



# International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: [www.ijarcsse.com](http://www.ijarcsse.com)

## Implementation of OMR Technology with the Help of Ordinary Scanner

\* Garima Krishna, Hemant Ram Rana, Ishu Madan, Kashif, Narendra Sahu,

Department of Computer Science And Engineering

College of Engineering, Roorki, India.

**Abstract**—In this fast and furious time, people don't want to invest their time in processing. They just want to give an input and take an output immediately. This project is actually an application that we are developing which enables implementation of OMR using an ordinary scanner. It provides tools to the user to design an OMR sheet based on the layout they want. The design of the sheet will be stored as image file format (JPEG). The user can take as many print outs as required, distribute it among others from whom information is desired, and get the filled sheets scanned. The scanned image files will then be provided as input to the software, processing will be done, value of filled fields will be extracted and then the data will be manipulated as instructed by the user. The implementation is done using Java. We are using Net beans IDE so as to easily manage the graphical layout and all the resources. For the GUI java was chosen mainly because of its platform independence and lightweight of the components in java swing.

**Keywords-**

### I. INTRODUCTION

Optical Mark Reader (OMR), also called “mark sensing”, is a method of scanning technology in which data is input via marks made in predefined positions on a form and entering data into a computer system. Therefore, OMR is best for handling discrete data, where values fall into a limited number of values. For examples, gender, occupation, religion, etc.

OMR is a technology that detects the absence or presence of a mark, but not the shape of the mark. OMR software interprets the output from the scanner, and translates it into the desired ASCII output. Forms are scanned through an OMR scanner. The forms contain small circles, referred to as bubbles, or boxes that are filled in by the respondent. Optical Mark Reader (OMR) reads marks written by pencil or ballpoint pen in the pre-defined positions on the questionnaire sheet. The OMR can judge the existence of written marks by recognizing their depth (darkness) on the sheet.

The simple model of thinking was applied, this work was intended to develop through three phases: (1) Input, (2) Process and (3) Output. First phase, there is only a module for covering the key objective of this phase that is an arrangement file system in searching and selecting the desired working folder and a file assigned to be the key answer sheet. Secondly, Process phase consists of two modules: recognizing and analysis. Recognizing module is the critical of whole program carrying the procedure that translates the image files to the digital data. The last Phase, called Output, responses in facilitating the end user to provide the necessary report. This is the simplest phase that follows the principle of data structure process.

A mark is response position on the questionnaire that is filled by survey collectors or respondents. If the position is filled (marked) then the scanner will registers significantly less light than the surrounding questionnaire. In order to be detected by a scanner, a mark has to be positioned correctly on the questionnaire and be significantly darker than surrounding of the questionnaire. Since most scanners use infrared light, an acceptable mark should be dark enough to absorb a high proportion of infrared light contrasted to questionnaire.

### II. OBJECTIVE

- Our main objective is reduce cost of implementation of OMR technology, cost of sheets is around 0.25
- To make our application available on a wide range of scanners. Any ordinary scanner will work with our application
- The application can work on damaged sheets only up to a certain level. Damage may be due to inclusion of stains in the sheet or wetting of sheets or degradation due to use of improper scanner.
- 

### III. DIGITAL IMAGE PROCESSING

An image may be defined as a two-dimensional function,  $f(x,y)$ , where  $x$  and  $y$  are spatial (plane) coordinates, and the amplitude  $f$  at any pair of coordinates  $(x,y)$  is called the intensity or gray level of the image at that point. When  $x$ ,  $y$  and the intensity values are all finite, discrete quantities, we call the image a digital image, and digital image processing refers to processing digital images by means of a digital computer.

The smallest element of an image is called a pixel.

There are different types of images used for digital image processing. They are:

**A. GRAYSCALE IMAGE**

This type of image uses 1 byte per pixel, and each pixel intensity ranges from 0-255.

**B. COLOR IMAGE**

It uses 3 bytes per pixel, as it has three components Red, Green and Blue. The intensity range of each component is also 0-255.

**C. BINARY IMAGE**

It uses only 1 bit per pixel. The pixel value is either 0 (for black pixel) or 1 (for white pixel).

