

Group Emotion Modelling and the Use of Middleware for Virtual Crowds in Video-Games

Olivier Szymanczyk¹, Grzegorz Cielniak¹

Abstract. In this paper we discuss the use of crowd simulation in video-games to augment their realism. Using previous works on emotion modelling and virtual crowds we define a game world in an urban context. To achieve that, we explore a biologically inspired human emotion model, investigate the formation of groups in crowds, and examine the use of physics middleware for crowds. Furthermore, we assess the realism and computational performance of the proposed approach. Our system runs at interactive frame-rate and can generate large crowds which demonstrate complex behaviour.

1 INTRODUCTION

Large virtual worlds with open-world game-play have been widely used for recent video-games, as they are often graphically impressive and help achieve an immersive experience. However, these genres of games have often been criticised for feeling lifeless. Particular examples of games which suffer from a lack of ‘life’ are “FarCry 2” [1], “Crysis” [2] and “Grand Theft Auto VI” [3]. One way of improving the experience of playing of these types of games could involve the simulation of virtual crowds. Implementing crowds in real and non-real time applications is a difficult and challenging task. One of the greatest challenges is the constant need for efficient variety management at every level, whether it is visualisation, motion control, animation, behaviour or sound [4]. Individuals in a crowd should look, move and react differently [4]. Even though the available computing power is constantly growing, it is still not possible to create real-time simulations filled with thousands of simulated agents necessary to achieve a realistic result. Current research of virtual crowds ranges from the visual diversity of simulated individuals [5], rendering [6], to the crowd behaviour generation [7, 8] and animation [9, 10]. Despite significant progress in these areas there are still several issues which need to be addressed (e.g. performance issues when simulating larger crowds with complex behaviour [7, 8], individuals passing through each other [10]).

In recent years, several solutions to develop virtual crowds to populate video-game worlds have been realised. For example, the video-game “Assassin’s Creed” [11] simulates crowded medieval cities during the third crusade in the Holy Land. The cities are filled with merchants, knights and peasants who fulfil daily routine activities (up to 150 onscreen characters [12]). The player can interact directly with the crowd. In return they generate realistic reactions as part of the game-

play. “MotoGP07” [13] also uses virtual crowds to simulate the crowd excitement of motorsport racing events. Even though the player does not interact with the crowd, the look and feel of the game is improved [13]. Although solutions for virtual crowds in video games exist [10, 12, 13], further research in the use of simulated crowds is required to achieve genuinely believable virtual worlds in video-games. Virtual crowds have already been widely used in other non video-games related applications. For example, crowd safety applications use real-time evacuation simulations to identify possible health and safety hazards. Another area where large virtual crowds have been used are non-real-time productions such as feature films including “Lord of the Rings” and “I, Robot” [14]. Virtual crowd solutions in those domains could be re-implemented in video-games situations. However, due to their high computational requirements [4], non real-time solutions have limited applicability to videogames.

In this work, we propose a method for simulating a real-time crowd in an urban context based on Millington’s goal-oriented algorithm [15] and investigate the feasibility of PhysX for behaviour modelling and animation. Furthermore, we suggest an approach to recreate the behaviour of groups in crowds. We demonstrate that with the proposed method up to 30,375 individuals that show complex behaviour within interactive frame-rate (>30FPS) can be simulated. Both qualitative and quantitative evaluation of the realism and performance of our simulation are presented and discussed. Our system is comparable to other real-time crowd simulations [8, 10]. This paper is structured as follows: we start with a review of related work in Section 2. Section 3 presents our virtual world including a method for crowd behaviour modelling. Section 4 details the implementation of our algorithm. Various metrics used for evaluation of our approach are presented in Section 5. Section 6 presents the results. The paper concludes with discussions about future work in Section 7.

2 RELATED WORK

Several methods for implementing virtual agents have been proposed [9, 10, 16, 17]. Reynolds [7] simulates a flock of birds. They navigate according to their perception of the environment, a set of predefined behaviour rules and the simulated physics that dictates a bird’s motion. The Sony Playstation 3 (PS3) multiprocessor architecture was recently used to further improve his approach [10] and a virtual crowd up to 15,000 agents at 60FPS was simulated. It was concluded that further hardware optimisations to generate larger crowds at interactive frame-rate were required and he proposed improvements in the collision avoidance approach to enhance the quality of his simulation.

