Arias Fisteus, Jesús; Pardo, Abelardo; Fernández García, Norberto (2013). Grading Multiple Choice Exams with Low-Cost and Portable Computer-Vision Techniques. *Journal of Science Education and Technology*, 22(4), pp.: 560-571.

DOI: https://doi.org/10.1007/s10956-012-9414-8

Journal of Science Education and Technology manuscript No.
(will be inserted by the editor)

Grading multiple choice exams with low-cost and portable computer-vision techniques

Jesus Arias Fisteus · Abelardo Pardo · Norberto Fernández García

Received: date / Accepted: date

Abstract Although technology for automatic grading of multiple choice exams has existed for several decades, it is not yet as widely available or affordable as it should be. The main reasons preventing this adoption are the cost and the complexity of the set-up procedures. In this article, *Eyegrade*, a system for automatic grading of multiple choice exams is presented. Whilst most current solutions are based on expensive scanners, *Eyegrade* offers a truly low-cost solution requiring only a regular off-the-shelf webcam. Additionally, *Eyegrade* performs both mark recognition as well as optical character recognition (OCR) of hand-written student identification numbers, which avoids the use of bubbles in the answer sheet. When compared to similar webcam-based systems, the user interface in *Eyegrade* has been designed to provide a more efficient and error-free data collection procedure. The tool has been validated with a set of experiments that show the ease of use (both set-up and operation), the reduction in grading time, and an increase in the reliability of the results when compared with conventional, more expensive systems.

Keywords Automatic assessment · computer assisted assessement · automatic image recognition · computer supported learning

1 Introduction

The use of technology is now present in numerous aspects of any learning experience. Assessment has been one of these aspects where technological solutions were first considered. As early as 1965, a system by which students submitted homework answers through punched cards that were automatically processed is described (Forsythe and Wirth, 1965). Nowadays, the variety of tools to support assessment is enormous. But this adoption is uneven when dividing assessment into formative (informal, providing feedback for teacher and student with no academic effect) and summative (oriented towards obtaining a measure of the learning process).

J. Arias Fisteus, A. Pardo (Corresponding author. Phone: +34 91 624 5940, Fax: +34 91 624 8749), and N. Fernández García
Department of Telematic Engineering, University Carlos III of Madrid, Avenida Universidad 30, 28911, Leganés (Madrid), Spain, E-mail: {jaf, abel, berto}@it.uc3m.es

Multiple automatic formative assessment solutions are widely available and frequently used within learning management systems. Students may take tests derived from large pools of questions which are then automatically graded and feedback is returned (Karavirta et al, 2006). Clearly, the time to create, grade and provide feedback is greatly reduced with these systems (Twigg, 2003). More sophisticated tools such as intelligent tutoring systems or adaptive hypermedia systems are also used to supervise the learning process in a specific context and provide students with the appropriate feedback to increase the learning effectiveness (Verdú et al, 2008).

But this level of automation decreases significantly in the context of summative assessment, and remains even lower in face to face learning environments. For example, final exams for high enrollment courses are typically scheduled by institutions in a fixed time, date and location. The high number of students makes the use of computer-supported assessment unfeasible. In these scenarios, pencil and paper are still being widely used. The main reasons behind this difference is that computer-supported exams do not scale when performed synchronously and they pose special security requirements (Apampa et al, 2009). Pencil and paper exams still are widely used in educational institutions for partial or final exams. The production of a physical document where students reflect their answers that are then graded to obtain a score is the essential aspect of this type of assessment when deployed in large clases. Besides, although the use of computer-based tests translates into a significant cost-reduction, it may have undesired effects on the students. Although active learning methodologies promote student participation and continuous evaluation of student performance, written tests are still present even in this context. A number of studies about the use of computers for assessment is presented by Norris et al (2007), concluding that the issue of equivalence between computer-based and paper-and-pencil assessments has not been conclusively solved. There seems to be some evidence that factors such as computer familiarity, attitudes toward computers, or computer anxiety have a negative effect on students when taking computerized tests. Although the presence of this negative impact in computer tests is still unclear, the possibility of a highly automated grading procedure for paper-and-pencil assessments seems an adequate trade-off to consider.

Automation of pencil and paper exams is restricted to multiple choice questions (MCQ) or, in general, questions with answers encoded in a so called “answer sheet”. These sheets are then processed and compared with a correct sample. Finally, a grading scheme is applied to obtain the final score. Optical Mark Recognition (henceforth simply OMR) tools are currently used to automate the grading for a large number of exams. However, the cost of the required equipment (a specific scanner) and the set-up time for these applications (access to the equipment and scan preparation) restricts its use to mainly high enrollment courses, where the time and cost reductions are significant (Kubo et al, 2004).

Early OMR systems required especially designed scanners and forms and were only available to institutions with a sizable budget. The optical recognition imposed severe restrictions on the type of paper, the color of the ink, or even the layout of the answer sheet. But current technologies favor the appearance of “low-cost” OMR solutions and their presence has increased in learning environments. Today users may print their own forms and use software tools to process the results.

Commercial tools such as Remark Office OMR Software[1] offer a solution covering the management of question pools, exam and answer sheet creation and the subsequent answer processing. Similar solutions are available as open-source tools such as QueXF (Zammit, 2009). Deng et al (2008) and Saengtongsrikamon et al (2009) show how a low-cost OMR

[1] http://www.gravic.com/remark/officeomr/ (accessed 10-1-2011)

tool can be used in a conventional educational scenario. Although requiring some adjustments, especially in the scanning phase, these tools can now be easily integrated with conventional Learning Management Systems. A module to support "off-line assessments" has been created for the Moodle Learning Management System (Rane et al, 2009). This module integrates question bank management, grading sheet creation and the processing of the scanned answers. A system for OMR named MarkSense is presented by Winters and Payne (2005). It uses computer vision techniques and the authors report a high level of adoption in their institution. Unfortunately, it is presented as part of a larger system, and the authors detail neither the capabilities of the system nor its technical implementation.

The majority of solutions rely on the use of a scanner to obtain a computer representation of the answers to be further processed by software tools. Although a regular scanner can be used for this task, the presence of these devices is not as ubiquitous as desired. In this paper an OMR-based solution for summative assessment based on computer-vision technology, named *Eyegrade*, is presented[2].

