Crowd Control with Swarm Intelligence

Ying-Yin Lin

NCLab Report No. NCL-TR-2007011
June 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/

國立交通大學

多媒體工程研究所

碩 士 論 文

利用群體智慧演算法來模擬控制人群的移動

Crowd Control with Swarm Intelligence

研 究 生：林盈吟

指導教授：陳穎平　教授

中 華 民 國　九 十 六　年 六 月

利用群體智慧演算法來模擬控制人群的移動
Crowd Control with Swarm Intelligence

研 究 生：林盈吟　　Student：Ying-Yin Lin

指導教授：陳穎平　　Advisor：Ying-Ping Chen

國 立 交 通 大 學

多 媒 體 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

利用群體智慧演算法來模擬控制人群的移動

學生：林盈吟　　指導教授：陳穎平

國立交通大學多媒體工程研究所碩士班

摘　　要

本論文以粒子最佳化演算法(Particle Swarm Optimization)為原型，提出了一個統一概念的模型，來模擬電腦圖學中人群的移動。根據粒子最佳化演算法的機制，一個人物(Particle)在群體(Swarm)中，可取得周圍環境的資訊，自動地尋得一條前往特定目標(Optimum)的路徑，然而，粒子最佳化演算法傳統上用以獲取最佳解，而非如本論文著重於產生粒子的路徑，因此，為了能夠讓人群中的個體有合適的路徑，我們提出了一個方法與粒子最佳化演算法的固有功能結合。我們所提出了模型簡單、統一且容易實做，不管是有障礙物的環境、動態目標物的環境、非單一群體的模擬或是不同地形的變化，利用此模型裡，我們皆可產生合理的、非決定性的(Non-deterministic)且無碰撞(Non-colliding)的路徑。

Crowd Control with Swarm Intelligence

student：Ying-Yin Lin　　Advisors：Dr. Ying-Ping Chen

Institute of Computer Science
National Chiao Tung University

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.

誌　　謝

時光匆匆，到新竹求學已屆滿二年，能夠如此順利地完成學業，實受到不少人的照顧，在此，致上我最誠摯的謝意。

首先要感謝我的指導教授－陳穎平教授。對老師的第一印象是和善、幽默與風趣，二年來，老師並沒有給我壓力。初進校園時，老師總是鼓勵我，給予我自信；爾後，在研究的過程中，老師總是細心地指導我做研究的方法與態度，讓我能夠順利完成本篇論文。由衷的感謝老師的諄諄教誨，使我獲益良多。

感謝自然計算實驗室的學長們、同學及學弟妹。柏甫、明昌及長泰學長，謝謝你們對我的關心與指導；宏偉和秉竹，謝謝你們曾給予我的幫助；文瑜、志銘和仲堯，你們的加入，讓實驗室更顯熱鬧。因為有你們每一個人，實驗室才能常常充滿歡笑！

感謝我的摯友們。普萊里，互相打氣、一同加油的好伙伴，謝謝妳陪伴我走過這二年；墾丁，謝謝妳對我的關心，別忘了也要多多照顧自己；阿囉哈，謝謝妳帶給我的歡樂，讓我的生活增添許多笑聲。

感謝我的親友們。老爸老媽，謝謝你們對我的支持與關懷，是我持續前進的動力；老哥，謝謝你對我的照顧與幫助；新竹的伯父伯母，謝謝你們的關心與照顧，讓我在新竹也能感受到家的溫暖；高雄的阿姨和姨丈，謝謝你們的關心和鼓勵，未來我也會繼續加油的！

最後，我要感謝國立交通大學，因為有你，我才得以遇到這麼多人、事、物，感謝每一堂課的老師，感謝熱心的同學們，如今我即將踏入不同的旅程，而在此生活中的點點滴滴將銘記我心，永遠支持陪伴我前進！

Contents

摘 要 ..... **i**

ABSTRACT ..... **ii**

誌 謝 ..... **iii**

CONTENTS ..... **iv**

LIST OF FIGURES ..... **vi**

SYMBOLS ..... **vii**

I. INTRODUCTION **1**

1.1 MOTIVATION 1

1.2 GOAL 2

1.3 ORGANIZATION 2

II. BACKGROUND **3**

2.1 RELATED WORKS 3

2.2 SWARM INTELLIGENCE 4

2.3 PARTICLE SWARM OPTIMIZATION 4

2.4 SUMMARIZATION 5

III. PROPOSED MODEL **7**

3.1 PSO ESSENCE 7

3.2 OBJECTIVE FUNCTION 8

3.2.1 Cost function for a unit obstacle *9*

3.2.2 Local search for collision avoidance *11*

3.3 SUMMARIZATION 13

IV. MOVING PATH GENERATION **14**

4.1 CROWD SIMULATION BY USING THE ORIGINAL PSO 14

4.1.1 Without Obstacles *15*

4.1.2 With Obstacles *16*

4.2 CROWD SIMULATION WITH COLLISION AVOIDANCE 16

4.2.1 With Dynamic Obstacles *16*

4.2.2 With Static Obstacles *17*

4.3 OTHER CROWD SIMULATIONS 18

4.3.1 Dynamic Target *18*

4.3.2 Multiple Crowds *19*

4.3.3 Geography *19*

V. CONCLUSIONS **21**

BIBLIOGRAPHY **22**

A. *STATE FIGURES* **24**



List of Figures

FIGURE 1. A FLOW CHAT OF PARTICLE SWARM OPTIMIZATION. 6
FIGURE 2. THE EXPONENTIAL MODEL FOR AN OBSTACLE. 10
FIGURE 3. THE PROBABILITY TO ACCEPT A NEW POSITION AROUND AN OBSTACLE WITH DIFFERENT K. 11
FIGURE 4. THE LOCAL SEARCH METHODS IN OUR SYSTEM. 12
FIGURE 5. THE PATHS OF INFLUXES CAN BE SIMULATED BY THE ORIGINAL PSO. 15
FIGURE 6. THE PATHS OF STREAMS ARE GENERATED BY THE ORIGINAL PSO. 15
FIGURE 7. THE ORIGINAL PSO IS LIKE TO HAVE A COLLISION. 16
FIGURE 8. TWO STREAM CASES ARE DISPLAYED IN OUR SYSTEM. 17
FIGURE 9. THE SITUATIONS FOR COLLISION AVOIDANCE IN OUR SYSTEM ARE PRESENTED. 17
FIGURE 10. A CIRCLE CROWD AND A WAVE CROWD ARE SIMULATED. 18
FIGURE 11. THREE DIFFERENT TOPOGRAPHIES. 20
FIGURE 12. THE CONDITIONS FOR A INFLUX BY THE ORIGINAL PSO. 25
FIGURE 13. THE CONDITIONS FOR A STREAM BY THE ORIGINAL PSO. 26
FIGURE 14. THE CONDITIONS FOR COLLICTIONS BY THE ORIGINAL PSO. 27
FIGURE 15. THE CONDITIONS FOR A STREAM IN OUR SYSTEM. 28
FIGURE 16. THE CONDITIONS FOR COLLISION AVOIDANCE IN OUR SYSTEM. 29
FIGURE 17. FOUR GROUPS IN THE FOUR CORNERS MOVE TOWARD THEIR OPPOSITE CORNERS. (CASE 1) 30
FIGURE 18. FOUR GROUPS IN THE FOUR CORNERS MOVE TOWARD THEIR OPPOSITE CORNERS. (CASE 2) 31
FIGURE 19. THE CONDITION FOR A CROWD MOVES FOLLOWING A DYNAMIC TARGET. 32

Symbols

V_i	:	The velocity of the *i*th particle.
P_i	:	The position of the *i*th particle.
D_i	:	The direction of the *i*th particle.
S_i	:	The speed of the *i*th particle.
ϖ	:	The weight for the previous velocity.
P_{BLS}	:	The best position where particle had been.
P_{BGS}	:	The overall global best position ever achieved by the swarm.
c_1	:	The cognitive parameter.
c_2	:	The social parameter.
r_i	:	Random number
(Q_X, Q_Y, Q_Z)	:	The position of the object *Q*.
$(\sigma_{Q_X}, \sigma_{Q_Y}, \sigma_{Q_Z})$	:	The scope of the object *Q*.
O	:	The set of all obstacles.
g	:	The target of the particle.
c_{obs}	:	A constant to adjust the relative importance between *O* and *g*.

I. 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.

1.1 Motivation

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. On the

other hand, if the target is the best position, the movement of a person is a process to find a walkable path to the destination. For these essential reasons, we propose the model to work with the original PSO for path creation.

1.2 Goal

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 model can be used in several scenarios, including static obstacles, moving targets, multiple crowds and geography.

1.3 Organization

The remainder of this paper is organized as follows.

- Section II describes the related works on the crowd control to give the readers some backgrounds and briefly introduces swarm intelligence and the methodology of particle swarm optimization.
- Section III proposes the idea as well as the framework to utilize PSO for controlling crowds.
- Section IV demonstrates the simulation results in several different scenarios.
- Finally, Section V concludes this paper.

II. Background

Collective behavior had been studied for a long time in many different research domains but was applied to computer simulation only recently. The section digests some related works and describes Swarm Intelligent and methods of particle swarm optimization.

2.1 Related Works

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.

2.2 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.

2.3 Particle Swarm Optimization

Particle swarm optimization (PSO) is proposed by *Kennedy* and *Eberhart* 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

$$x = [x_1, x_2, \cdots, x_D]^T \tag{1}$$

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. Table 1 shows the basic structure of a particle.

Table 1. The structure of a particle.

