

# Automatic Grading of 3D Computer Animation Laboratory Assignments

Fabrizio Lamberti, *Senior Member, IEEE*, Andrea Sanna, Gianluca Paravati, and Gilles Carlevaris

**Abstract**—Assessment is a delicate task in the overall teaching process because it may require significant time and may be prone to subjectivity. Subjectivity is especially true for disciplines in which perceptual factors play a key role in the evaluation. In previous decades, computer-based assessment techniques were developed to address the above-mentioned challenges and to automatically grade students' work in a variety of educational fields. In this paper, the application of automatic assessment strategies in the unexplored domain of computer graphics is discussed. In particular, a tool that is designed to evaluate student assignments for a 3D computer animation course taught at Politecnico di Torino University is presented. During laboratory examination sessions, students are requested to individually operate on the open-source Blender suite and to recreate a 3D animation similar to a reference one. Student's output is assessed against a set of similarity indicators, which are specifically designed to capture the technical and perceptual factors that would be blended in a traditional teacher's evaluation. A comparison of the results that are achieved by the computer-based tool with grades assigned by visual inspection confirms the effectiveness of the designed approach and displays a high-quality concurrence between automatic and manual evaluations.

**Index Terms**—Computer-managed instruction, evaluation methodologies, computer graphics, 3D animation

## 1 INTRODUCTION

STUDIES on computer-based assessment are reported in the literature from the 1960s, when the use of an automatic system to grade punch card-based programming assignments was first demonstrated [1]. In the following decades, with the increase of class sizes and the diffusion of e-learning solutions, the above-mentioned paradigm has been tested in a growing number of domains. Interesting applications exist in many disciplines of computer science [2], [3], [4], [5], [6], [7], [8], [9] as well as in mathematics [10], physics [11], biology [12], languages and literature fields [13].

Often, a primary motivation for the development of automatic assessment techniques is saving time [14]. Teachers spend a significant amount of time in grading activities especially in educational areas where assessment relies on the frequent evaluation of periodic home or in-class assignments [15]. Hence, students could wait for several days before receiving their assignment evaluation, and this delay could have a negative impact on their learning achievements. Automatic grading solutions can provide students with immediate feedback. Another key reason is evaluation fairness. Scenarios exist (e.g., when a significant number of evaluations are performed in a short time) in which a comparable student's work could obtain a conflicting grade due to the adoption of different grading standards (e.g., when different teachers are involved), the use of inadequate (e.g., incomplete or too general) assessment criteria or, more

frequently, the difficulty of removing subjective factors in the evaluation [16], [17].

Early automatic assessment tools were used as multiple-choice test checkers [18]. With technological advancements, they evolved significantly and were integrated into comprehensive solutions designed to optimize the overall learning process [11], [19], [20], [21].

Indeed, the application of automatic assessment tools in scenarios requiring natural language processing (e.g., for the evaluation of essays) recently opened an intense debate on the opportunity for adopting computer-based solutions to address tasks at a high cognitive level [22], [23]. Nevertheless, the results reported in the literature confirm that current technology allows automatic graders to provide rather accurate and reliable results, especially in the computer science domain. However, a particular area of computer science, namely, computer graphics, has been partially tackled by recent developments in the field. Only a few solutions are reported in the literature for the assessment of multimedia content, sketches, engineering drawings (ED), computer-aided design (CAD) or computer graphics models [24], [25], [26], [27], [28].

By moving from the promising results reported in the above works, this paper presents an automatic tool for the assessment of 3D computer animation assignments. The designed tool replaces the former assessment strategy used to evaluate students' work for the Virtual Animation course at Politecnico di Torino university.

The approach pursued relies on the identification of a set of measurable indicators that are computed on the students' source files and can capture both the technical and the perceptual factors that characterized the previous visual inspection-based assessment method. Based on a linear regression analysis on the grades assigned by a panel of teachers, the above indicators are transformed into a grading formula that mimics human behavior. Each student is

• The authors are all with the Dipartimento di Automatica e Informatica at Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy. E-mail: {fabrizio.lamberti, andrea.sanna, gianluca.paravati, gilles.carlevaris}@polito.it.

Manuscript received 5 Sept. 2013; revised 25 June 2014; accepted 15 July 2014. Date of publication 17 July 2014; date of current version 12 Sept. 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TLT.2014.2340861

provided with the final grade, as well as with the values of the above indicators. Results that are obtained by assessing approximately 130 animations showed a high coherency between manually and automatically assigned grades.

The remainder of the paper is organized as follows. In Section 2, the background of the computer-based assessment is reviewed by focusing on the computer graphics field. The context, motivations and requirements for the proposed automatic grading tool are discussed in Section 3. The overall design and step-by-step implementation details are presented in Section 4. The experimental results are analyzed in Section 5. Lastly, Section 6 concludes the paper by outlining the future research directions.

## 2 BACKGROUND

Because of the intrinsic ability of computer systems to perform repeatable and quantifiable measurements, automatic grading solutions were considered, since from their introduction, as an excellent means to perform timely, justified and objective evaluations. The field with the widest diffusion of tools and prototypes for automatic assessment is computer science and, more specifically, computer programming [2]. Nonetheless, computer-based grading systems have been applied in many ways in this domain from the evaluation of basic IT skills [3] to automatic assessment in the areas of security [4], networking [5], control systems [6], formal logic [7], digital design [8], and databases [9].

In the computer science field, one discipline that has been scarcely addressed is computer graphics [29], [30]. A key reason is that the most common automatic assessment strategy, which consists in comparing students' work with a "specimen" (reference) solution [26], cannot be easily extended to graphics data produced in this domain.

Hence, only a few applications were developed to support the automatic grading of computer graphics students' works. In many cases, the solution adopted consisted of working only onto textual or numerical data available. For instance, in [31], the assessment of user interfaces developed in programming assignments is performed by analyzing the underlying description rather than the graphics output. In [32], students' 2D and 3D Matlab plots are compared to a reference solution by working with attributes recorded in the source files, such as coordinates, drawing line features, and axes' scale without actually displaying them. A comparable approach is pursued in [33] to assess presentations and word processing documents.

In several cases, ad-hoc tools are developed that collect easier-to-compare information [34]. For instance, solutions have been created to automatically assess geometry assignments [35] or design works based on formal notations, such as the unified modeling language (UML) [36] or the entity relation diagram (ERD) [37]. In these cases, students are provided with simplified diagramming environments that easily compute the number of correct knots (e.g., model blocks), linkages (i.e., relationships among blocks) and related properties. A comparable approach is applied in [26] and [27] to assess ED and CAD works. Grades are assigned based on strict analytical rubrics [30] that consider the number of objects found in the student's solution, the accuracy of their geometry, the layer they have been placed

upon, etc. Because different procedures can lead to the same design, an automatic assessment could be complemented by visual inspection as shown in [26] and [38]. Another interesting research field is represented by the evaluation of sketches. In [25], computer-based worksheets allow students to draw glyphs (i.e., polylines) and the relations among them by also adding conceptual labels. Next, worksheets can be assessed by checking both qualitative and quantitative constraints (e.g., on spatial and conceptual relations as well as on "ink" usage).

Even when the goal is to evaluate drawing and animation abilities (e.g., with tools such as Adobe's Photoshop and Flash [24]) and the importance of the artistic and creative levels (rather than the technique and content) is stressed, the grading metrics proposed rely on the procedural aspects. Thus, for instance, drawings are assessed by considering the size of the fonts used, the application of a given filter, the color of a particular region, etc. Similarly, animations are graded by checking the presence of a key frame, the use of a layer, the change in an object's size, etc.

