New Developments with Collision Avoidance for Posture Prediction

Ross Johnson, Carl Fruehan, Matt Schikore, Tim Marler, Karim Abdel-Malek
Virtual Soldier Research (VSR) Program, University of Iowa

ABSTRACT

A substantial advantage of predictive virtual human models is the ability to adapt to changes in a virtual environment automatically, and with respect to posture prediction and analysis, this ability hinges on collision avoidance. Collision avoidance must be robust enough to accommodate various types of geometry, must apply to the avatar (self-avoidance) as well as virtual objects, must not detract from real-time operation, and must be suitable for a variety of real-world scenarios. Thus, while leveraging optimization-based posture prediction and a unique method for collision avoidance with increase computational speed we present new developments in this arena. A new sphere-filling algorithm is presented with increased speed and fidelity for creating surrogate geometry, which is critical for any type of collision avoidance or detection. The collision avoidance algorithm is implemented for self-avoidance. And, the new capabilities are demonstrated on automotive and motorcycle examples for ergonomic analysis. The results not only involve realistic predicted postures and novel forms of human-performance feedback, but also reflect real-time operation.

Keywords: Human Modeling, Optimization, Collision Avoidance, Sphere Filling

INTRODUCTION

A critical advantage to using digital human models is the ability to evaluate new products and processes virtually. However, fully recognizing this advantage requires the virtual human to interact with digital models in a 3-D environment. This interaction can be useful for identifying design issues relating to human factors and ergonomics, and can provide insight into human behavior. Such interaction between virtual humans and products often depends on predictive capabilities and the tendency of changes in the virtual environment to affect predicted responses.

Thus, this paper presents new developments with predictive collision avoidance capabilities, in the context of optimization-based posture prediction. Collision avoidance includes the avoidance of collisions with external objects, as well as the avoidance of collisions between one part of the digital human and another. The collision avoidance described here does not describe just the ability to detect when two objects collide, which is known as collision detection, but entails formulating the simulation in such a way that obstacles are actually considered as input to the problem, and therefore affect the results of a given task.

The optimization-based posture prediction has many advantages and can be adapted to solve a variety of digital human modeling problems (Yang et al., 2006; Marler et al., 2007; Marler et al., 2009). One main advantage of this technique is the relative ease with which one can expand the accuracy and utility of a simulation by adding new mathematical constraints that represent different real-world factors. These constraints can represent anything from location-specific targets (Farrell et al., 2005) to equations of static equilibrium (Liu et al., 2009), but there are practical limits to the number of constraints that can be added while still maintaining acceptable software runtimes.

Consequently, this paper presents new developments that increase both the speed and accuracy of collision avoidance for the optimization-based posture prediction approach. This work builds and expands on posture prediction capabilities for Santos™, a high-fidelity predictive human model (Abdel-Malek et al., 2006; Marler et al., 2008), so the optimization-based method for predicting posture is first summarized. Fundamental to most collision detection or avoidance approaches is the use of surrogate geometry. Consequently, we outline a new sphere-based algorithm for approximating geometry, which can be integrated in the optimization formulation. We then extend the multi-run obstacle avoidance method (Johnson et al., 2009) to function with self-avoidance. These improvements increase the accuracy, speed, and value of human modeling and simulation capabilities, and these improvements are demonstrated with practical examples.

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.