● Position
● Vector
● Objective Value
● Best Local Solution (BLS)
● Best Global Solution (BGS)

For per iteration, each particle updates its velocity and position. The velocity is varied according to Equation (2) and the position can then be updated according to Equation (3) :

$$V_i(t+1) = \omega * V_i(t) + c_1 * r_1 * \left(P_{BLS} - P_i(t)\right) + c_2 * r_2 * \left(P_{BGS} - P_i(t)\right) \quad (2)$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \quad (3)$$

In Equation (2), P_{BLS} and P_{BGS} make particle move toward to the best position. 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 uniformly distributed in [0, 1]. The stochastic scheme makes the velocity more diverse.

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. Figure 1 is a flowchart of particle swarm optimization.

2.4 Summarization

This section introduces the related works about group behaviors and the basic method of particle swarm optimization. In general, particle swarm optimization is usually used to find the best “result” in search space. But we can find the fundamental concept of PSO is quite similar to that of the movement of pedestrians according to

section 2.3. We would like to simulate a crowd with the optimization process that particles move toward the expected and currently known best solutions. 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.

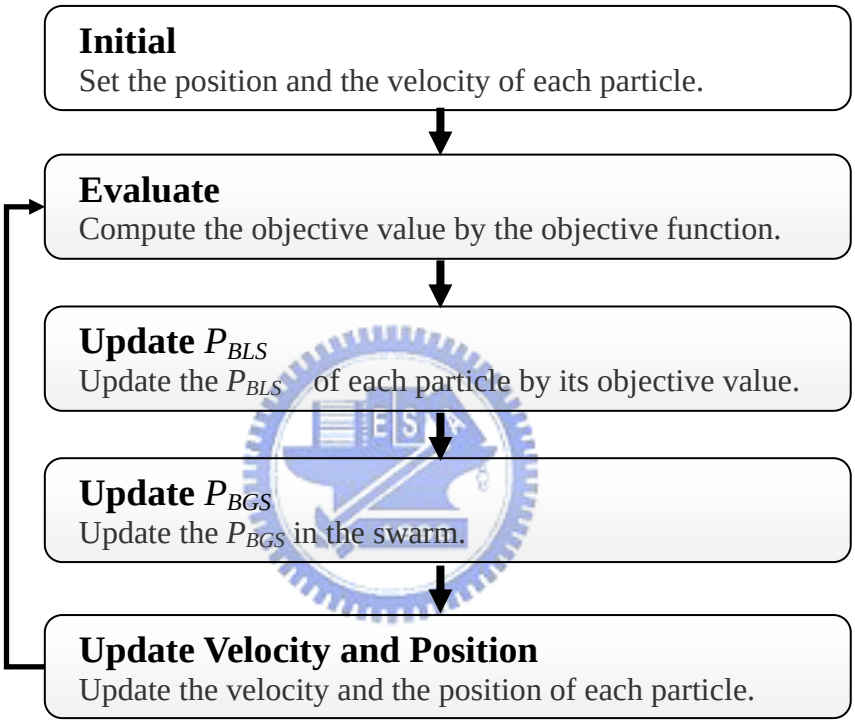


Figure 1. A flow chat of particle swarm optimization.

III. Proposed Model

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, 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. To resolve the aforementioned incompatibility issues, we propose a framework to work with PSO such that the movement of particles can be similar to that of real people.

3.1 PSO Essence

In the light of the analogy between swarms and crowds, we may consider a person as a particle and a group as a swarm. The structure of a particle is similar the Table 1., but we separate the velocity into a direction component *D* and a speed component *S* for convenience. Each particle holds the information about itself, including a position, a direction, a speed, and an objective value. The position and the direction can be represented by 3D vectors[1].

$$P_i = \left[P_{ix}, P_{iy}, P_{iz}\right]^T \tag{4}$$

$$D_i = \left[D_{ix}, D_{iy}, D_{iz}\right]^T \tag{5}$$

[1] Because the objects are simulated in 3D space, we observe the behaviors in x, y and z axes.

The direction and position are computed as

$$D_i(t+1) = \omega * D_i(t) + c_1 * r_1 * \left(P_{BLS} - P_i(t)\right) + c_2 * r_2 * \left(P_{BGS} - P_i(t)\right) \quad (6)$$

$$P_i(t+1) = P_i(t) + D_i(t+1) * S_i(t+1) \quad (7)$$

where $\left(P_{BLS} - P_i(t)\right)$ and $\left(P_{BGS} - P_i(t)\right)$ are both unit vectors for indicating the direction only. Other parameters are defined as the same parameters of the velocity in PSO.

The speed component *S* models and matches the variant paces of different people and has a maximum limit. 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 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.

3.2 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.

3.2.1 Cost function for a unit obstacle

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 in 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 goes up 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

$$Cost(P,Q) = \exp\left(-\left(\frac{(P_X - Q_X)^2}{(\sigma_{P_X} + \sigma_{Q_X})^2} + \frac{(P_Y - Q_Y)^2}{(\sigma_{P_Y} + \sigma_{Q_Y})^2} + \frac{(P_Z - Q_Z)^2}{(\sigma_{P_Z} + \sigma_{Q_Z})^2}\right)\right) \quad (8)$$

Q can be a target, an obstacle, or a particle. For example, if an obstacle's area is 30, 40, and 50, we can set $(\sigma_{Q_X}, \sigma_{Q_Y}, \sigma_{Q_Z}) = (30,40,50)$ or set each edge with smaller combinations. If it is the target, $(\sigma_{Q_X}, \sigma_{Q_Y}, \sigma_{Q_Z})$ is set to the whole search area. Figure 2 shows the exponential model for an obstacle.

The proposed exponential model is similar to the 3D normal distribution with a mean and a standard deviation[2]. 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

[2] (Q_X, Q_Y, Q_Z) is the mean and $(\sigma_{P_X} + \sigma_{Q_X}, \sigma_{P_Y} + \sigma_{Q_Y}, \sigma_{P_Z} + \sigma_{Q_Z})$ is the standard deviation.

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 Equation (9).

$$F_{obj}(P) = c_{obj} * \max_{o \in O}\left(Cost(P,o)\right) + \frac{1}{Cost(P,G)} \tag{9}$$

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 developing 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 accomplish its assigned task, and the particles converge to the desired goal via reasonable paths.

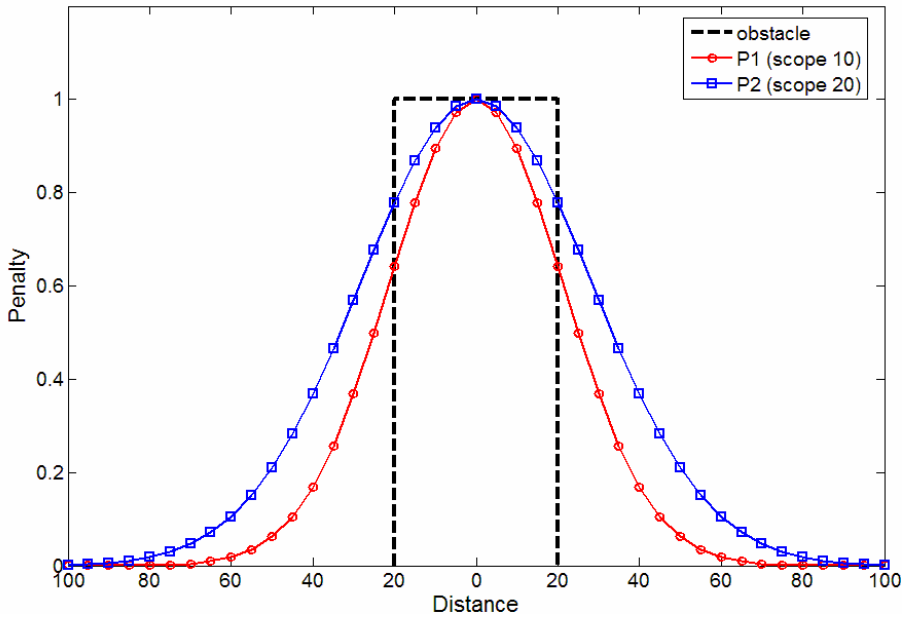


Figure 2. The exponential model for an obstacle. The box indicates the obstacle to be represented by the function with different object P_i.

3.2.2 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

$$prob(f) = 1 - \frac{f}{e^{-k}} \tag{10}$$

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 $k = 1$ according to Equation (8), because the probability will be zero 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.

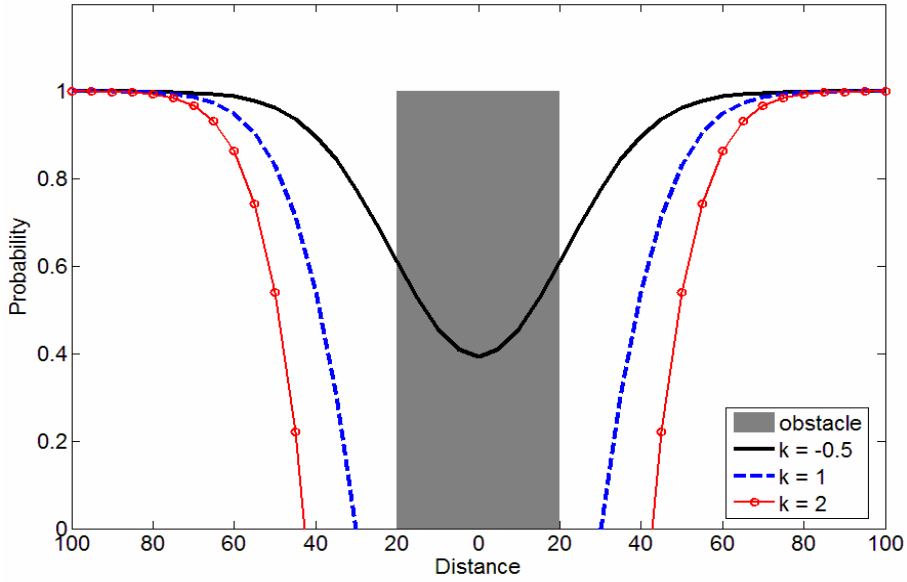


Figure 3. The probability to accept a new position around an obstacle with different k. The scope of object P is 10.

