Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Novel Curvilinear Collision Avoidance Scheme for Interacting Avatars in a Virtual Environment

Ziad Sakr(1), Chanan Sudama(1), Marc Doumit(2)
(1)School of Information and Communication Technology, University of Trinidad and Tobago, Trinidad and Tobago
(2)School of Information Technology and Engineering, University of Ottawa, Canada

***Abstract*— As the number of Interactive Virtual Environments over the Internet for gaming or training purposes has increased dramatically in recent years, there has been the need for making these environments appear more natural and real. Avatar locomotion is an important contributor to improving the realism of interactive 3D virtual environments. This paper presents a novel collision detection and avoidance algorithm that prevents avatars from colliding with each other or obstacles in their bath while moving in a linear as well as in a curvilinear motion with changing velocities.**

***Keywords*—avatar, collaborative virtual environments, collision detection, curvilinear motion**

I. Introduction

With the growing popularity of online gaming, social networking, training, and entertainment over collaborative virtual environments, has led to the research demand for making the user's presence and participation in these environments appear more natural and real. Avatar locomotion is an important contributor to improving the realism of collaborative virtual environments (CVEs). One of the interests of many researchers is investigating the locomotion characteristics of Avatars in a CVE, with a primary focus on avatar collision detection and avoidance in virtual scenes [6][12].

Many collision management solutions for avatars in CVEs have been proposed. Cohen et al [5], Patrick et al. [7], Wilson et al. [9] and Xiao et al. [10] – all proposing techniques to improve collision detection in virtual environments. Some of these include direction indicators, maps, path restrictions, and artificial force field algorithms. These algorithms though, focused on a collision detection of an avatar with static objects in its path and not with other moving avatars present in the same virtual scene.

A more recent Avatar-Avatar collision detection scheme, AACD [13], uses the Voronoi theory [4], to provide an effective linear solution for collision-free movement of an Avatar in a CVE. AACD however, has as a major limitation, where in order for two Avatars to effectively exercise horizontal collision management, they must be in direct line of sight with each other.

This paper extends the AACD scheme by proposing a Curvilinear Avatar with Avatar Collision Detection scheme (CAACD) to achieve a collision-free collaborative virtual environment for avatars going in a linear as well as curvilinear motion [3] with changing velocity.

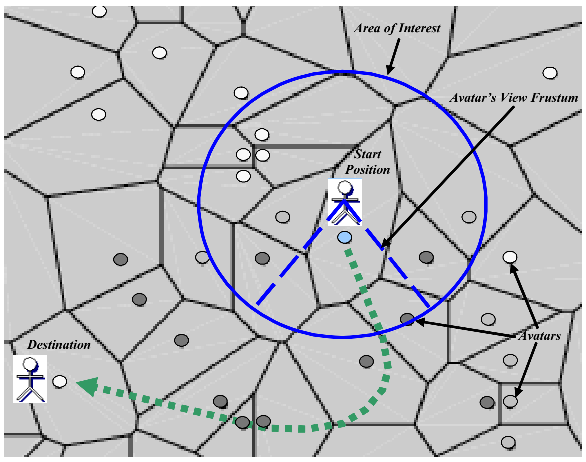


Figure 1. Avatar's motion in a virtual scene

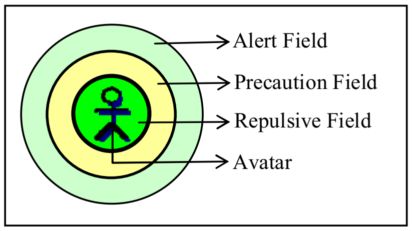


Figure 2. Avatar representation in CAACD

The rest of this paper is organized as follows. Section 2 lists the design concepts for the CAACD. Section 3 explains the CAACD algorithm. The CAACD architecture design is covered in section 4. Some of the simulation results are revealed in section 5. Section 6 concludes the paper with a discussion of future works.

