A Digital Human Model for Performance-Based Design

Tim Marler[1], Steve Beck[1], Uday Verma[1], Ross Johnson[1],
Victoria Roemig[1], and Behzad Dariush[2]

[1] Santos Human Inc.
Coralville, IA 52241, USA
[2] Honda Research Institute USA
Mountain View CA 94043
{tim.marler,steve.beck,victoria.roemig}@santoshumaninc.com,
{uv,ross}@mazira.com, bdariush@gmail.com

Abstract. Real-time optimization-based posture prediction has fostered the development of zone differentiation, whereby performance-measure values resulting from a predicted posture are evaluated and displayed for large volumes of target points surrounding an avatar. To date, this tool was limited with respect to computational speed and practical applications. This paper presents a series of improvements and new features, including new algorithms for sphere filling and collision avoidance, significant increases to computational speed, incorporation of whole-body posture prediction, new methods for visualizing results, and multi-dimensional zone differentiation, which is the ability to automatically calculate multiple zones for various sets of problem parameters. These new tools collectively advance human systems integration. They are successfully applied to three example problems and demonstrate the ability to direct product design virtually, based on human performance.

Keywords: Digital human modeling and simulation, zone differentiation, reach analysis, posture prediction.

1 Introduction

Often, the concepts of human systems integration (HSI) or human-centered design relate primarily to fit, access, and range-of motion, but not necessarily broader aspects of human performance. Yet, in order to design products for ease of use and safety, even when focusing on static analysis, one must consider performance. This is true whether one works with experimental protocols or with virtual simulation and analysis. As digital human models (DHMs) become more mature and more prevalent in the virtual design process, the amount and form of the analysis data they present becomes more critical. Distilling a variety of results from many different cases (i.e. millions of potential reach targets) into palatable data that can be used for practical design changes continues to be a challenge. Furthermore, although basic DHMs can be used for geometric package analysis, predictive and analytical capabilities are critical for improving designs as effectively as possible and thus for increasing safety and ease of use. In this vein, optimization-based posture prediction provides a means of not only

V.G. Duffy (Ed.): DHM 2014, LNCS 8529, pp. 136–147, 2014.

studying the interaction of virtual humans with their environment, but also studying their performance as well. Advances with this technology have been substantial over the past decade, but only recently have extensive real-world use cases for digital human models been published.

In response to this state of the art, Santos Human Inc. has developed the next generation *zone differentiation*, which allows one to conduct concurrent virtual design and analysis, and has used this new tool for improving the designs of seats in amusement park rides, hand breaks in automobiles, and seats in heavy equipment. In all cases, one is able to consider biomechanical performance measures like joint displacement or discomfort, when studying products for potential design changes. This in turn provides a novel approach to performance-based design.

Zone differentiation is based on optimization-based posture prediction implemented within the Santos DHM, whereby optimization is used to determine joint angles that optimize a specified combination of performance measures, subject to constraints that represent the task being simulated. This approach is computationally fast, so it is possible to predict and evaluate postures for large numbers of scenarios in a relatively small period of time. Zone differentiation entails automatically running posture prediction with millions of target points, and recording the consequent performance-measure values for each posture. The values are then presented as a 3D contour plot.

To date, this tools use with complex real-world problems and whole-body DHM models has been impractical, because it has not been fast enough, especially with cases that require collision avoidance. A variety of work has been completed with optimization-based posture prediction as summarized by Marler (2005), and zone differentiation is a natural outgrowth from such capabilities. One of the first developments in this regard is provided by Yang et al (2006), in the context of a 21-degree-of-freedon (DOF), one-arm, upper body system. Similar methods and results are presented by Yang et al (2008), albeit with a more extensive description of the discomfort model from Marler et al (2005). Yang et al (2008) extend this work with applications to discomfort analysis within an automobile cab, and with the ability to view preliminary results before the complete zone is calculated. Again, the tool is used for upper-body analysis. The same work surfaces in Yang et al (2009) and Yang and Abdel-Malek (2009), although the latter focuses on analytical determination of reach envelopes.