The tool with a functionality similar to the one proposed by *Eyegrade* is *GradeCam*[3]. *GradeCam* is a commercial OMR tool also based on a webcam. Although the approach is similar to the one presented in this paper, *Eyegrade* offers several improvements: arbitrary number of answers in a sheet, possibility for the students to change the answer to questions despite using non-erasable pens, no specialized hardware is required, recognition of hand-written student identifiers, and most importantly, an interface that seamlessly blends the functionality of supervising the data captured by the system. An in-depth comparison between these two tools is included in Section 4.

More precisely, the advantages of *Eyegrade* with respect to existing tools are:

- Portability. As opposed to current systems requiring a scanner, the system can be used where a webcam is present. In addition, webcams can be easily transported due to their compact size and reduced weight.
- Low-cost. Webcams are significantly less expensive than scanners. Furthermore, when used in OMR settings, scanners require automatic document feeders that increase their cost. Webcams, on the other hand, are commonly found on mid-level laptop computers and require no special enhancement to be used in the proposed tool.
- Speed. Scanning a single page in a low-end scanner takes a considerable amount of time, in the order of 20 to 30 seconds per page. Regular web cameras can capture data at greater speeds. There exist faster scanners equipped with automatic page feeding, with scanning times per page below 10 seconds or, the most expensive models, below 2 seconds. However, the price (in the order of ten to one hundred times the price of a regular webcam) clearly favors the approach proposed in this work.

Due to these features, the solution presented in this paper can be used in some scenarios in which conventional OMR systems are unfeasible. For example, medium and small institutions often lack of resources to acquire them. This is the case, for instance, of many secondary schools, in which the equipment needed for using Eyegrade is more likely to be within the immediate reach of teachers. A second scenario in which Eyegrade offers a convenient solution appears when exams need to be shipped to another location in order to be graded (for example, when an instructor is visiting other institutions, or when an institution has several sites but the OMR system is not available at some of them). In this case, the use

[2] Available at http://www.it.uc3m.es/jaf/eyegrade

[3] http://www.gradecam.com (accessed 10-1-2011)

of Eyegrade avoids the cost and inconveniences of shipping, as well as the delay it would introduce in the process of grading the exams.

The main advantage of using scanners instead of webcams for OMR is their superior resolution and quality of image. However, the experiments documented in this paper with *Eyegrade* show that the resolution and image quality of a regular webcam do not limit its application in a real academic context up to a reasonable number of questions per answer sheet.

The proposed tool allows an exam and its corresponding answer sheet to be easily created, and the results to be captured, reliably checked and recorded in a reduced amount of time. Hardware requirements are simply a normal off-the-shelf webcam and a computer.

The rest of the paper is organized as follows. Section 2 describes the technical details of the proposed system, mainly the image processing procedures. Section 3 describes the validation experiments that were carried out in real-life courses. A discussion of the obtained results is included in Section 4, and the conclusions as well as some future lines of research are outlined in Section 5.

2 Material and Methods

The system is based on well-known computer vision techniques. The webcam captures a continuous stream of images, and the system looks for a properly framed answer sheet. Once it is detected, its marks (in this case, student's answers and ID number) are extracted from the image, stored and shown to the instructor, who can review and correct them when necessary. This section explains the process in depth, as well as its technical details.

2.1 System overview

After instructors have selected a set of questions for an exam, several exam versions are created by shuffling both questions and answers within the questions. Each of these shuffled version of the exam will be called a *model* for the remainder of the paper. An answer sheet has to be produced along with the shuffled questions. Figure 1 shows an example of the part of the answer sheet that is subject to image recognition. The *Eyegrade* system automates these tasks by using the LaTeX document preparation system. Instructors, though, are free to use other environments to produce answer sheets as long as they have a similar format. As opposed to other OMR systems, answer sheets may be printed or photocopied on regular white or recycled paper.

Students write down their personal data (not shown in the figure), their student ID number and mark with a cross their answers in the appropriate cells. A student may choose to clear a cross by filling the cell entirely with ink. The system ignores these cells allowing students to change a question to "un-answered".

After the exam has taken place, instructors slide the answer sheets under the camera one at a time. For each sheet, the program displays an image capture augmented with information about the detected answers, the number of correct and incorrect answers, and the student's ID number. Figure 2 shows an example of a capture and Figure 3 shows the same capture augmented by *Eyegrade*.

If the system incorrectly detects an answer (either a false positive or false negative), the user interface allows the instructor to correct it by simply clicking in the appropriate cell.

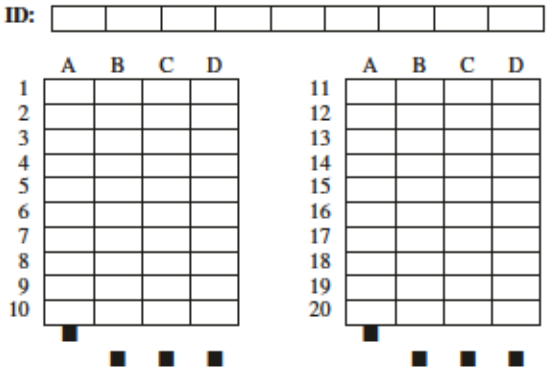

Fig. 1 Sample answer sheet for 20 questions with four alternatives per question. Black squares at the bottom are used to encode the exam model.

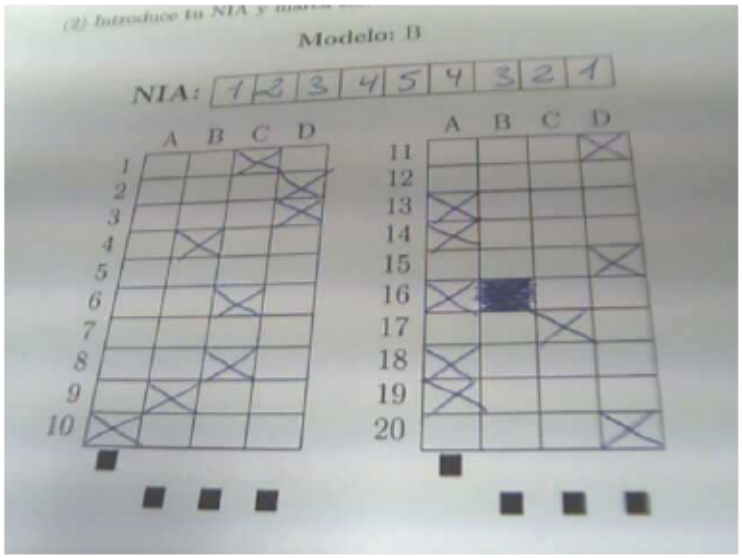

Fig. 2 Example capture of an answer sheet.

When the system renders the ID number incorrectly, the user interface allows the instructor to correct it with a few keystrokes. Three different mechanisms are provided for this correction:

- with a single keystroke, the instructor can walk through the list of student IDs in the class, ordered by their probability to represent the handwritten number (when a detection error occurs, the correct number is often the second or third in the list);
- by typing a sequence of digits (not necessarily the complete ID number) the system shows the ID in the given list containing that sequence with the highest probability;
- when the list of IDs is not available, or the student is not in it, by typing the complete ID number.

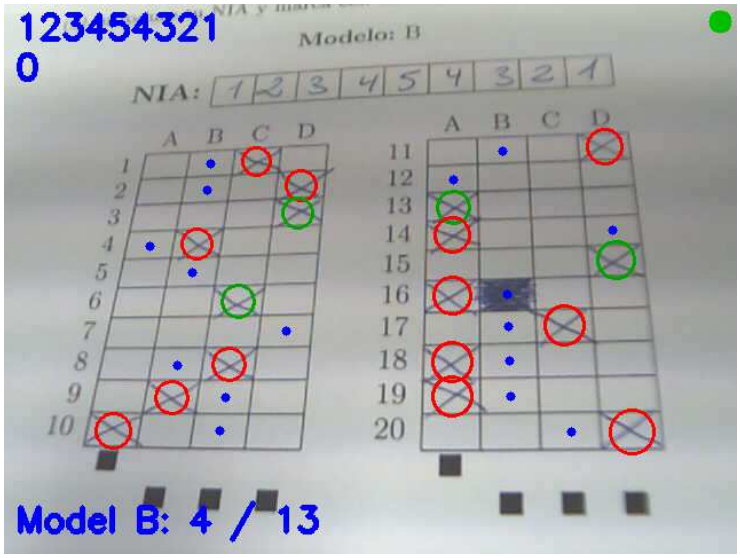

Fig. 3 Capture of Figure 2 with augmented information. Correct answers are marked with a green circle. Wrong answers, with a red circle. For wrong or blank answers, the correct answer is marked with a small blue dot. Model and total number of correct and incorrect answers are shown at the bottom left. The student's ID number is at the top left. The exam sequence number (a sequential number the system assigns to each processed exam) is just below it.

Once the instructor confirms that the captured data are correct, the system stores the augmented image capture along with a row in a formatted data file with several fields: student ID, exam model, number of correct and incorrect answers, and the answers for every question.

The data file produced by the tool is in Comma Separated Values (CSV) format and can be easily imported into the system used to store and manage scores (for example, a spreadsheet, or the Learning Management System used at the institution). This data file can also be used to produce detailed statistics for questions, groups of students, etc.

The augmented image capture can be useful in several ways. For example, it can be automatically emailed to students in order to let them review their score and the answers they failed. It can also be used to review an exam when the instructor has no physical access to it (for example, if students complain about their score while the instructor is in a trip, working at home, etc.)

The video in figure 4 shows a demonstration of the system.

2.2 Technical description of the system

The design of the system was driven by the following criteria:

1. Precision in the detection is more important than detection time.
2. The system captures a continuous stream of images from the webcam. This contrasts with other systems (for example, those based on scanners) in which only one capture is available.
3. Users must be able to review and correct, if necessary, automatic decisions of the system.

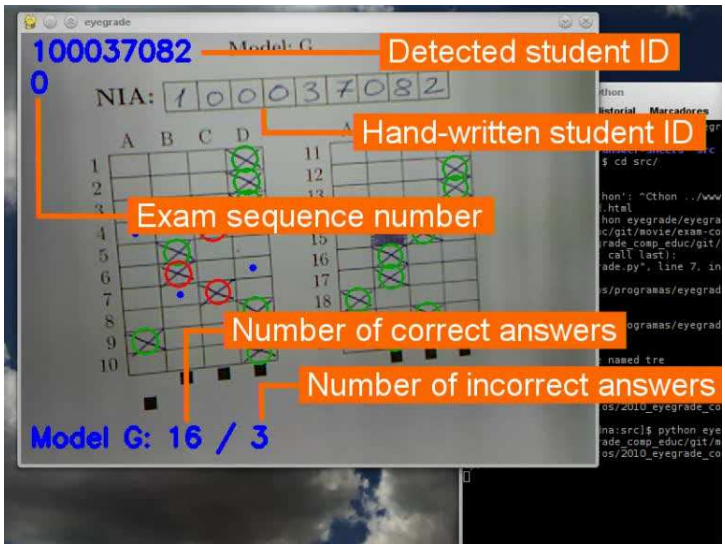

Fig. 4 Demonstration video of Eyegrade. *Note for referees: video attached to the submission.*

A consequence of combining assumptions 1 and 2 is that, if the marks in the answer sheet cannot be accurately detected in a given capture, the system can simply drop that capture and analyze the next one. This lowers the probability of incorrect detections, because only *good enough* captures are used.

Once an image is captured from the webcam, the system performs the following steps to capture the marks:

- Image pre-processing: the color image is transformed into a monochrome image. This step consists of an RGB to grayscale transformation followed by adaptive thresholding.
- Line detection: straight lines are detected by using the Hough transform (Duda and Hart, 1972).
- Answer table detection: the geometry of the table(s) in which answers are written is detected by intersecting the lines obtained in the previous step. The result of this step is the position of the corners of every cell in the answer tables.
- Decision making: the program analyzes the part of the image within each cell, and decides whether the cell has been marked or not.
- Model detection: the system supports the encoding of the exam model in the answer sheet. It is detected in the captured image.
- Student ID detection: the exam may include a field for students to write their ID number. The system applies OCR techniques to detect that number from handwritten digits.

The following sections describe each of these steps in detail.

2.2.1 Image pre-processing

The image processing algorithms used by the system work on one monochrome channel. This step prepares the image for those algorithms by first converting the image to a single grayscale channel and then applying an adaptive threshold algorithm, which transforms the grayscale channel to monochrome and, at the same time, reduces noise. Figure 5 shows the result of the threshold algorithm for the capture in Figure 2.