#### IV. TECHNOLOGY

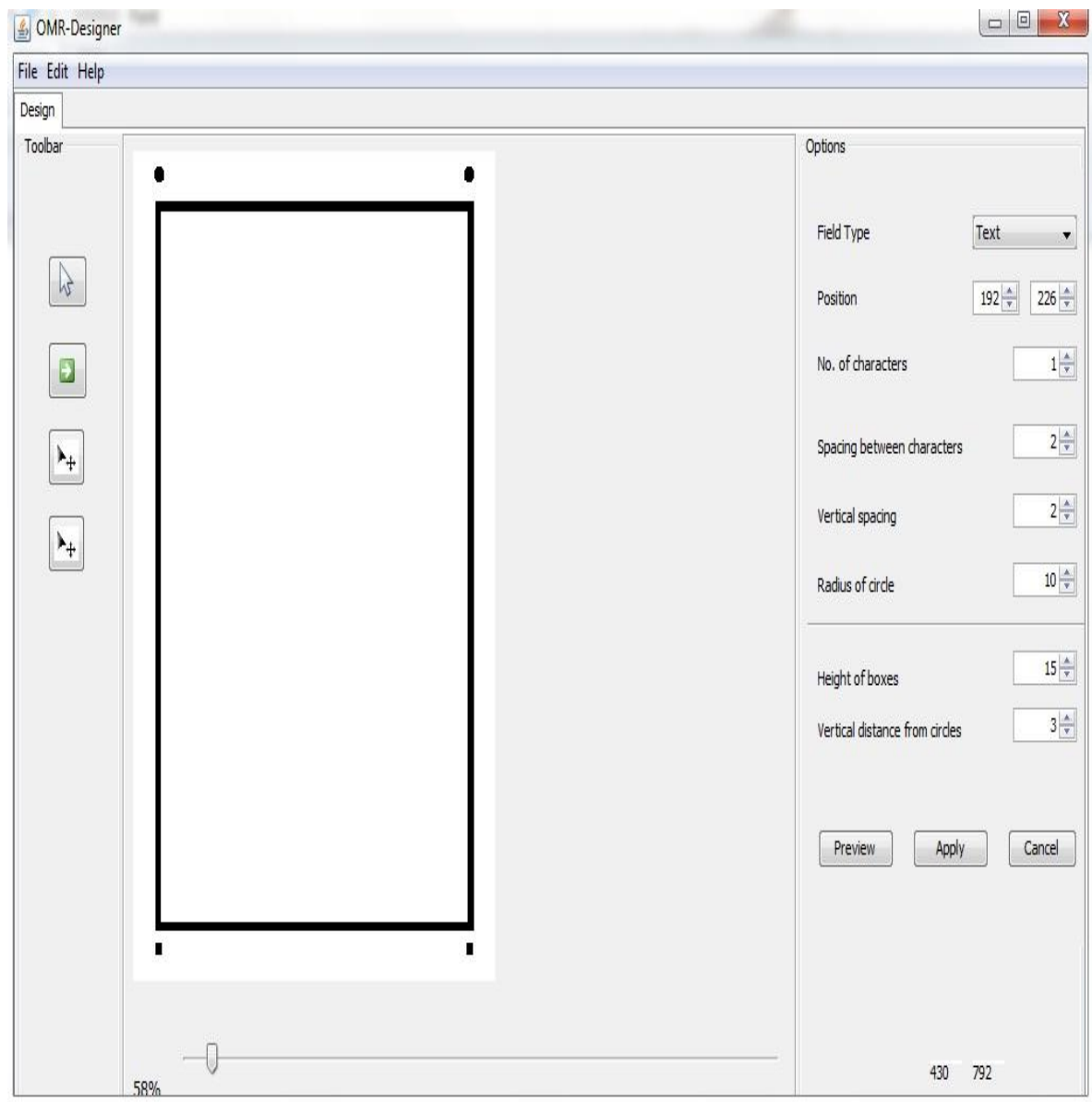
**A. REQUIREMENTS**

Programming Language: Java

Extra software: Netbeans IDE

Hardware: Simple scanner, printer

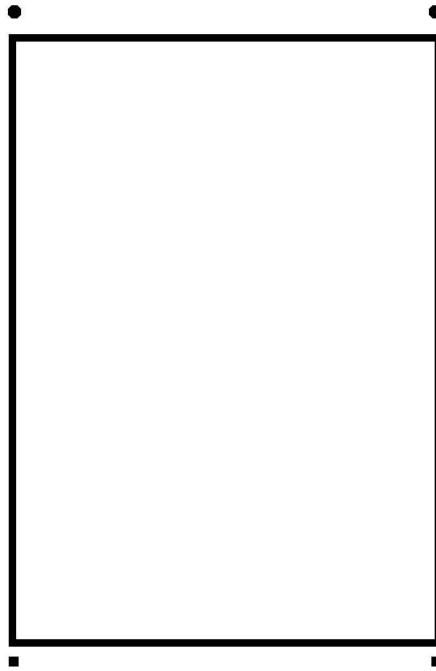
Here is a screenshot of our application



**Figure 1**

**B. ALGORITHMS**

The layout of OMR sheet to be used is as shown in figure 2:



**Figure 2**

- 1) *Finding corner points of bounding box:* To find the corner points we need to design algorithm for those cases when the sheet is not being kept straight during scanning, due to human error. In those cases rotation up to 11 degree is acceptable.

Now for finding the corner points we first need to know whether our sheet was rotated towards the right, or towards left, or not rotated at all. For this a random point from the top is taken, and traversed downwards till a black pixel is encountered. From that point traversal is done towards the left. If there are maximum black pixels, it means that the sheet is rotated towards the right; otherwise it is towards the left.

Using the appropriate algorithm the corner points can be found.

The algorithm for finding the appropriate points in case the image is tilted towards the left is:

img: Image to be analyzed

w: Width of image

h: Height of image

ratio: Minimum limit for number of black pixels

blackCount: Number of black pixels

findPoints(): Finds the corner points.

