Automatic Assessment of 3D Modeling Exams

Andrea Sanna, Fabrizio Lamberti, *Member*, *IEEE*, Gianluca Paravati, and
Claudio Demartini, *Member*, *IEEE*

Abstract—Computer-based assessment of exams provides teachers and students with two main benefits: fairness and effectiveness in the evaluation process. This paper proposes a fully automatic evaluation tool for the Graphic and Virtual Design (GVD) curriculum at the First School of Architecture of the Politecnico di Torino, Italy. In particular, the tool is designed for the 3D modeling course, taught during the second year, where students have to prove their ability to model static scenes using the open source modeler Blender. During the final exam, students are required to create a 3D model as similar as possible to a reference object proposed by the teacher and shown through a set of 2D views; the similarity of the images is judged according to both model shape and materials. The traditional assessment process is particularly slow and strongly based on teachers subjective evaluation; the proposed solution efficiently implements an objective assessment mechanism that exploits computer vision and image analysis algorithms to automatically extract similarity indices. These indices are related to partial evaluation grades, which are then combined to obtain the final mark. A comparison with the traditional assessment process shows robustness and trustworthiness of the designed approach.

Index Terms—Evaluation methodologies, teaching and learning strategies, higher education, computer graphics, 3D modeling.

✦

---

1 INTRODUCTION

THE Graphic and Virtual Design (GVD) curriculum at the First School of Architecture of the Politecnico di Torino, Italy (three years, roughly corresponding to the BS in the American System) aims to train students in disciplines concerning the design and implementation of graphic and multimedia content. Students are taught the fundamentals of art history and communication techniques as well as the fundamentals of computer science, computer graphics, and multimedia.

In particular, three courses are focused on topics in computer graphics: the first course aims to teach the fundamentals of computer graphics, and it functions as an introduction to 2D concepts, the second course presents methodologies and tools for modeling 3D static scenes, while the last course is devoted to teaching basic techniques for animation and interactive graphics.

The 3D modeling course considered in the present work counts for four credits in the European ECTS system [1], which are equivalent to 60 hours of teaching. Given the current schedule of the School, the courses have severe time constraints; in particular, the considered course fits into a 12-week slot and is structured in lessons of 5 hours each. Each lesson is taught in a laboratory and organized as follows: the teacher introduces the topic of the lesson and presents the basic theoretical concepts; for instance, when explaining concepts related to solid modeling, Constructive Solid Geometry (CSG) rules [2] are illustrated. Afterward, students work through one or more tutorials step-by-step using the open source modeling tool Blender [3]; finally, the

teacher asks the students to build a model starting from a set of screenshots (2D rendered views of a 3D model) and individually supports the students in completing their task.

The last phase of each lesson allows students to simulate the final exam; the goal of the exam is to evaluate the students ability to create a 3D model as similar as possible to a reference model provided by the teacher and represented through a set of screenshots. Students can choose, in general, among three to five different models (some examples are shown in Fig. 1); a target number of polygons is specified for each model (low-poly modeling is a worthwhile task), and images for texture mapping are provided, when necessary. Students are asked to create a model as similar as possible to the teachers reference in terms of geometry, materials, and number of polygons. At the end of the exam, each student delivers his or her own Blender project, together with a set of well-defined renderings of the generated model.

An academic board, usually comprised of three teachers, assesses the students work. Each teacher examines the model and the renderings and proposes a mark. The final mark is obtained after a discussion of individual judgments. Unfortunately, this approach has two main drawbacks: the assessment process can be very time consuming and, most importantly, grading is based on a similarity evaluation that is often strongly subjective.

The current work aims to tackle the above issues by proposing a fully automatic assessment tool able to efficiently provide teachers and students with objective evaluations of works of 3D modeling. The results of previous examination sessions using the traditional evaluation method were used to infer objective assessment criteria for use in fine tuning the tool. A comparison between evaluations by the academic board and results generated by the automatic tool is also provided.

The organization of the paper is as follows: Section 2 summarizes the local context and educational goals for the considered course and presents the motivations for the development of an automatic evaluation tool. Related work

---

- *The authors are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. E-mail: {andrea.sanna, fabrizio.lamberti, gianluca.paravati, claudio.demartini}@polito.it.*
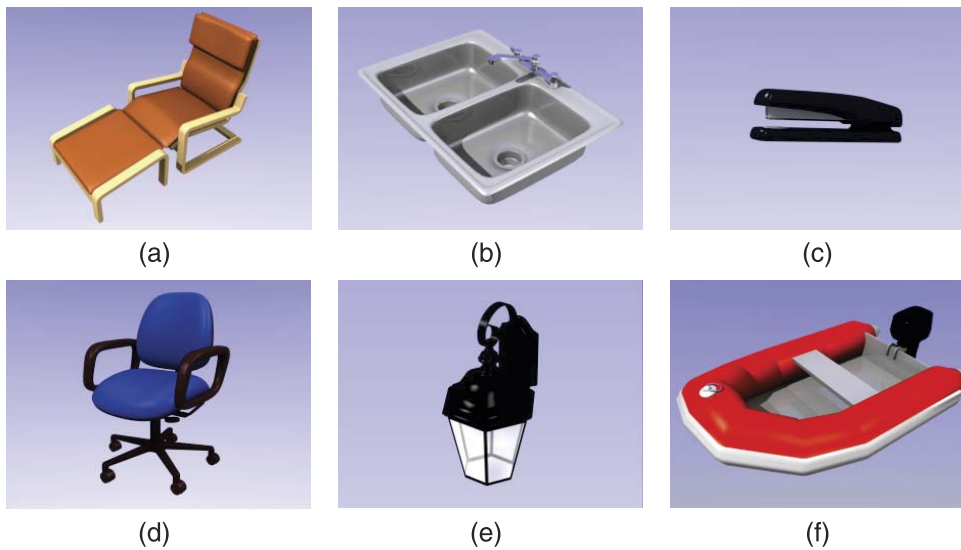
Fig. 1. Examples of screenshots provided by the teacher representing reference objects to be modeled during the exam.

is discussed in Section 3, and the proposed assessment solution is illustrated in Section 4. Finally, results and remarks can be found in Section 5.

