Scientific Research Publishing

Journal of Transportation Technologies, 2022, 12, 296-309
https://www.scirp.org/journal/jtts
ISSN Online: 2160-0481
ISSN Print: 2160-0473

Methodical Approach to Integrate Human Movement Diversity in Real-Time into a Virtual Test Field for Highly Automated Vehicle Systems

René Degen[1,2*], Alexander Tauber[1], Alexander Nüßgen[1,2], Marcus Irmer[1,2], Florian Klein[3], Christian Schyr[4], Mats Leijon[2], Margot Ruschitzka[1]

[1]CAD CAM Center Cologne, Institute of Automotive Engineering Cologne (IFK), Faculty of Automotive Systems and Production, Cologne University of Applied Sciences, Cologne, Germany
[2]Division of Electricity, Department of Electrical Engineering, Uppsala University, Uppsala, Sweden
[3]HHVISION GbR, Cologne, Germany
[4]Advanced Solution Lab, AVL Deutschland GmbH, Karlsruhe, Germany
Email: *rene.degen@th.koeln.de

How to cite this paper: Degen, R., Tauber, A., Nüßgen, A., Irmer, M., Klein, F., Schyr, C., Leijon, M. and Ruschitzka, M. (2022) Methodical Approach to Integrate Human Movement Diversity in Real-Time into a Virtual Test Field for Highly Automated Vehicle Systems. *Journal of Transportation Technologies*, **12**, 296-309.
https://doi.org/10.4236/jtts.2022.123018

Received: April 1, 2022
Accepted: June 5, 2022
Published: June 8, 2022

Abstract

Recently, virtual realities and simulations play important roles in the development of automated driving functionalities. By an appropriate abstraction, they help to design, investigate and communicate real traffic scenario complexity. Especially, for edge cases investigations of interactions between vulnerable road users (VRU) and highly automated driving functions, valid virtual models are essential for the quality of results. The aim of this study is to measure, process and integrate real human movement behaviour into a virtual test environment for highly automated vehicle functionalities. The overall system consists of a georeferenced virtual city model and a vehicle dynamics model, including probabilistic sensor descriptions. By motion capture hardware, real humanoid behaviour is applied to a virtual human avatar in the test environment. Through retargeting methods, which enable the independency of avatar and person under test (PuT) dimensions, the virtual avatar diversity is increased. To verify the biomechanical behaviour of the virtual avatars, a qualitative study is performed, which funds on a representative movement sequence. The results confirm the functionality of the used methodology and enable PuT independence control of the virtual avatars in real-time.

Keywords

Advanced Driver Assistance Systems/Automated Driving (ADAS/AD),

Autonomous Mobility, Virtual Testing, Motion Capture

1. Introduction

The successful development of new driving systems and highly automated vehicles requires testing and validation of automated driving functions. These systems provide resource conserving driving, noise and traffic reduction, but also the intelligent linking of all mobility participants for more safety. In 2019 48% of all new cars sold in Germany were equipped with a lane keeping assistant, 38% were delivered with an adaptive cruise control and 39% have an autonomous emergency brake [1]. In 2020, a study found that 90% of respondents felt that ADAS increased vehicle safety and 89% found that assistance systems made driving more comfortable [2]. Although the data refer to the German market, a similar result can be expected internationally. This leads to the expectation that the market for ADAS will continue to grow in the future. Besides the opportunities ADAS offer to vehicle safety, they also increase the vehicles complexity and the testing effort. [3] gives a theoretical approach for the approval of ADAS with regard to an autonomous highway pilot. On German motorways a deadly accident happens every 662 million kilometers. The authors say this situation needs to be reconstructed ten times to approve an ADAS system. This leads to a theoretical demand of 6.62 Billion test kilometers. This need cannot be met by test drives alone. One fundamental verification approach funds at hardware-in-the-loop (HiL) and software-in-the-loop (SiL) methods. Here, real hardware or software developments are connected to simulation environments, to verify specific functions. Approaches, where complete vehicles are in the loop (ViL) and their sensors are stimulated virtually become popular in shortly. Brissard *et al.* firstly present a new approach, which combines a highly-dynamic test bed for the powertrain, steering and brake system integrated in the complete vehicle with a detailed vehicle model operating in a virtual environment [4]. Gadringer *et al.* published a similar approach in 2018, where a real vehicle at the vehicle test bench is combined with an environmental simulation in virtual reality (VR) [5]. In both approaches real vehicle behavior is captured by vehicle test bench technologies and the simulation of the environment is done in virtual. Thus, it is possible to integrate the real vehicle into the virtual environment. For this, interfaces of the surrounding sensors are needed. In the given example by Gardinger *et al.*, target simulators are used to stimulate the real radar sensor. Further, in 2018 Al-Sadi *et al.* [6] presented a similar approach, where the real sensors were stimulated by specific actuators, which transmit the virtual detected information to the real sensor. Here, the sensor stimulation could completely be done virtually, by importing the sensor‘s information directly to the vehicle’s bus system. HiL approaches, where real sensors are stimulated by virtual detected information like [7] are already known before.

