Forschungsseminar aus Computergraphik und Digitaler Bildverarbeitung (SS2009)
Jana Dadova, Norbert Ketterl (Editors)

Crowd Simulation

J. Dadova[1] and N. Ketterl[1]

[1]Institut für ComputerGraphik und Algorithmen, TU Wien, Austria

---

Abstract

The ABSTRACT is to be in fully-justified italicized text, between two horizontal lines, in one-column format, below the author and affiliation information. Use the word "Abstract" as the title, in 9-point Times, boldface type, left-aligned to the text, initially capitalized. The abstract is to be in 9-point, single-spaced type. The abstract may be up to 3 inches (7.62 cm) long.

Leave one blank line after the abstract, then add the subject categories according to the ACM Classification Index (see http://www.acm.org/class/1998/).

---

1. Introduction

Both crowd behavior and crowd animation are quite newish research topics in computer science. Within the last three decades a wide range of scopes are focused on. Crowd animation is used in entertainment, commercials and simulations as well. The fields of application in simulation range from crowd control and traffic density to evacuation planning and safety aspects. The primary objective is to achieve a high-level simulation of both socially unconnected collectives and connected groups both in behavioral terms and in a visual way.

Dealing with crowd behavior focuses on much more topics than considering only a singular scientific discipline. