Avatar Staging: an evolution of a real time framework for theater based on an on-set previz technology.

Cédric Plessiet
INREV AIAC (EA4010)
Université Paris VIII
France
cedric.plessiet@univ-paris8.fr

Georges Gagneré
Scènes du monde, création, savoirs critiques(EA1753)
Université Paris VIII
France
george.gagnere@univ-paris8.fr

Remy Sohier
INREV AIAC (EA4010)
Université Paris VIII
France
remy.sohier@univ-paris8.fr

ABSTRACT

We will describe in this article our work on a framework mixing game, mixed reality, and motion capture technology for the theater. It will focus on two main technical problematics (motion retargeting, event-based control) and a new job-oriented organization leading to the arrival of new roles in the staging team.

CCS CONCEPTS

• **Human-centered-computing** → **Interaction design** → Empirical studies in interaction design; • **Computer systems organization** → **Real-time systems** → Real time system architecture; • **Applied computing**→ **Arts and humanities**→ performing arts • **Computing methodologies** → **Computer graphics** → **Animation** → Motion capture

KEYWORDS

Mixed reality, Motion capture, Performing art, Avatar

ACM Reference Format:

Cedric Plessiet, Georges Gagneré and Remy Sohier. 2018. Avatar Staging: an evolution of a real time framework for theater based on an on-set previz technology In *Virtual Reality International Conference - Laval Virtual VRIC '18 (VRIC '18), April 4–6, 2018, LAVAL, France,* Simon Richir (Ed.). ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3234253.3234300

VRIC '18, April 4–6, 2018, LAVAL, France © 2018 Association for Computing Machinery.ACM ISBN 978-1-4503-5381-6/18/04…$15.00 https://doi.org/10.1145/3234253.3234293

1 INTRODUCTION.

After conducting research on **on-set previs** for cinema[1], we decided to reorient our work around the theater, this in order to refocus on the work of the actor, which we had a little lost in our previous work.

This thought led us to reorganize some of our work, and develop new tools around capturing the body and triggering events. Our work led us to the need to create new professions and to rethink the roles of the plateau.

2 AN EVOLUTION OF AN ON-SET PREVIZ FRAMEWORK APPLIED TO THEATER PROBLEMATIC.

In a previous work we have developed an on-set previz platform in which a real actor could immerse himself in a virtual world, that allow actors, on stage director and the other members of filming crew to collaborate in the early phase of the conception of a movie [2]. This platform was built around different positions (actor, director, director of photography) and allow rapid prototyping of a scene before the final shooting.

The modular approach of the set allowed us to reuse much of our development and the methodology established earlier in this project to a new field: the theater. The choice of applying this technology to the theater is due to the necessity to refocus some of our problematic to the actor himself, and especially to the "real-time aspect" of his work and its skill of adaptation and improvisation.

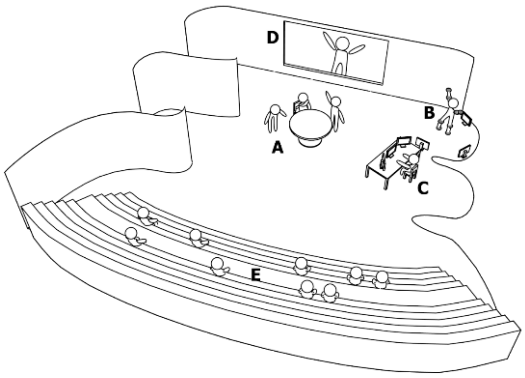

Figure 1: Avatar staging organization with the (A) actor[1], located in the center of the stage; (B) The mocaptor[2] who controls the avatar; (C) The digital artist (including manipulactor[3]) who works on the computer and uses midi and joystick devices;(D) Avatars performing on the screen.

In the experimental setup, the staging is organized in two parallel levels that strongly interact: the actor moves in the physical space in connection with the avatar who moves in the 3D space. The physical staging uses the theatrical rules of the theater, but the 3D real time avatar staging needs specific tools:

- Positioning the avatar independently of the *mocaptor* position to organize the way the avatar appears in the image through the 3D scene
- Correcting the *mocaptor* movement in real time in location and rotation to keep eye contact between actor and avatar independently from the *mocaptor*
- Using IA pathfinder system to combine complex avatar movements in 3D scenery with constrained *mocaptor* movements on the physical stage.

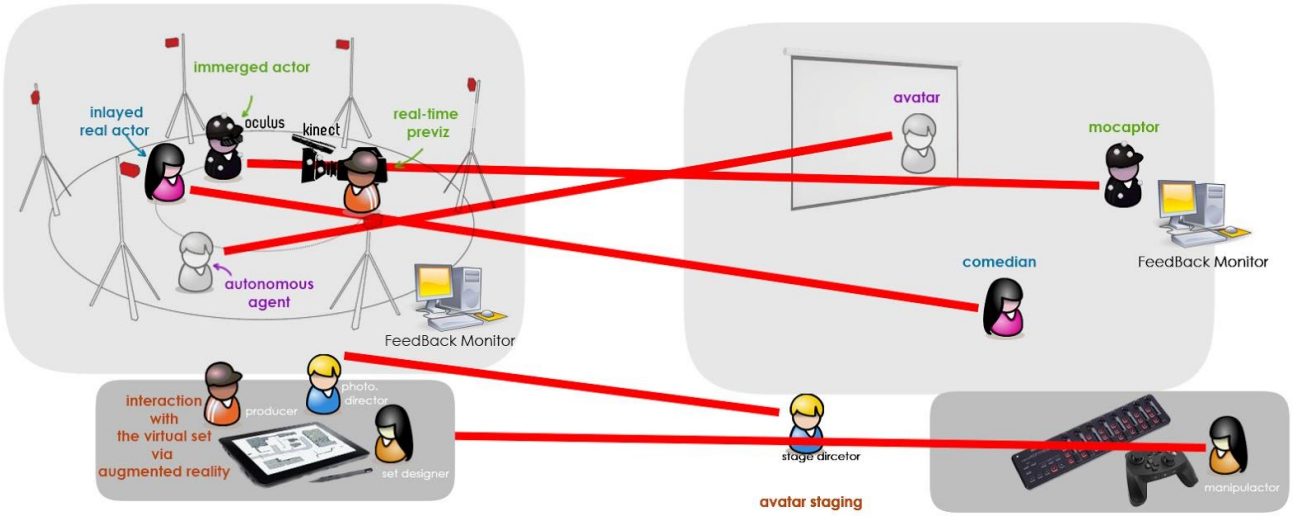

Figure 2: the adaptation of the outilnum to the avatar staging setup

Indeed, rather than ad hoc development, we have rather rearranged our modules to resolve to new issues. This is the reason why we can note some similarities between these two projects.

The first similarity is a multi-client / multi-server. Indeed, our device has several servers, each dedicated to a specific task. It shares with the entire device. We have such a world server that manages the 3D world, one or more servers that manages motion capture, one or more dedicated to the set designer.

In our new architecture the immersed actor becomes the *mocaptor*, and no more need a head-mounted display to immerse himself to a virtual actor, the inlayed actor becomes the actor, and we need no more to inlayed himself in the virtual world, the photo director becomes stage director and the set

[1] *Actor*: the physical actor who acts with the avatar projected on the screen

[2] *Mocaptor*: by mocaptor we design the physical actor who controls the avatar through a motion-capture suit

[3] *Manipulactor*: the artist who controls the avatar through a device (midi device e.g. Korg Nanokontrol or gamepad e.g. Microsoft Xbox Controller)

designer become the *manipulactor* and the autonomous agent become a simple avatar that is driven by the *mocaptor*.