In the field of highly automated and autonomous driving, the focus is now increasing on digital environments for testing ADAS/AD [8] [9] [10] [11]. Such vehicle systems require continuous advancement and validation, which is a time-consuming and cost-intensive process. They form a basis for the reduction of serious accidents especially for vulnerable road users such as pedestrians. Special consideration must be given to these road users in the development and validation of ADAS/AD.

Motion capture is a method of capturing and recording human or animal movements as well as the movement of rigid bodies by computers. Nowadays, it is mainly used in the production of films and video games for animation and digital interaction as well as in medical applications and high-performance sports to analyze the biomechanical movements of persons [12].

The innovative approach of this work is to combine motion captured person behavior in real time with a virtual test environment for highly automated vehicle systems. This enables the investigation of critical as well as realistic traffic situations between VRUs and automated vehicle systems without endangering people or vehicles. The diversity and authenticity of urban traffic reality can thus be fully represented and simulated.

For this purpose, motion capture systems are integrated into digital environments for the development and validation of ADAS/AD [8] [10] [11]. In this way, different digital avatars with biomechanically correct movement sequences are displayed in the environment. In addition, direct interaction of a pedestrian with an approaching digital vehicle is possible.

To avoid the loss of a degree of freedom in three-dimensional space *i.e.* gimbal lock [13], quaternions are used throughout this approach. Quaternions extend the number system of complex numbers. A quaternion is denoted as [14]:

$$q = w + xi + yj + zk = (w, x, y, z)^{\mathrm{T}} = (w, v)^{\mathrm{T}}$$

Whereas w, x, y, z are real numbers and i, j, k are called imaginary units. The fundamental multiplication rule for the imaginary units is denoted as [14]:

$$i^2 = j^2 = k^2 = ijk = -1$$

To multiply two quaternions q_1 and q_2, the following equation is given, where $(\cdot)$ represents the dot product and $(\times)$ the cross product of two vectors [13]:

$$q_1 q_2 = \begin{pmatrix} w_1 w_2 - v_1 \cdot v_1 \\ w_1 v_1 + w_2 v_2 + v_1 \times v_2 \end{pmatrix}$$

Finally, to rotate a vector with a quaternion it well known that [13]:

$$a' = qaq^{-1}$$

where q^{-1} denotes the inverse of the quaternion q with $qq^{-1} = q^{-1}q = 1$ [13]. An inverse quaternion is calculated as [4]:

$$q^{-1} = \frac{1}{\|q\|^2}\overline{q} = \frac{1}{\|q\|^2}\begin{pmatrix} w \\ -v \end{pmatrix}$$

where $\|q\|$ denotes the Euclidean vector norm for a four-dimensional vector. It follows that for a quaternion q with unit length, the inverse quaternion q^{-1} equals its conjugate $\overline{q}$.

Further information on quaternions can be found in [13] and [14].

2. Test Field Setup

The overall virtual test environment consists of three essential parts: the virtual reality environment, the motion capturing and the vehicle testbed. To enable a valid test environment including interactions and real traffic behavior, real-time capability is necessary. To achieve this, a co-simulation network is used, where all system parts have an individual calculation unit, which frequently sent and emits required data.

The schematically setup of the virtual test field for testing highly automated vehicle systems is shown in **Figure 1**.

Central element of the setup is the virtual environment. It connects the real human measured by motion capturing hardware and the real vehicle at the vehicle testbed. Both measurement systems provide their data to the virtual reality (VR) scene. Thereby the VR environment represents a real inner city traffic behavior including buildings, roads and external conditions.

The focus of this paper is the capture, progression and implementation of real human movements, to include real pedestrian behavior into the testing procedure of real highly automated vehicle systems.

2.1. Motion Capture

Motion Capture is the process of recording real sequences of movements and transforming them into usable mathematical expressions by tracking reference points in space over time to represent a time to represent a three-dimensional representation of the motion [15]. In the frame of this work, an optical motion capture system with passive markers as reference points is utilized. In detail *OptiTrack* Prime^{x} 22 motion capture hardware from *NaturalPoint* is used [16]. The cameras illuminate the capture volume with infrared light to light up the retroreflective markers attached to a person's body. The markers are then captured by multiple cameras with a resolution in sub-millimeter space. Based on stereo

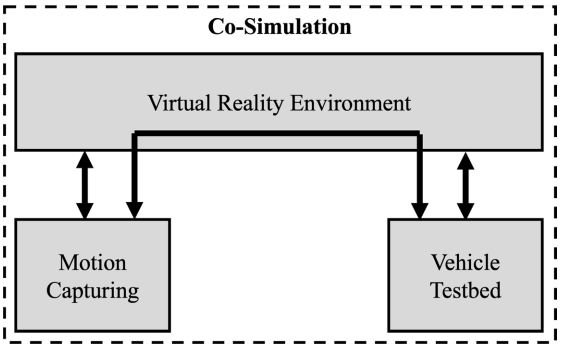

Figure 1. Schematically illustration of the test field setup.

triangulation the corresponding software *Motive* with its integrated skeleton solver estimate the position and rotation in quaternions of 21 body joints of a hierarchical humanoid skeleton. Tracking is possible with a native frequency of 360 Hz and thus provides a latency of 2.8 ms.

The schematical progression principle is shown in Figure 2.

Twelve cameras capture the PuT in an area of approximately 5 m × 5 m. This means, that every movement is captured in a highly accurate way, whereby the basis for highly realistic behavior of the virtual avatar in the test field is given.

2.2. Avatar Design

The setup of the avatar is essential for the biomechanical functionality in the virtual environment. Especially the three-dimensional position and rotation of the joints, as well as the number of joints is necessary to feed the information from the measured person to the avatar. Figure 3 shows schematically the progressions steps.

To enable a fully functional virtual humanoid avatar behavior, three fundamental development steps are important. First is the setup of the skeleton. To achieve a precise mapping of the motion capture skeleton, the avatars skeleton must be similar in relation to hierarchical structure and joint location inside the body. Therefore, every joint has six degrees of freedom (DoF), which enables every biomechanically movement possible. The individual length of the bones depends on the body dimensions. The joint located in the hip center is the hierarchical topmost joint of the body. Thus, a transformation of this joint affects all other body joints. In order to keep the proportions of the skeleton, displacement

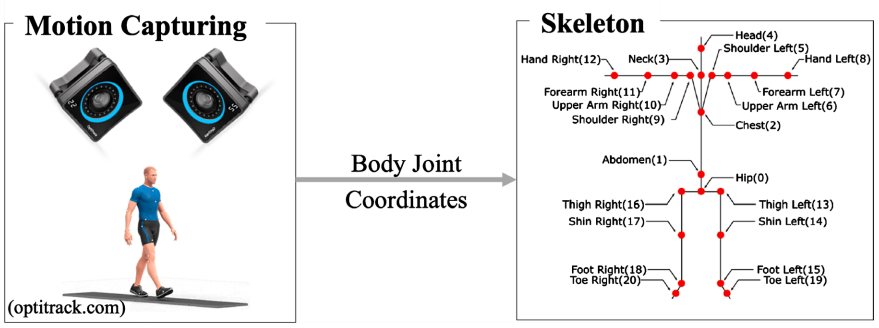

Figure 2. Schematical illustration of the measuring process.

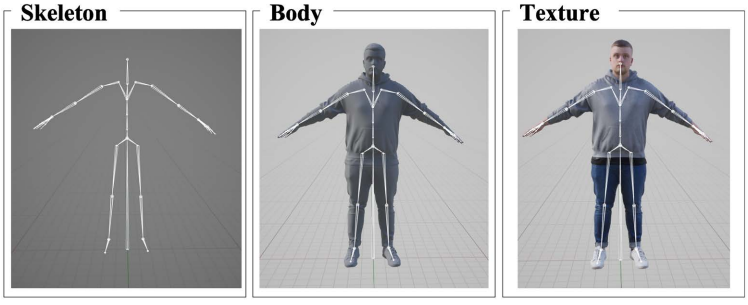

Figure 3. The three fundamental parts of the avatar design in VR.

is only applied to this specific joint. All other joints are affected by the parent-child-relationship and are transformed alone by the rotational data. Thus, a holistically determined system is given.

Second is the body development. This means the design of the mesh represents the body. It has all body dimensional information and represents the general appearance. The mesh consists of polygons, to which the following texture is applied. The mesh as well as the texture is determined by photogrammetry. Photogrammetry is the process of reconstructing position and shape of an object from photos [17].

An important part of merging the mesh and the skeleton is weight-painting. Here it is defined how much a deformation of a certain joint affects certain areas of the mesh. For example, the shoulder/armpit area, where movements of the shoulder joint cause deformation in the corresponding area of the mesh. By the individual value of the weight-painting, the distortion of the mesh in a certain area is determined.

Third is the texturing. Texturing gives the mesh the unicolor mesh an illustration by wrapping a two-dimensional image onto the three-dimensional mesh. The process used is called UV mapping. U are the horizontal and V are the vertical coordinates of the two-dimensional image [18]. Hereby polygons of the mesh are assigned to parts of the image and as a result are also affected by the weight-painting.

2.3. Remapping

The goal of this method is the universality in usage of the measured data. This means, that the virtual avatars dimensions are independent of those from the person under test in the motion capturing hardware. The progression steps of the remapping methodology are shown in **Figure 4** schematically.