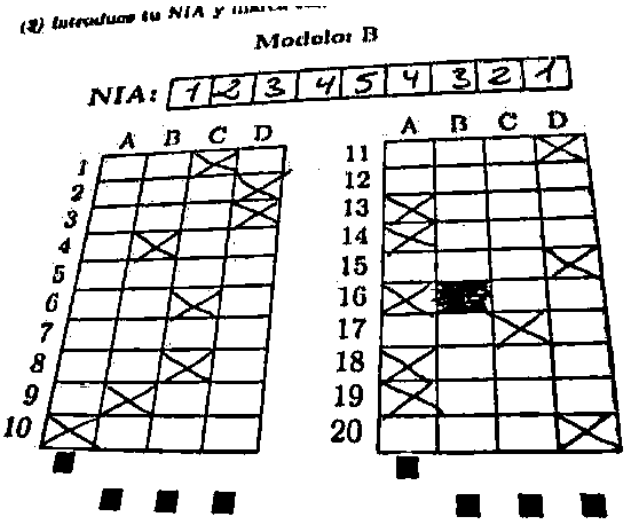

Fig. 5 Result of the adaptive threshold algorithm for an example capture.

2.2.2 Line and answer table detection

The Hough transform (Duda and Hart, 1972) is a widely used feature-detection technique. It was first designed for detecting straight lines, but later generalized for curves and other kinds of shapes.

The original straight line detection technique is applied to the pre-processed image in order to detect the straight lines that form the table in the answer sheet. For each detected line, the algorithm returns its direction and position, but not its bounds.

Once a set of lines has been identified in the captured image, the exact position of the answer tables and their cells have to be determined. The main difficulties that need to be solved are:

- Barrel distortion: depending on the optics of the camera and its relative position to the answer sheet, the captured image may suffer from a slight barrel distortion, which makes straight lines appear curved. In these cases, the Hough transform may detect more than one line for the same actual line, with small changes in direction.
- Perspective distortion: parts of the table that are closer to the camera are bigger. Tables are seen as trapeziums rather than rectangles in this case. As a consequence, horizontal and vertical lines may not be perpendicular in captures. Figure 2 is an example of perspective distortion.
- Undetected lines: some lines of the tables may have not been detected. This happens, for example, when a line is out of focus or intense light is reflected by a line.
- Lack of line bounds: as explained before, bounds of lines are not detected in the line detection step.
- Spurious lines: the set of detected lines may include other lines in the answer sheet not belonging to the table (for example, the baseline of some text lines, lines belonging to other boxes, oblique lines made from crosses written down by the student when they are aligned, etc.)

First, lines have to be classified according to whether they are horizontal or vertical. In order to do so, lines whose direction differ by 0.3 radians or less are grouped together. This tolerance in the angles allows a certain degree of perspective distortion. In normal conditions, two groups (horizontal and vertical) should result.

After that, barrel distortion effects have to be cancelled: lines that are very close together in terms of distance to the origin of coordinates and direction are collapsed in only one line, parameterized by their average distance to the origin and angle.

Then, the system checks whether the number of detected lines is in the expected range. Given the importance of precision, it is assumed that the geometry of the answer tables (number of tables, and number of rows and columns in each table) is known *a priori* (it is specified in an exam-specific configuration file).

Once the number of lines has been checked to be correct and the lines that delimit the tables are identified, the system computes the corners of every cell in those tables by intersecting horizontal lines with vertical lines.

Finally, distances between consecutive cell corners in a given horizontal line are checked to follow a regular sequence. If not, the capture is discarded because detection of some lines in the table may have failed. This check was added to the algorithm because our first experiments showed that, while vertical lines are generally detected with high accuracy, detection of horizontal lines in the upper rows of the answer tables may not be reliable due to the combination of perspective and barrel distortion.

2.2.3 Decision making

Once the corners of every cell in the answer tables have been detected, the system has to decide, for each cell, whether it has been marked or not.

The system expects cells to be marked with a cross sign (two lines joining opposite corners of the cell). In order to detect these marks, the image is masked by a thick cross placed where the mark is expected. The percentage of marked pixels behind the mask is computed. If it is higher than a given threshold, the system decides that the cell is marked.

Sometimes students need to *clear* an answer that they had already marked on the form. If they fill in the whole cell, it is considered not to be marked. In order to do that, for those cells the system considered to be marked, the percentage of marked pixels in the whole cell is computed. If it is higher than another given threshold, the cell is considered to have been cleared by the student. See question 16 in Figure 2 and Figure 3 for an example.

2.2.4 Model detection

It is frequent for MCQ exams to have different versions (models) with the same questions shuffled. The solution for these cases is to print the model identifier on the answer sheet and let the system read it from the captures.

One possibility is to perform OCR on the model identifier. However, we decided in our system to implement a simpler solution: black squares printed at the bottom of the answer sheet (see Figure 1) encode, with a high degree of redundancy, a binary number that represents the model identifier. Those squares are aligned with the columns of the answer tables in order to make it easier for the system to identify situations in which the geometry of the answer tables has not been accurately detected.

2.2.5 Student's ID number detection

A simple OCR technique is applied to detect students' ID numbers. It is based on computing where a written digit intersects a grid of horizontal and vertical straight lines. For a given handwritten digit, number and positions of those intersections are matched against a series of regular expressions that encode the expected positions and number of intersections for each digit. A score is given to each possible digit and the digit with the highest score is selected.

When the system is given a list of student IDs, this detection is significantly improved. First, a score is computed by the OCR system for each digit, reflecting the probability of a correct recognition. Then, a score is computer for each ID from the given list. The ID with the highest probability is selected. As it is shown in Section 3, this mechanism allows the system to significantly increase the reliability of the OCR to the point where IDs are recognized with minimum impact on performance.

3 Validation and Results

The *Eyegrade* system was implemented using the *Python* programming language[4] and its standard library, as well as three additional libraries: *OpenCV*[5] for image capturing, implementation of the Hough transform, thresholding algorithm and mask drawing; *Tre*[6] for approximate regular expression matching; and *Pygame*[7] for the user interface. All these libraries are multi-platform, available under *free software* licences and can be easily installed in conventional personal computers.

Several experiments were carried out in order to estimate the system performance in terms of:

1. Precision in the detection of answers.
2. Precision in the detection of the model identifier.
3. Precision in the detection of student ID numbers.
4. Average time needed to process an answer sheet.
5. Maximum number of questions per answer sheet.
6. User satisfaction.
7. Software stability.

