Schema for Motion Capture Data Management

Ali Keyvani[1,2], Henrik Johansson[1], Mikael Ericsson[1],
Dan Lämkull[3], and Roland Örtengren[4]

[1] University West, Department of Engineering Science, 46186 Trollhättan, Sweden
[2] Innovatum AB, Production Technology Centre, 46129 Trollhättan, Sweden
[3] Volvo Car Corporation, Dept. 81022/PVÖS32, SE-405 31 Göteborg, Sweden
[4] Chalmers University of Technology, Department of Product and Production Development,
SE-412 96 Göteborg, Sweden
`{Ali.Keyvani,Mikael.Ericsson,Henrik.Johansson}@hv.se`
`dlamkull@volvocars.com, roland.ortengren@chalmers.se`

Abstract. A unified database platform capable of storing both motion captured data and information about these motions (metadata) is described. The platform stores large motion captured data in order to be used by different applications for searching, comparing, analyzing and updating existing motions. The platform is intended to be used to choose a realistic motion in simulation of production lines. It is capable of supporting and handling different motion formats, various skeleton types and distinctive body regions in a uniform data model. Extended annotating system is also introduced to mark the captured data not only in the time domain (temporal) but also on different body regions (spatial). To utilize the platform, sample tests are performed to prove the functionality. Several motion captured data is uploaded to the database while MATLAB is used to access the data, ergonomically analyze the motions based on OWAS standard, and add the results to the database by automatic tagging of the postures.

Keywords: Motion Capture Database, Virtual Production Systems, Digital Human Modeling, Computerized Ergonomic Analysis.

1 Introduction

Many production systems face challenges because of costly changes imposed by late modifications in the product design/production line. When there is a need to simulate such production systems, one challenge is to integrate the human behavior together with other production elements in the simulation environment. Digital Human Modeling (DHM) tools address this problem by integrating human motions into the simulation platform. Process planners often use these DHM tools to simulate different work postures for ergonomic analysis or to investigate possible hand reach and line of sight. The advantage with DHM tools is the ability to make ergonomic analysis of work environments to prevent musculoskeletal disorders and thereby increase quality and productivity [1].

However, generating natural human motions is not an easy task. Redundancy, caused by additional degrees of freedom, restrains conventional mathematical

V.G. Duffy (Ed.): Digital Human Modeling, HCII 2011, LNCS 6777, pp. 99–108, 2011.

solutions. It is identified that motion-solving algorithms based on real motion captured data is one promising solution. This paper describes how database techniques can be used to facilitate the usage of motion capture (Mocap) data within DHM tools. The database techniques provide a framework for handling motion data, not only for DHM tools but also for every tool that uses motion data. The outputs from Mocap systems are generally stored in datasets that include the motion data and information about the captured data. In section 4, a general workflow is presented of how datasets are used to populate the database. Annotation methods are also introduced to describe information about the motions. The experiments in section 5 verify that the database can handle data according to the desired requirements.

2 Background

The need to use manikins and DHM tools in the production processes often comes from big companies such as car manufactures. As a result, many of today's process simulation vendors have integrated DHM tools into their software, such as Jack from Siemens PLM software [2]. One of the first applications of DHM was to analyze if it is possible to reach a specific location or control [3]. Today, DHM models are commonly used during the design of vehicle. Common industrial applications are to model the interaction between driver, passengers and the interior in a car [4-6]. Another example of an industrial task is to analyze if a human can reach a specified hand position in an assembly operation in a digital simulation of a factory [7, 8]. A large sector of DHM outside the industrial usage is entertainment, such as games and movies [9].

The simulation of human motions and postures in the production line is often done using mouse and keyboard. A natural posture of the manikin is often judged by a production engineer via visualization and assumptions [10]. However, manual manipulation of manikins are both time consuming and not always reliable. To help end-user generating postures, several techniques have been introduced to predict realistic postures. Some of them are based on pure mathematical methods, normally referred to as inverse kinematic methods [11]. Optimization tools are also used in order to minimize joint stresses and predict a 2D posture based on these [12, 13].