The real persons movement is captured as described before. The PuT therefore wears a specific motion-capture-suit on which the retroreflective markers at specific points of the body (see: **Figure 4** "Sensors View") are attached to. In the next step, the scaling of the PuT-dimensions to the avatar-dimensions regarding the lateral displacement takes place. For this purpose, the leg length relation between the two is used to derive a scaling factor. The basic assumption to this is that the maximal step width is determined by leg length. In relation to this factor, the step range of the avatar is adapted. As described before, just the central joint—the hip center—is able to additionally perform lateral displacements and all joints are controlled by the rotational data of the motion capture. This means, that the adoption of the step width is realized by a scaled lateral displacement of the avatar.

Additionally, due to an arbitrary placement of the avatar in the virtual environment before initialization, the offset between the feet and ground is not necessarily zero. Thus is it not given that feet are on ground. To fix this, the global height of the ground underneath the avatar is determined. This information is

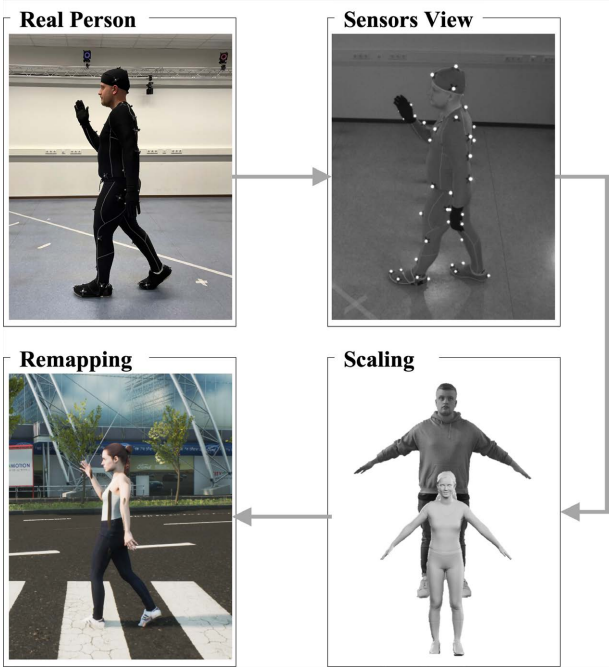

Figure 4. Schematically illustration of the remapping method.

used to position the avatar's hip center initially at the correct height above ground and thus the whole avatar standing on the ground.

3. Implementation in Virtual Reality

Next to the functional methodology, the way of implementation is important. Especially due to the fact of real time functionality, it is essential to increase the data efficiency of the communication between the motion capturing hardware and the VR environment.

In this approach the *Unreal Engine* 4.26 by *Epic Games* is used to visualize the motion capture data from the *OptiTrack* System. In Unreal, a mesh with an inherited skeleton is called a *Skeletal Mesh* [19]. To depict movements of a mesh performed by the real person, body joint coordinates are updated with data in the frequency defined by the motion capturing system. The data is then sent to Unreals animation pipeline to be visualized on the mesh. For this reason, an animation blueprint is used and an animation node inherits the following functionality. The functional process is shown in **Figure 5**.

At first, the data from the *OptiTrack* system is provided to the *Unreal Engine*. Due to different orientations of the coordinates systems and length units used, the data from the OptiTrack system has to be transformed into the coordinate system in *Unreal*, before the movement can be applied to the virtual avatar.

The skeleton of the *OptiTrack* system consists of 21 body joints with fixed names and identifiers. Since hierarchy, joint names and ID's of the Skeletal Mesh in Unreal must not fit necessarily, a manual allocation of the joints needs to be performed. Consequently, it is required to assign similar joints between the two skeletons that describe the same body joint. It is required, that the skeleton

