19th International Computer Science and Engineering Conference (ICSEC)
Chiang Mai, Thailand, 23-26 November, 2015

Physical Form Handling in Challenging Imaging Environment

Teera Siriteerakul
Faculty of Science
King Monkut's Institute of Technology Ladkrabang
Email: teera@tboxteam.com, kfteera@kmit.lac.th

***Abstract*—Even in a highly digitized world most of us live in, paper or physical forms (survey, multiple choices exam, etc.) are still playing an important roles. They are not only cheap and convenient, but also the only legal form of documents according to laws in many countries. Currently, these physical forms are mostly handled by hand. Otherwise, we need special forms with special filling instructions. To ease this process, this research proposed a non-supervised learning technique for handing physical forms in challenging imaging environment. This technique works on an assumption that a typical filled form has more blank entries than filled entries and image in each group (blank and filled) are closer to each order in image space. This technique considers special designed forms which can be printed in any paper and filled without any special instruction. With a standard webcam, the proposed method is demonstrated its capability of physical forms handling in challenging imaging conditions.**

Keywords—Optical Mark Reader, Physical Form Digitization, Multiple Choice Grader, Survey Handling, QR Pattern Finder

I. Introduction

Information gathering is always important in both academic and business world. One type of these information gatherings can be multiple choice exam or survey. Although digitized forms has been developed and utilized, paper-based or physical forms are still playing an important roles in many countries dues to its cheapness and tangible nature. However, processing physical forms can be challenging. Typically, we can utilize manual labor or we need a special form with special instruction together with special machine to extract information from these physical forms.

There are several attempts to solve this problem. One notable attempt was presented by Dell et al. [1]. In their work, paper forms recoding vaccine statistics in rural health center in Mozambique was digitized by using a smart phone with camera. First, a picture of a form to be processed was taken by a smart phone camera attach on a plastic stand to stabilize the image. Then, an image registration process was done using RANSAC and SURF. After that, each bubble was classified to be filled, partially filed, and unfilled using trained Support Vector Machine (SVM). Although the result accuracy were roughly 99%, there are several limitations to this approach. First, the camera has to be stabilized by a special device in order to take images. Second, SVM needs to be trained which may cause the system to be non-adaptive. Lastly, their system took 25 seconds to process each form.

There are also several commercially available OMR applications. For example, ZipGrade [2] can grade multiple choices exam taken with one of their three available types of answer sheet. The system seems to work under an assumption that the biggest difference of intensity between two closest choices within one question determine filled and unfilled. This is a speculation since there is no publication of this system. In any case, after some evaluation, their system cannot handle all filled choice in one question and the oval must be filled enough to make the threshold. See fig. 1 for illustrated example.

To ease the process of physical form handling, this paper proposed a method to digitized physical forms with three target objectives. First, the method must be fast. Second, it must be able to handle low quality imaging without any special device. Lastly, the method must be adaptive enough to handle forms filled without any special instruction.

The first and second objectives can be handled simply by utilizing QR Finder Pattern [3]. Upon placing four QR Finder Pattern on a physical form, we can correct perspective transformation relatively fast without any special device. Then, to make the system adaptive in low quality imaging, this paper proposes a two-parts method which can identify filled answer without any training.

The first part of the proposed detection method pre-classifies answers in filled and blank answers by assuming two facts: 1) that there are more blank entries than filled ones and 2) each group (blank/filled) are closer within its group in the image space. Thus, if we calculate differences between one answer and its neighbors, a filled answer will have more high difference values computed from that answer to blank ones

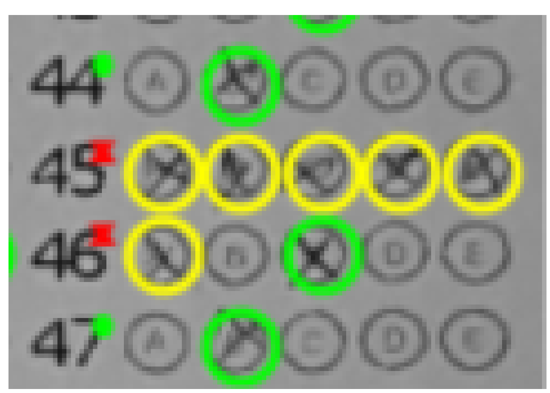

Fig. 1. An example of failure case of ZipGrade where all filled question and the first answer of the question after were not detect (show in yellow).

This project is funded in part by grant from King Mongkut Institute's of Technology Ladkrabang.

