Appl. Math. Inf. Sci. 6 No. 1S pp. 167S-175S (2012)

Applied Mathematics & Information Sciences
An International Journal

Authorable Dense Crowd Simulation based on Smoothed Particle Hydrodynamics

Shin Jin Kang[1] and SooKyun Kim[2]
[1]School of Games, Hongik University, Korea
[2] Dept. of Game Engineering, Paichai University, Korea
Email Address: kimsk@pcu.ac.kr

Received: Received May 02, 2011; Revised July 25, 2011; Accepted September 12, 2011

Abstract: The control of dense crowd is a time-consuming and challenging task. There are always conflicts between detailed movements and the control of large-scale crowd. To solve this problem, we suggest the smoothed particle hydrodynamics (SPH) technique to express continuous flow with two authorable interfaces. To reflect the connectivity that occurs in a crowd, we use an authorable sociogram model. And we present a new type of shape map to produce various shapes of crowd in real time under interactive control.

Keywords: Crowd simulation, Smoothed particle hydrodynamics

1 Introduction

In recent years, crowd simulation has expanded from a research topic to the commercial production of animation and interactive games. To simulate the movement of a crowd realistically, many techniques need to be integrated efficiently. Even with modern computer hardware, the complexity of crowd behavior makes it difficult to model precisely.

Most of previous crowd simulations adopted agent-based scheme to produce natural movements of individuals. In such a system, each individual agent perceives its environment and other individuals, and decides its actions based on several rules with variable parameters. The advantage of this model is that the integration of individual rule-based actions allows the synthesis of collective group behavior of a high complexity. The disadvantage of this approach is that the behavior of the crowd cannot be controlled directly, making it difficult to produce a particular style of group behavior. This approach requires intensive manual process, operating in a pre-processing time to make specific behaviors. In commercial production of animation and interactive game[16,17], this can cause high decrease in productivity.

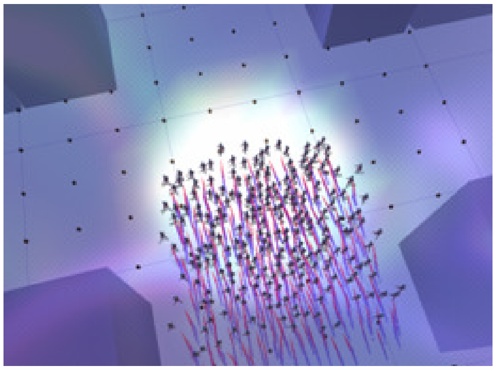


Fig. 1 Our dense crowd simulation result

Our aim is to produce dense crowd movement while providing an intuitive method of overall control. This can be achieved by using scalable

---

[2]**Corresponding Author: SooKyun Kim, kimsk@pcu.ac.kr**

agents within a two-level concept: 1) low-level individual locomotion and 2) high-level mass control. For low-level locomotion, we use SPH to simulate the detailed movement of a crowd, which is in many ways similar to that of a fluid. For high-level control, we introduce two interactive control methods: a sociogram model of inter-group connectivity and a shape map for modeling the overall geometry of a crowd. These two authoring techniques can be applied scalably to the individual agents by interactively changing the combination of components.

2 Related Work

Agent-based crowd modeling has been suggested in various research areas such as transportation science, robotics, and computer graphics. In the transportation science and robotics, most of agent-based crowd simulations have focused on finding fast, dynamic, and optimized path of individuals to the goal with pre-defined behaviors for the practical purpose. However, in the computer graphics, to find a natural human-like movement of individual is as important as to find an optimal path to the goal.

Reynolds[11] showed that flocking is a dramatic example of global behavior arising from the interaction of simple rules. Various behavioral rules have subsequently been suggested to achieve more realistic crowd simulations and integrated with other approaches. Helbing's[4] empirical social forces model which applies repulsion and tangential forces to simulate interactions between people and obstacles, achieving realistic pushing behaviors and variable flow. Anderson [1] suggested a method for imposing constraints on the paths of agents at specific times to producing constrained flocking motion. These agent-based approaches can produces realistic emergent behaviors but usually require a lot of computation and parameter tuning and limit the ability of a crowd model to express complicated organized movement. Another method of crowd simulation has emerged from fluid dynamics.