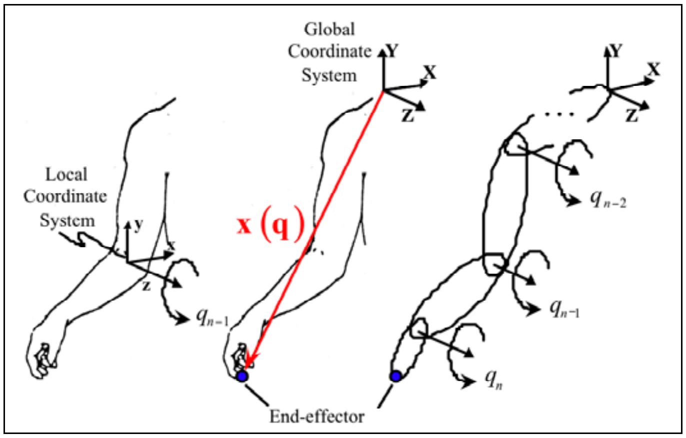

FIGURE 1 A kinematic chain of joints

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). $\boldsymbol{q} = [q_1 \quad \cdots \quad q_n]^T \epsilon \mathbb{R}^n$ is the vector of joint angles in an *n*-DOF model and represents a specific posture. Each skeletal joint is modeled using one or more kinematic revolute joints. $\boldsymbol{x}(\boldsymbol{q}) \epsilon \mathbb{R}^3$ is the position vector in Cartesian space that describes the location of an end-effector with respect to the global coordinate system. For a given set of joint angles $\boldsymbol{q}$, $\boldsymbol{x}(\boldsymbol{q})$ is determined using the Denavit-Hartenberg (DH)-method (Denavit and Hartenberg, 1955). With this work, a 55-DOF model for the human torso, arms, legs, and neck is used. This also includes six global DOFs, three for translation of the hip point and three for rotation about the hip point. The posture of this model is determined by solving the optimization problem formulated as follows.

The design variables for the problem are q_i, measured in units of radians. One 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, and q_i^L represents the lower limit. The basic benchmark performance measure, which serves as the objective function in the optimization problem, is joint displacement. This performance measure is proportional to the deviation from a *neutral position*, which is selected as a relatively comfortable posture, and is denoted q_i^N for a particular joint. Because some joints articulate more readily than others, a weight w_i is introduced to stress the relative stiffness of a joint.

The optimum posture for the system is then determined by solving the following problem:

$$\begin{aligned}
&\text{Find: } \boldsymbol{q} \epsilon \mathbb{R}^n \\
&\text{To minimize: } f_{JointDisplacement}(\boldsymbol{q}) = \sum_{i=1}^{n} w_i (q_i - q_i^N)^2 \\
&\text{Subject to: } distance = \left\| \boldsymbol{x}(\boldsymbol{q})^{end-effector} - \boldsymbol{x}^{target\ point} \right\| \leq \varepsilon \\
&\qquad\qquad q_i^L \leq q_i \leq q_i^U; i = 1,2, \dots, n
\end{aligned} \tag{1}$$

where ε is a small positive number that approximates zero and DOF is the total number of degrees of freedom. (1) is solved using the software SNOPT (Gill et al., 2002), which uses a gradient-based method. Thus, analytical gradients are determined for all objective functions and for all constraints.

SPHERE-FILLING ALGORITHM

The optimization formulation for posture prediction can be extended to include constraints that prevent an avatar from intersecting objects in the environment; however, these constraints must be continuous, differentiable functions. Objects in the 3D environment are internally stored as sets of primitive polygons, but attempting to formulate constraints that use these primitives would prove to be computationally intractable, especially when striving for real-time performance. The proposed approach represents an object in the environment with a set of spheres, such that the union of the sphere volumes approximates the shape of the object. The sphere representation allows for a relatively simple constraint formulation. In addition, it allows for variable fidelity, depending on the requirements of the problem being solved. Sphere filling algorithms are numerous, each with advantages and disadvantages, and many different techniques have been tested in the context of the above-mentioned posture prediction problem.

The current approach for representing surrogate geometry is sphere shelling (Johnson et al., 2009), which generates a large number of spheres that cover the surface of the object. Sphere shelling is fast but generates a large number of spheres, which greatly increases the runtime of posture prediction. Another disadvantage of sphere shelling is that it overestimates the shape of the object, as the spheres are on the surface and are not necessarily contained inside the object's surface. This overestimation prevents the digital human from getting close to object edges and occasionally creates an infeasible problem where conceptually there should be a solution. Thus, an advanced sphere filling algorithm has been implemented that specializes in generating efficient representations of geometry. This adaptive medial-axis approximation is based on three-dimensional Voronoi diagrams (Bradshaw et al., 2004) and generates close representations of objects with as few spheres as possible. The drawback of this medial-axis algorithm is that it takes a relatively long time to generate spheres, even for very simple meshes. Consequently, a two-step hybrid sphere-filling algorithm was created that utilizes sphere inflation and sphere culling, described as follows.