2 CONTEXT AND MOTIVATIONS

The 3D modeling course considered in the present work builds upon a previous course aimed at teaching the fundamentals of computer graphics. The course aims to provide students with basic techniques for generating (photo-realistic) 3D computer generated images. In particular, according to the first two Dublin descriptors [4], the goal learning outcomes can be summarized as follows:

- knowledge of how to create and manipulate models by curve modeling, solid modeling, and sculpt modeling and ability to apply this knowledge to develop complex 3D geometries;
- knowledge of techniques for assigning and changing attributes related to a model (color, transparency, reflectivity) and ability to use this knowledge to setup object materials;
- knowledge of shading models and light-material interaction techniques and ability to apply this knowledge to control the appearance of rendered objects;
- knowledge of 2D and 3D texture mapping techniques and ability to use this knowledge for controlling/changing object appearance;
- knowledge of rendering algorithms (local and global models) and ability to apply this knowledge to obtain photo-realistic images;
- knowledge of direct and inverse kinematics and ability to pose a virtual character and to develop constrained systems; and
- knowledge of particle systems and ability to apply this knowledge to simulate phenomena such as fire, smoke, rain, etc., and to manage soft bodies and fluids.

The 3D modeling course is mandatory for the GDV curriculum, and about 120 students are enrolled each year. Eight exam sessions are scheduled per year: three sessions at the end of the course, three sessions before the summer holiday, and two sessions in September before the beginning of the new academic year.

An in-depth evaluation requires no less than 10 minutes per each exam. The Blender project must be opened and analyzed; then, each rendered image has to be compared with the reference screenshots. Based on the outcomes of the above steps, each teacher belonging to the academic board makes a judgment of the exam work, and a final mark is assigned after a collective discussion. If the individual marks are significantly different, the collective discussion can be quite time consuming. Finally, marks are published on the website of the School and students have, in general, one week to accept or reject the mark. Students can also ask for an explanation of the assessment, thus involving the academic board in a further commitment.

Indeed, a fully automatic assessment tool could strongly speed up the evaluation process, thus relieving the academic board of a significant burden. However, even though the reduction of the time required for assessment was an important reason behind the design and development of the proposed tool, the main motivation arose from the need to improve the fairness of the evaluations.

In fact, despite the presence of an academic board, individual evaluations are traditionally based on subjective judgments. Teachers have to take into account the similarity of the student's model to the reference model. Similarity encompasses the model's shape (mesh) as well as the uses of materials and textures. Finally, the number of polygons in the student's model is considered against a target number in the reference model (however, even though low-poly modeling is an important task, this issue has a marginal weight in the overall evaluation).

While similarity in terms of the number of polygons can be easily expressed by means of a numerical value, finding and (manually) applying objective evaluation criteria for shape and material-based similarity factors is a far more complex task. This complexity is due to the fact that qualitative perception is strongly influenced by a marked dependency between model shape and visual appearance.

As an example, consider Fig. 2; Fig. 2a shows one of the screenshots the teacher provided as a reference for an exam,
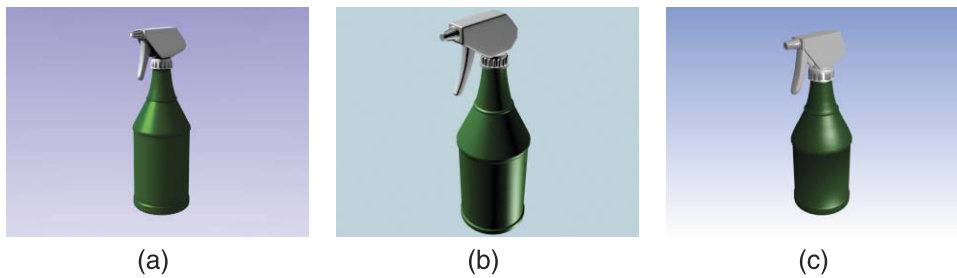
Fig. 2. Views of a 3D model from an exam session: (a) reference model, and (b)-(c) rendered images delivered by two students.

whereas Figs. 2b and 2c display the 2D rendered views of the models created by two students. It should be evident that determining whether the model in Fig. 2a is more similar to the model in Fig. 2b or in Fig. 2c is a hard task. Nonaligned objects and the use of materials and textures introduce further complexity. Fig. 3 presents this issue: the spray bottle in Fig. 3c is the same in Fig. 2c, but it has been rotated and scaled, and a different material has been assigned to the bottom part of the model. From a subjective point of view, the spray bottle in Fig. 3c seems to be more different from the reference model in Fig. 3a than the model in Fig. 3b, even though the mesh was not changed.

The above examples confirm that, although the collective discussion of individual marks may contribute to an increased fairness of the final mark, objectiveness can only be achieved by a computer-based system capable of isolating similarity factors and implementing measurable assessment rules in an automatic way. The following sections review the state of the art in computer-based assessment and present the strategy pursued for the design and implementation of the proposed tool.

3   RELATED WORKS

Computing systems have been used in education since their earliest appearance. However, it was only in more recent years, with the widespread diffusion of personal computers and the evolution of network infrastructures, that the use of the associated technologies significantly impacted all the facets of teaching and learning processes. In the context of learning assessment, software tools started to be developed for the evaluation of test-based assignments and were applied in a variety of educational contexts [5], [6], [7]. Various technological solutions were progressively exploited to increase assessment effectiveness [8], mainly by focusing on strategies for the automatic construction of test sheets [9], [10], [11], [12].

Despite guarantees in terms of objectivity, when the evaluation of learning outcomes requires considerations of applying knowledge in a specific context, test-based assignments are generally replaced by assessment techniques based on performance evaluation [8], [13]. The potential for unfair assessment for this type of assignment is higher than for test-based evaluations. As stated in [14], the main issue with computer-based performance assessment is related to the complexity of translating very specific grading methodologies relying on measurable objectives (e.g., rubrics) into an automatic evaluation logic.

Thus, vertical solutions, each tackling a particular assessment problem, have been proposed. Although a few solutions aimed at computer-based assessment outside scientific domains exist, e.g., automatic essay or pronunciation evaluation [15], [16], [17], most of the works reported in the literature focus on the assessment of technical subjects. Indeed, most of the researchers' attention has been devoted to the automatic assessment and grading of computer programming assignments. Solutions proposed in this field differ in the set of programming languages addressed, in the strategy used for weighting error severity, in the ability to evaluate code quality, in the degree of integration with existing development environments, and in the availability of antiplagiarism features [18], [19].