In this approach, a crowd is viewed as a dense particle model of a continuous fluid. Hughes [5] used fluid dynamics to describe crowd locomotion, producing impressive large-scale effects. Treuille [14] extended this work to simulations driven by dynamic potential fields, in which global navigation is integrated optimally with local collision avoidance.

Crowd control is another research area for producing simulation and animation of crowd with authoring method in an efficient way. Flow tiles [3] method shows that large, designed crowd flows can be required by offering user interface for tile design. Crowdbrush[15] is an intuitive method for authoring real-time crowd scene by supporting graphical 2D interface. Jin[6] proposed vector field based governing tool, which is realized with sketching interface of velocity in the scene. Kwon[7] presented an approach to editing group motion as a whole while maintaining its neighborhood formation and individual moving trajectories. Lee [8] recorded the motion of a human crowd from an aerial view using a camcorder, extracted the two-dimensional moving trajectories of each individual in the crowd, and then learned an agent model from observed trajectories.

In this paper, we suggest crowd authoring method for specific condition. Our goal is generating realistic dense crowd simulation. Rahul [10] presents a novel, scalable approach for simulating such dense crowds, using a dual representation both as discrete agents and as a single continuous system. We try to achieve authoring goal for dense crowd simulation with different approach. We combined SPH technique with a sociogram to produce plausible individual movement, and a shape map to allow the pattern created by the crowd as a whole to be controlled. SPH is one of Lagrangian approaches for fluid dynamics. It is numerical technique which involves the approximate integration of partial differential equations.

Müller [9] proposed an interactive method based on SPH to simulate fluids with free surfaces. Tantisiriwat [12] recently presented an interactive crowd simulation in which automatic path construction in combined with SPH to allow each individual in the crowd to search for their destination without using pre-computation of their path. We have been motivated by the way in which SPH is able to model continuous fluid movement in detail.

3 Background

In computer graphics, SPH is used to model fluid motion. In SPH, a flow volume is discretized into particles that are used as nodes in a simulation. Each particle $\boldsymbol{p}_i$ is defined by its position $\mathbf{x}_i$, a support radius $\boldsymbol{r}_i$, and a mass $\boldsymbol{m}_i$. The scalar properties of the quantity $\boldsymbol{A}$ can then be interpolated at a location $\mathbf{r}$ as a weighted sum of the contributions of all the particles:

$$A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} W(r - r_i, h) \qquad (1)$$

where W(r, h) is a weighting function, which is called the smoothing kernel in the SPH method. This kernel function smoothes out the field by averaging it over several particles. The choice of kernel function depends on the purpose of the simulation. SPH has several benefits over traditional grid-based techniques. First, SPH guarantees conservation of mass without extra computation since the particles themselves represent mass. Second, SPH computes pressure from weighted contributions of neighboring particles rather than by solving linear systems of equations. In SPH, the influence of neighboring particles is computed and results in a pressure acts to maintain a uniform density. In a similar way, people in a crowd move to low-density areas to escape the crush of high- density areas. SPH can describe basic behaviors such as avoidance of a crowd well but they cannot be applied directly to crowd simulation without considering the characteristic movement of individuals. Therefore we add the following properties to make SPH applicable to a crowd:

- Direction

Particles in SPH do not move in any preferred direction. But any individual in a crowd may attempt to make progress in a particular direction. If this is not considered, individuals will appear to vacillate implausibly. To prevent this, we design new SPH terms.

- Mechanism of collision avoidance

Particles in SPH experience repulsive and attractive forces which can move them in any direction to maintain a uniform density. But individuals in a crowd rarely move backwards; instead they tend to swerve or stop to avoid collisions.

- Connection

Individuals in a crowd have social relations which affect their movement. There are many types of relation, including friendship, hostility, leading, and following. Each of these has different effects on the movement of a crowd as a whole. We use the links in a sociogram to achieve this property.

- Path finding

Individuals in a crowd try to find the best path to their goal. This path-finding process is the most time-consuming step in a simulation. The path need to be optimized both locally and globally to avoid the formation of bottlenecks. We use Dijkstra's algorithm to construct a directed graph which is integrated into our shape map approach.

