Point cloud-based automatic assessment of 3D computer animation courseworks

Point Cloud-based Automatic Assessment of 3D Computer Animation Courseworks

Gianluca Paravati, *Senior Member, IEEE,* Fabrizio Lamberti, *Senior Member, IEEE,* Valentina Gatteschi, Claudio Demartini, *Senior Member, IEEE,* and Paolo Montuschi, *Fellow, IEEE*

Abstract—Computer-supported assessment tools can bring significant benefits to both students and teachers. When integrated in traditional education workflows, they may help to reduce the time required to perform the evaluation and consolidate the perception of fairness of the overall process. When integrated within on-line intelligent tutoring systems, they could provide students with a timely feedback and support self-assessment activities. The current work presents an alternative approach (and not just a “yet-another-implementation”) to the problem of automatically evaluating technical skills needed to create 3D computer animations. Although some solutions have been reported already in the literature, their applicability is partially constrained, as they require the teaching staff to define evaluation criteria that are strictly linked to the particular animation technique being assessed. Students are forced to operate in environments where they can only perform a part of the required animation steps, by using a pre-defined set of techniques and tools. To address such limitations, the proposed system exploits shape- and time-based features extracted from the 3D point clouds (i.e., the set of data points) describing animated geometries, which are independent of the particular animation techniques used. Experimental observations collected in the evaluation of course assignments in which students were asked to recreate 3D animations of deformable meshes prepared by the teaching staff showed a good correlation between automatic and manual evaluations. Obtained results confirmed the ability of the proposed approach to cope with heterogeneous evaluation tasks in which the relevant learning outcomes can be properly considered.

Index Terms—Automatic assessment, computer graphics, computer animation, feature extraction, machine learning, point cloud.

1 Introduction

Automatic assessment of learning assignments is a topic of increasing interest today. A number of approaches have been experimented already in different education domains, including mathematics [1], physics [2], [3], biology [4] and literature [5], though the majority of solutions reported in the literature have been developed in the area of computer science [6]–[11]. It is worth observing that the importance of computer-supported assessment is not to be regarded as strictly limited to the education domain. For instance, there exist use cases where automatic assessment tools have been used, e.g., to implement hiring procedures in software development companies [12], to design intelligent platforms for physical rehabilitation [13], [14], to score sport and art performances [15], [16], etc.

Focusing on the education scenario, automatic evaluation of students’ courseworks can indeed bring a number of advantages in traditional classroom settings, but can also have a great potential for learning at scale. In fact, technological evolutions like those associated with the spread of Massive Open Online Courses (MOOCs) [17]–[19] are urging researchers to find suitable ways for monitoring the progress of thousands of students. Moreover, solutions able to cope with personalized education contents as well as to provide both learners and instructors with a prompt feedback about learners’ achievements, e.g., through interactive and flexible tutoring mechanisms [20], are also required.

Benefits expected from automatic evaluation tools concern both instructors and learners. Specifically, instructors’ workload can be alleviated by reducing the overhead associated with grading activities, especially when large classrooms or on-line courses are considered. In this respect, tools devised so far have been used either to directly determine the final grade for a given assignment or to support the instructor in the decision-making process by providing him/her with quantitative measures about learner’s performance in a specific task. Moreover, automatic tools could represent a key factor in reducing the risk of inconsistent or possibly unfair evaluations. Furthermore, they could also be programmed for detecting fraudulent activities such as cheating and plagiarism [21]. With respect to learners, the significance of computer-supported evaluation lays in the possibility to provide them with a prompt and timely feedback, which represents an invaluable support for self-assessment [3], [20], [22].

This paper specifically focuses on the computer-based assessment of technical skills that are expected to be possessed at the end of a computer animation course (that is, artistic and other soft skills that complete the profile of a computer graphics professional are not in the scope of the present work). Assessment is performed by asking students to reproduce a reference animation prepared by the teacher, and by automatically comparing results obtained. It is worth observing that, although the assessment is not meant to evaluate creativity, in the field of computer animation results that may be considered as perceptually similar might have been obtained through very different approaches. For

- *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: {gianluca.paravati, fabrizio.lamberti, valentina.gatteschi, claudio.demartini, paolo.montuschi}@polito.it.*

Manuscript received XXXX XX, XXXX; revised XXXX XX, XXXX.

instance, the same mesh deformation could be produced, e.g., by using a technique based on the parenting of rigged armatures, but also through vertex editing by means of shape keys, lattices and mesh modifiers, rigid-body and soft-body physics simulations, etc.

Hence, automatic assessment should be capable to deal with the largest number of animation techniques possible, in order to avoid the need to use different ad-hoc modules depending on the specific technique that has been adopted and enable consistent evaluations. Based on a review of related literature, in the last years the domain of automatic assessment in the area of computer graphics has been addressed through a limited, though significant, set of works, which considered the evaluation of hand sketches and other multimedia contents, as well as of the products of 2D and 3D modeling and animation tasks [23]–[28]. Unfortunately, methods developed so far to deal with 3D animated contents are not general-purpose, and have been tightly tailored to the specific software used (e.g., the free and open source Blender suite, like in [28]).

In this paper, the above issues are addressed through the design of an automatic assessment methodology that, by relying on the analysis of shape variations occurring in animated 3D geometries, is made independent of the particular technique adopted by the student as well as of the software used to create the animation. Moreover, the proposed methodology allows the instructor to consider in the evaluation process all the steps that are characteristics of the various animation techniques, including mesh preparatory operations (e.g., the rigging and skinning phases in the case of armature deformation, the definition of shape keys in the case of shape deformation through vertex editing, etc.). This way, the range of technical skills that can be automatically assessed is significantly extended compared to existing solutions.

To remove the dependencies of alternative approaches, animations are described in the form of dynamic 3D point (i.e., vertex) clouds, which are used to compute a feature vector describing their similarity w.r.t. the teacher-provided reference animation. A machine learning approach is then adopted to train the automatic system and to identify suitable regression models able to fit the strategies implemented by the instructors in traditional assessment procedures. Similarity features that have been designed are meant to provide a numeric measure of local mesh deformations. Hence, they can immediately be applied to assess animations in which the location of the 3D model does not change over time. This choice allows the proposed method to be easily integrated in other assessment tools where global coordinates are considered. Similarly, it would be easy to generalize the proposed methodology to moving objects by simply taking into account also the metrics proposed, e.g., in [28].

Experimental results obtained by testing the system with animations created by using different techniques showed that the automatically-generated results present a good correlation with teachers' manual evaluations, thus confirming the ability of the proposed approach to serve as a valid support in the assessment phase.

The remaining of the paper is organized as follows. Section 2 reviews works related to automatic assessment in computer science, with a specific focus on the area of computer graphics. Section 3 introduces the learning context the designed approach has been targeted to, presents the traditional assessment process adopted so far and defines the requirements for the proposed methodology. Section 4 gives a general overview of the designed system, whereas Section 5 reports on implementation details. Section 6 is devoted to present and discuss experimental results that have been obtained. Lastly, Section 7 sketches the main conclusions that can be drawn based on the experience made, and provides some indications about research directions to be possibly investigated in the future.

2 Related Works

The problem of finding suitable solutions to support the evaluation of students' learning achievements through computer-based mechanisms has been tackled in a number of different domains in the last years. This problem is getting ever more relevant today with the diffusion of e-learning that, on the one hand, makes education become ever more widespread and, on the other hand, increases the amount of learning contents (including examination products) that are available in digital form.

Indeed, computer science has been one of the main application areas for computer-supported assessment techniques, and several tools have been developed for automating the evaluation of both basic and advanced skills. For instance, software solutions have been used to automatically grade computer science entrance examinations [29] and to assess basic abilities in the use of office automation tools [7]. Similarly, the mastery of advanced learning outcomes has been automatically evaluated in the areas of computer programming [6], [12], [30], [31], security [11], control systems [8], formal logic [9], digital logic [10], and databases [32].