Figure 3: the mocaptor, the actor and hidden behind the manipulactor

If the new avatar staging setup is simple to the **outilnum** setup, we have to refine the avatar driven mechanism and a more powerful event system. Because, where in **outilnum** we have only a prototyped previz tool, avatar staging was used as student learning support for course and theater creation. By the way, we have to enhance a certain number of tools.

3 ACTING INPUT, PUPPETEER AND AVATARS.

First of all, the movement of the avatar is the sum of three kinds of input:

- The *mocaptor* input: a motion issued from a motion capture device (optical system, kinect system, or inertial matrix motion-capture suit).
- The *manipulactor* input: the action that the *manipulactor* can give from a keyboard, a joystick, a midi system or another.
- A computer-driven input like a pathfinding algorithm, or a look-at algorithm

So we decide to represent all this kind of input by an "***acting input***" object. This *acting input* can drive the reference position of the avatar, the rotation of the full body, or the rotation of a specific bone of the body. We can easily combine each acting input to adapt the control of an avatar depending on the need of the scene. They are based on a discrete event system[7] that use boxed value[9]

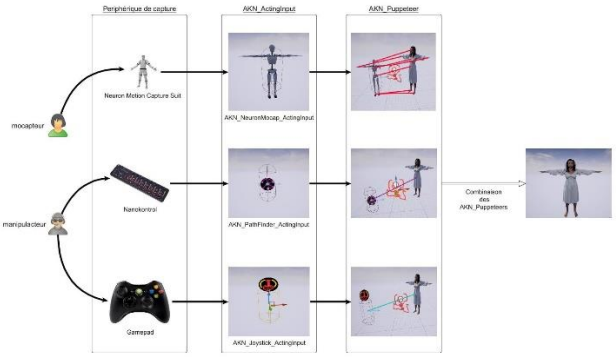

Figure 4: three different acting input and the puppeteer

We developed a motion retargeting system to control different avatars with the same real actor. To do that we developed a system that normalizes the animation issued from an *acting input* by applying it to a neutral character (for the moment, the default unreal "robot" neutral character, but we will create our own one, by adding facial data).

We direct later a second motion retargeting to project the movement of the unified avatar on a controlled virtual character. There are two common types of major motion retargeting, who focuses on the transcription of the movement of different body parts, and those who focus on the purpose of the movement (grabbing an object for example). We decide to use and mix these two approaches, by combining the motion retargeting in two phases, the first phase goal is to formalize data coming from different devices, the second to retarget this movement to the body of the avatar.

The first part of the algorithm works is based on a profile which makes links between the data send by the motion capture device and the bones of the neutral character. We have to take care of graduation change and rotation problems (gimbal lock…). So, each device has is own specified motion retargeting algorithms to the neutral character. The second has to solve the motion retargeting strictly speaking by sending the data from the neutral character to the avatar.

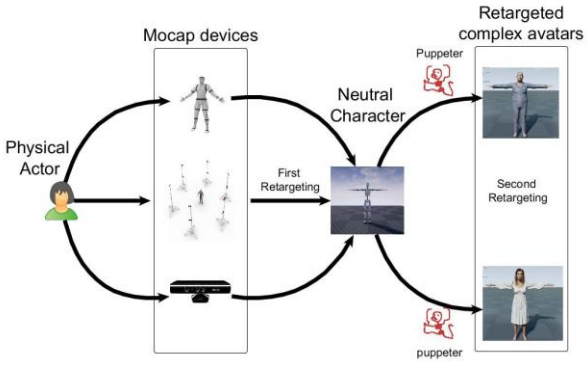

Figure 5: the motion retargeting in two steps

In OutilNum, we have developed an algorithm call "*comportmental layer*" that allows us to combine movement from different sources of motion. It will not answer our problematic. First of all, because the *comportmental layer* mix the animation of a full body, second, because the ergonomic of the previous **outilnum** project was based on an iterative process: we make a movement, and we correct it by adding or correct the recorded movement to obtain a new movement that will be the base of the new iteration.