4. Assessment Method

For the purpose of representing crowd locomotion, we write the equation 2. which determines its acceleration. The term $\vec{f}$ is a summation of four forces; which include the *avoidance*, *group*, *social* and *goal-directed* forces. Where α, β, ω and τ are parameters which control the relative strength of the *avoidance, group, social* and *goal-directed* forces. The *avoidance* and *group* forces determine the effect of interactions within a group. They are responsible for the smoothness of particle movement and the avoidance of collisions. The *social* forces model the effect of links between individuals. And the *goal-directed* force propels the particle towards a target position which is generated by Dijkstra's algorithm and shape map.

$$(2) \sum \vec{f}_i = \alpha \vec{f}_i^{\,avoidance} + \beta \vec{f}_i^{\,group} + \omega \vec{f}_i^{\,social} + \tau \vec{f}_i^{\,goal}$$

$$(3) \vec{a}_i = \frac{d\vec{v}_i}{dt} = \frac{\sum \bar{f}_i}{\rho_i}$$

Basic movement: the avoidance force

In SPH, a simple repulsive force is sufficient to produce the local motion of the particles. But in a crowd simulation, people move forward in the appropriate directions to avoid congestion. To simulate this behavior we modify the pressure term used by Müller, as follows:

$$(4) f_i^{avoidance} = -\nabla p(r_i) = -\sigma_i \sum_j \vec{R} m_j \frac{p_j}{\rho_j} \nabla W(r_i - r_j, h)$$

$$(5) p_i = k(\rho - \rho_0)$$

where $\vec{R}$ is the avoidance vector, and k is a stiffness parameter. We introduce a pressure P_i which is proportional to the difference between the current density and rest density. The avoidance vector $\vec{R}_i$ can be of three types:

a) A rotation vector which makes individuals avoid others by changing direction. The extent of this swerving effect is proportional to the distance between the individuals.

b) A translation vector which causes an individual to move backwards or step aside to make way for another individual.

c) A vector of appropriate direction and magnitude to cause an individual to stop, so as to allow another individual to pass.

The choice of avoidance vector can be expressed as follows:

$$\vec{R}_i = \begin{cases} Rotate(\theta_i) \;\; if\; (\nabla W(r_i - r_j, h) \cdot \vec{v}_i) > 0 \\ Translate(Dir_i) \;\; if\; (\nabla W(r_i - r_j, h) \cdot \vec{v}_i) < -0.5 \\ 0 \qquad \text{Otherwise} \end{cases} \qquad (6)$$

When the density of a crowd is high, individuals may oscillate to allow other individuals to pass. This oscillation phenomenon can be reduced by adding a constraint term σ_i, which makes individuals wait for a specified period before moving. In this case a density threshold is defined by counting the number of individuals within a radius h_2 of the individuals who is about to move:

$$\sigma_i = \begin{cases} 0 \quad \text{if (number of agent in h2 } > \text{maximum number)} \\ 1 \quad \text{Otherwise.} \end{cases} \qquad (7)$$

To create this avoidance force, we use the following spiky gradient kernel, which is proposed in [9]:

Our model allows individuals to exhibit the following behaviors: a) avoidance by rotation; b) avoidance by translation; c) waiting – the individual waits until another has passed. Equations 4, 5 and the spiky gradient kernel generate these behaviors in the context of crowd density. Individuals in our system always try to avoid others by moving to an area of relative low density, causing the crowd to assure its characteristic uniform density overall, and preventing local bottlenecks being caused by low-level individual locomotion.

Basic movement: the group force

People usually try to form a group within a certain distance of each other if they have some sort of relationship. This group distance is represented by the radius h_2 in the following equation, which produces a grouping force:

$$(9)\; f_i^{group} = \mu \sum_j m_j \vec{R} \frac{|v_j - v_i|}{\rho_j} \nabla^2 W(r_i - r_j, h_2)$$

The strength of this force is proportional to the relative velocity of two individuals. For the grouping force, we use another Laplacian of polynomial kernel:

$$(10)\; \nabla^2 (\mathrm{K(r)}) = -\frac{45}{\pi \mathrm{h}^6}(\mathrm{h} - \mathrm{r})$$

This grouping force allows us to model the following behaviors: a) basic grouping, b) lane forming – individuals make lanes to minimize collisions by following others; and c) distance maintenance. Unlike a Reynold's [11] model of grouping behavior, this approach propels individuals together to fill low-density areas.