¹ Lincoln School of Computer Science, University of Lincoln. LN6 7TS, UK. Email: {oszymanczyk, gcielniak} @lincoln.ac.uk



Figure 1. Our simulated world with 25,000 agents at interactive frame-rate (30FPS). Individuals and goals (e.g. shops) are rendered as textured 2d quads. The icons above each individual represent their current goal, icons above shops symbolises their offered actions and service.

Helbing's Social Force Model [9] describes pedestrian dynamics as a force model. This model takes into account an individual's acceleration force, possible repulsive effects, interactions with other individuals, boundaries and other possible attraction effects of pedestrian traffic. This approach has been implemented and tested. However some undesired features (unnatural shaking and vibrating) were reported [18].

A Cellular Automaton (CA) is an effective technique for modelling complex emergent collective behaviour of crowds [17]. The principle of CA is the division of the virtual world in a checkerboard pattern. Each cell can contain up to one individual, and contains localised neighbourhood rules of occupancy. The CA layers can be extended with sets which contain information such as behavioural rules of an individual that is occupying the cell. A crowd simulation using extendable layers CA was also developed [8]. It generates around 40,000 agents within interactive frame-rate on a 1.6 GHz computer with 2GB RAM. Despite CA models being simple and fast to implement, their disadvantage is the disallowance of individuals to come in direct contact with each other. Each agent can only move to a free adjacent cell. CA crowd are considered to have unrealistic animation which can only handle low to medium densities [18].

The use of a biologically inspired emotion model to generate realistic behaviour of virtual individuals has been widely discussed [16, 19, 20]. Emotions influence and dictate the way many decisions are taken, and how plans for the future are made by individuals. A framework for modelling emotionally driven context aware crowd behaviour was implemented in a train station simulation [18]. The conclusion argued that it is reasonable to assume that emotions widely increase the believability of simulated human behaviour. Delgado-Mata [16] also used an emotional model to simulate the behaviour of autonomous creatures. Agents communicated their emotions with help of artificial pheromones. He concluded that the use of emotions offer complex individual behaviour which in return offers greater believability. Emotion modelling has also been used in the popular video-game "The Sims", in which a small

number of individuals use personality descriptors to drive their actions and goal selection [15]. Millington [15] developed a goal-oriented behaviour algorithm to implement this goal/action selection behaviour. Characters are presented with a range of possible actions and the algorithm chooses the one that best meets their immediate needs.

Recently, in the video-game industry, game developers have started to take advantages of the use of middleware physics engines such as Havok [21] or PhysX [22] to simulate in-game characters and objects. The advantages of middleware physics engines are their high computational performance which utilise multithreading and GPUs, and their accurate physics simulation [23]. For example, the video-game "Assassin's Creed" uses Autodesk Maya's Human Inverse Kinematics middleware for collision detection and character interaction with the environment [12].

In our approach we investigate the feasibility of emotion modelling (using Millington's goal-oriented algorithm [15] presented in Section 3) and the usability of PhysX in a scenario that involves a larger number of in-game characters. Furthermore, we investigate the influence of groups (shared emotions between individuals) on the behaviour realism and computational performance.

3 THE VIRTUAL WORLD

The virtual world W consists of a set of N individuals I , M goals S and L actions A :

$$W = \{I, S, A\}, \quad (1)$$

where $I = \{I_1, \dots, I_N\}$, $S = \{S_1, \dots, S_M\}$ and $A = \{A_1, \dots, A_L\}$. Each Individual I_n is described by its physical state \mathbf{p} , the emotional state \mathbf{e} and the current goal g :

$$I_n = \{\mathbf{p}, \mathbf{e}, g\}. \quad (2)$$

The physical state $\mathbf{p} = \{position, velocity, mass, shape\}$ describes the position, velocity, mass and shape of an individual. These quantities are used to model character movement.

The emotional state $\mathbf{e} = \{e_1, \dots, e_K\}$ represents the current values of character's mood (e.g. weights describing the current strength of different emotions). The emotional states are used for goal selection. The goal $g \in S$ is the current goal of an individual.

