09DHM-0060

Real-Time Obstacle Avoidance for Posture Prediction

Ross Johnson, Brian Lewis Smith, Rajeev Penmatsa, Tim Marler, Karim Abdel-Malek
Virtual Soldier Research (VSR) Program, University of Iowa

ABSTRACT

Collision avoidance in digital human modeling is critical for design and analysis, especially when there is interaction between the avatar and his/her environment. This paper describes a new algorithm for obstacle avoidance with optimization-based posture prediction. This new approach is motivated by a need for decreased computational time and increased fidelity for modeling and analysis of collision avoidance tasks. Posture prediction is run in an iterative loop while conducting collision detection to dynamically update collision avoidance constraints. It is shown that this approach is substantially faster than the basic method involving a fixed number of sphere-based avoidance constraints with a single optimization/posture-prediction run. The method is demonstrated using an upper-body virtual human model in a cab setting.

INTRODUCTION

A key element in the development of human modeling capabilities is the interaction between the virtual human and his/her environment. Although an independent virtual human (avatar) can help one better understand human behavior, enabling an avatar to interact with a virtual environment increases human-modeling capabilities significantly and opens the door to more human-centric product design. A key element under the topic of virtual human-environment interaction is collision avoidance, which includes both obstacle avoidance (i.e. the avatar avoiding other geometry) and self avoidance (i.e. the avatar avoiding its self).

One advantage to optimization based posture prediction (Yang *et al*, 2006; Marler *et al*, 2007; Marler *et al*, in press) is the relative ease with which collision avoidance can be incorporated. Essentially, collision avoidance entails adding a set of constraints to the optimization problem. However, the approach to forming and incorporating such constraints can significantly affect predictive accuracy and computational speed.

Consequently, this paper presents a new algorithm for collision avoidance with optimization-based posture prediction. This new approach increases computational time substantially and thus allows one to increase fidelity with which a virtual world is modeled. With most computational models, increasing speed and fidelity tend to be conflicting objectives.

A basic, preexisting collision avoidance approach provides the foundation for this work. This basic approach involves representing the avatar and relevant geometry with spheres. Spheres used to model the avatar are called body spheres, whereas spheres used to represent geometry are called obstacle spheres. Constraints are then incorporated in the posture-prediction formulation such that no two spheres in the virtual environment can overlap. Although functional, this approach becomes impractical and slow with more complex obstacles or larger numbers of obstacles.

The new proposed algorithm for obstacle avoidance is a multi-run approach and involves cycling through multiple posture-prediction problems, noting relevant obstacles, and then considering only the associated spheres for avoidance constraints in the subsequent problems. The hypothesis is that the increase in computational time from reducing the number of constraints will more than compensate for the increase resulting from having to solve multiple optimization problems. Although this algorithm only applies to obstacle avoidance, not self avoidance, an increase in computational speed can allow for increased fidelity in the body spheres and thus improved performance with self avoidance. Ultimately, a reduction in computational time could provide real-time posture prediction with relatively low sensitivity to the number of and complexity of obstacles.

The primary intent of this work is algorithm development. This is considered a first step towards a robust validated tool for real-time posture prediction with collision avoidance. As a tertiary goal, this work involves the use of and comparison of two different sphere-generating algorithms. Given that any collision-avoidance approach in the context of optimization-base prediction likely involves representing geometry with spheres, the method by which spheres are generated becomes critical.

This paper first discusses the basics of optimization-based posture prediction. Two sphere-generating algorithms are then summarized. The proposed

algorithm is then outlined in reference to the existing basic approach. Finally, a series of example problems are presented and discussed.

OPTIMIZATION-BASED POSTURE PREDICTION

In this section, an overview of human optimization-based posture prediction is discussed. This includes a brief description of the skeletal model, as well as the final optimization formulation.

Simulating human posture depends largely on how the human skeleton is modeled. One way to view a skeleton is as a kinematic system, or series of links with each pair of links connected by one or more revolute joints. Therefore, a complete human body can be modeled as several kinematic chains, formed by series of links and revolute joints, as shown in Figure 1.