This paper presents new capabilities for whole-body zone differentiation, including refinements that increase computational speed. First, multi-dimensional zone differentiation allows one to compute and compare multiple zone differentiation volumes calculated with different constraints. Secondly, iso-contour surfacing allows the user to specify desired thresholds for visualizing the performance measures used to calculate zone volumes, thus reducing the final size of the volume. It is then possible to shrink wrap the resulting volume and export and/or manipulate the consequent geometry. Finally, a series of computational enhancements have been implemented to increase the speed of zone differentiation, including parallel processing, with the potential for cloud-based use.

2 Background

This section provides an overview of the underlying human model and the formulation for posture prediction. The work presented in this paper uses the Santos human model (Abdel-Malek et al, 2004) as a platform for further development. The underlying skeletal structure for Santo is modeled as a series of links with each pair of links connected by one or more revolute joints (Figure 1). There is one joint angle for each DOF, and the relationship between the joint angles and the position of points on the series of links (or on the actual avatar) is defined using the Denavit-Hartenberg (DH)-notation (Denavit and Hartenberg, 1955). This structure is becoming a common foundation for additional work in the field of predictive human modeling and has been successfully used with other research efforts (Ma et al, 2009; Howard et al, 2010).

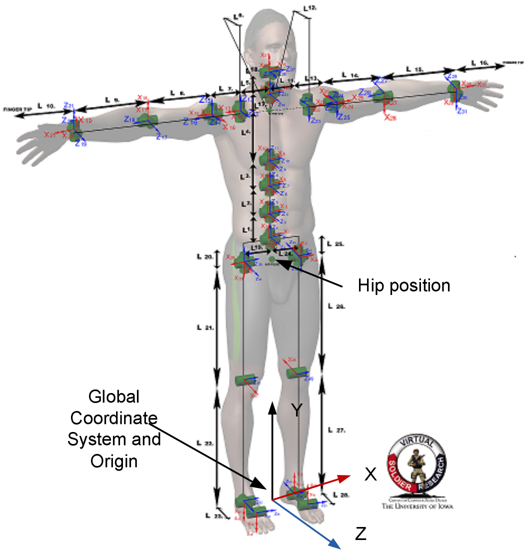

Fig. 1. The Santos Model

Given the structure in Figure 1, postures are predicted using an optimization-based approach first detailed by Marler (2005). Joint angles serve as the design variables, which are incorporated in various objective functions and constraints, the fundamental formulation for which given as follows:

Find: $\mathbf{q} \in R^{DOF}$

To minimize: $f(\mathbf{q})$

Subject to:

$$\text{Distance} = \left\| \mathbf{x}(\mathbf{q})^{\text{end-effector}} - \mathbf{x}^{\text{target point}} \right\| \leq \varepsilon$$

$$q_i^L \leq q_i \leq q_i^U;\; i = 1, 2, \ldots, DOF$$

$\boldsymbol{q}$ is a vector of joint angles, $\boldsymbol{x}$ is the position of an end-effector or point on the avatar, ε is a small positive number that approximates zero, and DOF is the total number of degrees of freedom. With this study, a single model with 113 DOFs is used for the human torso, arms, legs, neck, hands, eyes, and global position and orientation. Including the global DOFs as additional design variables allows one to predict the position and orientation of the body as well.

f(q) can be one of many performance measures (Marler *et al*, 2005; Marler *et al*, 2005b; Marler, 2005; Marler *et al*, 2009). For these studies, joint displacement and discomfort are used. In general, the objective function, potentially composed of of many performance measures, models what drives human behavior.

The primary constraint, called the *distance constraint*, requires the end-effector(s) to contact a specified target point(s). q_i^U represents the upper limit, and q_i^L represents the lower limit. These limits are derived from anthropometric data. In addition to these basic constraints, many other constraints can be used as boundary conditions in order to represent the virtual environment. In particular, in order to model collision avoidance, spheres are used to represent all geometry and all avatars, and *collision constraints* are added such that no one sphere overlaps another. Additional constraints can be created by the user on the fly, in order to simulate infinitely many tasks.

Typically, posture is predicted for a single scenario (a single set of constraints). A single objective function is used, although multiple performance measures may be aggregated to form what is technically a multi-objective optimization problem. However, regardless of the objective function being used, any performance measure can be evaluated at a consequent posture. In fact, zone differentiation involves predicting postures for potentially millions of target points surrounding the avatar, recording the consequent set of performance-measure values for each target, and then displaying the performance values corresponding to each target point within a volume around an avatar. This volume is called the *zone volume*.

3 Method

3.1 Multi-dimensional Zone Differentiation

Although zone differentiation has been a powerful tool for ergonomic analysis, there is been a need to automatically evaluate zone volumes for different conditions. For instance, one may need to evaluate a discomfort zone for different wrist orientations or different loading condition. This is called *multi-dimensional zone differentiation.* This functionality provides one with ways to batch more than one zone differentiation computation based on certain user configurable parameters. With respect to initial use cases, this tool provides a way to batch and compute several different volumes based on changing rotation of a 3D world entity, such as a lever of hand break (Figure 2).

The user specifies the range of rotation around the X, Y, and Z axes. Then, once the user specifies the geometry of interest, the tool automatically detects the axis on which the user may intend to rotate the geometry. The user may then choose the number of volumes for each axis. The total number of zones computed is based *L*M*N* where L, M, and N indicate the number of volumes chosen per axis. The user may then submit the problem for solution, and the submission window allows the user to preview the orientation of the geometry of interest.

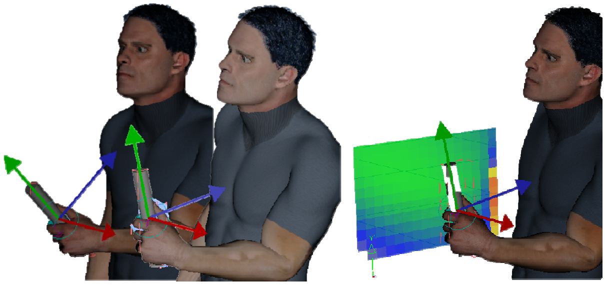

Fig. 1. Computing multiple zone volumes with changes in entity orientation

Multi-dimensional zone differentiation is a compute-intensive process and relies on partial-zone evaluation, parallelization, and batch computation techniques to deliver results in a reasonable amount of time, all of which are discussed in the following sections.

3.2 Computational Speed

Although posture prediction typically requires less than one second to run, running multiple zone volumes with millions of target points can be computationally demanding. Therefore, the following steps have been taken to increase the speed. First, we provide the feature of *sub-zone volume* calculations. This allows the end-user to take a more focused approach and define reduced zone volumes that do not necessarily include the complete volume around the avatar, some areas of which may not be important for the problem at hand.

A single instance of posture prediction can require more time when collision avoidance (Johnson *et al*, 2010) is involved. This is because, in order to incorporate collision avoidance, all avatars and geometry are represented by sphere-based surrogate geometry, and many additional constraints are included in the optimization problem to ensure one sphere does not overlap another. Thus, the second step in increasing the efficiency of zone differentiation entails a modified approach to selecting sphere constraints. Since we know that at most, an avatar will only intersect the spheres in the zone volume, all other spheres are discarded before zone differentiation (and posture prediction) begins. Note that previous work presents a multi-run approach to collision avoidance, which significantly reduces the number of spheres used in collision avoidance (Johnson *et al*, 2009). This method is "smart" in determining which spheres need to be used in avoidance constraints, but it still has to determine which spheres are colliding with the computed posture, and then update the constraint set. Although this approach is faster than considering all spheres in the environment, it is not fast enough for zone differentiation.

