Crowd Control with Swarm Intelligence

Ying-yin Lin
Ying-ping Chen

NCLab Report No. NCL-TR-2007007
March 2007

Natural Computing Laboratory (NCLab)
Department of Computer Science
National Chiao Tung University
329 Engineering Building C
1001 Ta Hsueh Road
HsinChu City 300, TAIWAN
`http://nclab.tw/`

Crowd Control with Swarm Intelligence

Ying-yin Lin and Ying-ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
{yylin, ypchen}@nclab.tw

March 30, 2007

Abstract

This paper presents a uniform conceptual model based on the particle swarm optimization (PSO) paradigm to simulate crowds in computer graphics. According to the mechanisms of PSO, each person (particle) in the crowd (swarm) can adopt the information to search a path from the initial position to the specified target (optimum) automatically. However, PSO aims to obtain the optimal solution, while the purpose of this study concentrates on the generated paths of particles. Hence, in order to generate appropriate paths of people in a crowd, we propose a method to employ the computational facilities provided in PSO. The proposed model is simple, uniform, and easy to implement. The results of simulations demonstrate that using PSO with the proposed technique can generate appropriate non-deterministic, non-colliding paths in several different scenarios, including static obstacles, moving targets, and multiple crowds.

1 Introduction

Today, virtual crowds created with the techniques of computer graphics can be oftentimes seen in movies, games, advertisements, and the like. It is applied to our daily life largely but in fact, creating virtual human beings behaving like human beings is not an easy task. Many research fields and sophisticated techniques are involved. In order to create high quality virtual human beings or animals, at least three facets must be taken into consideration [1]: The first one is appearance modeling. Lots of computer graphic techniques are developed to create a vivid human including the shapes of face and body, skin textures, hairstyle, and clothes. Appearance largely affects people to judge how much the computer creation is similar to a person or not. Then, the second facet is to make realistic, smooth, and flexible motions in any possible situation. Most existing methods for creating motions are parameter-based models with several parameters to control the motions. It is difficult to have a flexible, versatile model which can fit in all situations. Finally, realistic high-level behaviors have to be generated for the virtual human being. It is undoubtedly a difficult problem because defining what behaviors are human itself is worth a philosophical debate. To resolve the issue technically, many artificial intelligence and agent-based techniques are used to achieve the goal, while the techniques are still being improved and enhanced.

Particle swarm optimization (PSO) [2] is an optimization paradigm proposed in the field of evolutionary computation for finding the global optimum in the search space. The concept of PSO is easy to comprehend, and the mechanism is easy to implement. The ability of PSO to reach the position of the optimum creates the possibility to automatically generate non-deterministic paths of virtual human beings from one specified position to another. In this study, we focus on creating a realistic smooth and flexible motion for virtual human beings by utilizing the computational facilities provided in PSO. In particular, we present a uniform conceptual model based on particle swarm optimization (PSO) to simulate the motion of all persons in a crowd according to the analogy between a swarm and a crowd. A person can be considered as a particle, which would like to find a way to reach the best solution. The proposed framework can be used in various scenarios, including static obstacles, dynamic obstracles, moving targets, and multiple crowds.

The remainder of this paper is organized as follows. Section 2 describes the related works on the crowd control to give the readers some backgrounds. Section 3 briefly introduces swarm intelligence and the methodology of particle swarm optimization. Section 4 proposes the idea as well as the framework to utilize PSO for controlling crowds. Section 5 demonstrates the simulation results in several different scenarios. Finally, section 6 concludes this paper.

2 Related Work

Collective behavior had been studied for a long time in many different research domains but was applied to computer simulation only recently. In the field of computer graphics, Reynolds [3, 4] created a distributed behavior model to simulate the aggregate motion of the flock. Bouvier [5] presented an application specifically oriented to the visualization of urban space dedicated to transportation. Brogan [6] described an algorithm for controlling the movements of creatures that travel as a group. Still [7] developed a model to simulate the crowd as an emergent phenomenon using simulated annealing and mobile cellular automata. Helbing [8] used a model of pedestrian behavior to investigate the mechanisms of panic and jamming by uncoordinated motion in crowds.

Moreover, there are many studies on the realistic and real-time performance for crowd control. Aubel [9] used a multi-layered approach to handle muscles of varying shape, size, and characteristics and does not break in extreme skeleton poses. Tecchia [10] showed a real-time visualization system based on image-based rendering techniques for densely populated urban environments. Aubel [11] presented a hardware-independent technique that improves the display rate of animated characters by acting on the sole geometric and rendering information. Ulicny [12] defined a modular behavioral architecture of a multi-agent system allowing autonomous and scripted behavior of agents supporting variety. Treuille [13] presented a real-time crowd model based on continuum dynamics. Stylianou [14] used a flow grid to measure flow over an area and navigate the crowd.