Nonetheless, many fields other than computer science have been explored. For instance, in [20], a tool for assessing automata-based assignments is proposed. Automatic assessment of formal specification coursework is addressed in [21]. Computer-based grading in the field of database design has been studied in [22] and [23]. A software tool allowing an automatic check and verification of a student's laboratory work in the design and simulation of digital circuits is illustrated in [24]. In [25], a system for checking exercises in the field of dynamic geometry systems is illustrated. In [26], a software platform for self-assessment in the automatic control systems domain is presented. In
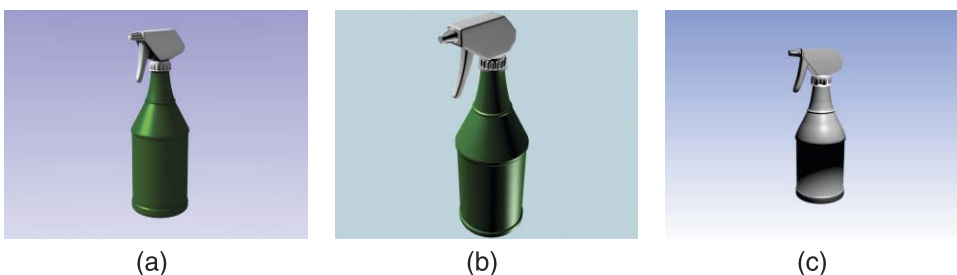

Fig. 3. Views of a 3D model from an exam session: (a) reference model, and (b)-(c) rendered images delivered by two students; the model in (c) is the same as Fig. 2c, but it has been rotated, scaled, and a material has been changed.

[27], a computer-based system is used to teach and assess industrial robot path planning and control.

Most of the assessment techniques above use some kind of computer graphics and multimedia techniques to enhance their communication potential [28]. In some cases, distinctive features of computer graphics are even used to teach other subjects, as in the case of [29], where computer game development is exploited to teach object-oriented programming. Furthermore, ever more powerful applications of computer graphics techniques, like virtual and augmented reality, are being exploited to build effective virtual laboratories and further enhance learning experience through virtual tutors [30], [31], [32], [33], [34], [35]. However, despite the pervasiveness of the above techniques in the framework of education, only a few works in the literature have focused on the issue of automatically assessing the outcomes of computer graphics-related courses.

In particular, Jiang et al. [36] present a method for autoevaluating Photoshop images and Flash animations. The system considers the content of the electronic file as well as the log of the student's operating processes, and it extracts technical information such as image size, number of layers, font size, etc. Artistic factors are only marginally taken into account by considering the organization of the work in terms of color and object location. Assessment is performed by executing a matching algorithm based on fuzzy logic against a reference in the form of text.

Some efforts have also been devoted to investigating computer-based assessment in the field of Engineering Drawing (ED) and Computer Aided Design (CAD). In fact, whereas this kind of evaluation used to depend only on the final printouts, today the amount of information generated by existing software tools is impressive, and objective evaluation has become an onerous task. However, ED and CAD learning outcomes are generally assessed according to consolidated and structured criteria [37]. Thus, based on the effective rubrics available [38], several computer-based grading systems were developed.

In particular, in [39] and [40], an automatic grading tool for a freshman solid modeling course is presented. The tool is designed as an integrated instrument allowing teaching staff to manage the class (e.g., by supervising student to section allocation, generating web pages for the course syllabus, verifying the status of each student, etc.) as well as providing students with a means to check their grades and interact with the instructor in case of doubts about a given result. The tool is implemented as a collection of software modules working on Excel spreadsheets (storing grades), SolidWorks data (models and engineering drawings submitted by the students to a database manager), and web pages (containing student's information). The system handles different types of assignments (quiz, laboratory recitation, hand sketching, and final problems). Some of the assignments are graded automatically. For instance, for laboratory works the student is provided with a reference print-out and a limited set of very specific grading criteria [41], e.g., all dimensions change together, all cross sections have the same number of edges, etc. The student's goal is not to produce a nice looking picture, but to create a mechanical drawing respecting rigid constraints. The designed tool works on the file produced by the student and checks whether the dimensions and views are present and whether placement of objects is correct. Despite the

promising approach, the authors underlined the fact that the design of a fully automatic grading routine for drawings is a complex task. Thus, some of the assignments, like for instance hand sketches, need to be graded visually: in this case, an image file is created and the instructor adds the grade manually.

In [42], different grading criteria were defined by referring to the "tracing technique," a common method of assessment in the ED domain. In this technique, the teacher's drawing is traced on the student's drawing, and accuracy is measured to arrive at an initial grade. Specifically, this initial mark is computed by taking into account the following "accuracy elements": accuracy of object type (a comparison of the number of objects of the same type in the student's drawing to the number in the teacher's drawing), the accuracy of object measurement (a comparison of the number of object entities with similar attributes between the drawings of the student and the teacher), and the entity of object attribute (a comparison of the number of object entities with the same attribute in the drawings of the student and the teacher). Unfortunately, even if the student's drawing shares the exact number of accuracy elements with the teacher's drawing, it is not guaranteed that the student's drawing is exactly similar to the teacher's drawing (and vice versa). Thus, to give the final mark, the initial mark is adjusted by means of visual comparison manually performed by the teacher.

The above works confirm the need for automatic tools to provide an efficient and objective assessment in the field of computer graphics. However, the proposed methods cannot be applied for the evaluation of 3D modeling assignments. For instance, despite the availability of the necessary assessment information, the approach in [36] would not be effective in the domain tackled by this paper. In fact, the student could follow multiple routes to reach the same result; thus, the analysis of the modeling process would be of little to no help. Moreover, technical and artistic factors in 3D modeling (e.g., similarity with a template shape, materials usage, etc.) go beyond the strict engineering requirements of ED and CAD considered in [42]; thus, they could not be judged individually by looking at the content of the electronic file, but would have to be considered as a whole by taking into account the concurrent effect of various factors such as camera position, lights configuration, etc. These various factors would necessarily require working with the resulting image rendering. As anticipated, given the importance (and complexity) of visual inspection, a computer-based assessment technique for 3D modeling should, therefore, automate this step.