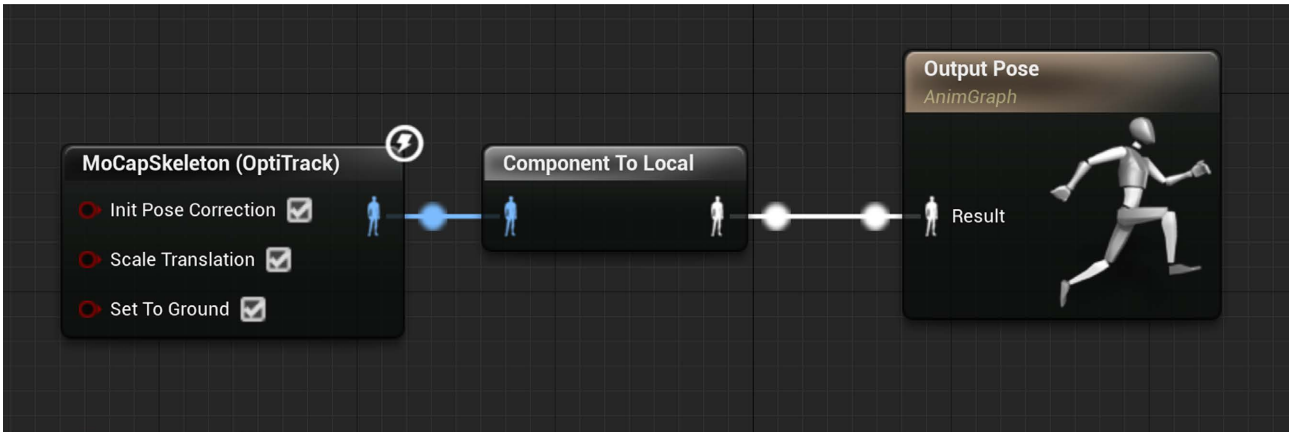

Figure 5. Dataflow from the motion capturing data to the virtual avatar

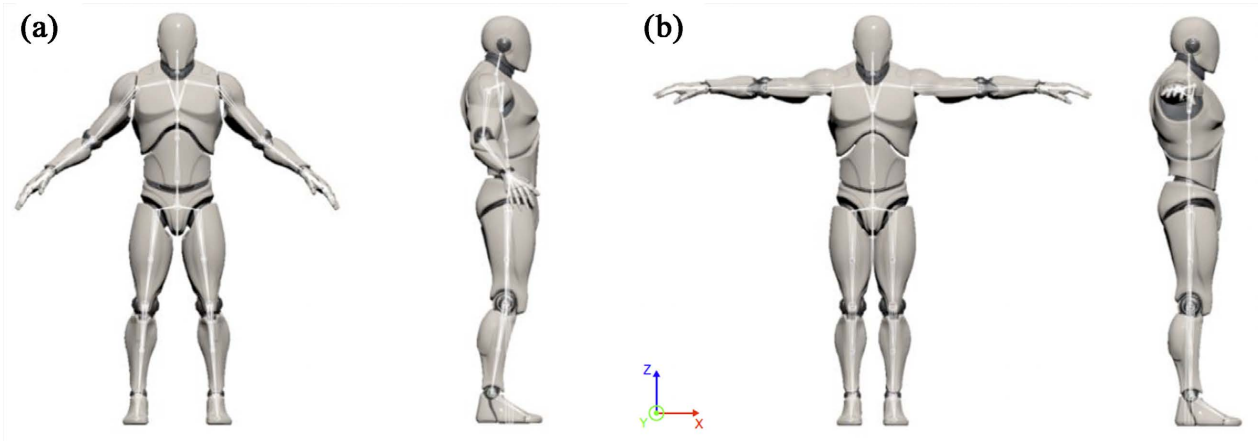

Figure 6. (a) Arbitrary initial pose of a humanoid avatar and (b) desired ideal T-pose.

joints need to be roughly in the same area in the mesh as they are in the *OptiTrack* skeleton.

Furthermore, the initial pose of the different avatars can differ. As shown in Figure 6 an avatar can have an arbitrary initial pose, whereas *OptiTrack* relies on a T-pose.

At first, the Unreal avatar is transformed into the ideal T-pose of the *OptiTrack* skeleton. This is important since rotation is applied additionally and thus relies on the initial pose. If the initial poses differ, rotational offsets occur. The transformation is performed by aligning specific body parts to certain axes of the component coordinate system. The torso needs to be aligned along the positive Z-axis and the legs along the negative Z-axis. The left arm is aligned along the positive X-axis and the right arm along the negative X-axis. The clavicle bones are not transformed since a deformation in most cases looks unnaturally. It is important to respect the hierarchical structure of the skeleton starting from the hip center. The joints of the feet, hands and head are a special feature. For this reason, the axis of the local coordinate system is used for the head and hands, which corresponds to the longitudinal axis of the bone in the superordinate joint. This is based on the assumption that all local coordinate systems of the

joints of the limbs and the neck and head are aligned in the same way in the starting pose. It is assumed that the feet are already on the floor in the starting pose. Accordingly, after the transformation into T-pose, the alignment of the feet is rotated back to the initial orientation via the ankle joint.

For the transformation, first the connection vectors *k*, which can be seen as a bone, between two hierarchical subordinated joints are determined and normalized. A rotation of a certain joint directly affects the bone pointing towards its subordinate joint. Therefore, the three-dimensional position vector of each joint P_{parent} and the superordinated joint P_{child} are subtracted and divided by the magnitude:

$$k = \frac{P_{child} - P_{parent}}{\left\| P_{child} - P_{parent} \right\|}$$

With this normalized connection vector k the orientation of each bone is known and the difference rotation q_{diff} in quaternions in relation to the orientation vector v is computed. v represents the axis of alignment for the several body parts as described above and q_{diff} the geodesic rotation between v and k. Computation is done via the dot product $\cos^{-1}(v \cdot k)$ for the angle α and the cross product $v \times k$ for the axis of rotation:

$$q_{diff} = \cos\left(\frac{\alpha}{2}\right) + \sin\left(\frac{\alpha}{2}\right)(v \times k)$$