leftUp(): Finds upper left corner point.

rightUp(): Finds upper right corner points.

leftDown(): Finds down left corner point.

rightDown(): Finds down right corner point.

findPoints()

1. ratio= 0.8\*w;
2. blackCount=0;
3. leftUp();
4. rightUp();
5. leftDown();
6. rightDown();

leftUp()

1. for i =2 to h/4 do
2.     for j=2 to w/4 do
3.         if( img[j][i] is BLACK &&   img[j-1][i] is WHITE then
4.             while j<w do
5.                 if img[j+1][i] is BLACK then blackCount++

```

6.         else if img[j+1][i] is WHITE then count number of consecutive white pixels and store in
           whiteCount
7.         if( whiteCount > 5) then
8.             i=i-1; /*Jump 1 pixel upwards to restrict control to move along the boundary(Refer figure
           3) */
9.         end while
10.        end if
11.        if( blackCount > ratio)
12.            print i,j;
13.            return;
14.        else
15.            continue;
16.        end for
17.    end for

```

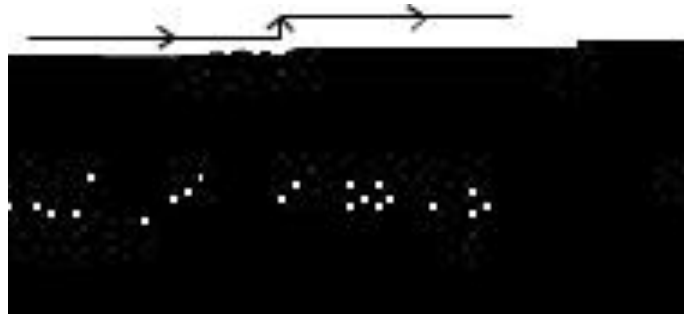


Figure 3

In the similar manner other points can be found

- 2) *Checking orientation:* Once the corner points of the bounding box are found, the exact angle of rotation needs to be calculated. For this, 2 corner points at the top need to be taken. Refer figure 4 for two types of rotation possible. Let  $(x_1, y_1)$  be co-ordinates of top left corner, and  $(x_2, y_2)$  be co-ordinates of top right corner.

