Real time whole body motion mapping for avatars and robots

Conference Paper · October 2013

DOI: 10.1145/2503713.2503747

Real Time Whole Body Motion Mapping for Avatars and Robots

Bernhard Spanlang[1] bspanlang@ub.edu
Xavi Navarro[1] xavi.navarro@ub.edu
Jean-Marie Normand[1,2] jean-marie.normand@ec-nantes.fr
Sameer Kishore[1] s.kishore@ub.edu
Rodrigo Pizarro[1] rodrigo.pizarro@ub.edu
Mel Slater[1,3] melslater@ub.edu

[1]EVENT Lab, Facultat de Psicologia, Universitat de Barcelona, Spain
[2] CERMA, UMR 1563, École Centrale de Nantes, France
[3]Institució Catalana de Recerca i Estudis Avançats, Spain

Abstract

We describe a system that allows for controlling different robots and avatars from a real time motion stream. The underlying problem is that motion data from tracking systems is usually represented differently to the motion data required to drive an avatar or a robot: there may be different joints, motion may be represented by absolute joint positions and rotations or by a root position, bone lengths and relative rotations in the skeletal hierarchy. Our system resolves these issues by remapping in real time the tracked motion so that the avatar or robot performs motions that are visually close to those of the tracked person. The mapping can also be reconfigured interactively at run-time. We demonstrate the effectiveness of our system by case studies in which a tracked person is embodied as an avatar in immersive virtual reality or as a robot in a remote location. We show this with a variety of tracking systems, humanoid avatars and robots.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality;

Keywords: Avatars, Robots, Motion Capture, Virtual Reality

1 Introduction

Until recently, systems for tracking whole body motions of a person used to be accessible to high end movie studios and wealthy research institutions only. With the decreasing cost of sensors more tracking systems are becoming available at a fraction of their previous cost. A number of different technologies are employed ranging from optical systems with retro-reflective markers to inertia based sensors there is a large variety of systems available now that can provide different quality of motion data from high end for the movie industry to low end game controllers. Tracking systems also vary in the effort that is required for the user to be tracked, from fairly restrictive marker or sensor based suits, to completely marker-less and sensor-less tracking based computer vision and range imaging.

Thus, a reliable mapping of a person's movements to an artificial counterpart of the physical body is an extremely useful tool to study the mechanisms of self and body perception in virtual reality. The purpose of this paper is to describe a generalized tool that can map motion capture data to a variety of representations, from avatars to humanoid robots, but where only a single software interface is suitable for any type of representation. We call this system HALCA Unity3D Mapping Avatars through Network (HUMAN).

VRST'13, October 06 - 09 2013, Singapore, Singapore

http://dx.doi.org/10.1145/2503713.2503747

2 Related Work

There are three strands of work that relate to the described system, software for merging and transforming tracking data, motion retargeting software and systems to control robots from motion capture data.

A popular library loosely related to the one described here is the Virtual Reality Peripheral Network (VRPN) [Taylor II et al. 2001] system. VRPN abstracts data from various peripheral devices to be accessible in Virtual Reality applications transparently over the network. Devices range from simple button inputs, over various tracking devices to eye trackers, etc.

Similarly OpenTracker [Reitmayr and Schmalstieg 2001] is a software library that aims at fusing information from different tracking sensors, with the possibility of easy reconfiguration and simple integration with existing VR systems. A further development of OpenTracker is the Ubiquitous Tracking (Ubitrack) [Newman et al. 2004] system which also aims at merging tracking data from multiple tracking sources, for example, to provide tracking data in a large factory environment.

Potentially our system could have been implemented on top of one of the above. For simplicity and since none of them provide specialized functionality to deal with skeletal hierarchies and to transform motion data in a structured way we decided to implement our system from scratch.

There are several possibilities on how to retarget motion capture data to creatures of different skeletal morphology. For example in [Hecker et al. 2008] a system for animating creatures created in the game Spore is presented. Animations for these creatures are designed in the game and created at run-time by a real time inverse kinematics (IK) solver. The system is not designed to work with real time motion capture input though, nor is it used to control robots.