Furthermore, other methods try to use real human motions as a reference to predict postures. A common need in all these techniques, which use real human motions, is to have structured databases of recorded human motions. These databases shall include changes in body joints during time and information about the characteristics of the motions (annotations).

Mocap is a common way to collect real human motion data. Several different commercial Mocap systems are available today and can be categorized in three main groups: Optical systems, magnetic systems and mechanical systems. Other methods exist such as ultrasonic and internal systems [14]. As a result, considerable work has been done to introduce efficient techniques for collecting and manipulating the human Mocap data. These techniques have been divided into two main approaches: database techniques [15], and annotation of motion sequences [16]. Furthermore, to generate a posture from real human data, different methods have been suggested. Examples are: using statistical models [17], treating recorded motions as a manikin memory [18], warping an existing motion by adjusting the motion graphs [19], synthesizing recorded data with the help of annotations [20], and mixing recorded motions with inverse kinematic techniques [21, 22].

3 Problem Description

Mocap data is recorded using a human subject performing a specified task. The subject usually wears a special suit with markers. The position of each marker is tracked by a Mocap system. It is both time consuming and expensive to use Mocap system to acquire and record new motion for each specific task.

Recorded Mocap data from different sources can normally not be reused because of a number of reasons. One reason is that different file formats are used to save the Mocap data due to different types of systems. Another reason when reusing Mocap data is the unique nature of each motion, which in turn is sensitive to changes in the skeleton configuration. Depending on the application, different skeleton configurations are used. Variations in these skeleton configurations can be e.g. number of segments, joint configuration and the length of individual segments. Fig. 1 shows an example of two skeletons with different number of segments. A third reason is the variations in naming convention due to lack of standards. Each research group can have its own naming conventions both for segment parts and/or for tagging the performed action. For example, use of trunk/chest in naming of segments and walk/pace/stride in naming an action.

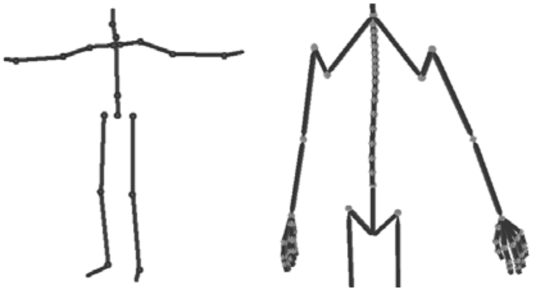

Fig. 1. Two different types of human skeleton, where lines and dots representing the segments and the joints

Generally, Mocap data are collected in datasets, which are usually folders named with a proper term describing the action and populated with Mocap files. One problem with large datasets of motions is the limited possibility to search within Mocap files. One possibility to facilitate search will be to integrate the Mocap data in a database. A database is a system provides means to manage, store, search, retrieve and process large amount of structured data [23]. Such a system should also allow the user to combine several existing motions and store it as a new entity.

In this article, a unified database platform is described capable of storing both Mocap data and information about these motions (metadata). It stores Mocap data that can be used, by different applications, for searching, comparing, analyzing and updating existing motions. A new annotating system is proposed to mark the captured data. Compared to existing methods [1], the proposed technique applies annotations both on the time domain (temporal) and on the different body regions (spatial). The advantage of this annotation method is that it will improve the search functionality by providing better-segmented search criteria.

4 Database Architecture

The main concept of the proposed platform is to provide a generic workspace, which can manipulate Mocap data in an organized way. It also provides efficient ways for other applications, especially industrial virtual production/ergonomic tools, to access the data, analyze them and feedback the results to the platform. Later on, these data will be the source for DHM tools to find and generate realistic and valid motions. For doing so, a standard database approach with a three-tier (Data, Query, and Application) architecture was implemented [23], see Figure 2. The inner layer, called data layer, keeps the data within tables. The middle layer, called query layer, is the link between the inner and outer layer. It handles the queries generated from the outer layer and communicates with the data layer. The query layer also insures that the data is consistent and well defined by means of relationships, attributes and rules. In fact, accurate presentation of this layer, called data schema, is one of the contributions in the paper. The outer layer, called application layer, is the interface communicating with the user. It can be any application that needs to communicate with the data layer. The data can travel in both directions (publish and update, or search and retrieve) as long as the applications talk to the query layer in the standard language.

One benefit of having such architecture is that the application layer can access the data regardless of the application type. This means that many applications can communicate with the data core using a standard query builder (SQL in this case) without worrying about how the data is handled within the system. Furthermore, different functions such as visualization, analysis, updates and data combination can be done by different applications simultaneously.

Preparations are needed to populate the database with motion files. This means that for one time, each Mocap file shall be interpreted and parsed into the database according to a specific workflow. To accomplish this, two separate procedures will be discussed in the rest of the paper, data storage and data retrieval. However, before going into these details, different data types to be handled by the database are presented in the next subsection.

4.1 Data Classes/Objects Definition

In this section, the implementation of the data schema is presented. The target of the schema is to satisfy a number of key requirements for a Mocap database to be efficiently utilized. These key requirements are:

- Support of skeleton variations
- Support of customized body segmentation and partial (upper body, hands, etc.) motion description
- Support of two domains (spatial and temporal) for annotating
- Support of categorized annotation system

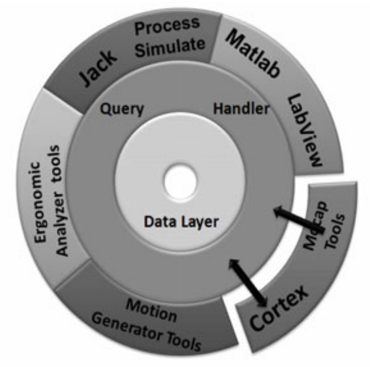

Fig. 2. Data, Query and Application layers in a three-tier architecture

The data schema is based on several data objects (D.O.s), described as follows:

Marker D.O.: Markers (in optical Mocap) are reflective objects attached to the subject (actor) body, and are the targets for the Mocap device to track them accurately.

To define a motion from the Mocap output, one needs to map the marker data to the joint centroids of the subject body. This means that first there should be a Skeleton (hierarchy) defining how the body segments are connected to each other; and second there should be a table defining the length of these segments for each subject (person) performing the motion. Finally, the relational movements of these segments shall be defined by means of changes in the joint values (rotations). Regarding this, three different D.O.s are needed in the schema to precisely define a motion independent of the motion subject.

Skeleton D.O.: As mentioned, the hierarchy of the body segments shall be defined for a specific motion in the Skeleton Data object. Notice that, depending on the application, different skeletons may be defined. This type of D.O. defines a hierarchy tree of parent-child segments connected to each other.

Actors D.O.: Assuming identical skeleton hierarchy, different subjects can be distinguished by assigning their specific segment lengths to the same skeleton type. An Actors D.O. contains these data in addition to other general data about the motion subject such as gender, weight, height, etc.

Joint Value D.O.: having a skeleton and subject's segment length data, a motion can be sufficiently defined by the relative rotational values in each joint during time. Generally, three degrees of freedom are needed to define a generic rotation, and therefore Joint Value D.O.s are pieces of data containing the joint name and three rotation values at a specific time.

BodyZone D.O.: In addition to above, motions are not necessarily performed full body and retrieval of full body motions are not needed all the times. Therefore, there should be a way to define a motion partially based on different body zones. The BodyZone D.O. defines how a number of joints can be grouped to form a zone. Moreover, it is possible to form a new body zone by combining existing body zones (Both_Hands = Right_Hand + Left_Hand).

Annotation D.O.: The proposed data model is able to annotate motions both temporally (frame range) and spatially (body zones). Therefore, an Annotation D.O. contains information about the type of annotation, tag name, target motion object, frame range and the applicable body zone.

4.2 Data Storage Workflow

In order to make the Mocap data generic and reusable, several actions need to be performed while they are being populated into the database (Fig. 3).

When receiving a Mocap file, a new object called Basic_Motion D.O. is created. This object is the core object in the database placed in a table called "Basic_Motion". This table consists of some mandatory and optional attribute fields, which primarily describe the motion and consist of relational fields, which connect the Basic_Motion object to other tables. Examples of mandatory attribute fields are the unique motion ID (primary key), number of frames and data format. Examples of optional fields are frame rate, primary motion description, date and number of cameras.

Examples of mandatory relational fields are a relation key to the "Skeleton_Type" table and a relation key to the "Body_Zone" table. Example of optional relation is relation keys to the "Actors" tables which show who was the primary subject of the motion. Notice that the schema assumes that the motions can be considered independent of the subject and can be applied to a new subject by changing the reference key in the segment data table. As a result, this is an optional and not a mandatory relation when defining the Basic_Motion object.

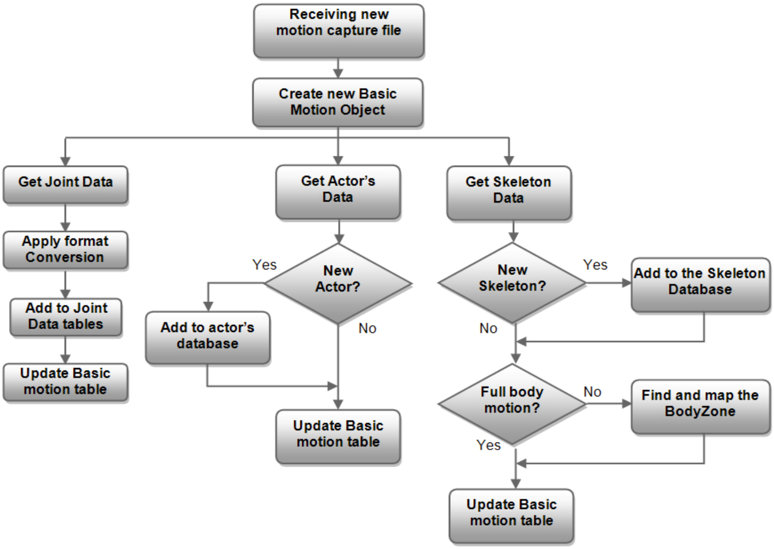

Fig. 3. Data Storage Workflow

For the next step, skeleton data, subject data (segment length) and joint data are separated, analyzed and stored in the database. Most of the current commercial

Mocap file formats (.htr by Motion Analysis, .asf/amc by Acclaim) include these data. “htr” format keeps the hierarchy, segment lengths and joint values in one file and “asf/amc” keeps the skeleton information (.asf) and joint data (.amc) separately.

As mentioned before, there can be different skeleton definitions and a motion can be performed by many subjects. When receiving a new motion, the Skeleton hierarchy will be checked to see if it exists in the database. In case of new skeleton, the database is populated by adding the skeleton and mapping the previously defined body zones to it. For example if a body zone called Right_Hand was defined for previous skeletons, the definition shall be updated by assigning which joints from the new skeleton are also members of this body zone. Next, the basic motion D.O. is updated with the correct skeleton type and the assigned body zone (notice that not all the motions are full body motions). Similar scenario will be executed for a new subject definition.

Finally, the joint values are stored in tables. For each joint three values, corresponding to Euler rotation angles will be stored. In case the file format was not based on Euler angles, a conversion is performed on the data to find the equivalent Euler angles and store them in the joint data tables.

4.3 Annotate, Search and Retrieve

The main goal of storing motions in a database is to provide access for different applications, for example to further analyze the motion data. The results of these analyses are metadata, which is stored in the database as annotations (tags). The proposed model is able to assign tags, not only temporally (frame range) but also spatially (body zone). This means that the annotating system provides a 2D space to describe the motion pieces. This approach will help the user applications to request specific motions by searching for part of the body and part of the motion in the database.