However, scenarios exist in which procedure/content-related aspects and errors alone may not allow an automatic system to perform a correct evaluation (e.g., because the same aesthetic result could be obtained in many different ways, because the teacher's assessment could incorporate perceptual components, or because the analytic rubrics could not be easily defined) [39]. It is the case, for instance, of artistic disciplines, such as singing [40], dancing [41], and hand drawing [42]. As shown in [28], it is also the case of 3D modeling and rendering, where an assessment cannot be based solely on criteria such as the number and configuration of vertices used to create a mesh. In fact, a modeled objects' appearance could depend not only on the complexity of its geometry but also on the materials used, the lights' set, and the textures added. Hence, the difference between the result achieved by the student and the expected one could be determined only by analyzing, either manually or automatically, the rendered output.

Based on the evident analogies, considerations for an assessment of the 3D modeled contents can be easily extended to 3D animations. In this case, the challenge is determining the right combination of measurable indicators or features that act as a reliable proxy for both the procedural/technical and perceptual criteria that are blended in a teacher's assessment behavior [40], [43].

The problems of comparing the output produced by a computer animation rendering process are very similar to those faced in the field of content-based video retrieval (CBVR) [44], [45]. In CBVR systems, video frames are described using feature vectors. Features may refer to visual characteristics of the frames and contained objects (e.g., color, texture, and shape). Additionally, temporal features can be used (e.g., related to camera motion and the objects' trajectories). Next, retrieval is performed using various similarity measures, which are obtained by computing in the feature vectors space different distance functions that are intended to mimic human perception [46].

Some of the characteristics of the CBVR techniques may not be particularly suited for the particular goal tackled by this paper. In fact, being designed to be used over large datasets, they often implement optimization

strategies that make the extraction of fine-grained details of possible interest in the assessment process difficult to extract [47]. Moreover, only displacements in the 2D space are generally considered (e.g., obtained from MPEG motion vectors). Hence, rotations and part deformations, which are very common in computer-generated animations, would remain unaddressed [48]. Although future advancements in this field might reduce the impact of the above limitations, CBVR solutions could hardly be used alone for assessing procedural aspects, since they could evaluate the differences in the results achieved, but not in the techniques used. For instance, they could not identify situations where a student produces an animation that looks very similar to the reference one by using a method different than the one requested by the teacher. Based on these considerations, a strategy to address the above issues has been designed.

### 3 CONTEXT AND MOTIVATIONS

The Virtual Animation course that is considered in this work completes the computer graphics path for students enrolled in the Product Design curriculum of the Design and Visual Communication bachelor's degree program at Politecnico di Torino university. The course is attended by approximately 100 students each year, and it is blended into the so-called Laboratory on the Design for Industrialization. The Laboratory includes two other disciplines, namely Design for Serial Production and Mechanics for the Design. The Virtual Animation course aims to provide students with the competencies required for representing product features in the design, manufacturing, assembly and functioning phases, for studying usage scenarios and ergonomic aspects, and for analyzing dynamic constraints on mechanical parts.

Each year, a design theme is chosen for the Laboratory and is summarized in a design brief. Students are asked to work in groups and to develop product concepts by providing a graphical representation for them using 3D animation techniques. In particular, each group is asked to develop two animations: one that is more technical and tailored, e.g., to a digital catalogue, and one that is more concrete and shows the degree of usability under realistic conditions. A combined assessment is planned for the entire Laboratory. Teachers from individual disciplines assign a grade for the activities pertaining to their own subjects. Thus, the grade for the Virtual Animation course is based only on the quality of the animations developed and does not consider aspects pertaining to design or mechanical choices.

Unfortunately, the grading strategy adopted so far for this course is prone to subjectivity. In fact, groups choose the animation techniques that are the most effective for their particular work, which causes difficulty when comparing the final results. Moreover, to assign individual grades, the specific contribution of each student to the group's work can only be determined by asking ad-hoc questions on the content of the files submitted. Because of the significant burden associated with this evaluation step, the large number of students and the highly variable conditions (e.g., from a student, a group, or a yearly theme to another), objectiveness represents a major concern.

Starting with the 2012-2013 academic year, the grading strategy for the course was changed by adding an individual animation assignment prior to the end of the course. The goal was to improve the previous evaluation method by providing teachers with a means to selectively and promptly assess the students against objective criteria without increasing their workload. The assignment was designed by considering the following requirements:

- it should require students to operate directly on the animation tool adopted in the course (i.e., Blender);
- it should enable the evaluation of a precise set of basic animation techniques (advanced abilities are mobilized in the development of group animations);
- it should focus also on qualitative performance indicators, not just on quantitative and procedural criteria;
- it should be assessed automatically by a system that generates immediate and fair feedback.

The solution developed consists of an integrated package, which can be downloaded by each student on a personal laptop or on one of the PCs in the laboratory. The package includes several videos showing a 3D animation rendered from the front, side, top and user camera views as well as a Blender file with all of the objects used in the animation (in a pre-defined configuration). The task for the students is to focus on the output expected and recreate an animation similar to the reference one by working with the objects provided and properly applying methods learned in theoretical lessons, e.g., for setting key frames, configuring poses with direct/inverse kinematics, creating actions, etc. After a reasonable amount of time (from half an hour to one hour based on the complexity of the reference animation), students are asked to submit their work. A computer-based assessment tool has been designed to extract relevant information from the source files submitted and to automatically compute a similarity score on a 1-16 scale. Similarity score is combined with the result of the assessment on the group animations, which are evaluated on the same scale. Next, a final grade in the range of 1 to 30 cum laude (corresponding to 32) is provided to each student.

## 4 THE PROPOSED TOOL

In this section, the design of the proposed tool is presented. In Section 4.1, the exam procedure is described. In Section 4.2, the approach devised to measure the similarity between the animations is analyzed. Lastly in Section 4.3, the manner in which the above indicators have been used in the implementation of the overall assessment method is illustrated along with the experimental observations.

### 4.1 Exam Procedure

Prior to the end of the course, teachers assemble the examination package by generating several 3D models and by producing a 10- to 15-second animation with them. Basic animation tools are used: transformations, path parenting, keyframing, constraining, armature deformation, and posing. The animation is rendered from four typical points of view and five videos are produced: one for each view and one with all four views. Videos are included in the package



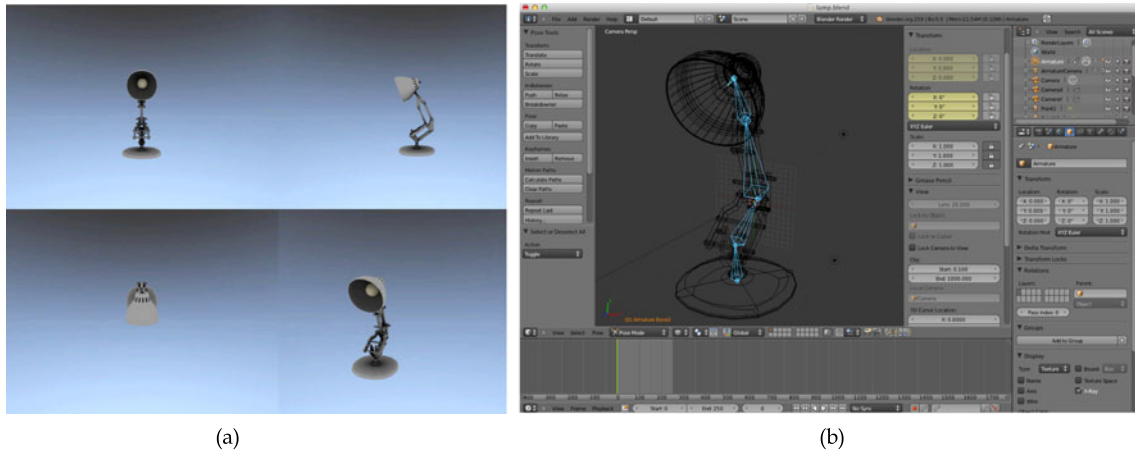


Fig. 1. Content of the examination package for the lamp sequence: (a) reference video incorporating all of the views, and (b) the Blender file.

along with the Blender file for the exam (with all of the 3D objects in a pre-defined state). Although the videos could contain more than one animated object, during the exam students will be requested to work only on one object (the main character with its own armature). An example is the *lamp* animation,<sup>1</sup> which is illustrated in Fig. 1.