4 THE PROPOSED TOOL

In the following, the proposed assessment tool is analyzed in detail. In particular, Section 4.1 presents the overall evaluation strategy. Section 4.2 illustrates the mechanisms controlling the determination of the mark. Finally, Section 4.3 discusses the configuration steps.

4.1 Assessment Strategy

According to the methodology illustrated in Section 1, for each exam session, the teacher selects a set of objects to be chosen by the students; objects are grouped in three

(a)

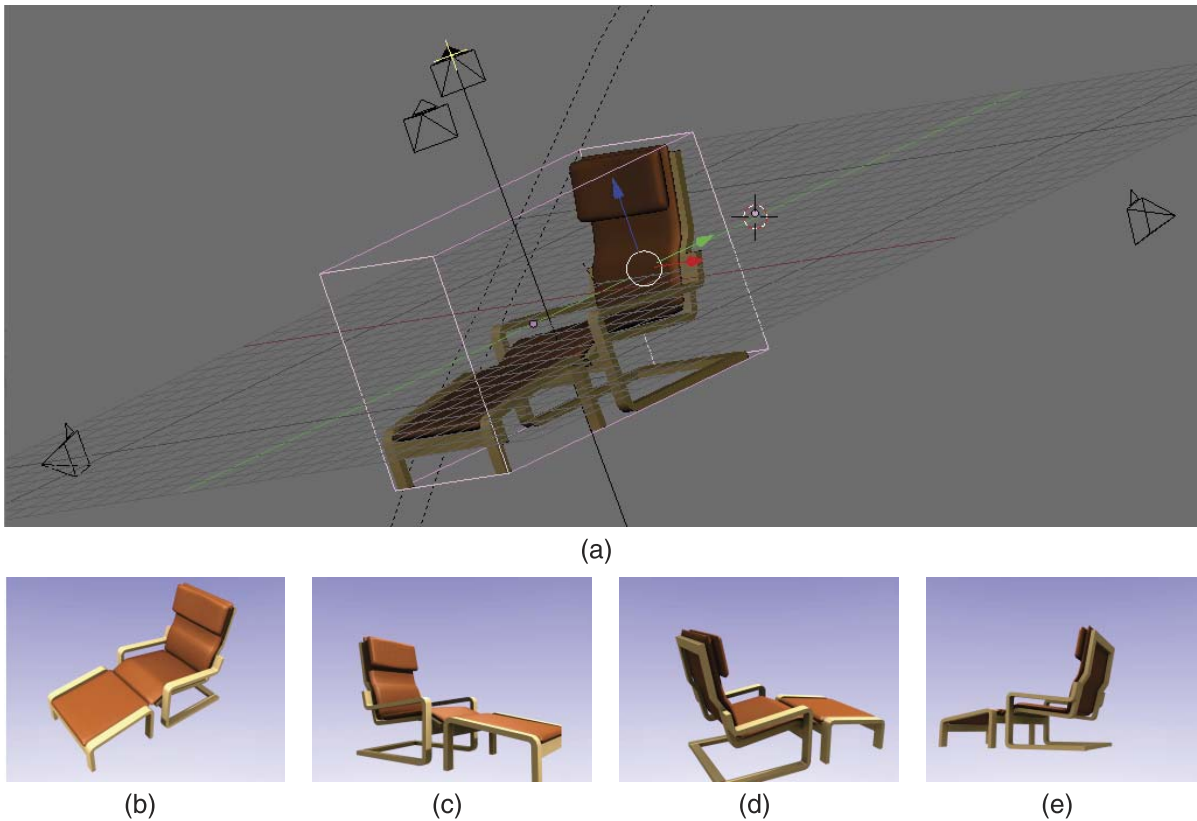(b)   (c)   (d)   (e)

Fig. 4. An example of exam package: (a) reference model aligned and encapsulated into the bounding box, and (b-e) related screenshots.

categories (easy, medium, and difficult), and each category is associated with a difficulty coefficient k that is used to fairly compare modeling works of different complexity. For each possible choice, a package `object_name.zip`, composed of the following items, is delivered to students:

1. a file named `empty.blend` containing: a set of N cameras, a set of L lights and a bounding box;
2. a set of N screenshots representing the object to be modeled; each screenshot is rendered using one of the N cameras set in the file `empty.blend`, and it is named `object_name_1.jpg`, `object_name_2.jpg`, ..., `object_name_N.jpg`, respectively.
3. a further screenshot, named `object_name_bounding.jpg`, showing the object to be created within the bounding box; and
4. a value representing the reference number of polygons (NP_{ref}) for the object to be modeled and the required resolution for the rendered images.

Lights, cameras, and the bounding box provided in `empty.blend` cannot be altered. This ensures uniformity in light-material interaction as well as in the reference point of the views used for generating the rendered images. Moreover, even though the strategy used to measure shape similarity is robust against translation, rotation, and scaling (see Section 4.2), the bounding box represents a further constraint ensuring that modeled objects are aligned and scaled to match the configuration in the screenshot `object_name_bounding.jpg`; in this way, the check of material similarity can be carried out without any model rescaling and/or realignment.

As an example, the package provided for the model `lounge_chair` in Fig. 1a is shown in Fig. 4: in the upper part the screenshot `lounge_chair_bounding.jpg` is shown (the bounding box containing the chair, the reference system for the alignment and the four preset cameras are clearly visible), while in the lower part of the figure, the four screenshots `lounge_chair_1.jpg`, `lounge_chair_2.jpg`, `lounge_chair_3.jpg` and `lounge_chair_4.jpg` are depicted. In this case, the resolution required for the rendered images was $1,024 \times 768$ pixels, and the reference number of polygons was set to $NP_{ref} = 1,640$.

Each student is required to create his or her own model within the `empty.blend` file and save it using his or her identification code as a filename, e.g., `s123456.blend`; in the same way, N rendered images at the required resolution have to be generated using the preset cameras and saved as `s123456_1.jpg`, `s123456_2.jpg`, etc.