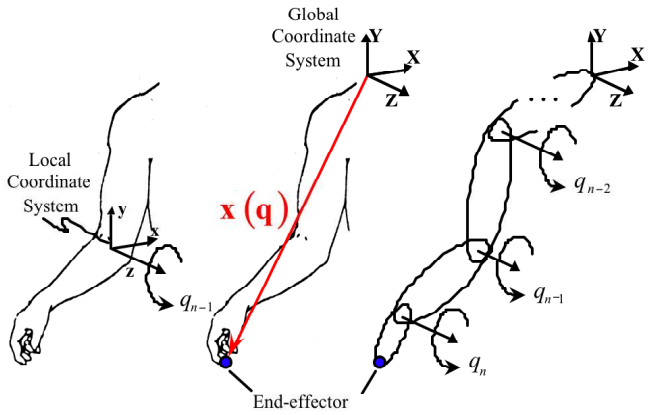

Figure 1: A Kinematic Chain of Joints

q_i is a *joint angle* and represents the rotation of a single revolute joint. There is one joint angle for each degree of freedom (DOF). $\mathbf{q} = [q_i, \ldots, q_n]^T \in R^n$ is the vector of joint angles in an n-DOF model and represents a specific posture. Each skeletal joint is modeled using one, two, or three kinematic revolute joints. $\mathbf{x(q)} \in R^3$ is the position vector in Cartesian space that describes the location of the end-effector as a function of the joint angles, with respect to the global coordinate system. For a given set of joint angles $\mathbf{q}$, $\mathbf{x(q)}$ is determined using the Denavit-Hartenberg (DH)-method (Denavit and Hartenberg, 1955).

Using the DH-method $\mathbf{x(q)}$ is expressed in terms of a series of transformations ${}^{i-1}\mathbf{T}_i$ and is given by:

$$\mathbf{x(q)} = (\prod_{i=1}^{n} {}^{i-1}\mathbf{T}_i)\mathbf{x}_n \qquad (1)$$

where $\mathbf{x}_n$ is the position of the end-effector with respect to the n^{th} frame and n is the number of DOFs. Note that the rotational displacement q_i changes the value of θ_i.

With this study, a 35-DOF model for the human torso, right arm, left arm, and neck is used and is shown in Figure 2, where each cylinder represents a rotational DOF.

q_1 through q_{12} represent the torso. q_{13} through q_{17} represent the right shoulder and clavicle. q_{18} through q_{21} represent the right arm. q_{22} through q_{30} represent the left arm. q_{31} through q_{35} represent the 5-DOF neck model. The link lengths between each of the joints are variable and can be set based on anthropometric data, thus representing various population variations.

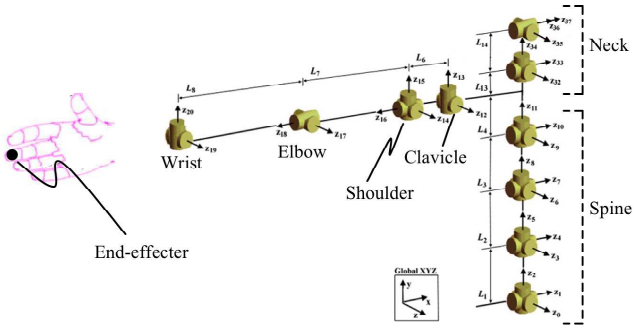

Figure 2: Human Model

OPTIMIZATION FORMULATION

The posture of the above-described model is determined by solving the optimization problem formulated in this section. The design variables for the problem are q_i, measured in units of radians. The vector $\mathbf{q}$ represents the consequent posture.

The first constraint, called the *distance* constraint, requires the end-effector to contact a target point. In addition, each joint angle is constrained to lie within predetermined limits. q_i^U represents the upper limit for q_i, and q_i^L represents the lower limit. These limits are derived from anthropometric data and ensure that the avatar does not assume an unrealistic posture.