Each goal S_m is described by its physical state \mathbf{q} , its available actions \mathbf{a} and a service limit γ :

$$S_m = \{\mathbf{q}, \mathbf{a} \subseteq A, \gamma\}. \quad (3)$$

The physical state $\mathbf{q} = \{position, influence_range\}$ describes the position and influence range of a goal. There is a limit γ on a number of individuals that a goal can service.

Each action A can influence (positively or negatively) any number of available emotions \mathbf{e} . Each single action A_o is composed out of κ amount of emotional influences μ :

$$A_o = \{\mu_1, \dots, \mu_K\}. \quad (4)$$

A goal is represented by an influence area in which a subset of all available actions can be taken (Eq. 3). For example, consider a kettle as a goal. The actions of a kettle are 'switch on/off, fill/pour water'. If a character's goal is to 'get boiling water', they need to go into the range of the kettle and use its available actions to achieve their goal. In the urban context of our virtual world, a goal is represented as a shop, which offers certain actions e.g. food related actions. Individuals in our virtual world are emotion driven. They eat when they're hungry and drink when they're thirsty. If individuals are hungry, their goal would be to go into the influence range of a shop which offers an action related to food and to perform this action to satisfy their hunger.

Action calculation	<p>For $i = 1$ to L</p> $d_i = \sum_{k=1}^P (\mathbf{e}_k + A_i(\mu_k))^2$ <p>$bestaction = \arg \min(d_i)$</p>
Goal calculation	<p>For $j = 1$ to M</p> <p>If $(S_j \subseteq bestaction) \& \overline{\gamma_j}$</p> $g = S_j$

Algorithm 1. Discontent/Action/Goal selection algorithm for one individual I_n .

Algorithm 1 explains the goal-oriented algorithm for one individual I_n that result in the emotional driven behaviour. To find which action to take, an individual calculates which action is the best suited for his current set of emotions. The aim is to find the action which would reduce the discontent value as much as possible. An individual looks at all possible actions present in the world and predicts (using Algorithm 1) which one produces the lowest possible discontent value. This action is then considered as optimal. The final step for an individual is to find a goal which has not reached its service limit that offers this action.

When the current goal is selected, an individual turns and moves towards their selected goal. When a collision with other individuals occurs, the collision is resolved, and then the individual pursues their route towards the goal again. An individual takes the chosen action while passing through the goal's influence range. The more crowded the goal is, the longer it takes to complete the action. It simulates the complex behaviour of busy urban areas. When leaving the goal, the action is completed and the person's emotional state is updated. At this point the new goal is calculated using Algorithm 1.

3.1 Group Emotion Modelling

We introduce the concept of groups in our virtual world. In this context, group can be considered to contain either a small number of individuals (e.g. friends, family) or a larger category of people (e.g. participants of a larger event such as a concert, riot or demonstrations). Each member tends to have the same goal as other members of their group. If we split our virtual crowd into groups, we can allow one instance of Algorithm 1 to be applied between a subset of U individuals who are in the same social group. Each individual in the group has their emotions as equally weighted as other group members, thus allowing Algorithm 1 to be used once to calculate the next goal for all individuals in the same group. Individuals in a group do not share the same physical state. While using groups, the world W (Eq. 1) is redefined to contain V groups of individuals G :

$$W = \{G, S, A\}, \quad (5)$$

where $G = \{G_1, \dots, G_V\}$. A group G is composed out of U individuals who share the same goal g and emotional state \mathbf{e} :

$$G = \{I, g, \mathbf{e}\}, \quad (6)$$

and $I = \{I_1, \dots, I_U\}$. The presented concept can be supposed to mimic social groups. In addition the complexity of the algorithm is reduced.

4 IMPLEMENTATION

We implemented the virtual world in Visual Studio 2008 using C++ with the help of the GLUT libraries and PhysX (version 2.7.3). No manually coded optimisations, e.g. no multi-processor support, was utilised at this point however we did use the hardware support for PhysX. A simple rendering system for evaluation purposes was implemented. The object-oriented code allows every object of the virtual world W to be parameterised.