Although there are many approaches for controlling crowds in computer graphics, only a few researchers try to use evolutionary algorithms for the purpose. Kwong [15] presented breeding experiments of dynamic swarm behavior patterns using an interactive evolutionary algorithm. Kim [16] incorporated several specifically designed mechanisms into the conventional particle swarm optimization methodology for simulating decentralized swarm agents. Instead of modifying the conventional PSO and designing different mechanisms for different issues, we propose a conceptual model to work with PSO to create a non-deterministic, non-colliding path for each agent with a uniform approach. The characteristics of PSO can make the particles act as a group and simulate crowds by these particle groups.

3 Swarm Intelligence

Swarm intelligence is a technique used in artificial intelligence, possibly first proposed by Beni and Wang [17] in 1989. It studies the collective behaviors of agents interacting in the environment. There is no centralized control to manage the agents, but all agents exchange their information to cooperate and emerge group behaviors. Many swarm intelligence systems are inspired by nature, including ant colonies, bird flocking, and fish schooling. They have been adopted in a lot of applications, such as ant colony optimization (ACO) [18], stochastic diffusion search (SDS) [19], particle swarm optimization (PSO) [2], and the like. Even for NP-hard problems, methods developed based on swarm intelligence can deliver good results. Among these swarm intelligence systems, PSO models a solution as a point on a surface and conducts continual movements in the search space.

Particle swarm optimization (PSO) is proposed by Kennedy and Eberhart [2] in 1995, inspired by the social behavior of bird flocking or fish schooling. PSO is a population-based optimization method and qualifies each potential solution as a particle. In a D-dimensional problem, a particle can be represented as

$$x = [x_1, x_2, \ldots, x_D]^T \ .$$

Each particle has a position, a velocity, and an objective value determined by the objective function. It uses the experiential and social metaphor to move toward the currently known best solution. The velocity is varied according to Equation (1):

$$\begin{aligned} v_i(t+1) &= \omega * v_i(t) \\ &\quad + c_1 * r_1 * (p_{BLS} - p_i(t)) + c_2 * r_2 * (p_{BGS} - p_i(t)) \ , \end{aligned} \tag{1}$$

where v_i and p_i are the velocity and position of the ith particle. ω is the weight for the previous velocity. p_{BLS} is the best position where this particle had been, and p_{BGS} is the overall global best position ever achieved by the swarm. c_1 and c_2 are the cognitive and social parameters, controlling the level of influence of p_{BLS} and p_{BGS} to make different movements. r_1 and r_2 are random numbers in $[0, 1]$. The stochastic scheme makes the velocity more diverse.

After computing the velocity, the position can then be updated according to Equation (2):

$$p_i(t+1) = p_i(t) + v_i(t+1) \ . \tag{2}$$

For every iteration, each particle updates its velocity and position. The new position is evaluated by the given objective function, and an objective value is assigned to the particle accordingly. Based on the objective value, p_{BLS} and p_{BGS} might be updated and have influence in the next iteration. Because the fundamental concept is quite similar to that of the movement of pedestrians, we would like to simulate a crowd with the optimization process that particles move toward the expected and currently known best solutions.

Although PSO does possess some characteristics of the crowd behavior, it is still incompatible with the use for crowd control. Firstly, the particle in PSO is absolutely free to fly through everywhere in the given multidimensional space. However, the environment for a crowd may have obstacles, and the pedestrians in the crowd must avoid collisions as shown in Figure 1, including the collision with the given obstacles and the collision with the fellow pedestrians, where other pedestrians can be considered as dynamic obstacles. These dynamic obstacles are not predictable and may appear and disappear in the environment at any moment. Moreover, it is important to make a virtual pedestrian to walk smoothly and naturally, instead of just oscillate uncertainly and strangely. The walking path must be reasonable and appropriate. For these essental reasons, we propose the model to work with the original PSO for path creation in the next section.

4 Proposed Model

In the light of the analogy between swarms and crowds, we may consider a person as a particle and a group as a swarm. In order to resolve the aforementioned incompatibility issues, we

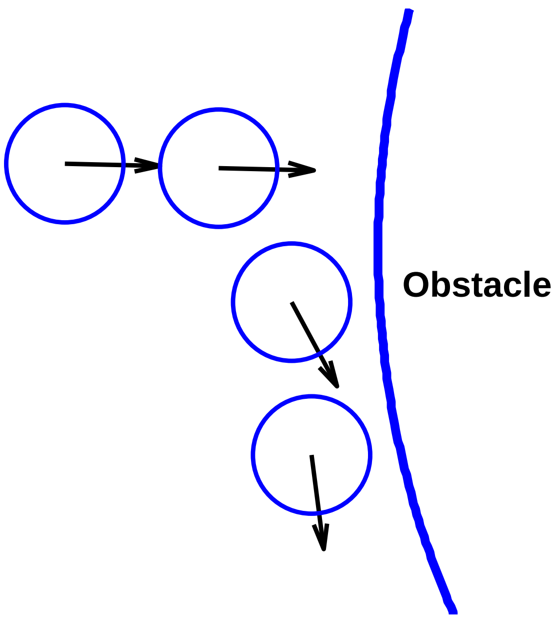


Figure 1: Virtual human beings must be capable of avoiding obstacles.

propose a framework to work with PSO such that the movement of particles can be similar to that of real people.