The basic benchmark performance measure represents joint displacement (Jung *et al*, 1994; Yu, 2001; Mi *et al*, 2002). This performance measure is proportional to the deviation from the *neutral position* (Smith *et al*, 2009), which is selected as a relatively comfortable posture, typically a standing position with arms at one's sides. q_i^N is the neutral position of a joint, and $\mathbf{q}^N$ represents the overall neutral posture. Because some joints articulate more readily than others, a weight w_i is introduced to stress the relative stiffness of a joint. Although these weights are typically based on trial-and-error, the consequent performance measure provides a baseline performance measure with consistently reasonable results and thus a foundation for new posture-prediction developments. The final joint displacement is given as follows:

$$f_{JointDisplacement}(\mathbf{q}) = \sum_{i=1}^{n} w_i (q_i - q_i^N)^2 \qquad (2)$$

The optimum posture for the system shown in Figure 2 is then determined by solving the following problem:

Find: $\mathbf{q} \in R^{DOF}$ (3)

to minimize: $f_{jointDisplacement}(\mathbf{q})$

subject to:

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

$$q_i^l \leq q_i \leq q_i^u ; i = 1,2, \ldots, DOF$$

where ε is a small positive number that approximates zero. (3) is solved using the software SNOPT (Gill *et al*, 2002), which uses a sequential quadratic programming algorithm. Analytical gradients are determined for the objective function and for all constraints. Note that the absolute values of the performance measures are not significant.

SPHERE FILLING ALGORITHMS

The formulation in (3) can be altered to incorporate collision avoidance by restricting various items from touching each other. This is done by adding constraints to the formulation. The question is how to model such constraints mathematically. Given that a gradient-based optimization method is used to solve (3), all objective functions and constraints should be continuous and differentiable. Thus, spheres are chosen as primitives for representing the avatar and the geometry. They lend themselves to simple and continuous equations that restrict the distance between objects based on eth sphere radii. The shape and size of the avatar is generally constant, although anthropometry may vary. However, geometry in the virtual environment (other than eth avatar) can vary significantly in shape and size. Thus, the method by which geometry is represented with spheres is critical to the performance of any collision avoidance approach.

In the Santos environment all imported geometry is internally represented by triangles as the basic building block. Collectively, these triangles form a mesh for an object. Using these triangles, however, in an optimization constraint presents a number of difficulties. Unlike spheres, which are represented by a point and a radius, a triangle is represented by an equation of a plane and three line equations. This complicated representation makes calculating whether two triangles intersect much more computationally difficult and expensive than simply determining whether two spheres intersect.

Any geometry in 3D space can be represented by a collection of spheres, and the methods and techniques for generating this approximation are numerous. Two different techniques have been tested for use with posture prediction, and they are illustrated in the following subsections. The first method uses spheres with equal radii and is called sphere shelling. The second method uses spheres with unequal radii and is called the adaptive medial-axis approximation method.

METHOD 1 – SPHERE SHELLING

With this method, a given 3D mesh's bounding box is filled with spheres. A representation of the process is given in Figure 3. The box is filled with spheres of a given radius line by line, column by column. The green spheres represent spheres that contain at least one vertex (spheres intersecting with at least one triangle on the geometry). Red spheres are empty, i.e., they do not contain any vertices. This results in the geometry being 'shelled', i.e., only the outside 'skin' of an object is approximated by spheres. This algorithm continues until the entire bounding box of the object is filled with spheres.

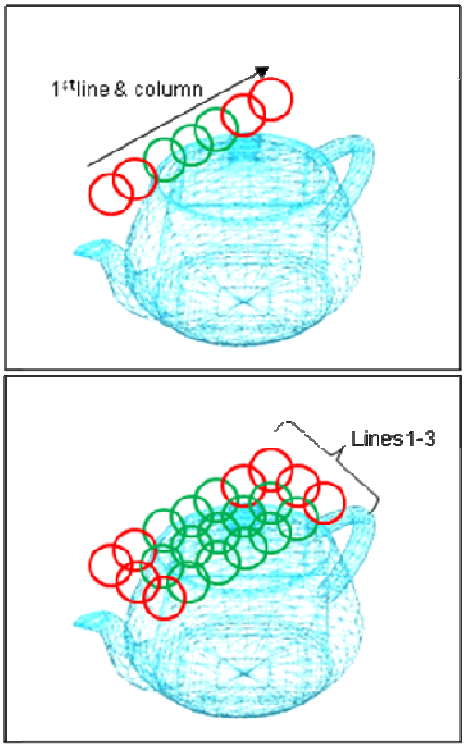

Figure 3: Sphere Shelling Procedure

As illustrated in figure 4, the red spheres are then deleted, which results in the 'shell' of the given 3D object.