The lamp character is placed in the center of the scene. A green ball enters the scene from the right side. The lamp turns to look at the ball and then jumps to the right. Several representative frames from the user camera view are reported in Fig. 2. In this case, the source file distributed to the students included the rigged lamp with its five-bone armature in the rest pose as well as the key frames that control the animated ball (as a time reference).

During the exam, students can determine whether to play the videos in a loop, advance them step-by-step, or even use them as a background in the Blender scene. They can check the renderings from the various views while developing their own animation with the more appropriate techniques to ensure that the work produced matches the videos provided. Once satisfied or when time for the exam expires, students can submit a copy of their work using a form that has been embedded into the Blender interface. The system is configured to avoid double submissions and to inform the students about the status of the upload.

## 4.2 Measuring Similarity Between Animations

The goal of the students during the laboratory exam is to reproduce an animation as close as possible to the reference file. Hence, the assessment was aimed at measuring the similarity between the two animations rather than evaluating the quality of the student's work according to absolute criteria [49] (e.g., about the application of generally agreed upon principles for effective animation [50]).

Thus, the identification of indicators that could best approximate the concept of similarity in terms of human perception while limiting the contribution of subjective factors was necessary. This task required to work in a controlled scenario, which was defined by considering the *fish*<sup>2</sup> sequence in Fig. 3 and by producing a number of

animations with ad-hoc errors embedded.<sup>3</sup> In the reference animation, the fish character moves over a circular path, while a hook (used for synchronization, such as the ball in the *lamp* sequence) enters the water. The armature controlling the fish's motion is composed of nine bones with each experiencing a rhythmic movement.

At first, four other animations were created, including *fishGood*, *fishMedium*, *fishWrong* and *fishVeryWrong*. In the *fishGood* animation, small errors were introduced on the bones' angles and on the character's global position. In *fishMedium*, parts of the fins were blocked to produce a more evident error. The *fishWrong* animation was built upon the *fishMedium* one by embedding additional errors into the fish's position and orientation. Lastly, in the *fishVeryWrong* animation, the character was partially blocked and placed in an extremely wrong position.

Based on previous studies of human motion [51], the similarity between generic character animations is related to the relative position and orientation of the moving parts. Hence, a preliminary similarity measure was computed between the reference and each of the four controlled animations by applying the metric used in [41] and [52] to compare the human character poses in a dance and in a martial arts training application, respectively. The similarity was computed by averaging over the whole animation the mean euclidean distance  $d$  (in 3D absolute Blender units) between the position of each joint in the two animation files as

$$S_{JP} = \frac{1}{F} \sum_{f=1}^F \sum_{j=1}^J d(P_j^f, \dot{P}_j^f) / (J \cdot F). \quad (1)$$

In (1),  $J$  is the number of joints in the character's armature,  $F$  is the number of frames, whereas  $P_j^f$  and  $\dot{P}_j^f$  are the position of joint  $j$  ( $1 \leq j \leq J$ ) at frame  $f$  ( $1 \leq f \leq F$ ) in the student and teacher animations, respectively.

The results for the four animations are reported in the upper portion of Table 1 (second column). Despite the evident discrimination power of the joints' position-based indicator  $S_{JP}$ , scenarios exist in which trivial animation errors are overestimated and provide results that are extremely

1. <http://youtu.be/hDqkBEIshOc>

2. [http://youtu.be/xiizA\\_KISVI](http://youtu.be/xiizA_KISVI)

3. <https://www.youtube.com/channel/UC2jzoqFzP9Stc6gsYCaTA9A/videos>

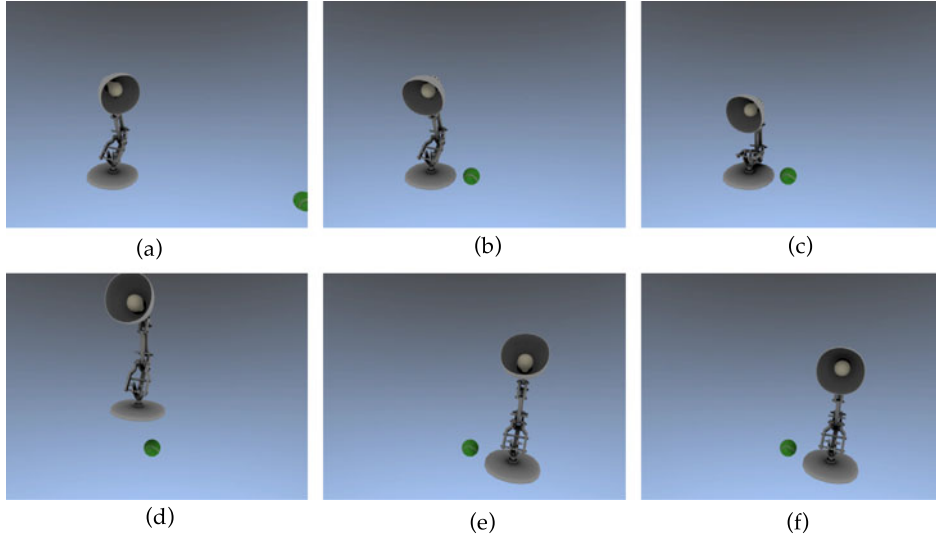


Fig. 2. Representative shots for the *lamp* animation rendered from user camera at frame (a) 30, (b) 120, (c) 150, (d) 170, (e) 200 and (f) 250.

different from those obtained by a human observer. To display this undesirable behavior, an animation named *fishGoodStopped* was added to the set, where the swim motion is exactly the same as *fishGood*, but the character remains fixed in the initial position. The value of  $S_{JP}$  is reported in the last row of Table 1. Although most of the frames are correct, the indicator strongly penalizes the position error.

A possible way to address the scenarios similar to the one above could be to measure the similarity as the average difference between the bones' orientations (relative to the root bone), namely,

$$S_{BO} = \sum_{f=1}^F \sum_{b=1}^B |\theta_b^f - \dot{\theta}_b^f| / (B \cdot F), \quad (2)$$

where  $B$  is the number of bones in the armature, whereas  $\theta_b^f$  and  $\dot{\theta}_b^f$  are the orientations of bone  $b$  ( $1 \leq b \leq B$ ) at frame  $f$  in the student and teacher animations.

