On-line Generation of Customized Human Models based on Camera Measurements

Koen Buys*†, Dorien Van Deun‡, Tinne De Laet†, Herman Bruyninckx†

† Katholieke Universiteit Leuven, Belgium, Division PMA – Department Mechanical Engineering

‡ Katholieke Universiteit Leuven, Belgium, Division BMGO – Department Mechanical Engineering

Abstract

This paper presents two algorithms, one monocular-camera and one multiple-view camera based, to estimate several anthropometric parameters of human skeletons underlying the human avatar consisting of a human skeleton and a mesh model. This results in an on-line implementation, which can use cheap sensors to give adequate results for biomechanical tracking with only four to seven parameters in the estimation process.

The first, single-view camera based (monocular-camera), algorithm uses the MakeHuman (Bastioni and Flerackers, 2007) mesh model to estimate several anthropometric parameters that are directly or indirectly observable, such as length, hip and waist circumference, age … .

The second, multiple-view camera based, algorithm uses a voxel-based method to estimate both the length parameters and the weight of the person. This is achieved by a multiple-view visual hull reconstruction that overfits the person in order to iteratively increase the number of voxels to better match the actual human's shape. This voxel-technique is furthermore used for person segmentation (head, torso, arms, and legs).

The proposed methods are currently being extended to combine visual images with a depth image (Microsoft Kinect) or a laser range image (Hokuyo), since these sensors directly provide a point cloud of the person, hereby replacing the otherwise computational expensive voxel technique.

Keywords: Anthropometry, Motion Capture, Perception and Cognition

1. Introduction

Nowadays more and more robots tasks are targeted to function in common household environments, hence increasing the number of possible human-robot interactions. These interactions require perception of the human and raise the need of individually adjusted human skeleton and mesh models (or combined in 'avatars'). These models are needed for gesture recognition, identification, … and need to be personalized.

The need for automatic avatar (human skeleton and mesh) calibration in order to automatically obtain a personalized avatar, is also becoming more and more common in the entertainment industry (motion animation, games (Microsoft Kinect), teleconferencing, …) and in the medical sector (gait analysis, physiotherapy, revalidation, …).

Although the human motion capture research is quite advanced and still ongoing (Poppe 2007), most of the avatars are still created manually and provide an ad-hoc solution for those applications. Only a small part of the human capture research focuses on the automatic calibration and creation of these avatars (Ahmed et al. 2005, Villa-Uriol et al. 2003). Furthermore, most of them are animation or game-oriented while very few evaluate the usability of automatic avatar calibration for biomechanical purposes. In this paper we present two algorithms for automatic avatar creation and evaluate their applicability for biomechanical purposes (Van Deun et al. 2011). Both presented algorithms can be applied on-line with the use of cheap camera

*Corresponding author. Email: koen.buys@mech.kuleuven.be sensors as the Microsoft Kinect or the Primesense PSDK 5.0. Both algorithms consider the achieved accuracy versus the calculation time, where on-line calculation was considered of higher importance in this paper.

Section 2 of this paper presents the two algorithms for automatic avatar calibration: Section 2.1 first discusses the method using a monocular-camera and a MakeHuman mesh, while Section 2.2 presents the voxel-based method using multiple camera views. Section 3 presents the experimental results and gives the specific details of the experimental setup. Section 4 discusses the achieved results in the context of this research and the accompanying paper (Van Deun et al. 2011), while Section 5 formulates the general conclusion and elaborates on future work.

2. Materials and Methods

2.1. Avatar creation based on the MakeHuman mesh with a monocular camera

The goal of this algorithm is to estimate anthropometric parameters like length, width, … along with an initial estimate of the internal joint locations of the human skeleton underlying the avatar. This is done by a readily available monocular camera, hereby facilitating the use of the algorithm in a lot of (for instance robot) applications.

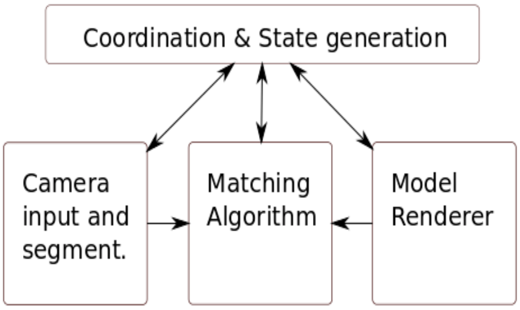


Illustration 1: The global workflow

The algorithm consists of a computer vision segmentation algorithm, a model renderer, a matching function and a coordinator program. The **Coordinator** program implements a state machine that coordinates the other parts. The applied work-flow is shown in illustration 1. For each captured frame the coordination will create candidate sets of parameters and poses of the human. These will be rendered by the **Model Renderer** and compared to the input camera image, which is first fed to a **Segmentation Algorithm**, with a **Matching Algorithm**. This algorithm will give weights to the different candidate sets of parameters and poses, hereby allows the Model Renderer the coordinator to choose the most appropriate candidate set, resulting in the most likely avatar.

