An Application for Automatic Multiple-Choice Test Grading on Android

Rodrigo Teiske China, rodrigo.china@aluno.ufabc.edu.br
Francisco de Assis Zampirolli, fzampirolli@ufabc.edu.br
Rogério Perino de Oliveira Neves, rogerio.neves@ufabc.edu.br
José Artur Quilici-Gonzalez, jose.gonzalez@ufabc.edu.br

Universidade Federal do ABC, Santo André, Brasil

Submetido em 01/09/2015
Revisado em 15/10/2015
Aprovado em 22/02/2016

Abstract:
There are currently several commercial automatic graders for multiple-choice test sheets, usually composed by a software and scanner bundle, resulting in high cost solutions. The popularity of devices such as laptops, tablets and smartphones with integrated cameras offer new possibilities to perform the same task at low costs. This paper presents an application for Android named “MCTest – Multiple Choice Test”, developed using the OpenCV library. The app was used for correcting hundreds of multiple-choice test sheets in real conditions and showed excellent results, precisely grading tests instantly and also providing statistical tools to analyze the results.

Keywords: Multiple-Choice Tests. Mobile Devices. Image Processing.

Um Aplicativo Android para Correção Automática de Testes de Múltipla-Escolha.

Resumo:
Atualmente existem diversos sistemas de correção automática de provas de múltipla escolha, geralmente compostos por um software e um escâner dedicado, resultando em custos altos. A popularização de dispositivos como *laptops, tablets* e *smartphones* com câmeras integradas abriu novas possibilidades para realizar a mesma tarefa com baixo custo. Este trabalho apresenta o aplicativo *“MCTest – Multiple Choice Test”* desenvolvido para sistemas Android utilizando a biblioteca OpenCV. O aplicativo foi usado em condições reais na correção de centenas de testes, demonstrando excelentes resultados, instantaneamente corrigindo testes com precisão, além de fornecer ferramentas para análise estatísticas dos resultados.

Palavras chave: Teste de Múltipla-Escolha. Dispositivos Móveis.
Processamento de Imagens.

Introduction

Multiple choice tests are the main evaluation tool used to grade individual performances in a test, particularly when the number of individuals are large, rendering other kinds of tests impractical There are several popular programs used to generate tests (CIENCIAMAO, 2015; FISTEUS et al., 2013), as well as software and devices aimed to this specific task, generally involving a significant investment. In addition, to the best of our knowledge, there is no specific literature describing methods and techniques to implement test grading or similar image processing problems. Using a technique known as *Optical Mark Recognition* (GREENFIELD, 1993) it is possible to acquire discrete data contained in predefined forms and, with an image *scanner*, detect the presence of marks in the reserved spaces. This technique often employs OMR specific scanners or image scanners, in which case software does the processing, sacrificing performance to the advantage of lower costs and flexibility when employing custom forms (SPADACCINI, 2009; NGUYEN et al., 2011; SEN et al., 2010; SAIPULLAH, et al., 2012). These examples would be hard to implement in places with limited access to technology, such as in 3rd world countries or in organizations running on low budgets.

With the modern popular technology of mobile devices with high processing capabilities and integrated digital cameras, such as smartphones and tablets, there is a low cost alternative to the task of grading multiple choice tests. On the other hand, the use of these devices makes the process subject to errors and increases the demanded time, New mobile technologies allow small, widely available devices with built in digital cameras and good processing power to become a viable alternative to the above mentioned solution, despite software solutions tend to be more time consuming and error prone, due to the manual acquisition of images, difficulty in positioning the sheets at the right angle and distance, parallel to the camera lenses. Furthermore, factors related to image quality, such as uneven lighting, shadows and camera distortions add new technical difficulties. Nevertheless, there are several OMR and OCR programs available in the market, including applications for smartphones and tablets capable of recognizing marks, barcodes and QR-codes (SPADACCINI, 2009;

NGUYEN et al., 2011; SEN et al., 2010; MOLLAH, et al., 2011; CHINNASARN; RANGSANSERI, 1999) which have dealt with such limitations in a variety of ingenious manners.

There are also several papers and application quizzes demonstrating how to use mobile devices and web services, such as the iQuiz (RODRIGUES, et al., 2013), an integrated assessment environment to improve Moodle quiz. The MobiMonitor (LUCENA, et al., 2014) is a mobile App for Monitoring distance courses in the Amazon region. An authoring verification environment for quizzes applied on the web is presented in (GORDILLO, et al., 2014) and an interactive environment suitable for classrooms that use mobile devices based on problem-solving skills is presented, (DEB, et al., 2014) showing how mobile devices and distance learning can mutually benefit from each other.

In this paper we describe the implementation of the App “MCTest”, an automated multiple-choice test grader for mobile devices running Android OS. MCTest as adapted for mobile devices from an earlier version developed for desktop PCs in MATLAB (ZAMPIROLLI; QUILICI-GONZALEZ; NEVES, 2013), which introduced a simplified answer sheet. When compared to regular OMR forms, it is compact and can be rescaled to variable sizes, according to the desired number of questions and possible answers for each question. Using any text editing software, such as LibreOffice or MS-Word, a table can be constructed to include rows and columns according to the test specifications, even differentiating among multiple templates, for variations of the same test, also supporting weighted questions. In the android version the image processing techniques used in the earlier version had to be adapted and optimized to be able to run in mobile processors and to use of the OpenCV library instead of the Matlab Image Processing Toolbox.

MCTest aims to help teachers and evaluators in the time consuming task of grading student tests, but other areas may benefit from its use, such as quickly accessing candidate scores in selection processes. Test correction data is stored for further analysis, giving evaluators the possibility to obtain qualitative information about the test and its overall efficiency.

Characteristics and Methods

A. MCTest characteristics

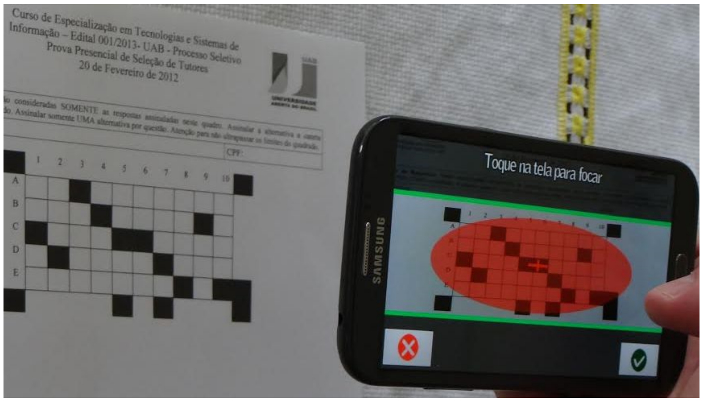
The MCTest App was developed in Java using the OpenCV library (Open Computer Vision) (OPENCV, 2015), imported into the developing environment ADT (Android Developer Tools, ver. 22.3) (ANDROID, 2015), which includes essential components for Android development. The App can be used with any device with a built in camera running Android OS version 4.0 or higher and is right now available on *Google Play Store* for free. Nonetheless, the camera resolution might limit the maximum number of questions and answers in the test sheet.

Given the printed tests and templates are ready to use, the grading process in the app is composed by three main stages:

- First it’s required to create a New Project or load a previously saved Project inside the App. The Project is stores its data in a subfolder with the project’s name;
- Then it’s required to save the templates with the correct answers;
- The tests are then compared with the corresponding template.

Figure 1 shows the app being used on a printed test and Figure 2 summarizes all the activities taking place within the App. Details for each figure will be further described in this section.

The answer sheet can be seen in Figure 3. In this example, the test has 10 questions and 5 possible answers for each question.



Curso de Especialização em Tecnologias e Sistemas de Informação – Edital 001/2013- UAB - Processo Seletivo
Prova Presencial de Seleção de Tutores
20 de Fevereiro de 2012

...o consideradas SOMENTE as respostas assinaladas neste quadro. Assinalar a alternativa e ...do. Assinalar somente UMA alternativa por questão. Atenção para não ultrapassar os limites do quadrado.

CPF:

Figure 1. Demonstration of capturing from printed test sheets.

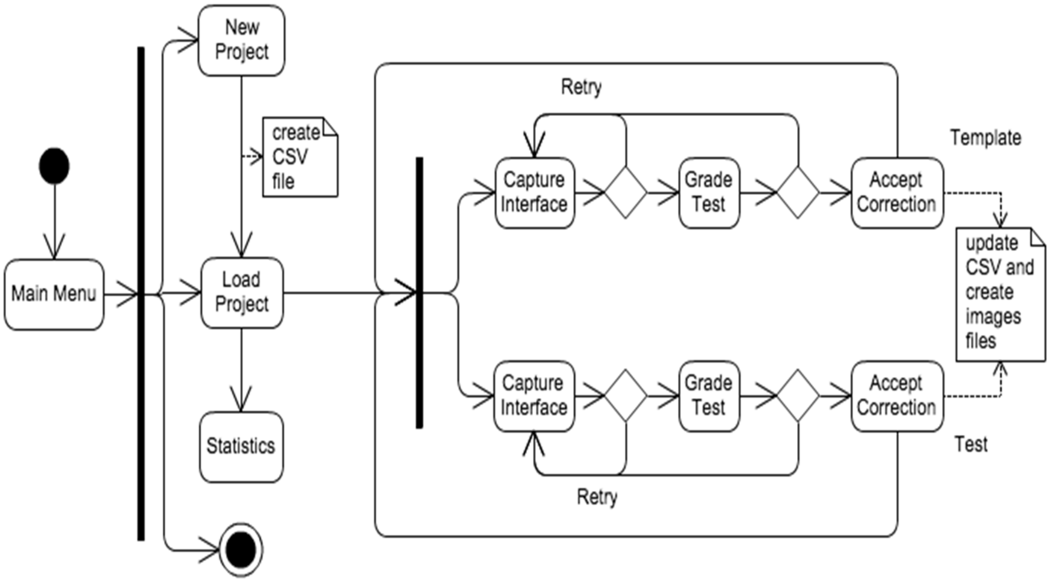
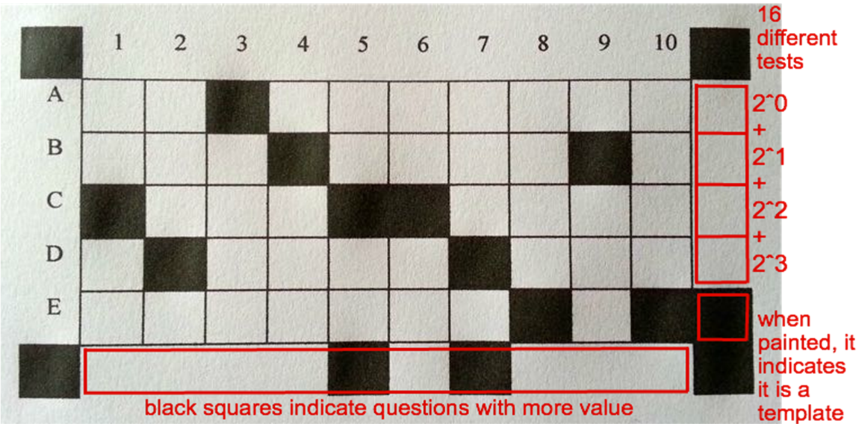


Figure 2. All stages of the MCTest App can be seen in this fluxogram.



1 2 3 4 5 6 7 8 9 10
A
B
C
D
E
16 different tests
2^0
+
2^1
+
2^2
+
2^3
when painted, it indicates it is a template
black squares indicate questions with more value

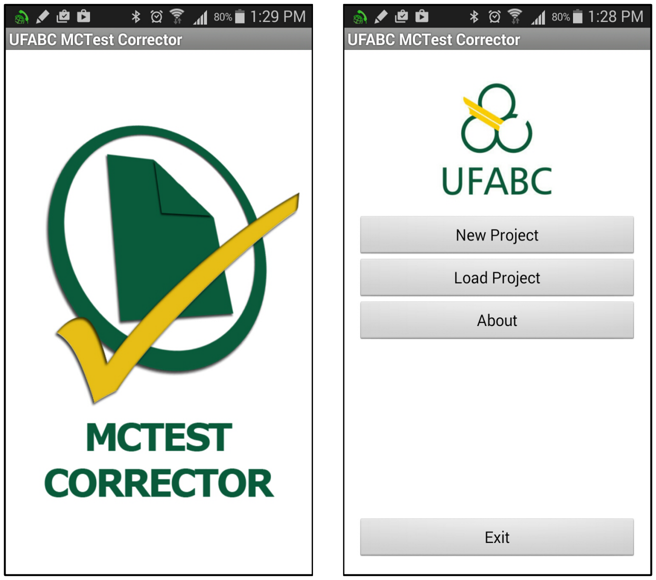
Figure 3. The red marks highlighted in the table indicate added value for weighted evaluation (bottom), different templates (right, top) and template identification (under).

In this Figure 3, the main form table is located in the centre and has 5x10 squares, but the complete answer form adds two lines (one above and one bellow) and two columns (one on the left and one on the right) to the form to hold table info. For instance, the corner squares are always painted to guide the block recognition. The right column is used to identify the test type, being the bits 2-5 of the column used to identify the number of the test type in binary format, 16 types in this model, and the next square to differentiate regular tests from a template (filled). This allows for different tests in a given project to have both questions and answers shuffled in order to prevent cheating, using different templates for grading each variation. The number of possible test types is given by the number of answer alternatives. For N alternatives for each question, the number of possible test types is $2^{(N-1)}$. In this example with 5 alternatives, there are $2^{5-1} = 16$ possible test types. The last line in the table allow for weighted evaluation, adding extra value for of specific questions with marked columns.

The first step in grading tests is to acquire the templates containing the correct answers for each type of test, e.g., in our case studies we used five squares for choices, therefore allowing 16 different templates used for grading. Having registered all templates, the user can start to grade tests randomly, the software being aware of what template to use. Trying to grade a test without having scanned the proper template will produce an error dialog.

The following figures (screen captures) illustrate the procedure for grading multiple-choice tests using MCTest. The device used for capturing the screens was a *Samsung Galaxy Note 2* smartphone running Android 4.4 (*KitKat*). Figure 4a shows the opening screen. Figure 4b shows the “Main Menu”, bringing the choices “New Project”, “Load Project”, “About” and “Exit” to finish execution (see workflow in Figure 2).

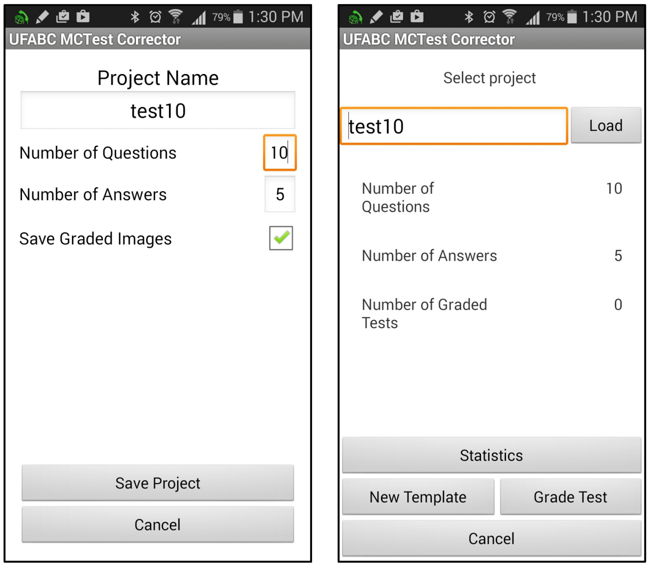
Figure 5a brings the “New Project” menu, where the user is required to name a project and specify the answer sheet size, for the number of questions and answers in the table. A checkbox gives the option to save the captured test images for later reference. After configuring the project the user is presented Figure 5b where it is possible to start capturing template sheets and test sheets. By clicking “Save Project”, the App creates a folder inside the application to store the project related files. Then the user is required to provide the templates for each test, using the option “New Template”, as shown in Figure 5b. Finally, having processed all the templates, the user can start grading tests using the option “Grade Test”.



(a) (b)

Figure 4. Welcome screen upon opening. b) App main menu.

In the capture interface, the user is require to position the table in the sheet alongside the guidelines in the delimited target area, using touches in the screen to focus the camera. The four corner black squares delimiting the table should remain outside the red ellipse and some extra white background added around the image to improve detection. After obtaining a satisfactory fit, capture an image by pressing the green button. Figure 6 shows the interface used to capture template sheets.



UFABC MCTest Corrector

Project Name

test10

Number of Questions 10

Number of Answers 5

Save Graded Images ☑

Save Project

Cancel

UFABC MCTest Corrector

Select project

test10 Load

Number of Questions 10

Number of Answers 5

Number of Graded Tests 0

Statistics

New Template

Grade Test

Cancel

(a) (b)

Figure 5. a) New Project and b) Load Project screens. After naming a new project and capturing at least one template sheet, "Grade Test" button is enabled.

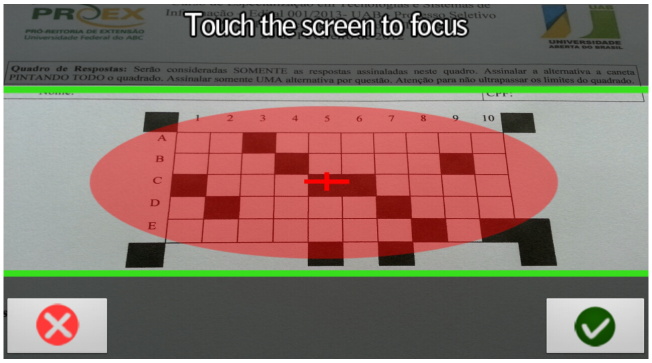
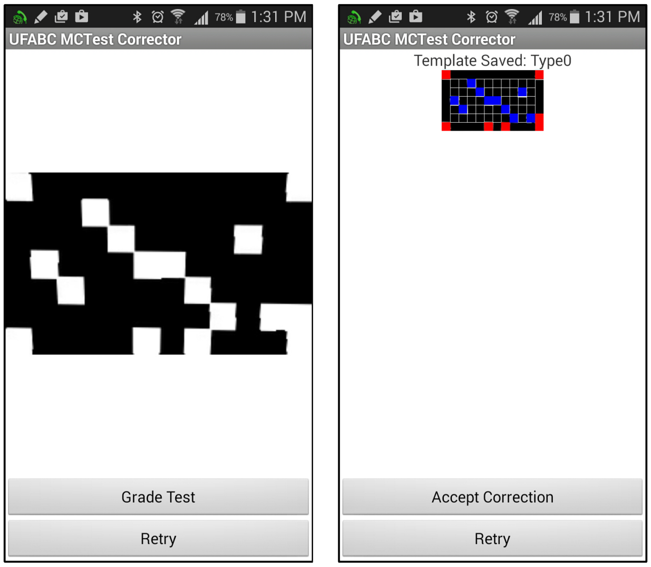


Figure 6. Capture interface. Test example with 10 questions.

The project website in (MCTest, 2015) contains several examples of test sheets in different formats for using with regular word processing applications. Figure 7 shows an example of captured template sheet before and after processing of the OMR (grading).

After proper detection, the option to "Grade Test" sends the image for processing, as shown in Figure 7b. If the capture was not successful, the user is given the option to "Retry". By clicking on "Accept Correction" a template sheet file is generated in the project folder.

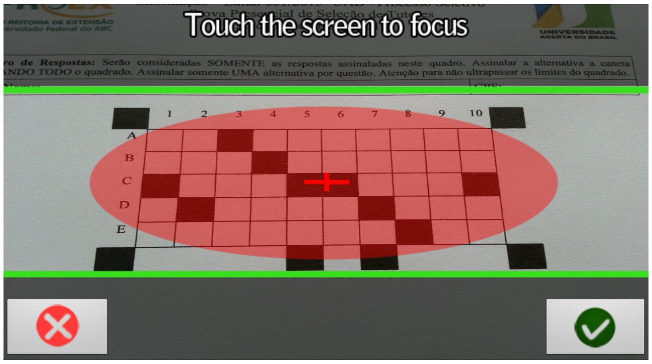


Template Saved: Type0

(a) (b)

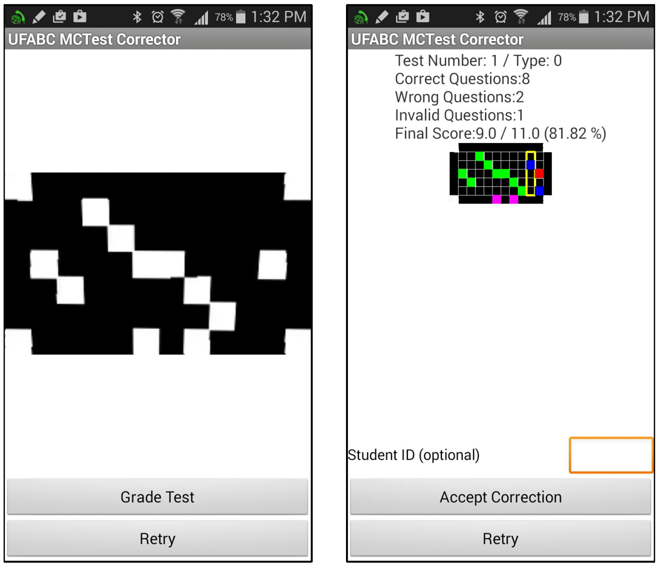
Figure 7. Captured image of a template sheet. a) before the grading. b) After sent for grading, the options are given to accept or retry the detection.

After capturing one or more template sheets, capturing and grading a test follows the same procedure (notice the bottom-left mark distinguishing between template sheets and test sheets). Figure 9 shows the correction and grading of the captured test. When the detection is satisfactory, the user can "Retry" or choose "Accept Correction" to add the grade to the spreadsheet.



Touch the screen to focus

Figure 8. Capturing a test with the capture interface.



Test Number: 1 / Type: 0
Correct Questions:8
Wrong Questions:2
Invalid Questions:1
Final Score:9.0 / 11.0 (81.82 %)

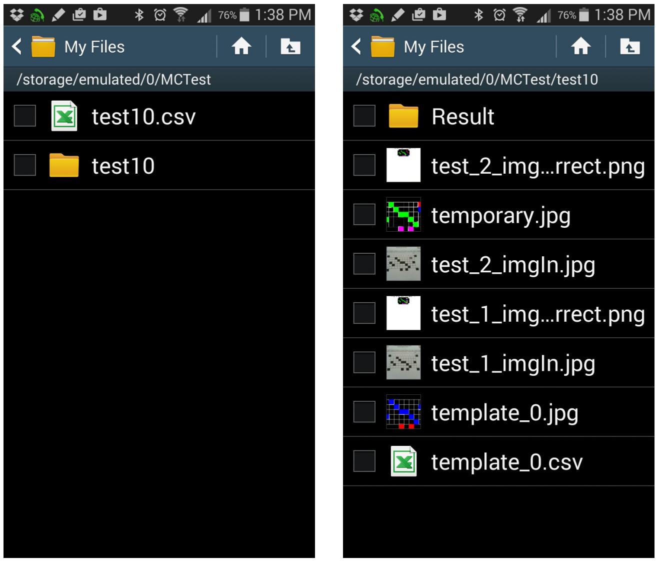
(a) (b)

Figure 9. Captured test sheet a) before grading. Selecting the option “Grade Test” generates the correction shown in b).

The corresponding template sheet is loaded and used to compare the data extracted from the two images, generating an image showing coloured blocks for correct answers (green), wrong answers (red) and highlighting columns left blank or with two or more answers for the question (yellow rectangles).

The program creates a spreadsheet to store the results sequentially, allowing the evaluator to perform test analysis and generate statistics later on. By clicking “Accept Correction”, the user adds an entry containing the test results to a spreadsheet named “ProjectName.csv”, a text file in “Comma Separated Values” (CSV) format. To retrieve the file, the user can send it from the device (e-mail or similar) of physically connect the mobile device to a PC to access the project folder. Figure 10 and Figure 11 shows the contents of the project folder and the spreadsheet file.

In addition, if the choice “Save Graded Images” (Figure 5a) is set, the program also saves the images as in Figure 10b to the project folder.



76% 1:38 PM

My Files

/storage/emulated/0/MCTest

- test10.csv
- test10

76% 1:38 PM

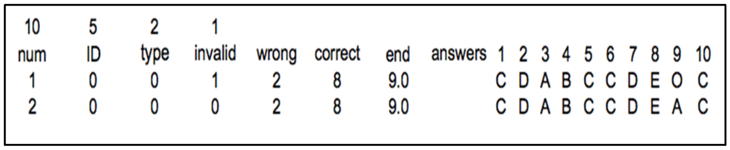
My Files

/storage/emulated/0/MCTest/test10

- Result
- test_2_img...rrect.png
- temporary.jpg
- test_2_imgIn.jpg
- test_1_img...rrect.png
- test_1_imgIn.jpg
- template_0.jpg
- template_0.csv

(a) (b)

Figure 10. a) Image folder for project "test10" and the spreadsheet containing results; b) images corresponding to the template sheets and tests graded.



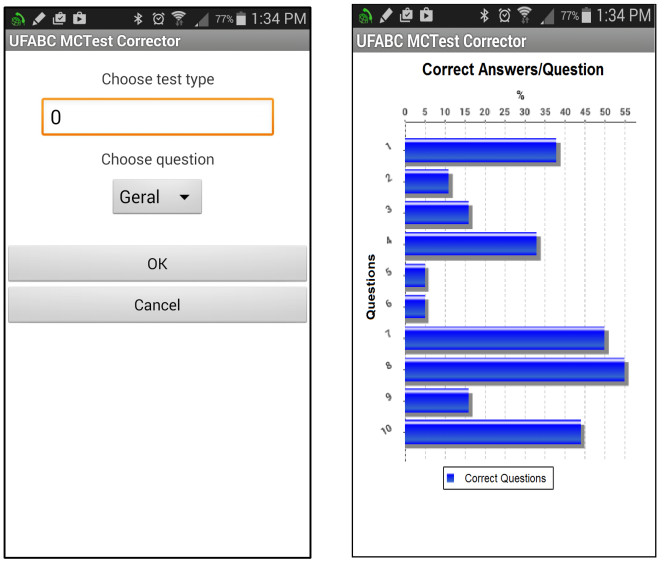
10	5	2	1														
num	ID	type	invalid	wrong	correct	end	answers	1	2	3	4	5	6	7	8	9	10
1	0	0	1	2	8	9.0		C	D	A	B	C	C	D	E	O	C
2	0	0	0	2	8	9.0		C	D	A	B	C	C	D	E	A	C

Figure 11. Contents of "test10.csv", the spreadsheet with results.

Performance statistics allow one to analyse test results after grading all tests, detecting weaknesses and improving the contents of teaching material. The option "Statistics" (in Figure 5b) opens the screen shown in Figure 12, where analysis can be generated for each test type, with "Choose test type", for each individual question or for all questions (from the pull down menu "Choose question", choosing "General"). After choosing test type and questions (or general), pressing "OK" MCTest will present a graph as in Figure 12b.

B. Image processing

The image captured with the device camera is a colour image with 24 bits/pixel. The resolution may vary depending on the camera specifications on each particular device. For the sake of compatibility, the raw image (Figure 13, upper panel) is cropped (Figure 13, lower panel) and has its colour map reduced to binary, where 0 is black an 1 is white) after inversion of the values for better visualization. This also saves space in the limited device’s storage. These steps are explained in detail in section D “Algorithms”.


UFABC MCTest Corrector

Choose test type: 0

Choose question: Geral

(a) (b)

Figure 12. a) Statistics for a test of type 0, considering all questions (general). b) Statistics relative to the number of correct answers.

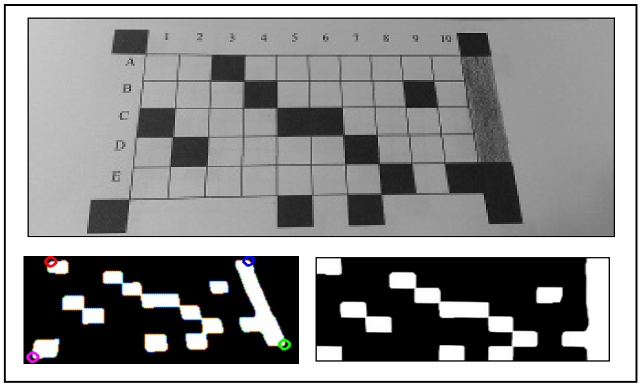


Figure 13. Example of capture and perspective correction of a template sheet.

In fact, once acquired (Figure 13), the image passes through a series of filters and transformations (GONZALEZ; WOODS, 2010) in order to improve quality while removing noises and reducing the effect of shadows. Some of the filters and transformations used are listed:

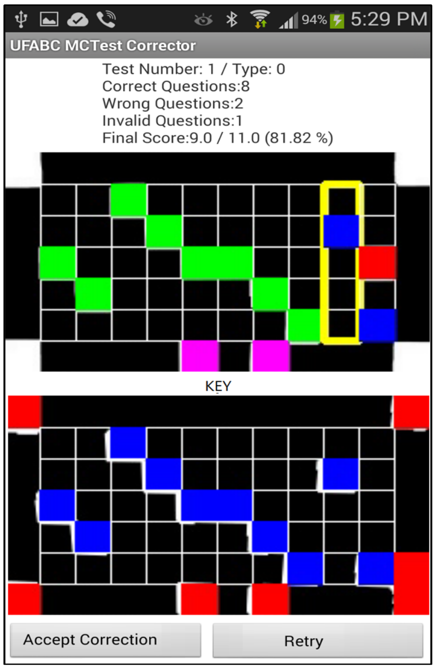
- Decrease resolution and colour depth to greyscale;
- Gaussian blur: smoothens the image to reduce artefacts;
- Adaptive threshold: adjusts the histogram and normalizes the light and dark areas in order to reduce the effect of shadows;
- Negative image: invert black and white;
- Morphological opening: consists of an erosion followed by dilatation, to further eliminate white dots or noises;
- Morphological closing: a dilatation followed by erosion, similar to the previous one, but to eliminate black dots;
- Distance transformation: Using the position of each corner, projected into an ellipsoid, it is possible to obtain the angle information (Figure 13, bottom left);
- Perspective transformation (WarpTransform) (OPENCV, 2015; GONZALEZ; WOODS, 2010): based on the information obtained in the previous step, adjusts the viewing angle of the image correcting the perspective (Figure 13, bottom right).

After successful pre-processing, the result should be a binary image showing only marks in white, as presented in Figure 13, bottom right image. The image is ready for translation into a table and subsequent comparison with the appropriate template or, in case of a template, create a new table for comparison.

C. Optical Mark Recognition

The final step is to compare the information presented in the image with the appropriate template, or to create a new template, in case the template identifier bit is marked (at the bottom of the right column, see Figure 3).

Upon comparison with the appropriate test template, the App calculates the score and displays the results along with information on each question. In the upper panel of Figure 14, green marks denote right answers, red denotes wrong placed marks and a yellow rectangle denotes a column where a mark is absent, undetected or with more than one choice. Notice that some questions with a mark at the bottom line have 50% extra value (1.5) in this test, whilst the remaining questions are rated 1 point. It is possible to adjust the extra value in "Project Configuration". The information atop Figure 14 identifies test number (ordinary) and type (template number used), number of correct and wrong answers, the final score in points and percentage. The lower table in Figure 14 is the image of the template used for comparison. If the user accepts the correction, the program records the data into the spreadsheet "ProjectName.csv" as previously described.



Test Number: 1 / Type: 0
Correct Questions:8
Wrong Questions:2
Invalid Questions:1
Final Score:9.0 / 11.0 (81.82 %)

Figure 14. Evaluation example, with test results displayed on top.

D. Algorithms

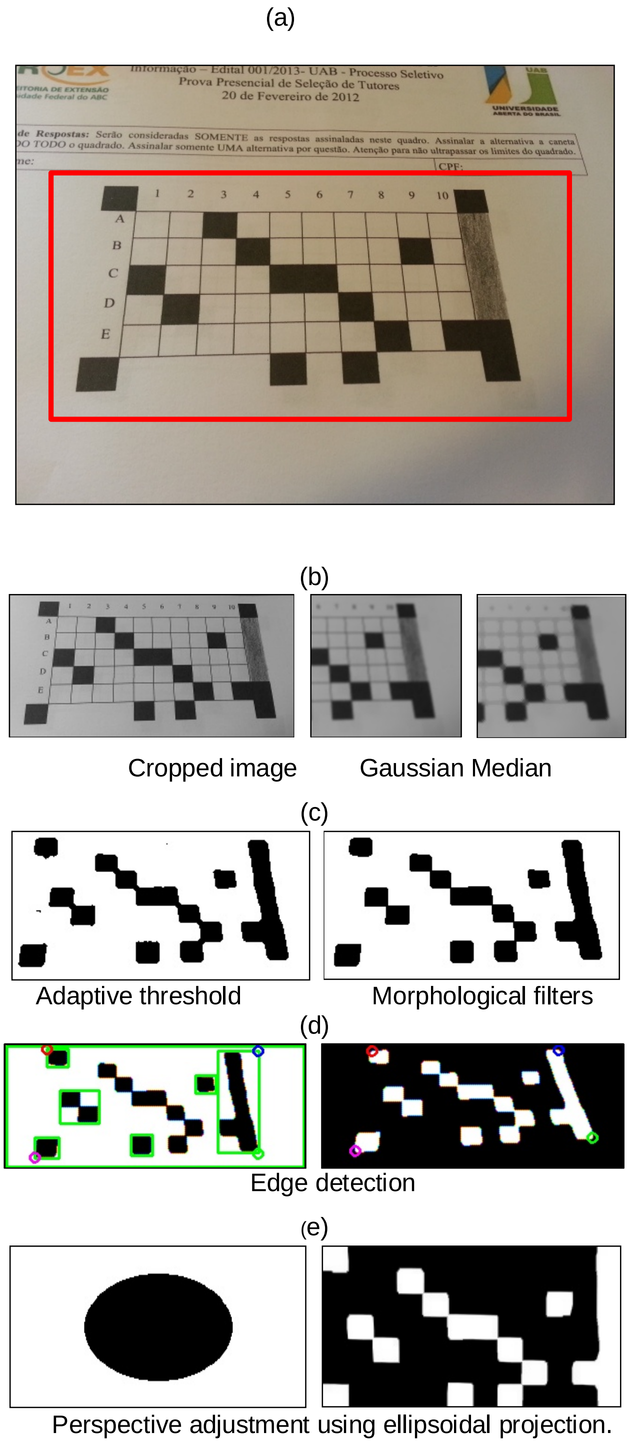
Figure 15 briefly emphasizes some steps taken to process the information from the captured camera image (a) to the final binary bitmap used to extract the answer table (e). After cropping (b, left) the original image (a) to fit inside the green guidelines (as in Figure 6 and Figure 8) it has its colour depth reduced to greyscale. Independent of the camera resolution, the image is resized to a fixed, manageable dimension. Both steps help to address compatibility issues across multiple devices and camera configurations. Then, the image is successively subjected to a Gaussian Softening (b, centre) and a Median Filter (b, right) to eliminate high frequency noise or salt and pepper noise. To reduce the impact of shadows, an adaptive threshold (c, left) is applied, followed by a combination of morphological filters (c, right) to eliminate more noises. After these steps, the image is ready for segmentation (d and e). All these processes in image segmentation are detailed in (ZAMPIROLLI, et al., 2015).

Discussions and Results

A. New Tools for Education

This work expands the idea that smartphones can boost productivity when employed in Education, not only for students, but specially for teachers (GRAHAM, 2015). In fact, the convenience of easily accessing the digital world has shifted the emphasis to the ability of finding information, rather than memorizing it. On the other hand, the increase in the amount of information available everyday makes the kind of knowledge acquired by most of the youths increasingly superficial.

When properly used, smartphones and gadgets in general can facilitate learning and teaching (DEB, et al, 2014; JEFFREYS, 2015). For instance, it is possible to teach mathematics and physics by means of games. Some important aspects of smartphones as tools for education are:


(a)
Informação – Edital 001/2013- UAB - Processo Seletivo
Prova Presencial de Seleção de Tutores
20 de Fevereiro de 2012
ITORIA DE EXTENSÃO
idade Federal do ABC
UAB
UNIVERSIDADE ABERTA DO BRASIL
de Respostas: Serão consideradas SOMENTE as respostas assinaladas neste quadro. Assinalar a alternativa a caneta
DO TODO o quadrado. Assinalar somente UMA alternativa por questão. Atenção para não ultrapassar os limites do quadrado.
me:
CPF:
1
2
3
4
5
6
7
8
9
10
A
B
C
D
E
(b)
(c)
(d)
(e)

Cropped image Gaussian Median

Adaptive threshold Morphological filters

Edge detection

Perspective adjustment using ellipsoidal projection.

Figure 15. a) Original image ("Temp"), as captured by the device camera, with the cropped section highlighted in red (ROI). b) The image is resampled, colour depth reduced to greyscale and filtered. c) The binary image as result of an adaptive threshold and a combination of morphological filters to eliminate noise. d) Finally, the perspective is adjusted using e) distance transformation in an ellipsoidal projection.

- They offer the possibility of mobile learning, i.e., one can get information everywhere, anytime, answer tests, upload homework, communicate with other students, send and answer questions, read emails, listen to podcasts, classes, etc.;
- Teachers can rapidly give feedback to their students about their grades;
- Enhance learning by interacting with objects, such as taking pictures of places, animals, zoom in small insects and use this material to document lessons learned;
- As a learning tool, the smartphone can make education cheaper, as a single teacher can reach a large number of students;
- They can make learning funny and a more familiar activity;
- As a grading tool for teachers, as described in this work.

On the other hand, there are also studies suggesting that the wrong use of gadgets, particularly in excess, may compromise the capacity of concentration, hinder intellectual development and even make some people dependent on technology (STRANG, 2014).

B. Preliminary tests

Before applying a test to a particularly large number of students, we recommended printing an answer sheet and some tests to evaluate its accuracy, given the parameters of shape and size of the built table. It is also a good idea to test the mobile device one is planning to use for grading.

Test models are available for download from the project site and can be used as a reference or modified for use in a particular scenario (MCTest, 2015).

C. Filling the answer sheet

The experiments indicate that malformed markings (blurred, not properly filled, e.g. just an X), leaving much of the background undetected, results in lower grades than expected. This requires the intervention of the grader to do a proper marking or to increase the grade on a need basis. To spot possible misdetection, the App shows a yellow rectangle around every question left blank, improperly filled or containing more than one marked square (invalid).

D. Accuracy

We made available as examples two tests graded with MCTest (2015) (under the names classA2 and classB2), for 41 and 64 students, respectively. The applied tests of 4 different types (variations) all have 16 questions, each containing 5 possible answers. In the first example, student number 19 wrongly filled some squares with an X instead of completely painting the square. To resolve this sort of problem, it is necessary to fill in the square with a pencil and repeat the correction (shown in correction n. 42 in subfolder Experiments / classA2 in (MCTest, 2015)). The same problem occurred in the second example (classB2) with student 49. The names and identifiers of the students were blurred to protect their identity. In both case studies, performed in real scenarios, MCTest had 100% accuracy when students correctly filled in the answer sheets. MCTest considers blocks “filled” if they contain at least 50% of the area painted. That is why it is important to include instructions in the test, explaining how to fill the answer table completely, using black pen or pencil, to avoid lower grades.

E. Include a white buffer

It is important to leave a blank area around the table (particularly in between the green detection lines in Figures 6 and 8) to avoid additional noise on the area surrounding the ROI (Region of Interest).

F. Lighting and distortions

Dim light might affect detection, resulting in misdetections due to shadows and camera artefacts. Choose an environment with uniform ambient light and preferably no reflections in the sheet. It is also important to capture the test in a plane surface, avoiding unnecessary distortions in the image.