Next, one of the most significant improvements in computational time entails eliminating consideration of avatar/object collision spheres that are already in collision when the zone diff volume calculations begin.

In addition to reducing the number of sphere constraints, we implement a *reduced target set*, whereby target points (for posture prediction) that happen to fall within existing geometry are automatically disregarded. It would be impossible to reach such points.

Zone differentiation provides an ideal opportunity for parallelization. Target points inside the zone volume are evaluated independently thereby eliminating data dependency chains. The parallelization of zone volumes is achieved by splitting the volume into smaller fixed size volumes, which are then computed separately and in parallel on all available CPUs. Interestingly, paralleling zone volumes only provides a significant benefit for larger volumes. The amount of effort required to split, compute, and combine smaller volumes outweighs the effort required to compute a small volume using non-parallelized zone differentiation. This break-point volume is approximately 8x8x8.

Since zone differentiation is computationally intensive it makes little sense to block a user's access to the 3D program while the volume is computed. Thus, zone differentiation batch processing provides a way to run zone differentiation out-of-process, whereby a separate program accepts computation requests and executes them one after another.

How fast zone runs is now is determined by how many processors a machine has. The performance boost will be approximately 0.8 * (number of cores).

3.3 Sphere Filling

An integral component of collision avoidance, and thus the effectiveness of posture prediction, is the process by which the avatar and environment are represented with sphere-based surrogate geometry. New developments with the underlying approach to filling objects with spheres have led to substantial improvements in zone differentiation. Sphere filling is more accurate, flexible, and orders of magnitude faster. Specific improvements are summarized as follows.

Now, the avatars are actually filled with spheres based on their morphology, whereas previously, the body-based spheres were fixed. Sphere representation of an avatar can now be completed by separating the avatar geometry into two regions defined by the hands and then the rest of the avatar. This provides the end-user with the ability to obtain highly accurate representations of the avatar hands and fingers for collision avoidance, without significantly increasing the number of spheres used to represent the rest of the avatar body.

In addition, the avatar spheres can be re-oriented on the fly, depending on the posture. Sphere representations of avatars can be dramatically different based on posture. For instance, sphere filling based on a seated posture eliminates overestimation in the hip region, rather than using standing posture and then moving the avatar joints into a seated posture.

With respect to the use of spheres to represent general imported geometry, the underlying inflate algorithms has been improved. First, the algorithm can now lump together various geometry components and consider the results as one piece for filling purposes. This increases the speed with which complex composite objects can be represented. There is a preprocess step to group triangles (fundamental elements for creating geometry) to the grid points (points where spheres are placed) they intersect. Then, grid points that are inside an object are determined differently. Instead of evaluating the orientation of nearby triangles as it fills at every grid point, it now assumes all grid points on the outside of a bounding box are outside the object, and conducts a breadth first search marking grid points as outside until a grid point intersects geometry. Since the algorithm knows which points are inside and outside, it only fills (places spheres at) inside points. With the geometry grouped to grid points, the algorithm only calculates distance to the nearby triangles instead of to all of them.

The culling portion of the algorithm, which subsequently determines which spheres to retain/use, has been redesigned. The old method used a greedy approach, picking the spheres that covered the most grid points and then recalculating the coverage of the other spheres. The new method makes a list of the largest spheres providing coverage to each grid point. It then starts filling by using the smallest spheres first, but only retaining spheres that are the largest spheres for covering a specific point.

Computation time for sphere filling used to dedicate 70% of the time to distance calculation and then 30% to culling. Now time is allocated as 10% for preprocessing, 88% for distance calculation, and 2% for culling (the new culling method is much faster). Overall, sphere filling is more than twice as fast.

3.4 Zone Differentiation Compute Instance (ZDCI)

In addition to developing new zone-differentiation capabilities, an overarching management system has been implemented. This program is responsible for accepting compute requests, batching, monitoring progress, enabling cancel requests, and presenting computation progress. This program is automatically instantiated when it is needed. ZDCI accepts connections over HTTP using a REST-full interface. All needed parameters to run a zone computation are delivered using JSON. The Santos software submits tasks by connecting to ZDCI over HTTP and pushing a request using the POST method with the associated JSON payload.