The results obtained for the set of animations with the  $S_{BO}$  indicator are provided in the last column of Table 1. With this indicator, the correct elements of the *fishGoodStopped* animation are properly acknowledged. The use of bones' orientations was proposed in [41] and [53] as an alternative to the joints' positions. However, Table 1 shows that in the scenario considered in this paper (where animations could be very heterogeneous), the orientation should be used in combination with position rather than as an alternative only.

Because of the implicit dependency between the angles defining a bone and the corresponding joints' positions, the  $S_{JP}$  and  $S_{BO}$  indicators cannot simply be used together. The solution consisted of selecting only one metric (the second one), and adding two other indicators linked to the global position/orientation of the armature's root joint/bone (thus, recovering the contribution blended in the first one):

$$S_{GP} = \sum_{f=1}^F d(P_r^f, \dot{P}_r^f) / F, \quad (3)$$

$$S_{GO} = \sum_{f=1}^F |\theta_r^f - \dot{\theta}_r^f| / F. \quad (4)$$

In (3),  $P_r^f$  and  $\dot{P}_r^f$  are the positions of the root joint at frame  $f$  in the student and teacher animations, respectively. In (4),  $\theta_r^f$  and  $\dot{\theta}_r^f$  are the orientations of the root bone.

A preliminary comparison with teacher's behavior confirmed that with a combination of  $S_{BO}$ ,  $S_{GP}$  and  $S_{GO}$ , an estimate of the overall similarity that was closer to human perception could be obtained. Nonetheless, the human observer could identify more subtle features that better characterize the distance between the animations, such as the movements fluidity, the presence of glitches, and the motion velocity. Hence, indicators that measure the differences in velocity and acceleration for the global position and orientation ( $S'_{GP}$ ,  $S'_{GO}$ ,  $S''_{GP}$  and  $S''_{GO}$ ) as well as for the individual bones' orientations ( $S'_{BO}$  and  $S''_{BO}$ ) were

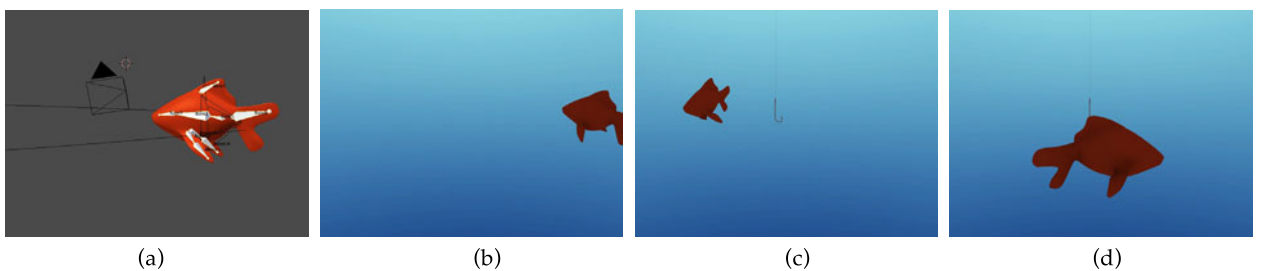


Fig. 3. The reference *fish* scene: (a) Blender file and representative shots rendered from user camera at frame (b) 40, (c) 150 and (d) 230.

TABLE 1  
Similarity Measures Based on Averaged Distances between  
Joints' Positions (in Blender Units) and between Bones'  
Orientations (in Degrees)

Animation compared to <i>fish</i>	Joints' positions similarity ( $S_{JP}$ )	Bones' orientations similarity ( $S_{BO}$ )
<i>fishGood</i>	0.31	1.50
<i>fishMedium</i>	1.01	3.90
<i>fishWrong</i>	2.77	5.47
<i>fishVeryWrong</i>	6.69	11.14
<i>fishGoodStopped</i>	29.81	1.50

added. The indicators above were tested on an extended set of animations, including the sequences in Table 1 and the new ones generated by introducing other errors in the reference animation. In the newly added *fishStopped* sequence, the character performs the same swim motion of the fish animation without moving from the initial position. In *fishGap*, a small constant difference in the character's position is added with respect to the fish animation, whereas in *fishBlocked*, the character correctly moves along the path but does not swim. In *fishGlitch*, a glitch has been introduced in the swim motion, which reduces the overall smoothness of the animation. In *fishStretched*, the character has a swim cycle that is two times slower than in *fish*. Finally, in *fishFaster*, the model moves over the path at twice the speed of *fish*, whereas in *fishOpposite*, the swim motion is reversed. The similarity scores computed for all of the indicators on the extended set of animations are reported in Table 2.

For instance, *fishStopped* is considered equivalent to *fish* based on the metric for individual bones' orientations. However, the global position and orientation-based indicators immediately outline the error. Similarly, in *fishGap* the error is detected only by considering the global position. As expected, indicators based on global position and orientation in *fishBlocked* exhibit no difference in the movement along the path, while angle errors in the swim motion are detected by the metric utilizing the individual bones' orientations. Velocity and acceleration information are particularly effective in numerically outlining errors that affect

only a few frames but are particularly important (i.e., they would be immediately detected by a human observer) as in the case of the *fishGlitch* animation.

According to the results obtained, the human similarity measures were less sensitive to slight but continuous changes in motion timing than the automatic ones. In fact, small errors in the positions and orientations at every frame could be difficult to perceive with a teacher's eyes but are accumulated by the computer-based metric and could lead to an extremely low similarity score (as seen in last three animations in Table 2). A technique commonly used to find an alignment between two time series known as dynamic time warping (DTW) [56] (used in multimedia retrieval and gesture recognition) was adopted to incorporate this perceptual behavior in the devised metric. The results obtained by applying DTW to the selected indicators are reported in Table 3 (indicated by ^).

By comparing the similarity scores for the *fishStretched*, *fishFaster* and *fishOpposite* sequences in Tables 2 and 3, the DTW-based indicators are far less sensitive to those errors that would almost be undetected by a human observer (because of their nature, they should not contribute negatively to the final grade).

In order to use the above set of indicators to automatically assess animations that are developed by students in different exam sessions, a normalization is required. Thus, global position values are divided by the overall character's displacement, whereas corresponding velocities and accelerations are divided by their maximum values in the reference animation.

Additionally, a comparable normalization factor could be applied to orientation-based indicators (e.g., by considering the angular range a single bone can move into). Thus, a significant error made on a bone undergoing large rotations in the reference animation would be weighted less than a small error on a bone with little or no changes in orientation. Other aspects could be considered as well, such as the bone's size and the number of child bones inheriting a parent's rotation. However, the strategies above would disregard perceptual aspects, which could be canceled by the normalization. According to [55] and [56], in these scenarios a perceptual weight should be applied to each bone, which

TABLE 2  
Similarity Measures Based on Averaged Distances between Global Positions and Orientations, and between  
Individual Bones' Orientations

