

Parametric Modeling of the Human Body Using Kinect Measurements

Jorn WIJCKMANS*^a, Dorien VAN DEUN^a, Koen BUYS^b,
Jos VANDER SLOTEN^a, Herman BRUYNINCKX^b

^a Dept. of Mechanical Engineering, Biomechanics Section, Katholieke Universiteit Leuven, Belgium;

^b Dept. of Mechanical Engineering, Division PMA, Katholieke Universiteit Leuven, Belgium

<http://dx.doi.org/10.15221/13.117>

Abstract

Personalized digital human modeling is useful for a wide variety of applications. An obvious interest comes from the entertainment industry, where movies and video games explore the possibilities of this technology. In biomechanics, human modeling assists in the design of person-specific solutions to improve the human well-being and ergonomics. Other applications exist in virtual dressing rooms, human-robot interactions in robotics, etc. Existing technology, although performing well, has the disadvantage of being expensive, immobile, not fully customizable and possibly requiring external body markers. To tackle these issues, this paper presents a modeling technique using the open source software MakeHuman, based on body measurements obtained with Microsoft's Kinect. The current solution is able to retrieve these measurements when the person is standing in a calibrated scene, this means when the person's position is known a priori. In order to retrieve the measurement data as a point cloud, and to process this point cloud, the PCL (Point Cloud Library) software is used, leading to a fully open source implementation. With these tools, solutions for person segmentation, measuring and personalized modeling are proposed. It appears that the current Kinect technology on itself is not very accurate for measuring body sizes. However, this work shows that the Kinect information combined with the MakeHuman modeling tool is valuable. The final model incorporates measures like body height, arm span, hip, waist and chest width, completed with information such as age, gender and weight. Evaluation of the resulting human model shows moderate to good results in modeling body height, hip and waist width, whereas chest width modeling is rather poor due to difficulties in chest width extraction from Kinect images..

Keywords: Personalized digital human modeling, Kinect, MakeHuman, PCL.

1. Introduction

Personalized digital human models [1, 2, 3] are an interesting tool for a broad spectrum of applications. Originally, drawn human models were used as a tool to assist product and workplace design. This was a cumbersome and time-consuming way of working. Since the availability of CAD tools, the models were digitalized, allowing faster and cheaper response to design problems. From then on, different industries started to see the potential of these models [1]. Modern applications are found in product and workplace design [4], the entertainment industry [5], clothing [6], biomechanics research, robotics and many more.

In order to make such models more accessible to all kinds of users, this paper suggests and investigates the feasibility of an alternative methodology. The urge for this alternative approach originates from existing constraints linked to the current techniques. For example, state of the art methods are based on anthropometric databases, 3D scanners [7] or motion capture technology [8] to build customized human models. These methods require expensive infrastructure, hence are quite immobile and can even require the use of external body markers. Additionally, current modeling tools are provided as commercial software packages [1, 9, 10, 11, 12], which obstruct application-specific development by the user himself. To put these disadvantages to the past, the proposed methodology makes use of Microsoft's Kinect [13, 14] to acquire body measurements. This low-cost and mobile device yields point clouds with both depth and color information (rgb). A point cloud is a set of n-dimensional vectors. In this case, the vector elements are (x,y,z,r,g,b), the first three being the 3D coordinates and the last three being the rgb values.

Before being able to calculate body dimensions, the point cloud has to be filtered in order to segment the person from his environment. This is done through a process called segmentation. Segmentation is done by using a series of pass-through filters and a ground removal method based on point cloud width. Figure 1 shows how this step fits into the global solution provided by this work. Processing of the point clouds is performed by making use of the open source software of the Point cloud library (PCL) [16, 17].