The automatic assessment tool receives the following inputs: all the files delivered by the student, the N reference screenshots, the reference number of polygons, and a text file. The text file, named `coordinates.txt`, stores three integer values per row: the first value is an index identifying the screenshot (ranging from 1 to N), whereas the remaining values represent the coordinates of an image pixel; these coordinates identify specific parts of the model in a given screenshot, thus allowing the tool to evaluate the use of materials over the model. The constraint on alignment/scaling given by the reference bounding box allows the system to compare materials belonging to the same model part.

4.2 Determination of the Mark

In order to evaluate exam work, the automatic assessment tool performs several steps aimed at transforming subjective parameters into measurable performance objectives. These steps are as follows:

1. computation and comparison of the number of polygons;
2. computation of a mesh similarity index; and
3. computation of a material similarity index.

Step 1 is performed in order to assess the ability of the student at applying low-poly modeling techniques, and it produces a partial mark M_1, which can be either zero (if the complexity of the geometry is comparable to the reference one) or a negative integer value (up to -2).

Step 2 uses the algorithm proposed in [43] to produce a partial mark M_2 concerning mesh similarity, which is a value from 0 to -20. The main idea underlying this evaluative technique is that if two models are similar, they also look similar from all viewing angles. In other words, the similarity between two 3D models is estimated by measuring and summing the similarity of pairs of corresponding images obtained from the same points of view. In order to do this, an automatic system takes the two models to be compared and aligns, rotates, and scales them in order to obtain the maximum cross-correlation. Then, all lights are turned off (rendered images will be only silhouettes), and a set of cameras is placed on the vertices of a fixed regular dodecahedron. The silhouettes are rendered for both objects, and for each pair of images, an index of similarity is computed; the final similarity index is the sum of all the partial indexes. This algorithm is robust against translation, rotation, scaling, noise, decimation, and model degeneracy; an implementation is freely available for trial use at: http://3d.csie.ntu.edu.tw.

Finally, Step 3 computes a sort of "color distance" to assess material similarity. For each row of the file coordinates.txt, the tool compares the corresponding pixels of the ith reference screenshot and of the ith rendered image and computes a distance in color space. The RGB space is perceptually nonlinear; thus, equally sized distances in different portions of the RGB color cube appear as different distances to the human visual system. It is possible, however, to evaluate the distance in a more perceptually oriented space, such as the Hue Saturation Value (HSV) space [44]. RGB values are, therefore, converted in HSV values by means of a simple procedure [2]. The average euclidean distance of all the considered pixels provides a partial mark M_3, which is a value from 0 to -10. Given the HSV colors of two pixels $P_1 = (H_1, S_1, V_1)$ and $P_2 = (H_2, S_2, V_2)$, the euclidean distance ED is calculated as

$$ED = \sqrt{(H_1 - H_2)^2 + (S_1 - S_2)^2 + (V_1 - V_2)^2}. \quad (1)$$

The final mark is computed as

$$M = 33 + M_1 + M_2 + M_3. \quad (2)$$

The highest mark is 33, which corresponds to 30 *cum laude*. Marks lower than 18 are considered failing. The tool writes the student's identification code and final mark in the file results.txt, on separate rows.

4.3 Configuration

In order to tune the mechanisms for determining M_1, M_2, and M_3, the results of past assessments performed by the academic board were used; in particular, the exam sessions of the academic year 2008-2009 were considered. As previously mentioned, low-poly modeling is not the main focus of the course; thus, given NP_{ref} and NP (i.e., the number of polygons of the model to be evaluated), M_1 is computed by the simple rule below

$$M_1 = \begin{cases} 0, & if \quad NP \leq 10 \cdot NP_{ref}, \\ -1, & if \quad 10 \cdot NP_{ref} < NP \leq 100 \cdot NP_{ref}, \\ -2, & if \quad NP > 100 \cdot NP_{ref}. \end{cases} \quad (3)$$

In other words, the penalty is negligible if the student has modeled his or her geometry using a number of polygons of the same order of magnitude of NP_{ref}, is equal to -1 if NP is more than an order of magnitude larger than NP_{ref} but lower than two orders of magnitude, and equal to -2 otherwise.

Concerning M_2, as illustrated in Section 4.2, the technique proposed in [43] compares two models and returns an indication of mesh similarity. The above indication is expressed by means of a complementary integer index DI denoting model dissimilarity: a larger index number represents a larger dissimilarity (the algorithm returns $DI = 0$ when the model is compared with itself). For the automatic assessment, the index DI is weighted by the coefficient k, which considers the complexity of the model to be created: it is $k = 1$ for objects categorized as easy, $k = 1.5$ for objects categorized as medium, and $k = 2$ for difficult objects (for instance, for the spray bottle in Fig. 2 it was $k = 1$, whereas for the two chairs in Fig. 1 it was $k = 1.5$).

Several tests were performed to correlate DI with M_2. It was experimentally found that indexes lower than 1,000 indicate objects that are almost indistinguishable, whereas for larger dissimilarities M_2 can be expressed as

$$M_2 = \begin{cases} 0, & if \quad \frac{DI}{k} \leq 1000, \\ -\lfloor (\frac{DI}{k} - 1000)/500.0 + 0.5 \rfloor, & if \quad \frac{DI}{k} > 1000. \end{cases} \quad (4)$$

In other words, a -1 penalty is added for each block of 500 units (or fractions thereof) of the weighted dissimilarity index DI exceeding 1,000. If $M_2 < -20$, then M_2 is set to -20.

Finally, to determine M_3, the average euclidean color distance ED is evaluated for all pixels/images specified in coordinates.txt; M_3 is computed as

$$M_3 = \begin{cases} 0, & if \quad ED \leq 100, \\ -\lfloor (ED - 100)/100.0 + 0.5 \rfloor, & if \quad ED > 100. \end{cases} \quad (5)$$

In other words, average distances in the HSV color space lower than 100 are considered as negligible, whereas a -1 penalty is added for each block of 100 units (or fractions thereof) of the euclidean color distance ED exceeding 100. If $M_3 < -10$, then M_3 is set to -10.