4.1 PSO Essence

We control the movement on the x-z plane[1], just like to have a bird's eye view on the top of people's heads. The position and the velocity can be represented by 2D vectors, such as $p_i = [p_{ix}, p_{iz}]^T$ and $v_i = [v_{ix}, v_{iz}]^T$. Although it is unnecessary, for convenience, we separate the velocity into a direction component, $D_i = [D_{ix}, D_{iz}]^T$ and a speed component, S. The speed component S models and matches the variant paces of different people and has a maximum limit. Each particle holds the information about itself, including a position, a direction, a speed, and an objective value. The direction is computed as

$$D_i(t+1) = \omega * D_i(t) + c_1 * r_1 * (p_{iBLS} - p_i(t)) + c_2 * r_2 * (p_{BGS} - p_i(t)) \ ,$$

where $(p_{iBLS} - p_i(t))$ and $(p_{BGS} - p_i(t))$ are both unit vectors for indicating the direction only. Other parameters are defined as the same parameters of the velocity in PSO.

After deciding the direction of the particle, the position is then computed as

$$p_i(t+1) = p_i(t) + S_i(t+1) * D_i(t+1)$$

where $S_i(t+1)$ is the speed component. The range of speed is $[0, V_{max}]$ and scales by the particle's objective value. The V_{max} is set by a step size with a random number. It is able to fit the different pace of each person and to make the environment more dynamic. If a particle approaches an obstacle, the speed will be slower to avoid it. The speed also decreases gradually when a particle is close to the specified goal. The mechanism can eliminate the collision issues and reduce the oscillatory situation that occurs near the optimal solution in PSO.

At each step, each particle gets an objective value to measure the current position. It is synchronous to update the local best position and the global best position based on the objective value. Such a PSO mechanism will make the particles in the group converge to the

[1]Because the y-axis represents the height in the 3D space, the ground surface is the x-z plane.

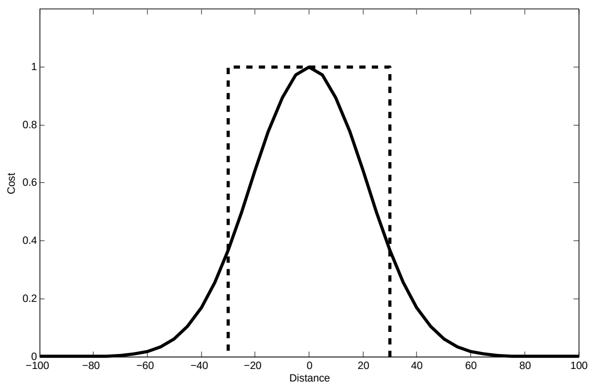


Figure 2: The exponential model for an obstacle. The box indicates the obstacle to be represented by the function.

target. Moreover, we can adopt different objective functions to arrange the final state of a group, such as a line, a circle, or other possible shapes.

4.2 Objective Function

As a matter of fact, PSO is only interested in the final state of particles and cares nothing about the particle paths at each step. It is very different from crowds in the real world. In the real world, not everywhere on the ground can be stepped on or gone through. There are obstacles, such as holes and buildings. Moreover, a person normally also cannot step on another person. These situations contribute to the incompatibility of PSO for crowd control. In order to resolve these incompatibility issues with an uniform approach, we design an objective function to represent the specified target, the static obstacle, as well as other particles.

4.2.1 Unit function as building blocks

The objective value of a particle is affected by two factors: the target and the obstacles. For the purpose to use the optimization ability of PSO, we make the target as the minimum on the x-z plane, considered as the mathematical search space. If a particle approaches the target, it should get a lower (better) objective value. All particles move toward the lower region just like water flows downward. On the other hand, the objective value raises as penalty if the particle comes close to or even touches obstacles. Since PSO used in this study for solving a minimization problem, we will view the objective value as "cost" in the remainder of this paper.

We use an exponential function to represent everything in the search space, including the target and the obstacles. The function for calculating the cost for an object p relative to an object q can be given as

$$Cost(p,q) = \exp\left(-\left(\frac{(p_x - q_x)^2}{(\sigma_{p_x}+\sigma_{q_x})^2} + \frac{(p_z - q_z)^2}{(\sigma_{p_z}+\sigma_{q_z})^2}\right)\right) , \tag{3}$$

where (q_x, q_z) is the the position the object q, and $(\sigma_{q_x}, \sigma_{q_z})$ is the scope of q. q can be a target, an obstacle, or a particle. For example, if an obstacle's area is 30 by 50, we can set $(\sigma_{q_x}, \sigma_{q_z}) = (15, 25)$. If it is the target, $(\sigma_{q_x}, \sigma_{q_z})$ is set to the whole search area. (p_x, p_z) is the position of a particle for which the cost is calculated. $(\sigma_{p_x}, \sigma_{p_z})$ is the scope in the search space occupied by p. Figure 2 shows the exponential model for an obstacle.