* Jorn Wijckmans. Email: jornwijckmans@gmail.com

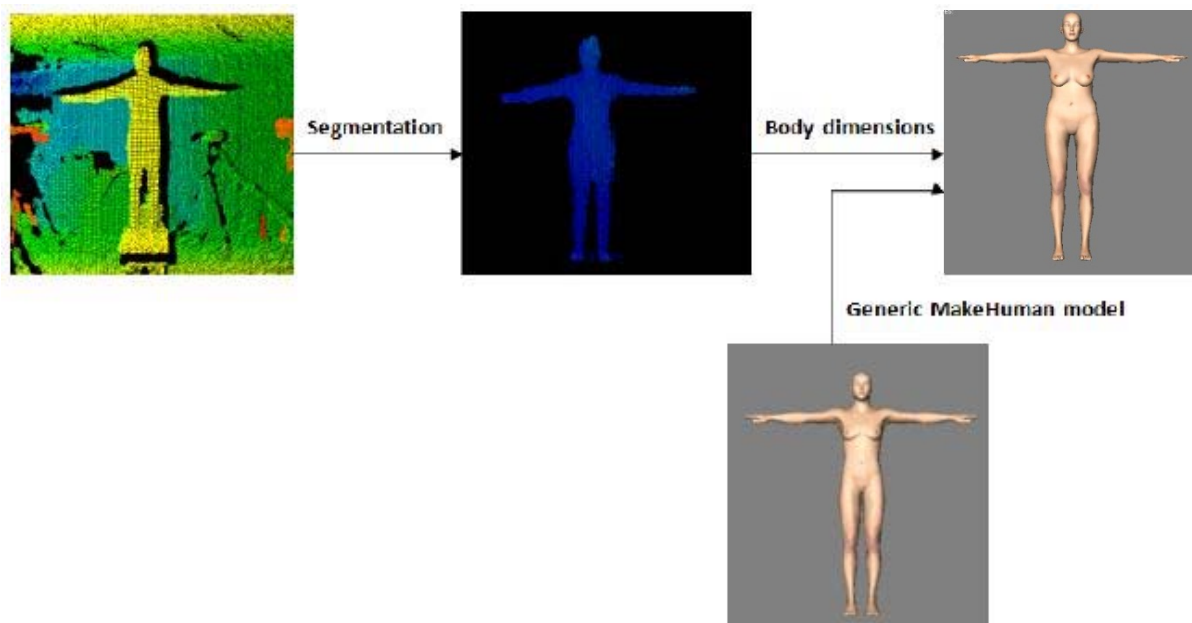


Fig. 1. Overview of the different steps implemented in the global solution.

After correct segmentation of the person, an extraction algorithm is developed and applied to obtain body measures from the person. This algorithm focuses on retrieving the following body dimensions: body height, arm span, hip, waist and chest width, completed with parameters as age, gender and weight. Those last three parameters are obtained through user input. Following Van Deun et al. [18] this number of body parameters can suffice for decent model characterization.

Once all Kinect-based dimensions are measured, they are implemented into a model, provided by the open source human modeling tool MakeHuman [19] (see figure 1). A generic mesh of a human being is customized - based on a person's Kinect measurements - through a morphing process [20]. MakeHuman incorporates correlations between anthropometric measures in its morphing targets. These correlations help MakeHuman to determine the boundaries of realistic human appearance. This makes MakeHuman a powerful tool for human modeling. Of course it is only safe to rely on these correlations when another base parameter can be modeled sufficiently accurate. In this case, BMI is used as base parameter, hence weight modeling is crucial in the suggested solution.

Finally, this paper investigates the modeling results. Vialux data [15] and manual measurements, obtained together with the Kinect data, are used for this purpose.

2. Related work

Since the Kinect was not launched until the end of 2010, its contributions to research were still limited at the time this work started. However, the Kinect's interesting properties have drawn much attention to it from research groups, realizing it is more than just a gaming controller. This explains the rapid increase of academic publications making use of the Kinect.