Despite the proliferation, in the computer science field, of software tools able to automatically identify errors in students' assignments and providing suitable feedback, only few of them dealt with the area of computer graphics [28]. A common strategy pursued in this domain to perform automatic assessment consists in analyzing the differences between a student's solution with respect to a reference. This is the case, for instance, of the correction of computer aided design (CAD) assignments, in which differences between geometric primitives of students' drawings are matched against those provided in a solution prepared by the teaching staff [25], [26], [33]. In [27], computer-supported assessment of 3D modeling courseworks was tackled in a different way, by grounding the comparison on the rendered outputs produced rather than on the input primitives used. This way, evaluation was extended to objects appearance, making it possible to consider in the automatic assessment also material properties and texturing.

Approaches reviewed so far focused on the evaluation of static computer graphics contents, like drawings, renderings, etc., i.e., on the design, or modeling, phase.

The application of automatic techniques for the assessment of computer animation assignments has not been thoroughly investigated yet. Indeed, the dynamic nature of involved digital contents poses more challenges compared to the static scenario.

Computer generated animations can be evaluated by working either on the output produced, i.e., the rendered video sequence, or on raw data, i.e., on data used by the software to produce the final result (a mix of the two approaches could be also considered). The first approach can be assimilated to that exploited on static contents, and can be easily implemented by matching the rendered movies produced by the students with the reference solution provided by the teacher [34]. Video comparison techniques are largely used in content-based video retrieval (CBVR) scenarios [35], [36]. However, they are not fully suitable to evaluate the technical skills of 3D animators. First, these methods are dependent on the appearance of animated objects, which is determined by the application of techniques liked modeling, texturing, etc., which are not necessarily part of the examination goals. For instance, the comparison of video sequences representing the same scene rendered with different parameters (such as lighting, material properties, etc.) would produce inconsistent results. Second, even when the assignment consists in replicating a given reference solution, the goal of the automatic technique is to evaluate animators' skills rather than, simply, the visual coherency of animations produced by the students and the teaching staff. In fact, in some cases, the use of particular animation techniques could produce unwanted modifications to 3D objects geometry that cannot be appreciated by considering only the expected visual output, since they might not be visible from the viewpoint used for rendering the video sequence. To deal with the above issue, multi-camera approaches like the one reported in [37] could be used, which however would make the application of video comparison techniques much more complicated and, in any case, would not represent the ultimate solution to the considered problem since there could be still details that cannot be appreciated in the rendering.

As said, the alternative approach consists in analyzing raw data, i.e., data that can be obtained directly from the animation software without requiring (and depending on the results of) actual rendering. For instance, in [28], a strategy relying on the computation of a set of similarity indicators based on the position and orientation of joints used to animate 3D objects via armature deformation was presented. Despite the interesting results obtained, the main drawback of such approach is that it can only be exploited to assess animations created using the selected technique. It is also characterized by significant limitations in the set of technical skills that can be assessed since, for instance, a ready-to-use armature should be embedded already in the examination package provided to the students. Thus, only students' *posing* skills could be evaluated (i.e., the ability to apply the required transformations to the geometry), but not their ability in carrying out the preparatory steps used to create the armature (through a process known as *rigging*, i.e., defining the bones) and configure it (in a process known as *skinning*, i.e., associating bones to objects vertices).

The goal of the present work is to build upon the advantages offered by raw data analysis to design a methodology for automatic assessment that is independent both of the particular objects to be animated and of the technique used by a given student to create the animation.

In this respect, the adoption of a methodology based on point clouds representing object geometries and their deformations is able to meet both the requirements, thus making the proposed tool able to deal with different types of animations in a way that is natively viewpoint-independent.

From the perspective of point cloud analysis, the problem to be addressed for automatic assessment of 3D animation assignments can be traced back, on the one side, to 3D mesh comparison and, on the other side, to motion analysis. A number of works proposed similarity techniques incorporating many different distance measures. However, most of them considered only features useful for a static analysis, e.g., tailored to object recognition and retrieval [38]–[41]. Static information should be complemented by motion similarity measures based on the analysis of object trajectories, which have been widely studied, e.g., for human activity classification and recognition [16], [42], [43]. In this context, approaches capable of guaranteeing a comparison of motion trajectories that is invariant to local non-linear accelerations and decelerations, i.e., to warping [44], should be considered.

3 Context

The education context that motivated the work reported in this paper concerns a Virtual Animation course that is held in the Product Design curriculum of the Design and Visual Communication bachelor's degree program at Politecnico di Torino. This course is geared within a highly interdisciplinary laboratory (with the term laboratory referring to a collection of learning modules), which includes two other disciplines, namely Design for Serial Production and Design Mechanics. In the laboratory, students take part to a professional simulation in which they are asked to carry out all the steps of a typical design process, from product idea to pre-engineering, by working on a precise theme chosen by the panel of teachers of the three courses. For instance, the theme for the 2015–16 edition of the interdisciplinary laboratory was the design of innovative suitcases.

In the above framework, the Virtual Animation course is devoted to make the students acquire and improve both soft and technical skills required to produce 3D computer animations that are meant to support the visualization of design, manufacturing, assembling and functioning stages of products lifecycle. The assessment of soft and technical skills is carried out in two distinct phases, each characterized by specific assignments.

The first phase involves small groups of students, which are requested to produce one or more short videos in computer graphics showing features, functionalities and use cases for the product they designed in the interdisciplinary laboratory[1]. In this phase, the assessment takes into account students' creativity as well as other soft skills like teamwork and communication, which are essential since from the early design steps in order to share the personal vision of the product and its visual representation. These skills are evaluated throughout the entire duration of the course, by means of review sessions with the whole panel of teachers.

[1] Examples of the videos created in the 2015-16 edition of the course for the assessment phase focusing on soft skills: https://youtu.be/sBhO2aePXJk, https://youtu.be/S0cKNcmNFQs, https://youtu.be/WbKO4GuTblI

While the first phase is aimed to assign a score for the group work based on an evaluation of soft skills, the second phase is devoted to the assessment of individual technical skills. To this aim, every student is administered another assignment, which consists in replicating a teacher-provided 3D animation in a given timeframe. This paper specifically focuses on the development of a computer-based methodology for making the second assessment phase automatic.

The procedure that is traditionally adopted in the Virtual Animation course for the assessment of individual technical skills is based on a manual (i.e., visual) analysis of the animations created by the students. The reference animation to be reproduced is prepared by the teaching staff and it is presented to the students in the form of several videos rendered from different viewpoints (front, side, top and user camera views). Students are provided with a virtual scene containing the static 3D models to be animated, and they are asked to produce an animation as similar as possible to the reference one. Animation tasks consist in applying some kinds of deformations to the given geometries over a specific period of time. 3D contents are assumed to be ready for animation, i.e., students do not need to modify their topology or appearance, and they should just focus on animation steps. However, 3D models have not been prepared for receiving deformations. Hence, students have to carry out some preparatory steps before actually starting to animate. Depending on the animation to be recreated (and/or on other requirements possibly set by the teacher), students may have to apply different techniques, each requesting possibly diverse preparatory actions.

Besides adapting to different animation techniques, the automatic methodology to be designed is requested to reflect the traditional evaluation process, which requires evaluators to answer the three categories of questions reported below, concerning completion rate, pose similarity and timing similarity between reference and students' animations.

- Was the student able to complete the assignment? If not, what has been the total progress on the task?
- Was the animation technique chosen by the student able to reach the expected result for the portion he/she was able to complete? How much are the deformations of the considered model similar to the reference ones? How much are they exaggerated or inappropriate with respect to the given geometry?
- Was the student able to properly manage the timing of the animation? Does the duration of the animation agree with the reference? Or, how long does the animation last? Does keyframing occur at the same time of corresponding frames in the reference animation for the portion he/she was able to complete? Are some parts too fast/slow?

Answering these questions clearly requires a careful work for the teacher, who is requested to examine source files to analyze the students' work in great detail.

In particular, the first set of questions allows the teacher to evaluate the amount of work a student was able to produce during examination.

The second set of questions concerns the evaluation of both technical choices made by the student to obtain specific results (such as the appropriateness of the selected animation technique), and his/her ability to replicate key poses by properly deforming provided geometries. This set of questions represents a sort of "static" analysis, since the teacher investigates about (and evaluates the similarity between) the reference and students' animations at relevant and specific keyframes.