The system was evaluated in two stages. In the first stage, four lecturers at *Universidad Carlos III de Madrid* volunteered to test a prototype and answered a questionnaire about it (*experiment A.1* in section 3.1). Their feedback was used to produce an improved prototype of the system.

In the second stage, four experiments were carried out using the improved prototype. In the first experiment (*experiment B.1* described in Section 3.3), the first four performance parameters were measured by scanning 233 answer sheets collected in an exam of a regular B.Sc. course.

In the second experiment (*experiment B.2* in Section 3.4), the system was tested with different dispositions and sizes of answer tables in the answer sheet in order to determine

[4] http://www.python.org/ (accessed 10-1-2011)
[5] http://opencv.willowgarage.com/ (accessed 10-1-2011)
[6] http://laurikari.net/tre/ (accessed 10-1-2011)
[7] http://www.pygame.org/ (accessed 10-1-2011)

the maximum amount of questions per answer sheet the system can handle with a reasonable error rate.

In the third experiment (*experiment B.3* in Section 3.5), the same four lecturers interacted again with the system in order to evaluate improvements with respect to the first prototype.

In the fourth experiment (*experiment B.4* in Section 3.6), the software stability of Eyegrade was tested.

The setup in all the experiments consisted of a low-end 19€ webcam (model Conceptronic Chatcam 2), with resolution 640x480, mounted on top of a cardboard tray made from the lid of a box. The time needed to set up that infrastructure and align the camera was approximately 3 minutes. Although a webcam with better resolution would offer a more reliable reading, this model was chosen to show the results in a worst case scenario.

3.1 Experiment A.1

Four instructors of *Universidad Carlos III de Madrid* tested the first prototype of the system and answered a questionnaire about it. None of the selected instructors was part of the team that developed *Eyegrade*. The questionnaire included questions about:

- Mechanics of the system: setup of the camera and tray, ease of aligning of the camera for the first exam, the manual process of feeding the system, etc.
- Quality of the detection system: time needed to scan an exam and accuracy.
- User interface: ease of use and time needed to learn how to use it.
- Overall recommendation: whether they would use the system or recommend the system to colleagues.

Answers to those questions are summarized in Table 1. In general, volunteers were very satisfied with the accuracy of the system and its user interface. However, they found the physical setup of the system to be rudimentary and subject to improvements. All the volunteers would be happy with using the system in courses in which they already have MCQ exams, and recommend it to colleagues.

In addition, they were asked to write down a list of strengths and weaknesses of the system, as well as suggestions for new features or improvements.

Users identified as strong points the accuracy in detecting answers, the speed to process the answers, and the intuitive user interface.

Their main concerns were regarding the setup procedure for the system (stability of the tray and the time needed to align the camera), the time needed to detect some exams (in some cases they had to modify the position of the sheet until the system captured the exam correctly), and the reliability in the detection of student IDs (although they agreed the interface for fixing incorrectly detected IDs was easy to use).

One user was also concerned about the possibility of an incorrectly detected answer going unnoticed, but later admitted that the probability of this scenario was really low. She suggested the inclusion of a mechanism to *flag* borderline decisions of the system (i.e. decisions based on values very close to the threshold).

Other user suggestions were about the configuration (e.g. user interface for providing the system with the number of questions, correct answers for each model and other configuration parameters; ability to insert solutions or number of questions in the system just by pointing the webcam to an answer sheet filled with the correct answers), the user interface (in-line help system) and post-processing of results (connection to LMS, generation of reports).

Physical setup	μ	σ
Setup of the camera	3.50	0.866
Adjusting camera for 1st exam	3.50	0.500
Placing/removing exams from tray	4.25	0.433
Physical stability of the setup	3.25	0.433

Automatic detection system	μ	σ
Time needed to detect an exam	4.75	0.433
Accuracy (student's answers)	5.00	0.000
Accuracy (ID numbers)	4.50	0.500

Ease of use	μ	σ
Time needed to learn how to use it	4.75	0.433
Interaction with the interface	5.00	0.000
Reviewing student's answers	5.00	0.000
Fixing wrongly detected answers	5.00	0.000
Reviewing student's ID number	4.75	0.433
Fixing wrongly detected IDs	4.75	0.433

Overall recommendation	μ	σ
Saves time for MCQ exams?	5.00	0.000
Would you use it regularly?	4.25	0.829
Would you recommend it to others?	4.50	0.500

Table 1 Average (μ) and standard deviation (σ) of the volunteers' answers to the questionnaire. Ranges from 1 (very poor) to 5 (fantastic). In *Overall recommendation* results range from 1 (definitely not) to 5 (definitely yes).

3.2 Second prototype

From all the collected information about this first experiment a second prototype of the tool was produced. The task focused on improving the detection time of some answer sheets. The problem was detecting cell bounds in the student ID field of the answer sheet. Several image captures were discarded because the system was not able to accurately identify these cell bounds. The discarded captures forced the user to waste time changing the orientation of the answer sheet. In the second prototype, the algorithm to detect student IDs was redesigned to raise the probability of detection in the first capture, thus significantly reducing the time to detect the marks in the answer sheet.

Furthermore, for those cases in which ID numbers were difficult to detect, the user interface was extended with the possibility of capturing the image without ID detection and directly typing the number with the keyboard. This option was particularly useful for those answer sheets with student IDs totally unreadable or incorrect (the student wrote a different number).

In addition, the second prototype included other minor improvements such as a mode to help align the camera with the exam, or the ability to introduce student IDs even when they are not in the given list (for example for late enrollments).

3.3 Experiment B.1

In this experiment, 233 exams obtained from an undergraduate course at *University Carlos III of Madrid* were processed using the improved prototype of the *Eyegrade* system. Each exam had 20 multiple choice questions with 4 choices per question. ID numbers contained 9 digits each. Exams were photocopied using $80\,g/m^2$ A4 recycled paper. The answer sheet was stapled together with the exam. At the beginning of the exam, students were instructed to mark out their answers with a cross joining opposite corners of the cell.

