

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:

2021-09-17

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Baptista, F., Lopes, P. F. & Santana, P. (2016). MotionDesigner: Augmented artistic performances with kinect-based human body motion tracking. In Goncalves, G., e Bessa, M. (Ed.), 2016 23rd Portuguese Meeting on Computer Graphics and Interaction (EPCGI). Covilhã: IEEE.

Further information on publisher's website:

[10.1109/EPCGI.2016.7851195](https://doi.org/10.1109/EPCGI.2016.7851195)

Publisher's copyright statement:

This is the peer reviewed version of the following article: Baptista, F., Lopes, P. F. & Santana, P. (2016). MotionDesigner: Augmented artistic performances with kinect-based human body motion tracking. In Goncalves, G., e Bessa, M. (Ed.), 2016 23rd Portuguese Meeting on Computer Graphics and Interaction (EPCGI). Covilhã: IEEE., which has been published in final form at <https://dx.doi.org/10.1109/EPCGI.2016.7851195>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

MotionDesigner: Augmented Artistic Performances with Kinect-Based Human Body Motion Tracking

Filipe Baptista¹, Pedro Faria Lopes^{1,2}, and Pedro Santana^{1,3}

¹ISCTE - Instituto Universitário de Lisboa, Portugal

²ISTAR-IUL, Portugal

³Instituto de Telecomunicações, Portugal

Email: {fmsba,pedro.lopes,pedro.santana}@iscte.pt

Abstract—In the last two decades the use of technology in art projects has proliferated, as is the case of the interactive projections based on movement used in artistic performances and installations. However, the artists responsible for creating this work typically have to rely on computer experts to implement this type of interactive systems. The tool herein presented, *MotionDesigner*, intends to assist the design of these systems by providing artists with higher levels of autonomy and efficiency during the creative process, allowing them to specify the rules by which a human body interacts with both the audio and the visuals used in their interactive art work. The presented tool relies on an RGB-D camera to modulate the multimedia content according to the performer’s body motion. *MotionDesigner* is extensible so as to accommodate additions required by artists. The tool has been tested with dancers, choreographers, and architects. Results show that *MotionDesigner* is a valuable aid to artists, working as a catalyst of their creative process.

Index Terms—Human-Computer Interaction, Interactive Multimedia Projection, RGB-D Camera, Kinect, Dance, Art

I. INTRODUCTION

SINCE the 1950s artists and computer scientists work together to create art pieces for different media [1]. Nowadays, there is many art work based on the interaction between a person, typically an art performer, and a computer system [2][3][4]. The most explored type of human-machine interaction in artistic work involves the use of RGB-D cameras (or simply depth cameras) to capture the position and movement of the user/performer, allowing him/her to directly modulate projected multimedia content [5].

To create interactive art work, artists often collaborate with skilled programmers and software engineers to implement the envisioned system [6][5]. Currently, there are many different tools and libraries that help programmers and engineers to prototype and develop interactive art work [7][8][9][10]. Through early interviews with the target audience, we found that artists need to have immediate control over the projection content and over how the use of the technology is explored during the performance. This level of control is essential to enable improvisational work. This means that the technological tools used by the artist should be intuitively set, monitored, and adapted without having to rely on too many time-consuming artist-engineer interactions.

The tool herein proposed, *MotionDesigner*, is a computer program designed to allow creative people to autonomously

control how the audiovisual content required for a given interactive projection is modulated according to the body movement of a third-person (performer, person from the audience, etc.) by an RGB-D camera. A set of evaluation sessions conducted with performance and multimedia artists, as well as with architects, confirmed the ability of the tool to boost the creative process. By discarding the engineer-in-the-loop, *MotionDesigner* allows artists to rapidly design and test their ideas.

This paper is organized as follows. Section II presents the related work. Section III gives an overview of the proposed system, describing its principles, features and design. Then, section IV describes the evaluation method chosen to validate our work and the results of these evaluation tests with the final users. Finally, section V presents the final conclusions and proposes a number of features that can be added in the near future.

II. RELATED WORK

Before the RGB-D cameras were available on the market for artists and researchers, interactive systems based on motion capture were built using different technologies. For instance, O’Neal et al. [11][12] used motion capture suits to record dancers’ motion, which was then fed into their interactive system. Latulipe and Huskey [13] used, instead, portable USB mice which was held by the performers in order to send spatial input streams to the implemented system, which would translate these spatial data into visuals to be projected on a screen. Hewison et al. [14] used, instead, vision sensors integrated in the performers suits.

The appearance of RGB-D cameras opened a new space of opportunity for multimedia computing, allowing the implementation of systems based on motion-capture without the need for intrusive hardware [15]. Currently, one of the most commonly used depth camera is the Microsoft’s Kinect [16].