The last set of questions is aimed at analyzing "dynamic" characteristics of the entire animation, which specifically pertain timing aspects. To this aim, the teacher inspects the animation timeline, and evaluates the similarity between the reference and students' animations from the point of view of keyframes position.

In the traditional assessment process, answers provided by the teacher to the three sets of questions translate into an overall score regarding technical skills, which is added to the score assigned for soft skills to compute the final grade. However, the assessment of questions in each set is not based on individual rubrics. Hence, although the above questions identify specific aspects to be investigated in the evaluation, the design of a computer-based method able to automatically implement the traditional assessment process requires a method for comparing two animations and for assigning a similarity score by mimicking as close as possible teacher's behavior.

4 System Overview

The system for automatically evaluating technical computer animation skills reported in this paper requires the definition of an assessment estimation model able to predict, given in input a reference and one or more students' animations, what would be the grade(s) assigned by the teaching staff based on the strategy described in the previous section. The underlying process can be considered as split in two phases, later referred to as the *training* phase and the *assessment* phase.

In the training phase, a set of animations that have been previously graded by the teaching staff is exploited to create an assessment estimation model capable to maximize the correlation between automatic and manual scores. The training phase can be performed only once (when the first assignments of a course have been completed), or repeated several times (e.g., using other animations from next assignments) in order to possibly improve estimation performances. In the assessment phase, the estimation model trained with graded animations is exploited to automatically get an estimate of the teacher's score for each student's animation.

The functioning of the training phase is illustrated in Fig. 1. The reference (R) and training (T) animations are first processed by a software module devoted to PCF Extraction, which is in charge to translate the files produced by the specific animation tool used by the students to so-called point cloud frame (PCF) sequences (the module is basically a plug-in for that tool). The use of point cloud data guarantees that the creation of the assessment estimation model and the following processing steps are independent of the particular software and animation technique used. All the PCF sequences are stored in a Point Cloud Database.

Since the goal of the examination procedure consists in reproducing an animation given a reference one, PCF sequences are processed in order to extract a set of features

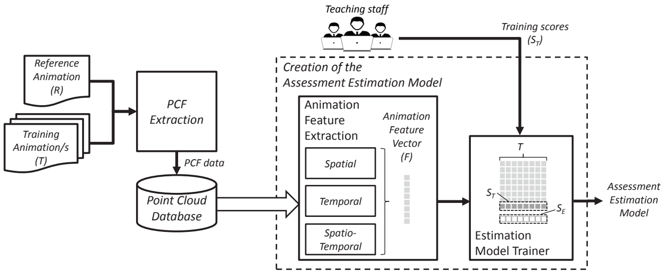


Fig. 1. Training phase: an assessment estimation model is created using animations previously scored by the teaching staff.

(described by the feature vector F) capable to characterize the similarity between R and T. Feature extraction is performed by the Animation Feature Extraction module.

Each feature is meant to characterize aspects that are tackled in questions considered in the traditional assessment process as described in Section 3. Three categories of features have been designed, referred to as *spatial*, *temporal*, and *spatio-temporal*. Spatial features are aimed to analyze static information, such as mesh similarity in a specific PCF. Temporal features and spatio-temporal features focus on dynamic aspects, e.g., on timing similarity between two animations (temporal) and movement similarity when a mesh is deformed (spatio-temporal). Details about the above features will be given in Section 5.

Given the fact that, in the considered scenario students are provided with the 3D model to be animated (after rigging, if necessary) and a reference animation, no editing of the mesh is actually required. That is, the size and topology of the student's and reference point clouds are the same and remain constant in the animation. Hence, PCF sequences of the reference and comparison animations can be created by simply recording, for every frame of the sequence, the position of each vertex belonging to the source 3D model. Similarly, features can be defined in a simplified way, by taking into account that a vertex belonging to one of the clouds can always be paired with exactly one vertex of the other cloud. Nonetheless, more complex strategies taking into account also changes both in the size and topology of the animated mesh could be implemented [45]. It is worth observing that similarity features could be influenced by the density of the point clouds which, at present, is not considered (precisely, it is simply assumed that the source 3D model has been modeled with the number of vertices that is needed to represented the intended deformations). Future activities will be devoted to investigate this issue and to possibly identify suitable strategies to mitigate its effects.

The assessment estimation model is finally created by using statistical regression analysis, in order to correlate scores assigned by the teaching staff (S_T) to automatically-computed (i.e., estimated) ones (S_E). Different regression methods could be used at this stage. Section 6 will investigate the applicability of several methods, i.e., linear, Bayesian linear, decision forest, Poisson, neural network, boosted decision tree, and support vector machine regression. Estimated scores are produced based on animation feature vectors (F) extracted for every training animation and associated training scores (which are combined in the feature matrix illustrated in Fig. 1). The Estimation Model Trainer is responsible for determining the coefficients required to configure the selected estimation model.

Once the estimation model has been trained, the assessment phase is carried out as illustrated in the sequence diagram reported in Fig. 2. The diagram shows the interactions between the teacher and the different modules composing the designed tool. In order to automatically determine the score for a given student's animation, the teacher provides the system with the identifiers of the particular student and of the reference animation. It is assumed that both the student's (later referred to also as the "comparison") and the reference animations have been converted and stored as PCF sequences in the database. The assessment tool requests to the Animation Feature Extraction module the feature vectors describing the two animations, which are calculated on the retrieved point clouds. Feature vectors are sent to the assessment estimation model trained during the previous phase, which finally produces the requested score.

5 Feature Extraction

This section illustrates the Animation Feature Extraction module previously introduced, which is used to compute the metrics describing the similarity between students' and reference animations. As per design requirements, these metrics have been designed by trying to mimic the teacher's cognitive processes activated when he/she assesses a student's work by following the traditional (question-based) evaluation strategy discussed in Section 3.

The designed metrics define the components of the animation feature vector (F) used both for creating the estimation model (in the training phase) and for obtaining the estimated scores (in the assessment phase). All the metrics are defined in a normalized $(0, 1]$ range. As said, metrics refer to three domains, namely spatial, temporal, and spatio-temporal. Spatial metrics concerns the analysis of 3D geometries at a specific time (intra-frame analysis). Temporal metrics are obtained by analyzing time-related animation data, like duration, keyframes position, etc. Finally, spatio-temporal metrics extend the analysis of 3D geometries to a sequence of frames (inter-frame analysis).

From an algorithmic point of view, metrics are computed in two different steps of the feature extraction algorithm, i.e., *Sequence Alignment* and *Point Cloud Comparison*.

The Sequence Alignment step is needed to account for the fact that animations to be analyzed could be produced with a different timing, but a comparison would only be possible if the correspondence between frames is known in advance. Hence, the role of this step is to normalize the length of one sequence with respect to the other one. The assumption is that, in general, two animations should be compared by considering that they may differ in duration or speed. As a matter of example, an animation task could be fully completed and the comparison animation could look like very similar to the reference one in terms of deformations, but it could be quite diverse from the point of view of timing (e.g., it could last a fraction of the total duration of the other one). Similarly, a comparison animation could be perfectly synchronized with the reference one for the portion that has been completed, but the animation could

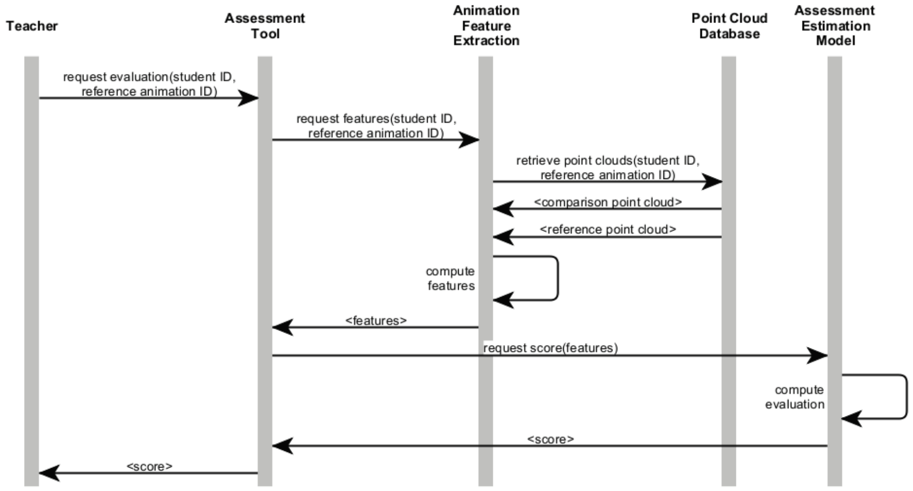

Teacher
Assessment Tool
Animation Feature Extraction
Point Cloud Database
Assessment Estimation Model
request evaluation(student ID, reference animation ID)
request features(student ID, reference animation ID)
retrieve point clouds(student ID, reference animation ID)
<comparison point cloud>
<reference point cloud>
compute features
<features>
request score(features)
compute evaluation
<score>
<score>

Fig. 2. Sequence diagram of the assessment phase. Students' animations are scored using the assessment model created in the training phase.

be unfinished. Mixed situations could clearly occur. The alignment phase is based on the application of the well-know Dynamic Time Warping (DTW) algorithm [46], which produces a warped version of the two input sequences. A description of this step is given in Appendix A.