This tool allows one to run zone differentiation independently from other Santos-related processes. It also provides the opportunity of scaling zone differentiation to external, more powerful, dedicated compute clusters. Since ZDCI communicates over HTTP, any external software package can submit computation requests over the Internet, thereby allowing for cloud computing.

3.5 Iso-contour Surfacing

The color gradients used to present zone differentiation results are called *iso-contour surfaces*, and a suite of capabilities have been developed for manipulating these surfaces. Iso-contour surfacing provides the ability to generate surface geometry from zone differentiation data via a marching-cubes algorithm and an end-user definable zone differentiation data threshold value. The threshold value can be interactively modified to indicate what is of interest (and what is not). Values below the threshold value are considered "of interest" and are visible, while values above the threshold value are made invisible, as shown in Figure 3. Once the desired threshold value is identified, invoking the iso-contouring algorithm creates 3-dimensional surface geometry that encompasses the zone differentiation data. Removal of the orthographic cutting planes, which can help the end-user gain insight into the volumetric data, shows the complete, resulting surface, which can be exported for use with third party systems.

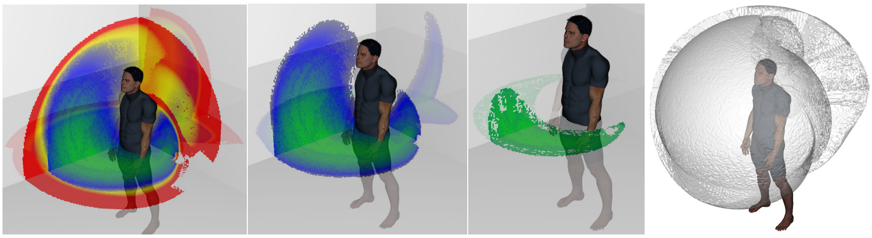

Fig. 2. Modification of threshholds with iso-contour surfaces

4 Results

In this section, three basic case studies are presented to further demonstrate the types of problems to which the new zone differentiation capabilities can be applied. The first example depicts Santos in an amusement park ride. As shown in Figure 4, collision avoidance is used to predict Santos's posture while reaching over a restraint, as if to help another passenger for safety reasons.

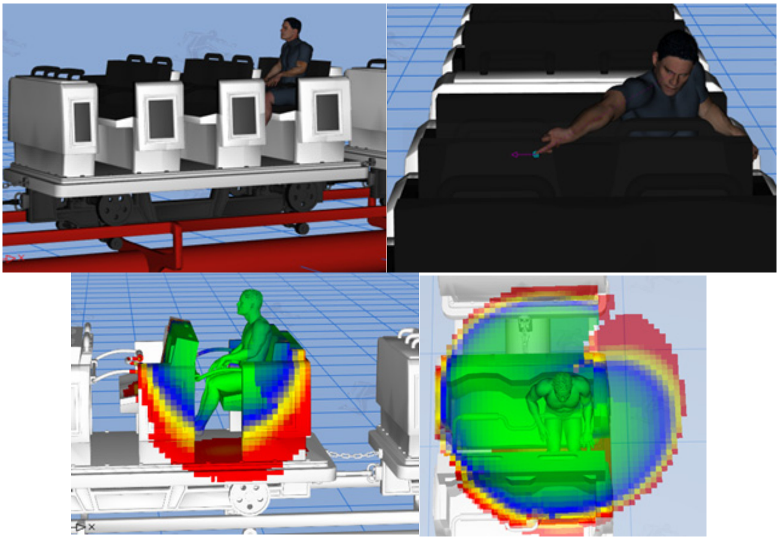

Fig. 3. Zone differentiation used for the analysis of an amusement park ride

Zone differentiation is then used to illustrate the joint-displacement values for all points around Santos. Ranging from green to red, the color scheme demonstrate Santos's difficulty, with red being the most difficult. A fixed volume around the ride was initialized before the test was run. The test shows the difficulty of reaching around the top of the front seat and the ease of touching the seat next to Santos. With respect to design considerations, in order to make sure a parent could reach his/her child in front of them, for instance, the seat would have to be redesigned.