II. CAACD Design Concepts

The Avatar-Avatar collision avoidance scheme, AACD, uses the Voronoi theory, which provides an efficient, componentized mechanism, modeled around real world human collision avoidance behavior [13]. The concept of Voronoi theory adopted in the AACD model is based on the

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

partitioning of a plane with multiple points into either convex polygons, swept volumes or bounding spheres. Each partition in the plane contains exactly one generating point and every point within such partition is closer to its generating point compared to any other generating point in that plane. For example, from Fig. 1 it can be seen that each avatar represents a generating point housed within its own convex polygon. While noticeably, each avatar is not in the "ideal" center of its polygon, it is in an optimal, strategically located position, within its own partition relative to the location of all other avatars in the plane.

The decision on whether to use convex polygons, swept volumes or bounding spheres in our CAACD model was based on the following analysis.

Convex polygons are able to deliver a relatively high degree of Voronoi boundary and generating point accuracy per Voronoi region (partition). However, computational intensity is relatively high and number crunching processes are time consuming, requiring expensive high-end hardware and software resources for large-scale implementation [2][11].

Swept volumes present an underlying formulation for characterizing the volume generated as a result of the motion of a geometric entity in space. However, this is currently a hot spot research area of study with researchers focusing on reducing the calculation of properties associated with a resulting motion sweep [1]. Algorithms to compute and visualize swept volume boundaries have been formulated, but they are not fast enough for interactive applications [8].

Bounding sphere implementations on the other hand, are able to deliver a relatively acceptable degree of Voronoi boundary and generating point accuracy per Voronoi region. Computational intensity is relatively tolerable and number crunching processes are delivered in shorter time frames, utilizing standard hardware and software resources [1][2][11]. That is why the bounding sphere concepts were found to be the most appropriate and where chosen in our CAACD algorithm.

Fig. 1 depicts an avatar's movement from a starting point to its intended destination with a dotted line that passes by other avatars or entities that are either in an immediate (● = critical), or future potential collision threat (◍ = alert, ○ = tolerable).

The CAACD uses the concentric bounding sphere concept for collision management, which include the following protection fields: an Alert field, a Precaution field, and a Repulsive field, as shown in Fig. 2, and a Repulsive zone, represented in Fig. 7 by the dashed rectangles.

The Alert radius, Precaution radius and the Repulsive radius can be defined as follows: $r_{alert} = A_{alert} + B_{alert}$, $r_{pre} = A_{pre} + B_{pre}$, and $r_{rep} = A_{rep} + B_{rep}$ respectively.

If the distance between avatar A and avatar B is greater than the Alert radius, $d(A,B) > r_{alert}$, shown in Fig. 3, the Avatar state is set to *Tolerable*. If $r_{pre} < d(A,B) \leq r_{alert}$, Avatar state is set to *Alert*, as shown in Fig. 4. From Fig. 5, it can be seen