Once the alignment phase has been completed, two temporal metrics can be computed. The first metric, referred to as *Timing similarity* (T), indicates how much a sequence has been warped. The second one, named *Completion index* (C), is used to express the amount of work carried out by the student. Basically, the objective of the timing similarity metric (T) is to capture how timing has been managed by the student in the creation of the comparison animation. The more the two sequences are misaligned, the more the comparison and reference sequences are different from the point of view of the timing. The completion index (C) is aimed to indicate the amount of work carried out by the student. It represents the percentage of completion of the assigned animation task, and is computed as the ratio between the last animated frame for which a correspondence has been found by the sequence alignment step in the reference sequence and the duration of the sequence itself.

Warped sequences of the reference and comparison animations obtained in the Sequence Alignment step are passed to the Point Cloud Comparison step, which is meant to compute a set of measures describing their correlation based on a frame-by-frame comparison. Two categories of metrics have been devised for this task, focusing on mesh (spatial) and motion (spatio-temporal) similarity, respectively.

The mesh similarity category contains metrics that implement a static analysis of mesh deformations. In fact, their focus is only on pose similarity, independently of animation timing. Two separate metrics have been designed, in order to consider both parts of 3D models that should be animated and, correspondingly, parts that should not. Fig. 3 reports an example of mesh similarity computation between reference and comparison animations. Fig. 3 (a) and (b) show two examples of point clouds extracted, respectively, for a specific frame of a reference and comparison sequence (in which a 3D character is moving its eyelids and lips). In the particular frame considered, vertices that should move are those shown in Fig. 3 (c), i.e., corresponding to the character's lips and eyelids. However, as illustrated in Fig. 3 (d), the analysis of the comparison mesh reveals that other vertices (i.e., in the back of the neck) have been erroneously moved by the student in the posing process.

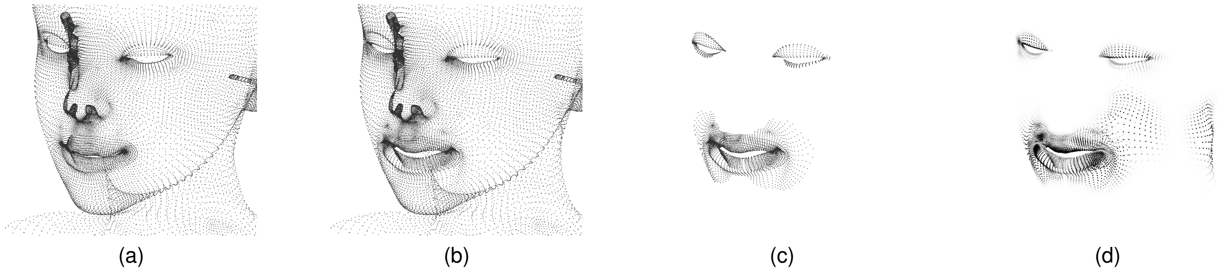
By grounding the analysis on so-called moving and stationary clouds $\mathcal{V}_M$ and $\mathcal{V}_S$, it is possible to define the *moving cloud mesh similarity* (M_m) and *stationary cloud mesh similarity* (M_s) metrics. While the first metric computes the similarity between the point clouds made up of vertices that should move in a specific frame, the latter considers the complementary set of vertices, giving an indication of how well meshes have been prepared for receiving deformations (e.g., in the rigging process, the application of lattices, etc.).

The motion similarity category contains metrics that carry out a dynamic analysis of mesh deformations. In this case, moving clouds have been analyzed for computing a *motion orientation similarity* (O_m) metric and a *motion range similarity* (R_m) metric. The first metric compares the average motion direction of the two moving clouds, whereas the latter provides information about the completeness of mesh movements. Practically, the motion orientation metric (O_m) is aimed to evaluate the similarity between the average motion direction of aligned reference and comparison point cloud sequences. That is, it provides a measure of the instantaneous error between the motion vector orientations of the input clouds and is independent of their magnitude. The motion range similarity metric (R_m), which is computed again on the moving cloud, compares the speed of the reference and comparison point cloud sequences, i.e., the magnitude of the velocity, throughout the animation.

A mathematical description of the metrics involved in the feature extraction process is given in Appendix B.

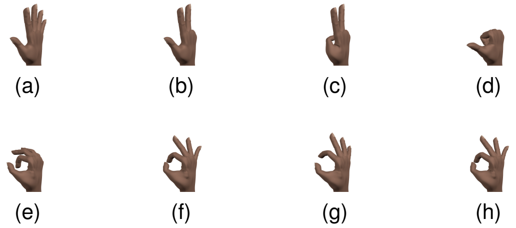
6 Results

This section presents the experimental results obtained on a dataset that was created by collecting students' works produced in three separate animation assignments (participated by 74, 74 and 79 students, respectively). In each assignment, students were asked to carry out the animation task by following the procedure described in Section 3.



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

Fig. 3. Examples of point clouds for reference and comparison animations at a given frame. (a) Reference point cloud. (b) Comparison point cloud. (c) Moving point cloud. (d) Distance between reference and comparison point clouds: darker vertices are farther away from the reference mesh.



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

(e) (f) (g) (h)

Fig. 4. Representative renderings of the *hand* reference animation at frame (a) 1, (b) 41, (c) 71, (d) 101, (e) 123, (f) 131, (g) 154, and (h) 166.

The first reference animation[2], which in the following will be referred to as the *hand* sequence, was produced by rigging, skinning and posing a publicly-available 3D model of a human hand. In particular, students were asked to recreate the sequence of gestures illustrated in Fig. 4. The second animation[3] dealt with facial expressions and, for this, it will be later referred to as the *face* sequence. Some representative frames showing deformations applied to the selected human face model are illustrated in Fig. 5. Finally, the third animation[4], referred to as *lamp*, was based on the Pixar-like lamp character depicted in Fig. 6. In this case, the feature extraction module only takes into account mesh deformations by working on local coordinates (as said, global transformations could be tackled through other approaches already reported in the literature).

In all the assignments, students were provided only with a raw 3D geometry of the object to be animated, and were allowed to chose the preferred animation technique to apply in order to obtain the expected result. As said, compared to previous approaches, this fact has an important impact on flexibility offered by the devised technique.

As a matter of example, for recreating the *hand* and *lamp* animations, almost all the students chose to work with armature deformations. This was actually the same technique used to create the reference animation. However, before moving to armature posing, students had to build their own armature rig, and to parent it to the hand mesh. Thus, even though, in principle, an automatic method based on armatures comparison like the one reported in [28] could be used to assess animations created using this technique, it would require the rigged and skinned mesh to be included already in the assignment package provided to the student. In fact, for the method in [28] to provide consistent results, the armature in the reference and student's animations should be characterized by the same topology, and bones should be assigned to the same vertices. On the contrary, the proposed method can live without the above requirements, and is capable to cope with skills like rigging and skinning that, with posing, are implicitly related to the considered animation technique.