As the second example, multi-dimensional zone differentiation is used to study the placement of a hand break in an automobile. With traditional zone differentiation, the user would have to manually re-orient the hand-brake and re-run zone differentiation for each orientation of interest. However, multi-dimensional zone differentiation automates this process.

Results in Figure 5 illustrate the difficulty of numerous beginning, middle, and ending postures during the hand break motion. A given restriction of X, Y, and Z was assigned to the hand brake object a priori. Then, for each plane, a number of zones were calculated. This example includes four zones. As shown from beginning to end of the avatar's hand brake motion, the green color begins to fade to red, where the red color corresponds to a less comfortable posture. The current location for the hand brake's motion is easily seen as acceptable, since all four postures are located in a green zone. The results do indicate, however, that there could be some benefit to moving the hand break forward slightly.

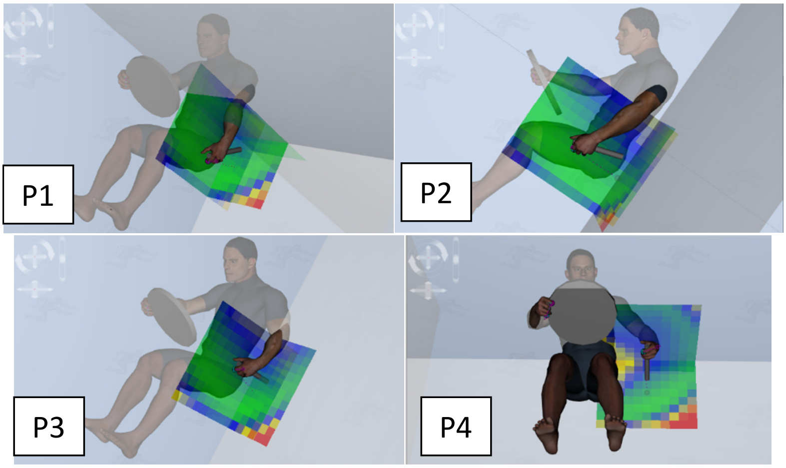

Fig. 4. Multi-dimensional zone differentiation for hand-break placement

As shown in Figure 6, iso-contours can be altered in order to vary the presentation of the scale of the underlying numerical results (for joint displacement values). Any portion of the zone volume can be selected based on performance-measure scale or contour color, and the consequent volume can be shrink wrapped and exported for use in third party design packages.

A third use case demonstrates the use of whole-body zone differentiation for concurrent seat design (Figure 7), whereby multiple ergonomic constraints (hand reach for joysticks, foot reach for pedals, visual target point, etc.) are considered concurrently.

The design criteria were as follows. The location of a point midway between the operator's hips when seated is provided. The range of motion of the seat is 7.2 cm forward of default location and 7.2 cm upward of default location. The joysticks are designed to be used while the elbows rested on the arm supports. The arm supports are not attached to the seat and are not adjustable. The left hand joystick rotates 15-degrees forward and aft. The brake is fully deployed at 17-degrees.

