Laboratório VISGRAF
Instituto de Matemática Pura e Aplicada

Motion Creation from Motion Capture Data

Louise Roy
Luiz Velho (orientador)

Technical Report TR-15-02 Relatório Técnico

January - 2015 - Janeiro

Instituto Nacional de
Matemática Pura e
Aplicada

Technical Report

Motion Creation from Motion Capture Data

Author
Louise Roy

Instructor
Luiz Velho

Rio de Janeiro, January 16, 2015

Contents

Introduction

« In looking for new movement, I would look for something I didn't know about rather than something I did know about.[1] »

Merce Cunningham was undoubtedly one of the first choreographers to explore the innumerous possibilities of the interaction between new technologies and arts, and his sentence sumarizes well his passion for innovation. In 1968, he already had imagined a computer technology that could display three-dimensional figures on a computer screen.[2] About twenty years later, Lifeforms was born : a pioneer software tool that enabled him to enlarge possibilities in movement creation in many ways. He said about it that one could easily choreograph with such a device.[3]

Since then, we understood that « computer technology can enrich the creative experience of choreographers by providing new methods to explore movement compositionally and recommends further directions for choreographic research.[4] »

The software tool proposed here is to fit into this framework : a real dancer generates, through his movements and in real time, the creation of a virtual dancer with which he can interact.

FIGURE 1 – Hand-drawn Spaces by Merce Cunningham, Paul Kaiser, and Shelley Eshkar, 1998[1]

---

[1] Hand-drawn Spaces is a virtual dance installation that represents a mental landscape in which motion-captured hand-drawn figures perform intricate choreography in 3D

1 Creating a Motion Database

Motion capture, or MoCap, is the process of recording an entity, whether it is an object or people. It has many applications, mainly in entertainment, validation of computer visions and robotics, but also in sports or in the medical field.

The choice of using Mocap data was quite obvious : indeed, a realistic animation of the human movement is a really challenging task for several reasons. First, human movements are very complex because they present a lot of joints and degrees of freedom, then we are surrounded by them in our everyday life and are therefore naturally skilled to notice their subtle details. For instance, we can recognize someone at a quite long distance just by the way he or she is walking, or it clearly appears to us what mood someone is in just by looking at his body language. That's why it is so obvious to us when movements are not a hundred percent accurate.

The synthesis of danced movements may even be harder : dancers spend indeed their whole life perfecting their movements and furthermore, others concepts more complex are to be taken into account such as energy or weight sensation. Using MoCap data therefore seems the most natural and effective way to create an improvisation tool for dance.

1.1 MoCap Technique

There are several types of MoCap systems. The one we used is an optical one : sensors are placed on the body of a dancer and the recognition sources are cameras. This is important because sensors won't constraint the movements of the dancers during their performance. The system uses the data acquired by the image of the sensors to triangulate the 3D position of a subject, using the fact that cameras have overlapping fields of view. According to the number of cameras, it is possible to record "flawless" data, or in other words that doesn't need to be edited after the recording.

There are three main steps. First, the sensors are to be identified on the 2D data recorded by the cameras : we try to keep track of them on each frame. The complexity of this step comes from the fact that sensors can overlap, change relative position one to another, be hidden by another body part or even change position on the body due to the movement being realized. The second step is the 3D reconstruction from the 2D data. We need therefore to calibrate the positions and orientations of the cameras in order to reconstruct the 3D data from at least two different cameras that overlap. Finally, the third step is to apply these data to a virtual skeleton.

This technique records human movements and can be used to draw a performer in real time as well as record a database for a later use.

1.2 Creating the database

The Motion Capture was performed at the Visgraf Laboratory using OptiTrack, the MoCap system of NaturalPoint. It is composed by 12 infrared cameras at 100 fps that keep track of the retroreflective sensors which are placed on the dancer's body.