Animations compared with <i>fish</i>	Global position			Global orientation			Bones' orientations		
	$S_{GP}$	$S'_{GP}$	$S''_{GP}$	$S_{GO}$	$S'_{GO}$	$S''_{GO}$	$S_{BO}$	$S'_{BO}$	$S''_{BO}$
<i>fishGood</i>	0.79	0.01	0.01	0.00	0.00	0.00	1.50	2.74	0.32
<i>fishMedium</i>	2.13	0.03	0.01	0.00	0.00	0.00	3.90	2.98	0.49
<i>fishWrong</i>	5.41	0.07	0.01	8.86	0.07	0.13	5.47	2.90	0.72
<i>fishVeryWrong</i>	3.17	0.03	0.01	66.46	0.43	0.13	11.14	4.00	1.42
<i>fishGoodStopped</i>	69.10	0.91	0.02	84.60	0.72	0.09	1.50	2.74	0.32
<i>fishStopped</i>	69.73	2.12	0.02	84.55	3.11	0.09	0.00	0.00	0.00
<i>fishGap</i>	20.97	0.95	0.02	0.00	0.00	0.00	0.00	0.00	0.00
<i>fishBlocked</i>	0.00	0.00	0.00	0.00	0.00	0.00	12.97	5.67	1.55
<i>fishGlitch</i>	0.00	0.00	0.00	0.00	0.00	0.00	12.27	8.40	4.17
<i>fishStretched</i>	0.00	0.00	0.00	0.00	0.00	0.00	13.13	5.28	1.48
<i>fishFaster</i>	59.72	1.78	0.05	53.41	4.92	0.13	0.00	0.00	0.00
<i>fishOpposite</i>	0.00	0.00	0.00	0.00	0.00	0.00	11.91	5.37	1.23

TABLE 3  
Similarity Measures Based on Averaged Distances between Global Positions and Orientations, and between Individual Bones' Orientations Obtained by Applying Dynamic Time Warping (Indicated by the  $\hat{\cdot}$  Symbol)

Animations compared to <i>fish</i>	Global position			Global orientation			Bones' orientations		
	$\hat{S}_{GP}$	$\hat{S}'_{GP}$	$\hat{S}''_{GP}$	$\hat{S}_{GO}$	$\hat{S}'_{GO}$	$\hat{S}''_{GO}$	$\hat{S}_{BO}$	$\hat{S}'_{BO}$	$\hat{S}''_{BO}$
<i>fishGood</i>	0.13	0.00	0.01	0.00	0.00	0.00	1.42	0.52	0.30
<i>fishMedium</i>	0.33	0.01	0.01	0.00	0.00	0.00	3.56	1.04	0.46
<i>fishWrong</i>	1.44	0.02	0.01	3.34	0.03	0.05	4.62	1.58	0.70
<i>fishVeryWrong</i>	2.31	0.01	0.01	66.46	0.26	0.06	11.06	3.82	1.42
<i>fishGoodStopped</i>	69.10	0.91	0.02	84.60	0.72	0.09	1.42	0.52	0.30
<i>fishStopped</i>	69.73	2.12	0.02	84.55	3.11	0.09	0.00	0.00	0.00
<i>fishGap</i>	4.77	0.29	0.01	0.00	0.00	0.00	0.00	0.00	0.00
<i>fishBlocked</i>	0.00	0.00	0.00	0.00	0.00	0.00	12.97	5.67	1.55
<i>fishGlitch</i>	0.00	0.00	0.00	0.00	0.00	0.00	3.36	3.19	2.90
<i>fishStretched</i>	0.00	0.00	0.00	0.00	0.00	0.00	5.82	3.47	1.24
<i>fishFaster</i>	0.45	0.17	0.04	0.34	0.15	0.07	0.00	0.00	0.00
<i>fishOpposite</i>	0.00	0.00	0.00	0.00	0.00	0.00	0.87	0.61	0.17

would be obtained by subjective studies. Although the above approach has been successfully used in animations with human characters, it could not be applied to the considered context because a setup step would be required to configure the normalization for each specific model (i.e., at each exam session).

## 5 EVALUATION AND DISCUSSION

Normalized similarity indicators illustrated in the previous section were used in the development of an automatic assessment tool that can process student Blender files and compute a grade similar to the one that would be assigned by teachers.

To train the system, the *fish* and *lamp* animations presented above were used because they incorporate all of the various features to be assessed. A group of 80 students were asked to produce their own animations by following the same procedure devised for the exam. A training set that contained 40 *fish* and 40 *lamp* animations was created. The animations in the training set were assessed separately by a panel of three course teachers, and each teacher assigned a similarity score in the range of 1 (poorly similar) to 5 (very similar). The inter-rater agreement of the similarity scores assigned by the three teachers was measured by Cohen's  $\kappa$  using linear weights. Results are reported in Table 4

(columns two through four). For each student's animation, an average score was computed by rounding values every 0.25 step. Thus, an equivalent 16-step evaluation scale was obtained, which ensured an adequate discrimination power. In fact, while the 1-5 scale would be extremely coarse grained, the 16-step scale has a resolution greater than the one for an individual teacher and can be directly matched to the scale used for the final grade.

Starting with the averaged teachers' scores above, an overall formula for controlling the automatic assessment was obtained by multiple linear regression (i.e., by finding the weighted combination of the nine similarity indicators that provide the best fit). To verify the accuracy that could be achieved with the weights found, the formula was first applied to assess the animations in the training set. By following the approach in [13], an estimate by the automatic tool was considered *correct* if the difference with respect to the average human grade was in the range of  $\pm 0.25$ . An estimate was considered *average* if the difference was in the range of  $\pm 0.50$ . In all of the other cases, the estimate was marked as *wrong*. The results are shown in the upper part of Table 4. By considering the training set as a whole, the percentage of *correct* estimates was 63.75 percent, whereas the percentage of *average* estimates was 92.50 percent. The mean error was 0.23, which corresponds to 5.75 percent on the 16-step evaluation scale.

TABLE 4  
Comparison of Assessment Results by a Panel of Teachers and by the Automatic Tool

Anim. considered	$\kappa$ A-B	$\kappa$ B-C	$\kappa$ C-A	Percentages of <i>correct</i> est.	Percentages of <i>average</i> est.	Mean Error (%)	Standard deviation	$\kappa$ Man.-Aut.
<i>fish</i> (train.)	0.420	0.473	0.458	52.50%	85.00%	0.30 (7.50%)	0.22	0.710
<i>lamp</i> (train.)	0.666	0.585	0.690	75.00%	100.00%	0.16 (4.00%)	0.12	0.920
<i>fish+lamp</i> (train.)	0.583	0.550	0.596	63.75%	92.50%	0.23 (5.75%)	0.19	0.855
<i>fish</i>	0.612	0.424	0.481	52.50%	90.00%	0.26 (6.50%)	0.18	0.700
<i>lamp</i>	0.530	0.564	0.699	72.50%	92.50%	0.19 (4.75%)	0.20	0.858
<i>fish+lamp</i>	0.582	0.526	0.616	62.50%	91.25%	0.23 (5.75%)	0.19	0.817
<i>seagull</i>	0.638	0.62	0.473	63.83%	85.11%	0.25 (6.25%)	0.21	0.834
<i>fish+lamp+seagull</i>	0.636	0.594	0.598	62.99%	88.98%	0.24 (6.00%)	0.20	0.838



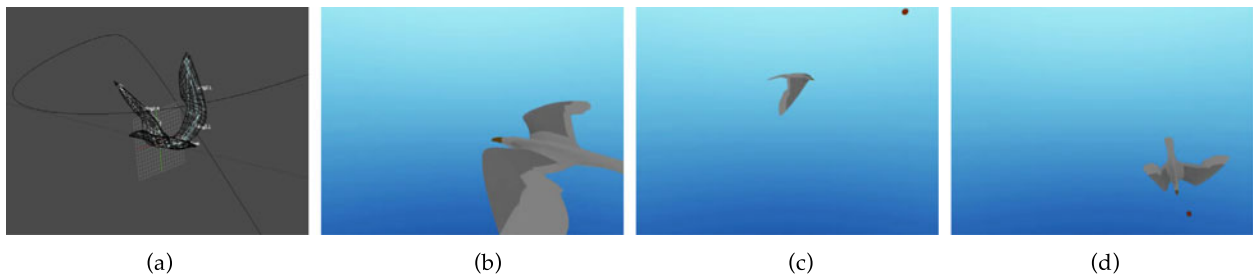


Fig. 4. The reference *seagull* scene: (a) Blender file and representative shots rendered from a user camera at frame (b) 40, (c) 190 and (d) 230.