Kulpa et al. [Kulpa and Multon 2005] describe a system that enables them to control up to 10 virtual characters in real time with motion capture data. They convert the original rotational representation to partially Cartesian coordinates (using for example a vector between hands and shoulders to describe the elbow angle) and combine this with an adapted cyclic coordinate descent (CCD) method.

In this paper we describe a system that can transfer the motion stream of various motion capture systems onto avatars with various skeletal representations or various robots. The system is modular which enables us to add new tracking devices, avatars and robots with little (if any) change in the code.

3 The System

In this section we first define the problem, then we enumerate and describe the system components and finally discuss our implementation of the system.

3.1 Problem Definition

The term motion in Virtual Reality systems refers to a sequence of states of an entity expressed in positions and rotations. We can express this motion using relative data or absolute data. Relative data refers to rotations or positions expressed in the parent's coordinate system in a skeletal hierarchy whereas absolute data refer to rotations or positions expressed in the global coordinate system. HUMAN deals with three types of skeletons: Motion Capture skeletons, Avatar skeletons and Robot skeletons. HUMAN converts the data from the Motion Capture to relative and absolute data in order to apply it to avatars, which depending on the avatar, will be either in relative or in absolute coordinates. Because of the way actuators are built in robots, they are usually driven by Euler angles for each joint instead[1].

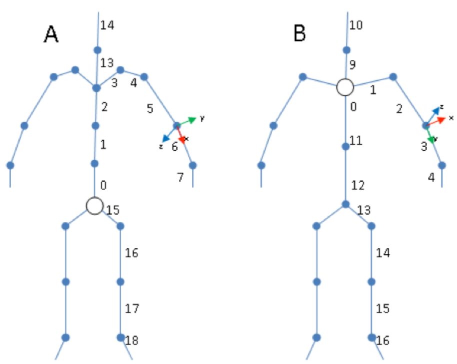

Figure 1: *Skeletal representations can differ in the number of skeletal segments and in which joint describes the root joint for skeletons with a relative rotation hierarchy. A: A skeleton with higher definition and the hip as root joint. B: A skeleton with less joints and the shoulder centre as the root joint.*

Figure 1 shows a skeletal representation of an avatar. For each joint of the hierarchy, we create a representation in relative and absolute coordinates. Computations are carried out depending on the input Motion Capture data being relative or absolute. In case of relative Motion Capture data, we need to be able to compute absolute quaternions for each joint of the skeleton. For joint j of the skeleton this is achieved, by traversing the bones hierarchy from the root bone until j and multiplying the relative quaternions along the kinematic chain ($\mathbf{K}_j$) to the target bone j. This allows us to compute absolute quaternions for each of the skeleton's joints.

For example to compute the absolute shoulder (joint 5) rotation in skeleton A of Figure 1 the following multiplications have to be performed:

$$\mathbf{q}_5^{abs} = \prod i \in \mathbf{K}_5 \mathbf{q}_i^{rel} \tag{1}$$

where $\mathbf{K}_5 = (0, 1, 2, 3, 4, 5)$.

[1]Euler Angles are required for the robots that we have tested with the HUMAN system but of course there may be other robot systems that require other representations for joint rotation.

Similarly absolute joint positions are computed by summing the rotated skeletal segment vectors.

In the case that the motion capture system sends absolute data, we compute relative quaternions for each joint in the skeleton. This is achieved by multiplying the absolute rotation for joint j by the inverse rotation (the quaternion conjugate) of its preceding joint in the hierarchy.

In addition, owing to the way artists design virtual characters and motion capture system designers decide about their skeletal coordinate systems it can be frequently observed that there is a mismatch between the axes representations in the different systems. By ordering and redirecting the axes the correct mapping from one system to the other is established.