As stated, at the time the research presented in this paper started, almost no scientific literature on anthropometric measuring with the Kinect existed. However, at that same time, some similar projects started. Aitpayev et al. [21] for example did research on the creation of 3D human avatars based on Kinect measurements. The way their work differs from ours is that they use the Kinect output and make a model out of the point cloud immediately. The model's face represents the face of the person being modeled. Our model, on the other hand, calculates body parameters from the Kinect point cloud and implements them in a generic model. Another research project, by Weiss et al. [22], presented in November 2011, looks at 3D scans in a home environment from noisy Kinect data. Their approach combines low-resolution image silhouettes with coarse range data to estimate a parametric model of the body. Combining multiple monocular views of a person in front of the Kinect then enables them to accurately estimate the 3D shape of the person.

3. Methodology

The introductory section already described the general workflow. The Kinect yields point cloud data, from which a person needs to be segmented. The next step is to extract body dimensions from it and implement them in a MakeHuman model. The different steps and the outcome of each step are visualized in figure 1. Each of the following sections elaborates on these steps.

3.1. Person segmentation

The Kinect yields raw point cloud data, i.e. a point cloud containing both the person – standing in a Vitruvian man-like pose - and his environment. This environment is of no interest and hence needs to be removed, a process called segmentation. The segmentation algorithm proposed in this paper is suited for a calibrated scene, meaning that it is capable of separating the person from his environment if his position in the scene is known a priori. The segmentation algorithm consists of three consecutive filters, completed with a ground removal algorithm. The person is, with the exception of the ground, completely unconnected to his environment. Therefore, making use of the points' 3D coordinates, simple pass-through filters suffice for removal of all environmental points lying behind, next to and above the person. For depth and side filters, scene-specific filter thresholds have to be set, i.e. ranges in which points are to be withheld. Filtering the upper part of the point cloud is easier, as we assume only the ceiling of the room is present here. A fixed value of 2.5 centimeters for example is enough as a threshold for ceiling subtraction. This means that only points lying more than 2.5 centimeters below the highest point in the scene, are withheld in this segmentation step.

The final part of the segmentation comprises the separation between the person and the ground. This is a lot harder to do because one needs to find the exact height at which the feet of the person touch the ground. The principle developed to do this (visualized in figure 2) looks at the body width by iterating over the body in a vertical direction in steps of one millimeter. For each iteration height, the number of points lying only 1mm higher or lower than the iteration height is counted and saved into a vector. This represents the cloud width at each height. The reason why we look in this window of 2 mm is because the coordinates of the points in the cloud are not necessarily an integer. When a sudden change in point cloud width, indicated by the number of points with a vertical coordinate (y-axis) deviating maximum one millimeter from the iteration height, is observed, the transition between the feet and the ground is found.

Applying this series of steps shows that the segmentation algorithm has a success rate of 100 percent once it is appropriately tuned. For completeness, figure 3 shows the segmentation algorithm in pseudo code.

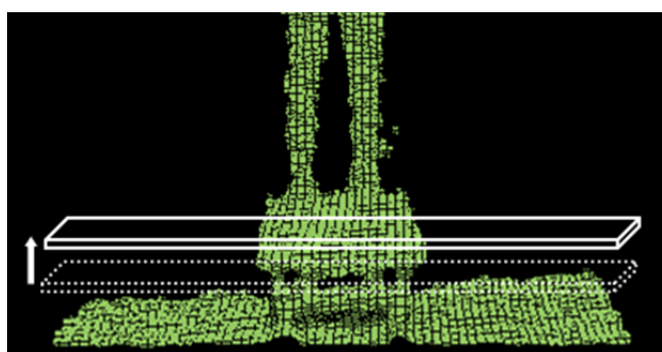


Fig. 2. Principle to segment the person from the floor or an object he is standing on.

```

DEPTH FILTER
for all points in cloud do
  1) search for point closest to Kinect (= point with smallest z coordinate).
end for
2) pass-through filter: withhold all points for (closest point < z < closest point +
user value)
SIDES FILTER
3) pass-through filter: withhold all points for (user value 1 < x < user value 2)
CEILING FILTER
4) pass-through filter: withhold all points for y > (minimum y value + 2.5 cm)
GROUND FILTER
for i = 0; i < 3000; ++i do
  5) iterationHeight = 0 + 0.001*i
  6) search all points lying at most 1 mm from the iterationHeight, count them
  and put the number in vector PointCount.
end for
for all indices k in PointCount do
  if PointCount[k] > threshold and PointCount[k+1] > threshold then
    7) groundlevel = 0 + 0.001*(k-1) (corresponding to PointCount[k-1])
  end if
end for

```

Fig. 3. Segmentation algorithm in pseudo code.