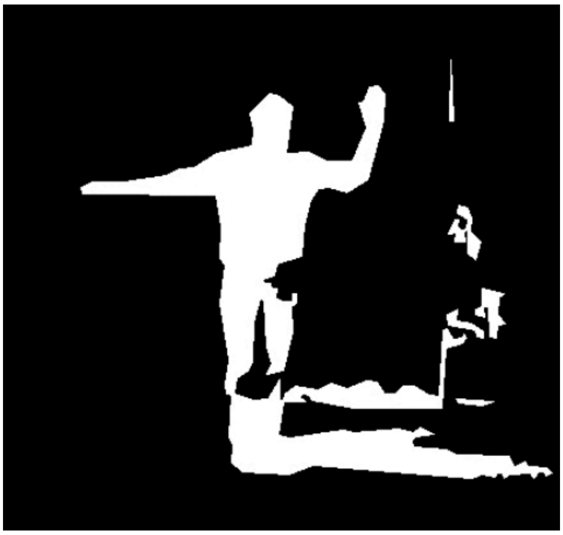


Illustration 2: The shadow problem

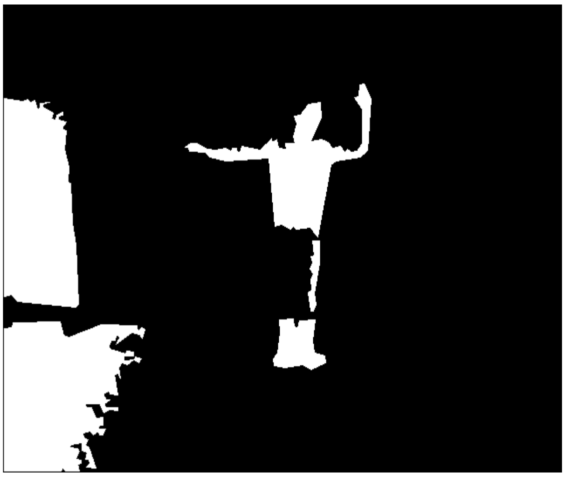


Illustration 3: The over-segmentation problem and background noise

The first, **Camera input and Segmentation**, part segments the human out of a camera image: pixels originating from the human are colored white against a black background, resulting into a binary image (Illustration 2 and 3). The background subtraction algorithm used is based on a multiple learned background model allowing to cope with illumination changes. After the learning step a simple thresholded delta subtraction (Piccardi 2004) proved to be the most performance/time efficient.

This binary segmentation however still requires additional aid to overcome the shadow problems and the over-segmenting and background noise (Illustration 2 and 3). To improve the segmentation, the camera image, together with the binary segmentation, are fed into the GrabCut algorithm (Rother et al. 2004). The binary image gives a probabilistic prior to the foreground input of the GrabCut algorithm. The background output of the learning background method gives a probabilistic input to the background input of the GrabCut algorithm. The output of the GrabCut algorithm is an improved segmentation of the camera input image which is suitable for comparison with the model input.

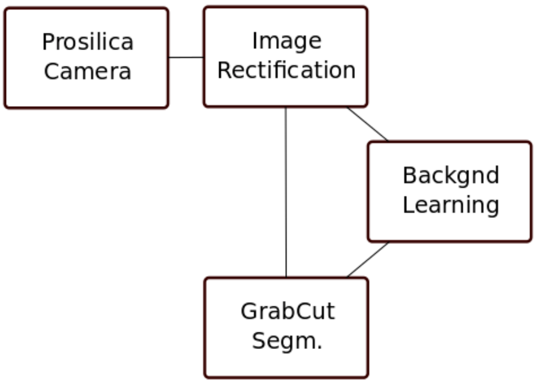
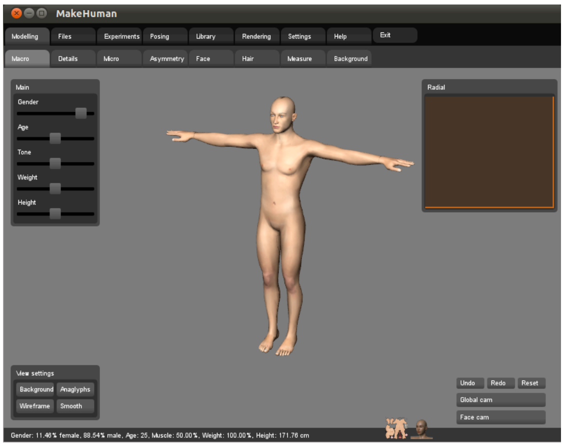