We identified four types of skeletal mismatches:

- The tracked skeleton has less joints than the target skeleton: This issue can be resolved by either mapping only the relevant joints and keeping joints in between mapped joints rigid, or by dividing the rotation for the intermediate joints.
- The tracked skeleton has more joints than the target skeleton: In this case the system multiplies the rotations in between the selected joints that are being tracked.
- The tracked skeleton has a different root to the target skeleton: In this case the tracked data is first transformed to absolute rotations and relative rotations are computed for the new root.
- The initial tracked pose (usually a T-Pose) is different to the initial pose of the avatar (for example quite frequently avatars are modelled with their arms down 45 degrees): This can be resolved by adding a rotational offset which can be described in our XOF file (more details below).

For the robot, we compute absolute positions of all joints and compute the rotations as described in Section 3.2.4.

3.2 System Components

The system relies on three main components: motion capture systems, humanoid entities and HUMAN, the middleware library that connects the two, as illustrated in Figure 2. HUMAN is responsible for remapping the hierarchical transformation information of one skeletal system to another.

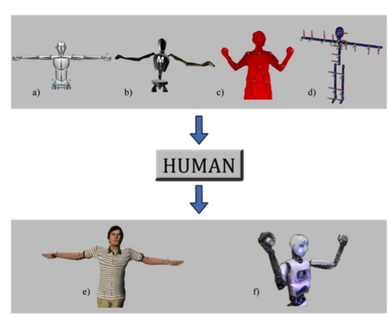

Figure 2: *The entire workflow described. Real-time tracking data from various motion capture devices such as a) Optitrack, b) XSens c) Kinect, d) Organic motion is obtained by HUMAN, which converts this data, taking into account the different configurations, into meaningful values that can be mapped to e) an avatar and/or f) a humanoid robot.*

3.2.1 Motion Capture Systems

A Motion Capture system aims at capturing the movements of the whole body of usually human participants. There are various technologies for capturing motion. Some require the user to wear a special suit with markers, for example NaturalPoint's Optitrack [2], Vicon's systems [3] and PhaseSpace's [4] multi camera based systems or inertia based sensors as for example in the MVN suit by XSens [5] or Animazoo [6]. A few systems can track the person in casual, ideally tight fitting clothing. For example the multi camera system from Organic Motion [7] or the structured light sensor in the Microsoft Kinect [8]. Such systems are called vision based markerless systems. Multi Camera systems require a calibration step in order to compute the relative camera positions and orientations (these are relative to a frame of reference). For the user this usually involves waving a marker configuration in the tracking area for a few minutes and defining a frame of reference by placing a triangle of markers.

All systems require another calibration step for finding the initial rotations and joint lengths of the tracked skeleton. All systems also have in common that they deliver the motion data as a stream of joint positions and/or orientations. Typically motion is described in skeletal hierarchies as shown in Figure 1. Most systems deliver their data in a simple UDP based network protocol that can be accessed either locally or remotely or by providing Software Development Kits (SDKs) that give access to the motion data stream.

3.2.2 HUMAN Motion Mapping Library

HUMAN is a thread based dynamic library that retrieves the information from the motion capture systems transforms this and passes it on to avatars or robots in real time. It uses an XML based Orientation File (XOF) and an avatar skeleton file (XSF) to map positions and orientations of the corresponding bones and joints.

XOF describes the coordinate system transformation from the axes of each joint of the system to the avatar's coordinate system for the different motion capture systems and it relates the joint numbers of the tracking system with the joint numbers (or joint names) of the avatar or robot. During run-time, the library parses the file to perform the required computations for a specific system configuration. This file can be edited at run time to see how changes affect the result of the motion on the avatars or robots.

HUMAN uses the Cal3D [Heidelberger 2006] XSF format to describe the skeletal hierarchy along with the initial joint rotations and joint lengths. The library reads the XSF file to create an internal structure of the bones of the avatar, which is going to be used in order to store, refresh and compute all the transformations from the original data received from the motion capture system.