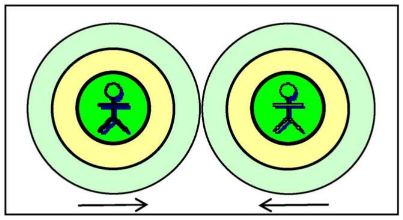


Figure 3. Avatars in a Tolerable State

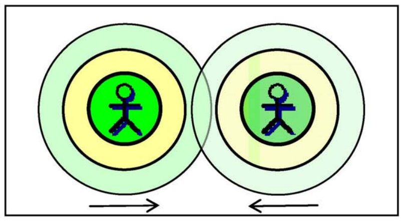


Figure 4. Avatars in an Alert State

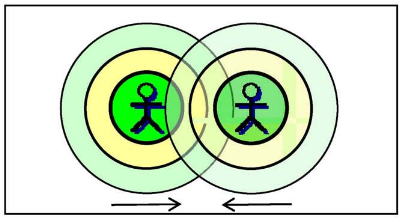


Figure 5. Avatars in a Critical State

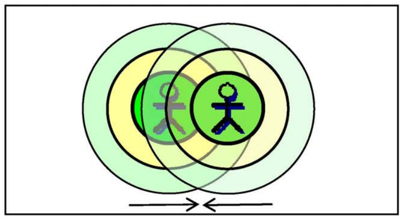


Figure 6. Avatars in a Critical State with Avatar Collision

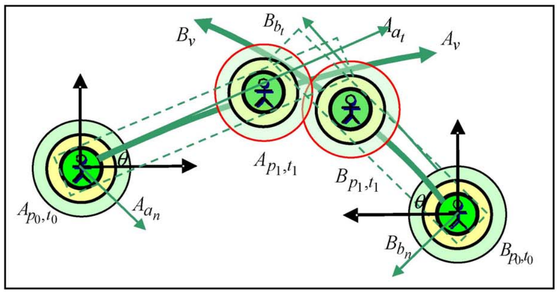


Figure 7. Avatars in motion

that if $r_{rep} < d(A,B) \leq r_{pre}$ the Avatar state is set to *Critical.* A collision occurs when the distance $d(A,B) \leq r_{rep}$, as shown in Fig. 6.

If the repulsive zones Z_1 and Z_2 of both avatars intersect at point in time P_n, t_n along the radii of their precaution fields, an avoidance event will occur to alter the elements of motion - v (velocity), θ (angular direction), a_t (acceleration in direction of velocity to change the velocity's magnitude), and a_n (centripetal acceleration perpendicular to velocity to change the velocity's direction), for either *A* or *B*, or both. Similarly, for an avatar *A* approaching an Obstacle *O*, if the repulsive zone Z_1 of *A* overlaps the precaution field of *O*, an avoidance event will occur to alter the parameters of motion for A.

Additionally, each avatar is equipped with a steering parameter and navigation probes (an avatar's projected repulsive zone). These probes acquire data, to approximate collision and avoidance factors that contribute to determining locomotion parameters before taking the next step.

III. Proposed CAACD Algorithm

The Curvilinear Avatar with Avatar Collision Detection scheme (CAACD) divides the overall collision detection into three phases: Collision Determination, Collision Avoidance and Path Amendment. Based on the distance between two Avatars, and their velocity and correlative information, Collision Determination estimates whether a potential collision exists, and generates a Collision Avoidance event if it does. Collision Avoidance amends an Avatar's navigation speed and direction to avoid a potential collision and generates a Path Amendment event. Finally, Path Amendment aims at maintaining an Avatar's original path and navigation destination to avoid unnecessary spatial loss while guaranteeing the user's spatial awareness while interacting in the scene.

A. *Collision Determination*

Each Avatar in our simulation was assigned its own target destination. These target destinations were implemented randomly such that one or more Avatars may have interest in the same target destination P_n. If two Avatars arrive at a common, or in close proximity to a destination P_n, they must negotiate which Avatar is allowed to proceed, and which must wait. In our model, a random generator is used to set Avatar target destinations such that, all destinations are set outside the radius of all Obstacle Alert fields in the scene.

As avatars *A* and *B* converge on P_n it is important to note that their Tolerable and Precaution fields will overlap before their Repulsive fields intersect at P_n. As a result, it is important to estimate the relative speeds and converging distance between *A* and *B* as they approach P_n if Avatar locomotion is to be collision free. Initially, at t_0, $A_{state}, B_{state} = Tolerable$ and $A_\tau, B_\tau = T_3$, a threshold minimum *Alert* arrival time. T_3 is a reasonable threshold time estimate since the Alert protection field has a larger radius than the repulsive or precaution field.

In the CAACD model, avatar *A*'s approximate arrival time τ is determined as a function of the relative projected distance R_d between *A* and *B* and the relative speed R_s of *B* to *A*. Therefore, at t_1, approximate convergence distance A_d between *A* and *B* is determined using the relative velocity $\vec{R}_v$ and relative position $\vec{R}_p$ locomotion components of A, B, and relative speed R_s is determined as the magnitude of $\vec{R}_v$. Therefore, if relative velocity