The devised formula was then applied to assess the animations that were not included in the training set. Thus, the remaining 40 *fish* and 40 *lamp* students' animations were considered. Moreover, another group of 47 students was asked to work on a new animation referred to as *seagull*,<sup>4</sup> which was included in the examination package illustrated in Fig. 4. In this case, students had to force the character to follow a path (circular, and then quickly descending) by rotating and translating it as well as by applying some type of constraints. Then, a number of parameters were set and key frames were added to make the character move along the path at the right speed. Finally, keyframing was used again to pose the character and to define the periodic flying motion. Students could use the falling ball animation included in the examination package to set up the timing. Teachers assessed students' *seagull* animations, and the average scores were obtained.

Automatic grades were then computed on all animations by using the configuration for the training set. The results obtained are reported in the bottom portion of Table 4. The percentage of *correct* and *average* estimates remains comparable with those obtained with the training set, thus confirming the validity of the approach. For instance, the automatic tool provided 52.50, 72.50 and 63.83 percent of *correct* estimates and 90.00, 92.50 and 85.11 percent of *average* estimates on the *fish*, *lamp* and *seagull* animations, respectively. The agreement between the manually and the automatically assigned grades is shown by Cohen's  $\kappa$  coefficients reported in the last column of Table 4.

Nonetheless, in some cases, a difference between the automatic and the human grades greater than  $\pm 0.50$  was observed. A possible reason could be that, in the proposed assessment method, the emulation of teachers' behavior has been obtained by replicating the perceptual factors that define, in human terms, the concept of similarity between computer animations. Despite the key role played by such aspects, they could introduce subjectivity into the evaluation. For instance, an observer would generally evaluate the quality of a character's animation by checking its naturalness and plausibility, i.e., its consistency with known behaviors, with the laws of physics, etc. [57], [58]. However, two observers could quantify a dissimilarity based on the above factors in two different ways.

To determine the effect, Table 5 could be considered, which compares automatically and manually assigned scores for several representative students' animations. In the last three columns, the error percentages for the global

position and orientation as well as for the individual bones' orientations are reported.

In the first two rows, the *lamp1*<sup>5</sup> and *lamp2*<sup>6</sup> animations are depicted, which received nearly the same evaluation by the computer-based system (4.18 and 4.20) but obtained significantly different scores by the teachers (4.38 and 3.66). In this case, errors made by the students are rather small from a numerical point of view, whereas the manual evaluation of the second animation is negatively influenced by the particularly unnatural timing of the character before the jump. An additional animation with distant scores is *lamp3*,<sup>7</sup> which was rated 2.84 by the teachers and 3.36 by the automatic tool. The difference occurs during the jump, since the lamp moves as if no gravity is present. A human observer penalizes this effect, even though the motion is very similar to the reference.

An opposite scenario can be observed in *fish1*,<sup>8</sup> where the averaged teachers' score is higher than the automatically assigned one. By viewing the video, the character appears to be well animated. However, many imperceptible errors exist on the bones' orientations. An additional case is represented by *fish2*,<sup>9</sup> which shows a risk for a human observer to focus only on some parts of the animation. In this case, the fish is correctly rotating around the hook, but it does not swim. Teachers did not properly consider the fact that, despite the evident motion error, the fish is following the path throughout the whole animation perfectly.

In *seagull1*,<sup>10</sup> only small errors are present in the seagull's flight cycle and in the global position and velocity of the character. However, the timing of the last portion of the sequence is totally wrong because the seagull does not accelerate to follow the falling ball. A human observer penalizes the incorrect timing significantly, while the automatic system manages the situation in a more balanced way by considering it with the proper weight in the context of the overall animation. Lastly, in *seagull2*,<sup>11</sup> the scenario is similar to *fish1*. The character's animation is affected by several position errors (seagull's motion is significantly anticipated) that cannot be appreciated by the human observer because of the lack of known references.

The examples considered above show that in the approach adopted, the indicators chosen captured the

4. <http://youtu.be/36rpF2Te-Ic>

5. <http://youtu.be/kKTIwwWelXE>

6. <http://youtu.be/Vwvh6Pkt9tw>

7. <http://youtu.be/sh1T-wsBJu4>

8. [http://youtu.be/-ZS\\_NX78W2U](http://youtu.be/-ZS_NX78W2U)

9. <http://youtu.be/ENzVv14WcyI>

10. <http://youtu.be/5be3zztUnNg>

11. <http://youtu.be/rw7a-QIsJhE>



TABLE 5  
The Effect of Subjectivity on Perceptual Aspects  
for the Similarity-Based Evaluation

Anim. considered	Teach. score	Aut. tool score	Global pos. error	Global or. error	Bones' or. error
<i>lamp1</i>	4.38	4.18	2.70%	0.01%	1.00%
<i>lamp2</i>	3.66	4.20	2.40%	0.01%	0.96%
<i>lamp3</i>	2.84	3.36	5.13%	0.00%	2.16%
<i>fish1</i>	4.95	4.13	0.20%	0.01%	5.77%
<i>fish2</i>	2.84	3.35	0.20%	0.01%	10.2%
<i>seagull1</i>	2.75	3.59	4.20%	0.01%	2.86%
<i>seagull2</i>	3.50	2.61	8.80%	0.30%	4.30%

perceptual factors in the teachers' similarity measures while effectively smoothing the contribution of the subjective aspects at the same time.

## 6 CONCLUSIONS

In this paper, an automatic tool that supports the assessment of 3D animation assignments of a Virtual Animation course is presented. An assessment is performed by comparing the animations developed by individual students with a teacher's reference one. A comparison leads to a similarity score that considers a set of numerical indicators designed to replicate both the technical and perceptual factors that are blended in a visual inspection-based assessment. The similarity score is translated into a grade that is expressed on a 16-step scale. The grade is notified to each student together with the values of the above indicators.

The experimental observations confirm the accuracy of the designed approach. By considering the aggregate results, in 62.99 percent of the cases the automatically computed grade is the same assigned by the teachers. In 88.98 percent of the cases, the grade is only one step below or above the teachers' one. In addition, the methodology used for tuning the grading metric succeeds in limiting the impact of subjectivity in the teachers' evaluation.

Currently, the system has been used to complement the assessment for the students attending the 2012-2013 edition of the course, which provided students with fair, detailed and immediate feedback. The plan is to continue the testing in the 2013-2014 edition by enriching the training set and by further tuning the grading metric. In this context, the opportunity to consider also the similarity scores obtained by crowdsourcing or CBVR approaches and to compare with them will be analyzed. Moreover, strategies for embedding the evaluation procedure directly into the downloadable Blender package will be explored, with the aim to provide students with an effective tool for self-assessment and exam training. Lastly, in order to improve the effectiveness of the feedback and provide the students with a further help in identifying and correcting their errors, the possibility to automatically generate student-specific videos showing in a graphics way the reasons for the value of a given indicator will be considered. For instance, such videos could compare a student's animation<sup>12</sup> with a reference one in terms of global position and orientation error,<sup>13</sup> as well as of

individual bones' orientation error<sup>14</sup> (hiding the effect of the error on the character and showing directly joints and sticky bones of the two animations overlapped).