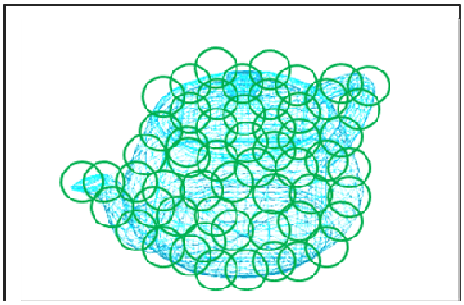

Figure 4: Sphere Shelling Result

METHOD 2 – ADAPTIVE MEDIAL-AXIS APPROXIMATION

This method, although slightly slower than sphere shelling, provides two advantages. First, it is a filling method, so it represents the complete geometry, not just the boundary. Second, it is adaptive, which means the filling spheres are adjusted to optimally fill any geometric form. The construction of the spheres is based on three-dimensional Voronoi diagrams (Bradshaw *et al*, 2004). The algorithm is based on the Voronoi diagram proposed by Hubbard (1996). A Voronoi diagram in a two-dimensional context involves a set of points $\boldsymbol{P}$, called Voronoi sites, given on a plane. Each point p has a Voronoi cell $V(p)$ consisting of all points closer to p than to any other point. This concept is extended to the third dimension where a Voronoi cell would consist of all points closer to p in three-dimensional space. A medial axis is defined as the set of centers of a set of maximally sized spheres filling an object. The Voronoi points inside the object define the medial axis, and the spheres are formed based on the medial axis. Then, the spheres are automatically merged, removed, and expanded to refine the results. An example of spheres resulting from this method are illustrated in Figure 5.

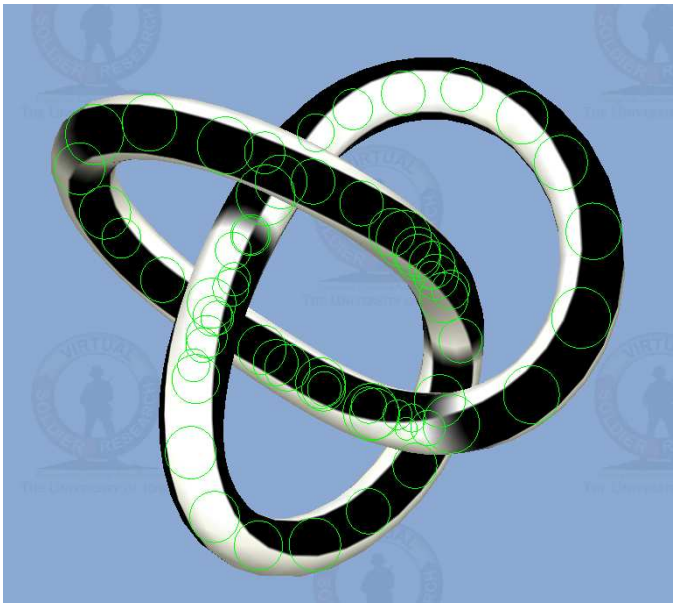

Figure 5: Sphere Filling Result

Note that the medial axis approach is much more accurate than the shelling algorithm. In addition, there is much less over approximation of the geometry. The medial axis approach is slightly slower than shelling, but if all geometry is represented with spheres as it is imported into the virtual environment, and if the spheres are generated only once, then computational time for generating spheres is not necessarily a critical issue.

COLLISION AVOIDANCE ALGORITHM DEVELOPMENT

Given the algorithms for representing a virtual environment (or part of an environment) with spheres, the formulation in (3) can be extended to include collision avoidance (self avoidance and obstacle avoidance). Extra constraints are added to the optimization formulation and prevent an avatar's body spheres from intersecting other body spheres and obstacle spheres. Body spheres are grouped based on their locations, and spheres in the same group are not constrained with other spheres in the same group. For example, there is no constraint preventing a sphere in the torso and another sphere also in the torso from intersecting, as the joint limits alone will prevent this collision. Obstacle spheres are represented by a global position and a radius, and body spheres are represented by a local position relative to a joint, and a radius. Then, sphere-based constraints simply require that no two spheres intersect, and this condition is modeled by limiting the distance between sphere centers based on sphere radii. Thus, the sphere avoidance constraints are similar to the end-effector-target constraints. The final updated formulation is stated as follows:

Find: $\mathbf{q} \in R^{DOF}$ (4)

to minimize: $f_{jointDisplacement}(\mathbf{q})$

subject to:

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

$$\text{obstacledDistance} = \left\| \mathbf{x}(\mathbf{q})^{\text{body sphere}} - \mathbf{x}^{\text{obstacle sphere}} \right\| \geq r^{\text{body sphere}} + r^{\text{obstacle sphere}}$$

$$\text{selfAvoidanceDistance} = \left\| \mathbf{x}(\mathbf{q})^{\text{bodysphere1}} - \mathbf{x}(\mathbf{q})^{\text{bodysphere2}} \right\| \geq r^{\text{bodysphere1}} + r^{\text{obstacles}}$$

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

Although this formulation provides reasonable results, there is room for improvement with regards to computational speed. With this basic approach, every body-sphere is constrained with every obstacle sphere in the environment. This means that for n obstacle spheres and m body spheres, there will be $n \times m$ constraints added to the optimization problem. This can make it difficult to obtain real time results, especially when a relatively large number of spheres (and consequent number of constraints) is used to increase the fidelity with which the avatar and/or geometry is modeled/represented. Consequently, only nine spheres are used to represent Santos' upper-body, and no more than a few hundred spheres can be used to represent obstacles in the environment.

The new method for obstacle avoidance reduces runtime by drastically reducing the number of sphere-to-sphere constraints that are included in the optimization problem. In general, the method uses a multi-run approach,

whereby the optimization solver runs (the formulation in (4) is solved) multiple times, each time including additional necessary sphere-to-sphere constraints. The method is summarized with the flowchart shown in Figure 6.

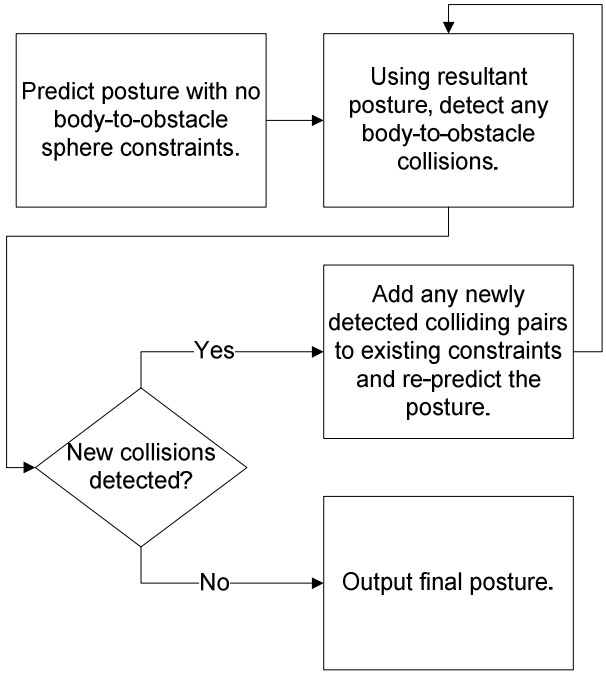

Figure 6: Multi-Run Approach Flow Chart

First, posture prediction is run with no collision-avoidance constraints. Then, using the resultant posture, body-to-obstacle collisions are identified. For every colliding set of spheres a corresponding constraint is incorporated in the posture-prediction formulation for obstacle avoidance, and the problem is re-run with the new constraints. Any new body-to-obstacle collisions are identified, and the process is repeated until there are no collisions. Thus, the final optimization run includes the minimum necessary body-to-obstacle sphere constraints. Note that the multiple runs of the optimization problem are completed in the background; the user only sees the final results (the final posture).

The main benefit from this method arises from the fact that simply checking for sphere collisions is significantly faster than including all of the sphere pairs as constraints in the optimization problem. Even though the optimizer is running multiple times, the significant reduction in the number of constraints allows us to predict realistic postures that avoid collisions in real-time.