$$\vec{R}_v = \vec{B}_v - \vec{A}_v \tag{1}$$

then relative speed

$$R_s = |R_v| \tag{2}$$

Using the relative normalized tangential velocity

$$\vec{R}_t = \left(\frac{1}{|\vec{R}_v|}\right) * \vec{R}_v \tag{3}$$

and the relative position

$$\vec{R}_p = \vec{A}_p - \vec{B}_p \tag{4}$$

the projected distance R_d between *A* and *B* is defined as the dot product of their normalized tangential velocity and relative position, such that

$$\vec{R}_d = \vec{R}_t . \vec{R}_p \tag{5}$$

Hence, R_d relative to P_n is computed as $\vec{R}_t . \vec{R}_p = \vec{R}_{t.x} * \vec{R}_{p.x} + \vec{R}_{t.y} * \vec{R}_{p.y} + \vec{R}_{t.z} * \vec{R}_{p.z}$

With R_d and R_s known, the estimated convergence arrival time A_τ of A to B is

$$A_\tau = R_d / R_s \tag{6}$$

If $A_\tau > T_3$, $A_{state}, B_{state} = Tolerable$ and no avoidance initiatives are undertaken. However, if $0 \leq A_\tau < T_3$, the positions of all avatars in close proximity to A are computed in order to determine the avatar which poses the greatest threat to A. This proximity information coupled with arrival time estimates computed above is used to establish the possibility of collisions, and negotiate collision avoidance.

Using the estimated arrival time A_τ computed in equation (6), and the actual Avatar positions A_p, B_p, the distance B_d (distance of the Avatar with greatest collision threat to a specific Avatar) is computed and compared with

$r_{rep}, r_{pre}, r_{alert}$ to determine if B, in its current position, poses a potential *Critical* or *Alert* threat. Therefore, at t_1, if A, B current convergence positions are

$$\vec{A}_{cp} = \vec{A}_p + \left(A_\tau * \vec{A}_v\right) \tag{7}$$

and

$$\vec{B}_{cp} = \vec{B}_p + \left(A_\tau * \vec{B}_v\right) \tag{8}$$

respectively, then the relative convergence position $\vec{B}_{Rcp}$ of B relative to A is

$$\vec{B}_{Rcp} = \vec{A}_{cp} - \vec{B}_{cp} \tag{9}$$

Hence, by computing the approximate length of vector $\vec{B}_{Rcp}$, the relative convergence distance B_d of the Avatar with the greatest threat to A is:

$$B_d = \left(\vec{B}_{Rcp}\right)_{Max} * \lambda_1 + \left(\left(\vec{B}_{Rcp}\right)_{Sum} - \left(\vec{B}_{Rcp}\right)_{Max}\right) * \lambda_2 \tag{10}$$

where $\left(\vec{B}_{Rcp}\right)_{Max}$ is the coordinate (x, y or z) with the highest numeric value for vector $\vec{B}_{Rcp}$, $\left(\vec{B}_{Rcp}\right)_{Sum}$ is the sum of the coordinate values (x, y and z) for vector $\vec{B}_{Rcp}$, and λ_1, λ_2 are decimal constants whose numeric sum approximates to 1.0.

If $B_d \le r_{pre}$ a *Critical* threat is signaled. If $B_d \le r_{alert}$ an *Alert* threat is signaled. However, if $B_d > r_{alert}$ there is no immediate threat to be reported.

If a potential *Critical* or *Alert* threat is detected, the CAACD system shifts into collision avoidance mode where the best evasive collision avoidance option is determined and adopted. Avatar parameters for current and relative convergence positions, velocity, speed and directional forces are integrated in order to derive an amicable solution to a potential threat.

B. Collision Avoidance

Collision avoidance for Avatar A is managed using a combination of three directional vectors $\vec{A}_{up}, \vec{A}_{fwd}, \vec{A}_{side}$ for *upward*, *forward* and *lateral* motion respectively. These are used to influence $A's$ current velocity by injecting *upward*, *forward* and *lateral* forces, as and when required, into its current trajectory causing angular shifts in motion in a particular direction, and hence collision avoidance. Initially, all directional vector components are set to zero values: $\vec{A}_{up} = \langle 0,0,0 \rangle$, $\vec{A}_{fwd} = \langle 0,0,0 \rangle$, $\vec{A}_{side} = \langle 0,0,0 \rangle$.