The physical state \mathbf{p} of individuals I is handled by PhysX (NxBoxShapeDesc class). The emotional state \mathbf{e} consists of five different components which symbolise the value of strength of Hunger, Thirst, Health, Time and Bathroom. Their values are initialised randomly at start-up of the application. Upon initialisation, individuals are positioned in a square pattern around the virtual world.

The physical state \mathbf{q} of goals S is also handled by PhysX. (NxBoxShapeDesc class, flag NX_TRIGGER_ENABLE set). Each goal contains a randomly defined subset of available actions \mathbf{a} . Goals are positioned randomly around the virtual world. In our implementation we defined $M = 100$ goals S and $L = 23$ actions. The number of Individuals N was varied between 1

and 65,000. Actions A are related to the emotions e of individuals. They influence positively and negatively the emotional needs of hunger, thirst, health, time and bathroom. For example, the ‘get a large drink’ action greatly satisfies the thirst strength of an individual, but negatively affect the bathroom value. These values were manually generated.

The size of the virtual world was defined and rendered as a 2d surface of 500x500m. There are no physical urban obstacles (e.g. no cars, no buildings, no roads). Goals were rendered as 2d shops and individuals were rendered as 2d characters. Actions and the current goal of individuals were rendered as small 2d icons (see Figure 1)

5 EVALUATION METRICS

To assess the believability of our approach, we applied the following set of evaluation metrics. The first metric is based on the fact that there has been shown to be a linear relationship between the average speed and the flow of the pedestrian density in crowds [24]. The maximum speed of pedestrians is achieved when the density is at its lowest. As the density increases, individuals slow down to finally reach the point of congestion.

Levels of service were used to define the quality of the pedestrian flow. This metric details a person’s freedom to choose their speed, their possibilities to overtake, their ability to walk in the direction opposing the major flow and their capacity to manoeuvre without conflicts [24]. Seven different levels of service were defined: jammed, congested, crowded, constrained, impeded, unimpeded and open (see Figure 2).

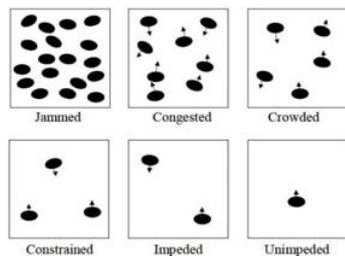


Figure 2. Illustrative depiction of the levels of service [24]. Each blob represents an individual with direction of movement depicted by arrows.

The formation of platoons [24] and the formation of lanes [9] is another observable feature of crowd behaviour. Platoons are temporary clusters of pedestrians behind slower groups e.g. people leaving a train. The formation of lanes occurs in crowds moving in opposite directions. In some circumstances it is possible to observe the formation of lanes of people moving in the same direction. The reason for lane formation is the related decrease in the frequency of required deceleration and avoidance manoeuvres, which increases the efficiency of the pedestrian flow [9].

6 EXPERIMENTS

Our objective is a qualitative and quantitative evaluation of our virtual world. To achieve this, a series of experiments were conducted involving different conditions while varying and controlling the most important system variables. We evaluate the

realism of the simulation using a set of metrics presented in Section 5 based on informal observation. Furthermore, we compare and evaluate the computational performances by measuring the frames per second (FPS) performance in different scenarios. The scenarios are: varying number of individuals, breakdown performance between different parts of the system and effect of group modelling on the performance. We also describe the effect of group modelling on crowd behaviour. The experiments were carried out on an Intel Core 2 Quad CPU Q6700@2.66GHz, 2048MB Ram, GeForce 8800 GT (256MB).

6.1 Crowd Realism

We observed a relationship between crowd density and average speed. The higher the density, the slower a person is (see Figure 3). There are considerable differences between the estimated speed/density values from [24] and values obtained in our simulation. This is most probably caused by the crude shape of individuals in our system.

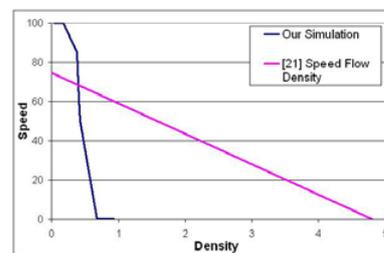


Figure 3. Average speed/number of people in an area (crowd density).