Similar considerations apply to the *face* animation. In this case, the reference animation was prepared by using shape keys to modify several sub-parts of the original mesh (i.e., eyelids and lips), and by applying drivers to control face deformations through the bones of an external armature. Since mesh deformation in the reference animation was not directly controlled by a rig, methods like the one in [28] cannot be applied. Moreover, since the same animation could be produced through more than one of the techniques learned in class (e.g., shape keys, lattices, etc.), the relevance of an automatic assessment method like the proposed one, which focuses on the final result rather than on the animation procedure adopted, becomes particularly evident.

It is worth observing that the three sequences are intentionally very different, not only in the animation techniques used. For instance, from the point of view of movement ranges, the *face* sequence presents more subtle deformations with respect to the *hand* and *lamp* sequences. Extracted animation features and related normalization strategies illustrated in Section 5 were designed to take the above aspects into consideration, with the aim to let the designed system account for details that might be hard to quantify in the traditional assessment based on visual inspection.

In the following, a preliminary assessment of the extracted animation features will be firstly performed, with the aim to evaluate their effectiveness in characterizing the degree of similarity between reference and students' animations. Then, results obtained by training different assessment estimation models on various configurations of the students' animations and by using them to automatically perform the evaluation will be discussed.

6.1 Verification of Similarity Metrics

In order to validate the discrimination capabilities of the designed similarity metrics, the algorithm implemented by the Animation Feature Extraction module in Fig. 1 was initially tested in controlled conditions, by working with an ad-hoc set of verification sequences that were created by modifying the reference *hand* and *face* animations.

[2] https://youtu.be/Ambuiy48DlM
[3] https://youtu.be/mytt8IPPru4
[4] https://youtu.be/ehBxzgFq1hA

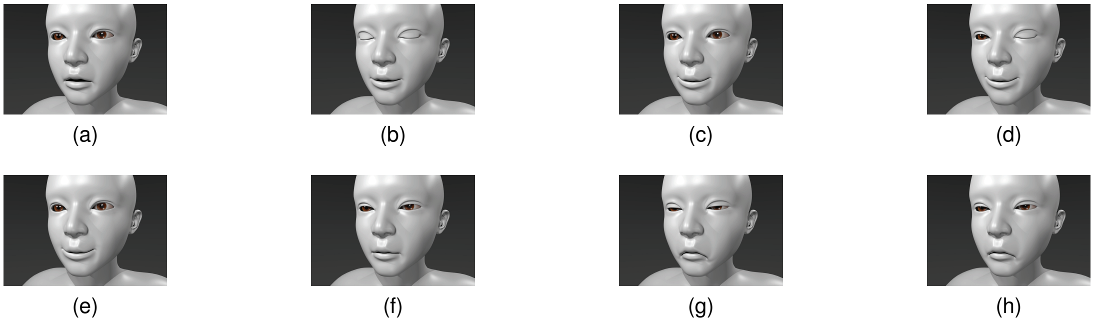


Fig. 5. Representative renderings of the *face* reference animation at frame (a) 1, (b) 25, (c) 50, (d) 87, (e) 112, (f) 150, (g) 175, and (h) 200.

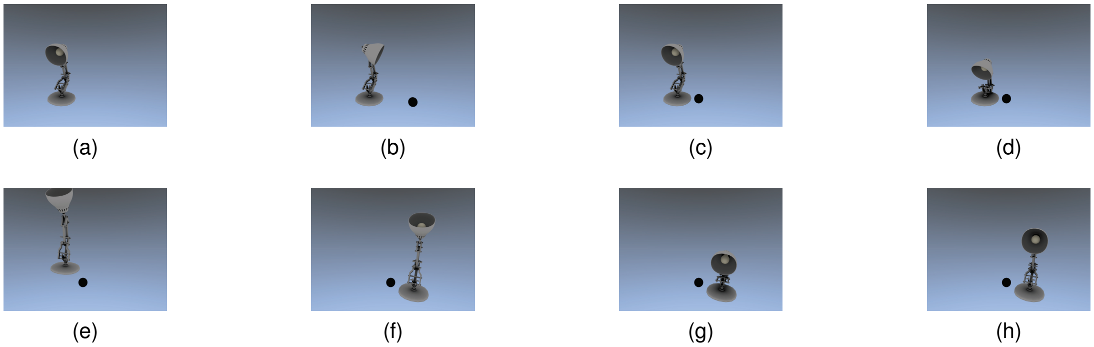


Fig. 6. Representative renderings of the *lamp* reference animation at frame (a) 1, (b) 78, (c) 131, (d) 145, (e) 164, (f) 198, (g) 211, and (h) 228.

TABLE 1
Features Computed on a Verification Set of Sequences Obtained by Introducing Controlled Modifications into the Reference Animations

		Sequence Alignment step		Point Cloud Comparison step			
Verification sequence		Timing similarity (T)	Completion index (C)	Moving cloud Mesh sim. (M_m)	Stationary cloud Mesh sim. (M_s)	Movement Orientation similarity (O_m)	Motion Range similarity (R_m)
Hand	unchanged mesh, 5% wrong timing	**0.951**	1.000	0.995	0.999	0.995	0.993
	unchanged mesh, 50% wrong timing	**0.520**	0.986	0.972	0.993	0.957	0.995
	changed mesh, right timing	**0.982**	1.000	0.966	0.990	0.965	0.907
	changed mesh, 20% wrong timing	**0.808**	1.000	0.966	0.989	0.965	0.911
	changed mesh, right timing, frame 101 (46.75%)	0.961	**0.468**	0.951	0.994	0.980	0.901
Face	unchanged mesh, 5% wrong timing	**0.947**	1.000	0.976	0.999	0.987	0.972
	unchanged mesh, 50% wrong timing	**0.521**	0.990	0.874	0.958	0.873	0.818
	unchanged mesh, right timing, frame 50 (25%)	1.000	**0.250**	1.000	1.000	1.000	1.000
	changed mesh, right timing	**0.990**	1.000	0.768	0.928	0.972	0.980
	changed mesh, 20% wrong timing	**0.825**	1.000	0.766	0.928	0.971	0.981

The main aim was to check the functioning of the Sequence Alignment step (from which the first two features T and C are derived), so that to ensure that spatial and spatio-temporal features related to the Point Cloud Comparison step (M_m, M_s, O_m, and R_m) can be properly computed by considering the right correspondences between frames of the reference and students' sequences. In fact, a misalignment would lead to incorrect values for both spatial and spatio-temporal features, and independence between computed metrics could not be guaranteed.

Given the above strategy, modifications to the reference animations concerned both timing and content. In particular, verification sequences were produced by changing the speed of the reference animation (scaling the duration by a certain factor), varying the degree of completion (cutting the animation at a given frame), and altering poses (slightly modifying the position of mesh vertices at particular keyframes).

Results obtained on these verification sequences are reported in Table 1. For each sequence, values obtained for all the components of the feature vector are tabulated in separate columns. The notation "unchanged mesh" is used to refer to verification sequences in which the position of vertices was not varied w.r.t. the reference animation. Sequences in which vertices were moved are identified by the "changed mesh" notation.

Labels in the form "$s\%$ wrong timing" in sequence names are exploited to describe the factor s used for scaling (down, in this case, though considerations would be the same for upscaling) animation duration. For instance, a "20% wrong timing" label identifies an animation that is 20% faster than the reference one. Sequences for which the original timing was not altered contain the label "right timing" in their name. Scaling factors were applied to both animations with changed and unchanged meshes.

Verification sequences that were created for testing the correctness of information regarding animation completeness are identified by the "frame f (p%)" notation. The f value indicates the keyframe at which reference animation was cut, whereas $p\%$ describes keyframe position (i.e., completion rate), in percentage. Specifically, the "frame 101 (46.75%)" label identifies a sequence that was obtained by cutting reference animation (*hand*, in this case) precisely at frame 101. Since the reference sequence is 216-frame long, completion rate is 46.75%. Similarly, label "frame 50 (25%)" is used to identify a sequence obtained by cutting the reference animation (*hand*, in this case, whose length is 200 frames) at frame 50 (i.e., 25% completion rate).