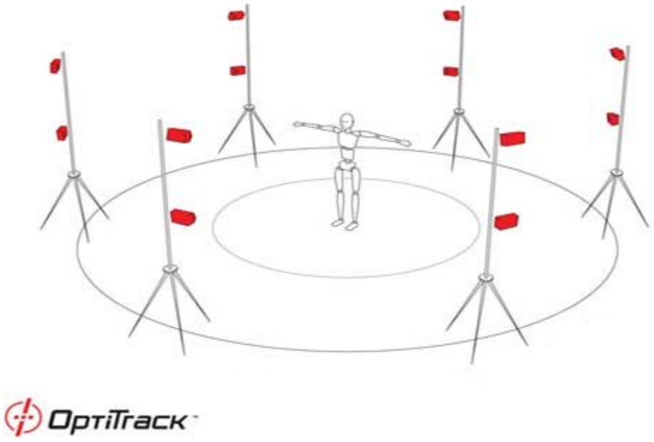

FIGURE 2 – 12 Camera setup for the Motion Capture

Firstly, we had to calibrate the cameras in order to fullfill the second step of the 3D reconstruction mentioned above. Then, it is necessary to check the eventual reflections in the recording area, and to remove them so that it won't interfere with the recognition of the actual sensors. Thanks to the software tool associated to this system, Motive, we can then place the sensors on performer's body, following the instructions of placement given by Motive according to the number of sensors chosen.

FIGURE 3 – Retroreflective sensors placed on a performer

It is now possible to begin the acquisition. In practice, we realized that it is necessary for our use - that is, to record dance movements - to let the performer beat his hands to the music so that we can synchronize it later.

2 Drawing a Streaming Dancer

2.1 NaturalPoint's NatNet SDK

The NatNet SDK allows us to send motion tracking data into our application.

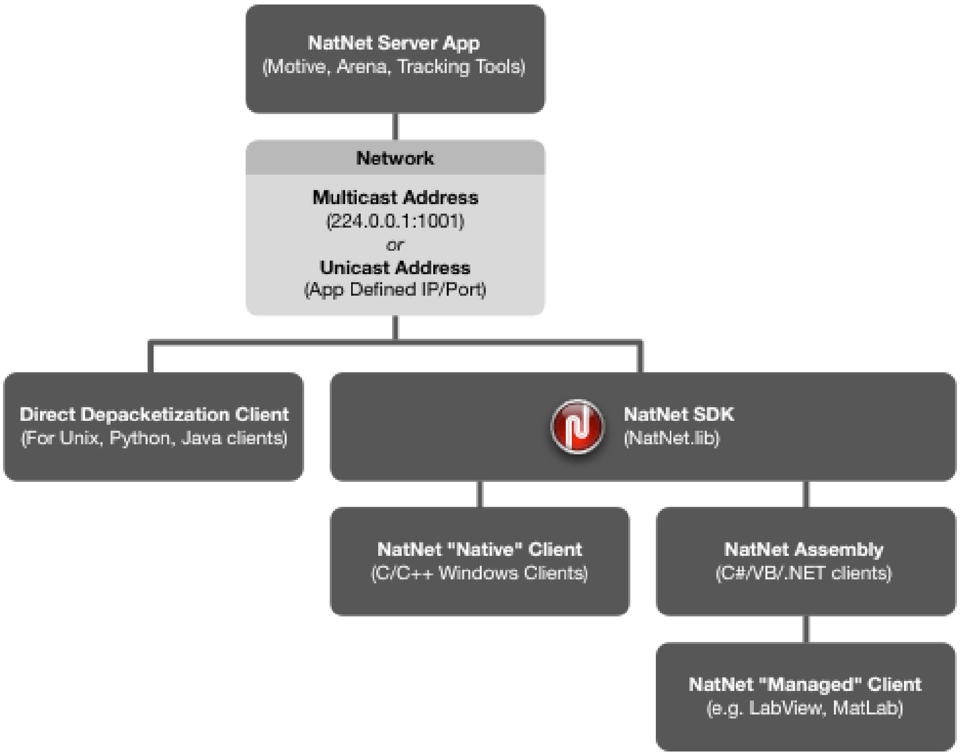

FIGURE 4 – Functioning of the NatNet SDK

A library, done by Djalma Lucio, was integrated into the project and enabled us to draw in real time the performance of a dancer. Data is sent by the MoCap system program Motive in packages and collected into our project. It is then read by our application similarly to a BVH file[1]. By applying the rotations to the joints, we can easily draw the dancer in real-time.