LSVA is designed so that directional vector coordinate parameters computed in this step are used to influence an Avatar's trajectory during its next step. Therefore, values for $\vec{A}_{up}, \vec{A}_{fwd}, \vec{A}_{side}$ computed at t_0, are used in determining the locomotion parameters for Avatar A at t_1.

Lateral avoidance $\vec{A}_{lavoid}$ for Avatar A is computed as the multiplicative product, of the dot product of the difference of $A's$ current position and $B's$ relative convergence position in relation to $A's$ upward directional force $\vec{A}_{up}$ as revealed in equation (11).

$$\vec{A}_{lavoid} = \left(\left(\vec{A}_p - \vec{B}_{cp}\right) \cdot \left(\vec{A}_{up}\right)\right) * \vec{A}_{up} \tag{11}$$

Therefore, we compute all lateral avoidance forces $\vec{A}_{lavoid}$ as a function of an Avatar's position.

Forward avoidance vector coordinate components are also computed as a function of Avatar positions. These directional vectors can be used to either speed up, or slow down an Avatar based on its current velocity relative to a predetermined threshold $MaxSpeed$. If $A's$ current speed, taken as the magnitude of its velocity, is lower than the threshold $MaxSpeed$, $\left(\left|\vec{A}_v\right| < \left(MaxSpeed * \lambda_3\right)\right)$, $A's$ current forward directional vector is adopted as its new forward directional vector $\vec{A}_{favoid} = \vec{A}_{fwd}$.

$$\vec{A}_{favoid} = \vec{A}_{fwd} \text{ if } \left(\left|\vec{A}_v\right| < \left(MaxSpeed * \lambda_3\right)\right) \tag{12}$$

However, if $A's$ current speed is greater than the threshold $MaxSpeed$, $\left(\left|\vec{A}_v\right| > \left(MaxSpeed * \lambda_3\right)\right)$, forward avoidance becomes a function of the multiplicative product, of the dot product of the difference of A, B relative convergence positions, and $A's$ forward directional vector component such that

$$\vec{A}_{favoid} = \left(\left(\vec{A}_{cp} - \vec{B}_{cp}\right) \cdot \left(\vec{A}_{fwd}\right)\right) * \vec{A}_{fwd} \quad \text{if } \left(\left|\vec{A}_v\right| > \left(MaxSpeed * \lambda_3\right)\right) \tag{13}$$

After finalizing directional vector components, $\vec{A}_{lavoid}, \vec{A}_{favoid}$ for lateral and forward avoidance respectively, a combined navigational steering force $\vec{A}_{steer}$ is applied to $A's$ velocity $\vec{A}_v$ prior to A taking its next locomotion step. Therefore if

$$\vec{A}_{steer} = \vec{A}_{lavoid} + \vec{A}_{favoid} \tag{14}$$

then $A's$ new velocity

$$\vec{A}_v = \vec{A}_v + \vec{A}_{steer} \tag{15}$$

The resultant $\vec{A}_{steer}$ force is designed to skew the original trajectory taken by Avatar A, away from potential threats, as it navigates towards its target destination.

C. Path Amendment

Destination parameters are introduced in the path amendment model so that regardless of the avoidance, or penetration constraint action taken, the optimum path for the next step of the journey is determined. Hence, consideration is now given to $A's$ target destination, relative to its current locomotion parameters. For every target destination a threshold deceleration distance ψ_{decel} is defined. Hence, Avatar A is constantly decelerated until its Repulsive field origin completely overlaps its corresponding target destination origin. At this point $A's$ velocity is then set to zero.