Illustration 4: The segmentation blockdiagram

This multiple stage segmentation diagram (Illustration 4) allows for combinations of multiple segmentation types like hand and face detection, skin segmentation, … allowing to combine the best of the different segmentation approaches.



Gender: 11.46% female, 88.54% male, Age: 25, Muscle: 50.00%, Weight: 100.00%, Height: 171.76 cm

Illustration 5: The MakeHuman Graphical User Interface

The second, M**odel Renderer**, part of the algorithm is a plug-in on the MakeHuman project (Bastioni and Flerackers, 2007). The MakeHuman project has its origin in the animation community where it was started to facilitate the generation of animation characters. It consists out of a rigged human mesh that can be morphed with input parameters like age, gender, muscle tone, ...

Our extension to MakeHuman consists of a MakeHuman core extension in C and a plug-in addition in Python. The plug-in allows the user to configure all MakeHuman parameters (gender, age, length, …) and export this as a picture and an **M**ake**H**uman e**X**change format file (MHX)(Illustration 5). The main advantage of MakeHuman is that the different configuration parameters underlying the human avatar are coupled: for instance of the estimated age is changed, the other parameters such as the length of the avatar will automatically change accordingly. This coupling allows the user to minimize the number of parameters needed for each specific application in order to decrease the calculation time.

In the last, **Matching Algorithm**, step of the algorithm, a camera image is simulated from the estimated MakeHuman avatar. The obtained image is fed in the background subtraction method and this output is, in a last process, compared to the real camera segmented image of the human in order to obtain a goodness of fit. This comparison is based on moving a small three-by-three pixel Gaussian kernel over the input image in order to compare it the simulated MakeHuman avatar image (Buys et al. 2010).

The **Coordination and State Generation Algorithm** coordinates the complete calculation. This program chooses the model state and sends this to the renderer (MakeHuman). The range of motion is based on active motion (Winter 1990) while average values for the body segment lengths (Drillis and Contini 1966) were taken as a prior. This program generates the configuration state in a 'brute force calculation' way (it iterates over all possible parameter configurations) and requests the result of the image comparison to take the end decision.

After the model creation the exported MHX file can be used for tracking in Blender (Illustration 6) (Buys et al. 2010, Roosendaal 2011) or exported to other simulation and visualization frameworks by the Collada specifications (Barnes and Finch 2008).

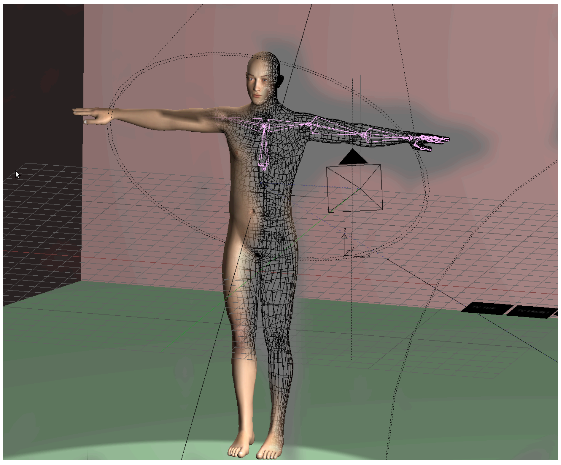


Illustration 6: After creation the avatar is imported in Blender

2.2. Multiple-view voxel-based 3D perception and avatar rigging

The goal of the second implementation is to estimate body weight and to be able to automatically adjust the found character rigging of Section 2.1 to more appropriate joint locations for the skeletal model underlying the avatar. The second algorithm is based on a multiple view approach. Furthermore, it can be seen as an optional addition to the first algorithm (Section 2.1) when multiple cameras are available.