Finally, q_{diff} has to be normalized to ensure it is a unit quaternion. To rotate the joint and the corresponding bone into the correct orientation, the inverse of the quaternion q_{diff} is computed. Since it is a unit quaternion, the inverse corresponds to the conjugate of the quaternion with an inverted vector part. Finally, it is multiplied with the initial rotation from the left to get the needed orientation for the T-pose:

$$q_{parent_{rotated}} = q_{diff}^{-1} q_{parent}$$

Next the vector k has to be rotated into the orientation necessary for the T-Pose. By the rules of quaternion algebra, k is multiplied it from the left with $q_{parent_{rotated}}$ and its inverse or conjugate from the right:

$$k_{rotated} = q_{parent_{rotated}} k q_{parent_{rotated}}^{-1}$$

This process is done for every joint inside the avatar's skeleton in hierarchical order to achieve the desired T-Pose. The corresponding body joint orientation in quaternions transformed into euler angles in initial pose (**Figure 6(a)**) and T-pose (**Figure 6(b)**) in the local coordinate systems are shown in **Table 1**.

4. Verification

To analyze the quality of the function, in the frame of this chapter a verification study is performed. Therefore, the same motion capture data is applied to three different dimensional virtual avatars. The avatar "Faye" has a body height of circa 130 cm, "Juliette" circa 165 cm and "Alex" a height of circa 190 cm. The

Table 1. Body joint angles in local coordinate system.

Name	Initial pose			T-pose		
	X (roll)	Y (pitch)	Z (yaw)	X (roll)	Y (pitch)	Z (yaw)
pelvis	90.00˚	89.79˚	90.00˚	−90.00˚	84.87˚	−90.00˚
spine_01	0.00˚	0.00˚	7.15˚	0.00˚	0.00˚	15.89˚
spine_02	0.00˚	0.00˚	−14.06˚	0.00˚	0.00˚	−9.59˚
spine_03	0.00˚	0.00˚	−2.78˚	0.00˚	0.00˚	−3.03˚
neck_01	0.00˚	0.00˚	23.51˚	0.00˚	0.00˚	3.48˚
head	0.00˚	0.00˚	−15.35˚	0.00˚	0.00˚	−2.25˚
clavicle_l	108.72˚	−61.85˚	−101.54˚	108.72˚	−61.85˚	−101.54˚
upperarm_l	7.67˚	−40.30˚	17.02˚	4.72˚	8.71˚	26.87˚
lowerarm_l	−3.61˚	10.40˚	30.36˚	−6.44˚	0.00˚	0.00˚
hand_l	−76.36˚	−2.50˚	0.41˚	−76.35˚	0.00˚	0.00˚
clavicle_r	108.72˚	−61.85˚	78.46˚	108.72˚	−61.85˚	78.46˚
upperarm_r	7.67˚	−40.30˚	17.02˚	4.72˚	8.71˚	26.87˚
lowerarm_r	−3.61˚	10.40˚	30.36˚	−6.44˚	0.00˚	0.00˚
hand_r	−76.36˚	−2.50˚	0.41˚	−76.35˚	0.00˚	0.00˚
thigh_l	8.56˚	7.03˚	1.52˚	8.75˚	0.00˚	4.50˚
calf_l	−5.74˚	−1.79˚	7.61˚	−5.62˚	0.00˚	0.00˚
foot_l	−0.42˚	−3.70˚	−8.06˚	−0.51˚	0.89˚	0.69˚
thigh_r	8.56˚	7.03˚	−178.48˚	8.75˚	0.00˚	−175.50˚
calf_r	−5.74˚	−1.79˚	7.61˚	−5.62˚	0.00˚	0.00˚
foot_r	−0.42˚	−3.70˚	−8.06˚	−0.51˚	0.89˚	0.69˚
ball_l	0.00˚	−0.01˚	91.88˚	0.00˚	−0.01˚	91.88˚
ball_r	0.00˚	−0.01˚	91.88˚	0.00˚	−0.01˚	91.88˚

Figure 7. Avatar sizes in direct comparison.

different avatars are shown in Figure 7.

To apply the same movement data set to all avatars, one movement is recorded in the motion capture studio. In this case a normal walking movement is chosen. It starts with the movement of one leg by lifting it and performing a step

forward. After another step with the other leg, the movement ends with the stillstand of both legs next to each other. The movement data set for the left leg is shown in Figure 8.

The most interesting part to rate the quality of the retargeting functionality of the walking movement to is the lower body section starting with the hip. Therefore, the investigation area focuses on the body joints 13 to 20 (see Figure 2). Figure 9 shows a comparison movement diagram of the main toe joint of the three avatars with the forward movement at the X-axis and the vertical movement at the Y-axis.

In summary, the same movement characteristics can be detected by all three avatar movements. The main toe joint lifts to perform a step and ends by rolling the foot of over the heel. This happens twice for normal step size and once to move the rear foot next to the front one before the stillstand. The step sizes in forward direction vary from about 110 cm for "Faye" to circa 165 cm for "Alex". In vertical direction it varies between 16 cm for "Faye" until 22 cm for "Alex". "Juliette" performs forwards step-sizes of 150 cm and vertical step-sizes of 20 cm.