3.2.3 Character Animation Libraries

In virtual reality, avatars are usually represented by textured meshes with vertex attributes that define the weight of how much a vertex is influenced by the movements of a skeletal segment. For efficiency reasons the most commonly used mesh deformations are linear blend skinning or dual quaternion skinning [Kavan et al. 2007]. We use a hardware accelerated library for character animation (HALCA) [Gillies and Spanlang 2010] for avatar animation and visualisation in XVR [Tecchia et al. 2010]. HALCA uses the Cal3D [Heidelberger 2006] XML file format to describe skeleton weighted meshes, animations, and materials. Exporters for this format are available for 3D Studio Max, Maya and Blender. HALCA was built to exploit programmable graphics hardware by allowing hosting applications to dynamically change GLSL shader programs [22]. It supports the delivery of skeletal transformation information to the GPU in matrix or dual quaternion form and the skinning shader can be modified at run time.

Alternatively HUMAN can also use the Unity3D[9] games engine which provides avatar rendering through its SkinnedMeshRenderer component. This component renders skeleton-rigged meshes based on the relative transformation of their bones. Importers for 3D Studio Max, Maya and Blender are built in.

3.2.4 Robots

HUMAN can also map a user's movements on to humanoid robots. The field of robotics has come a long way in the last few years in terms of building highly advanced, autonomous humanoid robots. Many humanoid robots now come equipped with several degrees of freedom for their limbs and are also able to walk. Being able to control a robot remotely has many applications, particularly, in teleoperation.

However, mapping real-time tracking data to a robot's limbs is very different from doing the same task with an avatar. Various robots are built differently, with various kinds of motors and actuators controlling their limbs, which in turn define the constraints of the system. The degrees of freedom associated with a robot's limb will dictate the degree of accuracy and control the user will have while controlling it.

The algorithm that has been developed for real-time mapping, modifies data obtained from the tracking system into 'robot-friendly' angles. The whole process was developed initially for the Nao robot, manufactured by Aldebaran Robotics[10], but has also been ported for the Robothespian[11] developed by Engineered Arts.

Developing a solution for implementing real-time walking depends heavily on the configuration and physical ability of the robot. Usually, since robots have a low speed, and do not have the same dimensions as a human (usually smaller), directly mapping the movement of the legs is not a feasible solution. It also may result in loss of balance.

The algorithm we have developed for the walking, tracks the torso position of the user instead of individually tracking the legs. Initially, the user's torso position (2D coordinates on the X-Z plane) is obtained, and at every frame, it is compared with the updated position of the torso. Whenever a robot-based threshold is crossed, the walking mechanism is triggered. The advantage of this algorithm is that it is robust and can be used with any robot that has the ability to move, and does not depend on how the robot moves. It could have any mechanism, such as legs, wheels, a movable platform, and the algorithm would still work.

4 Results

In this section we show results of our system with various combinations of tracking systems, and rendering engines and robots.

[2] http://www.naturalpoint.com/optitrack/
[3] http://www.vicon.com/
[4] http://www.phasespace.com/
[5] http://www.xsens.com/en/general/mvn
[6] http://www.animazoo.com/
[7] http://www.organicmotion.com/
[8] http://www.microsoft.com/en-us/kinectforwindows/
[9] http://unity3d.com/
[10] http://www.aldebaran-robotics.com/
[11] http://www.robothespian.co.uk

Finally we enumerate several research studies in which the system was used to evaluate the consequences of virtual embodiment and telepresence.

4.1 Current configurations

For various research demos the system was tested with a combination of motion capture devices, robots and avatars from different companies in Unity3D and HALCA. In the accompanying video we demonstrate Kinect control of a Rocketbox avatar in Unity3D (also shown in Figure 3), Optitrack control of a Rocketbox avatar in HALCA simultaneously with Nao and Robothespian robots and XSens control of Rocketbox avatar in HALCA simultaneously with Nao and Robothespian robots (Figure 3).