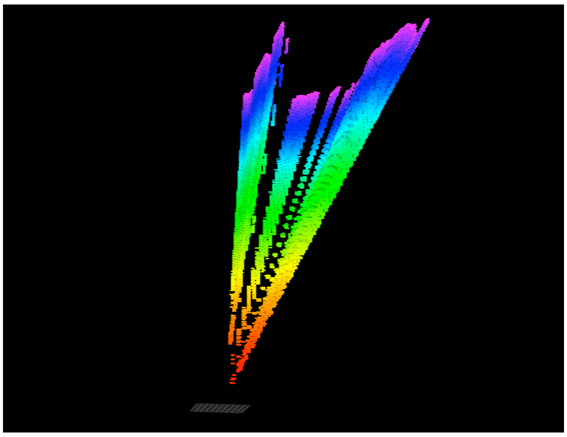


Illustration 7: Space carving example for one camera. The cones created by image pixels are simplified by lines.

This implementation is based on the visual hull (Laurentini 1994) or space carving technique. It uses a calibrated multi-camera setup to recreate the visual hull of the person. This is achieved by segmenting the person out with a binary mask that is created as explained in Section 2.1. This mask covers the original rectified camera image. For each of the remaining pixels a process similar to ray casting (Appel 1968) is executed. This is done by combining the pixel coordinates with the combined camera calibration matrix (intrinsic and extrinsic) which gives a function for a 3D line in space (that from a camera point of view can be thought of as a cone but we simplify this to limit calculation effort (Illustration 7)). This line is discretized in the positive Z-camera axis and each of these points are back projected with the respective camera calibration matrix into the other segmented camera images.

This process is repeated for each of the segmented camera images. For each of these points a probability (that the point is belonging to the visual hull of the person) is calculated based on the number of successful back projections. This proves to be useful in cases where the background segmentation process was unable to completely filter out a cluttered background.

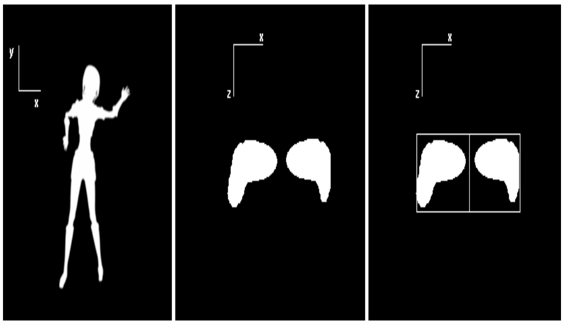


Illustration 8: Example of a visual hull sliced in the horizontal (transverse) plane

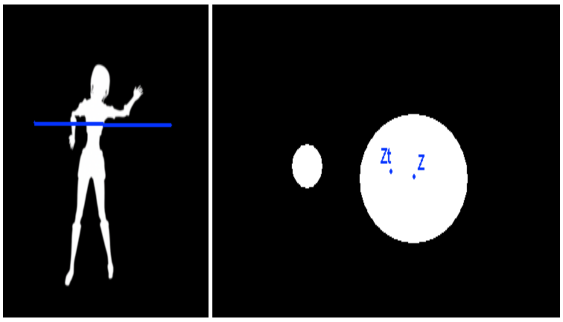


Illustration 9: Centroid calculations

From the found probabilities, a threshold is calculated in each iteration to take a minimum number of found voxels into account (this leads to the assumption that there is always a person in the image). This group of voxels form the visual convex hull that is fitting over the person. The accuracy is related to the number of cameras, the resolution in respect to the field of view, and the discretization steps taken in the back projection phase. Our experiments show that the contribution to the accuracy when adding an extra camera to a setup using 5 or more camera's is limited and is only interesting if the field of view needs to be expanded.

Once the visual hull is recreated, it is sliced in both horizontal and vertical directions (sagittal and transverse plane). On each of these slices a blob detection algorithm segments the binary slice into multiple parts (Illustration 8). Each of these contours are considered both separately and together to give both a local centroid and a uniform centroid for the slice (Illustration 9). This allows the algorithm to evaluate the local centroid distances (Z) in correspondence to the uniform centroid (Zt) and take decisions on possible 'voxel-noise'. A learning approach for this is considered as possible future work.