If the cost is not accepted, another direction must be taken to create a smooth path. We use two steps to find an acceptable position. First, according to the direction decided by Equation (6), a random angle less than 90 degrees is added to or subtracted from the direction vector of the particle. This kind of the random direction would be generated several times and the position with the lowest cost is chosen. Following, based on this direction, do a grid search with a fixed angle. The better position is chosen to be the new position. If only with random angle to do local search, it is possible to find no way to walk. Particles often stop if meeting an obstacle. On the contrary, if only fixed angle, it is changeless when particles locate in same position. Therefore, we integrate both. The random search can be regard as a viewpoint to find a walkable area. The grid search can be regard as eyeshot to find a walkable position in the area. Figure 4 shows the local search area. The gray area is the range of random search base on V determined by PSO. The yellow area is the range of grid search base on V' which is selected by random search.

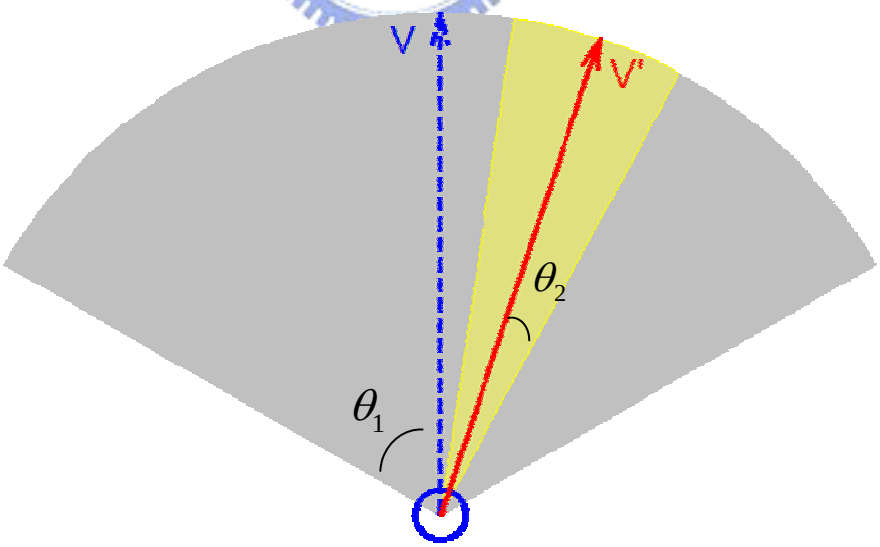


Figure 4. The local search methods in our system.

3.3 Summarization

This section dwells on our proposed model and methods. It is reliable to guide particles toward their target with particle swarm optimization. An identical model makes the system more flexible. The mechanisms designed in local search allow particles to qualify the ability to avoid collisions. Hence, particles in our model can move with a reasonable path.



IV. Moving Path Generation

With the proposed framework, we simulate several showcases on the *x*-*z* plane[3], just like to have a bird's eye view on the top of people's heads. For expediently observing, we show the walking paths generated by PSO in 2D space. Each simulation includes two cases. There are two kinds of figures to show our results. *Path figure* displays the paths of all particles in whole process. *State figure* depicts the condition in a moment. By these two kinds of figures, we can image the dynamic scenery in the simulation. Each simulation has one state figure at least. For convenience, we collect the state figures in appendix. Table 2. The Indication for each symbol in figures.

Table 2. The Indication for each symbol in figures.

Symbols	Indication
Start	Goal Position
Solid line	A path of a particle.
Filled area	Obstacles.
Circle	A person.
Quiver	The direction of a person.

4.1 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 $\varpi = 1.0$, $c_1 = 0.5$ and $c_2 = 0.5$. The population size is 20, and the initial positions are assigned. The velocities are initialized at random.

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

4.1.1 Without Obstacles

Figure 5 shows an influx. Each solid line indicates a path of a person. People initially are scattered and finally converge to the target. Figure 6 shows a stream. People who 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 the original PSO.

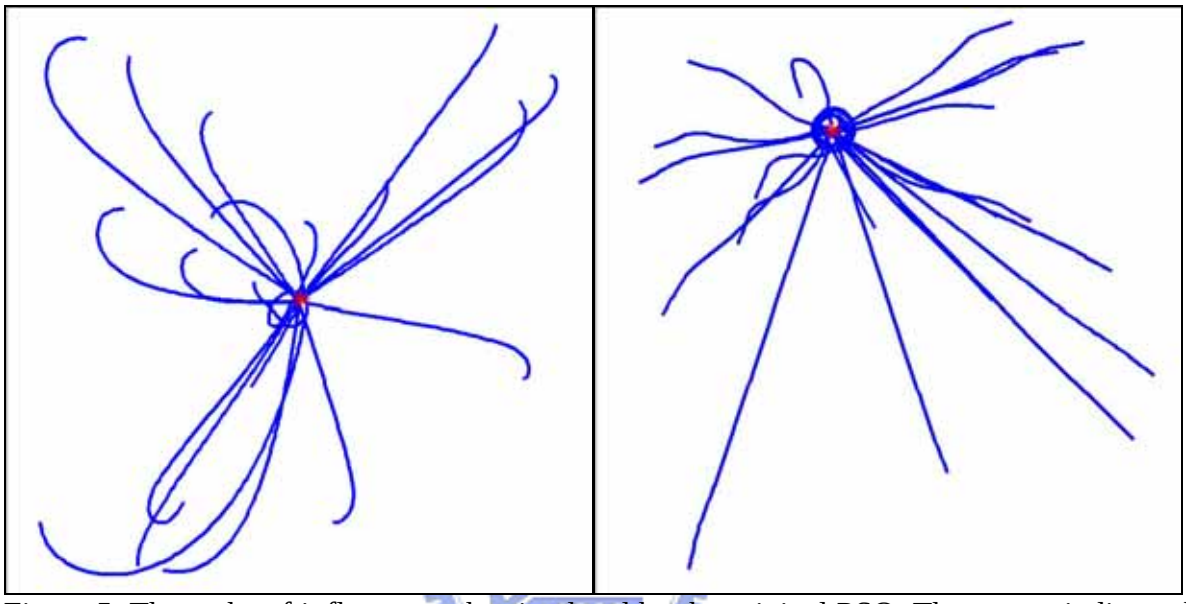


Figure 5. The paths of influxes can be simulated by the original PSO. The curves indicate the paths of people in the crowd.

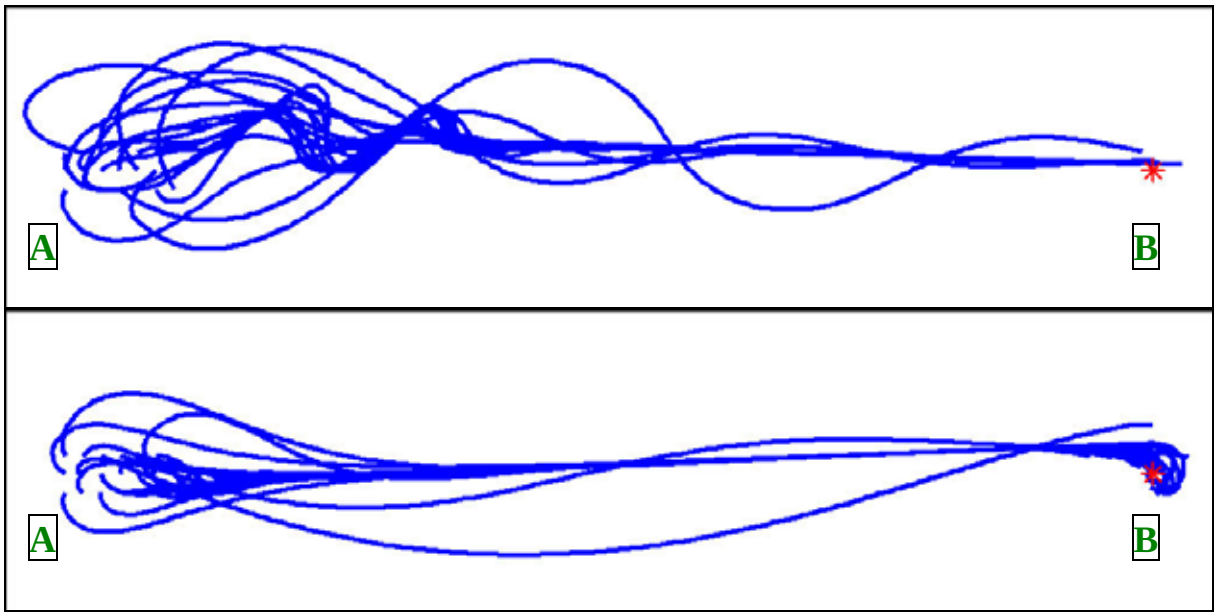


Figure 6. The paths of streams are generated by the original PSO. They simulate that a crowd moves from position A to position B.

4.1.2 With 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 7. We set an obstacle on the way from position A to position B. It can be observed that the original PSO with the mechanism of the penalty can guide the crowd to select a better way to avoid the obstacle. Sometimes it is successful to avoid the obstacle successfully, but it is not implicit.