Interactive performance *.cyclic*. [2] uses depth cameras to capture the movement of a dancer so as to modulate computer graphics content. In *.cyclic.*, a pre-sequenced set of images to be iterated are synced with the music and drawn to the screen according to the performer’s position. Another related project, known as *Divided By Zero* by Hellicar and Lewis [4], considered an interactive dance performance that used a depth

sensor to track the dancers body silhouette, which would, in turn, affect the visuals that were generated in real-time. In this project the graphical elements projected were all computer graphics calculated and generated in real-time, and no pre-rendered video or preset of images were used. Conversely, the employed soundscape was pre-recorded, instead of generated or affected in real-time. Our system, instead, considers the transformation of digital sound samples, controlled according to the performer’s body motion.

Berg et al. [17] explored sound interactivity. They produced a music generator, in which each joint of the art performer was responsible to transform a specific sound sample. Joey Bargsten brought together interactive sound and graphical elements modulated by the feedback produced by a Kinect sensor [18]. He used PureData and Quartz Composer software to control audio and graphical content, respectively. However, his solution is based on two separate applications, one dedicated to sound and another to graphics. Conversely, our system provides the artist with the ability to simultaneous exploit computer generated graphics and sound.

MotionDraw [5] is one of the few examples of existing artists-oriented tools for creating interactive performances/installations. It is focused on enhancing the experience of the artist when conducting an interactive projection in real-time. MotionDraw only considers the graphical interactivity, i.e., no sound interactivity is explored. In MotionDraw, the artist can directly manipulate the visual aspect of the virtual scene - which conceptually is something similar to brushes painting on a canvas - by interacting with a simple GUI that allows to configure the aspect of the brush for each body joint being tracked by the Kinect sensor in real-time.

eMotion [19] is another example of a tool that was developed for the artists to directly manipulate the projection content by interacting with a simple and intuitive GUI. It allows displaying virtual objects chosen by the user and lets the body motion of a performer to interact with those virtual objects.

Table I provides a side-by-side comparison of the major features provided by MotionDraw, eMotion, and our system *MotionDesigner*. The table shows that both MotionDraw and eMotion do not handle more than one specific set of graphical objects (brushes painting on a canvas or a specific palette of virtual objects that can be used) and, consequently, cannot sequence multiple scenes exhibiting different content. Moreover, both tools do not include sound manipulation. Conversely, to enrich the artist palette, our system *MotionDesigner* provides several graphical and sound elements that can be sequenced and shaped according to the performer’s activity.

TABLE I
COMPARISON BETWEEN PREVIOUS WORK AND MOTIONDESIGNER

	MotionDraw	eMotion	MotionDesigner
Interactive Graphics	Yes	Yes	Yes
Multiple Scenes	No	No	Yes
Interactive Audio	No	No	Yes

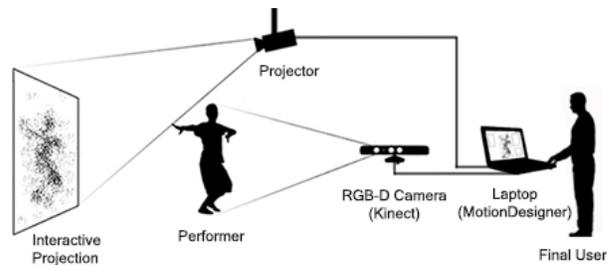


Fig. 1. System overview. The proposed system is composed of an RGB-D camera for full body motion capture, a projector that maps computer generated graphics onto a plain surface, and a software module (*MotionDesigner*) that allows the artist to control how visual and sound content is shaped by the performer’s activity.

III. THE PROPOSED SYSTEM

The tool herein presented, *MotionDesigner*, was developed having in mind the creative person as the final user of the software. The goal is to allow the artist to intuitively control how visual and audio content is projected onto the environment as a function of the performer’s (e.g., a dancer) full-body pose, without the need for any programming skills.

Figure 1 depicts the main elements of the proposed system. The system is composed of an RGB-D camera, a Kinect, for full body motion capture of the performer, a projector that maps computer generated graphics onto a plain surface behind the performer, and the software tool *MotionDesigner*, which allows the artist to specify, monitor, and adapt how visual and sound content is projected according to the performer’s activity.

Figure 2 depicts *MotionDesigner*’s software architecture, which is based on the software packages openFrameworks 0.8.4, OpenNI 2.2, NiTE 2.2 middleware, as well as ofxUI and ofxSecondWindow add-ons. The figure also shows how each software component integrates with both sensor and projector. The user interacts with the system through a GUI (implemented using the ofxUI add-on) to set the graphical aspect of the current graphical scene and the rules for the motion capture process. OpenGL renders the scene onto the screen according to the users parameterization. OpenNI is used to turn the camera’s depth sensor on when the user chooses to project the sequence, whereas NiTE is responsible for tracking the performer’s joints.