3.2. Measurement extraction

From a segmented point cloud, body measures can be calculated. Body height and arm span are quite straightforward to measure, as they are the largest distances measured in the vertical (y) and horizontal (x) direction, respectively. By making use of their 3D coordinates, these extreme points are not hard to find. Height and arm span are then simply calculated as the difference in y, respectively x value between these extreme points.

Measuring hip, waist and chest width requires a method which is less obvious. From anatomical knowledge, it is known that the hip region forms the broadest part of the trunk, at least, when the person is not too heavy. Therefore, similar to the ground removal principle, the body width is calculated by iterating in steps of one millimeter in the y direction. At each iteration height, the largest x distance between points lying at most one millimeter higher or lower than the iteration height is sought. This distance is then defined as the trunk width at that height. However, application of this principle would yield the arm span, as this is the broadest width of the total body (see figure 1). Therefore, a starting point close to the hip is searched to start the iterations. This is done by making use of an algorithm designed by Buys et al. [23]. They are able to segment a person from a non-calibrated scene, and assign a body part label to each pixel. We use this body part information to find a characteristic hip point. As mentioned, this characteristic point is the starting iteration height, and the extraction algorithm searches for the broadest trunk width, lying not more than ten centimeters higher or lower than the starting point. Results show that this principle is very robust in determining the hip width.

The waist is known to be the smallest part of the trunk. Therefore one could use the same principle as before for the hip width. Again, there still are smaller widths over the entire body, like the neck for example. It is possible to define a characteristic starting point like before, however, this would be less efficient, as a starting point is already known actually (the hip), and as we know the waist is lying above the hip. For women the waist is often easy to determine, for men this is a little harder. To solve this problem, the smallest trunk width is searched, lying maximum thirty centimeters above the hip. This methodology is quite robust. It only fails to find the correct waist width when there is a small abnormality in the point cloud between the hip and the waist. Figure 4 shows such an example.

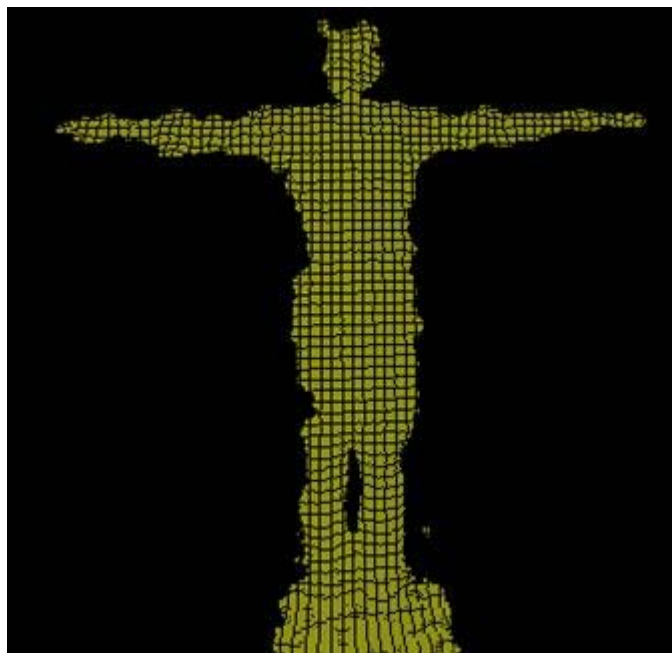


Fig. 4. Poor waist width retrieval due to an incomplete point cloud

The chest width, finally, is measured by looking downwards, starting from the arm pits. The arm pits are defined as the height where there is a sudden decrease in body width (same principle as before), corresponding to the transition between stretched arms and the trunk. Chest width is measured at the level of the nipple. This height is assumed to be lying approximately ten centimeters below the arm pits. This, however, is a rough estimation of the correct measuring height. Chest width is then measured as the trunk width at this particular height. More accurate chest results can be obtained by using a more correct routine to find the chest height. Winter et al. [24] present several body ratios, amongst which a body to chest height ratio. This particular ratio equals 72 percent. This approach is more accurate in defining chest height than the one currently used (as a fixed distance underneath the arm pits). This improvement will be implemented in future work.