Interconnectivity: sociogram links

A sociogram is a graphic representation of a person's social links, which allows the structure of the interpersonal relations in a group situation to be plotted. There are many types of sociogram for different purposes. Our sociogram is designed for real-time implementation, and consists of three types of social domination area and three types of social relation links, as shown in Fig. 2. The social areas are: a) 'personal distance', which is the minimum distance between individuals; b) 'group distance', which is the range of an attractive effect between members of a group; and c) 'social distance', which is the range of specific relations between linked individuals. The social relations are: 'mutual select', which represents mutual friendships; 'follow', which is a one-way friendship; and 'exclude', which is a hostile relationship. Two 'exclude' links can be used to denote mutual hostility.

In our system, three different types of social force attract or repel individuals. These forces are generated from a sociogram link structure with five components: a source agent, a target agent, an initial relative length l_k^{social}, an initial relative angle vector v_k^{social}, and a social force type. These forces act within the social distance h_3, which is greater than h_1 and h_2, so that the social force is dominant. We used a modified spring-damper model to implement these social links so as to reduce any oscillations. Our system supports formation created by sociogram links which allow more coordinated movements. Every social link is assigned an initial status by the user, and the system attempts to minimize the difference between the current and initial status of each link. In shape maintenance mode, the system generates two types of relative acceleration, a_i^{length} and a_i^{angle}, which result in a restoring force that acts to maintain the shape of a group of linked individuals.

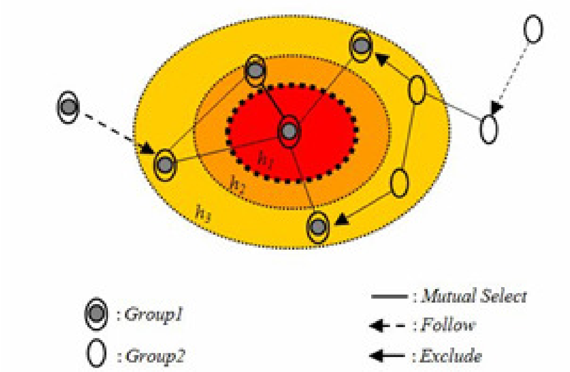


Fig. 2 Our sociogram. The domination areas: h_1 is 'personal distance'; h_2 is 'group distance; and h_3 is 'social distance'. The social relations: solid lines are 'mutual select' links; dashed arrows are 'follow' links; and solid arrows are 'exclude' links

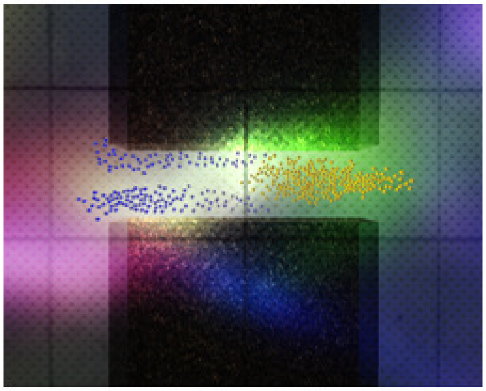


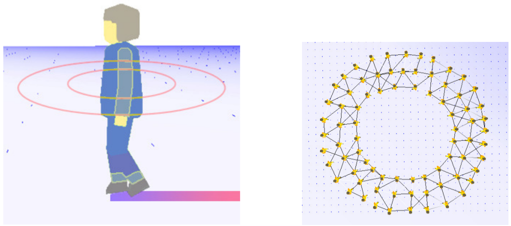
Fig. 3 Natural bi-directional flow augmented with sociogram links

Fig. 3 shows the bi-directional flow produced by sociogram links, and Fig. 4 shows the sociogram link structure used in our simulation. This social force allows our system to create the following types of crowd behaviors: a) assembling - individuals can assemble in a specific pattern; b) leading and following – the whole crowd follows the appointed leader; c) formation restoration – a crowd regains its original formation from an unorganized status; d) moving in formation; e) convex hull formation – a crowd tries to maintain the shape of its convex hull against outside pressure. This allows the user to move a whole crowd by placing a few outlying individuals.