The following sub-sections describe the various elements that compose the software component of *MotionDesigner*, with an emphasis on the Graphical User Interface (GUI). The GUI was devised targeting simplicity and modularity. Simplicity is key to allow intuitive control over the multimedia content by non-programmers, whereas modularity facilitates future extensions to accommodate novel content.

A. User Interface

The GUI allows the artist to control how to sequence and shape the graphical and audio elements that are to be projected onto the environment according to the performer’s motion. The artist specifies the sequence of audiovisual content pre-

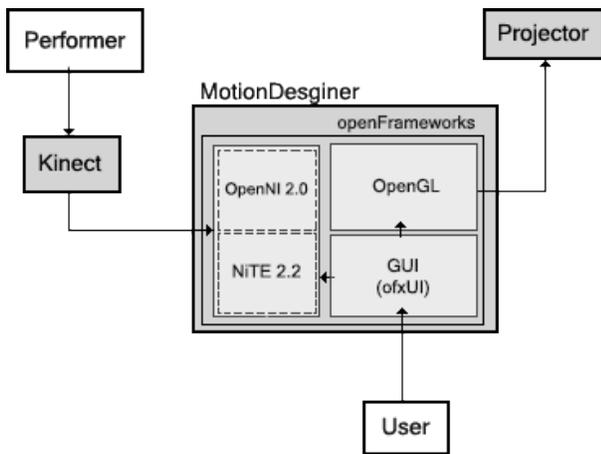


Fig. 2. System architecture. The performer interacts with the Kinect, which feeds the software with body movement data, processed by OpenNI and NiTE modules. The graphical content is rendered (through OpenGL) according to this data and the current parameterization set by the user through a GUI. A projector renders the whole graphical content in a plain surface.

sentation via the Editing Studio, a timeline-based environment depicted in Figure 3.

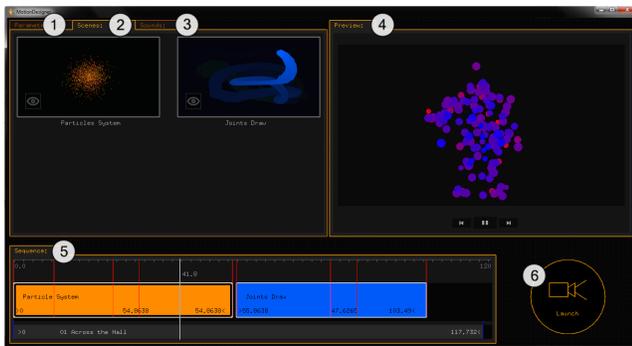


Fig. 3. Editing Studio interface. 1 - Parameterizations Panel, through which the user can add timestamps to a scene from the timeline and set the desired parameters for that time instant; 2 - Scenes Palette, which displays the graphical scenes the user can drag to the timeline; 3 - Sounds Palette, which displays the sound samples the user can drag to the timeline; 4 - Preview Player, where the user can preview the scenes representation over time according to the current parameterisations; 5 - Timeline, where the user can sequence the graphical scenes and set their duration; 6 - Launch Button, where the user can play/launch the sequence onto the projector.

The Editing Studio is the first environment to be presented to the artist, and is the software's main interface. In this environment the user can drag to a timeline a set of graphical scenes (visual content) to be projected and sounds samples to play along with the projection. The panel with the graphical scenes and the panel with the audio files are displayed according to the tab currently selected by the user. The user may manipulate the projection time of each audiovisual element and set their order of appearance in the timeline.

Besides sequencing the audiovisual elements and set their duration, the user is also able to set timestamps/markers during the projection time of a scene (represented by red lines on the timeline) and associate to that instant a set of values for the parameters associated to a given audiovisual content (e.g., the sound level). Setting these parameters can be done

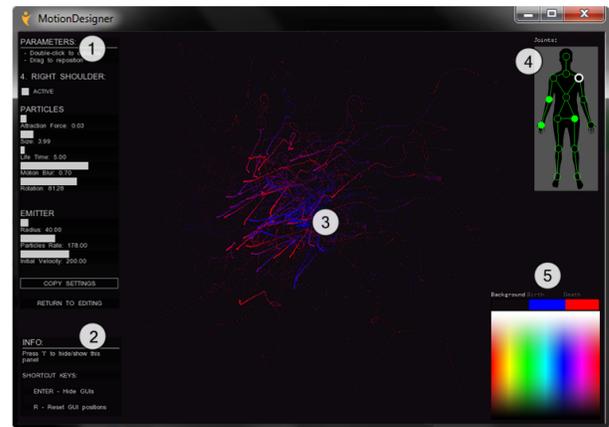


Fig. 4. Scenes editing interface. 1 - Parameters Control Panel, sliders-based panel where the user can set a different value for each scene parameter; 2 - Info Panel, panel with useful info, namely short key commands; 3 - Scene Content, where the scene is displayed as is being rendered on the projection; 4 - Joints Selector Panel, where the user can choose which of the performers body joint is going to be parameterized (how much that joint affects the projection); 5 - Color Picker, where the user sets the color for each scene element