---

[1] A BVH file can be found in the appendix

3 Structure for Danced Improvisation

Motion Capture is a very powerfull technique, for it allows us to record and reproduce natural and realistic human movements. However, having a database isn't enough to combine them and improvize randomly a piece of dance. We now want to be able to reorganize and reuse pieces of movements, in order to create new and more complex ones. We chosed the Motion Graph Structure introduced by Lucas Kovar, Michael Gleicher and Frédéric Pighin.[5]

3.1 Motion Graph Construction

3.1.1 Basic Construction

While creating a Motion Graph Structure, we want to achieve a certain connectivity within a single file or among a database in order to be able to improvize dance movements. We interprete each movement as a clip of frames, each frame being represented as a sampling of the subject's parameters such as the position and rotation of each joint.

In such a graph, the edges - which means the graph is oriented - represent sequences of frames, and the nodes are certain strategic frames that can be used to connect two clips between them. We can construct a trivial graph in placing all of the movements as the initial edges, all disconnected between one another.

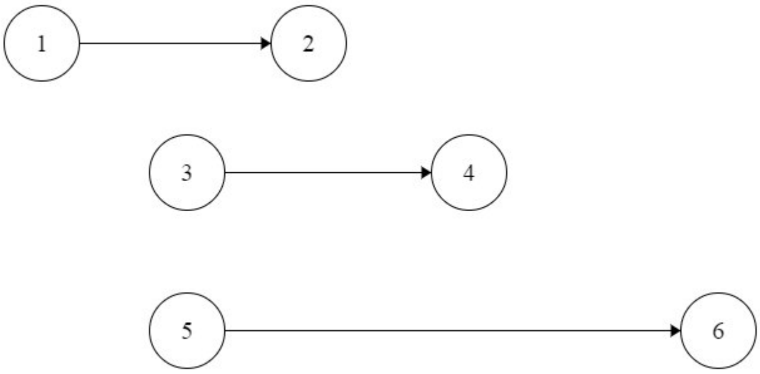

FIGURE 5 – Trivial Motion Graph for three motion clips

3.1.2 Increase of connectivity

For the moment, we can only play a single motion. We now need to increase the connectivity of the graph.

We have to compute the distance between two different frames i and j to know if they are good candidates to add a transition. For this purpose, we consider the point clouds formed at each frame by the body pose of the skeleton. We can indeed notice that the pose stays unchanged whether we deplace it along the floor or rotate it along its vertical axis, so we only need to consider the point clouds formed by the pose and driven by the skeleton's center of gravity - we usually choose the hips.

We group them in k point clouds, corresponding to a k-length motion clip, k being chosen by the user to be the length of the computed transitions. To connect i and j, we have to compare the k-length window of frames beginning at i, and the k-length window of frames ending at j. Given the transformation that minimizes the distance between the two body poses, the distance between the two frames is then the usual distance calculation. That is, we want to find the following sum, θ, x_0 and z_0 being the parameters of the aforesaid transformation, and p_i and $p'i$ the two point clouds :

$$\min_{\theta,x_0,z_0} \sum_i w_i \|\mathbf{p_i} - \mathbf{T}_{\theta,\mathbf{x_0},\mathbf{z0}}\mathbf{p'i}\|^2$$

The w_i are the weights affected to each value that compose the sampling of the subject, for it is obvious that the rotation of a hand shouldn't influence the distance the same way the position of a foot is. This optimization has a closed-form solution that was first found by Lucas Kovar, Michael Gleicher and Frédéric Pighin.[6] We can then decide whether to keep the transition or not, according to a limit value.

Once the transition is kept, we have to interpolate the two windows of frames using quaternions in order to blend the two movements and create a new one, as seen on the folowing figure.

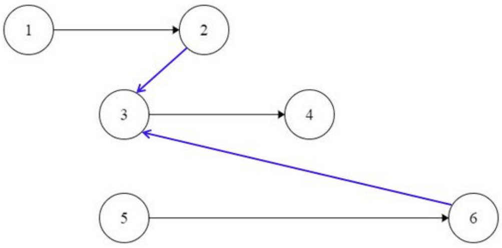

FIGURE 6 – Adding transitions to the basic graph

3.2 Adaptation for Dance Improvisation

3.2.1 Measure synchronous

This project is no study on movement itself and as such, but tries to explore the creativity of a real live dancer reacting to a virtual one, and the interactions between the two. It appears then interesting to keep a certain rythm so that the two dancer can have a creative dialogue, and the most adaptated is of course music.