Here, we describe the first stage of the hybrid approach. This method uses a grid of points (voxels) to create the initial spheres, as with sphere-shelling, but then expands spheres inside the object until they touch the edge of the mesh. The positioning of the spheres may not be as optimal as with the medial-axis method, but the slight loss of representation efficiency is compensated for with the speed increase. The following pseudo-code describes this inflation method.

inflate(K: *mesh*, M: *integer*)
VOXELS: *grid* ← impose M×M×M grid over bounding box of K
SPHERES: *sphere set* ← ∅

for each P: *point* **in** VOXELS
DIST: *float* ← signed distance from P to closest triangle in K (Bærentzen & Aanæs, 2002)
if (DIST < 0)
// The voxel is inside the mesh
SPHERES ← SPHERES ∪ { **sphere**(P,DIST) }

return SPHERES

This sphere inflating algorithm provides a set of spheres, the centers of which are placed on a Cartesian grid. As seen in FIGURE 2**Error! Reference source not found.**, there can be a large number of spheres, many of them redundant. Thus, the second phase of the hybrid approach involves culling redundant spheres.

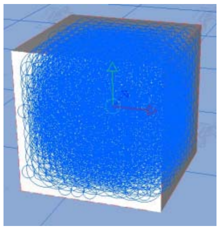

FIGURE 2 A sphere-filled cube before culling

When culling spheres, the algorithm ensures that the center of each voxel is filled (remains contained within at least one sphere). A greedy algorithm is used to select the final set of spheres. During each iteration the algorithm chooses the sphere that includes the highest number of previously unfilled voxels, and adds it to the final set of spheres. This is repeated until all voxels are filled. Various intermediate results for culling a cube are shown in FIGURE 3.

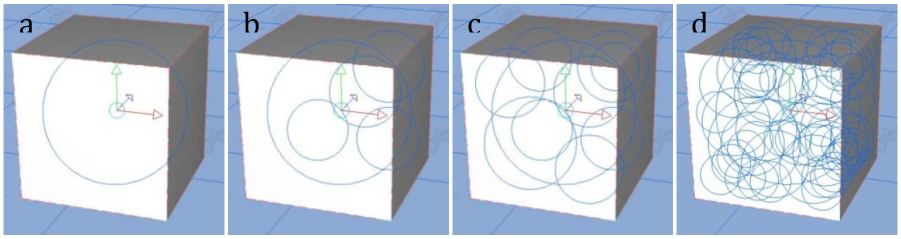

FIGURE 3 The first 1 (a), 5 (b), and 10 (c) spheres selected, and the final sphere set (d)

COLLISION AVOIDANCE ALGORITHM

Conceptually, collision avoidance is modeled by adding an additional constraint to the optimization problem (see equation (1)) for every pair of spheres that should not intersect. The use of spheres as surrogate geometry greatly increases the simplicity of calculating the constraint functions and their gradients. Given an obstacle sphere O with a global position and a body sphere B with joint-relative position, the constraint function preventing them from intersecting is given as:

$$f(\boldsymbol{q}) = \boldsymbol{position}(O) \cdot \boldsymbol{position}(B, \boldsymbol{q}) - \big(radius(O) - radius(B)\big)^2 \geq 0$$

Similarly, given two body spheres (approximating the avatar mesh) B_1 and B_2, with positions given locally, the constraint function preventing their intersection is:

$$f(\boldsymbol{q}) = \boldsymbol{position}(B_1, \boldsymbol{q}) \cdot \boldsymbol{position}(B_2, \boldsymbol{q}) - \big(radius(B_1) - radius(B_2)\big)^2 \geq 0$$