offline, prior to the performance, and online. This allows the artist to plan ahead the performance and, yet, leave room for improvisation.

B. Interactive Graphical Scenes

To facilitate the creative process, the artist needs an interface to freely experiment the effect of applying a given set of parameters to the projected content (e.g., to test which color of a given visual content is the most adequate to a given moment in the performance or to a given pose of the performer). This trial-and-error process is done through the Interactive Scenes Editing GUI, depicted in Figure 4.

The Interactive Scenes Editing GUI is presented to the user whenever the *Explore* button is pressed in the Editing Studio interface. This button is represented in Figure 3 as an eye icon on the bottom left corner of each scene from the scenes palette. When the user presses this button, the scene is rendered in full screen along with this GUI, so that the user is able to test the scene before the live performance. When the user decides to live project the content used in the timeline, the scenes are accordingly projected and the user can see the current scene content replicated on the computer screen along with this editing GUI, which allows real-time, live manipulation of the scene.

When using *MotionDesigner*, the user is presented with a set of graphical elements available in a library. These are the elements the user can use to build the graphical scenes. To determine which graphical elements the software should provide, a set of early interviews with a group of people from the target audience were carried out. As a result of these interviews, the library was populated with the following graphical elements:

- 1) **Particle System** - a set of 3D floating particles that stochastically track the performer's body joints in the visual canvas (see Figure 5);

- 2) **Joints Draw** - a set of 3D brushes associated to the performer's body joints that leave a vanishing trail on the visual canvas as the performer moves (see Figure 6).

1) *Particle System*: In the Particle System scene, the user is presented with a particle emitter, i.e., a point in the screen space from which the particles arise. These particles are colored filled circles which move independently from each other and will track a specific joint from the skeleton(s) detected by the depth camera. When each of the particles is born (spawned in a random position within the emitter radius) they are associated to a specific body joint. When associating the targets of each particle, each possible body joint is iterated from the first (the head) to the fifteenth (the right foot) and therefore, the first fifteen particles to be born have different target joints, the sixteenth has the same as the first, the seventeenth has the same as the second particle, and so on. Although each joint is linearly iterated, the user can deactivate the desired body joints from the joints selector panel (see top right corner of the UI in Figure 4) and these will not be considered as possible targets, being ignored by the algorithm that associates the particles with the skeleton joints. The parameters that characterize this scene as a whole, which are the variables adjustable by the final user, are [20]:

- 1) **Attraction Force** - the intensity with which the point of attraction (body joint) pulls the particles;
- 2) **Particles Size** - the size of the particles radius (in pixels);
- 3) **Life Time** - time that the particles take to disappear from the scene (in seconds);
- 4) **Motion Blur** - visibility of the trail left by the particles motion;
- 5) **Rotation** - centrifugal force of the attractor, i.e., controls the speed and orientation with which the particles orbit the target;
- 6) **Emitter Radius** - the size of the emitter radius (in pixels);
- 7) **Particles Rate** - the number of particles ejected by the emitter (in particles per second);
- 8) **Initial Velocity** - the maximum velocity with which the particles are ejected from the emitter.

By setting different values to each of these parameters, through the scenes editing UI, the user will affect the aspect and behavior of the particles being drawn and computed by the responsible algorithms, in real-time.

2) *Joints Draw*: In the Joints Draw scene the performers skeleton joints act like brushes painting on a canvas, which is the screen. Circle primitives are drawn to the screen in the position of each of the traceable skeleton joints. These circles are translated according to the associated skeleton joint position and the previous frames (which illustrate their previous positions) are stored in a buffer, which renders these frames with transparency (the older the frames the greater the transparency). This allows a simulation of a brush painting on a canvas, with the result of that painting being gradually erased as time passes. In this type of graphical scene, the parameters configurable by the user are:

- 1) **Brush Size** - the size of the brush radius (in pixels);

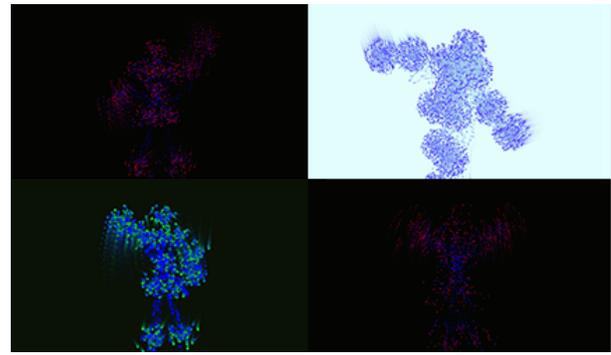


Fig. 5. Particle System. The particles track the performers skeleton joints and orbit them forming an anthropomorphic swarm of particles, which is more or less recognizable as a human form depending on the current active joints and their parameterization.