An added benefit of this method is that the time improvement allows for increased fidelity of the avatar's self avoidance spheres while still maintaining real-time results. The number of spheres used to represent the upper-body has been increased from 9 to 16, as shown in Figure 7.

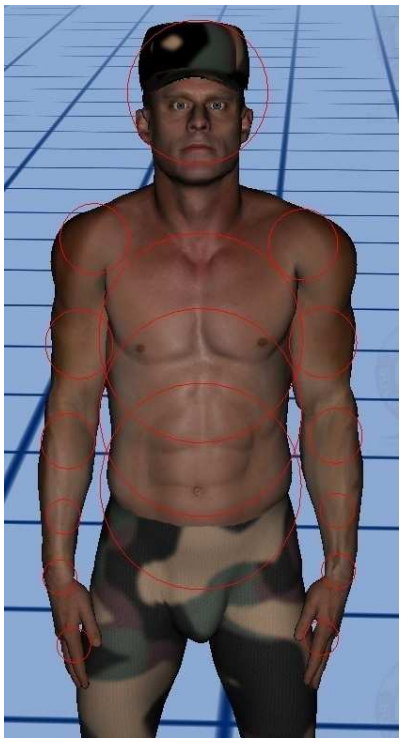

Figure 7: Santos' New Body Spheres

RESULTS

TEST CASE #1

The first test case provides a basic example of the proposed algorithm. Santos is placed in a seated posture inside a cab. One end-effector is placed on his right index finger, and the corresponding target point is placed on the dashboard. The purpose of this test case is to create a situation such that without collision avoidance, Santos' arm intersects the steering wheel. The steering wheel is shelled with spheres, and the posture is predicted with both the basic collision avoidance approach and the new multi-run approach. To keep the two methods as comparable as possible, the updated body spheres are used for both methods (shown in Figure 7). Since both the basic and new collision avoidance methods produce the same final posture, only one is shown. The predicted posture with no collision avoidance is also shown for reference. The predicted postures are shown in Figure 8. As one can see, with collision avoidance enabled, Santos' right arm no longer collides with the steering wheel. The left arm is still colliding with the seat because the seat has not been filled with spheres.

Table 1 shows numerical information about the problem formulations and results for test case #1. The number of iterations for the basic method is always 1, but this varies for the multi-run approach based on the problem. In this context, an iteration is defined as one run/solution of the posture prediction; it does not refer to a computational optimization iteration. The fourth row in Table 1 indicates the total number of body-obstacle sphere-pair constraints considered for the basic method,

and the final number of sphere constraints that were considered for the multi-run approach. Despite having to run two optimization iterations, the multi-run approach produced a solution much faster than the basic method. The multi-run approach only considered 0.86% of the total possible sphere constraints, and produced a result 13.2 times faster than the basic method.

Figure 8: Case #1 with no collision avoidance (left) and collision avoidance (right)

Table 1**:** Test case #1 results

	Basic Method	Multi-Run Approach
Total Obstacle Spheres	80	80
Total Body Spheres	16	16
Iterations of Optimizer	1	2
Sphere constraints (final)	1280	11
Total Runtime (seconds)	0.66	0.05

TEST CASE #2

The second test case involves a slightly larger optimization problem and addresses the issue of the left arm. The additional objects that are shelled with spheres are shown in Figure 9. Again, one end-effector is placed on his right index finger, and the corresponding target point is placed on the dashboard, such that without collision avoidance, Santos' arm intersects the steering wheel. The steering wheel, steering column, seat, arm rests, and various control knobs are shelled with spheres. Posture is predicted with both the basic collision avoidance approach and the multi-run approach. It can be seen that with collision avoidance enabled, Santos' left arm no longer collides with the seat. In this case, the left arm is not assigned an end-effector and target point.