With a basic implementation, one constraint is added to the optimization formulation for every pair of spheres that should not collide. Thus, for m body spheres and n obstacle spheres, $m \times n$ constraints are added for obstacle avoidance. Thus, the optimization running time is at best linear in the number of constraints, or $\Omega(MN)$. In many real-world simulations, it is not uncommon to have thousands of obstacle spheres present in the environment, which can greatly increase the time required to find a solution. Consequently, a multi-run collision avoidance method has been implemented, in which the optimizer is executed in a loop, with each iteration running posture prediction, performing collision detection, and then either adding new necessary constraints to the problem, or returning a satisfactory result (Johnson et al., 2009). This approach requires multiple executions of the optimizer, but it ultimately considers fewer constraints than the basic implementation.

Here, we outline how the multi-run approach is used for self-avoidance, where spheres representing an avatar are restricted from colliding with other spheres in the avatar. This multi-run approach is especially helpful at reducing optimization constraints, because with the basic implementation, the number of constraints required for M body spheres is $O(M^2)$. Thus, implementing the multi-run approach reduces constraints and allows for an increased number of defined body spheres.

One source of difficulty with self-avoidance is that there are certain body-sphere pairs that should never be constrained from colliding. For example, the body spheres that represent the avatar’s right forearm need not be constrained from intersecting spheres in the right hand, but they should not collide with spheres that represent the avatar’s torso. To handle this, a new grouping approach has been incorporated, whereby the body spheres are grouped based on their physical location and spheres in the same group are not checked for avoidance. In addition, there are specific pairs of body spheres that may reside in different groups but should still not be constrained to avoid one another. The user has the ability to alter the grouping, and this provides significant flexibility in tailoring speed and precision.

A new multi-run approach that incorporates these considerations for self-avoidance is shown in following pseudo-code:

```
avoid( BODY: body spheres ) : posture
    ENABLED: sphere pairs ← ∅
    DISABLED: sphere pairs ← ∅

    for all {X, Y} ∈ 𝒫(BODY)
        if ( group(X) ≠ group(Y) ∧ ignore ( X, Y) = false )
            DISABLED ← DISABLED ∪ {X, Y}

    do
        RESULT: posture ← optimizer_solve( ENABLED )
        NEW_COLLISIONS: boolean ← false

        for each  {X: body sphere, Y: body sphere} in DISABLED
            D: float ← |position( X, RESULT )−position( Y, RESULT )|
            if ( D < radius( X ) + radius( Y ) ) then
                NEW_COLLISIONS ← true
                DISABLED ← DISABLED \ {(X, Y)}
                ENABLED ← ENABLED ∪ {(X, Y)}
    while ( NEW_COLLISIONS = true)

    return RESULT
```

RESULTS

The performance benefits of the multi-run approach have already been discussed in a previous paper by Johnson et al., so this section takes the algorithmic advancements discussed thus far and shows their application to biomechanics, design, and analysis. The first two examples display the merits of the new sphere-filling algorithm and independently demonstrates the effects of self avoidance and obstacle avoidance on posture prediction results.

Figure 4 demonstrates a reaching task where the avatar is instructed to touch his seat belt buckle with his left hand. The resulting posture is shown with self-avoidance turned off (Figure 4.b) and with self avoidance enabled (Figure 4.c). The joint displacement objective function was used in this example, with the neutral posture shown in Figure 4.a. The numbers indicate the relative objective function value for the various postures. These numbers show that Figure 4.b is a better posture according to the objective function, but with the self avoidance constraints enabled, the avatar is forced to increase the objective value to avoid collisions.