It is worth recalling that material similarity is here evaluated by comparing the rendered surface colors; however, visual appearance of a surface is generally determined by lights, materials and many other "variables": the shading model, the procedural texture, the rendering

algorithm, the ambient occlusion, refraction, and reflection indexes, etc. Different combinations of these variables can result in very similar or very dissimilar visual appearances. In this context, the academic board wants to assess the ability of students to create objects that are visually similar to the reference objects. In other words, if the reference model is a low reflective, green spray bottle, a semitransparent red object will be considered as a very dissimilar model from the material point of view. On the other hand, an object exhibiting a comparable color tone will be considered as coherent with the reference model, independent of the status of the above variables.

5   RESULTS

An example of automatic assessment is illustrated in Fig. 5. In this case, students had to model a simple spray bottle with $NP_{ref} = 3,430$. For the image, $k = 1$, $N = 4$ renderings were required, and the image resolution was set to $1,024 \times 768$ pixels. Fig. 5a shows a reference screenshot, Fig. 5b illustrates the corresponding rendering by a particular student, and Fig. 5c reports the silhouette of the model where the pixel coordinates used to check material similarity have been emphasized (in this case, just one point of view has been considered as sufficient to evaluate material similarity; more complex objects would require, in general, a comparison of screenshots generated from all the N cameras). The number of polygons NP used for modeling the spray bottle was 4,861, the same order of magnitude of NP_{ref}, thus leading to $M_1 = 0$. The dissimilarity index DI was 2,075, thus producing $M_2 = -3$ (the two models are slightly different in the upper region). The average euclidean color distance ED for the considered pixels was equal to 79, thus leading to $M_3 = 0$ (the difference is mainly due to the different color saturation of the bottle). The final score was 30/30.

The proposed tool, implemented as an MS Windows application, is currently under validation: for each exam session, the academic board evaluates students' work in the traditional way and compares their marks with the results generated by the designed tool. This kind of comparison is necessary in order to evaluate the trustworthiness of the automatic process. As an example, Table 1 reports a comparison between the proposed tool and the traditional assessment process concerning several students enrolled
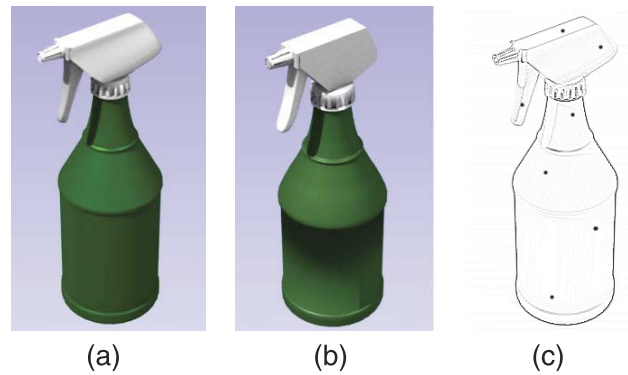

Fig. 5. An example of automatic assessment: (a) reference screenshot, (b) student's rendered image, and (c) pixels used to check material similarity.

into an exam session of the current academic year. Two models were considered: the spray bottle in Fig. 2 and the lounge chair in Fig. 4. Some screenshots delivered by students numbered 6-9 are shown in Fig. 6.

The mark determined by the academic board is sometimes expressed as a range, because it encompasses different judgments by the three members. It is noticeable that, in general, automatic and traditional evaluations may differ by one or two units; however, for student no. 1, the two marks were very different. In this case, the academic board re-evaluated the exam. It was found that teachers' evaluation had been strongly affected by the main body of the bottle: the student had not set a smooth rendering of the surface. It was a silly mistake that strongly impacted the overall visual appearance, but one that is not related to the ability of the student to model 3D objects. Therefore, the academic board decided to reconsider its evaluation and a 24/30 mark was assigned: an intermediate value between the initial mark and the automatic result.

It is worth observing that with the proposed software, like with automatic assessment tools in general, students may be drilled to focus on certain aspects which are checked through specific criteria hardwired into and mechanically checked by the system, whereas other aspects which a human grader would recognize at once may go undetected. As a matter of example, a model with a circular symmetry may be created, as any other model, by designing the polygonal mesh vertex by vertex. However, a more effective approach would require the application of a spin

TABLE 1
A Comparison between Traditional and Automatic Assessment

Student no.	Spray bottle $NP_{ref} = 3430$	DI $k = 1$	ED	M_1 M_2 M_3	Automatic mark	Traditional mark
1	6649	3526	134	0 -6 -1	26	20-22
2	765103	2143	45	-2 -3 0	28	26-29
3	4295	3186	308	0 -3 -3	27	25-27
4	4740	1221	231	0 -1 -2	30	30
5	4861	2075	79	0 -3 0	30	30

Student no.	Lounge chair $NP_{ref} = 1640$	DI $k = 1.5$	ED	M_1 M_2 M_3	Automatic mark	Traditional mark
6	1542	5674	71	0 -6 0	27	25-26
7	229576	6133	101	-2 -7 -1	23	25-26
8	1392	3393	22	0 -4 0	29	29
9	1532	4907	3	0 -4 0	29	26-27

(a)	(b)	(c)	(d)

Fig. 6. Screenshots provided by student (a) no. 6, (b) no. 7, (c) no. 8, and (d) no. 9 from Table 1.

operation on a specific profile. Similarly, the effect of a rough surface could be either obtained by an image texture, by a procedural bump mapping technique or by a true mesh deformation, and the choice of the best approach would depend on the particular modeling scenario being considered.

However, the designed tool is not able (actually, it is not meant to) evaluate the adequacy of the methodology used to reach a given modeling result, but, rather, it focuses on the result itself. In other words, given the fact that model representations are actually compared, two very similar objects, obtained using different modeling techniques, would be graded in the same way (and this may be undesirable in some cases). Nonetheless, the designed implementation has been developed to mimic as far as possible the traditional assessment based on visual inspection, i.e., to intentionally focus on the result rather than on the process. Hence, in the proposed similarity-based approach, the risk of "influencing" student's behavior is partially mitigated.

Nevertheless, should the evaluation strategy require to consider also methodology-based assessment criteria, ad hoc routines (like those developed in [36]) could be easily integrated. In this way, both process and result would be considered, making the resulting assessment technique suitable also for other disciplines, like 2D graphics and 2D/3D animation, among others.