5. Discussion

Due to the fact, that the movements of the different avatars are controlled by the

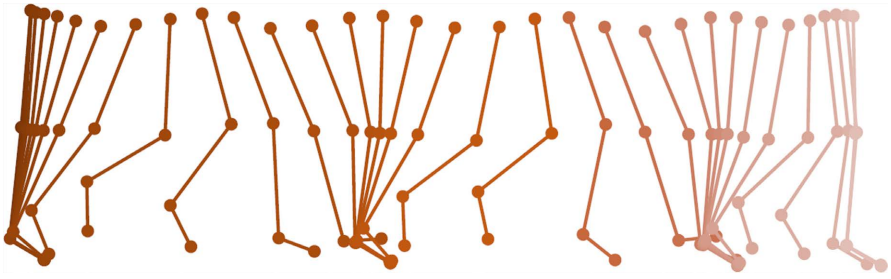

Figure 8. Walking movement data set.

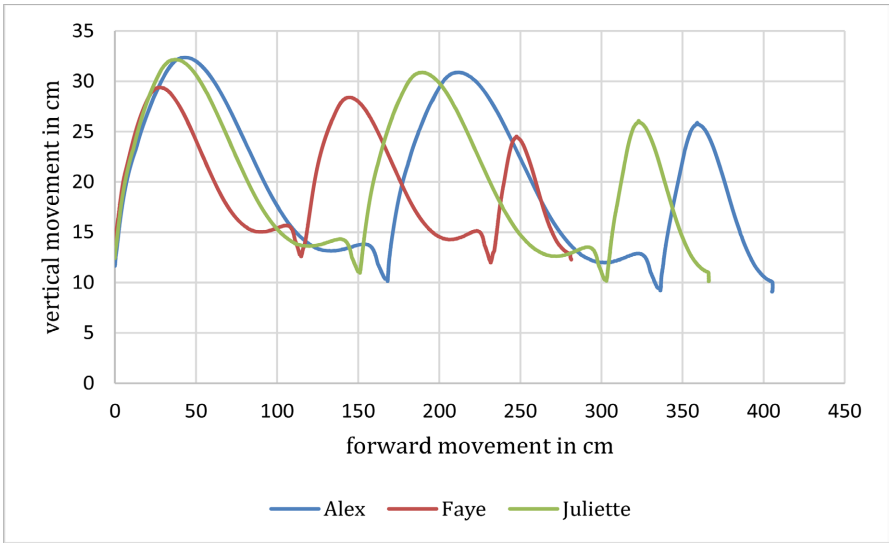

Figure 9. Comparison of walking movement data set for different avatars.

same movement data, the characteristics are the same. As described before, especially Figure 9 shows the functionality of the remapping algorithm. The step-size varies in vertical as well as in horizontal direction, depending on the different body-dimensions. This confirms the general plausibility of the used algorithm.

Additionally, the diagram in Figure 9 shows a variation of the ground truth data. At the start point, all avatars start with their main toe joint at about 12 cm above the floor. The next floor contact after the performed step is on average at about 2 cm below. This could be justified by the ground truth information of the georealistic virtual city model. Even when the PuT is walking on a flat ground, the avatar always uses the virtual map information for the ground truth.

Another point to discuss is the originality of the walking illustration and their fit to the different kinds of avatars. The used walking information of the PuT are based on a male walk, whereby these characteristics do not perfectly fit to a petite female avatar. In a further processing step, the algorithm could be extended to include gender-specific characteristics.

It is hard to establish an absolute real-time system, since large amounts of data have to be recorded, processed and output. For the evaluation of real-time capability, criteria must be defined in order to allow an assessment. For this purpose, it is assumed that the human eye perceives fluid movements from around 30 Hertz. The following framework conditions apply: The *OptiTrack* system consisting of Prime^{x} 22 cameras has a native frame rate of 360 Hz, which corresponds to a system latency of 2.8 ms.

6. Conclusion

The developed algorithm shows a valid functionality for the remapping as well as for the general data transition from the reality to the virtual environment. Independent of the PuT and their dimensions, all body joints were processed correctly to the different avatars of this study. Further, the procession works in system real time, which means a fluid sequence of images in the video stream. Finally, the step sizes were adopted correctly in context of the different body dimensions of the avatars.