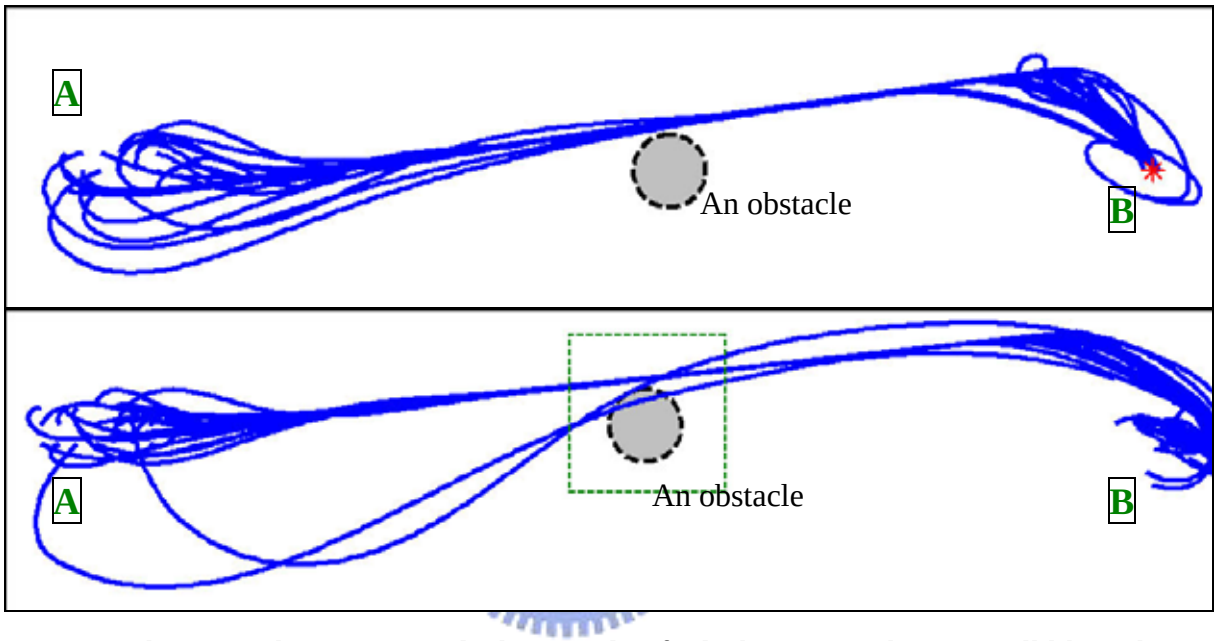


Figure 7. The original PSO can guide the crowd to find a better way, but it is still like to have a collision.

4.2 Crowd Simulation with Collision Avoidance

The collision avoidance mechanism has to be employed in the same situation.

4.2.1 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. Stream cases are simulated in Figure 8. It holds a distance between particles, and don't converge first.

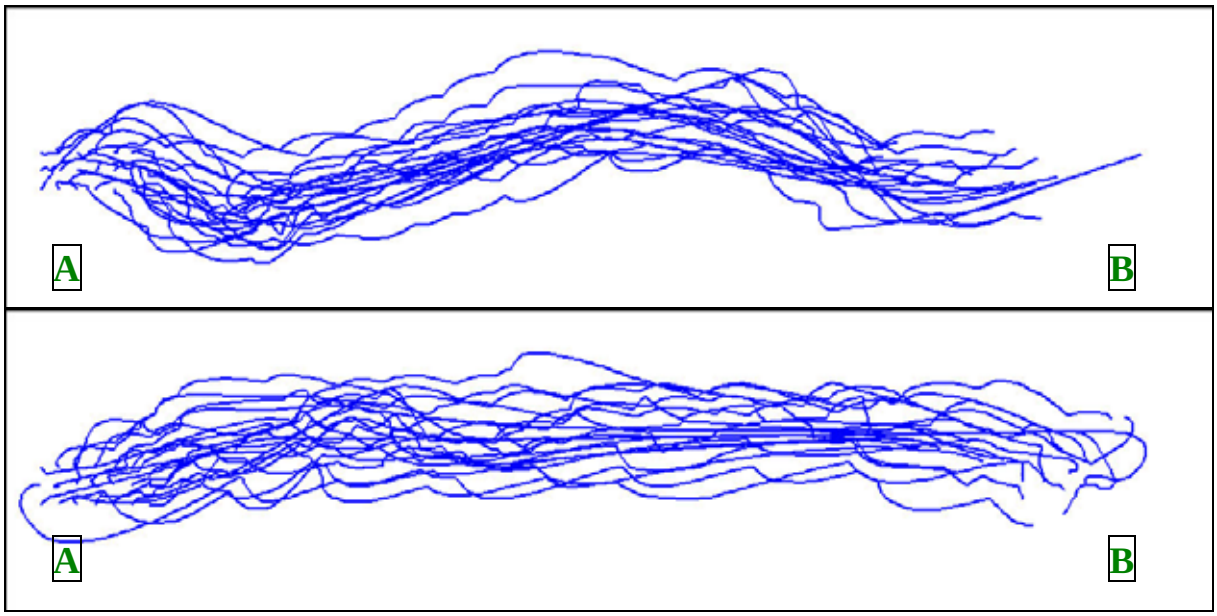


Figure 8. Two stream cases are displayed in our system.

4.2.2 With Static Obstacles

We also place an obstacle between position A and position B. Figure 9 shows a crowd passing by a static obstacle in our system. Based on the paths, we can see that the crowd can avoid the obstacle and reach the goal. The crowd will keep a distance to the obstacle, because the solid line is the position center of a particle, not the volume.

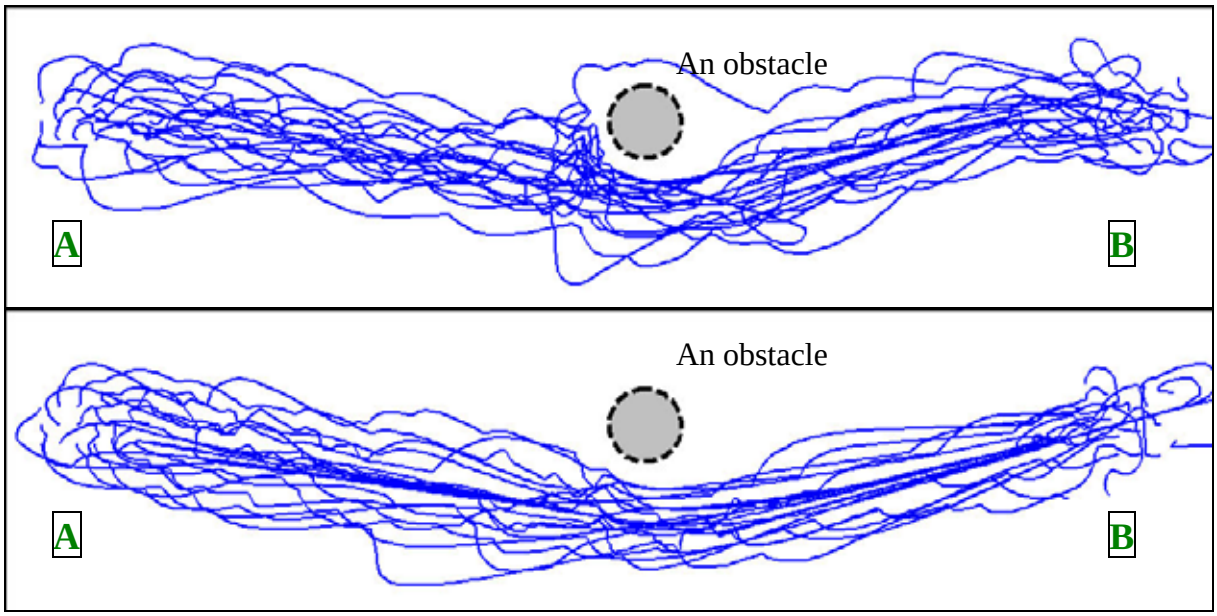


Figure 9. The situations for collision avoidance in our system are presented.

4.3 Other Crowd Simulations

Three types of crowds are simulated in our system, including Dynamic Target, Multiple Crowds and Geography.

4.3.1 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 10.