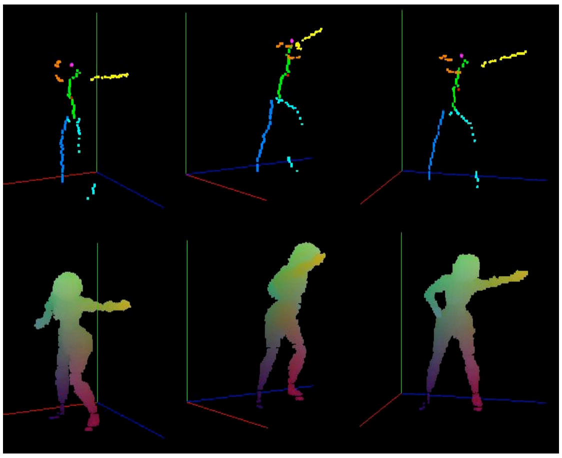


Illustration 10: Line fitting through the centroids

As a last step in the process the found centroids are imported in a line fitting algorithm, similar to curve skeleton analysis method (Lovato et al. 2009) where the intersection points of the lines prove to be good candidates for joint locations (Illustration 10).

The global centroid of the visual hull is a good indication for the center of gravity. Moreover the number of found voxels can be used to obtain a good weight estimate (provided that a weighted offset caused by the convex visual hull shape is subtracted).

The current implementation does not include tracking, it only performs frame by frame calculation. Since tracking can drastically reduce the search space of possible human poses and parameters by using the information from the previous frame, the extension of the algorithm to tracking will be handled in future work.

3. Experiments and Results

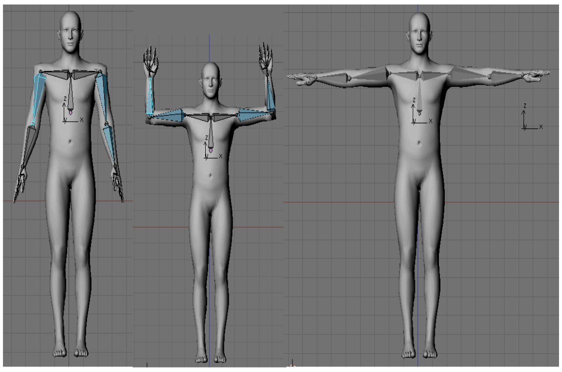


Illustration 11: Three of the requested body poses

During the experiments the subjects were asked to stand in five known poses with the 'neutral body position' as the starting pose (Illustration 11). These poses where recorded using five synchronized cameras for visual input, two Hokuyo laser scanners, and a Microsoft Kinect for point cloud generation. Afterwards the body measurements of the sagittal and coronal contour as well as weight were taken, combined with a high resolution 3D scan of the torso.

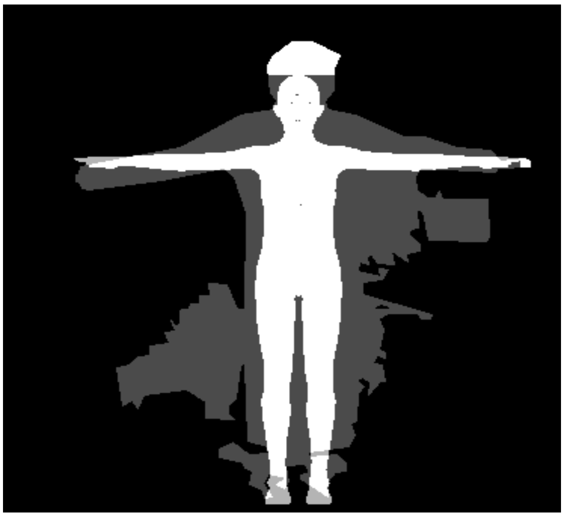


Illustration 12: MakeHuman pose matching with only arm length variation

The different parts of the algorithms are implemented in a component based framework using the ROS (Quigley et al. 2009, Willow Garage 2009) and Orocos frameworks (Soetens 2006) and interconnected using the ROS middle-ware. The different processes make use of the OpenCV (OpenCV 2011) and OpenGL libraries (OpenGL 2011) as additions to the MakeHuman plug-in.

*Corresponding author. Email: koen.buys@mech.kuleuven.be

3.1 Avatar creation experiments based on MakeHuman mesh

For this experiment the priors for the joints were taken from the five known body poses and only a slight variation was allowed. Furthermore the known gender, age, and geographical origin where fed into the prior state of the model.