$$a_i^{length} = -\kappa^{length} \sum_k n_{ij} \left| l^{current}{}_{ij} - l_k^{social} \right| \frac{l_{ij}}{l_k^{social}} \qquad (11)$$

$$a_i^{angle} = -\kappa^{angle} \sum_k (a^{current} - (v_k^{social} + v_{ij})) \qquad (12)$$

$$A_i^{social} = a_i^{length} + a_i^{angle} \qquad (13)$$



(a) (b)

Fig. 4 The 'personal distance' radius h_1 and the 'group distance' h_2 (a), and a network of sociogram links (b)

Forming a specific shape using the shape map

Creating crowds that fill specific shapes is a challenge for crowd simulations. An agent-based crowd system can only handle a limited number of individuals because of the cumulative unpredictability of the individual's behaviors. To solve this problem, we use a shape map for interactive shape control, consisting of three steps:

1) *Painting the shape map*

The position of each individual usually determined incrementally by a shortest-path algorithm. This approach can make fastest movement but cannot make specified shape of crowd. To make shape of crowd, we combine shortest-path algorithm with user created shape map. The shape map, which coordinates the incremental positions of every individual, consists of three types of cells: a) a *seed cell* is the starting point for the automatic positioning step; b) a *density cell* has a density level (low, medium, or high) which determines the radii h_1, h_2, and h_3 for the individuals on it. This equalizes the distribution at the automatic positioning step; and c) a *block cell* is an obstacle that limits the spread of particles for positioning in automatic positioning step. Every cell can be painted interactively by the user, and this controls the automatic generation of the potential field. Additionally, the system can read the field from terrain data. Fig. 5 shows the different types of cell in a shape map. Fig. 6 shows the shape map which is used for our number mass game in the movie. 400 individuals make numbers one after the others interactively.

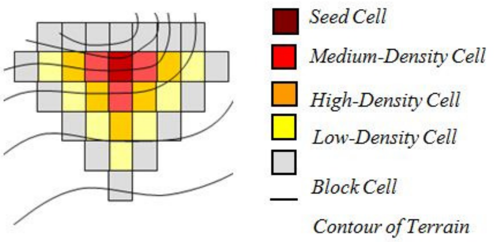


Fig. 5 Different types of cell in a shape map.

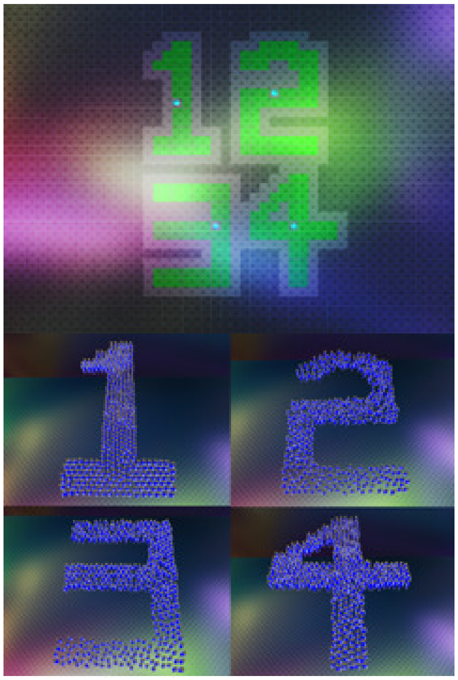


Fig. 6 Shape maps for the number mass game (above), and simulation results (below)

2) *Automatic positioning*

Every movement that contributes to the formation of a crowd shape is determined at the automatic positioning step. After the potential field has been constructed, our system uses it to create a vector field. Particles for positioning move towards its required positions by following the curvature of the vector field. When each particle reaches the equilibrium status, its position becomes every destination in the goal map. The equilibrium positions for shape formation need to satisfy several requirements: a) the position of individuals should be equally distributed. This is achieved using Equation 4 with a spiky gradient kernel. b) The positions of individuals inside the desired shape are more significant than those on the periphery in achieving a smooth change of crowd shape. We therefore weight the potential field to give a higher priority to the arrival of individuals inside the crowd. c) Individuals in the interior of the shape map need to have very similar connectivity if the movements required for a change of shape are to be stable. This challenging condition can be met by sociogram links between particles. Automatic positioning is achieved by an SPH simulation of particles that represents individuals. The result is a uniformly distributed target position in goal map which meets the relative arrival priorities of individuals.

