fig. 7. Then for a circle C, the sum of intensities of the binary image is calculated in the area of a square having C as its incircle. If the sum exceeds $area of square * 200$ it is not marked, else it is marked. The detected black bubbles are shown in Fig. 7. Now as we know
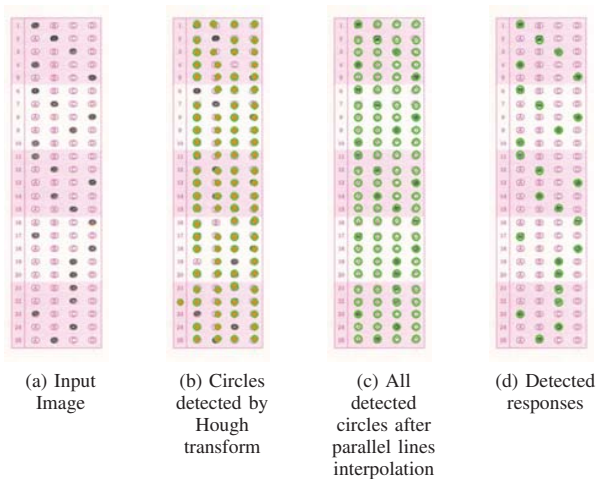
(a) Input Image | (b) Circles detected by Hough transform | (c) All detected circles after parallel lines interpolation | (d) Detected responses

Fig. 8 Process for the straight image

(a) Input Image | (b) Circles detected by Hough transform | (c) All detected circles after parallel lines interpolation | (d) Detected responses
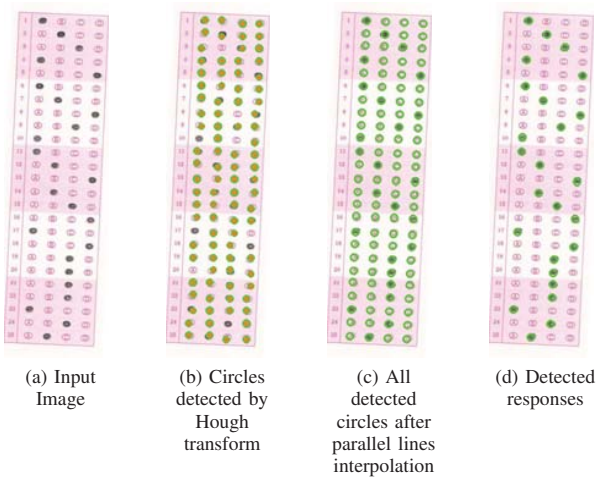
Fig. 9 Process for the tilted image

exactly which option of a question a student has marked, it is then compared to a database containing actual answers and the OMR sheet is evaluated.

IV. RESULTS

We tested the proposed algorithm on straight images as well as on tilted images. After full processing we were able to detect and evaluate these OMR sheets Figs. 8 and 9 without leaving any circles and responses undetected . However there were images where we were unable to detect few circles. We tested 10 different scanned images of OMR sheets and the summary of the results is given in the following Table I.

V. LIMITATION

Though the algorithm is more generalized than those already there, we need to define some hyper-parameters. Like we need to have a rough idea about the radius of the circles in

TABLE I
EXPERIMENTATION RESULT SHOWING AVERAGE NUMBER OF CIRCLES
DETECTED DURING DIFFERENT STAGES OF THE ALGORITHM

	Circles detected by Hough transform		Circles detected after all processes		Circles not detected after all process
	TP	TN	TP	TN	
Average	77.29%	2.10%	99.23%	0.0%	0.76%

[a] TP (True Positive) - Denotes the detected circles which were actually true circles, [b] TN (True Negative) - Denotes the falsely detected circles

pixels. This problem too could be generalized by choosing a wide range of radius while using Hough circle transform and choosing the range which produces the most circles. We also need to specify the size of the accumulator for hough transform. That again can be estimated based on the radius which can be estimated as mentioned previously. The algorithm can be easily extended to multiple valid options and multiple columns OMR sheets.We have considered the options to be circular in shape. However for the other shape OMR sheets we have to use specific algorithms to detect those shape in OMR sheet.

REFERENCES

[1] Sumitra B. Gaikwad, "Image Processing Based OMR Sheet Scanning," International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE).

[2] Rusul Hussein Hasan, Emad I Abdul Kareem, "An Image Processing Oriented Optical Mark Reader Based on Modify MultiConnect Architecture MMCA," International Journal of Modern Trends in Engineering and Research (IJMTER) Volume 02,Issue 07, (July 2015).

[3] Qi-Chuan Tian and Quan Pan and Yong-Mei Cheng and Quan-Xue Gao, " Fast algorithm and application of Hough transform in iris segmentation," International Conference on Machine Learning and Cybernetics, 2004.

[4] Gorgevic Dejan1, Grcevski Nikola2, Mihajlov Dragan1, "A Simple System For Automatic Exam Scoring Using Optical Markup Reader," Applied Automatic System AAS'2000.

[5] S, Rakesh and Atal, Kailash and Arora, Ashish, " Cost Effective Optical Mark Reader," International Journal of Computer Science and Artificial Intelligence, 2013.

[6] Deng, Hui and Wang, Feng and Liang, Bo, "A Low-Cost OMR Solution for Educational Applications," 2008.

[7] N. H. Lestriandoko and R. Sadikin, " Circle detection based on hough transform and Mexican Hat filter," 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Tangerang, 2016, pp. 153-157.

[8] OpenCv Documentation, (Online). Available: https://docs.opencv.org/3.1.0/da/d53/tutorial_py_houghcircles.html

[9] Sebastian Ruder, " An overview of gradient descent optimization algorithms," CoRR, abs/1609.04747, 2016.

[10] Puneet and Naresh Garg, " Article: Binarization Techniques used for Grey Scale Images." International Journal of Computer Applications 71(1):8-11, June 2013.

[11] Devi, H. K. A, " Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script. " Ancient Asia. 1, pp.161165. DOI: http://doi.org/10.5334/aa.06113.