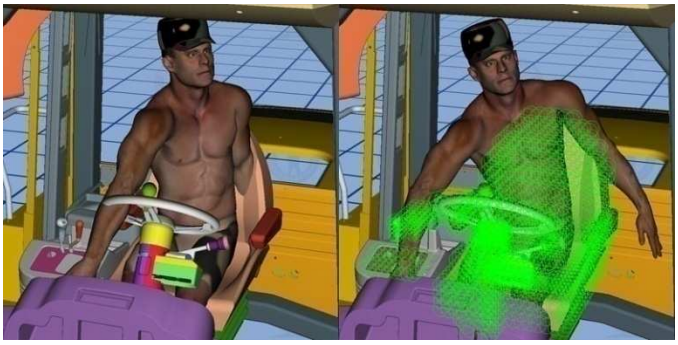

Figure 9: Case #2 with no collision avoidance (left) and collision avoidance (right)

Table 2 provides numerical output for this case. The multi-run approach considers just 0.47% of the total possible sphere constraints, and it results in a solution 301.5 times as fast as the basic method. This test case demonstrates that the multi-run approach works well in situations with many filled objects and a large number of obstacle spheres. When comparing this case to case #1, it can be determined that by increasing the number of obstacle spheres by a factor of 27.15, the computation time of the basic method increased by a factor of 118.8, while the multi-run computation time increased only by a factor of 5.2. This indicates that the multi-run approach scales much better than the basic approach.

Table 2**:** Test case #2 results

	Basic Method	Multi-Run Approach
Total Obstacle Spheres	2172	2172
Total Body Spheres	16	16
Iterations of Optimizer	1	3
Sphere constraints (final)	34752	163
Total Runtime (seconds)	78.39	0.26

TEST CASE #3

The third test case involves complex geometry in the form of a torus knot shown in Figure 10. One end-effector is placed on the pointer finger of each hand. Corresponding target points are placed on the surface of a torus knot. With this case, the adaptive medial axis sphere filling algorithm is used instead of the sphere shelling algorithm. The torus knot is filled with 64 spheres. Without collision avoidance, Santos' arms collide with geometry, but all geometry is avoided when collision avoidance is used.

Figure 10: Case #3 with no collision avoidance (left) and collision avoidance (right)

The multi-run approach eventually considered 1.56% of the total possible sphere constraints, and as shown in Table 3, it found a solution 42.7 times faster than the basic method.

Table 3**:** Test case #3 results

	Basic Method	Multi-Run Approach
Total Obstacle Spheres	64	64
Total Body Spheres	16	16
Iterations of Optimizer	1	3
Sphere constraints (final)	1024	16
Total Runtime (seconds)	6.41	0.15

CONCLUSION

This paper presents a new method for obstacle avoidance with optimization-based posture prediction. We have shown a substantial increase in computational speed with both theoretical and practical examples. The initial hypothesis, that a multi-run approach could increase speed despite the necessity for solving multiple optimization problems, was proven true. The new approach greatly reduces the number of sphere-to-sphere constraints that are active during the optimization process. The consequent increase in speed has allowed for an increase in fidelity with regards to modeling a virtual environment.

Concurrent with the development of a new obstacle-avoidance algorithm, we have compared two methods for generating spheres. Although sphere filling takes longer, the results are much more accurate than that of sphere shelling, and once spheres have been generated for a particular object, they do not need to be generated again, even if the object's orientation or direction is changed. The spheres essentially represent a meta-model of the human model and virtual environment, so their use is critical.

An ancillary improvement to our collision avoidance software is the improvement of body-sphere fidelity. Because the multi-run approach greatly increased the speed of collision avoidance posture prediction problems, the number of spheres used to represent the body of Santos can be increased without significantly reducing performance.

All of these improvements lead to a collision avoidance method that is fast enough to be used in real-time and biomechanically accurate enough to assist in ergonomic analysis of product design and workplace design. Due to the nature of the multi-run approach, increasing the number of spheres in the environment does not necessarily increase the run-time. This is a necessary development to allow collision avoidance to work automatically in large environments without requiring the user to specify which specific objects Santos should avoid.

Future work includes the use of sphere filling methods to generate spheres for Santos' body and skin. This would greatly increase the precision of obstacle avoidance and self avoidance. To further develop the obstacle avoidance capabilities of Santos, the predicted postures must be validated. To be sure, the presented results are evaluated subjectively, but full experimental validation is still needed.