But in **avatar-staging**, this methodology is not adapted to our problematic, it is why we have developed the *puppeter* concept. We decide to call it *puppeter* and not puppeteer, to make a difference between the job "puppeteer" and our concept, that even if there is a strong connection with the puppet concept has a slightly different philosophy, for example the fact that the *puppeter* will operate a second motion-retargeting.

When using a *puppeter*, we can decide which part of the avatar will be driven by, so for example we can affect the movement of the avatar's head by a *puppeter*, the movement of the left arm's avatar to a second **puppeter**. We can go further by deciding that a specific part will be the addition of two or more different **puppeters**. For the moment we have only three **puppeter's** mode: *addition*, *ignore*, *replace*, but we can imagine more mode like for example *interpolation*, *min max* and *subtraction*.

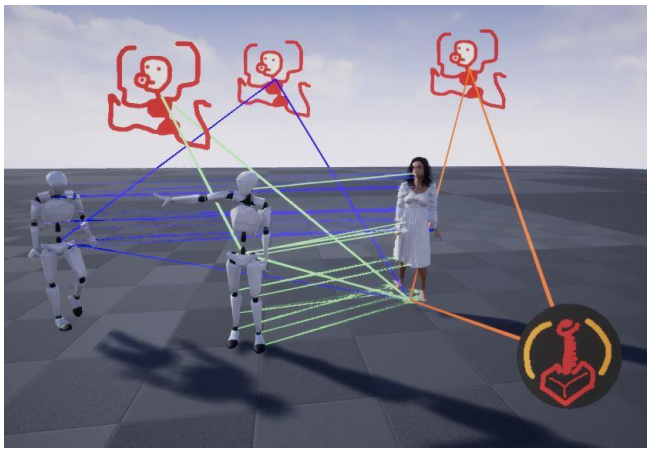

Figure 6: an example of an avatar driven by different pupetters, combining parts of two "full body acting input", and a "path finder acting input"

4 NEW USES.

Digital involvement.

In the digital 3D scene we used for our experimentations, the combination of both acting from actors, *mocaptors* and *manipulactors* requires building combinations of *acting inputs*, *puppeters* and avatars that follow real-time interactions, under the control of the stage director. The programming needs to evolve with the creative proposal of the varied collaborators of the process (stage director, actor, *mocaptor*, *manipulactor*, digital artist). And the first and fundamental creative responsibility for the digital artist is to keep reactive the real-time rehearsal process, that means coding stable modules fitting the avatar staging. In a second step, the modules are mixed together and parametrized according to the evolution of a storyboard.

Ronfard[6] indicates that in some theatrical direction methods, the tech manager devellope an expert assessment in the organization of the movement on stage that can be useful in the creation of an event management system applied to the movement in a 3D space. We propose an event manager architecture that uses goals to organize the blocking of the avatars and the framing of the camera points of view, and cue lists that allow to write the interactions between 3D effects. The main characteristic of this system is that it is not temporarily continuous as in a cinematic game moment, but organizes non-continuous events [7]

The avatar direction needs a deep understanding of motion capture constraints, decomposition of the avatar movement between *mocaptor* and *manipulactor* inputs, setup of 3D objects transform and assets. What the director sees is the result of a long programming work that needs a minimum of anticipation to respond to the creative imagination. We suggest that directing an avatar as a theatrical director made with an actor requires basic programming skill to play with the 3D constraining rules.

This results in the new position of the avatar director combining theatrical and programming skills. One aspect of this new function is the *manipulactor* position, that increase acting possibilities for the *mocaptor*.

Balance between mocaptor and manipulactor interactions for avatar staging.

One of our main issues was to allow movement freedom for the avatar in a constraint acting space for the *mocaptor*: indeed, the *mocaptor* could not use all the stage, mainly dedicated to the actors in front of the screen representing the 3D scenery and the avatar. We usually worked with two avatars, acting together relating to two actors acting on the physical stage.