From reported results it can be easily observed that the Sequence Alignment step produced values for the timing similarity and completion index metrics which changed as expected (meaningful values are highlighted in bold in the corresponding columns). As a matter of example, for the verification sequence named "unchanged mesh, 50% wrong timing" it was $T = 0.520$ and $C = 0.986$, i.e., slightly more than the expected 50% speedup, slightly less that full completion. Similarly, for verification sequence named "unchanged mesh, right timing, frame 50 (25%)" it was $T = 1.000$ and $C = 0.250$, i.e., exactly the awaited timing and completion rate.

Results concerning the four metrics obtained in the Point Cloud Comparison step are reported mainly for sake of completeness. In fact, mesh alterations quantified by the designed metrics would be hard to verify when changes are introduced simultaneously also to timing and sequence completion rate. Notwithstanding, validation tasks carried out in simplified conditions confirmed the appropriateness of calculations made and of normalizations applied once sequences have been properly aligned.

TABLE 2
Parameters of the Estimation Models

Regression Model		Parameter	Value
L	Linear	Solution method	Least squares
		L2 regularization weight	0.001
BL	Bayesian Linear	Regularization weight	1
DF	Decision Forest	Resampling method	Bagging
		Number of decision trees	8
		Max. depth of the decision trees	32
		Number of random splits per node	128
		Minimum number of samples per leaf node	1
P	Poisson	Optimization tolerance	10^{-7}
		L1 regularization weight	1
		L2 regularization weight	1
		Memory size for L-BFGS	20
NN	Neural Network	Number of hidden nodes	100
		Learning rate	0.005
		Number of learning iterations	100
		Initial learning weight diameter	0.1
		Momentum	0
		Type of normalizer	min-max
BDT	Boosted Decision Tree	Max. number of leaves per tree	20
		Minimum number of samples per leaf node	10
		Learning rate	0.2
		Total number of trees constructed	100
SVM	Support Vector Machine	Kernel	RBF
		Cost parameter C	10000
		Width parameter γ	0.01

6.2 Automatic assessment

This section reports and analyzes the results obtained by using the designed metrics with several statistical regression methods, with the aim to study their applicability in the considered scenario and get insights that could be used for guiding the process devoted to the creation of the assessment estimation model for a given animation course.

Results have been gathered by working with a dataset made up of the students' *hand*, *face* and *lamp* animations previously described (composed by 74, 74, and 79 elements, respectively). All the animations belonging to the dataset were assessed by a panel of two teachers, who scored each animation in a worse-to-better scale ranging from 0% (very low similarity) to 100% (very high similarity). The reasons for choosing a percentage scale range was two-fold. On one hand, teacher was left a fine-grained discriminating power. On the other hand, once an assessment is obtained, it would be straightforward to convert the percentage to any grading scale used in a particular course.

Teachers' scores were averaged and used as target values for the estimation models described in Table 2. Estimation models were first applied to parts of the overall dataset (referred to as *training sets*) to train the system. Afterwards, trained models were applied to other parts of the dataset (referred to as *test sets*) to compute the assessment error between predicted values (i.e., scores computed by a particular model based on the designed features) and target values (i.e., average scores assigned by the teaching staff).

An in-depth performance analysis of the designed approach can be carried out by looking at Table 3, where a wide range of configurations of the considered dataset is explored. Specifically, figures reported therein were obtained by changing the set of animations used to train the system (second column) and by computing automatic estimates on various test sets (third column). Both training and test sets were randomly extracted from the initial dataset with different percentages. For instance, 50% H in the training set column means that half of the *hand* dataset was used to train the system. Symbol * is used to indicate that a given training set was used also as test set. The first column indicates the experiment identifier for later reference, while the remaining columns refer to the percentage mean absolute error (MAE) and coefficient of determination R^2 describing the correlation degree between computed estimates and teachers' evaluations for each considered estimation model.

The first two experiments (E1 and E2) were conceived to evaluate the predictive power of the estimation models of interest when a limited number of animation sequences is available to perform the training (e.g., when the devised computer-based assessment method is firstly introduced in a given course). In experiment E1, the training set was

TABLE 3
Assessment Results by Applying Different Estimation Models. Comparison Between Automatic and Teachers Manual Evaluations.

Exp.	Training	Test	L		B		P		NN		DF		BDT		SVM	
ID	set	set	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
E1	50% H	50% H*	6.58	0.92	7.6	0.9	11.18	0.81	13.78	0.71	4.14	0.96	1.22	0.99	7.43	0.94
E2	50% H	50%H	6.32	0.89	6.67	0.9	8.95	0.84	13.34	0.71	6.37	0.89	6.64	0.87	7.51	0.88
E3	H	H*	6.27	0.92	6.73	0.91	9.5	0.86	7.42	0.87	3.02	0.97	0.56	0.99	6.7	0.94
E4	20% F	80% F	9.87	0.13	8.48	0.37	7.46	0.51	10.39	0.12	8.38	0.33	14.38	-0.67	7.39	0.57
E5	40% F	60% F	6.48	0.54	6.35	0.57	6.88	0.55	11.7	-0.05	7.36	0.46	13.21	-0.41	7.24	0.47
E6	50% F	50% F	5.79	0.58	5.74	0.6	6.29	0.59	11.41	-0.12	7.33	0.38	9.84	0.01	7.91	0.51
E7	50% F	50% F*	7.11	0.71	4.05	0.89	7.78	0.66	8.41	0.53	11.96	0.11	3.36	0.91	7.19	0.71
E8	20% L	80% L	8.37	0.16	6.9	0.58	8.21	0.38	9.71	-0.02	7.74	0.48	10.97	-0.54	8.38	0.49
E9	40% L	60% L	7.41	0.65	7.55	0.61	8.7	0.42	10.81	-0.03	9.92	0.02	9.19	0.38	6.81	0.73
E10	50% L	50% L	7.67	0.64	7.92	0.63	9.97	0.36	11.85	-0.03	10.42	0.39	10.49	0.23	6.67	0.81
E11	50% L	50% L*	6.54	0.91	3.72	0.96	7.64	0.89	11.18	0.81	12.68	0.76	1.57	0.99	6.37	0.91
E12	H	L	10.89	0.11	6.45	0.64	6.48	0.68	6.5	0.63	7.23	0.55	7.13	0.55	13.75	0.67
E13	H + F	L	7.26	0.57	5.83	0.73	6.08	0.69	6.06	0.71	6.46	0.63	6.8	0.61	6.97	0.63
E14	H + 20% L	80% L	6.23	0.71	6	0.74	6.07	0.75	6.15	0.75	7.49	0.57	6.55	0.63	8.66	0.68
E15	H + 40% L	60% L	6.26	0.78	6.03	0.78	5.86	0.81	6.13	0.78	7.58	0.65	6.81	0.67	8.52	0.73
E16	H + 50% L	50% L	6.39	0.8	6.21	0.8	5.63	0.86	7.01	0.77	7.33	0.72	5.56	0.74	8.48	0.76
E17	H+F+20%L	80% L	5.6	0.78	5.67	0.77	5.58	0.79	6.11	0.74	7.16	0.62	7.39	0.57	7.93	0.65
E18	H+F+40%L	60% L	5.88	0.8	5.84	0.8	5.5	0.83	5.99	0.78	7.1	0.69	7.05	0.67	8.03	0.7
E19	H+F+50%L	50% L	6.01	0.82	5.99	0.82	5.45	0.86	6.16	0.81	7.68	0.7	6.62	0.74	8.67	0.71

created by randomly selecting half of the *hand* animations and the predictive power was examined on the same set of sequences. In this experiment, correlation was always rather good (i.e., $R^2 \geq 0.81$) except for the neural network regression model. As expected, a slight decrease in performances was observed when estimation models were evaluated against the remaining sequences of the *hand* dataset (E2). Again, the worst model was the one trained with neural network regression.

Based on the above results, it could be argued that, in some cases, the size of the training set might be too small to perform a proper training. For this reason, experiment E3 was deemed to analyze the behavior of the estimation models on the entire *hand* dataset rather than just on half of the sequences. As expected, performances increased both in terms of MAE and coefficient of determination. The highest improvement was recorded for the neural network, although all the models showed a value of $R^2 \geq 0.86$.