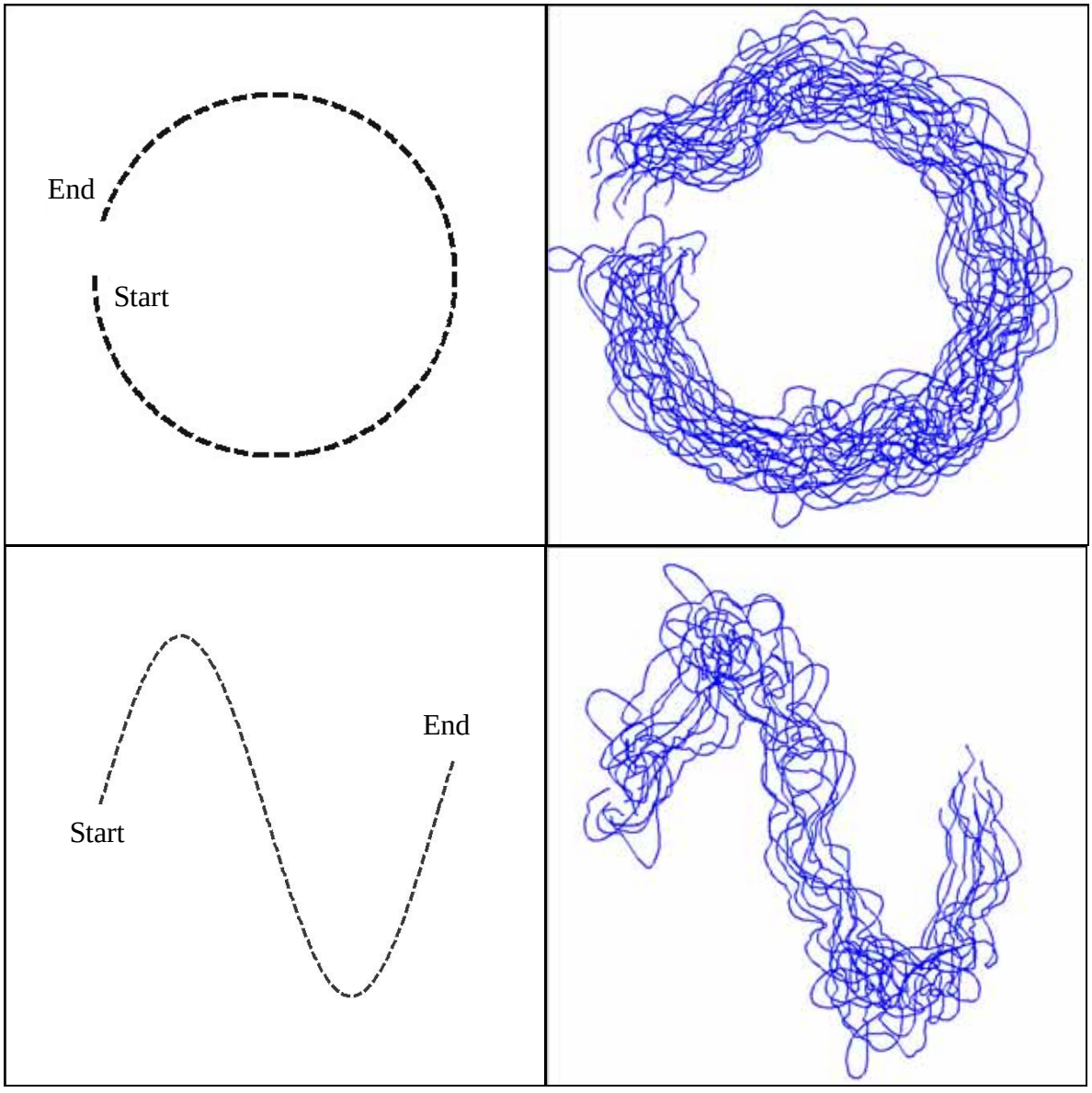


Figure 10. A circle crowd and a wave crowd are simulated.

4.3.2 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 17 and Figure 18 demonstrate 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 respective targets.

4.3.3 Geography

It is feasible to simulate a crowd in different topography in our system. We adjust the parameter *k* in Equation (10) to fit different situations. Figure 11 exhibits different topographies for people move from left side to right side. We present 3D scenery and each sphere represents a person. It is explicit to find the altitude is higher, the accept probability of people is lower.

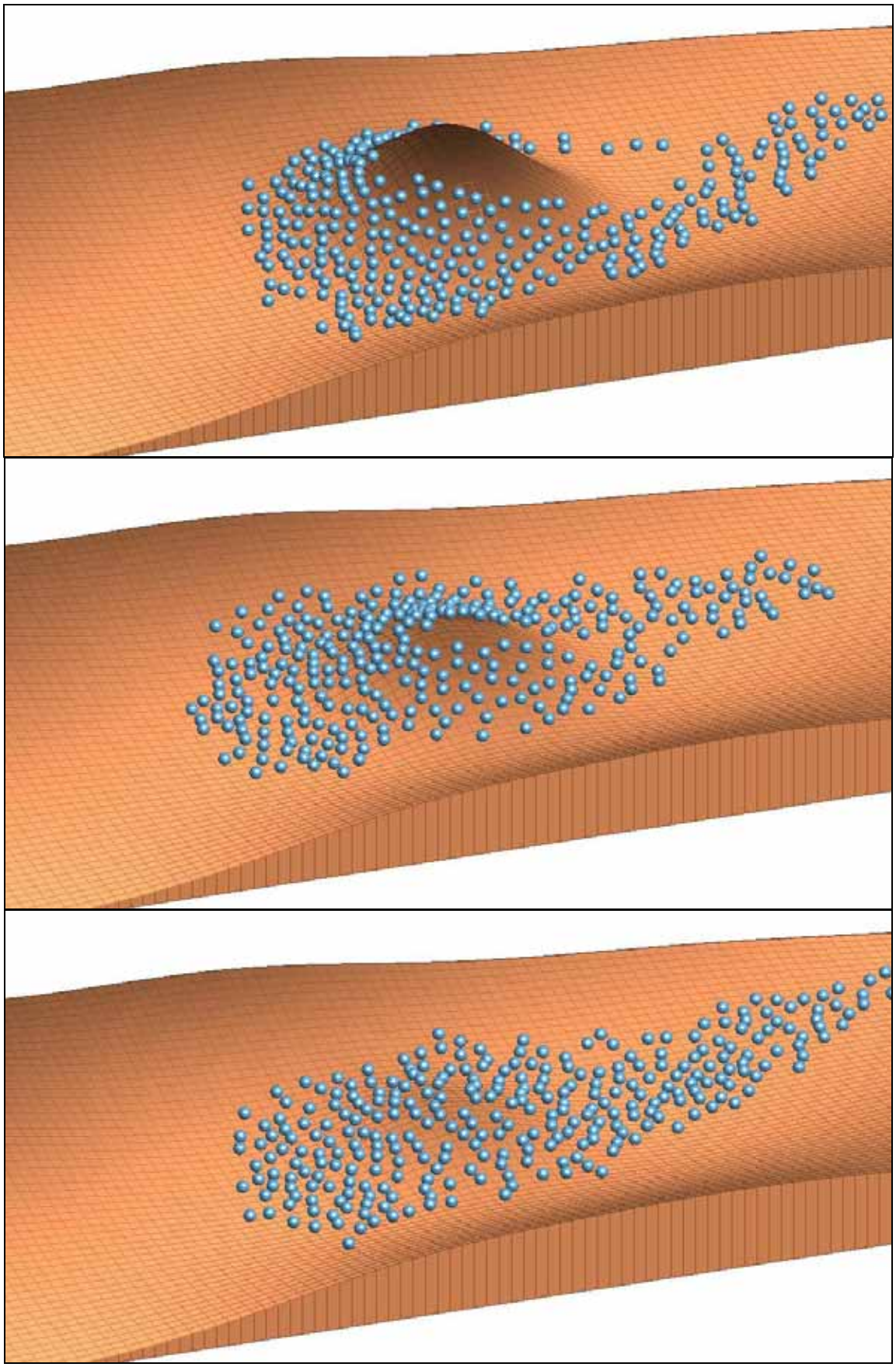


Figure 11. Three different topographies.

V. 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 environments 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 easily. Based on the uniform model, we can demonstrate a complex crowd-space 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 variety 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 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.

Bibliography

[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. G¨obel, 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. Popovi´c, “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.

A. *state figures*



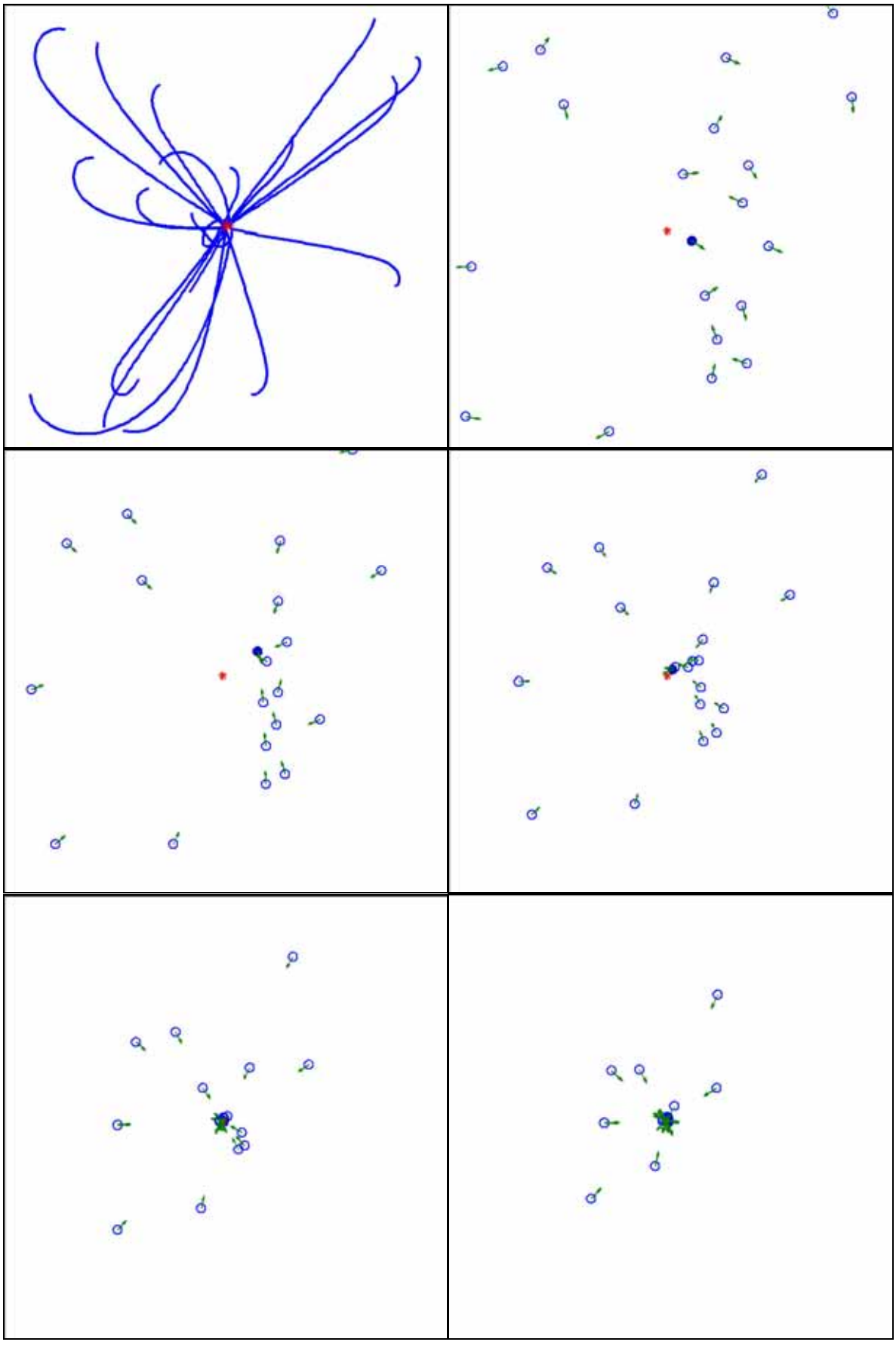


Figure 12. The conditions for a influx by the original PSO.