In addition, the proposed data schema provides flexibility to generate both simple and complex queries to search and retrieve motions. These queries are sent from the user applications to the database for such purposes as:

- In a simple scenario, the query is retrieving a complete motion or part of a specific motion for visualizing.
- Analysis tools can send queries to retrieve part of a motion, analyze it, and send back the results in form of annotations to the database.
- Motion synthesizers can generate queries to search for a specific action (previously annotated) with the closest match to a certain scenario.

5 Experiments

Sample motions were generated using an in-house Mocap laboratory. The laboratory consists of 10 Hawk cameras and Cortex 2.0 software from Motion Analysis. The generated motions were all full body motions of different lengths (150 to 2500 frames), and the generated output was “.htr” format. Two types of skeleton definitions (27 and 15 segments) were used, and four different subjects performed the motions. Moreover, eight different body zones were defined and assigned to the skeleton definitions. The data schema was implemented in Microsoft Access, and MATLAB was used as a user application to access the database using the SQL toolbox.

5.1 Human Motion Analysis and Data Update

The focus of this work is to reuse real human motions in different virtual production applications. Here MATLAB reuses motion data for ergonomic analyses. MATLAB evaluates and annotates postures frame by frame. Among other possible standards, The Ovako Working Posture Analysis System (OWAS) [24] is used as an example of an ergonomic tool to analyze the motion. The OWAS concerns four different categories, which are: Back, Upper limbs, Lower limbs and Carrying Load. Based on the joint values in each frame the position of the body segments are decided. This result in a 4-digit code, were each digit corresponding to a category. The final combination will decide the ergonomic condition of a posture. These results are updated into the database, so other applications can use them while searching for an ergonomically acceptable posture (Fig. 4 up-left).

5.2 Annotation Report

As mentioned, the OWAS analysis results were fed to the data schema by means of annotations. In addition, performed motions were also annotated based on the type of actions that subjects are doing. Tags such as walk forward, walk backward, seated, stand on one foot, etc. were applied to the corresponding frame ranges and body zones. The annotation tables enable the user applications to generate several types of reports and diagrams. Fig. 4 (right) shows the overall ergonomic condition of a selected motion. Based on the OWAS code each frame is situated in a colored zone.

5.3 Visualization

Again, MATLAB and Cortex 2.0 were used to search for part of a motion in the database, retrieve it and visualize it. Search functions can be any type of query and the result can be a single frame or a range of frames for a full body or a partial body motion. Figure 4 (down-left) shows a sample search result for a range of motion tagged with “Walk Backward”.

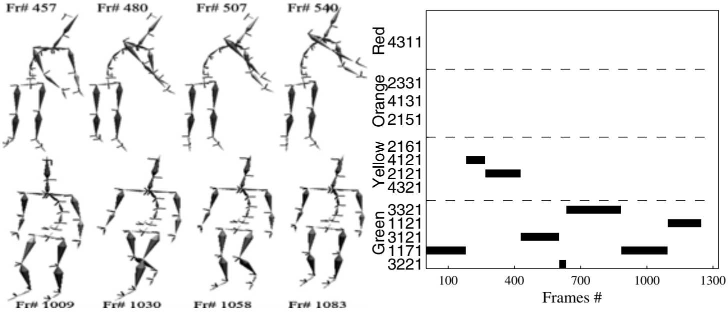

Fig. 4. Up-Left: frame samples presenting postures tagged with OWAS code 3121 (twisted, standing on both feet and with hands below shoulder). Down-left: frame samples presenting walk backward. Right: OWAS report of a sample motion during time (Frame #).

6 Conclusion and Discussions

The data schema presented in this article successfully stores and retrieves different motions regardless of variations in the skeleton types. It is also possible to annotate the motions based on customized body zones independent of the skeleton types.

The presented data schema is validated using different type of queries, such as motion level queries, which are searching for annotations, and joint level queries, which are searching for a specific posture. The implemented annotation system enables the applications to search for a specific motion in a part of the body.

The data schema is confirmed to be a suitable platform for the DHM and ergonomic tools. It is showed that via interpreters, ergonomic tools can communicate with the platform and examine motions frame by frame for unhealthy work postures. It is also shown that DHM tools can search among real human data for a required posture or motion.