Again, more details about the algorithm performing the described tasks are given in pseudo code (see figure 5).

Both algorithms are combined into one semi-automatic program. This program needs user input for setting the segmentation filter thresholds, and for person-specific information like gender, age and weight. With this information, the program executes everything starting from a raw point cloud, ending up with a list of all the aforementioned body sizes. To check the correctness of the calculated body sizes, the program also gives an image of which points in the cloud are actually used for measuring. Figure 6 gives an example of this additional output.

```

1) determine characteristic points with body labeling algorithm.
2) load file with characteristic points, here average y coordinate hip.
3) ask for user's age, gender and weight.
ARM SPAN
for all points in cloud do
  4) search for extremities in the x direction.
end for
5) put the 2 points into cloud destined for supervision
6) arm span = left extremity - right extremity
BODY LENGTH
for all points in cloud do
  7) search for extremities in the y direction.
end for
8) put the 2 points into cloud destined for supervision.
9) body length = lower extremity - upper extremity (because of orientation y axis).
```



```

TRUNK WIDTH
for j = 0; j<2000; ++j do
    11) iterationHeight = minimal y value + j * 0.001
    for all points in cloud do
        12) find left and right extremities
    end for
    13) trunkWidth = left extremity - right extremity
    14) put trunkWidth in vector widthVector
    15) put iterationHeight in vector 'height'
end for
HIP WIDTH
for all k = 0;k<2000; ++k do
    if height[k] - y value characteristic point < 0.0015 then
        for l = -100; l<101; ++l do
            if widthVector[k+l] > max hip width then
                16) hip width = widthVector[k+l]
            end if
        end for
    end if
end for
WAIST WIDTH
for m = hip height; m > hip height - 300; -m do
    if widthVector[m] < min waist width then
        17) waist width = widthVector[m]
    end if
end for
CHEST WIDTH
for all points in cloud do
    18) find maximal width = arm span
end for
for n = height arms; n < height arms + 100 ;++n do
    if widthVector[n] - widthVector[n+1] > 0.15 then
        19) chest width = widthVector[n+100]
    end if
end for
20) put indices points for hip, waist and chest in additional cloud for supervision.
21) BMI = weight / (body length^2)
22) write body dimensions to file.

```

Fig. 5. Extraction algorithm in pseudo code.

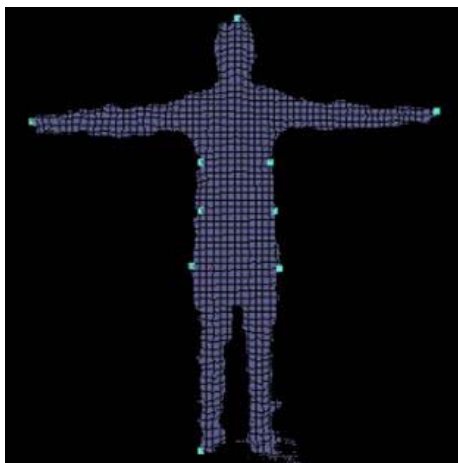


Fig. 6. Points used for measurement extraction.

3.3. Measurement implementation

After determining the body measures, they are implemented into a MakeHuman model through a newly developed plug-in. The MakeHuman interface uses morphing techniques to adapt the model, in which morphing targets are defined for several regions and alterations of the model [20]. These targets define which vertices of the model are addressed and how much and in which direction they are displaced. Each target is associated with a value between zero and one, determining the degree of morphing. For example, two targets, 'giant' and 'dwarf' are available for modeling of the body height. The tallest and smallest person achievable correspond to the target values of one and zero respectively.