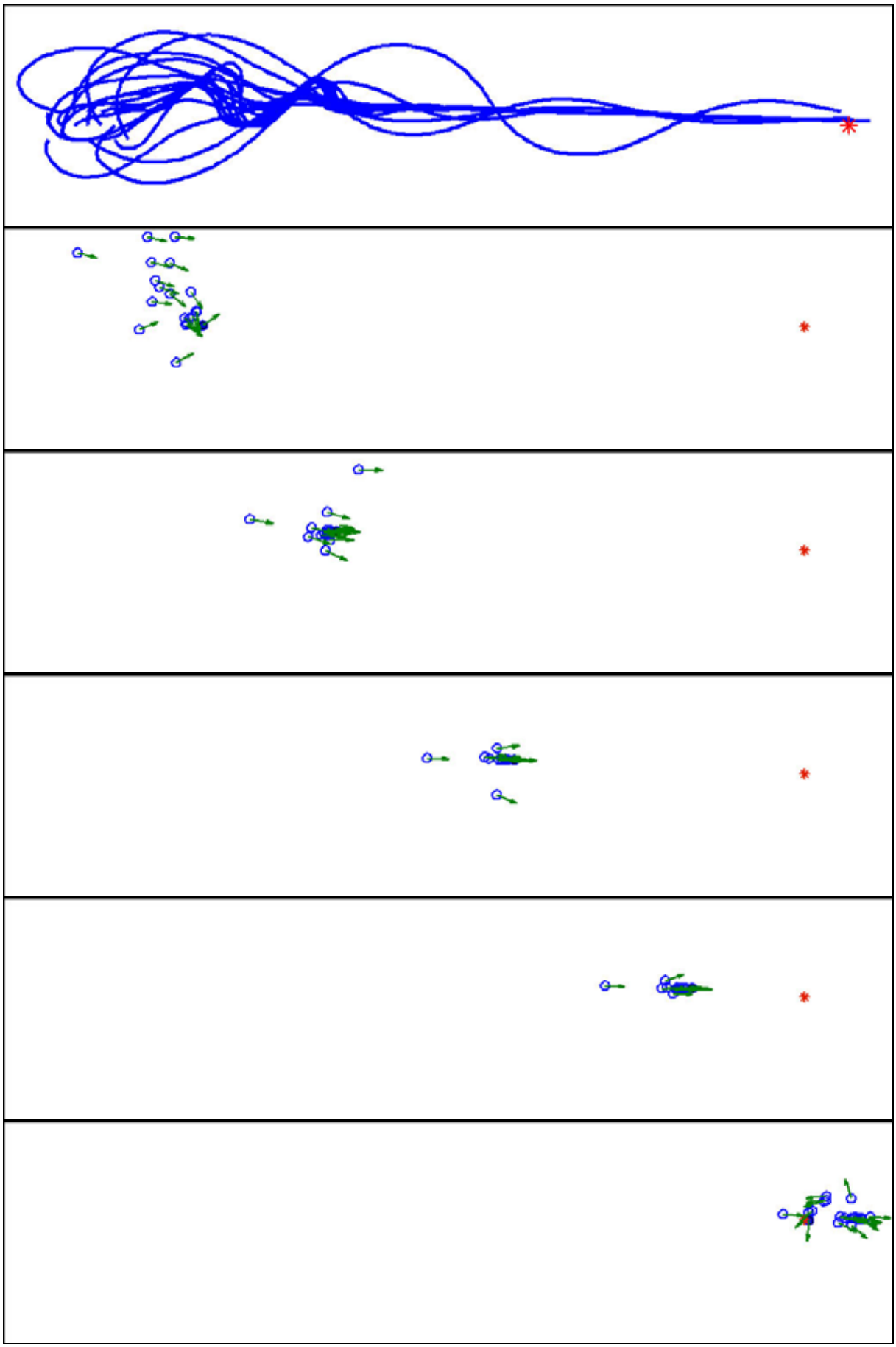


Figure 13. The conditions for a stream by the original PSO.

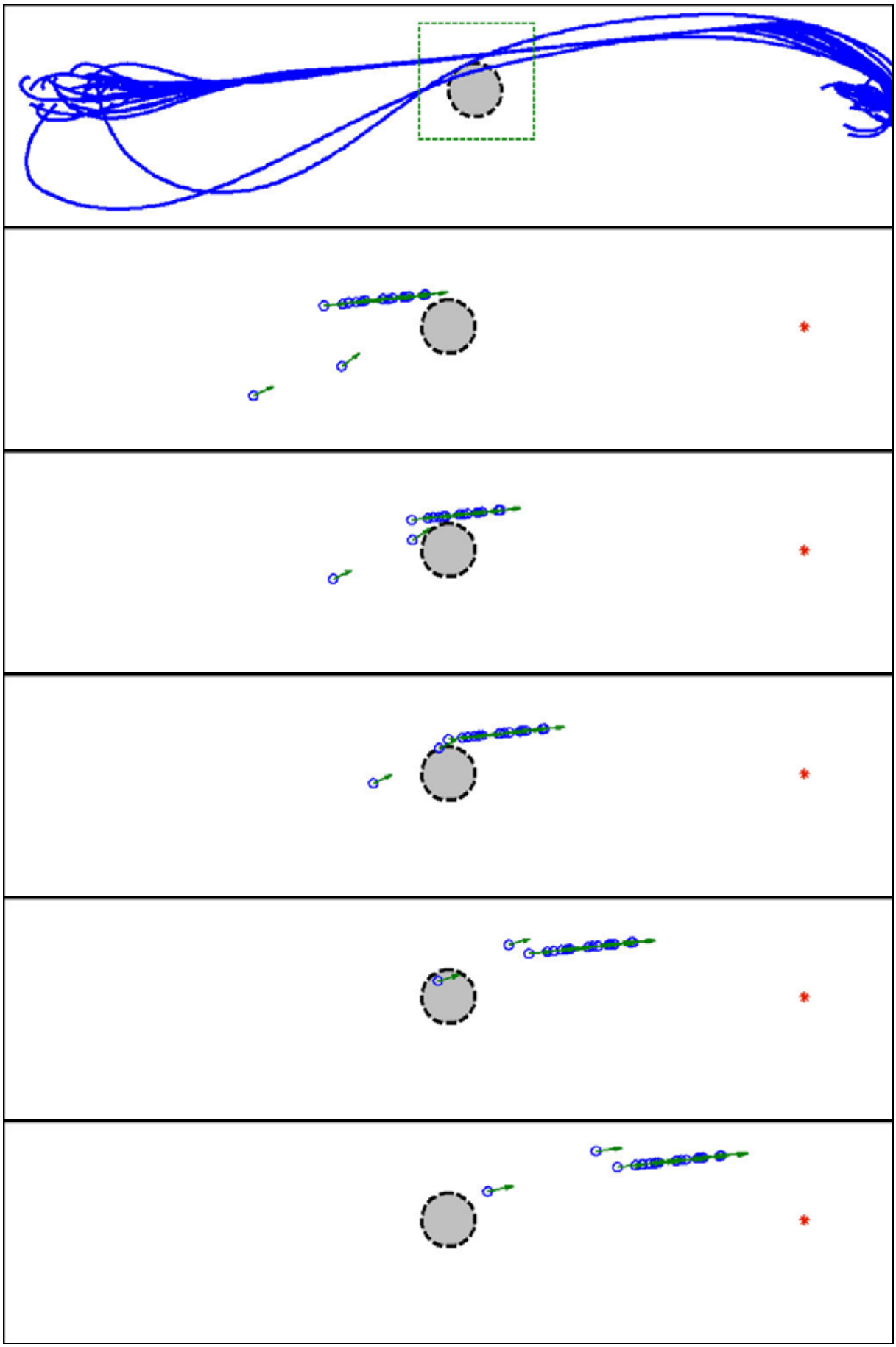


Figure 14. The conditions for collictions by the original PSO.