The proposed exponential model is similar to the 2D normal distribution with a mean and a standard deviation. The difference is that every exponential function representing an object in the system has its own volume size, while the volume size is always one for the normal distribution. The different sizes in volume make it relatively easy to model the landscape for the pedestrians to go through. Overlaying these exponential functions, a bumpy landscape with a minimum position as the target can be created. Therefore, the overall objective function proposed in this study can be described as

$$F_{obj}(p) = c_{obs} * \max_{o \in O}(Cost(p, o)) + \frac{1}{Cost(p, g)} ,$$

where O is the set of all obstacles, g is the target for the particle to reach, and c_{obs} is a constant to adjust the relative importance between the target and obstacles.

It has to be noticed that the set O contains not only all the specified obstacles, such as holes and buildings, on the landscape but also other people in the crowd. We adopt the identical model for everything in the scenario instead of develop different models for objects of different kinds. By doing so, no extra models have to be introduced into the system when new objects are included, such as moving cars or running animals. Furthermore, the proposed model induces an interesting situation for PSO. Every particle actually "sees" a different landscape due to the relative relations among particles. Such a situation does not exist in common optimization applications. Under this condition, according to the results of this study, PSO can still appropriately accomplishes its assigned task, and the particles converge to the desired goal via reasonable paths.

4.2.2 Local search for collision avoidance

Even with the carefully design objective function, the possibility for a particle to pass through an obstacle still need to be eliminated. In this study, we implement this functionality as a form of local search, which is a common operator used with PSO. When we get the cost for a new position of a particle, whether or not the new position is accepted should be checked. We use a stochastic mechanism to decide whether the new position should be accepted according to the cost. The probability to accept a newly generated position is computed by

$$Prob(f) = 1 - \frac{f}{e^{-1}} ,$$

where f is the cost for the newly generated position. The shape of the exponential function is quite appropriate for estimate whether or not a particle is too close to an obstacle. It helps the particle to avoid collisions and makes the path smoother. Moreover, there exists a hard boundary when $\sigma = 1$ according to Equation (3), because the probability will be 0 if $f = e^{-1}$. Therefore, we can theoretically verify that collisions under this checking mechanism can never happen. Figure 3 shows the probability to accept a new position around an obstacle.

If the cost is not accepted, another direction must be taken to create a smooth path. According to the original direction, a random angle less than 20 degrees is added to or subtracted from the direction vector of the particle to find an acceptable position. The better position is chosen to be the new position.

5 Moving Path Generation

With the proposed framework, we simulate several showcases on the x-z plane and show the walking paths generated by PSO in the following sections.

5.1 Crowd Simulation by using the Original PSO

We first attempt to simulate a crowd by using the original PSO. The parameter settings for PSO are $\omega = 0.8$, $c_1 = 0.4$, $c_2 = 0.4$, and the target is $(200, 0)$. The population size is 10, and

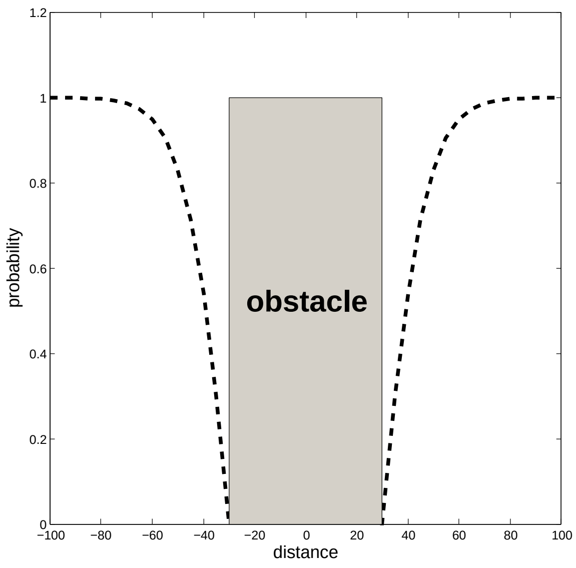


Figure 3: The probability to accept a new position around an obstacle.

the initial positions are assigned. The velocities are initialized at random. Figure 4(a) shows an influx. Each solid line indicates a path of a person. People initially are scattered and finally converge to the target. Figure 4(b) shows a stream. People on the left side move toward the right side. We can observe that they converge first and move toward the target together. For an influx or a stream, the appropriate paths can be created under the mechanism of PSO.