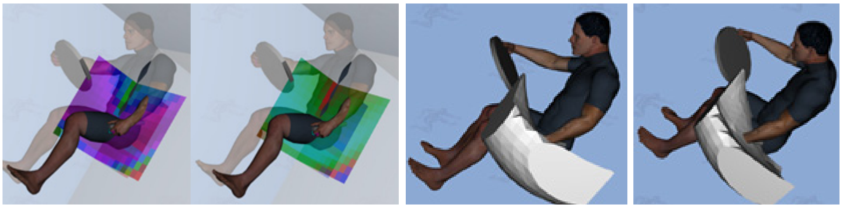

Fig. 5. Variation in iso-contour coloring scale, and shrink wrapped contours

Fig. 6. Seat design example

Posture is predicted by minimizing discomfort while requiring both feet to remain on the pedals and both hands to remain on the joysticks. Zone differentiation is used to determine the discomfort value for the predicted postures when the seat (and avatar) i.e. moved throughout the seat range of motion (ROM). Figure 8 displays the zone volume for the initial seat ROM, and then for a new ROM extended down and to the left. In the first case, the avatar actually has difficulty seeing the target and touching the pedals. When the ROM is extended, more comfortable postures become feasible, and the avatar is essentially able to relax.

5 Discussion

This paper presents new capabilities for optimization-based zone differentiation that yield a tool for human-centric, performance-based design. Improvements to collision avoidance, computational speed, and visualization allow zone differentiation to be used in a wider and more practical set of scenarios. This in turn allows one to automatically consider human performance when evaluating a virtual design. Most novel among the proposed capabilities is multi-dimensional zone differentiation, which allows one to consider variations in problem constraints when evaluating performance.

By leveraging real time posture prediction, this work provides one of the first DHM tools for automatic human systems integration. Especially with the final example, Santos actually determines beneficial design changes automatically while considering how a human interacts with the product. The inherent use of performance measures in the optimization-based posture-prediction construct is a key factor as to why this can be done.

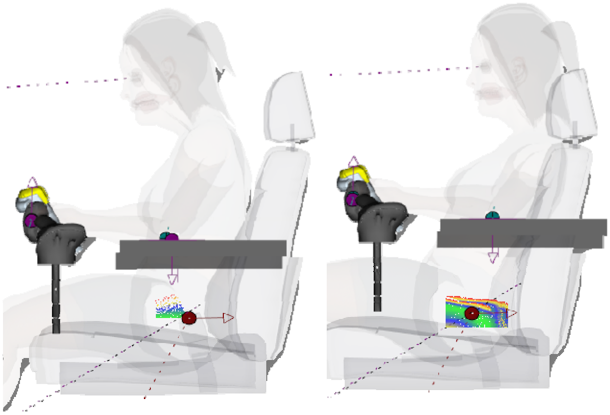

Fig. 7. Whole body seat-based zone differentiation

The presented work offers a new practical tool, and the implications of this work are significant. Ultimately, additional product parameters can be considered and automatically altered within an overarching optimal design loop. Furthermore, one can evaluate the differences in implied design changes when different performance measures are used. How does a seat designed to minimize discomfort differ from a seat designed to minimize joint displacement? Finally, these kinds of capabilities will not only be available within the Santos software, but given the provisions for ZDCI, they could be accessed by co-located users, thus fostering collaboration and concurrent design on a large scale.

References

1. Abdel-Malek, K., Yang, J., Kim, J., Marler, R.T., Beck, S., Nebel, K.: Santos: A Virtual Human Environment for Human Factors Assessment, In: 24th Army Science Conference, Orlando, FL, Assistant Secretary of the Army (Research, Development and Acquisition), Department of the Army, Washington, DC (November 2004)
2. Denavit, J., Hartenberg, R.S.: A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. Journal of Applied Mechanics 77, 215–221 (1955)
3. Howard, B., Yang, J., Gragg, J.: Toward a New Digital Pregnant Woman Model and Kinematic Posture Prediction. In: 3rd International Conference on Applied Human Factors and Ergonomics, Miami, FL (July 2010)
4. Johnson, R., Fruehan, C., Schikore, M., Marler, T., Abdel-Malek, K.: New Developments with Collision Avoidance for Posture Prediction. In: 3rd International Conference on Applied Human Factors and Ergonomics, Miami, FL (July 2010)
5. Johnson, R., Smith, B.L., Penmatsa, R., Marler, T., Abdel-Malek, K.: Real-Time Obstacle Avoidance for Posture Prediction. In: SAE Digital Human Modeling Conference, Goteborg, Sweden. Society of Automotive Engineers, Warrendale (June 2009)
6. Liu, Q., Marler, T., Yang, J., Kim, J., Harrison, C.: Posture Prediction with External Loads – A Pilot Study. In: April, D.M. (ed.) SAE 2009 World Congress, Detroit, MI. Society of Automotive Engineers, Warrendale (April 2009)