$$f_i^{pressure} = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(r_i - r_j, h) + m_i \vec{V}^{grid} \qquad (14)$$

3) *Shape maintenance*

In the simulation step, every individual moves to their destination under the control of shape maps which are painted by the user. To make the transformation between shapes smooth and continuous, we weight the accelerations of individuals who have further to go to their target positions. Ideally, every individual would reach their target position at the same time, but this is not possible, so we must allow a period for the arrival of all the individuals in a group. Each group has an allocated time period for arrival. Individual in the group can get permission to go to next destination after majority of the group has reached its target position. This allows the crowd to form an almost perfect target shape. During this process the velocity of each individual is determined by the following equation:

$$\mathrm{cV_i} = \alpha_\mathrm{i}\left[\mathrm{pV_i} + \beta_i(\mathrm{cDist_i} - \frac{1}{\mathrm{N}}\sum_\mathrm{i}\mathrm{rDist_i}) + \mathrm{aV_i^P}\right] \qquad (15)$$

where $\mathrm{cV_i}$ and $\mathrm{pV_i}$ are the current and previous velocities of the individual, $\mathrm{cDist_i}$ is the distance currently remaining to the destination, and $\mathrm{aV_i^P}$ is a weighted velocity designed to achieve the arrival priority of this individual. The ratio α_i and β_i are waiting and relative arrival time adjustments.

The shape map method allows our system to support the following crowd behaviors: a) assuring a specified shape; b) smooth shape transformation; c) separation – one group can separate into several groups with specific patterns; d) multi-shape movement – one group creates several shapes one after the others.

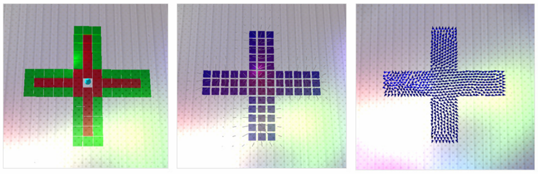

(a) (b) (c)

Fig. 7 Map and field used in the shape map technique. Shape map (a), Potential and vector field (b), and Goal map (c)

Simulation

To generate the force which will push an individual towards their goal, we use a goal map which is constructed from the shape map. Then we can generate the shortest path between two goal maps using Dijkstra's algorithm. Because we regard every node on the path and all obstacles as particles in an SPH simulation, we can unify the system within a single framework. The force driving a particle toward its goal also includes a force specified by the user, which allows interaction:

$$(16)\ \vec{f}^{\,goal}{}_i = \vec{f}_i^{\,path} + \vec{f}_i^{\,obstacle} + \vec{f}_i^{\,\mathrm{interacion}}$$

Crowds are always affected by their current circumstance. For instance, in an escape from a fire, private spaces are ignored and the crowd forces the movements of individuals. Conversely, when a crowd is sparse, individuals try to find other people to whom they can relate, forming small groups for psychological safety. To reflect this phenomenon, our system can change the length of its smoothing kernel dynamically, as suggested by Benz [2], using information from density checks. This adaptive kernel scaling makes the behaviors of individuals adapt to shifts in density. These changes affect the h_1 and h_2 radii as follows:

$$h_i = h_{i_0}\left(\frac{\rho_0}{\rho}\right)^{\frac{1}{k}} \qquad (17)$$

where h_i, h_{i_0} and ρ_0 are the initial radii and density, and k is a stiffness constant. In a basic SPH implementation we can calculate each quantity in the field by summing over all particles in the system. But such a naive implementation will have O(n^2) time complexity, making it unsuitable for a large crowd. The SPH technique can be accelerated by using a spatial hashing technique. This reduces the complexity to O(kn), where n is the number of particles in the system and k is the average number of neighbors of each particle.

At the simulation stage, our system runs with a variable selection of the individuals in the crowd to reduce the computation time. A user can change the parameters of every functional module – the sociogram links, shape maintenance system, and options such as adaptable scaling and its parameters during the simulation. This allows a user to create various styles of crowd, varying from a highly organized shape to an uncontrolled mass of people.