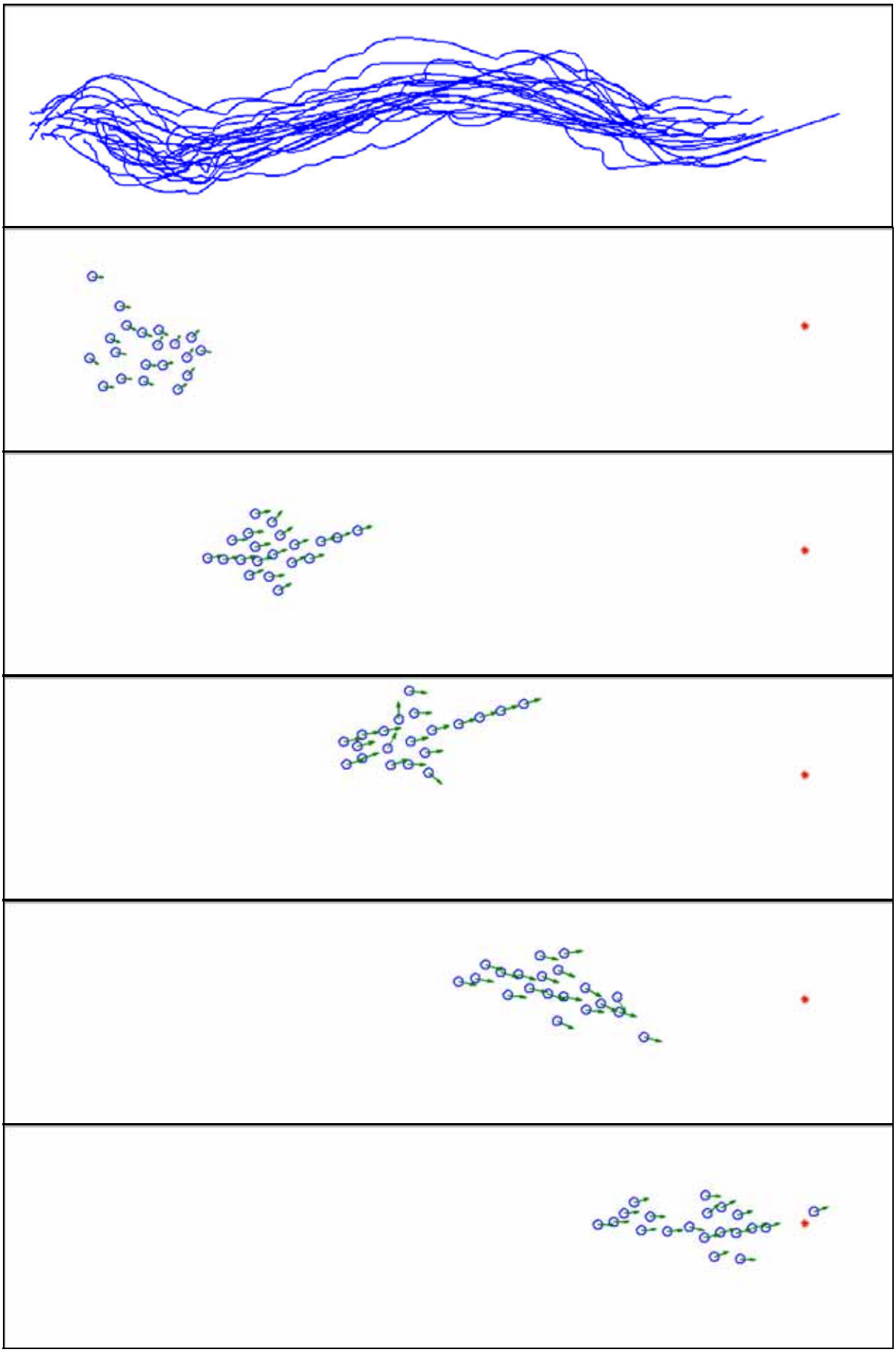


Figure 15. The conditions for a stream in our system.

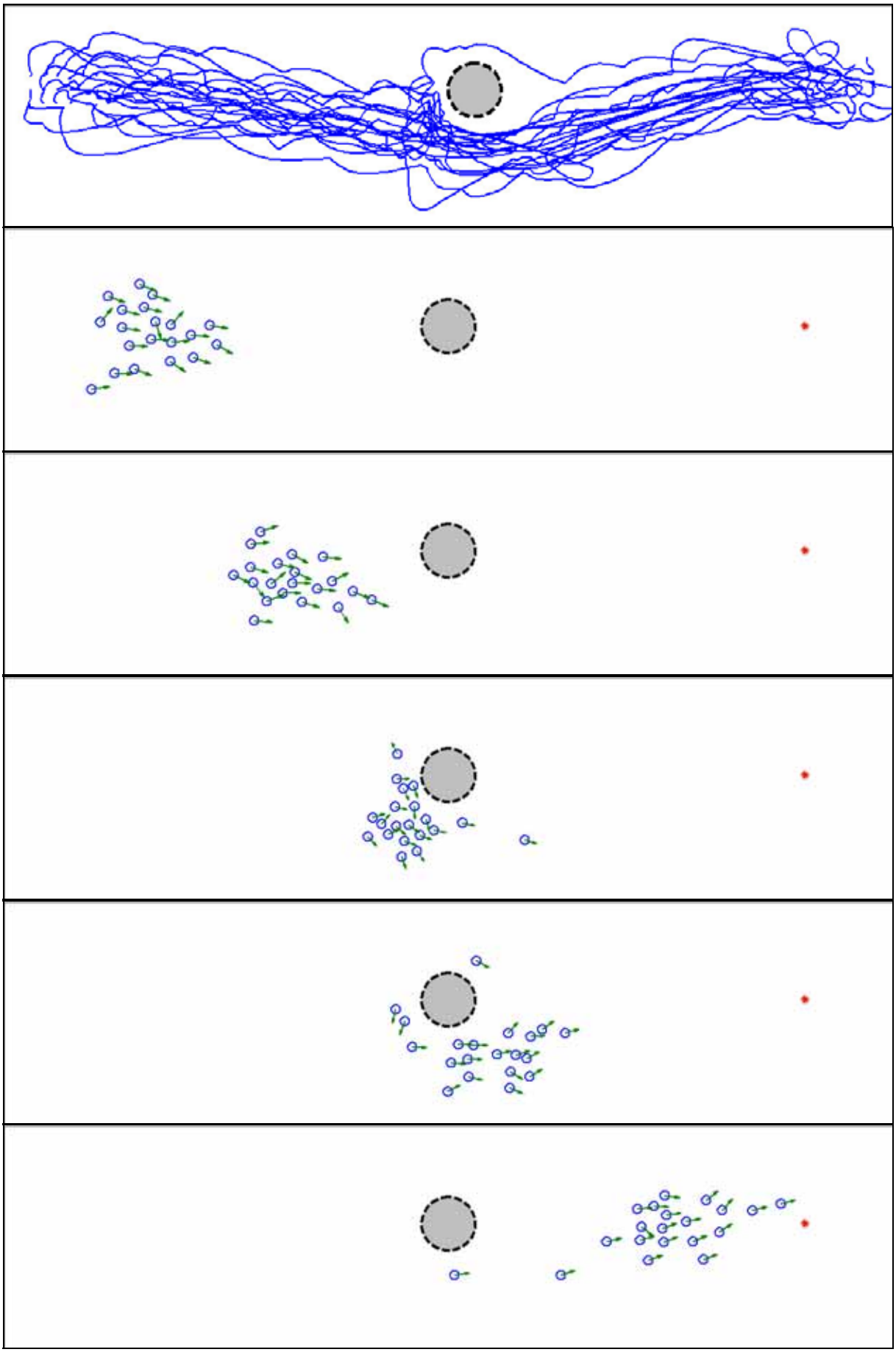


Figure 16. The conditions for collision avoidance in our system.