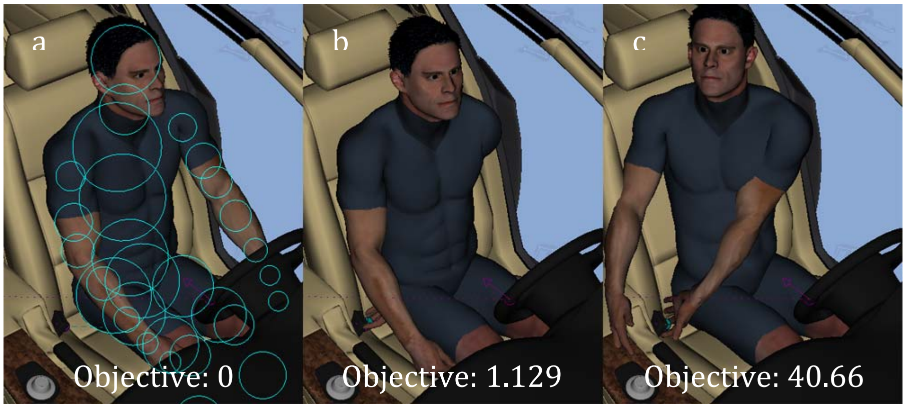

FIGURE 4 Posture prediction results with b. self-avoidance disabled and c. self-avoidance enabled using the neutral posture and body spheres in a.

The second example, shown in Figure 5, demonstates the new sphere filling method and its use with obstacle avoidance. Figure 5.a shows a steering wheel represented filled with 1206 spheres by the sphere-shelling algorithm. Figure 5.b shows the same steering wheel filled with 185 spheres generated by the inflate method. Figure 5.c and d show a posture prediction task where the avatar is reaching to the right of the steering column first without obstacle avoidance and then with obstacle avoidance. The obstacle avoidance in Figure 5.d uses the spheres from the inflate method (Figure 5.b).

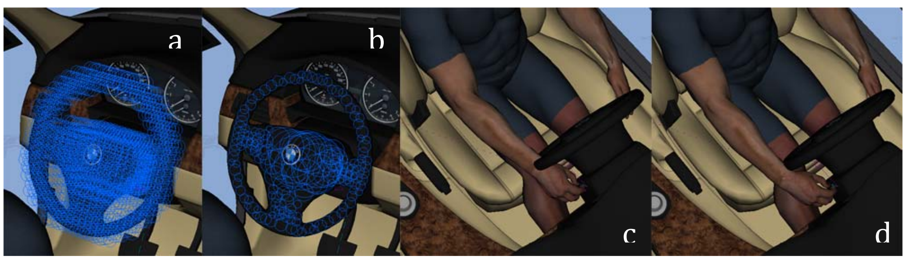

FIGURE 5 Obstacle spheres a. generated by shelling method and b. generated by inflation method. Posture prediction results c. with obstacle avoidance disabled and d. obstacle avoidance enabIde using spheres from b.

The third example, shown in Figure 6, shows a predicted posture on a motorcycle. The avatar Is constrained to touch his right knee with his left hand. Without collision avoidance, the avatar's wrist intersects with the motorcycle gas tank, as shown in Figure 6.a. With collision avoidance and self-avoidance, the avatar avoids the collision with the motorcycle, and he also avoids colliding with himself, as shown in Figure 6.b.

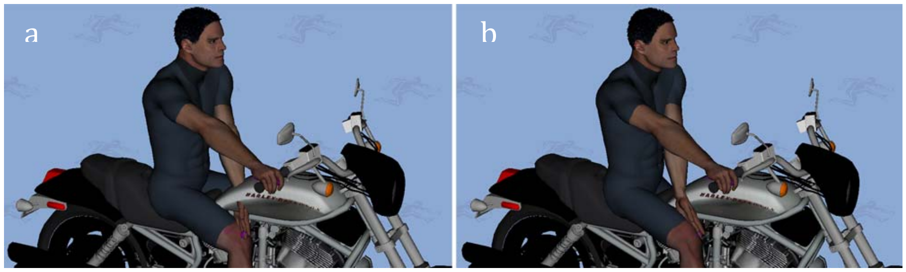

FIGURE 6 Posture prediction with self-avoidance and obstacle avoidance a. disabled and b. enabled.