- 2) **Trail** - the perseverance of the trail left by the brush motion (how many seconds does it take for the drawing to be erased);
- 3) **Drawing Speed** - the intensity with which the joints move the brushes.

Experimenting with these parameters values allows the user to achieve many different interesting scenarios. Figure 6 illustrates some possible imagery that can be achieved by interacting with this scene.



Fig. 6. Joints Draw. Some abstract shapes can be drawn on the screen by having a performer moving around and conducting each brush move on the canvas/screen. By changing the parameters values the user can set different aesthetics to drawings

C. Interactive Audio

In addition to the interactive graphical scenes, *MotionDesigner* also provides the artist with the possibility of playing audio samples along with the projection of the graphical elements. The developed software allows the users to modulate a set of audio properties of a given sound file, e.g., WAV or MP3 file, according to the performer's body joints when standing in front of the depth camera: (1) volume; (2) speed; and (3) panning.

The user can also choose which body joint of the performer will affect each of the available sound parameters. For instance, the head's position can be set to control the volume whereas the right hand's position to control the panning.

Position information can be defined in terms of displacement along a given axis of the coordinate system or as a distance between the joint and a given point in space. Distances can be computed in the 3-D world coordinate system or on the screen (projected) 2-D coordinate system.

An example of a possible setting is: the position of the head along the y-axis controls the sound volume, the position of the left hip along the x-axis will control the sound speed, and the distance (in pixels) from the x coordinate of the torso to the center of the screen will dictate the sound panning. This allows the user to explore different parameters for the audio content in real-time.

The audio files the user wants to play during the live projection can be dragged and placed in the Editing Studio's timeline, similarly to the graphical scenes case, and parameterized through a sliders based interface. This audio editing interface can be accessed through the same *Explore* button used in the graphical scenes palette. When the interactive audio editing scene is called, the loaded sound file is immediately played. The sound engine loads and plays the sound sample in a loop until the user exits the audio editing environment. The user can pause the sound sample playback at any time by pressing the Space key on the keyboard. The GUI for this audio editing environment can be seen on Figure 7.

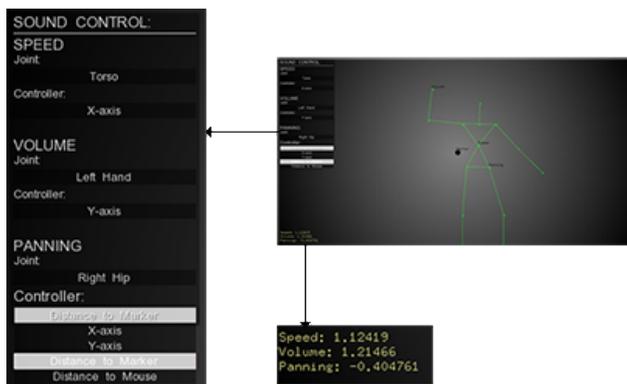


Fig. 7. Interactive Audio Editing GUI. The user can set the joints responsible for transforming the properties of the sound sample being played during the projection. The panel on the left allows the user to associate a specific joint to a sound property (speed, volume and panning) and specify its position reference. On the bottom left corner the current values for each sound property are shown.

IV. EVALUATION AND DISCUSSION

MotionDesigner has been tested with ten people from the target audience, including choreographers, dancers, architects and multimedia artists interested in creating interactive projections for live performances or artistic installations. The ages of the testers varied between 19 and 33 years and each of them participated in the test session without the presence of any other person in the room, apart from the development team. This means that each test session was private and the testers were unaware of the software they were about to evaluate.

The majority of these tests (six out of ten) were carried out at the end of the development process in order to validate the system as a whole. The remainder tests (four out of

ten) were carried out as check points of the development process. These intermediate tests were pivotal to better match the requirements of the target audience. Due to the user's creative facet, setting user requirements just from preliminary interviews would be insufficient. Users need to experience the tool and check how it influences their creative process. Only then they are able to provide sufficiently detailed feedback.