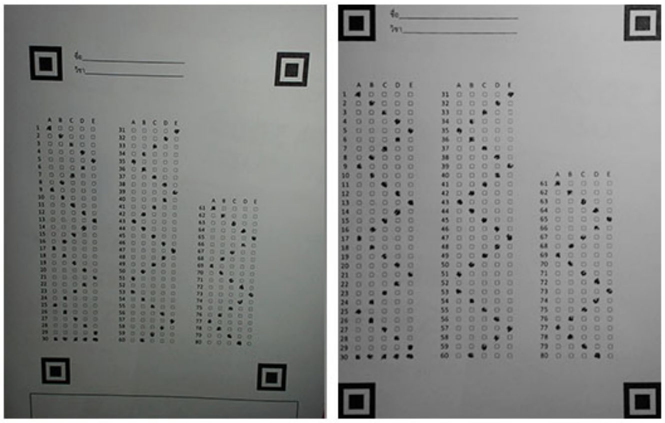

Fig. 2. An image taken by the proposed system (left) and the image after perspective correction (right).

(since there are more blank answers). On the other hand, a blank answer will be calculated to have fewer high differences. Thus, after we compute mean of differences within a local neighborhood, the filled answers will have higher mean of differences than the blank answers. We can use this fact to separate the answers into two groups. Then, the second part of the proposed method utilizes k-Nearest Neighbor classification process using the pre-classified labels from the first part.

The contribution of this paper lies in the techniques which together can handle a physical form without any special device or any special filling instruction. The proposed technique can automatically re-adjusted perspective transformation and adaptively search for filled entries. The process of the proposed method can be done in real-time (within 1-2 seconds after form imaging).

The rest of this paper is organized as follows. First, the proposed system is described in details in section II. Then, experiment setup, result, and discussion are provided in section III. Finally the concluding remarks is given on section IV

II. Proposed System

This section describes system overview as well as the methods and their rationalization of each processing step. The proposed system works on a specially designed form which an be printed on any paper and filled without any special instruction.

A. System overview

The system will start by using a camera to capture an image of the filled form. This can be done without any special device since the proposed method can handle some degree of perspective transformation (see Fig. 2, left). After that, the system will search for four QR Pattern Finders which the searching technique has been modified to handle some degree of perspective transformation. With four detected locations, the image is then transformed using recovered perspective transformation (describes in section II-B).

With the corrected perspective, images of all the answers can be found by simply looking at pre-determined locations.

Fig. 3. Overall process of the proposed system.

Once all answers are located, the proposed system can detect filled answer by process described in section II-C. See Fig. 3 for illustration of this overall process.

B. Resolving Perspective Transformation

First, an image of a physical form needed to be aligned in both position and orientation. This paper assumes that the paper-based form lied flat on a planar surface (such as table top). Thus, only perspective transformation is needed. We can follow works done by [**?**] and start with a claim that the camera model could be written as $X = Hx$ here X is a vector of world plane coordinate, x is a vector of image plane coordinate, and H is a matrix transform.

$$\begin{bmatrix} XW \\ YW \\ W \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

With some derivation, we can found X and Y to be:

$$X = ax + by + c - 0d + 0e + 0f - Xxg - Xyh \tag{2}$$

$$Y = 0a + 0b + 0c + xd + yd + f - Yxg - Yyh \tag{3}$$

Thus, the final system of linear equations is

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -X_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -Y_1x_1 & -Y_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -X_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -Y_2x_2 & -Y_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -X_3x_3 & -X_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -Y_3x_3 & -Y_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -X_4x_4 & -X_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -Y_4x_4 & -Y_4y_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \\ X_4 \\ Y_4 \end{bmatrix} \tag{4}$$

Therefore, the camera model can be recovered using four points form a projected image (image taken by a camera). These four points can be found robustly using four QR Finder Patterns as mentioned in the previous section. Finally, after the input form image has been aligned, the location of all the answers can be found simply looking at the pre-determined positions.

C. Filled/unfilled Answer Separation

The proposed method conducts a non-supervised learning by applying two-part process in order to separate between filled and blank answers.The process starts by

1) Computing The Mean of Neighboring Means of Differences: The mean of neighboring means of difference is a threshold value used for separating between filled and blank answers. This number is computed locally for each question in the form. This part of the process starts by defining n as a number of neighboring questions which are n question closest to the processing question. Then, differences between each answer of these neighboring question and the others are computed. Thus, if there are a answers to each question, there will be a total of $a * n - 1$ values of differences. After that, a mean of differences is computed from these values. As a result, there will be $a * n$ values of these means of differences, one for each answer in the neighborhood. Finally a threshold which will be used to separate between filled and blank answers is a mean of these means of differences. Then, the process is repeated for all questions to label all answers as filled or blank ones.This process can be describe as in Algorithm **??**.

Algorithm for pre-determine labels of all answers.

Parameters: Images of all the answers and size of neighboring question n.

Output: pre-determined label of all answers.