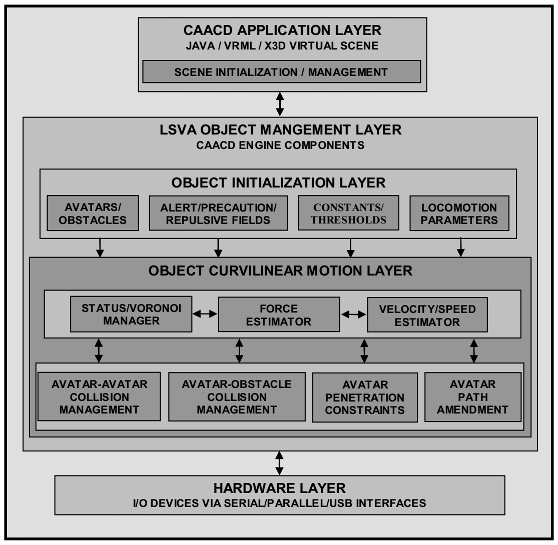
IV. CAACD Architecture Design

The CAACD is implemented as a partitioned, layered architecture with three (3) major layer components. These include the CAACD Application Layer (Layer 1), the Local Space Vector Analysis (LSVA) Object Management Layer (Layer 2), and the Hardware Layer (Layer 3), as displayed in Fig. 8.

The application layer presents the User Interface components, required for user interaction with the model. The controls on the user interface allow users to manipulate or adjust avatar locomotion parameters such as Mass, Force and Speed prior to launching an instance of the simulation. This allows the selection of the optimum parameter combination, for collision free Avatar locomotion to be determined, thereby ensuring maximum navigation efficiency and effectiveness for all Entities in the scene.

The LSVA Object Management layer contains two sub-layers; the Object Initialization Layer and the Object Curvilinear Motion Layer. The Object Initialization Layer ensures that all entities, protection fields, threshold constants and locomotion parameters are created and initialized, in preparation for scene interaction. Once initialization is complete, the Object Curvilinear Motion Layer activates and takes responsibility for the real time management of all scene Entities, as Avatars navigate around Obstacles and other Avatars. Avatar velocity, position, directional forces, status, and protection states are all monitored and regulated in this layer.

No hardware devices were implemented in our simulations. Therefore, instead of user controlled Avatar locomotion and target destinations via input devices, the simulation randomly generates an Avatar's initial locomotion parameters, and its target destination for each Avatar participating in the scene. However, Human Interface Devices such as joysticks and game pads can be easily integrated into the simulation for user interaction convenience.



CAACD APPLICATION LAYER
JAVA / VRML / X3D VIRTUAL SCENE
SCENE INITIALIZATION / MANAGEMENT
LSVA OBJECT MANGEMENT LAYER
CAACD ENGINE COMPONENTS
OBJECT INITIALIZATION LAYER
AVATARS/ OBSTACLES
ALERT/PRECAUTION/ REPULSIVE FIELDS
CONSTANTS/ THRESHOLDS
LOCOMOTION PARAMETERS
OBJECT CURVILINEAR MOTION LAYER
STATUS/VORONOI MANAGER
FORCE ESTIMATOR
VELOCITY/SPEED ESTIMATOR
AVATAR-AVATAR COLLISION MANAGEMENT
AVATAR-OBSTACLE COLLISION MANAGEMENT
AVATAR PENETRATION CONSTRAINTS
AVATAR PATH AMENDMENT
HARDWARE LAYER
I/O DEVICES VIA SERIAL/PARALLEL/USB INTERFACES

Figure 8. Architecture of CAACD

V. Simulation Results

A number of simulations were conducted on a two-dimensional (2D) space, as shown in Fig. 9 by varying the number of Avatars, ranging from five (5) to one hundred (100). Once activated, Avatars navigated the virtual space randomly while attempting to avoid fixed obstacles as well as each other. Many simulations were conducted and with each one, locomotion parameters including *Mass*, *MaxSpeed* and *MaxForce* were manipulated to determine the parameter combination that would yield the optimum result for minimum collisions.

Obstacles in the scene were represented by grey circles of varying sizes. The concentric inner and outer circles around an obstacle represent the obstacle's repulsive and precaution fields respectively. An obstacle's position is fixed in the virtual scene for the duration of the simulation. Avatars in the scene were represented by mobile small circles of equal sizes, as shown in Fig. 9. The concentric inner, middle, and outer circles around an avatar represent the avatar's repulsive, precaution, and alert fields respectively. The short straight line and the white parallel lines attached to an Avatar, represent the current navigation bearing and Repulsive zone respectively for the current locomotion of an Avatar. Each intended destination in the scene was represented by a black target symbol. Those targets were selected randomly and every avatar is aware of its target and must follow a path to reach it.