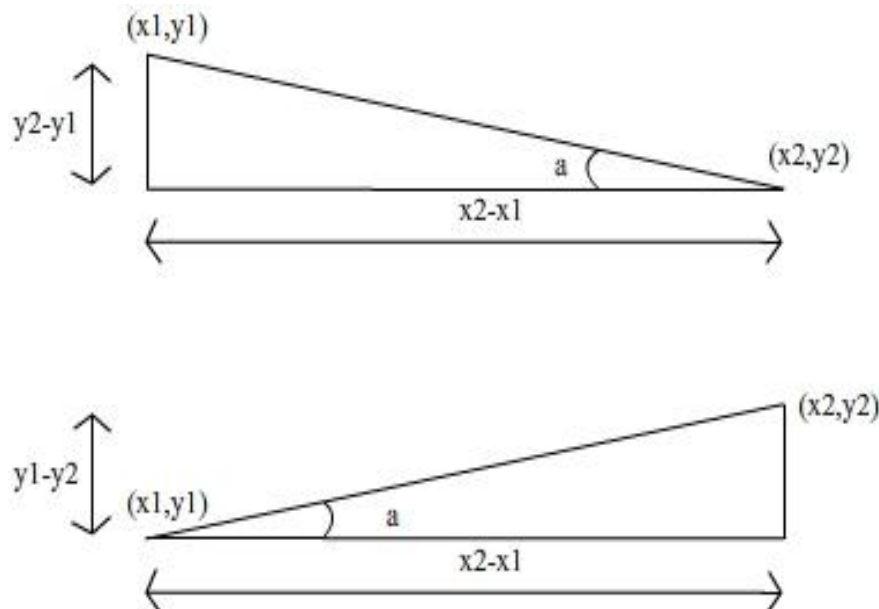


Figure 4

Angle 'a' needs to be found for finding the rotation

So,  $\tan a = \text{Perpendicular} / \text{Base}$

$$\tan a = (y_2 - y_1) / (x_2 - x_1)$$

$$a = \tan^{-1} (y_2 - y_1) / (x_2 - x_1)$$

After finding the angle we can then rotate the image to make it straight.

- 3) *Reading the fields:* While the OMR sheet was being designed by the user previously, the co-ordinates of the fields were being stored with other attributes, like horizontal and vertical space between circles, number of characters, type of field, radius of circles, etc.

A rough estimate of value of each attribute in the scanned image can be easily taken.

For example, let's assume we need to find the starting x co-ordinate of an attribute in the scanned image.

Then,  $X = (W/w) * x$ ,

where, X is x co-ordinate of scanned image needed to be estimated

, x is x co-ordinate of the original sheet designed by the user

, w is width of the original sheet

, W is width of the scanned image

Using these attributes starting point and ending points of each circle can be found. Then the task of finding whether it is bubbled or not is quite easy. Inside these bounds the number of black pixels are counted.

Let number of black pixels be b, and radius of the circle be r.

If  $b > 0.9 * \pi r^2$ , then the bubble was filled, otherwise it wasn't.

## V. RESULTS

By using the algorithms above we found our results to be nearly accurate. It was successful in reading the fields. However it won't work when the image is significantly damaged.

## VI. CONCLUSION

Previous analysis presented earlier shows clearly the advantages of using automated OMR procedures instead of manual. The main advantage of the presented software is the ability to work with every single scanner of the market, needing no extra drivers whatsoever.

The program is also small and compact, easily fitting in a diskette. No special equipment is needed thus retaining a very low cost. The forms themselves need no special printing (IR dedicated OMR scanners need red ink forms), and they can be printed in every printer and photocopied in large quantities. Finally the algorithms can work and manage the pivot driving areas by themselves, freeing the user from the tedious and error prone task of manually marking these points.