In order to evaluate a single pose/parameters set for nine parameters, the algorithm needs about 27 seconds. The calculation time furthermore grows exponentially with the number of parameters, making the algorithm too computationally intensive for on-line (for instance robotic) applications for more than about seven parameters. For most applications however, only 4 parameters (Van Deun et al. 2011) were needed to successfully create a mesh model that provides sufficient accuracy for applications such as human robot interaction. Estimating these four parameters can be done on-line using a currently single threaded program.

3.2 Avatar rigging estimation based on visual hull reconstruction

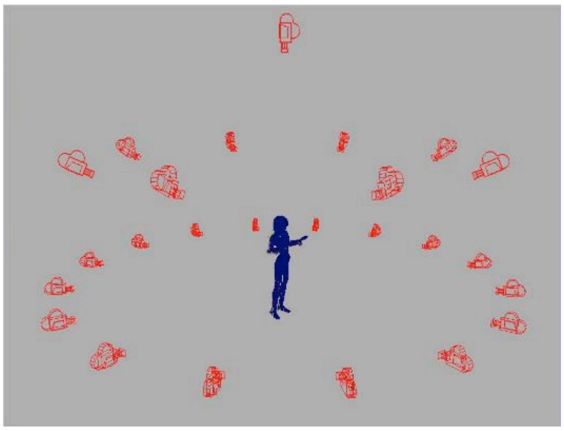


Illustration 13: The Kung-Fu dataset camera setup

The initial tests for this part are based on the "Kung-Fu Girl" dataset provided by the GrOVis research group (Graphics-Optics-Vision 2011). This dataset provides a virtual setup with some movements not common in the daily life. Together with the known ground truth they provide a good evaluation set. It has 24 cameras, placed circularly around a dancing model (Illustration 13), with known intrinsic and extrinsic calibration matrices.

Underlying the proposed algorithm are a couple of assumption that we will state explicitly. The algorithms assumes that all camera calibration matrices represent a simple pinhole camera model and second order distortion model. Furthermore, an upright position of the human is assumed, in order to be able to assign the body parts correctly to the human mesh. The person also needs to be fully visible in all camera images.

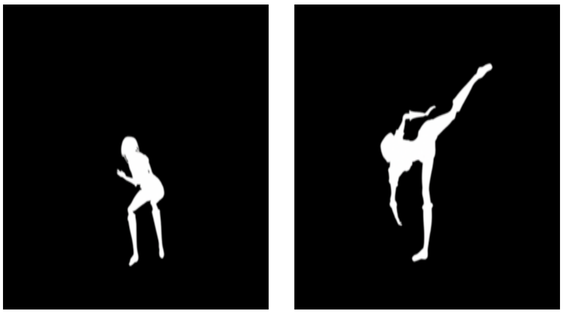


Illustration 14: Two difficult poses from the Kung-Fu dataset

Based on the results of the algorithm on the above dataset and the recorded datasets of real people in five poses we conclude that the algorithm functions fine as long as all assumptions are met (like legs are kept lower than arms). Illustration 14 shows a pose not meeting the assumptions, where the leg is higher than both arms. Although segmentation still takes place correctly, the body parts are not assigned correctly (the lifted leg is thought to be an arm).

The first results with the limited dataset showed that a 5-10% accurate body weight estimate could be accomplished using this method. For the joint locations further work with external verification needs to happen (based on measured markers).

4. Discussion

Both algorithms presented offer centimeter accuracy with on-line calculation as a main advantage. On-line calculation was set as the main priority with human robot interaction as an application.

As illustrated in our accompanying paper (Van Deun et al. 2011) it sufficed to take three or four good fit parameters to have a well fitting model for biomechanical purposes. The advantage of MakeHuman is that a prior for the model can easily be generated without doing body measurements, as a physician normally already knows parameters like age, weight, gender, origin, … thus limiting the need for extra body measurements and additional calculations.

5. Conclusion and Future Work

The paper illustrates two camera-based algorithms to automatically generate human avatars (skeletal model and human mesh) that are adequate for medical and robot-assistance purposes and avoid taking extra body measurements (Van Deun et al. 2011). It shows that the MakeHuman project is suitable to be adopted for biomechanical measurements.

Both presented algorithms can be calculated in an on-line manner and are therefore suited for implementations in robot perception algorithms, hereby contributing to for instance the application of human-robot interaction.

Future work includes avoiding the brute force calculation approach that our current implementation includes. This can be done by implementing an estimation algorithm (Extended Kalman filter, particle filter) (Kalman 1960, Gordon et al. 1993) if tracking on video feeds is required.