Table 1 compares the different levels of service with values obtained in our application. The overall tendency follows the values reported in [24]. However, results show that the denser an area is, the larger the difference between our system and the values specified in [24].

levels of service	Area/ped	avg speed (our world)	avg speed from [21]
Jammed	0.2 ~ 1.0	<40%	<70% of free flow
Congested	1.0 ~ 1.4	~ 60%	75% of free flow
Crowded	1.4 ~ 2.3	~ 75%	>80% of free flow
Constrained	2.3 ~ 3.7	~ 85%	Approaching free flow
Impeded	3.7 ~ 12	~ 99%	Virtually as chosen
Unimpeded	12 ~ 50	100%	As chosen
Open	>50	100%	As chosen

Table 1. Levels of service of the virtual crowd flow.

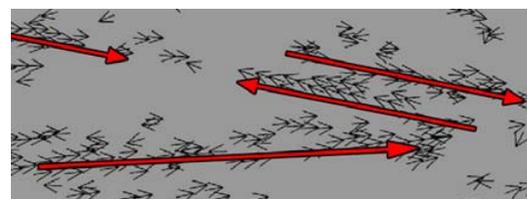


Figure 4. Examples of pedestrian flow (red arrow), each black arrow represents a pedestrian and their walking direction.

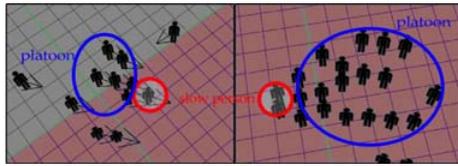


Figure 5. Temporary platoons (blue) and individuals causing it cause (red).

The formations of platoons and lanes can be observed in our application (see Figure 5). Figure 4 shows an example of the occurrence of pedestrian flow in our application

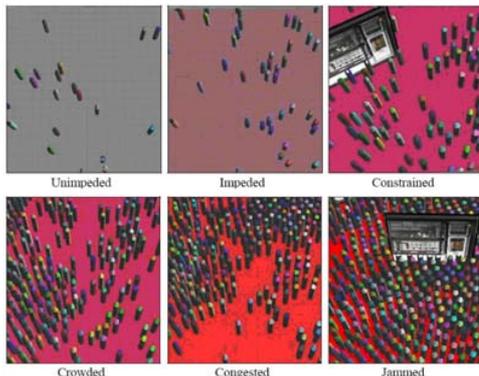


Figure 6. Levels of service in our application.

If we visually compare the levels of service in Figure 4 with the depicted illustration in Figure 2, similarities can be observed.

There is a difference of resulting behaviour between the world W defined by Eq. 1 (no groups) and Eq. 5 (group modelling). In the first case individuals fulfil their goal on their own and the formation of groups is temporary. In the second case, individuals stay in their attributed group. This behaviour is mostly visible with larger group. The distribution of individuals through of the simulated area is also influenced. A more uniform density is obtained when using smaller groups or the ‘no groups’ approach (see Figure 7).

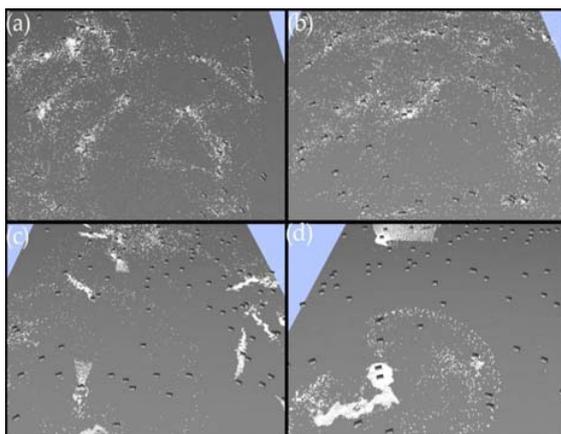


Figure 7. Our simulation after 10,000 iterations. (a) no groups (b) 50 individuals per group. (c) 500 individuals. (d) 15000 individuals.

6.2 Computational Performances

There are different computational costs related to different parts of the system (e.g. PhysX, Rendering, Emotions). For a crowd consisting of 25,000 individuals, the data collected showed that 52.07% of the update time is used to compute the discontent/action/goal algorithm, 18.35% is spent on rendering and 29.58% is used by PhysX.