A. Evaluation Method

To test *MotionDesigner*, different evaluation sessions were prepared. On each evaluation session a user (e.g., a choreographer) was asked to perform a set of tasks using *MotionDesigner*. These tasks involved parameterizing and sequencing the interactive graphical scenes (both the particle system and the joints draw), using the timeline of the editing studio, and interacting with the sliders-based interface for parameterizing each scene as it was being rendered in real-time.

Each evaluation session started with a brief explanation of the premise behind the creation of the tool the testers were about to use and the purpose of its implementation. Then, the testers were immediately invited to use a laptop that was running a build of the software and they were asked to perform some tasks in the Editing Studio. These tasks consisted on dragging a graphical scene to the timeline (the user was allowed to choose between the *Particle System* and the *Joints Draw* scenes), setting the duration of that scene, changing the instant at which the scene begins, dragging an audio file to the timeline, and repeating the previous operations for the sound files.

Once the users finished interacting with the Editing Studio environment, they were introduced to the Interactive Scenes Editing GUI. The testers were asked to run each of the Interactive Scenes Editing GUI by clicking on the *Explore* button (in the order they wanted) and perform the following set of tasks for each of the two scenes provided.

For the *Particle System* scene, the tester was asked to change the behavior of the particles and their emitter by interacting with the parameters panel. Testers had freedom to change the parameters they wanted to, but we suggested them to try to change the particles size, increase the area where the particles are born, change the particles rotation, decrease the delay between the movement of the performer and the movement of the particles, and decrease the number of particles in the screen. Once this interaction with the parameters panel was finished, the tester was asked to change the color of the scene's background and the color the particles should have when they are about to die. Finally, the tester was asked to change the number of joints that affect the particles motion, by making the head and the hands of the performer the only joints that affect the projection content (although they had the freedom to perform other joints combination).

For the *Joints Draw* scene, the tester was asked to try to deactivate the head brush, change the size of the hands brushes, changing the color of the brushes and the background, and, finally, making the drawing to be erased faster. During this phase of the evaluation, a dancer from the development team acted as the performer in front of the Kinect sensor, while

the tester was operating the computer. This was done so that the content being projected had a real-time response to the movement of a third person, as it is supposed to. The dancer performed large movements with the arms and legs and moved around the room from left to right, approaching the floor only a few times. Through this phase the tester could refine each scene in real-time as the dancer was affecting the projected graphics, until he/she achieved the desired result.

At the end of the test session a small questionnaire was handed to the users who tested the last version of the software, i.e., to 6 testers out of the total 10, in order to better understand how was their experience of interacting with *MotionDesigner* and how it helped them fulfill their ideas. The results extracted from these questionnaires are presented next.

B. Results

One of the questions asked to the testers addressed directly the intuitiveness of the presented system. Figure 8 shows that all the users that tested the final version of the software considered that our tool was relatively intuitive to use. However, 33.3% of the testers considered that, despite of being intuitive, there are some minor changes that could be done to the GUI in order to boost even more the user experience. They suggested a few design decisions, such as changing the position of sliders labels from underneath to above the slider itself, as well as providing pop up text notes to help the understanding the button's function.

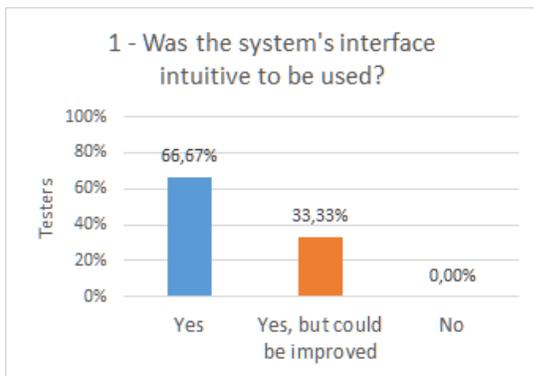


Fig. 8. Testers feedback regarding the intuitiveness of the system.

After confirming among the testers that the ability to materialize ideas is a fundamental process of any creative process, we asked the users if they felt any difficulty in carrying out their own ideas for the interactive projection when using the set of elements we provided them, concretely, the *Particle System* and *Joints Draw* scenes. Figure 9 shows that all the users felt they successfully achieved the results they were aiming at when creating their own interactive projections using the presented tool. Some testers (33.3%) felt that, mostly due to the possibility of performing interactive experimentation, the tool helped them in the creation of new ideas. This is a very useful aspect since artists often exploit some exploratory design.