5.2 Crowd Simulation with Static Obstacles

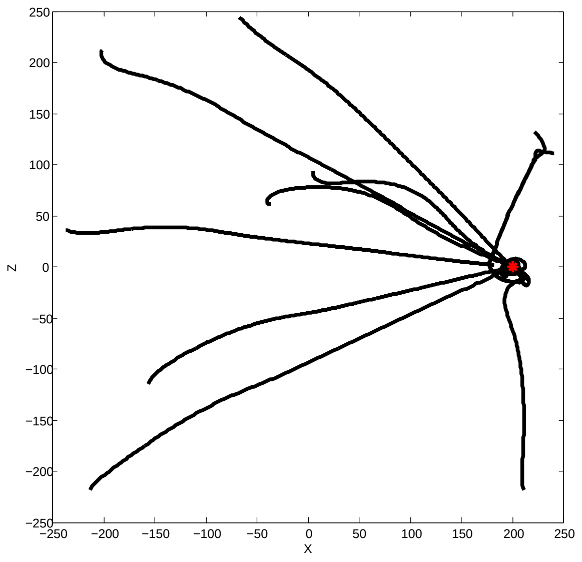
The original PSO can make all persons to reach the goal automatically, but it does not have the ability to make the particles to avoid obstacles as shown in Figure 5(a). Therefore, the proposed collision avoidance mechanism has to be employed. Figure 5(b) shows a crowd passing by a static obstacle in our system. The collisions between particles are not considered in this case. The obstacle is placed at $(0, 0)$, and its radius is 50. Based on the paths, we can see that the crowd can avoid the obstacle and reach the goal.

5.3 Crowd Simulation with Dynamic Obstacles

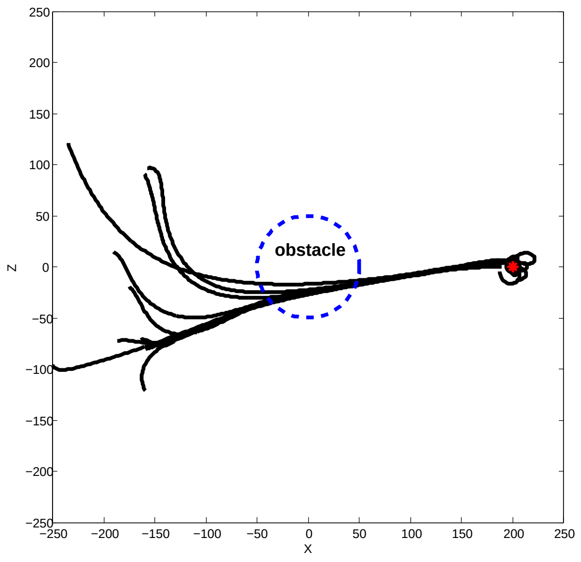
Avoiding collisions between persons in a crowd is a necessity for generating reasonable paths. In this study, we do not resort to any extra method or mechanism to solve this problem. In the proposed framework, each person is considered as an obstacle with $\sigma = 5$. If two persons come close, the cost of each person will be checked, and an acceptable position for each person will be determined. Figure 6(a) presents the paths when two persons almost collide. The dotted line is the path from A to B, and the solid line is the one from B to A. Figures 6(b) and 6(c) show the jinking situation. More people are simulated in Figure 7. Each circle represents a person, and the dotted lines are their waking paths. The simulation results demonstrate the generated paths without collisions.

5.4 Crowd Simulation with a Dynamic Target

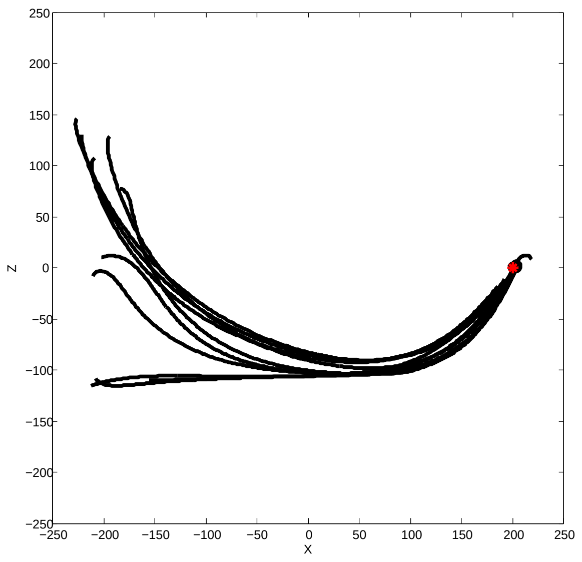
In addition to the fundamental path generation presented in the previous sections, we also conducted experiments on the crowd simulation with a dynamic target in our proposed framework. As the goal moves, the crowd is capable of following the moving target as shown in Figure 8.