Figure 8 shows the average performance of the application with respect to the number of simulated individuals (no groups, no rendering). The relationship between the number of individuals N and the performance of the application is exponential and indicates a possible bottleneck of the prototyped approach.

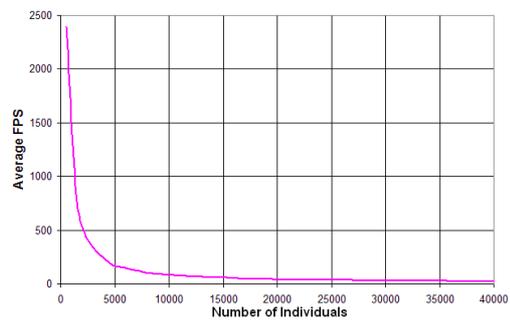


Figure 8. Average FPS performance with respect to the number of individuals N .

Figure 9 shows the computational performance between values of 60 to 30 FPS. There are different data sets representing, the computational effect of differently sized groups on the performance.

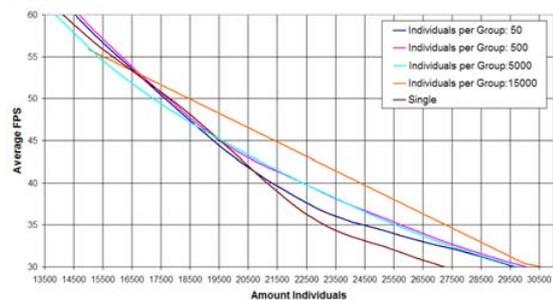


Figure 9. Performance of the application, no groups and groups of different sizes.

The FPS performance of the ‘no groups’ data set is the lowest. The maximum number of individuals the ‘no groups’ approach could handle on our test machine above 30FPS was 27,000. If groups are used, the performance gradually increases. The total number of individuals that can be simulated at interactive frame-rate increases as shown in Table 2. In other words, the larger the group size is, the higher the increase in performance.

Group size	Total individuals	Increase
50	29.500	7.62%
500	29.750	8.40%
5000	30.000	9.16%
15000	30.375	10.28%

Table 2. Performance increase when using different group sizes compared to no groups.

7 CONCLUSION AND FUTURE WORK

We have developed and evaluated a virtual crowd which uses a biologically inspired human emotion model, investigated the formation of groups, and examined the feasibility of PhysX in a large crowd simulation. The results demonstrate that PhysX [22] and Millington's goal-oriented algorithm [15] can be used to simulate a large virtual crowd. The formation of groups in real crowds was implemented by using the concept that emotions are shared between each member of the group. The use of groups improved the computational performance of the virtual crowd while also affecting the behavioural results of each individual. In our simulation, each person in a group tends to stay within close range of their attributed group, thus visibly reducing the spread of the crowd over the simulated virtual world. This behaviour was particularly visible in situations when larger groups (e.g. 15,000) of people were used. For a particular application, consideration of the group sizes should be made as they affect the behaviour. Larger groups of individuals can be used to model participants of a larger event whereas small groups can be used to simulate families or groups of friends.

On our test machine, we were able to generate up to 30,375 agents at interactive frame-rate. Although a precise comparison with other virtual crowd system is difficult (mainly due to hardware differences) our implementation offers similar performance results as virtual crowd applications used in [8, 10]. The qualitative evaluation showed that in a crowd of 25,000 individuals, 52% of the update time is used to compute the discontent/action/goal algorithm, 18% is spent on rendering and 29% is used on PhysX of the computational performances. The quantitative evaluation shows that our developed crowd demonstrated realistic crowd aspects such as the formation of lanes and platoons and the dependency between density and average speed.

In a future work, it would be interesting to investigate more complex models of group emotion. Currently, our application only allows groups of the same size to be used. Therefore, implementing the ability to vary the size of the groups would augment the realism of the application. The computational performance of the discontent/action/goal selection algorithm could be improved with the help of Millington's discussions about the computational performances of his algorithm [15]. Furthermore, it would also be interesting to apply a group model approach using different crowd behaviour and animation model. In addition, the virtual world could also be improved by adding an appropriate rendering system and complex urban elements (e.g. building, roads). Collision detection and response approaches could be improved as currently individuals seem to be 'pushing' other crowd members, giving the feeling that most individuals are busy, late or unmannered. More extensive metrics on crowd behaviour realism could be investigated, and the results used to further improve the system.