In order to confirm whether the developed tool really facilitates the creative process of interactive art work, the testers

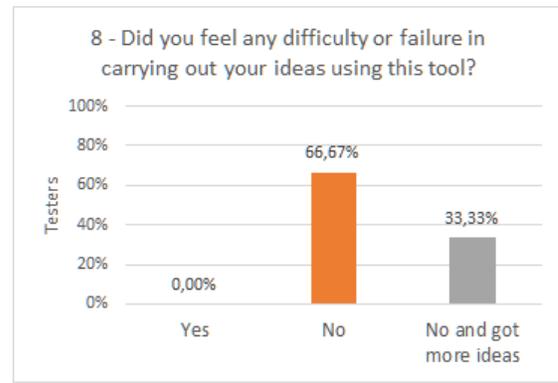


Fig. 9. Testers feedback on how easy was to realize their ideas with *MotionDesigner*.

were also directly asked if they felt this process facilitation. All testers answered positively, as seen on the results depicted in Figure 10. That is, the tool works as a catalyst of the creative process. However, 16.6% of the testers felt the tool as having a somewhat steep learning curve, mostly due to the difficulty of mastering the manipulation of each system's component. Despite this, they immediately stated that after some time spent interacting with the interface they could gradually get more efficient and productive. They also suggested making a brief introduction tutorial when running the software for the first time, which is something considered for further software iterations.

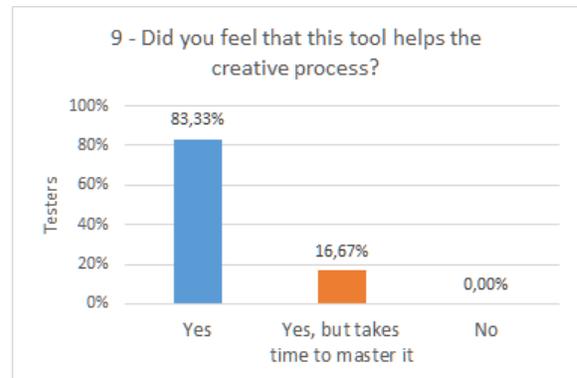


Fig. 10. Testers feedback on how much *MotionDesigner* boosts the creativity process.

After finishing the questionnaire, the users had freedom to suggest other features or simply give an overall feedback. Most of the users did not cover any issue that they did not address already when answering the questionnaire. However, some testers said they also wanted to try acting as the performer in front of the depth camera, interacting directly with the projected content. Two of them, which are dancers, experimented the system as performers for almost twenty minutes.

One of the dancers that tested our software, which is a dance degree student, showed interest in participating in the creation of a small interactive performance. Therefore, the developed tool was used to create a small dance performance, in which both the particle system scene and the joints draw scene were used as the projection content. Figures 11 and 12 show some

moments of the performance rehearsal with the dancer from our team acting as the performer and one of us as the conductor of the performance. All the projection aesthetic was found by following the dancers artistic vision.

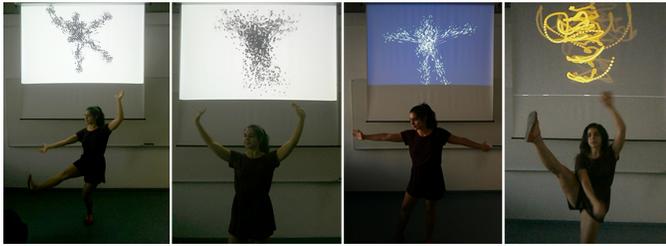


Fig. 11. Rehearsal moments for the small interactive performance that was prepared. The dancer is performing a choreography while the graphics projected on the wall react to her movement in real-time. On the first three images the Particle System scene is being projected and on the last one is the Joints Draw scene. The user can mirror the projection content in relation to the body pose if wanted, as seen on the first image.

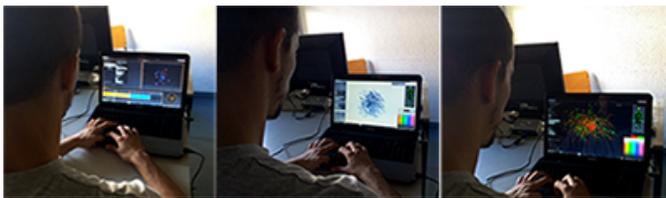


Fig. 12. The user sequences the scenes he is going to use in the projection and sets a few initial parameterizations (left image), then he controls the representation and behavior of the scenes by changing the parameters values in real-time.

V. CONCLUSION

MotionDesigner, a tool to assist the design of interactive projections was presented. The tool allows creative people (e.g., dance choreographers) to prepare performances in which multimedia (visual and audio) content is shaped and projected onto the environment according to the motion of a human performer (e.g., a dancer), estimated using a depth camera. A set of evaluation sessions with artists and architects showed that the tool is intuitive to use and works as catalyst of new ideas. Most importantly, the users managed to develop their ideas autonomously, that is, without the support of an engineering team. This allows artists to cut development time and avoid distractions that could otherwise hamper a proper creative process.