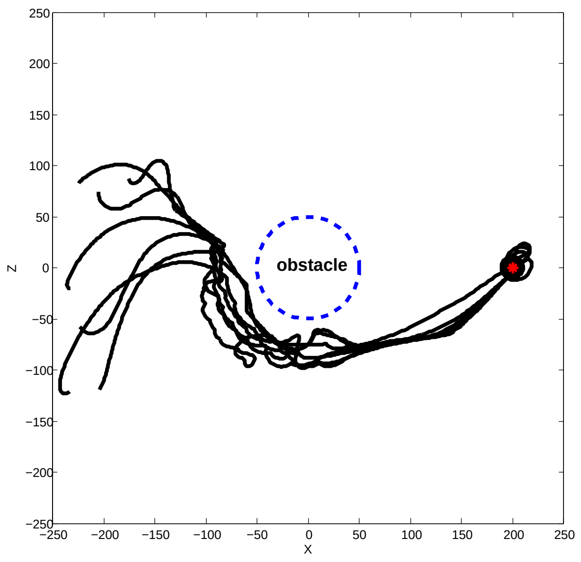
(a) Influx.



(a) The original PSO.



(b) Stream.



(b) PSO with collision avoidance.

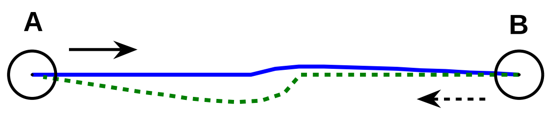
Figure 4: The paths of an influx and a stream can be simulated by the original PSO. The curves indicate the paths of people.

Figure 5: The original PSO does not guarantee to avoid obstacles. With collision avoidance, particles can go around the given obstacle.

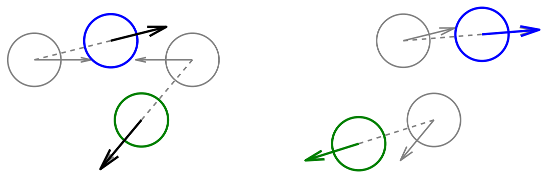
The path of the goal is designed as a circle with radius = 150. Figure 9 depicts the generated circular paths based on the moving goal.

5.5 Simulation for Multiple Crowds

Furthermore, in our framework, simulating multiple crowds going toward different targets is a trivial extension. Multiple groups can be overlaid on the same area such that more complex scenes are made possible. Figure 10 demonstrates that four groups in the four corners move toward their opposite corners. The different symbol represents the person in the different group, and each group consists of ten members. There are four obstacles with different sizes at different positions. As time goes by, each group can pass by these obstacles and reach their targets.



(a) The complete meeting paths.



(b) The first action. (c) The section action.

Figure 6: The dodge of two people with opposing directions is simulated.

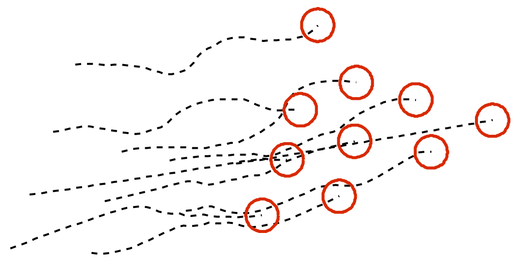
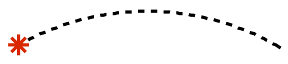


Figure 7: A walking crowd. A circle represents the of a person, and a line is the path of the person for the last 20 steps.

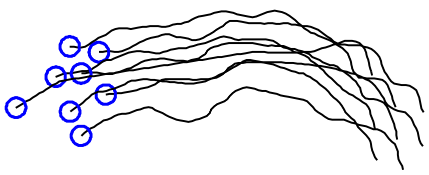
6 Conclusions

This paper proposed a uniform model to simulate crowd movements based on particle swarm optimization. We considered that people finding a walkable path to their goal as the process to find the optimal solution by PSO. The advantages of PSO are simple, fast, and easy to implement. By the PSO mechanism, each person can search for a path automatically. However, particles controlled by the original PSO may penetrate an obstacle. Hence, we developed the collision avoidance mechanism in a form of local search to work with PSO. Static obstacles, dynamic obstacles, and the target were all modeled with an exponential function. Combining these exponential functions, the scenario environment were constructed, and the particle paths were generated by the proposed framework.

The proposed method in this study is compact, coherent and controls the crowd movement



(a) A moving goal.



(b) The paths of the simulated crowd.

Figure 8: The moving goal can control a group to behave as expected.