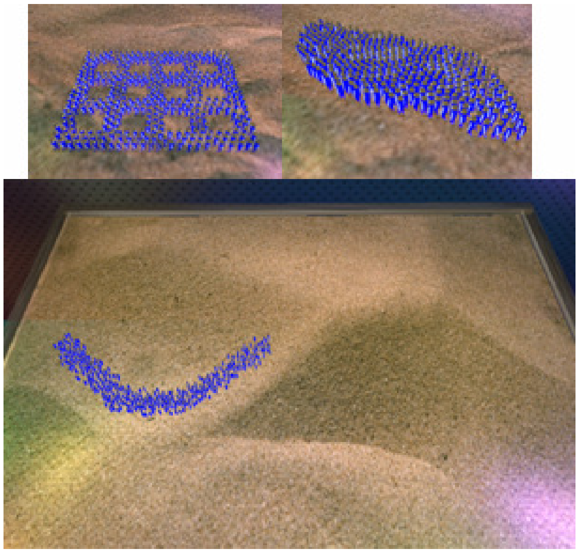


Fig. 8 An encircling army simulation in the desert

5. Implementation

We performed a number of simulations on our system, running on a 1.68GHz Intel Core2 Dual with an NVIDIA GeForce 8500 Mobile GT graphics card and 2GB memory on a notebook platform. This configuration can handle crowds counting up to 900 animated polygon agents, each constructed of 1,200 polygons, at 12 frame rate per second when using all components. The simulation was updated at every frame. Fig. 8 shows organized continuous crowd movements. 400 individuals make encircling formation on the desert terrain. Three formations are defined at preprocessing time, and during the simulation, transition among the formations is preceded smoothly in real time. This shows our stable basic movement of individuals.

Fig. 9 shows an idling crowd. 800 individuals make random movements. In this simulation, dense crowd did not show any bottleneck phenomenon at the high density center area. This shows our avoidance and group forces are working well without making any congestion.

Fig. 9 A simulation of tourists outside St. Peter's in Rome

Fig. 10 shows a movement involving several shapes. 920 individuals make circle formations one after the other. During the simulation, dense crowd shows few congestion at central high density area. The companion video demonstrates a range of behaviors including basic avoidance, pushing, waiting to go, and lane formation. These are all based on continuous fluid computations using the SPH framework. With the addition of a sociogram link force, leader and follower, collision avoidance with formation maintenance, convex hull control, natural bidirectional flow, and assembling behavior can all be produced in a stable way.

6. Conclusions

In this paper, we presented an approach to dense crowd control using authoring method based on SPH. We have used the SPH technique from fluid dynamics to express continuous flow within a detailed crowd simulation. To reflect the connectivity that occurs in a crowd, we use a sociogram model. To create a crowd that fills a given shape, we use a shape map which allows movements from on formation to another be created in an intuitive way. An advantage of our model is its ability to satisfy realistic microscopic locomotion and macroscopic crowd control with relatively few parameters. Our approach is especially useful for applications that require the simulation and animation of well-organized crowds, such as the mass game and army simulation. In the future, we would like to improve the SPH kernel function to model various small-scale movements made by individuals. In addition, we intend to integrate global path optimization with our shape map technique.

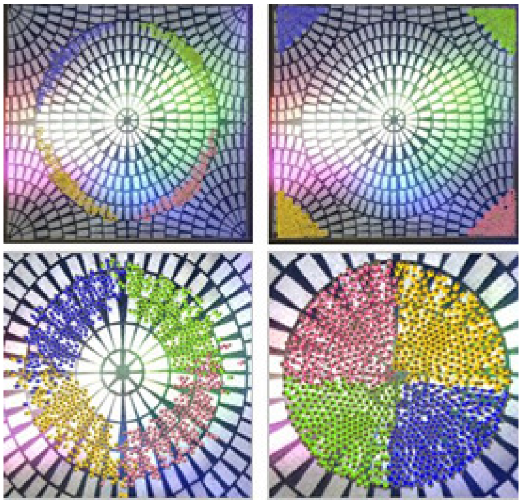


Fig. 10 Color mass game simulation

Acknowledgements

This work was supported by the Hongik University new faculty research support fund.

References

[1] Anderson, M., Mcdaniel, E., and Chenney, S., "Constrained animation of flocks," Proceeding of ACM Siggraph/Eurographics Symposium on Computer Animation, pp. 286–297, 2003.