7. Ma, L., Zhang, W., Chablat, D., Bennis, F., Guillaume, F.: Multi-objective Optimization Method for Posture Prediction and Analysis with Consideration of Fatigue Effect and Its Application Case. Computers and Industrial Engineering 57, 1235–1245 (2009)
8. Marler, R.T.: A Study of Multi-objective Optimization Methods for Engineering Applications, Ph.D. Dissertation, University of Iowa, Iowa City, IA (2005)
9. Marler, T., Arora, J., Beck, S., Lu, J., Mathai, A., Patrick, A., Swan, C.: Computational Approaches in DHM. In: Duffy, V.G. (ed.) Handbook of Digital Human Modeling for Human Factors and Ergonomics. Taylor and Francis Press, London (2008)
10. Marler, R.T., Arora, J.S., Yang, J., Kim, H.: –J., and Abdel-Malek, K., Use of Multi-objective Optimization for Digital Human Posture Prediction. Engineering Optimization 41(10), 295–943 (2009)
11. Marler, T., Knake, L., Johnson, R.: Optimization-Based Posture Prediction for Analysis of Box-Lifting Tasks. In: 3rd International Conference on Digital Human Modeling, Orlando, FL (July 2011)
12. Marler, R.T., Rahmatalla, S., Shanahan, M., Abdel-Malek, K.: A New Discomfort Function for Optimization-Based Posture Prediction. In: SAE Human Modeling for Design and Engineering Conference, Iowa City, IA. Society of Automotive Engineers, Warrendale (June 2005)
13. Marler, R.T., Yang, J., Arora, J.S., Abdel-Malek, K.: Study of Bi-Criterion Upper Body Posture Prediction using Pareto Optimal Sets. In: IASTED International Conference on Modeling, Simulation, and Optimization, Oranjestad, Aruba. International Association of Science and Technology for Development, Canada (August 2005b)
14. Yang, J., Abdel-Malek, K.: Human Reach Envelope and Zone Differentiation for Ergonomic Design. Human Factors and Ergonomics in Manufacturing 19(1), 15–34 (2009)
15. Xiang, Y., Chung, H.-J., Kim, J.H., Bhatt, R., Marler, T., Rahmatalla, S., Yang, J., Arora, J.S., Abdel-Malek, K.: Predictive Dynamics: An Optimization-Based Novel Approach for Human Motion Simulation. Structural and Multidisciplinary Optimization 41(3), 465–479 (2009)
16. Yang, J., Verma, U., Penmatsa, R., Marler, T., Beck, S., Rahmatalla, S., Abdel-Malek, K., Harrison, C.: Development of a Zone Differentiation Tool for Visualization of Posture Comfort. In: 2008 SAE World Congress, Detroit, MI (April 2008)
17. Yang, J., Sinokrot, T., Abdel-Malek, K., Beck, S., Nebel, K.: Workspace Zone Differentiation and Visualization for Virtual Humans. Ergonomics 51(3), 395–413 (2008)
18. Yang, J., Sinokrot, T., Abdel-Malek, K., Nebel, K.: Optimization-Based Workspace Zone Differentiation and Visualization for Santos. In: 2006 SAE World Congress, Detroit, MI (April 2006)
19. Yang, J., Verma, U., Marler, T., Beck, S., Rahmatalla, S., Harrison, C.: Workspace Zone Differentiation Tool for Visualization of Seated Postural Comfort. International Journal of Industrial Ergonomics 39, 267–276 (2009)