CONCLUSION

Using optimization-based posture prediction as a foundation, this paper has presented novel advances with collision avoidance, which is a critical advantage of predictive DHM capabilities. Real-time collision avoidance provides one more way in which a user can alter a virtual environment on the fly and see the effects on human performance. A new multi-run approach to obstacle avoidance has been extended to self avoidance. In addition, a new method for developing surrogate geometry has been developed and tested in conjunction with a grouping method for culling surrogate geometry used to represent avatars. The results, which are demonstrated in the context of automotive and motorcycle ergonomic analysis, are quite successful.

The presented method for sphere filling has potential applications that extend far beyond posture prediction. For instance, collision detection is another critical component of virtual modeling and simulation and also requires fast and accurate creation of surrogate geometry. Thus, the proposed sphere-filling method is also used in an algorithm for detecting collisions between the avatar and geometry, when collision avoidance is turned off. This provides users with an indication of geometry that restricts motion and thus area of focus for potential design changes.

With respect to future work, objective validation using motion capture is ongoing, to verify the accuracy of the predicted postures. In addition, it is possible to predict postures, not just of body segments, but of body location and orientation. This capability will be tested with collision avoidance as well. Finally, the ability will be developed to fill geometry automatically as it is loaded. Then, only those spheres within an avatar's immediate reach envelope will be considered for avoidance.

REFERENCES

Abdel-Malek, K., Yang, J., Marler, T., Beck, S., Mathai, A., Zhou, X., Patrick, A., Arora, J. (2006), “Towards a New Generation of Virtual Humans,” *International Journal of Human Factors Modelling & Simulation*, 1(1), 2-39.

Bærentzen, J. A., Aanæs H. (2002), “Generating Signed Distance Fields From Triangle Meshes,” Technical Report IMM-TR-2002-21, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Meshes Bradshaw, G., O’Sullivan, C. (2004), “Adaptive Medial-Axis Approximation for Sphere-Tree Construction,” *ACM Transactions on Graphics*, 23(1), 1-26.

Denavit, J., Hartenberg, R. S. (1955), “A Kinematic Notation for Lower-pair Mechanisms Based on Matrices.” *Journal of Applied Mechanics*, 77, 215-221.

Farrell, K., Marler, R. T., Abdel-Malek, K. (2005), "Modeling Dual-Arm Coordination for Posture: An Optimization-Based Approach," *SAE 2005 Transactions Journal of Passenger Cars - Mechanical Systems*, 114-6, 2891, SAE paper number 2005-01-2686.

Gill, P., Murray, W., Saunders, A. (2002), “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization.” *SIAM Journal of Optimization*, 12(4), 97-1006.

Johnson, R., Smith, B. L., Penmatsa, R., Marler, T., Abdel-Malek, K. (2009), “Real-Time Obstacle Avoidance for Posture Prediction,” *SAE Digital Human Modeling Conference*, June, Gothenburg, Sweden, Society of Automotive Engineers, Warrendale, PA.

Liu, Q., Marler, T., Yang, J., Kim, H. J., Harrison, C. (2009), “Posture Prediction with External Loads – A Pilot Study,” *SAE 2009 World Congress*, April, Detroit, MI, Society of Automotive Engineers, Warrendale, PA.

Yang, J., Marler, T., Beck, S., Abdel-Malek, K., 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.

Marler, R. T., Arora, J. S., Yang, J., Kim, H. J., and Abdel-Malek, K., (2009), “Use of Multi-objective Optimization for Digital Human Posture Prediction,” *Engineering Optimization*. 41(10), 925-943.

Marler, T., Arora, J., Beck, S., Lu, J., Mathai, A., Patrick, A., Swan, C. (2008), “Computational Approaches in DHM,” in *Handbook of Digital Human Modeling for Human Factors and Ergonomics*, Vincent G. Duffy, Ed., Taylor and Francis Press, London, England.

Marler, T., Yang, J., Rahmatalla, S., Abdel-Malek, K., 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.

ACKNOWLEDGEMENTS

This work has been partially funded by Caterpillar Inc. project: Digital Human Modeling and Simulation for Safety and Serviceability. This support is gratefully acknowledged.