The system was configured to log the overall time needed to process each exam. The complete period of time between two consecutive exams was measured. It included time

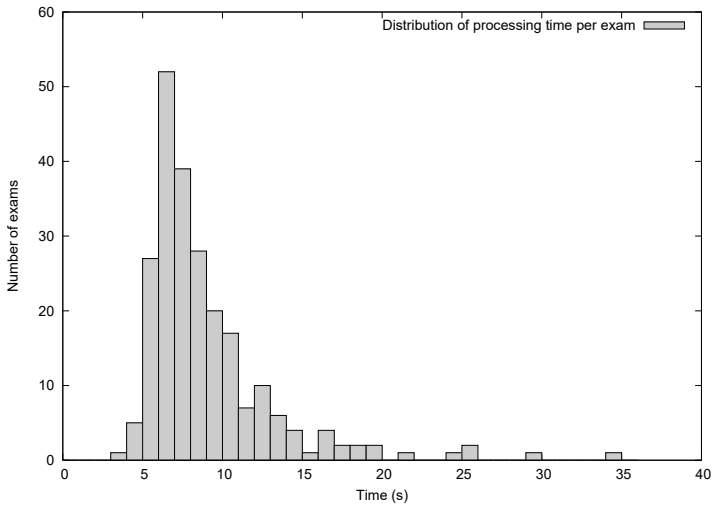

Fig. 6 Distribution of overall processing time per exam.

needed for: placing the answer sheet on the tray, waiting for the system to capture data, reviewing the answers and student's ID number detected by the system, manually fixing detection errors and removing the answer sheet from the tray. Additionally, the system counted the number of changes done by the instructor to correct erroneous decisions done by the system.

All 233 exams were processed. Only in one of them a piece of white paper was needed to cover text writen in the middle of the answer sheet that prevented a correct detection. Processing all 233 exams took 2110 seconds (35 minutes, 10 seconds). The median time per exam was 7.73 seconds, with an average of 9.05 seconds per exam. The distribution of processing times per exam is shown in Figure 6.

Table 2 shows the precision in the detection of correct answers. A total of 97.0% of the automatic decisions of the system were correct (no user intervention was needed). Of the remaining 3%, most of the erroneous decisions were due to students not following the instructions to mark their answers. Only 29 answer sheets (0.6%) failed despite being properly marked. Analysis of these sheets showed the following weaknesses in the detection system:

- Some crosses were excessively thick and provoked the system to detect them as answers the student had cleared (20 sheets in the experiment).
- Some answers cleared by the student (see Section 2.1) were not detected as such (2 sheets).
- Some crosses not properly centered were detected as blank answers (3 sheets).
- Some answers with crosses were detected as blank due to excessive light in the cell area (4 sheets). These cases were due to the limitations of the webcam to adapt to well lit environments. Other webcam models were tested in the same environments, and they did not suffer from this problem.

The distribution of the erroneous decisions per answer sheet is shown in Figure 7. That figure shows erroneous decisions due to incorrectly marked down answers as well as those attributable to the system itself.

	Total	Correct	Student errors	System errors
Num. decisions	4660	4519	112	29
% decisions	100%	97.0%	2.4%	0.6%
Num. exams	233	205	19	9
% exams	100%	88.0%	8.1%	3.9%

Table 2 Successful and erroneous decisions in the detection of answers. The *student errors* column shows the number of decision errors due to students not following the instructions to mark their answers. The *system errors* shows the decision errors attributable solely to the system.

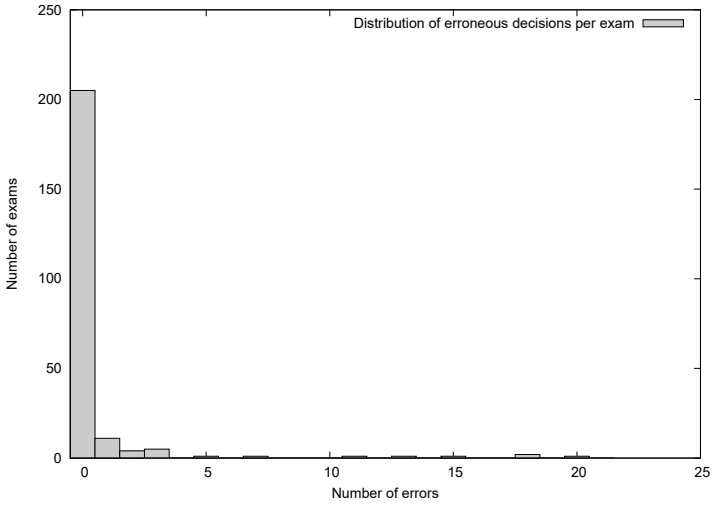

Fig. 7 Distribution of number of erroneous decisions per answer sheet.

Detection of the exam model was correct for all the exams. This is due to the high degree of redundancy of its encoding: when the checksum of the detected exam model fails the capture is discarded.

A total of 230 answer sheets contained a student ID. The system identified the correct ID numbers for 196 answer sheets (85.2%). A key factor to obtain that precision was the algorithm that, when provided with the list of IDs, chooses the most probable from the list. Without that algorithm, precision would have been only 13.5% (the OCR algorithm succeeded only for 76.7% of the digits).

In those answer sheets in which the ID was incorrectly detected, the user could correct it with a few keystrokes. Figure 8 shows the distribution of the number of keystrokes needed by the user to correct an incorrectly detected student ID. Sheets in which automatic detection succeeded are computed as 0 keystrokes. Sheets that needed the student ID to be corrected by the user were processed in an average of 16.71 seconds per exam. Sheets in which the student ID was correctly identified were processed in an average of 7.62 seconds. As a conclusion, the time needed to correct an incorrectly detected student ID is not excessively high.

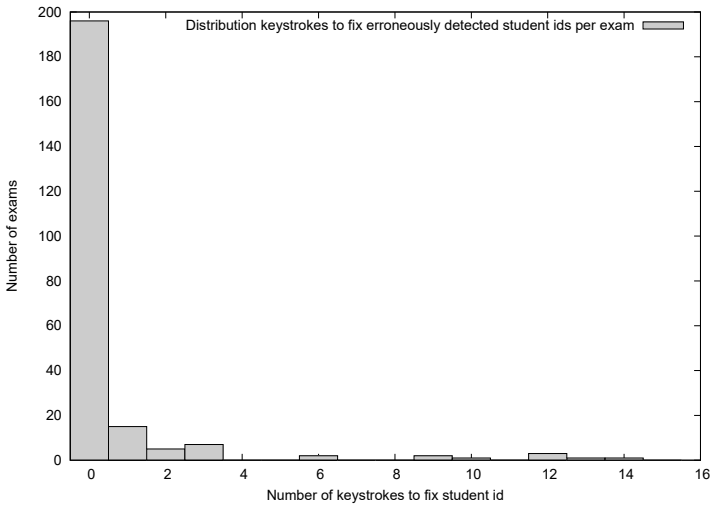

Fig. 8 Distribution of number of keystrokes needed to fix student ID numbers.