For each questions q in this form
  For each answer a_i within n closest questions
    Compute distance d_{ij} from a_i to a_j where $i!=j$
    Compute mean of differences m_i of d_{ij}
  EndFor
  Compute mean m of means of differences m_i
  For each answer a_i within question q
    If $m_i > m$ then
      Label a_i as filled
    Else
      Label a_i as blank
    EndIf
  EndFor
EndFor

Algorithm 1 Pre-determining label of all answers.

2) Answer Calibration: After all answers are pre-determined as filled or blank, we can then calibrate them using k-Nearest Neighbor technique. In this step, the pre-determined predictions are used as label in the kNN process. Thus, for each answer choice, we determine the final prediction by majority label of the k nearest pre-determined labels. An example of pre-labeled and final labeled form can be seen in Fig. 4.

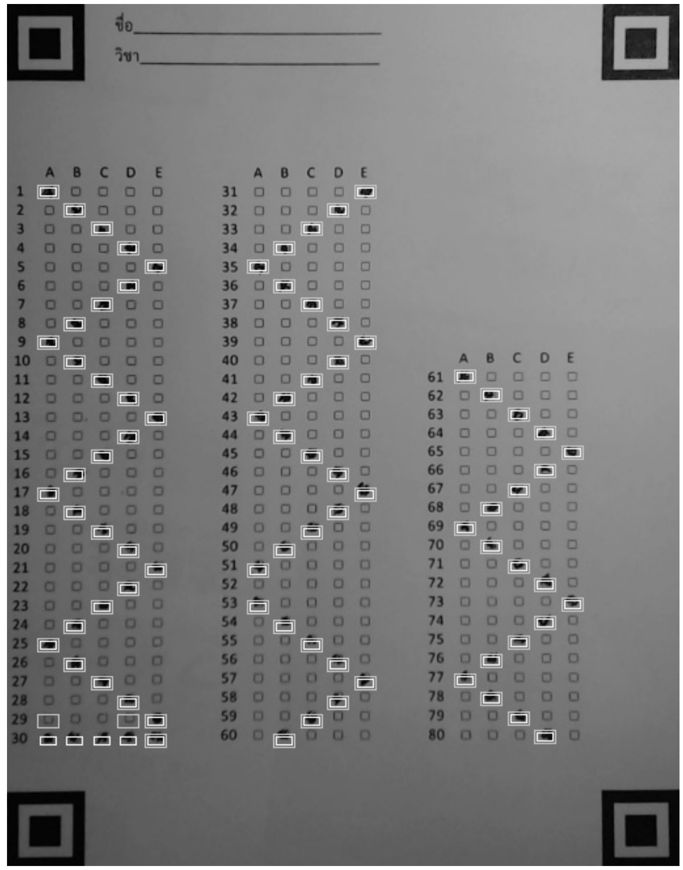

Fig. 4. An example of detection results. The gray outside boxes and the white inside boxes in the right image represent results from part 1 and part 2 of the detecting process respectively).

III. Experimental Results and Discussion

This section describes dataset, experimental setup and result, as well as discussion of results and future work of this research.

A. Dataset and Experimental Setup

Physical forms used in this experiment has been filled by 43 volunteers. Since each form contains 80 questions and there are 5 answers to a question, together there are 400 answers (represented as square) to be filled. With 43 volunteers, there are 17,200 total answers to be classified as filled or blank. The answers were filled without any special instruction. This resulted in wide variety of marking symbols (cross, check, filled, circle, etc.) which are illustrated in two pairs on the right in Fig. 5. Moreover, the forms were filled using pencil, dark, or blue pen.

The proposed technique has been implemented with C++ using Microsoft Visual Studio 2013 with OpenCV [5] library. The hardware setup including a non-dedicated Windows 32bits PC with 2.3GHz Intel Core i7 CPU and 16 GB of RAM. The camera used was c920 Logitech HD Pro Webcam. With this setup, the detection can be done in real-time (within 1-2 seconds) after taking an image of a form.

B. Experimental Setup and Results

There are two parameters which can be fine tuned in this method: size of neighbor n and k of kNN. Dues to highly coupling of the process, the parameters were tested together with variety of n and k equal to 1, 2, and 3.

Table I illustrates detection performances of the proposed method with difference combination of parameters. The highest accuracy reported was $k = 5$ with neighborhood size (n) of 3. Notice that the number of false positives (blank answers labeled as filled) reduced as the parameter k get higher. On the contrary, the higher the parameter k, the higher the number of false negatives (filled answers labeled as blank) which lower the accuracy. This may be the result of various blank answers contained enough noise to be confused as filled answers in the first-part of the detection step.