REFERENCES

1. Bradshaw, G., O'Sullivan, C. (2004), "Adaptive Medial-Axis Approximation for Sphere-Tree Construction," *ACM Transactions on Graphics*, January, ACM, New York, NY.
2. Denavit, J., and Hartenberg, R. S. (1955), "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, 22, 215-221.
3. Farrell, K., Marler, R. T., and Abdel-Malek, K. (2005), "Modeling Dual-Arm Coordination for Posture: An Optimization-Based Approach," *SAE Human Modeling for Design and Engineering Conference*, June, Iowa City, IA, Society of Automotive Engineers, Warrendale, PA.
4. Gill, P., Murray, W., and Saunders, A, 2002, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal of Optimization*, 12(4), 979-1006.
5. Hubbard, P. (1996), "Approximating Polyhedra with Spheres for Time-Critical Collision Detectioin," *ACM Transactions on Graphics*, July, ACM, New York, NY.
6. Hubbard, P. (1996), **"**Improving Accuracy in a Robust Algorithm for Three-Dimensional Voronoi Diagrams**,"** *Journal of Graphics Tools*, A. K. Peters, Natick, MA.
7. Jung, E. S., Choe, J., and Kim, S. H. (1994), "Psychophysical Cost Function of Joint Movement for Arm Reach Posture Prediction," *Proceedings of the Human Factors and Ergonomics Society 38th*

Annual Meeting, October, Nashville, TN, Human Factors and Ergonomics Society, Santa Monica, CA, 636-640.
8. Marler, R. T. (2005), "A Study of Multi-Objective Optimization Methods for Engineering Applications," Dissertation, University of Iowa, Iowa City, Iowa.
9. Marler, R. T., Arora, J. S., Yang, J., Kim, H. –J., and Abdel-Malek, K. (in press), "Use of Multi-objective Optimization for Digital Human Posture Prediction," *Engineering Optimization*.
10. Marler, T., Yang, J., Rahmatalla, S., Abdel-Malek, K., and Harrison, C. (2007), "Validation Methodology Development for Predicted Posture," *SAE 2007 Transactions Journal of Passenger Cars – Electronic and Electrical Systems*, SAE paper number 2007-01-2467.
11. Mi, Z. (2004), "Task-Based Prediction of Upper Body Motion," Ph.D. Dissertation, University of Iowa, Iowa City, IA.
12. Mi, Z., Yang, J., Abdel-Malek, K., Mun, J. H., and Nebel, K. (2002), "Real-Time Inverse Kinematics for Humans," *Proceedings of the 2002 ASME Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, 5A, September, Montreal, Canada, American Society of Mechanical Engineers, New York, 349-359.
13. Smith, B. L., Marler, T., Meusch, J., Smith, R., Rahmatalla, S., Abdel-Malek, K. (2009), "Studying Alternative Neutral Positions for Optimization-based Posture Prediction," *13th International Conference on Human-Computer Interaction*, July, San Diego, CA, Springer, New York.
14. Yang, J., Marler, T., Beck, S., Abdel-Malek, K., and Kim, H. -J. (2006) "Real-Time Optimal-Reach Posture Prediction in a New Interactive Virtual Environment", *Journal of Computer Sciience and Technology*, 21 (2), 189-198.
15. Yang, J., Marler, R. T., Kim, H., Arora, J. S., and Abdel-Malek, K. (2004), "Multi-objective Optimization for Upper Body Posture Prediction," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, August, Albany, NY, American Institute of Aeronautics and Astronautics, Washington, DC.
16. Yang, J., Rahmatalla, S., Marler, T., Abdel-Malek, K., and Harrison, C. (2007), "Validation of Predicted Posture for the Virtual Human Santos," *12th International Conference on Human-Computer Interaction*, July, Beijing, China, Springer, London, England.
17. Yu, W. (2001), *Optimal Placement of Serial Manipulators*, Ph.D. Dissertation, University of Iowa, Iowa City, IA.

ACKNOWLEDGEMENTS

This research was funded by the USCAR project: Development of a New Generation Digital Human Model for Ergonomics in Automotive Design and Manufacturing, and the Caterpillar project: Digital Human Modeling and Simulation for Safety and Serviceability. The authors would like to thank Brent Rochambeau and Uday Verma for their work with the interface and controls, and Andy Taylor for his work with the sphere shelling algorithm.