As future work, we intend to allow the use of more than one depth camera. This is essential to enable larger scale performances. We also intend to include a mechanism so that specific body poses can be used to start and end a given scene, rather than relying solely on time for that purpose. Finally, we also consider to include a basic sound synthesizer module or implementing OSC routers to allow integration of third-party software for music production, such as Ableton's Live [21].

ACKNOWLEDGMENT

We would like to thank the dancers, choreographers, architects, and multimedia artists that so kindly accepted to partic-

ipate in the evaluation sessions. The authors wish to thank the fruitful comments provided by the anonymous reviewers.

REFERENCES

- [1] A. Trifonova, L. Jaccheri, and K. Bergaust, "Software engineering issues in interactive installation art," *International Journal of Arts and Technology*, vol. 1, no. 1, pp. 43–65, 2008.
- [2] D. Jung, M. H. Jensen, S. Laing, and J. Mayall, "cyclic.: an interactive performance combining dance, graphics, music and kinect-technology," in *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*. ACM, 2012, pp. 36–43.
- [3] "Pixel - extrains on Vimeo," <https://vimeo.com/114767889>, 2015, [Online; accessed 21-September-2015].
- [4] P. Hellicar and J. G. Lewis, "Divided By Zero, Hellicar and Lewis Ltd.," <http://www.hellicarandlewis.com/divide-by-zero/>, 2015, [Online; accessed 19-January-2015].
- [5] D. G. Rodrigues, E. Grenader, F. d. S. Nos, M. d. S. Dall'Agnol, T. E. Hansen, and N. Weibel, "Motiondraw: a tool for enhancing art and performance using kinect," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 1197–1202.
- [6] B. Gonzalez, E. Carroll, and C. Latulipe, "Dance-inspired technology, technology-inspired dance," in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. ACM, 2012, pp. 398–407.
- [7] "openFrameworks," <http://www.openframeworks.cc/>, 2015, [Online; accessed 19-January-2015].
- [8] "Processing.org," <https://processing.org/>, 2015, [Online; accessed 19-January-2015].
- [9] "Cinder," <http://libcinder.org/>, 2015, [Online; accessed 19-January-2015].
- [10] "Max is a visual programming language for media Cycling 74," <https://cycling74.com/products/max/>, 2015, [Online; accessed 19-January-2015].
- [11] W. S. Meador, E. M. Kurt, and K. R. O'Neal, "Virtual performance and collaboration with improvisational dance," in *ACM SIGGRAPH 2003 Sketches & Applications*. ACM, 2003, pp. 1–1.
- [12] W. S. Meador, T. J. Rogers, K. O'Neal, E. Kurt, and C. Cunningham, "Mixing dance realities: collaborative development of live-motion capture in a performing arts environment," *Computers in Entertainment (CIE)*, vol. 2, no. 2, pp. 12–12, 2004.
- [13] C. Latulipe and S. Huskey, "Dance. draw: exquisite interaction," in *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 2*. British Computer Society, 2008, pp. 47–51.
- [14] H. Bailey, J. Hewison, and M. Turner, "Choreographic morphologies: digital visualization and spatio-temporal structure in dance and the implications for performance and documentation," in *Proceedings of the International Events in Visual Arts (EVA Conference), London, 2008*, pp. 9–18.
- [15] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proceedings of the IEEE on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2000, pp. 126–133.
- [16] Z. Zhang, "Microsoft kinect sensor and its effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.
- [17] T. Berg, D. Chattopadhyay, M. Schedel, and T. Vallier, "Interactive music: Human motion initiated music generation using skeletal tracking by kinect," in *Proc. Conf. Soc. Electro-Acoustic Music United States*, 2012.
- [18] J. Bargsten, "Gesture and Performance with Kinect, Quartz Composer, and PureData," http://fau3711.pbworks.com/w/file/attach/76805993/Gesture_and_Performance_with_Kinect_Quartz_and_PD.pdf, 2013, [Online; accessed 19-January-2015].
- [19] J. Backhus, "Dancer Bends Light in Stunning Projection-Mapped Performance — The Creators Project," http://thecreatorsproject.vice.com/blog/dancer-bends-light-in-stunning-projection-mapped-performance?utm_source=tcpsfbus, 2015, [Online; accessed 21-September-2015].
- [20] D. Perevalov, *Mastering openFrameworks: Creative Coding Demystified*. Packt Publishing Ltd, 2013.
- [21] "What is Live? Learn more about Ableton's music making software — Ableton," <https://www.ableton.com/en/live/>, 2015, [Online; accessed 29-September-2015].