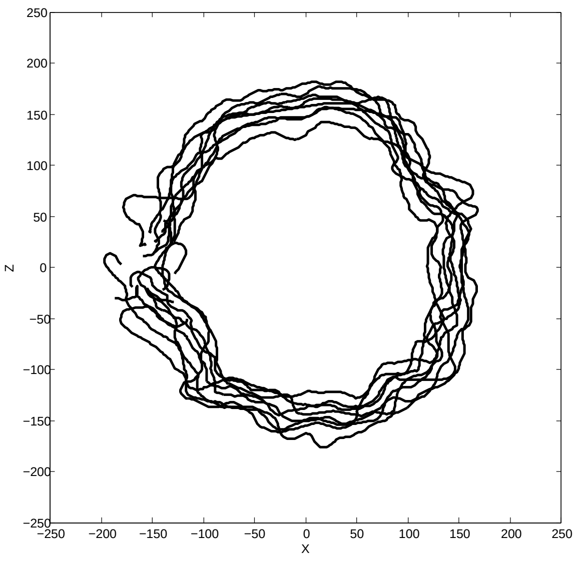
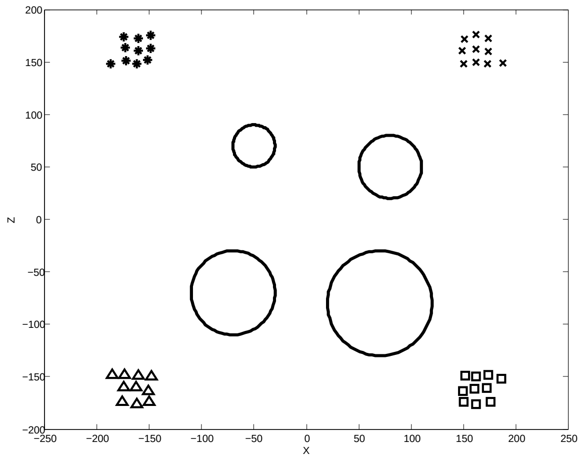
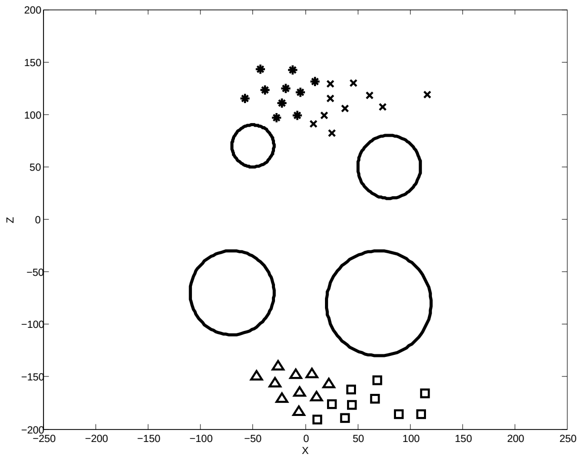


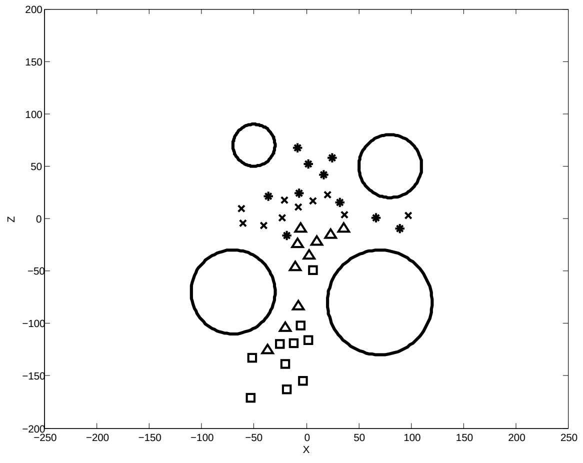
Figure 9: The paths of a crowd following a dynamic goal moving circularly.

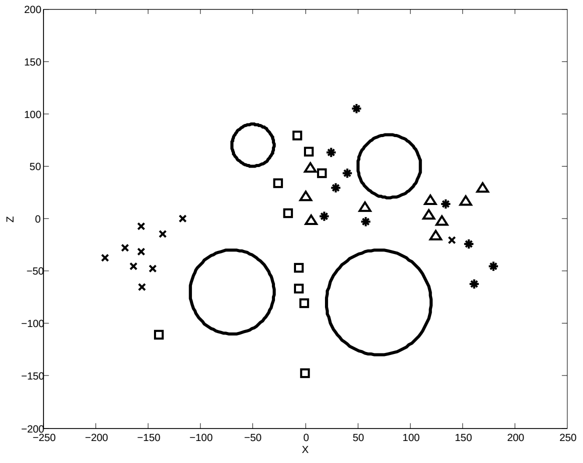
easily. Based on the uniform model, we can demonstrate a complex crowd-scape to stack up several different crowds, and the created paths are more dynamic, non-deterministic. Although this study is not the first to apply the concept of swarm intelligence on crowd control, to the best of our limited knowledge, it retains the most design of the original PSO and almost leaves PSO unmodified. The proposed model is flexible, versatile and can be used to represent a number of different kinds of objects.