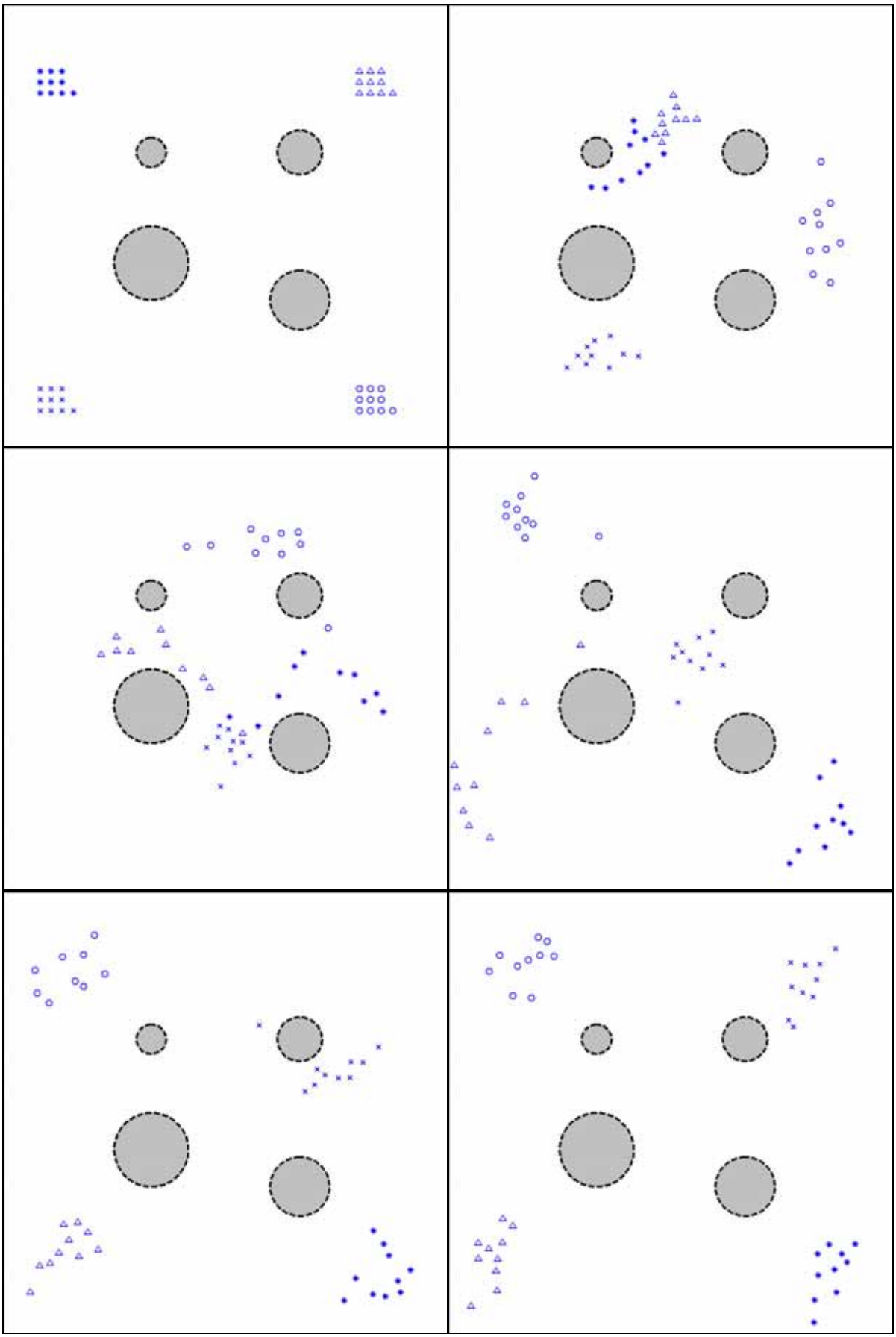


Figure 17. Four groups in the four corners move toward their opposite corners. (Case 1)

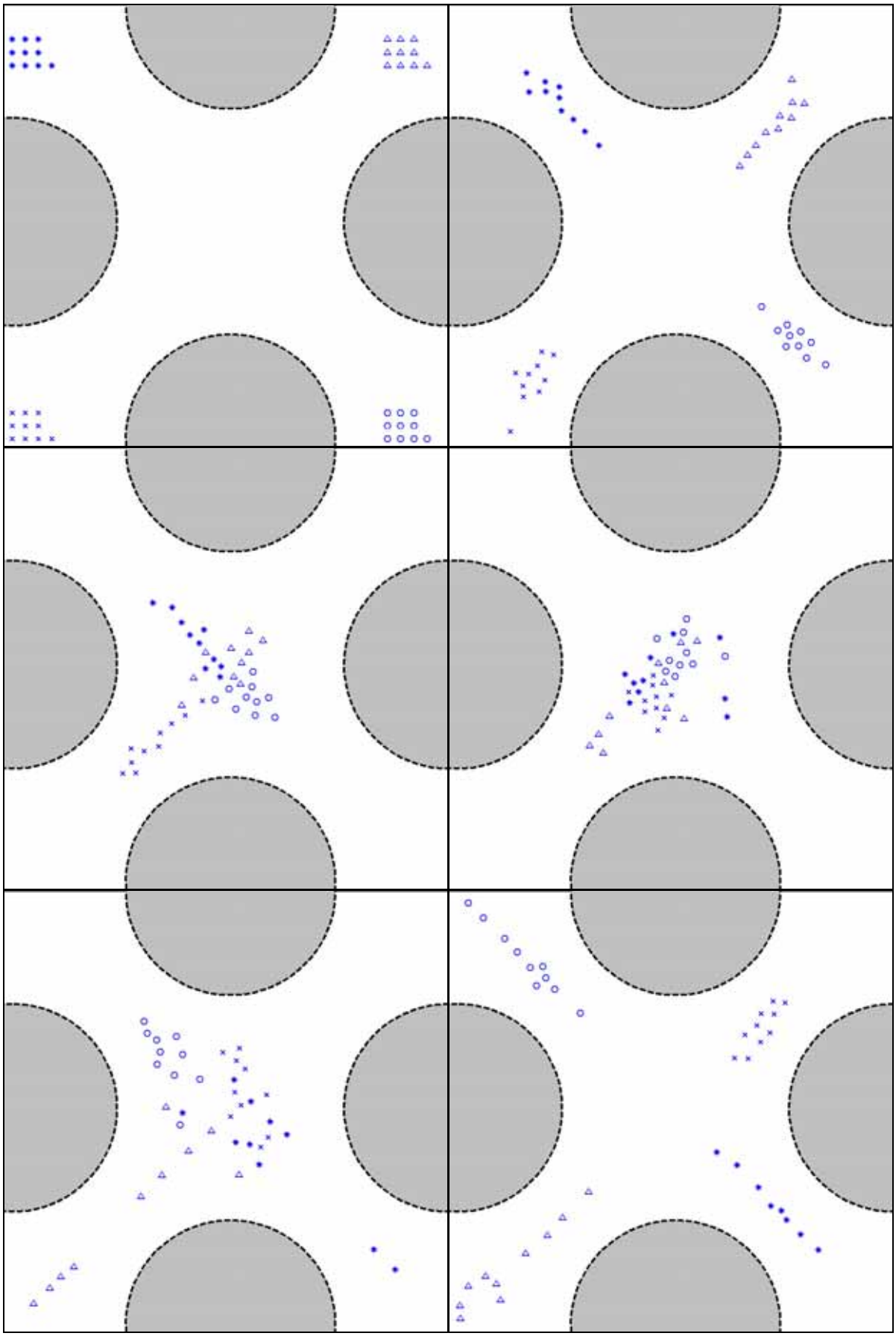


Figure 18. Four groups in the four corners move toward their opposite corners. (Case 2)

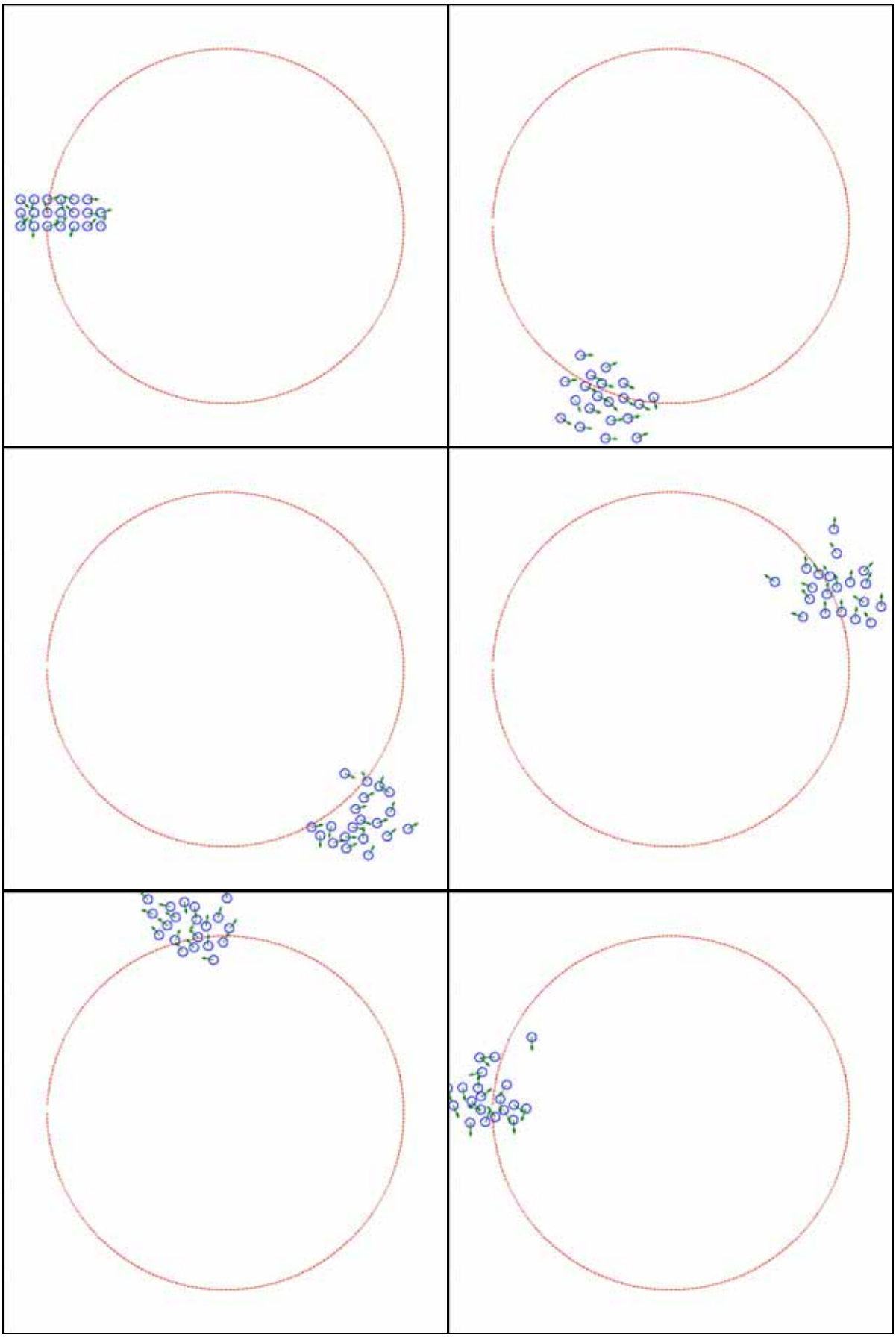


Figure 19. The condition for a crowd moves following a dynamic target.