G. Performance

Another important factor when grading a large number of tests is the processing time for each test. A preliminary version (in MATLAB) could return a result instantaneously, but required a predefined stable structure in place for capturing images (ZAMPIROLLI; QUILICI-GONZALEZ; NEVES, 2013). As expected, the processing time in mobile devices depend on the processing capabilities of each device. Yet, under conventional situations, it was still preferable to use mobile devices (even cheaper models) when compared to the PC-based solution. The added mobility and easiness to position the camera granted lower “per-test” processing times, varying from 3 (Motorola XT918) to 7 seconds depending on the device used. For instance, a Samsung Galaxy Note 4 takes less than one second from the capture of the test image to displaying the result. In other words, it is processing the result in real time.

H. Future works

We are currently working on an iOS (Apple iPhone/iPad) version of the App. A Windows Phone version is under consideration.

A server based correction engine and the archiving of test results are under study. Both will allow the use of protocols, such as FTP and VPN, to assist a team in order to split the efforts of grading large quantities of tests in a project. In this model, the mobile phone would send the image to the server to be graded and archived, adding the results to a shared spreadsheet. Parts of the code already exist to accomplish this task. In the server-based model, implemented using Python and OPENCV (2015) secure connections are established using access permissions to increase privacy and security.

Another study aims to integrate grading data with grade management systems, commonly used in schools and universities, taking special care about security issues involved in authoring authenticating and verifying tests and grades.

Conclusion

The image segmentation technique used in this work, known as Mathematical Morphology, demonstrated how limitations resulting from images captured under non-ideal conditions can be circumvented, giving even better results than experiments conducted under controlled conditions.

Despite the fact that the Operation System of choice was Android, development for any other platform in existence is possible. Given that OpenCV is available for most platforms nowadays, it would be relatively simple to port this solution to any another OS, mobile or desktop.

The trials performed in dozens of tests have shown that MCTest presents an accuracy rate of up to a 100%, even when considering user-related errors, such as poorly filled tests. Improvements are currently underway to enhance recognition rates in the worst case scenarios (bad lighting, poorly filled forms, poor quality cameras).

Acknowledgments

The authors would like to thank the *Universidade Federal do ABC* and also the financial support through the scientific initiation program PIBIC/CNPq.

References

ANDROID, Available in http://developer.android.com/sdk. Access in 22.02.15.

CHINNASARN, K.; RANGSANSERI, Y. An image-processing oriented optical mark reader. Conference Series, Society of Photo-Optical Instrumentation Engineers (**SPIE**) Conference Series, vol. 3808, pp. 702–708, 1999.

CIENCIAMAO. Available in www.cienciamao.usp.br. Access in 10.04.15.

DEB, D.; et al. Developing interactive classroom exercises for use with mobile devices to enhance class engagement and problem-solving skills. Frontiers in Education Conference (**FIE**). IEEE, 2014.

FISTEUS, J.A.; et al. Grading Multiple Choice Exams with Low-Cost and Portable Computer-Vision Techniques. **Journal Of Science Education And Technology**, vol. 22(4), pp. 560-571, 2013.

GONZALEZ, R.C.; WOODS, R. **Processamento Digital de Imagens**. São Paulo: Pearson Pentice Hall, 2010.

GORDILLO, A.; et al. Enhancing web-based learning resources with quizzes through an authoring tool and an audience response system. Frontiers in Education Conference (**FIE**). IEEE, 2014.

GRAHAM, E. Using smartphones in classroom. National Education Association. Washington, DC, USA. Available in http://www.nea.org/tools/56274.htm. Access in 29.11.2015.

JEFFREYS, B. Can a smartphone be a tool for learning? Available in http://www.bbc.com/news/education-34389063. Access in 18.11.2015.

GREENFIELD, Elizabeth. OMR and Testing Software: Saving Time and Getting Results. **THE Journal (Technological Horizons In Education)**, v. 20, n. 8, p. 12, 1993.

LUCENA, K.K.T.; et al. MobiMonitor: A mobile app for monitoring distance courses in the Amazon region. Frontiers in Education Conference (**FIE**). IEEE, 2014.

MCTest. Available in http://vision.ufabc.edu.br/MCTest. Access in 24.05.15.

MOLLAH, A.F.; et al. Design of an Optical Character Recognition System for Camera-based Handheld Devices. **International Journal of Computer Science Issues** 8(1), 2011.

NGUYEN, T.D.; et al. Efficient and reliable camera based multiple-choice test grading system**. International Conference on Advanced Technologies for Communications**, 2011.

OPENCV. Open Source Computer Vision. Available in http://opencv.org - Access in 08.04.15.

RODRIGUES, J.R.A.; et al. iQuiz: integrated assessment environment to improve Moodle Quiz. Frontiers in Education Conference (**FIE**), IEEE, 2013.

SAIPULLAH, K.M.; et al. Measuring power consumption for image processing on Android smartphone. **American Journal of Applied Sciences**, vol. 9(12), pp. 2052-2057, 2012.

SEN, D.J.; et al. Modern Database Technology Needs Optical Mark Reading. **International Journal of Pharmaceutical and Applied Sciences** 1(2), 56, 2010.

SPADACCINI, A. A Multiple-Choice Test Recognition System based on the Gamera Framework. **Document Image Analysis with the Gamera Framework**, vol. 8, pp. 5-15, 2009.

STRANG, T. Smartphone in class: Learning tool or distraction? Engagement and Motivation, Student Motivation. Cengage Learning. Available in http://blog.cengage.com/smartphones-in-class-learning-tool-distraction/. Access in 10.09.2015.

ZAMPIROLLI, F.A.; QUILICI-GONZALEZ, J.A.; NEVES, R. Automatic Correction of Multiple-Choice Tests using Digital Cameras and Image Processing**.** Workshop de Visão Computacional (**WVC**), Rio de Janeiro, 2013.

ZAMPIROLLI, F.A.; CHINA, R. T.; NEVES, R.; QUILICI-GONZALEZ, J.A. Automatic Correction of Multiple-Choice Tests on Android Devices**.** Workshop de Visão Computacional (**WVC**), São Carlos, 2015.