[2] Benz, W. "Smooth particle hydrodynamics: a review," The Numerical Modeling of Nonlinear Stellar Pulsations, pp. 269-288, 1990.

[3] Chenney, S., "Flow tiles," Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 233-242, 2004.

[4] Helbing, D., Farkas, I., and Vicsek, T., "Simulating dynamical features of escape panic," Nature, Vol. 407, pp. 487-490, 2000.

[5] Hughes, R. L., "The flow of human crowd," Annual review of fluid mechanics, Vol. 35, pp. 169-182, 2003.

[6] Jin, X., Xu, J., Wang, C. C. L. Huang, S. and Zhang, J., "Interactive control of large crowd navigation in virtual environment using vector field," IEEE Computer Graphics and Applications, Vol. 28, pp. 37-46, 2008.

[7] Kwon, T. S., Lee, K. H., Lee, J. H. and Takahashi, S., "Group motion editing", ACM Transactions on Graphics,

Vol. 27, No. 3, pp. 80-88, 2008.

[8] Lee, K. H., Choi, M. G.., Hong, O. Y., and Lee, J. H., "Group behavior from video: A data-driven approach to crowd simulation," ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 109-118, 2007.

[9] Müller, M., Charypar, D., and Gross, M., "Particle-based fluid simulation for interactive applications," Proceeding of ACM Siggraph/Eurographics Symposium on Computer Animation, pp. 154–159, 2003.

[10] Rahul N., Abhinav G., Sean C., and Ming C. L., "Aggregate dynamics for dense crowd simulation," In ACM Transactions on Graphics, Vol. 28, No. 5, pp. 122:1–122:8., 2009

[11] Reynolds, C. W., "Steering behavior for autonomous characters," Game Developers Conference, 1999.

[12] Tantisiriwat, W., Sumleeon, A., and Kanongchaiyos, P., "A crowd simulation using individual-knowledge-merge based path construction and smoothed particle hydrodynamics," Proceedings of the 15th International Conference in Central Europe on Computer Graphics, 2007.

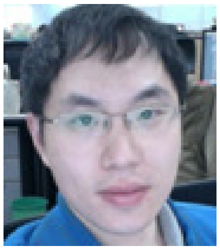
[13] Teschner, M., Heidelberger, B. and Müller, M., Pomeranets, D., and Gross, M. "Optimized spatial hashing for collision detection of deformable objects," Proceedings of the Vision, Modeling, Visualization Conference, 2003.

[14] Treuille, A., Cooper, S., and Popovic, Z., "Continuum crowds," ACM Transactions on Graphics, Vol 25, 2006.

[15] Ulicny, B., Ciechomski, P. H. and Thalmann, D., "Crowdbrush: interactive authoring of real-time crowd scenes," Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer Animation, pp. 243-252, 2004

[16] HwaMin Lee, Min Hong, "An Analysis of Learning Effects for Smartphone based Serious Game Applications," Journal of Future Game Technology, 2011. 9, Vol 1, No 1, pp. 1-8.

[17] Shin Jin Kang, Soo Kyun Kim, Sung Hyun Cho, "Procedural Game Level Generation with Interactive Evolutionary Computation," Journal of Future Game Technology, 2011. 9, Vol 1, No 1, pp. 9-20.



Shin Jin Kang received Ph.D. degree in Computer Science from Korea University in 2011. Since 2003, he has worked at Sony Computer Entertainment Korea and NCsoft as a lead game designer in various video games and MMORPGs including AION. He is now a professor at the School of Games at Hongik University. He is also the technical advisor of NCsoft.

SooKyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He is now a professor at Department of Game Engineering at Paichai University, Korea. Professor Kim has published many research papers in international journals and conferences. Professor Kim has been served as Chairs, program committee or organizing committee chair for many international conferences and workshops; Chair of ICCCT'11, ITCS'10, HumanCom'10, EMC'10, ICA3PP'10,,FutureTech'10, ACSA'09, Em-Com'09, CSA'09, CGMS'09, ISA'09, SIP'08,FGCN'08 and so on. Also Professor Kim is guest editor of the International Journal of "IET Image Processing" and "Multimedia Tools and Applications".

His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS and KIIT.