This distribution allowed development of dramatic conflicts either in the 3D and in the real world. Thus , we proposed to the *mocaptor* to move in a corridor along both sides of the stage, and two *manipulactors*, one for each *mocaptors* change the direction movement direction of the avatar to allow them to walk everywhere. Moreover, the *manipulactor*, placed in the

audience part has the mission of keeping the avatar address towards the actor, by rotating the avatar properly according to perspective constraints.

Figure 7: 2 avatars controlled by 2 mocaptors

We worked on combining several viewpoints on the 3D space:

- In his acting corridor, the *mocaptor* needs a permanent contact with the 3D rendering and the movement of his controlled avatar. He has an indirect contact with the actor and the audience.
- Among the audience, the *manipulactor* combines *mocaptor*, avatar and actor movements and is responsible for the continuity between physical and 3D stages.
- The actor follows the director indication with a distorted contact with the avatar partner due to perspective constraints.
- The stage director and the audience finally appreciate the interactions between physical and 3D realities. The stage-director addresses indication of the three above collaborators to build plausible theatrical stories.

Sharing scenic avatar actions between *mocaptor* and *manipulactor* is a complex issue that will need future interaction design developments.

CONCLUSIONS.

We have exposed in this article our new prototype for the theater: **Avatar Staging**, evolution of our previous work **OutilNum**. This tool allowed us to put the actor at the center of our problematic, and the dialogue between the director and the virtual universe, forcing us to imagine new digital control professions.

To do this, we developed a system of events management, the "*acting input*" which allows to associate a source that can be of hardware origin via a device, or software (algorithm, timer ...). In addition, we have developed a method of retargeting motion based on a two-stage operation: from the capture source to a neutral avatar, then from the neutral avatar to a complex avatar using a *puppeter*. This allows us to combine multiple source moves easier.

All this work allowed us to ask ourselves about a reorganization of the theater teams by integrating new professions and hybrid profiles halfway between the director, the actor, the operator and the puppeteer

Figure 8: avatar staging in action during a master class with the cnsad

REFERENCES.

[1] C.S.K.LesleyNortham, JoeIstead.Rtfx: On-setprevis with unrealengine3. In S. B. Heidelberg, editor, Entertainm

[2] Plessiet C., Chaabane S., and Khemiri G. 2015. Autonomous and interactive virtual actor, cooperative virtual environment for immersive Previsualisation tool oriented to movies. In Proceedings of the 2015 Virtual Reality International Conference (VRIC '15). ACM, New York, NY, USA, Article 5, 4 pages. DOI: https://doi.org/10.1145/2806173.2806177

[3] Gagneré, G., Plessiet, C., « Traversées Des Frontières », Frontières Numériques & Artéfacts; L'Harmattan),2015.

[4] Gagneré, G., Plessiet, C., « Perceptions (Théâtrales) de L'augmentation Numérique.», In Proceedings of the International Conference Frontières Numériques - Perceptions. Toulon (France), 2017.

[5] Gagneré G., Plessiet C., Sohier R. « Une recherche-création sur l'espace de jeu théâtral à l'ère du réseau » in Internet des Objets, Numéro 2, Volume 2, ISTE, 2017

[6] Katz D. S., Film Directing Shot by Shot: Visualizing from Concept to Screen, Focal Press, 1991.

[7] Cassandras, C.G. Lafortune, S., Introduction to Discrete Event Systems, Springer, 2008

[8] Patoli, M. Z., Gkion, M., Newbury, P., White, M. «Real time online motion capture for entertainment applications» Dans Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), Third IEEE International Conference on (pp. 139-145). IEEE. 2010

[9] Leroy, X. «The effectiveness of type-based unboxing.» Dans Workshop on Types in Compilation (Vol. 97), 1997.