Musics are divided into measures, which organize them according to the melody. That is why we decide to preserve the size of the melodic sentences during the composition, and we try to build a measure-synchronous graph as it was already done for a previous project in the laboratory.[7]

This means each motion clip is cut into n-length smaller motion clips, n being the number of frames contained in a music measure. This allows greater connectivity while preserving the rythm constraint, so that the real dancer can still interact with the virtual one on music.

3.2.2 Helping real-time choices

Since we are going to interact in real-time with a performer, we decide to store additionnal information on the graph that will help the virtual dancer to choose quickly his next move.

Among these informations, we hold the displacement of each move - for that will help us keep a safe distance between the two dancers -, but also other kinds of informations that would enable us to make the virtual dancer respond in a coherent manner to the movements of the performer, such as speed, acceleration, head orientation...

4 Resulting Program

Once this adaptated Motion Graph is created, improvizing dance is just walking randomly on the graph. We thought however, that we could create different kinds of responses from the virtual dancer according to the situation.

For instance, we thought interesting to let possible an interaction between the program and a choreographer. In this case, the virtual dancer would try to obey the choreographer and not only walk randomly on the graph.

4.1 Functionning

The virtual dancer created by the program has three main influences to improvize movement : the database he can chose his moves from, the real performer's movements and the choreographer's suggestions.

4.1.1 Database's influence

The first influence is the database's one. The Motion Graph is created from a database of movements, and therefore the possibilities are limited by the number and style of movements. The graph's connectivity also has a great influence on the possibilities at each step, and is conditioned by the clips in the database : if they are well planed - for instance to finish and end always the same way - the graph won't have any dead end and will be able to keep running endlessly.

4.1.2 Performer's influence

Secondly, the virtual dancer responds to the real dancer. For the moment, its only interaction is to avoid collision with the real dancer. This is, of course, the first thing to treat. At each new walk on the graph, the virtual dancer computes the future position he will have for each move, and checks if he'll be too close to the real dancer. If all of the moves lead him too close to the performer, he will try to choose the one that will keep him the further away from him.

4.1.3 Choreographer's influence

And finally, the virtual dancer can be influenced by the choreographer. The user interface enables him to choose a point in space that the virtual dancer will try to reach.

The figure below show the interactions between the different entities :