3.3.1 Comparison with manual grading

In order to compare the results of this experiment to manual grading, some exams were manually graded. The sample was extracted from 67 answer sheets divided into four models. The exams were previously classified according to their model (consecutive exams of the same model speed up the process). They were classified in 138 seconds, 2.1 seconds per exam. Out of this sample, only 15 exams of the first model were graded. A template with the correct answers was previously prepared for that model in 46 seconds. Finally, exams were graded and their marks introduced in the spreadsheet used as grade book.

Ignoring the time needed to setup the template with the correct answers (which is not significant for a large number of exams), the average grading time per exam was 55.3 seconds (2.1 seconds for classifying an exam according to its model, 41.1 seconds to grade it, and 12.1 seconds to introduce the number of correct and incorrect answers in the grade book). Comparing with the proposed tool, an average of 6 exams can be graded with *Eye-grade* in the same amount of time. The exams used in the experiments contained the answers for 20 questions. This difference would have increased for larger numbers of questions per exam.

3.4 Experiment B.2

The objective of this experiment was to evaluate the maximum amount of questions the system can reliably detect in an answer sheet in one single capture.

Results vary depending on whether the student ID has to be detected (its area in the answer sheet can be used for more questions if it is not needed) and the number of choices per question. The maximum sizes obtained were:

- With student ID number:
  - 60 questions with 4 choices each, placed in 4 tables of 15 questions per table.
  - 75 questions with 3 choices each, placed in 5 tables of 15 questions per table.

- Without student ID number:
  - 72 questions with 4 choices each, placed in 4 tables of 18 questions per table.
  - 90 questions with 3 choices each, placed in 5 tables of 18 questions per table.

Answer sheets with larger number of questions should be processed in more than one capture. In its current version, *Eyegrade* does not support this feature. However, its implementation is a simple extension of the user interface (no new functionality is needed in the core algorithms).

The previous results were obtained with answer cells of approximately 28x28 pixels in the captured image. Using webcams with more resolution would allow a larger number of questions as long as that minimum amount of pixels per cell is available. For example, doubling the resolution in each dimension (1280x960) would potentially allow to multiply by four the amount of questions per capture.

3.5 Experiment B.3

The same users that participated in *experiment A.1* were asked to evaluate the improved prototype of the system. In this experiment there were no formal questionnaire, their impressions were collected while they interacted with the new prototype.

All of them agreed that the problem that caused some exams to take too much time to detect had been solved. They found this time to be instantaneous almost always. In addition, they found the range of distances/angles between camera and answer sheet in which the system was able to detect the marks of an answer sheet to have been largely increased.

3.6 Experiment B.4

The complexity of the algorithms that process the captured images, added to the impossibility to control the kinds of images the program may have to analyze in production environments, make it important to check the stability of Eyegrade. In order to lessen the consequences of a failure while processing an image, Eyegrade has been programmed to react gracefully when unexpected exceptions happen. In that case, the current image is automatically discarded and a new one is taken from the camera.

Nevertheless, we conducted the experiment B.4 in order to test the stability of Eyegrade. The objective was checking that the system does not crash (e.g. abrupt termination due to unexpected exceptions, illegal memory accesses, hang-ups due to infinite loops, etc.) in long runs of varied image captures.

In order to process as many captures as possible, Eyegrade was configured to continuously process captures from the camera, i.e. without pausing when an exam is correctly processed. In addition, detection parameters were changed from one capture to the next. These changes allowed the system to process each exam several times with different parameters.

A batch of 153 exams was processed in Eyegrade in five rounds, each round with a different position of the camera in order to capture different perspectives of the exams. Lighting conditions were also changed in some rounds. After that, other documents and several exams with geometries different from the expected were processed. Finally, we processed also several minutes of video taken from the office and desk, in order to check the robustness of Eyegrade when presented with images that are not taken from documents.

The experiment run for approximately 120 minutes, in which 35,712 images were processed by Eyegrade. More than a third of them (13,766 images) were detected as exams. The rest were transitions between exams, other documents, exams with other geometries or just video taken at the office. The program ran continuously from the beginning to the end of the experiment. It suffered neither from hang-ups nor from unexpected exceptions (even those that the protection mechanism mentioned at the beginning of the section would catch). We conclude from this experiment, given the variety of images tested, that the system is stable.

4 Discussion

The system presented in this paper rivals other OMR solutions based on scanners in terms of performance at a much lower cost:

- Only a scanner equipped with an automatic feeder and the capacity to scan about 10 pages per minute minimum could compete in terms of time with *Eyegrade*. However, the price of such a scanner is currently higher than the webcam used by Eyegrde.
- Whereas a scanner is bulky, the instructor can easily take a webcam to class, home, a trip, etc.
- Although a typical webcam has a very low resolution compared to a scanner, experiment B.2. shows that up to 60 or 90 questions per answer sheet can be detected with a single image capture, which is an acceptable amount for most exams. The amount of questions can be easily doubled with a higher resolution webcam, available at prices as low as 30 US$.
- In contrast to other OMR systems, there is no need to use expensive answer sheets. The experiments presented in this paper used answer sheets printed by a normal printer and photocopied on non-white recycled paper.
- Supervised approach: *Eyegrade* adopted a supervised approach to grading. It allows instructors to be sure that no scanning errors occurred, to easily process exams not filled according to the instructions, and to avoid cheating. There are videos available in the net showing how to exploit errors in OMR systems to get high marks. These techniques are rendered useless when using a supervise procedure.

Another differentiating factor of *Eyegrade* with respect to many OMR solutions is the ability to perform handwritten digit recognition. With this feature, students fill in their ID number in a more convenient and intuitive way compared to marking bubbles in the answer sheet. Current results show that 85% of the student IDs are correctly detected, provided that the list of these numbers is available.