In the course studied for this paper, the tool is specifically used to assess basic animation skills, but the devised methodology could be easily utilized to cover an extended set of learning outcomes. Thus, future work may consider animations obtained by deforming the mesh or by physics-based techniques, which can be assessed by integrating similarity indicators based on the cloud of vertices of the reference model. Moreover, because the methodology defined is general, animation tools other than Blender could be considered. Thus, the proposed tool could be used in a wider set of educational scenarios.

## REFERENCES

- [1] J. Hollingsworth, "Automatic graders for programming classes," *Commun. ACM*, vol. 3, no. 10, pp. 528–529, Oct. 1960.
- [2] P. Ithantola, T. Ahoniemi, V. Karavirta, and O. Seppala, "Review of recent systems for automatic assessment of programming assignments," in *Proc. 10th Int. Conf. Comput. Edu. Res.*, 2010, pp. 86–93.
- [3] T. Lahtonen and V. Isomottonen, "PARSI: A tool for automatic assessment of office documents and basic it skills," *Proc. 11th Int. Conf. Comput. Edu. Res.*, 2012, pp. 174–180.
- [4] G. T. Ferreres, K. Wouters, B. Ramos Alvarez, and A. R. Garnacho, "EVAWEB: A web-based assessment system to learn X.509/PKIX-based digital signatures," *IEEE Trans. Edu.*, vol. 50, no. 2, pp. 112–117, May 2007.
- [5] A. Robles-Gomez, L. Tobarra, S. Ros, R. Hernadez, A. C. Caminero, and R. Pastor, "Automatic assessment for the e-learning of the network services in the context of the EHEA," in *Proc. Frontiers Edu. Conf.*, 2011, pp. 1–6.
- [6] V. Petridis, S. Kazarlis, and V. G. Kaburlasos, "ACES: An interactive software platform for self-instruction and self-evaluation in automatic control systems," *IEEE Trans. Edu.*, vol. 46, no. 1, pp. 102–110, Feb. 2003.
- [7] G. Molnar, V. Omrcen, and M. Cupic, "Automating the formal logic course," in *Proc. 35th Int. Convention MIPRO*, 2012, pp. 1124–1129.
- [8] Z. Stanisavljevic, B. Nikolic, and J. Djordjevic, "A module for automatic assessment and verification of student' work in digital logic design," in *Proc. 19th Int. Conf. Workshops Eng. Comput.-Based Syst.*, 2012, pp. 275–282.
- [9] J. Z. Kovacic and J. S. Green, "Automatic grading of spreadsheet and database skills," *J. Inf. Technol. Edu.: Innovations Practice*, vol. 11, pp. 53–70, 2012.
- [10] A. Rasila, M. Harjula, and K. Zenger, "Automatic assessment of mathematics exercises: Experiences and future prospects," in *Proc. Symp. Eng. Educ.*, 2007, pp. 70–80.
- [11] R. E. Stanley, "Using web based animation software with algorithmic parameters in order to simplify grading while still maintaining oversight of the student's work," *Comput. Edu.*, vol. 19, no. 1, Jan.–Mar. 2009.
- [12] Y. Wang, S. Zhong, H. Zhang, Y. Zhong, and L. Bai, "Research of automatic assessment system of virtual experiment in middle school biology based on virtual simulation technology," in *Proc. Int. Conf. E-Learn. Games*, 2010, pp. 345–352.
- [13] Y. E. Chen, C. L. Liu, T. H. Chang and C. H. Lee, "An unsupervised automated essay-scoring system," *IEEE Intell. Syst.*, vol. 25, no. 5, pp. 61–67, Sep./Oct. 2010.
- [14] A. E. Tshibalo, "The Potential impact of computer-aided assessment technology in higher education," *South African J. Higher Edu.*, vol. 22, no. 6, pp. 686–695, 2007.
- [15] S. Brandle, "Automated grading of student programming assignments," *J. Comput. Sci. Colleges*, vol. 25, no. 2, pp. 284–286, May 2010.
- [16] R. Alur, L. D'Antoni, S. Gulwani, D. Kini, and M. Viswanathan, "Automated grading of DFA constructions," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1976–1982.

12. [http://youtu.be/W\\_8994oCA9M](http://youtu.be/W_8994oCA9M)