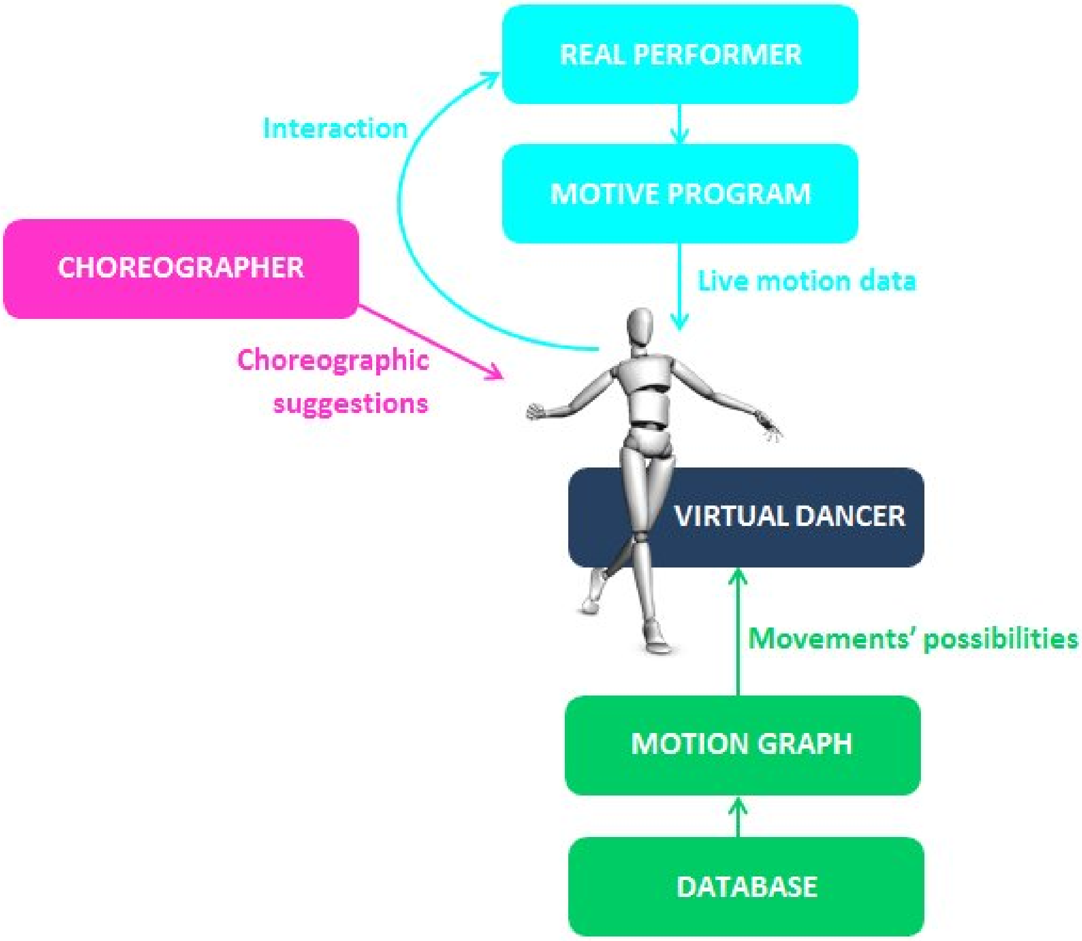

FIGURE 7 – Influences of the Virtual Dancer and Interactions

Discussion and Future Work

We have endeavoured to provide a new and interactive tool for Dance Improvisation based on Motion Capture. After recording dance movements, we created a graph structure and increase its connectivity to enable improvisation. We started to study the interaction between the virtual dancer and its environment, composed by the performance of the real dancer but also a possible choreographer.

This line is still to be explored. We created for the moment only two modes of Improvisation : the first one is a random walk on the Motion Graph, and the second one, triggered by the choreographer through the user interface, is a point in Space to be reached.

We could continue down this idea and invent new modes, which could be turned on and off by the real dancer and / or the choreographer, for example with the user interface or even with predetermined key movements that the real dancer could perform to change modes.

Theses modes could take into account other informations, such as the expressivity of the real dancer movements. We can indeed notice that sadness would make someone do movements that bring his body parts close to his center, - bending the head or bringing the arms close to the stomach - where as anger would make him do the opposite in energetic movements. Another kind of mode could take into account the music's modulations.

We could also record databases of different dance styles and make the virtual dancer react to the dance style the real dancer is performing.

All of theses ideas would enrich the program and, we hope, help to explore further human creativity and reactivity in the dance field.

Références

[1] Merce Cunningham, *The Dancer and the Dance/conversations with Jacqueline Lesschaeve*, New York and London, Marion Boyars, 1985.

[2] Merce Cunningham, *Changes : Notes on Choreography, edited by Frances Starr*, New York, Something Else Press, 1968.

[3] *Ibid*

[4] Thecla Schiphorst, *A Case Study of Merce Cunningham's use of the LifeForm Computer Choreographic System in the Making of Trackers*, B.G.S Simon Fraser University, 1986.

[5] Lucas Kovar, Michael Gleicher and Frédéric Pighin, *Motion Graphs*, In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2006.

[6] *Ibid*

[7] Adriana Schulz, Marcelo Cicconet, Bruno Madeira, Aldo Zang and Luiz Velho, *Techniques for CG Music Video Production : the making of Dance to the Music / Play to the Motion*, Technical Report TR-2010-04, Laboratorio VISGRAF - IMPA, March 2010