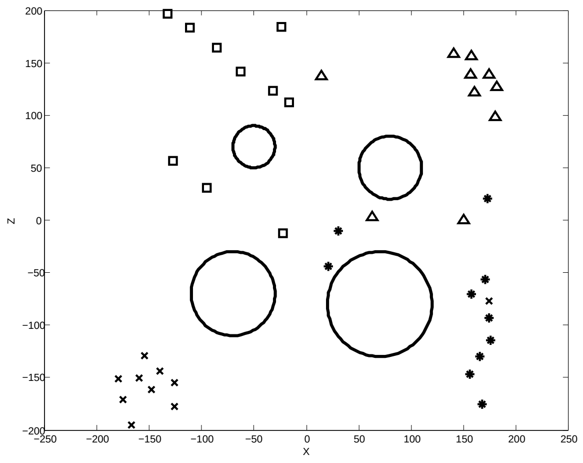
The future work includes understanding how the parameters affect the paths, determining whether functions of other classes can be employed for creating better paths, and integrating


(a) $t = 0$


(b) $t = 100$


(c) $t = 200$


(d) $t = 300$


(e) $t = 400$

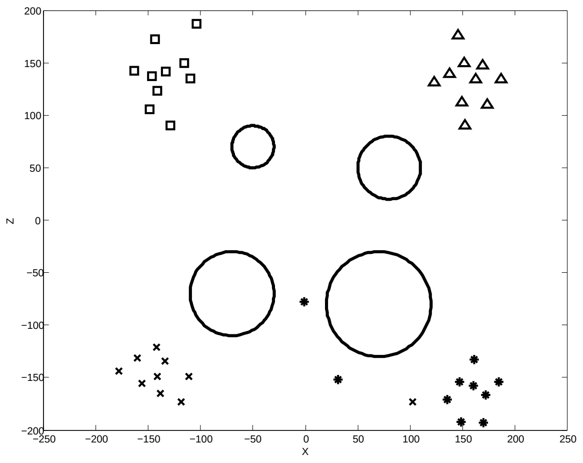

(f) $t = 500$

Figure 10: Four groups move toward the opposite corners without any collisions.

the framework into the existing computer graphics systems. Theoretical insights into the crowd behavior might also be obtained through the development of the proposed framework.

Acknowledgments

The work was partially sponsored by the National Science Council of Taiwan under grants NSC-95-2221-E-009-092 and NSC-95-2627-B-009-001 as well as by the MOE ATU Program. The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

References

[1] N. Magnenat-Thalmann and D. Thalmann, "Virtual humans: thirty years of research, what next?" *The Visual Computer*, vol. 21, no. 12, pp. 997–1015, 2005.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[3] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, July 1987.

[4] ——, "Steering behaviors for autonomous characters," in *Game Developers Conference 1999*, 1999. [Online]. Available: http://citeseer.ist.psu.edu/reynolds99steering.html

[5] E. Bouvier and P. Guilloteau, "Crowd simulation in immersive space management," in *Virtual Environments and Scientific Visualization '96*, M. Göbel, J. David, P. Slavik, and J. J. van Wijk, Eds. Springer-Verlag Wien, April 1996, pp. 104–110, virtual Enviroments '96.

[6] D. C. Brogan and J. K. Hodgins, "Group behaviors for systems with significant dynamics," *Autonomous Robots*, vol. 4, no. 1, pp. 137–153, March 1997.

[7] G. K. Still, "Crowd dynamics," Ph.D. dissertation, Warwick University, 2000.

[8] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, p. 487, 2000. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0009448

[9] A. Aubel and D. Thalmann, "Musclebuilder: a modeling tool for human anatomy," *Journal of Computer Science and Technology*, vol. 19, no. 5, pp. 585–595, September 2004.

[10] F. Tecchia and Y. Chrysanthou, "Real-time rendering of densely populated urban environments." in *Rendering Techniques*, 2000, pp. 83–88.

[11] A. Aubel, R. Boulic, and D. Thalmann, "Real-time display of virtual humans: Levels of details and impostors," *IEEE Trans Circuits Syst Video Technol, Special Issue on 3D Video Techology*, vol. 10, no. 2, pp. 207–227, 2000.

[12] B. Ulicny and D. Thalmann, "Towards interactive real-time crowd behavior simulation," *Computer Graphics Forum*, vol. 21, no. 4, pp. 767–775, December 2002.

[13] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1160–1168, 2006.

[14] S. Stylianou and Y. Chrysanthou, “Crowd self organization, streaming and short path smoothing,” *WSCG*, February 2006.

[15] H. Kwong and C. Jacob, “Evolutionary exploration of dynamic swarm behaviour,” in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC2003)*, vol. 1, 2003, pp. 367–374.

[16] D. H. Kim and S. Shin, “Self-organization of decentralized swarm agents based on modified particle swarm algorithm,” *J. Intell. Robotics Syst.*, vol. 46, no. 2, pp. 129–149, 2006.

[17] G. Beni and J. Wang, “Swarm intelligence in cellular robotics systems,” in *NATO Advanced Workshop Robots Biological System*, vol. 1, 1989, pp. 703–712.

[18] M. Dorigo, “Optimization, learning and natural algorithms,” Ph.D. dissertation, Politecnico di Milano, Italy, Italy, 1992.

[19] J. M. Bishop, “Stochastic searching networks,” in *Proceedings of the First IEE International Conference on Artificial Neural Networks (Conf. Publ. No. 313)*, 1989, pp. 329–331.