A clear dependence between the size of the training set and predictive power of the particular estimation model being used can also be observed by analyzing other configurations of the animations in the dataset. For instance, in experiments E4–E6, a growing number of training sequences is considered (with a training set made up of 20%, 40% and 50% of the *face* animations, respectively). In general, for almost all the estimation models the correlation between computed estimates and teachers' evaluations improved as the number of sequences in the training set increased. However, performances were influenced also by other factors. As a matter of example, results for the *face* animations were worse than with the *hand* dataset (E1–E3) and, in this case, neural networks and boosted decision tree regression were not able to produce meaningful results at all. Even when the trained model was run on the same data used for training (E7), these two methods definitely achieved poor results. In this case, the reason could be the quality of training data. In fact, the *face* dataset is the one showing the worst correlation coefficient among the teachers ($r_t^{face} = 0.89$, compared to $r_t^{hand} = 0.98$ and $r_t^{lamp} = 0.94$).

Similar considerations hold also when the *lamp* animation is considered (E8–E11). In this case, the observed trend is a general improvement in the predictive power as the number of considered animations increases, except, as before, for neural network and boosted decision tree methods.

After having worked with a dataset made up of the *hand* and *face* animations, experiments from E12 to E19 were performed to simulate a concrete scenario in which new assignments are carried out in a given course, and new students' animations (represented by the *lamp* dataset) become available. In this scenario, the teaching staff should decide whether to apply the assessment estimation model created with animations that were collected in previous assignments or to consider also (part of) the new ones.

Thus, in experiment E12, the assessment of the entire *lamp* dataset is performed by training the estimation models with all the animations in the *hand* dataset. By comparing MAE and R^2 obtained in this case with those achieved in experiments E8–E10, it could be observed that performances of most of the estimation models trained with the *hand* dataset were superior to those experienced when performing the training with part of the new *lamp* animations. This trend is maintained when, in experiment E13, the *face* dataset is added to the training set. Almost all the considered estimation models showed improved performances compared to the case in which the training set included only *hand* animations.

In experiments E14–E16, the inclusion in the training dataset of a growing number of animations from the *lamp* dataset is studied, with the aim to simulate a scenario in which the teaching staff decide to score part of the newly available animations. Results showed that, on average, the correlation between automatic scores and teachers' evaluations keeps improving, as in previous cases. The same trend was confirmed also by experiments E17–E19, when the new sequences were added to two complete datasets.

In summary, results obtained with the above experiments showed that, independent of the considered model, estimation performances can be improved by adding new scored animations to the training set as they become available. However, in the considered scenario, none of the selected estimation models was able to outperform the other ones. Hence, the choice of the model best suited for a given assessment task remains an open issue. Nonetheless, experiments carried out so far provide early indications that could be considered in order to apply the designed assessment methodology to a new course. In particular, the bayesian linear regression proved to be the method with the highest average correlation between automatic and teachers' scores and the lowest error. Thus, it could be considered as a reasonable choice, especially when the size of the training set is small (E1–E12). When the size of the training set increases (E13–E19), attention should be shifted towards Poisson regression, which proved to be capable to achieve slightly better results in the considered conditions.

7 Conclusion

In this paper, a system for automatically evaluating technical skills required for producing 3D computer animations is presented. The proposed tool relies on the analysis of point clouds describing deformation of the considered geometries in order to cope with different techniques that could be used to produce the animation. A set of features is extracted from the point clouds to assess their similarity to a reference animation. Extracted features are then used for training a statistical regression model to fit human judgements.

An experimental phase was carried out by applying different estimation models on a testbed encompassing three set of students' animations. Results confirmed that, independently of the model used, the correlation between automatic scores and teachers' scores improves as the amount of training animations increases. Moreover, several indications about estimation models capable to provide the best performances in the considered scenario have been also obtained. The plan is now to start using the tool to assign the final grades of the considered Virtual Animation course and to deploy it also in other courses taught at Politecnico di Torino where a similar examination procedure is currently used.

Future works will be aimed to study the dependency of the designed methodology on the characteristics and size of the training dataset, as well as on the method for the extraction of point cloud data. Moreover, efforts will be devoted to the introduction of new functionalities required in order to use the tool also as a tutoring system during the course. In particular, it will be necessary to provide the students with a detailed feedback about error made in terms, e.g., of amount of deformation, timing, etc., by also relating them to the specific part of the animation where they have occurred. Lastly, research activities will focus on the possibility to link the evaluation of technical skills to the assessment of other competences required in a computer animation profile, e.g., related to creativity and expressiveness.

References

[1] A. Rasila, M. Harjula, and K. Zenger, "Automatic assessment of mathematics exercises: Experiences and future prospects," in *Proc. Symp. of Engineering Education*, 2007, pp. 70–80.

[2] R. Stanley, "A way to increase the engineering student's qualitative understanding of particle kinematics and kinetics by utilizing interactive web based animation software," *ASEE Computers in Education Journal*, vol. 20, no. 1, pp. 1–18, 2010.

[3] S. Romero, M. Guenaga, J. Garcia-Zubia, and P. Ordua, "An automatic assessment model for remote laboratories," in *Proc. Frontiers in Education Conf.*, 2014, pp. 1–4.

[4] 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. 5th Int. Conf. on E-learning and Games*, 2010, pp. 345–352.

[5] Y. E. Chen, C. L. Liu, T. H. Chang, and C. H. Lee, "An unsupervised automated essay-scoring system," *IEEE Intelligent System*, vol. 25, no. 5, pp. 61–67, Sep./Oct. 2010.

[6] X. Ying, M. Lingling, and Z. Hongying, "Design of online automatic assessment," in *Proc. Int. Conf. on Computer Science and Software Engineering*, 2008, pp. 1275–1278.

[7] T. Lahtonene and V. Isomottonen, "PARSI: A tool for automatic assessment of office documents and basic it skills," in *Proc. 11th Int. Conf. on Computing Education Research*, 2012, pp. 174–180.

[8] V. Petridis, S. Kazarlis, and V. G. Kaburlasos, "ACES: An interactive software platform for self-instruction and self-evaluation in automatic control systems," *IEEE Transactions on Education*, vol. 46, no. 1, pp. 102–110, Feb. 2003.

[9] G. Molnar, V. Omrcen, and M. Cupic, "Automating the formal logic course," in *Proc. of the 35th Int. Conv. MIPRO*, 2012, pp. 1124-1129.

[10] Z. Stanisavljevic, B. Nikolic, and J. Djordjevic, "A module for automatic assessment and verification of student's work in digital logic design," in *Proc. 19th Int. Conf. and Workshops on Engineering of Computer Based Systems*, 2012, pp. 275–282.

[11] G. T. Ferreres, K. Wouters, B. Ramos Alvarez, and A. R. Garnacho, "EVAWEB: A web-based assessment system to learn X.509/PKIXbased digital signatures," *IEEE Transactions on Education*, vol. 50, no. 2, pp. 112–117, May 2007.

[12] S. Srikant and V. Aggarwal, "Automatic grading of computer programs: A machine learning approach," in *12th Int. Conf. on Machine Learning and Applications*, 2013, pp. 85–92.

[13] M. Janidarmian, A. R. Fekr, K. Radecka, and Zeljko Zilic, "Affordable e-rehabilitation monitoring platform," in *IEEE Canada Int. Humanitarian Technology Conf.*, 2014, pp. 1–6.

[14] A. Aguirre, A. Lozano-Rodero, L. M. Matey, M. Villamae, and B. Ferrero, "A novel approach to diagnosing motor skills," *IEEE Transactions on Learning Technologies*, vol. 7, no. 4, pp. 304-318, 2014.

[15] Y. Heryadi, M.I. Fanany, and A.M. Arymurthy, "A method for dance motion recognition and scoring using two-layer classifier based on conditional random field and stochastic error-correcting context-free grammar," in *IEEE 3rd Global Conf. on Consumer Electronics*, 2014, pp. 771–775.

[16] A. Ahmadi, E. Mitchell, C. Richter, F. Destelle, M. Gowing, N.E. O'Connor, and K. Moran, "Toward automatic activity classification and movement assessment during a sport training session," *IEEE Internet of Things Journal*, vol. 2, no. 1, pp. 23–32, 2015.

[17] A. Siegmar and L. Debiasi, "Automated assessment in massive open online courses," *Seminar aus Informatik*, University of Salzburg, 2013.