Appendix

A BVH file containing a single frame

```
HIERARCHY
ROOT Hips
{
    OFFSET 0.000000 0.000000 0.000000
    CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
        JOINT Spine1
        {
            OFFSET -0.000000 26.239704 0.000000
            CHANNELS 3 Zrotation Xrotation Yrotation
            JOINT Neck
            {
                OFFSET -0.000000 19.126398 1.738763
                CHANNELS 3 Zrotation Xrotation Yrotation
                JOINT Head
                {
                    OFFSET -0.000000 12.010013 -1.715716
                    CHANNELS 3 Zrotation Xrotation Yrotation
                    End Site
                    {
                        OFFSET -0.000000 21.446451 0.000000
                    }
                }
            }
            JOINT LeftShoulder
            {
                OFFSET 3.450920 17.462563 -0.165682
                CHANNELS 3 Zrotation Xrotation Yrotation
                JOINT LeftArm
                {
                    OFFSET 9.874601 0.000000 0.000000
                    CHANNELS 3 Zrotation Xrotation Yrotation
                    JOINT LeftForeArm
                    {
                        OFFSET 28.338211 0.000000 0.000000
                        CHANNELS 3 Zrotation Xrotation Yrotation
                        JOINT LeftHand
                        {
                            OFFSET 21.230946 0.000000 0.000000
                            CHANNELS 3 Zrotation Xrotation Yrotation
```

```
                        End Site
                        {
                            OFFSET 12.867871 0.000000 0.000000
                        }
                    }
                }
            }
        }
        JOINT RightShoulder
        {
            OFFSET -3.411501 17.462563 -0.165682
            CHANNELS 3 Zrotation Xrotation Yrotation
            JOINT RightArm
            {
                OFFSET -9.874601 0.000000 0.000000
                CHANNELS 3 Zrotation Xrotation Yrotation
                JOINT RightForeArm
                {
                    OFFSET -28.338211 0.000000 0.000000
                    CHANNELS 3 Zrotation Xrotation Yrotation
                    JOINT RightHand
                    {
                        OFFSET -21.230946 0.000000 0.000000
                        CHANNELS 3 Zrotation Xrotation Yrotation
                        End Site
                        {
                            OFFSET -12.867871 0.000000 0.000000
                        }
                    }
                }
            }
        }
    }
    JOINT LeftUpLeg
{
    OFFSET 8.578581 0.000000 0.000000
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftLeg
    {
        OFFSET -0.000000 -34.842335 0.000000
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT LeftFoot
        {
            OFFSET -0.000000 -40.778851 0.000000
```

```
                CHANNELS 3 Zrotation Xrotation Yrotation
                JOINT LeftToeBase
                {
                    OFFSET -0.000000 -5.576077 12.867871
                    CHANNELS 3 Zrotation Xrotation Yrotation
                    End Site
                    {
                        OFFSET -0.000000 0.000000 3.431432
                    }
                }
            }
        }
    }
    JOINT RightUpLeg
    {
        OFFSET -8.578581 0.000000 0.000000
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT RightLeg
        {
            OFFSET -0.000000 -34.842335 0.000000
            CHANNELS 3 Zrotation Xrotation Yrotation
            JOINT RightFoot
            {
                OFFSET -0.000000 -40.778851 0.000000
                CHANNELS 3 Zrotation Xrotation Yrotation
                JOINT RightToeBase
                {
                    OFFSET -0.000000 -5.576077 12.867871
                    CHANNELS 3 Zrotation Xrotation Yrotation
                    End Site
                    {
                        OFFSET -0.000000 0.000000 3.431432
                    }
                }
            }
        }
    }
}
MOTION
Frames:    3254
Frame Time:    0.010000
97.359428    77.723724    4.324814    -15.715580    14.386765    2.159513
-11.864145    -0.211646    -3.105822    1.112894    10.428071    -10.630816
20.588499    -12.817129    -5.943286    11.603930    -3.007114    -9.282147
```

```
-18.308559    8.539310    -5.637339    8.009450    -14.678966    -10.840744
58.208191    -12.084389    3.422480    -8.048749    1.724394    -0.939846
49.666428    -1.559678    -11.056878    -6.490510    -5.483145    5.510576
-2.454423    -5.120018    -2.426296    8.858532    -11.708108    0.192597
5.979720    -1.083734    15.335091    -1.970501    2.729267    10.856393
0.000000    0.000000    0.000000    -6.039531    -15.505136    -5.982050
3.756803    5.617821    -19.850521    10.313384    38.026775    -9.608858
0.000000    0.000000    0.000000
```