It must be noted that most commercially available programs depend on user intervention for the pivot marking

## REFERENCES

- [1] Saunders, M., Lewis, P., Thornhill, A., "Research Methods for Business Students", 3<sup>rd</sup> edition, Prentice Hall, England, 2003
- [2] Research Paper By NIKOLAOS I. ANDROULIDAKIS, Tanvi Sharma and Prof. Niket Bhargava (January-December, 2012)
- [3] Bergeron B P (1998) Optical Mark Recognition: Tallying information from filled- in bubbles. Postgraduate Medicine, 104(2): 23-24.
- [4] Bourke P (1998) BMP image format. <http://astronomy.swin.edu.au/~pbourke/dataformats/bmp/>.
- [5] Dillman D A (2000) Mail and Internet Surveys: The Tailored Design Method. Pp.418-19. John Wiley & Sons, New York.
- [6] Haigh S (1996) Optical Character Recognition (OCR) as a Digitization Technology. National Library of Canada - Network Notes #37. <http://collection.nlc-bnc.ca/100/202/301/netnotes/netnotesh/notes37.htm>. (DOV 18 June, 2003).
- [7] Hetzl S (1998) The .BMP file format. <http://www.fortunecity.com/skyscraper/windows/364/bmpffrmt.html>.
- [8] Kalley G and Plastock R A (1986) Theory and Problem of Computer Graphics. pp 80-86. Schaum's outline series, McGraw Hill.
- [9] Kia O E (1997) Optical Mark Recognition-FAQ. <http://tev.itc.it/people/modena/kia-ocr-faq.html>.
- [10] Lopresti F and Naphtali Z S (1996) Statisticians' Lib: Using Scanners and OMR Software for affordable Data. Input. <http://www.nyu.edu/its/pubs/connect/archives/96fall/loprestistats.html>.
- [11] Moore B (2002) Surprise! OCR and OMR Are Hotter Than Ever. <http://www.idat.com/a-ocromr.html> (DOV 16 Sep, 2003).
- [12] Nartker, Rice S V and Jenkins F R (1995) OCR Accuracy: UNLV's 4<sup>th</sup> Annual Test. Inform.
- [13] Schantz H F (1982) The History of OCR: Optical Character Recognition. Recognition Technologies Users (Association).
- [14] Srihari S N (1992) High-Performance Reading Machines. Proceedings of the IEEE, 80(7): 1120-32.
- [15] Zahniser J (1997) RAF technology helps U.S. Postal Service tame world's largest data stream, saving millions a year. Business Wire. <http://www-rcf.usc.edu/~wdutton/comm533/OCR-SHEE.htm>. No. 466-261, WSEAS NNA-FSFS-EC 2003, May 29-31, 2003
- [16] Jill Gilbert, "How to Do Everything With Your Scanner (How to Do Everything Series)" McGraw-Hill Companies, 2001
- [17] "Color Scanning Handbook, The: Your Guide to Hewlett-Packard Scanjet Color Scanners", Prentice Hall PTR, 1997

- [18] Haag, S., Cummings, M., McCubrey, D., Pinsonnault, A., Donovan, R. Management Information Systems for the Information Age, 3rd ed., Canada: McGraw-Hill Ryerson, 2006.
- [19] Gonzalez, R. C. and Wintz, P. Digital Image Processing, 2nd ed., Addison-Wesley, Reading, Massachusetts, pp. 30-31 and 130-134 , 1987.
- [20] McConnell, Steve . Software Estimation: Demystifying the Black Art. N.C., Microsoft Press, 2006.
- [21] Heywood John, Assessment in Higher Education. Jessica Kingsley Publishers, London, Pages 350-372, 2000.
- [22] TAWIP. Image Capture, Image Recognition and Image Processing.[Online],(n.d.),Available from : <http://www.tawpi.org/image-capture-recognition-processing.html>
- [23] Payne DA. The Assessment of Learning — Cognitive and Affective. Lexington, Ky: D.C. Heath and Company: 255,272, 1974.
- [24] Lowe Doug. Java™ All-in-One Desk Reference For Dummies. River Street, Hoboken, Wiley Publishing, 2005.
- [25] French, Christine L. “A Review of Classical Methods of Item Analysis”. Presened at the annual meeting of the Southwest Educational Research Association, New Orleans, 2001.
- [26] Raksakietisak Sunee. The Development of Online Item banking and Testing System. Masters Thesis, Faculty of Science, Srinakharinwirot University, 2003.
- [27] Pornsiriprasert Nisarath. Design and development of page segmentation program for character recognition. Masters Thesis, Chulalongkorn University, 2002.