First of all, parameters like age and gender are modeled. Gender for example is zero for women, one for men. Next, MakeHuman models between the age of twelve and seventy, corresponding to the zero and one values. Every age value in between can be found by linear interpolation.

Weight appears to be a very important parameter. MakeHuman namely has correlations between anthropometric measures integrated into its model. For example, a heavy person's maximal hip width will be larger than a skinny person's. When evaluating the results of our technique, it appears to be extremely important to make use of these correlations, as the Kinect overestimates hip, waist and chest width most of the time. Weight is the parameter against which these widths are correlated. The weight is modeled by tuning both the parameters 'weight' and 'muscle tone' in MakeHuman. The zero to one values of the parameter weight are set to BMI values of 16.5 and 40, corresponding to under- and overweight, and a value of 0.5 corresponds to a BMI of 20.75 in our implementation. In between the ranges 0 - 0.5 and 0.5 - 1, BMI is linearly distributed over the weight values. Figure 7 visualizes this distribution. Furthermore, as the parameter weight exceeds the 0.5 value, muscle tone is decreased at the same rate, starting from a 0.5 value. This appears to be an accurate model of body weight.

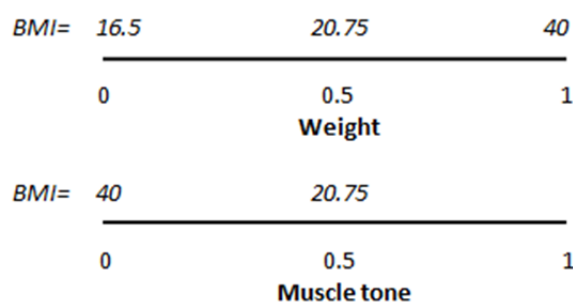


Fig. 7. BMI distribution for parameters 'weight' and 'muscle tone'.

When these first parameters are set, the modeling algorithm performs morphing on body height, hip, waist and chest width in an iterative way. As long as the difference between the MakeHuman dimensions and the ones dictated by the Kinect are not closer to each other than 0.25 cm, the target is further morphed. During each iteration, the target is morphed with 50% of the difference between the MakeHuman dimensions and the ones found with the Kinect. This way, the morphing steps become smaller when the MakeHuman dimensions approximate the Kinect's. This is summarized in figure 8, showing this algorithm.

As stated earlier, the Kinect mostly overrates hip, waist and chest width by a few centimeters. However, because of the body correlations incorporated in the MakeHuman target files, the morphing stops when the boundaries of realistic anatomy are exceeded. This way, the global solution is able to customize the MakeHuman model so it resembles relatively close the person standing in front of the Kinect with regards to the modeled body parameters. A quantitative evaluation is given in the next section.

```

1) load the target files for hip, waist and chest morphing
2) measure the dimension in the current MakeHuman model
while delta = (MakeHuman value - Kinect value) > threshold do
  3) change = x percent of delta
  4) depending on the sign of delta, in- or decrease the MakeHuman model's
  dimension with value 'change', using the appropriate target file
  5) recalculate the dimension in the adapted MakeHuman model and delta
  if delta did not change then
    6) stop morphing
  end if
end while

```

Fig. 8. Morphing algorithm to adapt the generic mesh to the desired body sizes

4. Results and Discussion

The quality of the MakeHuman model is evaluated by comparing its body height, hip, waist and chest width to reference values for these parameters. Table 1 summarizes both measuring and modeling errors for those body parameters.

Table 1. Evaluation of measuring and modeling errors.