[18] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, F. Ming, and F. Wong, "Learning about social learning in MOOCs: From statistical analysis to generative model," *IEEE Transactions on Learning Technologies*, vol. 7, no. 4, pp. 346–359, 2014.

[19] IITBombayX, Basic 3D animation using Blender [Online], Available: https://www.iitbombayx.in/courses/TISS/SKANI101x/2015-16/about, accessed on Sept. 15, 2016.

[20] D. Hooshyar, R. Binti Ahmad, M. Fathi, M. Yousefi, and M. Hooshyar, "Flowchart-based bayesian intelligent tutoring system for computer programming," in *Int. Conf. on Smart Sensors and Application*, 2015, pp. 150–154.

[21] C. Zhao, H. Yan, and C. Song, "A system for automatic assessment and plagiarism detection of student programs," in *Int. Conf. on E-Business and E-Government*, 2010, pp. 3619–3624.

[22] A. Lekshmi, and P. S. Nair, "SoftTutor - A XML based architecture for online questionnaire evaluation," in *Int. Conf. on Control, Instrumentation, Communication and Computational Technologies*, 2014, pp. 614–617.

[23] H. Jiang, L. Zhang, and W. Ye, "The automatic evaluation strategies and methods of multimedia work assignments," in *Int. Conf. on Comp. Intelligence and Software Engineering*, 2009, pp. 1–5.

[24] K. Forbus, J. Usher, A. Lovett, K. Lockwood, and J. Wetzel, "Cogsketch: Sketch understanding for cognitive science research and for education," *Topics in Cognitive Sc.*, vol. 3, pp. 648–666, 2011.

[25] Z. Shukur, Y. Away, and M. A. Dawari, "Computer-aided marking system for engineering drawing," in *Proc. of Society for Information Technology & Teacher Education Int. Conf.*, 2004, pp. 1852–1857.

[26] K. A. Hekman and M. T. Gordon, "Automated grading of first year student CAD work," in *Proc. ASEE Annual Conf.,* 2013, pp. 1–10.

[27] A. Sanna, F. Lamberti, G. Paravati, and C. Demartini, "Automatic assessment of 3D modeling exams," *IEEE Transactions on Learning Technologies*, vol. 5, no. 1, pp. 2–10, 2012.

[28] F. Lamberti, A. Sanna, G. Paravati, and G. Carlevaris, "Automatic grading of 3D computer animation laboratory assignments," *IEEE Transactions on Learning Technologies*, vol. 7, no. 3, pp. 280–290, 2014.

[29] F. M. Aymerich, N. Dessi, B. Pes, and A. Saba, "An automatic assessment system supporting computer science entrance examinations," in *Proc. IEEE Int. Conf. on Advanced Learning Technologies*, 2004, pp. 657–659.

[30] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Sepp, "Review of recent systems for automatic assessment of programming assignments," in *Proc. 10th Koli Calling Int. Conf. on Computing Education Research*, 2010, pp. 86–93.

[31] M. Amelung, K. Krieger, and D. Rosner, "E-Assessment as a service," *IEEE Transactions on Learning Technologies*, vol. 4, no. 2, pp. 162–174, 2011.

[32] J. Z. Kovacic and J. S. Green, "Automatic grading of spreadsheet and database skills," *Journal of Information Technology Education: Innovations in Practice*, vol. 11, pp. 53–70, 2012.

[33] D. H. Baxter and M. J. Guerci, "Automating an introductory computer aided design course to improve student evaluation," in *Proc. ASEE Annual Conf.*, 2003, pp. 1179–1180.

[34] M. Zaharieva, M. Zeppelzauer, D. Mitrovic, and C. Breiteneder, "Finding the missing piece: Content-based video comparison," in *Proc. 11th IEEE Int. Symp. on Multimedia*, 2009, pp. 330–335.

[35] S. D. Thepade and N. B. Yadav, "Assessment of similarity measurement criteria in Thepade's Sorted Ternary Block Truncation Coding (TSTBTC) for content based video retrieval," in *Proc. Conf. on Communication, Information & Computing Tech.*, 2015, pp. 1–6.

[36] S. Padmakala, G. S. Anandhamala, and M. Shalini, "An effective content based video retrieval utilizing texture, color and optimal key frame features," in *Proc. Int. Conf. on Image Information Processing*, 2011, pp. 1–6.

[37] P. Huang and A. Hilton, "Shape-colour histograms for matching 3D video sequences," in *Proc. IEEE 12th Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, 2009, pp. 1510–1517.

[38] S. Mittal, M. S. Karthik, S. Kumar, and K. M. Krishna, "Small object discovery and recognition using actively guided robot," in *Proc. 22nd Int. Conf. on Pattern Recognition*, 2014, pp. 4334–4339.

[39] H. Yuesheng and Y. Y. Tang, "Classification of 3D models for the 3D animation environments," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2009, pp. 3786–3791.

[40] Y. An, Z. Li, and C. Shao, "Feature extraction from 3D point cloud data based on discrete curves," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–19, 2013.

[41] S. Gumhold, X. Wang, and R. McLeod, "Feature extraction from point clouds," in *Proc. of 10th Int. Mesh. Roundtable*, 2001, pp. 293–305.

[42] A. R. Lee, H. I. Suk, and S. W. Lee, "View-invariant 3D action recognition using spatiotemporal self-similarities from depth camera," in *Proc. 22nd Int. Conf. on Pattern Recognition*, 2014, pp. 501–505.

[43] P. Ranacher and K. Tzavella "How to compare movement? A review of physical movement similarity measures in geographic information science and beyond," *Cartography and Geographic Information Science*, vol. 41, no. 3, pp. 286–307, 2014.

[44] G. E. Batista, E. J. Keogh, O. M. Tataw, and V. M. Souza, "CID: An efficient complexity-invariant distance for time series," *Data Mining Knowledge Discovery*, vol. 28, no. 3, pp. 634–669, 2014.

[45] Bulbul, A.; Capin, T.; Lavoue, G.; Preda, M., "Assessing Visual Quality of 3-D Polygonal Models," in Signal Processing Magazine, IEEE , vol.28, no.6, pp.80-90, Nov. 2011

[46] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

Gianluca Paravati (M'14-SM'16) is an Assistant Professor at Politecnico di Torino, Italy, where he received his Ph.D. degree in computer engineering in 2011. His research interests include real-time image processing, collaborative virtual environments, human-machine interaction, remote visualization, and distributed systems.

Fabrizio Lamberti (M'02-SM'14) is an Associate Professor at Politecnico di Torino, Italy. He received his M.S. and the Ph.D. degrees in computer engineering from Politecnico di Torino, Italy, in 2000 and 2005, respectively. His research interests are in the areas of computational intelligence, semantic processing, distributed computing, human-computer interaction, computer graphics, and visualization. He serves as an Associate Editor for IEEE Transactions on Emerging Topics in Computing and for IEEE Consumer Electronics Magazine.

Valentina Gatteschi is a Postdoctoral Research Assistant at Politecnico di Torino, where she received her B.Sc. and M.Sc. degrees in management engineering and the Ph.D. degree in computer engineering in 2005, 2008 and 2013, respectively. Her main research interests are in semantics and natural language processing. She has been involved in several European projects on education.

Claudio Demartini (M'02-SM'14) is a Full Professor at Politecnico di Torino, where he teaches information systems and innovation and product development. His research interests are in software engineering, architectures, intelligent systems, and education. He is the Chair of the Control and Computer Engineering Department and a member of the Academic Senate of Politecnico di Torino.

Paolo Montuschi (M'90-SM'07-F'14) is a Full Professor in the Department of Control and Computer Engineering at Politecnico di Torino, where he served as Chair of the Department from 2003 to 2011. His research interests include computer arithmetic and architectures, computer graphics, electronic publications, semantics and education, and new frameworks for the dissemination of scientific knowledge. He is an IEEE Fellow, an IEEE Computer Society Golden Core member, and serves as Editor-in-Chief of IEEE Transactions on Computers, as the Chair of the Computer Society Technical Achievement Award Committee, as a member of the IEEE Publications Services and Products Board, and the IEEE Products and Services Committee.