Locomotion parameters used in documenting the results of the various simulations were structured as *Mass/MaxSpeed/MaxForce* respectively. The individual parameter values varied depending on the parameter being tested in the current simulation. The range of values tested for each parameter are listed as follows: $1.0F \le Mass \le 2.0F$, $0.4F \le MaxSpeed \le 1.2F$ and $0.02F \le MaxForce \le 0.20F$, with F represent Float Units.

The locomotion parameters 1.0/1.0/0.06 and 1.0/0.8/0.06, for Mass/MaxSpeed/MaxForce respectively, produced a relatively high number of collisions when compared to the all other locomotion parameter combinations assessed in this simulation.. Generally for these parameters, the number of collisions was ≤ 2 for ≤ 10 Avatars, ≤ 4 for ≤ 30 Avatars, ≤ 8 for ≤ 40 Avatars, and ≤ 11 for ≤ 50 Avatars. However, the number of collisions increased significantly to a maximum of ≤ 109 when the number of Avatars ranged from 50-100 as depicted in Fig. 10.

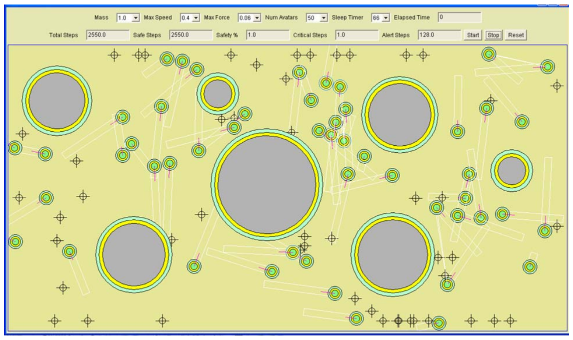

Mass 1.0 Max Speed 0.4 Max Force 0.06 Num Avatars 50 Sleep Timer 66 Elapsed Time 0
Total Steps 2550.0 Safe Steps 2550.0 Safety % 1.0 Critical Steps 1.0 Alert Steps 128.0

Figure 9. Simulation evironment with obstacles and avatars

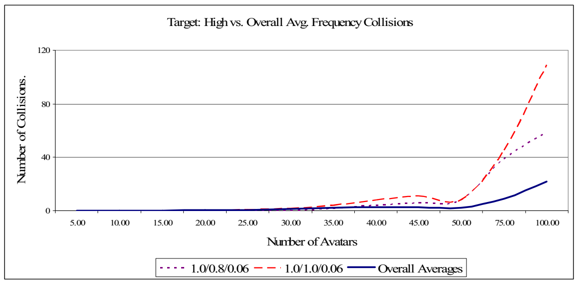


Figure 10. Target high vs. Overall avg. frequency collisions

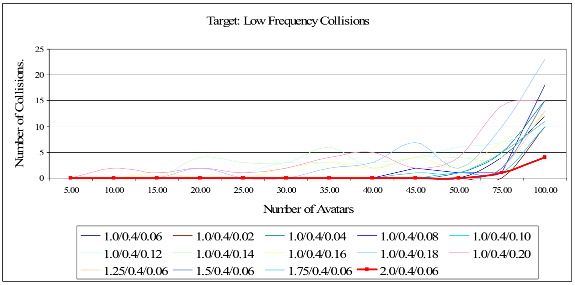


Figure 11. Target low frequency collisions

However, after removing the locomotion parameters 1.0/1.0/0.06 and 1.0/0.8/0.06 from the result set of the Target Curvilinear Motion simulation, the remaining combinations of *Mass/MaxSpeed/MaxForce* demonstrated results indicating a relatively Lower Frequency (LF) of Avatar collisions. Generally for these parameters, the number of collisions was ≤ 2 for ≤ 10 Avatars, ≤ 4 for ≤ 30 Avatars, ≤ 6 for ≤ 40 Avatars and ≤ 7 for ≤ 50 Avatars. However, the number of collisions increased to 23 when the number of Avatars ranged from 50-100 as depicted in Fig. 11. . The bold line, shown in Fig. 11, indicated the most effective locomotion parameter combination of 2.0/0.4/0.06, which across all parameter combinations investigated in the Target simulation, generated the least collision situations.