	Measuring error	Modeling error
Body height	Less than 3 cm (after error analysis) - underrated	Less than 3 cm (after error analysis) - underrated
Hip width	[2,00 cm ; 5,89 cm] (overrated)	error < 1,2 cm 66 % : error < 0,75 cm
Waist width	[-2,67 cm ; 5,91 cm]	[-2,41 cm ; 3,57 cm] 50% : error < 1 cm 72,5% : error < 2 cm
Chest width	[-4,92 cm ; 2,31 cm] 50% : [- 2 cm ; 1,69 cm]	[0,43 cm ; 8,53 cm] (underrated)

Body height is evaluated at an early level (before implementation) because modeling accuracy is not a problem at all, whereas measuring accuracy is the main issue here. It appears that body height is always underrated. The evaluation is performed by comparing the Kinect's estimations to values measured manually. This information is used for error analysis. Because only a small data set was at hand (twelve persons), a leave one out analysis was performed as to find correlations between the real and Kinect values of body height. By correcting height with the correlation coefficients, the maximum error encountered in modeling this parameter is reduced to three centimeters.

As stated before, hip, waist and chest width are mostly overrated. MakeHuman's incorporated correlations between body sizes, however, help to keep the implemented body sizes in the final model in range of realistic values. This only holds when weight modeling has been performed well. Vialux data of the trunk, obtained simultaneously with the Kinect data during a measuring experiment with 45 persons, are used as a reference measurement (measuring accuracy of 0,86 mm) to assess modeling quality of waist and chest width. The Vialux point cloud is turned into a mesh and its waist and chest width are compared to the ones implemented in the model. Regarding waist width, it is found that 50 percent of the persons in the database were modeled within one centimeter of the real values. Furthermore, in 72.5 percent of the cases, the modeling error was two centimeters at most.

When looking at chest width, the results are inferior to the ones for the waist. Measuring errors range from -4.92 to 1.69 centimeters. Fortunately, approximately 50 percent of the persons' chest are measured within the range [-2, 1.69 cm]. Chest modeling on the other hand is less accurate ([0,43, 8,53 cm]), because of the difference between our solution and MakeHuman in defining chest width (due to imperfect measurement extraction). Another cause is the fact that there is no perfect morphing target for the chest. It is possible to use a better one than the one currently implemented, when looking only at the chest. However this alternative target shows overlap with the waist region, and therefore deteriorates the waist modeling. Because waist width is modeled quite accurately, the original chest target is preferred.

Hip width, finally, cannot be measured using Vialux data, because the limited field of view of the camera setup. Therefore manual hip measurements of six persons are used. It is clear that the outcome of this evaluation has to be interpreted with care, as the data set is too limited for a strong statement to be based on. Moreover, because the reference measurement is obtained in a less accurate way compared to the Vialux measurements, it is very dangerous to put the same confidence in these results as in the waist and chest results. From error analysis it appears that modeling mistakes remain limited to 1.2 centimeters, whereas four of the subjects' hip width was even modeled within 0.75 centimeter of the real value. This accurate modeling clearly is only possible due to MakeHuman's correlations, as hip width tends to be overestimated the most of the three trunk parameters by the Kinect.

5. Conclusions

This paper presents a first step towards a possible approach to the problem of personalizing a digital human model, in an inexpensive, mobile and customizable fashion. Results are given for the modeling of body height, hip, waist and chest width. Height is fairly accurate implemented in the model. Unfortunately this is hard to improve, as the error is caused by gaps in the point cloud. Hip width appears to be modeled accurately, however, a more thorough evaluation is required to confirm this statement. Waist results are satisfying, whereas chest width results can be improved.

Future work should focus on optimizing the current modeling routines (especially segmentation) and on integration of new body parameters in the model, e.g. back curvature and body depth at hip, waist and chest height. Another important aspect for improvement is the processing of the Kinect point cloud. Preprocessing the point cloud potentially can ameliorate the Kinect measurements. Another point of interest is the expansion of the current segmentation algorithm to non-calibrated or continuously changing scenes.

As a final conclusion, one can state that personalized human modeling is feasible, though the Kinect on its own is not capable of acquiring accurate measurements in the current implementation. Hopefully this year's follow-up version of the Kinect is showing significant improvements in accuracy. A model incorporating correlations between body dimensions, like MakeHuman does, is required in order to make sure that over- or underrated Kinect measurements are not fully implemented but only until the boundaries of realistic values are reached. To conclude, the combination Kinect – MakeHuman is quite suited for the purpose of digital human modeling, showing medium to good results for the targeted body sizes.