6 CONCLUSION

This paper presents a software tool supporting the automatic evaluation of final exams of a 3D modeling course within the GVD curriculum at the First School of Architecture of the Politecnico di Torino, Italy. Computer vision and image analysis techniques are exploited to develop a fair and efficient evaluation strategy based on shape and material similarity criteria. Exam sessions of the past academic year have been used to infer objective assessment rules and to fine tune the proposed grading mechanism. Currently, the system is under validation, and it helps the academic board to smooth out the potential subjectivity of the results.

Experimental tests proved the trustworthiness and the efficiency of the proposed solution (evaluation of an exam requires a few seconds), which will be gradually used to reduce the burden associated with manual assessment. When the system has been completely tested on a significant number of exams (a test is planned for all exam sessions of the current academic year), a web-based version will be made available for students' self-assessment and exam training.

Students seem to appreciate this new approach: they know that teachers are now supported by an automatic assessment system, but they do not see the two marks. Students feel the tool, at this stage, improves the assessments made by the academic board, and the number of students asking for explanations after the publication of exam results has been significantly reduced.

REFERENCES

[1] European Commission, "European Credit Transfer and Accumulation System (ECTS)," http://ec.europa.eu/education/lifelong-learning-policy/doc48_en.htm, July 2006.
[2] J.D. Foley, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice.* Addison-Wesley, 1990.
[3] Blender Website, http://www.blender.org, July 2010.
[4] "Shared 'Dublin' Descriptors for Short Cycle, First Cycle, Second Cycle and Third Cycle Awards," working document, Joint Quality Initiative Informal Group, Oct. 2004.
[5] A.I. González-Tablas Ferreres, K. Wouters, B. Ramos Alvarez, and A. Ribagorda Garnacho, "EVAWEB: A Web-Based Assessment System to Learn X.509/PKIX-Based Digital Signatures," *IEEE Trans. Education,* vol. 50, no. 2, pp. 112-117, May 2007.
[6] M.S. Pérez, P. Herrero, F.M. Sánchez, and V. Robles, "Are Web Self-Assessment Tools Useful for Training?" *IEEE Trans. Education,* vol. 48, no. 4, pp. 757-763, Nov. 2005.
[7] A. Tartaglia and E. Tresso, "An Automatic Evaluation System for Technical Education at the University Level," *IEEE Trans. Education,* vol. 45, no. 3, pp. 268-274, Aug. 2002.
[8] G. Conole and B. Warburton, "A Review of Computer Assisted Assessment," *ALT-J Research in Learning Technology,* vol. 13, no. 2, pp. 17-31, Mar. 2005.
[9] P. Lira, M. Bronfman, and J. Eyzaguirre, "MULTITEST II: A Program for the Generation, Correction, and Analysis of Multiple Choice Tests," *IEEE Trans. Education,* vol. 33, no. 2, pp. 320-325, Nov. 1990.
[10] G.J. Hwang, "A Test-Sheet-Generating Algorithm for Multiple Assessment Requirements," *IEEE Trans. Education,* vol. 56, no. 2, pp. 329-337, Aug. 2003.
[11] H. Wainer, *Computerized Adaptive Testing: A Primer.* Hillsdale, 1990.
[12] E. Guzmán and R. Conejo, "Self-Assessment in a Feasible, Adaptive Web-Based Testing System," *IEEE Trans. Education,* vol. 48, no. 2, pp. 688-695, Nov. 2005.
[13] J. Whittington and K.J. Nankivell, "Teaching Strategies and Assessment Measures for Rapidly Changing Technology Programs," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques,* 2006.
[14] M. Al-Smadi, C. Guetl, and D. Helic, "Towards a Standardized E-Assessment System: Motivations, Challenges and First Findings," *Int'l J. Emerging Technologies in Learning,* vol. 4, no. 2, pp. 6-12, 2009.
[15] L. Indrayanti, T. Usagawa, Y. Chisaki, and T. Dutono, "Evaluation of Pronunciation by Means of Automatic Speech Recognition System for Computer Aided Indonesian Language Learning," *Proc. Seventh Int'l Conf. Information Technology Based Higher Education and Training,* pp. 553-556, 2006.
[16] H.C. Wang, C.Y. Chang, and T.Y. Li, "Assessing Creative Problem-Solving with Automated Text Grading," *Computers and Education,* vol. 51, pp. 1450-1466, 2008.
[17] T. Kakkonen and E. Sutinen, "Automatic Assessment of the Content of Essays Based on Course Material," *Proc. Second Int'l Conf. Information Technology,* pp. 126-130, 2004.

[18] C. Douce, D. Livingstone, and J. Orwell, "Automatic Test-Based Assessment of Programming: A Review," *J. Educational Resources in Computing,* vol. 5, no. 3, pp. 1-13, 2005.

[19] K.A. Ala-Mutkaa, "A Survey of Automated Assessment Approaches for Programming Assigments," *Computer Science Education,* vol. 15, no. 2, pp. 83-102, June 2005.

[20] Z. Shukur and N.F. Mohamed, "The Design of ADAT: A Tool for Assessing Automata-Based Assigments," *J. Computer Science,* vol. 4, no. 5, pp. 415-420, 2008.

[21] Z. Shukur, E. Burke, and E. Foxley, "The Automatic Assessment of Formal Specification Coursework," *J. Computing in Higher Education,* vol. 11, no. 1, pp. 86-119, 1999.

[22] P. Thomas, K. Waugh, and N. Smith, "Using Patterns in the Automatic Marking of ER Diagrams," *Proc. 11th Ann. Conf. Innovation and Technology in Computer Science Education,* pp. 83-87, 2006.

[23] H. Ke, G. Zhang, and H. Yan, "Automatic Grading System on SQL Programming," *Proc. Int'l Conf. Scalable Computing and Comm.; Eighth Int'l Conf. Embedded Computing,* pp. 537-540, 2009.

[24] E. Gutiérrez, M.A. Trenas, J. Ramos, F. Corbera, and S. Romero, "A New Moodle Module Supporting Automatic Verification of VHDL-Based Assigments," *Computers and Education,* vol. 54, pp. 562-577, 2010.

[25] S. Isotani and L. de Oliverira Brandao, "An Algorithm for Automatic Checking of Exercises in a Dynamic Geometry System: iGeom," *Computers and Education,* vol. 51, pp. 1283-1303, 2008.