From the reporting result in table I, the size of neighborhood, n, improved the accuracy when change from 1 to 3, but degrade the performance when $n = 5$. This can be explain by non-uniformity of light. When $n = 1$, we have too few data to

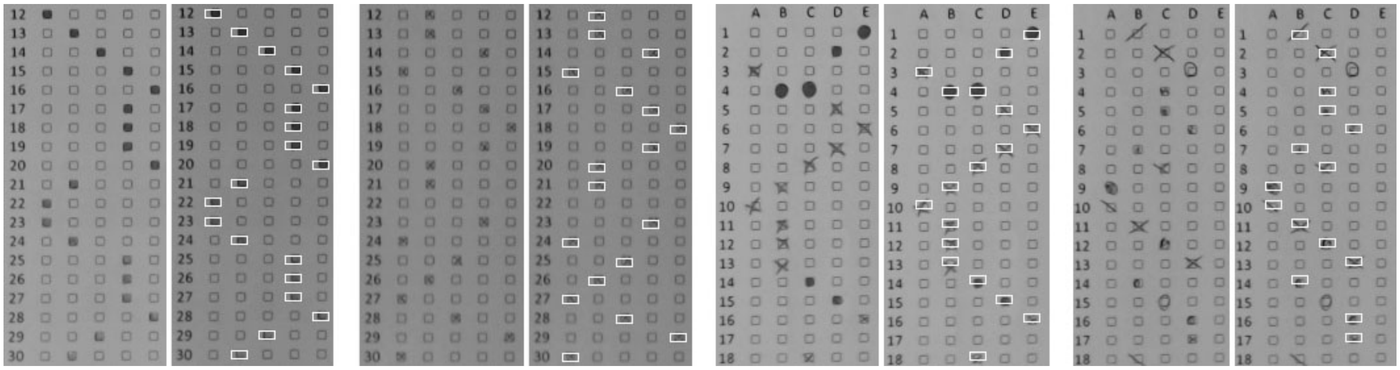

Fig. 5. An example forms that are easy (two pairs on the left) and hard (two pairs on the right) to detect.

TABLE I. DETECTING RESULT: REPORTING FALSE POSITIVE (F-P), FALSE NEGATIVE (F-N), AND ACCURACY (ACC) OF THE PROPOSED METHOD WITH n NEIGHBORHOOD AND k AS APARAMETER OF KNN. NOTE THAT THERE ARE 3,531 FILLED AND 13,669 BLANK ANSWERS.

n	k=1			k=3			k=5		
	F-P	F-N	ACC	F-P	F-N	ACC	F-P	F-N	ACC
1	344	50	97.7%	153	58	98.8%	93	98	98.9%
3	326	35	97.9%	129	61	98.9%	66	98	**99.1%**
5	334	38	97.8%	154	63	98.7%	78	96	99.0%

learn without supervision (unsupervised learning). Thus, when n increased to 3, we have more data to learn from which result in improving accuracy. On the other hand, when n increased to 5, we included answers from far away which may be lid by different light (or effected by shadow). Consequently, the answers from different lighting will be hard to learn using unsupervised method.

C. *Discussion and Future Work*

The overall accuracy of the proposed method is roughly 99%. This may be in the lower end comparing to bubble-filled OMR system. Nevertheless, if the form is filled as bubble-filled form (as in the left most pair in Fig. 5), the accuracy result were practically 100%. However, the proposed method is aiming to handle more challenging environment of form imaging without special instruction of form filling.

For future work, there are several area we could explore. We could go on to improve accuracy performance of processing forms filled naturally (without special instruction). On the other hand, one of the challenging questions to investigate is how can we detect someone marking an answer then crossing it of to mark another using unsupervised learning. Another direction we could explore is implementing the proposed method on a mobile device.

IV. CONCLUSION

The proposed technique has been demonstrated on physical forms filled without any special instruction. This can be useful for many countries where physical forms are still a majority. The proposed method can take picture using a Webcam without any special equipment or lighting. Then, with help of QR Finder Pattern, the method can automatically recover perspective translation. Then, the proposed non-supervised learning can be applied to detect filled answers with 99.1% accuracy.

REFERENCES

[1] Dell, Nicola, et al. ”Digitizing paper forms with mobile imaging technologies.” Proceedings of the 2nd ACM Symposium on Computing for Development. ACM, 2012.

[2] ”ZipGrade - IPhone and Android Grading App for Formative Assessment and Quizzes.” ZipGrade. Web. 9 Feb. 2015. https://www.zipgrade.com/.

[3] ”QRcode.com-DENSO WAVE, the Inventor of QR Code.” QRcode.comDENSO WAVE. Web. 9 Feb. 2015. http://www.qrcode.com/en/qrfeature.html.

[4] Criminisi, Antonio, Ian Reid, and Andrew Zisserman. ”A plane measuring device.” Image and Vision Computing 17.8 (1999): 625-634.

[5] Bradski, Gary. ”The opencv library.” Doctor Dobbs Journal 25.11 (2000): 120-126.