Acknowledgments

This work has been funded thanks to the financial support of KU Leuven's Concerted Research Action GOA/2010/011.

References

1. Peter Blanchonette, "Jack Human Modelling Tool: A Review", Australian Government, Department of Defence, Defence Science and Technology Organization, Air Operations Division, 2010.
2. Niels Moes, "Digital Human Models", in TMCE Symposium, Ancona, Italy. Delft University of Technology, Faculty of Industrial Design Engineering, 2010.
3. Hein Daanen, "Digital Human Models", in Workshop DHM, Tokyo, Japan. TNO Human Factors, 2004.
4. D. B. Chaffin, "Digital Human Modeling For Vehicle And Workplace Design", Society Of Automotive Engineers, 2001, 200 p.
5. "Microsoft Xbox", <http://www.xbox.com/nl-NL/>, 01/06/12.
6. F. Pereira, C. Silva, M. Alves, "Virtual Fitting Room Augmented Reality Techniques for e-Commerce", in Communications in Computer and Information Science 2011, 220: 62-71.
7. T. Weirich, M. Pauly, R. Keiser, S. Heinzle, S. Scandella and M. Gross, "Post-processing of scanned 3D surface data", in Proc. Symposium on Point-Based Graphics, 2004.
8. J. Bandouch, F. Engstler and M. Beetz, "Accurate Human Motion Capture Using an Ergonomics-Based Anthropometric Human Model", in Lecture Notes in Computer Science, 2008.

9. "Mannequin Pro", <http://www.nexgenergo.com> , 21/10/11.
10. "Anybody", <http://www.anybodytech.com> , 21/10/11.
11. "Madymo", <http://www.tass-safe.com> , 21/10/11.
12. "Safeworks", <http://www.3ds.com> , 21/10/11.
13. "Microsoft Kinect", <http://www.xbox.com/nl-NL/kinect> , 15/10/11.
14. Luke Plunkett, "Report: Here Are Kinect's Technical Specs", <http://kotaku.com/5576002/here-are-kinects-technical-specs> , 15/10/11.
15. "Vialux 3d scanner", <http://www.vialux.de/HTML/en-ddscan.html>, 10/03/12.
16. "The Point Cloud Library (PCL)", <http://www.pointclouds.org>, 09/11/11.
17. R. Bogdan Rusu, S. Cousins, "3D is here: Point Cloud Library (PCL)", in IEEE International Conference on Robotics and Automation (ICRA), May 2011, Shanghai, China.
18. D. Van Deun, V. Verhaert, K. Buys, B. Haex, J. Vander Sloten, "Automatic Generation of Personalized Human Models Based on Body Measurements", in International Symposium on Digital Human Modeling edition:1, Lyon, France, June 2011.
19. S. M. Bastioni, M. Flerackers, J. Capco, "MakeHuman, Open source tool for making 3D characters", <http://www.makehuman.org> , 05/03/12.
20. B. Ugur, A. Vahit Sahiner, "A Parametric Human Body Modeling Tool".
21. K. Aitpayev, J. Gaber, "Creation of 3D Human Avatar using Kinect", in Asian Transactions on Fundamentals of Electronics, Communication and Multimedia (ATFECM) (ATFECM ISSN: 2221-4305) Volume 01 Issue 05, January 2012.
22. Alexander Weiss, David Hirshberg, Michael J. Black, "Home 3D Body Scans from Noisy Image and Range Data", in 2011 IEEE International Conference on Computer Vision, November 2011.
23. K. Buys, C. Cagniard, A. Baksheev, C. Pantofaru, "An adaptable system for human body tracking", submitted for IROS conference 2012.
24. D.A. Winter, "Biomechanics and Motor Control of Human Movement". Wiley Interscience, 1990.