[26] V. Petridis, S. Kazarlis, and V.G. Kabrlasos, "ACES: An Interactive Software Platform for Self-Instruction and Self-Evaluation in Automatic Control Systems," *IEEE Trans. Education,* vol. 46, no. 2, pp. 102-110, Feb. 2003.

[27] Z. Doulgeri and T. Matiakis, "A Web Telerobotic System to Teach Industrial Robot Planning and Control," *IEEE Trans. Education,* vol. 29, no. 2, pp. 263-270, May 2006.

[28] L.C. Jacobs and C.I. Chase, *Developing and Using Tests Effectively: A Guide for Faculty.* Jossey-Bass, 1992.

[29] W.K. Chen and Y.C. Cheng, "Teaching Object-Oriented Programming Laboratory with Computer Game Programming," *IEEE Trans. Education,* vol. 50, no. 2, pp. 197-203, Aug. 2007.

[30] M.D. Koretsky, D. Amatore, C. Barnes, and S. Kimura, "Enhancement of Student Learning in Experimental Design Using a Virtual Laboratory," *IEEE Trans. Education,* vol. 51, no. 2, pp. 76-85, Feb. 2008.

[31] H. Delingette, X. Pennec, L. Soler, J. Marescaux, and N. Ayache, "Computational Models for Image-Guided Robot-Assisted and Simulated Medical Interventions," *Proc. IEEE,* vol. 94, no. 2, pp. 1678-1688, Sept. 2006.

[32] A. Johnson, T. Moher, Y.J. Cho, Y.J. Lin, D. Haas, and J. Kim, "Augmenting Elementary School Education with VR," *IEEE Computer Graphics and Applications,* vol. 22, no. 2, pp. 6-9, Mar./Apr. 2002.

[33] A.C. Graesser, P. Chipman, B.C. Haynes, and A.O.A., "AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue," *IEEE Trans. Education,* vol. 48, no. 2, pp. 612-618, Nov. 2005.

[34] P. Fournier-Viger, R. Nkambou, and A. Mayers, "Evaluating Spatial Representations and Skills in a Simulator-Based Tutoring System," *IEEE Trans. Learning Technologies,* vol. 1, no. 2, pp. 63-74, Jan.-Mar. 2008.

[35] I. Gustavsson, K. Nilsson, J. Zackrisson, J. Garcia-Zubia, U. Hernandez-Jayo, A. Nafalski, Z. Nedic, Ö. Goöl, J. Machotka, M.I. Pettersson, T. Lagö, and L. Håkansson, "On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories," *IEEE Trans. Learning Technologies,* vol. 2, no. 2, pp. 263-274, Oct.-Dec 2009.

[36] H.Q. Jiang, L. Zhang, and W.Q. Ye, "The Automatic Evaluation Strategies and Methods of Multimedia Work Assigments," *Proc. Int'l Conf. Computational Intelligence and Software Eng.,* pp. 1-5, 2009.

[37] T.J. Branoff, "Constraint-Based Modeling in the Engineering Graphics Curriculum: Laboratory Activities and Evaluation Strategies," *Proc. Midyear Conf. Eng. Design Graphics Division of the Am. Soc. for Eng. Education,* pp. 132-138, 2004.

[38] J.L. Colwell, J. Whittington, and J. Higley, "Assessment Measures and Outcomes for Computer Graphics Programs," *Eng. Design Graphics J.,* vol. 69, no. 3, pp. 24-33, 2005.

[39] D.H. Baxter and M.J. Guerci, "Automating an Introductory Computer Aided Design Course to Improve Student Evaluation," *Proc. Am. Soc. for Eng. Education Ann. Conf.,* 2003.

[40] D.H. Baxter, "Evaluating an Automatic Grading System for an Introductory Computer Aided Design Course," *Proc. 58th Ann. Eng. Design Graphics Midyear Meeting,* pp. 39-44, 2003.

[41] D.H. Baxter, "Evaluating Performance in a Freshman Graphics Course to Provide Early Intervention for Students with Visualization and/or Design Intent Difficulties," *Proc. Am. Soc. for Eng. Education Ann. Conf.,* 2002.

[42] Z. Shukur, Y. Away, and M.A. Dawari, "Computer-Aided Marking System for Engineering Drawing," *Proc. Soc. for Information Technology and Teacher Education Int'l Conf.,* pp. 1852-1857, 2004.

[43] D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung, "On Visual Similarity Based 3D Model Retrieval," *Computer Graphics Forum,* vol. 22, no. 2, pp. 223-232, Sept. 2003.

[44] A.R. Smith, "Color Gamut Transform Pairs," *Proc. Fifth Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 12-19, 1978.

Andrea Sanna graduated in electronic engineering in 1993 and received the PhD degree in computer engineering in 1997, both from Politecnico di Torino, Italy. Currently, he has an associate professor position at the First Faculty of Architecture of Politecnico di Torino, Italy. He has authored and coauthored several papers in the areas of computer graphics, virtual reality, distributed computing, and computational geometry. He is a senior member of ACM.

Fabrizio Lamberti graduated with a degree in computer engineering in 2000 and received the PhD degree in computer engineering in 2005 from Politecnico di Torino, Italy. Since 2006, he has been an assistant professor at Politecnico di Torino. He has published a number of technical papers in international journal and conferences in the areas of mobile and distributed computing, computer graphics, image processing, and visualization. He has served as a reviewer and program or organization committee member for several conferences. He is a member of the Editorial Advisory Board of international journals. He is a member of the IEEE and the IEEE Computer Society.

Gianluca Paravati received the BSc and MSc degrees in electronic engineering and the PhD degree in computer engineering from Politecnico di Torino, Italy, in 2005, 2007, and 2011, respectively. Currently, he is a research fellow with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy. His research interests include real-time image processing, collaborative virtual environments, human-machine interaction, remote visualization, and data distribution systems.

Claudio Demartini received the degrees in computer engineering and the PhD degree in software engineering from the Politecnico di Torino, Italy, in 1980 and 1987, respectively. He has published a number of technical papers in international journals and conferences in the areas of distributed computing. He is the vice dean of the Industrial Engineering and Management School at Politecnico di Torino. He is a member of the IEEE.