References

1. Eklund, J.A.E.: Relationships between Ergonomics and Quality in Assembly Work. Appl. Ergon. 26, 15–20 (1995)
2. Siemens Product Lifecycle Management Software Inc., `http://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/jack/index.shtml`
3. Ryan, P., Springer, W., Hiastala, M.: Cockpit Geometry Evaluation (1970)
4. Chaffin, D.B.: Digital human modeling for vehicle and workplace design. In: Society of Automotive Engineers, Warrendale, Pa (2001)
5. Yang, J.Z., Kim, J.H., Abdel-Malek, K., Marler, T., Beck, S., Kopp, G.R.: A new digital human environment and assessment of vehicle interior design. Comput. Aided Design. 39, 548–558 (2007)
6. Reed, M.P., Huang, S.: Modeling vehicle ingress and egress using the Human Motion Simulation Framework. In: SAE Digital Human Modeling for Design and Engineering Conference (2008)
7. Wenzel, S., Jessen, U., Bernhard, J.: Classifications and conventions structure the handling of models within the Digital Factory. Comput. Ind. 56, 334–346 (2005)
8. Mavrikios, D., Karabatsou, V., Pappas, M., Chryssolouris, G.: An efficient approach to human motion modeling for the verification of human-centric product design and manufacturing in virtual environments. Robot. Cim.-Int. Manuf. 23, 533–543 (2007)
9. Menache, A.: Understanding motion capture for computer animation. Morgan Kaufmann, Burlington (2011)
10. Lämkull, D., Hanson, L., Örtengren, R.: Uniformity in manikin posturing: a comparison between posture prediction and manual joint manipulation. International Journal of Human Factors Modelling and Simulation 1, 225–243 (2008)
11. Jung, E.S., Kee, D., Chung, M.K.: Upper body reach posture prediction for ergonomic evaluation models. Int. J. Ind. Ergonom. 16, 95–107 (1995)
12. Zhang, X.D., Kuo, A.D., Chaffin, D.B.: Optimization-based differential kinematic modeling exhibits a velocity-control strategy for dynamic posture determination in seated reaching movements. J. Biomech. 31, 1035–1042 (1998)
13. Lin, C.J., Ayoub, M.M., Bernard, T.M.: Computer motion simulation for sagittal plane lifting activities. Int. J. Ind. Ergonom. 24, 141–155 (1999)

14. Kitagawa, M., Windsor, B.: MoCap for artists: workflow and techniques for motion capture. Elsevier/Focal Press, Amsterdam, Boston (2008)
15. Awad, C.: A Database Architecture for Real-Time Motion Retrieval. In: 7th International Workshop of content-based Multimedia Indexing (CBMI 2009), pp. 225–230 (2009)
16. Kahol, K., Tripathi, P., Panchanathan, S.: Documenting motion sequences with a personalized annotation system. IEEE Multimedia 13, 37–45 (2006)
17. Faraway, J.J.: Regression analysis for a functional response. Technometrics 39, 254–261 (1997)
18. Park, W., Chaffin, D.B., Martin, B.J., Yoon, J.: Memory-based human motion simulation for computer-aided ergonomic design. IEEE T. Syst. Man Cy. A 38, 513–527 (2008)
19. Witkin, A., Popovic, Z.: Motion warping. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 105–108. ACM, New York (1995)
20. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. ACM T. Graphic. 22, 402–408 (2003)
21. Faraway, J.J.: Data-based motion prediction. Society of Automotive Engineers, New York, NY, ETATS-UNIS (2003)
22. Aydin, Y., Nakajima, M.: Database guided computer animation of human grasping using forward and inverse kinematics. Comput. Graph.-Uk 23, 145–154 (1999)
23. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book. Prentice Hall, Englewood Cliffs (2008)
24. Louhevaara, S.: OWAS—A method for the evaluation of postural load during work. Institute of Occupational Health and Centre for Occupational Safety (1992)