The most relevant comparison of the obtained results is with *GradeCam*, because it follows a similar approach: Although there are no available public results of the performace obtained with this product, the information made available by the vendors still allows the comparison of the main features of both systems:

- Grading time: demonstration videos of *GradeCam* suggest it is faster than *Eyegrade*. Time is in the order of 2 seconds per exam. However, those times are achieved with the user not checking whether answers have been correctly detected.
- Answer sheets: *GradeCam* answer sheets can only have 20, 40, 70 or 100 questions, whilst *Eyegrade* allows instructors to use forms with the exact number of questions they need. In addition, questions in *GradeCam* must have 4 or 5 possible answers, whilst *Eyegrade* does not impose any limit as long as the form fits in a camera capture with

enough size. The other advantage of *Eyegrade* is that answer sheets need no specific software to be produced. Any document editing environment can be used because the system is flexible with sizes and proportions of answer tables.
- Students' ID numbers: *GradeCam* is not able to perform OCR on handwritten digits. It needs students to mark their ID number in bubbles or instructors to print adhesive labels or answer sheets with student ID numbers already marked in the bubbles.
- Review and correct approach: *GradeCam* does not show the answers layered on top of the capture of the exam, but in a separate list, which makes it more difficult to review and correct the potential system errors.
- Ability to clear answers: in *Eyegrade* a student can clear an answer even when it is written with ink. In *GradeCam* it can be done only if the answer is written with pencil.
- Hardware requirements: *GradeCam* works with a specific hardware device including a tray and a camera. *Eyegrade*, on the other hand, can work with any webcam (even webcams embedded in laptop computers). Although the *GradeCam* device seems robust, a similar device could also be made for *Eyegrade*. Furthermore, with *Eyegrade* the camera does not need to be fixed. Users may hold it in their hand aiming at the answer sheet.
- Integration with grade books and LMSs: *GradeCam* is superior in that aspect because of its maturity as a commercial product. The current version of *Eyegrade* produces CSV files with scores that can be imported in spreadsheets and any other software that accepts this format.
- User interface: due to its prototype phase, *Eyegrade* has a command line interface to select options, whereas *GradeCam* has a window interface.

As a conclusion, the shortcomings of *Eyegrade* with respect to *GradeCam* are due to its stage of development. The current prototype is completely usable and has all the core functionality implemented, but needs some extra features to gain a widespread use (mainly, a more user-friendly interface, connection to LMSs and grade book applications, all-in-one installer, plug-ins for the major systems with which instructors edit their exams). Future enhancements will focus on those areas.

Experiments A.1 and B.3 show that the lecturers that used the system were highly satisfied, agreeing that it could be deployed in their courses as is. The aspects to improve are mainly the same listed above as shortcomings of *Eyegrade* with respect to *GradeCam*: physical system setup, graphical user interface for configuration, connection to LMSs and connection to exam authoring tools.

5 Conclusions

In this paper *Eyegrade* a low cost optical mark recognition based system to grade multiple choice tests has been presented. The system achieves a significant reduction in instructor grading time of multiple choice exams while lowering the adoption barrier.

With respect to other OMR systems, the advantages of *Eyegrade* are: its low cost, because only a regular webcam is needed; its portability, because of the small size and light weight of webcams; and its ability to recognize handwritten digits to read student IDs.

The tool follows a supervised model in which instructors manually feed answer sheets, review the decisions automatically taken by the system and correct them if necessary. The algorithm to detect marks and user interface were specifically designed to reduce to a minimum the time to grade an exam reliably. The mark detection algorithms were optimized to drop as few image captures as possible by making it tolerant to changes in distance, relative

angle and position between the camera and the answer sheet. Experimental results show that most exams are detected just with the first capture. The answers and the student ID detected by the system superimposed to the image capture so that the instructor can quickly review the result. The interface allows for quick changes of both the detected answers or the student ID.

The application has been validated with a set of experiments showing that an exam can be graded in an average of 9 seconds (almost 7 exams per minute), which is a reasonable amount of time even when compared to unsupervised systems based on scanners with fast automatic feeders. In addition, it already attracted the attention of several instructors of Universidad Carlos III de Madrid outside the development team, who use it in their undergraduate courses.

Future work will focus on improving the overall user experience: multi-platform installation packages, user interface for configuring the system and adaptors for major LMSs and authoring tools. In addition, we are exploring porting *Eyegrade* to smart-phones, which would make it more handy for some users.

We plan to publicly release *Eyegrade* with a free software licence when all the non-technical requirements are removed from the installation process. The platform will then be easily installable in at least Microsoft Windows and major GNU/Linux distributions.

Acknowledgements Work partially funded by the EEE project, "Plan Nacional de I+D+I TIN2011-28308-C03-01" and the "Emadrid: Investigación y desarrollo de tecnologias para el e-learning en la Comunidad de Madrid" project (S2009/TIC-1650).

References

Apampa K, Wills G, Argles D (2009) Towards security goals in summative e-assessment security. In: International Conference for Internet Technology and Secured Transactions

Deng H, Wang F, Liang B (2008) A Low-Cost OMR Solution for Educational Applications. In: IEEE International Symposium on Parallel and Distributed Processing with Applications, IEEE Press, pp 967–970

Duda RO, Hart PE (1972) Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM 15(1):11–15

Forsythe G, Wirth N (1965) Automatic Grading Programs. Communications of the ACM 8:275–278

Karavirta V, Korhonen A, Malmi L (2006) On the use of resubmissions in automatic assessment systems. Computer Science Education 16(3):229–240

Kubo H, Ohashi H, Tamamura M, Kowata T, Kaneko I (2004) Shared questionnaire system for school community management. In: International Symposium on Applications and the Internet, IEEE, pp 408–414

Norris J, Pauli R, Bray D (2007) Mood change and computer anxiety: A comparison between computerised and paper measures of negative affect. Computers in Human Behavior 23(6):2875–2887

Rane A, Kumar A, Saini H, Sasikumar M (2009) Extending Moodle to Support Offline Assessments. In: Proceedings of National Seminar on e-Learning & e-Learning Technologies (ELELTECH), pp 31–39

Saengtongsrikamon C, Meesad P, Sodsee S (2009) Scanner-Based Optical Mark Recognition. Journal of Information Technology (Thailand) (9):69–73

Twigg C (2003) Improving Learning and Reducing Costs. URL http://www.thencat.org/PCR/Rd1Lessons.pdf

Verdú E, Regueras LM, Verdú MJ, De Castro JP, Pérez MA (2008) An analysis of the research on adaptive learning: the next generation of e-learning. WSEAS Transactions on Information Science & Applications 5(6):859–868

Winters T, Payne T (2005) What do students know?: an outcomes-based assessment system. In: Proceedings of the first international workshop on Computing education research, ACM, New York, USA, ICER '05, pp 165–172

Zammit A (2009) quexf an open source, web based paper form verification and data entry system. http://quexf.sourceforge.net/