Future work will furthermore implement an avatar adaptation algorithm that can be executed also during tracking and can take multiple segmentation hints as well as learning approaches to minimize the background noise influences.

Acknowledgements

This study was carried out thanks to the financial support of K.U.Leuven's Concerted Research Action GOA/2010/011 real-time optimal control of autonomous robots and mechatronic systems, Tinne De Laet is a PostDoctoral Fellow of the Research Foundation - Flanders (FWO) in Belgium. Dorien Van Deun's research is funded by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT - Vlaanderen).

Further more the authors would like to acknowledge the master thesis work of Niels Van Malderen and Thijs Parmentier and the work performed during the internship of Jeroen Lecoutere.

References

Ahmed N. , de Aguiar E., Theobalt C., Magnor M. , and H.-P. Seidel. 2005. Automatic generation of personalized human avatars from multi-view video. In Proceedings of the ACM symposium on Virtual reality software and technology (VRST '05). ACM, New York, NY, USA, 257-260.

Appel A., 1968, Some techniques for machine rendering of solids. AFIPS Conference Proc. 32 pp.37-45

Barnes M. and Finch E. L., 2008, COLLADA—Digital Asset Schema Release 1.5.0. http://www.collada.org

Bastioni M. and Flerackers M., 2007 (0.9.1 Release Candidate), the MakeHuman project, http://sites.google.com/site/makehumandocs/.

Buys K., De Laet T., Smits R. and Bruyninckx H., 2010, Blender for Robotics: integration into the Leuven paradigm for robot specification and human motion estimation. In Proceedings of Simulation, Modeling and Programming for Autonomous Robots (SIMPAR) Conference, Darmstadt

R. Drillis,R. Contini, 1966, segment parameters. Technical Report 1166.03, New York University School of Engineering and Science, New York

Gordon, N. J.; Salmond, D. J. and Smith, A. F. M. (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". IEE Proceedings on Radar and Signal Processing **140** (2): 107–113.

Graphics-Optics-Vision, Max-Planck-Institut für Informatik, 2011, "Kung-Fu Girl" A synthetic test sequence for multi-view reconstruction and rendering research, http://www.mpi-inf.mpg.de/departments/irg3/kungfu/

Kalman, R. E., 1960, “A new approach to linear filtering and prediction problems”, Journal of Basic Engineering, vol. 8, pp. 35-45

A. Laurentini, 1994, "The Visual Hull Concept for Silhouette-Based Image Understanding," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 150-162

C. Lovato, U. Castellani, and A. Giachetti, 2009: “Automatic segmentation of scanned human body using curve skeleton analysis”, *Proceedings of the 4th International Conference on Computer Vision / Computer Graphics Collaboration Techniques*

C. Rother, V. Kolmogorov, and A. Blake, 2004, : Interactive foreground extraction using iterated graph cuts, ACM Trans. Graph., vol. 23, pp. 309–314

OpenCV, 2011, The Open Computer Vision library, http://opencv.willowgarage.com/wiki/

OpenGL, 2011, http://www.opengl.org/

M. Piccardi, 2004 , "Background subtraction techniques: a review," Systems, Man and Cybernetics, 2004 IEEE International Conference on , vol.4, no., pp. 3099- 3104 vol.4, 10-13

R. Poppe, 2007, Vision-based human motion analysis: An overview, Computer Vision and Image Understanding, Volume 108, Issues 1-2, Special Issue on Vision for Human-Computer Interaction, Pages 4-18

M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, 2009, ROS: an open-source Robot Operating System. In ICRA Workshop on Open Source Software

Roosendaal T., 2011, Blender: a free and open source tool for 3D graphics application, available on http://www.blender.org

Soetens P. , 2006, A Software Framework for Real-Time and Distributed Robot and Machine Control. PhD thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.

Villa-Uriol MC., M. Sainz, F. Kuester, N. Bagherzadeh, 2003, Automatic creation of three-dimensional avatars, Proceedings of the SPIE, Volume 5013, pp. 14-25.

Van Deun et al., 2011, Automatic Generation of Personalized Human Models Based on Body Measurements, internal report, accepted for publication in Proceedings of First International Symposium on Digital Human Modeling, Lyon

Winter, D.A., 1990, and Motor Control of Human Movement. Wiley Interscience, New York

Willow Garage, 2009 Robot Operating System (ROS). http://www.ros.org.