Figure 3: *Top: Rocketbox Avatar in Unity3D controlled by motions from Kinect; Left: RocketBox avatar in HALCA, Nao and Robothespian controlled by XSens motion data; Rigth: XSens controls Robothespian while Optitrack controls Nao robot.*

4.2 Studies that used the system

Kilteni et al. [Kilteni et al. 2013] used HUMAN to show that people virtually embodied in a Casual Dark-Skinned avatar performed drumming very differently to people embodied in a formal suited light-skinned avatar. They performed Full Body tracking with the Optitrack system and they used avatars from Rocketbox visualised with HALCA in XVR.

Another group of researchers used HUMAN in a demo for the BBC[12] to control a Robothespian robot in London by a real time motion capture stream from the XSens suit worn by someone in Barcelona. In this demo in addition to the motion stream the images from the robot's eyes (cameras built in the head of it) were transferred to a head mounted display (HMD). With this setup the person in Barcelona could physically interact with the BBC journalist in London.

5 Conclusions and Future Work

Since the system has been developed in a modular, portable and reusable way, we can add easily more tracking systems, graphics engines and robots. To add a new family of avatars (avatars with different skeletal hierarchies), we only need to adapt the XOF file corresponding to the new skeletal structure. To add a new tracking system, we have to adapt the XOF files of all the avatar families and create the module in the core of the system in order to add the required network protocol or SDK to access the motion data. If we need to add a new robot, we have to create the module in the core of the system to add the network protocol or SDK for the new robot.

[12]http://www.bbc.co.uk/news/technology-18017745

Acknowledgements

This work was funded under MS's Advanced European Research Council grant TRAVERSE Grant Number 227985. SK was supported by the FI-DGR grant from the Catalan Government (Gencat) co-funded by the European Social Fund (EC-ESF).

References

GILLIES, M., AND SPANLANG, B. 2010. Comparing and evaluating real time character engines for virtual environments. *Presence: Teleoper. Virtual Environ. 19*, 2, 95–117.

HECKER, C., RAABE, B., ENSLOW, R. W., DEWEESE, J., MAYNARD, J., AND VAN PROOIJEN, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 27.

HEIDELBERGER, B., 2006. Character Animation Library Cal3D.

KAVAN, L., COLLINS, S., ZARA, J., AND O'SULLIVAN, C. 2007. Skinning with Dual Quaternions. In *2007 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM Press, 39–46.

KILTENI, K., BERGSTROM, I., AND SLATER, M. 2013. Drumming in immersive virtual reality: the body shapes the way we play. *IEEE Trans Vis Comput Graph 19*, 4, 597–605.

KULPA, R., AND MULTON, F. 2005. Fast Inverse Kinematics and Kinetics Solver for Human-Like Figures. In *IEEE Humanoids*, 38–43.

NEWMAN, J., WAGNER, M., BAUER, M., MACWILLIAMS, A., PINTARIC, T., BEYER, D., PUSTKA, D., STRASSER, F., SCHMALSTIEG, D., AND KLINKER, G. 2004. Ubiquitous Tracking for Augmented Reality. In *In Proc. of International Symposium on Mixed and Augmented Reality (ISMAR'04*, 192–201.

REITMAYR, G., AND SCHMALSTIEG, D. 2001. OpenTracker - An Open Software Architecture for Reconfigurable Tracking based on XML. In *Proceedings of the Virtual Reality 2001 Conference (VR'01)*, IEEE Computer Society, Washington, DC, USA, VR '01, 285—-.

TAYLOR II, R. M., HUDSON, T. C., SEEGER, A., WEBER, H., JULIANO, J., AND HELSER, A. T. 2001. VRPN: a device-independent, network-transparent VR peripheral system. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, New York, NY, USA, 55–61.

TECCHIA, F., CARROZZINO, M., BACINELLI, S., ROSSI, F., VERCELLI, D., MARINO, G., GASPARELLO, P., AND BERGAMASCO, M. 2010. A Flexible Framework for Wide-Spectrum VR Development. *Presence: Teleoperators and Virtual Environments 19*, 4, 302–312.