8 REFERENCES

- [1] Ubi Soft, Farcry 2, <http://farcry.uk.ubi.com/> (2008)
- [2] EA Games, Crysis, <http://www.ea.com/games/crysis> (2007)
- [3] RockStar Games, Grand Theft Auto IV <http://www.rockstargames.com/IV/> (2008)
- [4] Thalmann D. and Musse S.R., *Crowd Simulation*, 1st ed. Springer, (2007)
- [5] McDonnell, R., Larkin, M., Dobbyn, S., Collins, S., and O'Sullivan, C. Clone attack! Perception of crowd variety. In *ACM SIGGRAPH pp 1-8* (2008)
- [6] Tecchia, F. and Chrysanthou, Y. Real-Time Rendering of Densely Populated Urban Environments. In *Proceedings of the Eurographics Workshop on Rendering Techniques* pp 83-88 (2000).
- [7] Reynolds, C. Flocks, herds and schools: A distributed behavioral model. In *Proceedings 14th annual conference on Computer graphics and interactive techniques*, pp 25 - 34 (1987)
- [8] Banerjee, B., Abukmail, A., and Kraemer, L. Advancing the Layered Approach to Agent-Based Crowd Simulation. In *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation* (2008).
- [9] Helbing D, Molnar P., Farkas I. J., and Bolay K. Self-organizing pedestrian movement. *Environment and Planning B*, 28:361--384, (2001)
- [10] Reynolds, C. 2006. Big fast crowds on PS3. In *Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames* (2006)
- [11] Ubi Soft, Assassin's Creed, <http://assassinscreed.uk.ubi.com/> (2006)
- [12] Monret, J. Code Hashashin Scimitar Engine ++ <http://ib0ssum.blogspot.com/search?updated-max=2008-12-19T23%3A10%3A00-05%3A00> (2008)
- [13] Jefferies, D. Postmortem: Black Rock Studios' MotoGP'07. http://www.gamasutra.com/view/feature/1971/postmortem_black_rock_studios_.php (2007)
- [14] Kavan, L., Dobbyn, S., Collins, S., Žára, J., and O'Sullivan, C. Polypostors: 2D polygonal impostors for 3D crowds. In *Proceedings of the 2008 Symposium on interactive 3D Graphics and Games* pp 149-155 (2008)
- [15] Millington I., *Artificial Intelligence for Games* (The Morgan Kaufmann Series in Interactive 3D Technology), Morgan Kaufmann Publishers Inc., San Francisco, CA (2006)
- [16] Delgado-Mata C.. *Emotion Signalling In Multiple Intelligent Virtual Agents for Believable Artificial Animals*. PhD thesis, University of Salford (2004)
- [17] Adler, J. Cellular Automata Microsimulation for Modeling Bi-Directional Pedestrian. In *Transportation Research Journal Part B*, Vol. 35/3 pp 293-312 (2001)
- [18] Pelechano, N., Allbeck, J. M., and Badler, N. I.. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM Siggraph/Eurographics Symposium on Computer Animation* pp 99-108 (2007)
- [19] D. Isla, B. Blumberg. New Challenges for Character-based AI for Games. In *Proceedings of the AAAI Spring Symposium on AI and Interactive Entertainment*, Palo Alto, CA (2002)
- [20] Dey, P. and Roberts, D. A Conceptual Framework for Modelling Crowd Behaviour. In *Proceedings of the 11th IEEE international Symposium on Distributed Simulation and Real-Time Applications* pp 193-200 (2007).
- [21] Havok Inc, Havok Physics engine, <http://www.havok.com/> (2009)
- [22] Nvidia, PhysX, http://www.nvidia.com/object/PhysX_new.html (2009)
- [23] Ballanytyne, R. Collision Overload: Reducing the Impact in Real-time Physics. <http://www.doc.ic.ac.uk/teaching/projects/Distinguished07/ianballanytyne.pdf> (2007)
- [24] F. Feurtey, *Simulating the collision avoidance Behavior of pedestrians*, Master's thesis, University of Tokyo, Department of Electronic Engineering, (2000)