Acknowledgements

The Project is supported by the Ministry of Economic Affairs, Innovation, Digitization and Energy of North Rhine-Westphalia, AVL Deutschland GmbH and by HHVISION GbR.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Statista (2020) Anteil der PKW mit Fahrassistenzsystemen in Deutschland. https://de.statista.com/statistik/daten/studie/1083873/umfrage/anteil-der-pkw-mit-f

ahrassistenzsystemen-in-deutschland/

[2] Statista (2020) Meinungsumfrage zu Assistenzsystemen in Autos in Deutschland. https://de.statista.com/statistik/daten/studie/1108736/umfrage/meinungsumfrage-zu-assistenzsystemen-in-autos-in-deutschland/

[3] Wachenfeld, W. and Winner, H. (2015) Die Freigabe des autonomen Fahrens. In: Maurer, M., Gerdes, J.Ch., Lenz, B. and Winner, H., Eds., *Autonomes Fahren*, Springer Vieweg, Berlin, Heidelberg, 439-464. https://doi.org/10.1007/978-3-662-45854-9_21

[4] Brissard, A. and Schyr, C. (2016) DrivingCube—A Novel Concept for Validation of Powertrain and Steering Systems with Automated Driving. *Proceedings of the* 13*th International Symposium on Advanced Vehicle Control* (*AVEC*' 16), Munich, 13-16 September 2016.

[5] Gadringer, M.E., Schreiber, H., Gruber, A., Vorderderfler, M., Amschl, D., Bösch, W., Metzner, S., Pflügl, H. and Paulweber, M. (2018) Virtual Reality for Automotive Radars. *Elektrotechnik & Informationstechnik*, **135**, 335-343. https://doi.org/10.1007/s00502-018-0620-9

[6] Al-Saidi, O., Müller, S., Zech, A. and Schyr, C. (2018) Bewertung und Analyse von Fahrzeugregelsystemen im fahrdynamischen Grenzbereich mit einem Vehicle-in-the-Loop Prufstand in: SIMVEC—Simulation und Erprobung in der Fahrzeugentwicklung, Seite 495-510.

[7] Weiskopf, M., Wohlfahrt, C. and Schmidt, A. (2015) Absicherung eines radarsensors im systemverbund mit der hardware-in-the-loop testtechnologie. In Klenk, H., Keller, H.B., Plodereder, E. and Dencker, P. (Eds.), *Automotive—Safety & Security* 2014. Gesellschaft fur Informatik e.V., Bonn.

[8] Degen, R., Ott, H., Overath, F., Klein, F., Hennrich, M., Schyr, Ch., Leijon, M. and Ruschitzka, M. (2021) Virtual Urban Traffic Infrastructure for Testing Highly Automated Mobility Systems. In: Krieger, J., Ed., *Fachkongress Digitale Transformation im Lebenszyklus der Verkehrsinfrastruktur*, Esslingen, 317-330.

[9] Degen, R., Ott, H., Overath, F., Schyr, C., Leijon, M. and Ruschitzka, M. (2021) Methodical Approach to the Development of a Radar Sensor Model for the Detection of Urban Traffic Participants Using a Virtual Reality Engine. *Journal of Transportation Technologies*, **11**, 179-195. https://doi.org/10.4236/jtts.2021.112012

[10] Degen, R., Ott, H., Overath, F., Klein, F., Schyr. C., Leijon, M. and Ruschitzka, M. (2021) Integration of Driving Physical Properties into the Development of a Virtual Test Field for Highly Automated Vehicle Systems. *NAFEMS World Congress* 2021, Salzburg, 25-29 October 2021, 2-20.

[11] Degen, R., Ott, H., Klein, F., Shankavaram, R., Leijon, M. and Ruschitzka, M. (2021) CityInMotion—A Virtual Urban Test Environment for Automated Mobility. *International Symposium for Development Metodology*, Wiesbaden, 9-10 November 2021, 1-13.

[12] Midori, K. and Brian, W. (2008) MoCap for Artists: Workflow and Techniques for Motion Capture. Elsevier/Focal Press, Amsterdam.

[13] Hanson, A.J. (2008) Visualizing Quaternions. Elsevier Morgan Kaufmann, Amsterdam.

[14] Hamilton, W.R. (1844) II. On Quaternions; or on a New System of Imaginaries in Algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **25**, 10-13. https://doi.org/10.1080/14786444408644923

[15] Menache, A. (2011) Understanding Motion Capture for Computer Animation. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann, Burlington.

[16] OptiTrack (2022): Prime^{x} 22—Specs. https://optitrack.com/cameras/primex-22/specs.html

[17] Karl, K. (2004) Photogrammetrie. Band 1: Geometrische Informationen aus Photographien und Laserscanneraufnahmen. 7., vollständig bearbeitete und erweiterte Auflage. Walter de Gruyter, Berlin.

[18] Oliver, S. (2005) Stereoanalyse und Bildsynthese: Mit 6 Tabellen. Springer-Verlag, Berlin.

[19] Epic Games Inc. (2022): Skeletal Meshes. https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Types/SkeletalMeshes/

List of Symbols

P_{child}	Position of a child joint
P_{parent}	Position of a parent joint
k	Bone vector
v	Axis of alignment
q_{diff}	Difference quaternion
$q_{parent_{rotated}}$	Rotated parent joint orientation
q_{parent}	Parent joint orientation
$k_{rotated}$	Rotated bone vector