From these simulations, one can observe that the CAACD algorithm was able to provide collision avoidance capability for a large number of avatars within a confined space. But as the speed of each avatar and the number of avatars increase within that limited space, the number of collisions generated will naturally increase. Therefore to minimize the number of collisions generated by avatars, the *MaxSpeed* threshold and the number of avatars allowed within a confined space should be set appropriately.

VI. Conclusion

The While the AACD algorithm did present a collision detection scheme for moving avatars, it took into consideration a horizontal collision detection scheme for avatars moving at constant velocity with a non-linear locomotion. This paper extended the work of the AACD algorithm by presenting the CAACD, which accommodates collision detection with changing speed and curvilinear motion.

Future research work will focus on extending the avatar state information to play a more integral role in the avatar collision management process by assigning several priority levels to each of the states. Hence, minimize the number of collisions in an environment with high volume of avatars.

References

[1] K. Abdel-Malek, D. Blackmore, and K. Joy, "Swept Volumes: Foundations, Perspectives, and Applications." *International Journal of Shape Modeling.*

[2] F. Aurenhammer, "Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure." *Institute fur Informationsverarbeitung Technische Universitat Graz, Sch iet!stattgasse 4a, Austria.*

[3] R.A. Best and J.P. Norton, "A New Model and Efficient Tracker for a Target with Curvilinear Motion", IEEE Trans. on Aerospace and Electronic Systems, vol. 33, issue 3, pp. 1030-1037, 1997.

[4] P. Chew, "Building Voronoi Diagrams for Convex Polygons in Linear Expected Time", Technical Report-PCS-TR90-147, 1990.

[5] J. Cohen, M. Lin, D. Manocha, M. Ponamgi, "I-Collied: An Interactive and Exact Collision Detection System for Large-scale Environment", *Symposium on Interactive 3D Graphics*, pp. 9-12, 1995.

[6] T. W. Kwon, Y. C. Choy, "A New Navigation Method in 3D VE", *6th International Conference on Virtual and Multimedia*, 2000.

[7] M. Patrick, C. Karin, F. Eddy, "Collision Avoidance and Map Construction Using Synthetic Vision", *Virtual Reality Journal*, Springer London, vol. 5, June 2000.

[8] S. Redon, Y.J. Kim, M.C. Lin, D.Manocha, and J. Templeman, "Interactive and continuous collision detection for avatars in virtual environments.", *Proc. on IEEE Virtual Reality,* pp. 117-283, March 2004.

[9] A. Wilson, E. Larsen, D. Manocha, and M. Lin, "Partitioning and Handling Massive Models for Interactive Collision Detection" *Proc. of Eurographics*, 1999.

[10] D. Xiao, R. Hubbold, "Navigation Guided by Artificial Force Fields", *CHI*, pp. 18-23, 1998.

[11] C. Yang, L. Wang, X. Wang, Y. Xu, X. Meng "A system framework and key techniques for multi-user cooperative interaction in virtual museum based on Voronoi diagram.", *Proc. of the Ninth International Conference on Computer Supported Cooperative Work in Design,* pp. 966-970, vol. 2, May 2005.

[12] R. Young, "Design a Group Navigation Tool for a Collaborative Virtual Environment", Master's thesis, University of Nottingham, 1996.

[13] C. Yu, D. Ye, M. Wu, Y. Pan., "A New Horizontal Collision Detection Scheme for Avatar with Avatar in Collaborative Virtual Environment", *Proc. of the Fourth International Conf. on Machine Learning and Cybernetics*, pp. 18-21, August 2005.