13. <http://youtu.be/PFObxHCFtdo>

14. <http://youtu.be/JO6A2bQ65Y0>

- [17] L. Yang, "An automatic grading model for learning assessment," in *Proc. Int. Conf. e-Education, e-Bus., e-Manage. e-Learn.*, 2010, pp. 217–220.
- [18] A. Tartaglia and E. Tresso, "An automatic evaluation system for technical education at the university level," *IEEE Trans. Edu.*, vol. 45, no. 3, pp. 268–275, Aug. 2002.
- [19] R. Singh, S. Gulwani, and A. Solar-Lezama, "Automated feedback generation for introductory programming assignments," in *Proc. 34th ACM SIGPLAN Conf. Program. Lang. Des. Implementation*, 2013, pp. 15–26.
- [20] M. L. Nguyen, S. C. Hui and A. C. M. Fing, "Web-based mathematical testing with automatic assessment," in *Proc. 12th Int. Conf. Artif. Intell.*, 2012, pp. 347–358.
- [21] L. de Alfaro and M. Shavlovsky, "CrowdGrader: a tool for crowdsourcing the evaluation of homework assignments," in *Proc. 45th ACM Tech. Symp. Comput. Sci. Educ.*, 2014, pp. 415–420.
- [22] S. Swithenby, "E-assessment for open learning," in *Proc. Annu. Conf. Eur. Assoc. Distance Teach. Universities*, 2006.
- [23] M. Mahana, M. Johns and A. Apte, "Automated essay grading using machine learning," *Mach. Learn. Session, Stanford University*, 2012.
- [24] H. Q. Jiang, L. Zhang and W. Q. Ye, "The automatic evaluation strategies and methods of multimedia work assignment," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, 2009, pp. 1–5.
- [25] K. D. Forbus, J. Usher, A. Lovett, K. Lockwood and J. Wetzel, "CogSketch: Sketch understanding for cognitive science research and for education," *Topics Cognitive Sci.*, vol. 3, pp. 648–666, 2011.
- [26] Z. Shukur, Y. Away and M. A. Dawari, "Computer-aided marking system for engineering drawing," in *Proc. Int. Conf. Soc. Inf. Technol. Teacher Educ.*, 2004, pp. 1852–1857.
- [27] K. A. Hekman and M. T. Gordon, "Automated grading of first year student CAD work," in *Proc. 120th ASEE Annu. Conf.*, 2013, pp. 1–10.
- [28] A. Sanna, F. Lamberti, G. Paravati and C. Demartini, "Automatic assessment of 3D modeling exams," *IEEE Trans. Learn. Technol.*, vol. 5, no. 1, pp. 2–10, Jan.–Mar. 2012.
- [29] J. English, "Automated assessment of GUI programs using JEWEL," in *Proc. 9th Annu. SIGCSE Conf. Innovation Technol. Comput. Sci. Educ.*, 2004, pp. 137–141.
- [30] D. H. Baxter and M. J. Guerci, "Automating an introductory computer aided design course to improve student evaluation," in *Proc. Annu. Meeting Am. Soc. Eng. Educ.*, 2003, pp. 1179–1180.
- [31] M. Thornton, S. H. Edwards, R. P. Tan and M. A. Perez-Quinones, "Supporting student-written tests of GUI programs," in *Proc. 39th SIGCSE Tech. Symp. Comput. Sci. Edu.*, 2008, pp. 537–541.
- [32] J. Ramos, M. A. Trenas, E. Gutierrez, and S. Romero, "E-assessment of Matlab assignments in moodle: Application to an introductory programming course for engineers," in *Proc. Comput. Appl. Eng. Edu.*, 2011, pp. 1–6.
- [33] T. G. Hill, "Word grader and PowerPoint grader," *ACM Inroads*, vol. 2, no. 2, pp. 34–36, Jun. 2011.
- [34] P. Thomas, N. Smith and K. Waugh, "Automatically assessing graph-based diagrams," *Learn., Media Technol.*, vol. 33, no. 3, pp. 249–267, 2008.
- [35] J. Masters, "Automated scoring of an interactive geometry item: A proof of concept," *J. Technol., Learn., Assessment*, vol. 8, no. 7, pp. 1–39, 2010.
- [36] P. Thomas, N. Smith and K. Waugh, "Automatic assessment of sequence diagrams," in *Proc. 12th Int. CAA Conf.*, 2008, pp. 1–14.
- [37] F. Batmaz and C. J. Hinde, "A Web-based automatic assessment tool for conceptual database design," in *Proc. 6th Conf. Web-Based Educ.*, 2007, pp. 427–432.
- [38] T. J. Branoff, "Integrating constraint-based CAD into an introductory engineering graphics course: Activities and grading strategies," in *Proc. Annu. Meeting Amer. Soc. Eng. Educ.*, 2003.
- [39] J. L. Colwell, J. Whittington and J. Higley, "Assessment measures and outcomes for computer graphics programs," *Eng. Des. Graph. J.*, vol. 69, no. 3, pp. 24–33, 2005.
- [40] Z. Jin, J. Jia, Y. Liu, Y. Wang and L. Cai, "An automated grading method for singing evaluation," *Recent Adv. Comput. Sci. Inf. Eng.*, vol. 5, pp. 691–696, 2012.
- [41] J. C. P. Chan, H. Leung, J. K. T. Tang and T. Komura, "A virtual reality dance training system using motion capture technology," *IEEE Trans. Learn. Technol.*, vol. 4, no. 2, pp. 187–195, Apr. 2011.
- [42] C. Li, and T. Chen, "Aesthetic Visual Quality Assessment of Paintings," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 2, pp. 236–252, Apr. 2009.
- [43] G. M. Sanchez-Torrubia, C. Torres-Blanc and S. Escribano, "GRAPHS: A learning environment for graph algorithm simulation primed for automatic fuzzy assessment," in *Proc. 10th Int. Conf. Comput. Edu. Res.*, 2010, pp. 62–67.
- [44] M., Flickner, H. Sawhney, W. Niblack, J. Ashley, H. Qian, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Comput.*, vol. 29, no. 9, pp. 23–32, Sep. 1995.
- [45] E. Chalom, E. Asa and E. Biton, "Measuring image similarity: An overview of some useful application," in *Proc. IEEE Instrum. Measurement Mag.*, 2013, pp. 24–28.
- [46] W. Hu, N. Xie, L. Li, X. Zeng and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 797–819, Nov. 2011.
- [47] S. S. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 59–74, Jan. 2003.
- [48] C. W. Su, H. Y. M. Liao, H. R. Tyan, C. W. Lin, D. Y. Chen, K. C. Fan, "Motion flow-based video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1193–1201, Oct. 2007.
- [49] Computer 3D. Modelling and Animation, Level 5, N23010, Further Education and Training Awards Council, 2013.
- [50] J. Lasseter, "Principles of traditional animation applied to 3D computer animation," *Comput. Graph.*, pp. 35–44, vol. 21, no. 4, Jul. 1987.
- [51] K. Miura, H. Furukawa and M. Shoji, "Similarity of human motion: Congruity between perception and data," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2006, pp. 1184–1189.
- [52] P. T. Chua, R. Crivella, B. Daly, H. Ning, R. Schaaf, D. Ventura, T. Camill, J. Hodgins and R. Paush, "Training for physical tasks in virtual environments: Tai-chi," *Proc. IEEE Virtual Reality*, 2003, pp. 87–94.
- [53] J. Chan, H. Leung, K. T. Tang and T. Komura, "Immersive performance training tools using motion capture technology," *Proc. 1st Int. Conf. Immersive Telecommun.*, art. no. 7, 2007.
- [54] M. Muller, Information Retrieval for Music and Motion. New York, NY, USA: Springer-Verlag, 2007.
- [55] S. Celebi, A. S. Aydin, T. T. Temiz and T. Arici, "Robust gesture recognition using feature pre-processing and weighted dynamic time warping," *Proc. Int. Conf. CVTA.*, 2013, pp. 620–625.
- [56] J. K. T. Tang, H. Leung, T. Komura and H. P. H. Shum, "Emulating human perception of motion similarity," *Comput. Animation Virtual Worlds*, vol. 19, no. 3–4, pp. 211–221, 2008.
- [57] R. Blake and M. Shiffrar, "Perception of human motion," *Annu. Rev. Psychol.*, vol. 58, pp. 47–73, 2006.
- [58] L. Ren, A. Patrick, A. A. Efros, J. K. Hodgins and J. M. Rehg, "A data-driven approach to quantifying natural human motion," *ACM Trans. Graph.*, vol. 24, pp. 1090–1097, 2005.



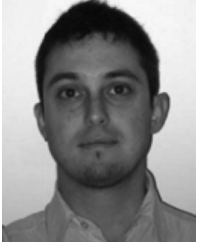
**Fabrizio Lamberti** received the MSc and the PhD degrees in computer engineering from Politecnico di Torino University, Italy, in 2000 and 2005, respectively. Since 2006, he has been an assistant professor in the Dipartimento di Automatica e Informatica, Politecnico di Torino University. He has published a number of technical papers in international books, journals, and conferences in the areas of computer graphics, visualization, human-machine interaction, and mobile computing. He has served as a reviewer, and a program or organization committee member for several conferences. He is a member of the Editorial Advisory Board of international journals. He is a senior member of the IEEE and the IEEE Computer Society. He is a member of the IEEE Computer Society Technical Committee on Visualization and Graphics.



**Andrea Sanna** received the MSc degree in electronic engineering and the PhD degree in computer engineering from Politecnico di Torino University, Turin, Italy, in 1993 and 1997, respectively. Currently, he is an associate professor in the Dipartimento di Automatica e Informatica, Politecnico di Torino University. 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 the ACM.



**Gilles Carlevaris** received the BSc and MSc degrees in computer engineering from Politecnico di Torino University, Turin, Italy, in 2009 and 2012, respectively. He is currently a research assistant in the Dipartimento di Automatica e Informatica, Politecnico di Torino University. His current research interests include human-machine interaction, computer graphics, and virtual reality.



**Gianluca Paravati** received the BSc and MSc degrees in electronic engineering and the PhD degree in computer engineering from Politecnico di Torino University, Turin, Italy, in 2005, 2007, and 2011, respectively. He is currently an assistant professor in the Dipartimento di Automatica e Informatica, Politecnico di Torino University. His current research interests include real-time image processing, collaborative virtual environments, human-machine interaction, remote visualization, and distributed systems.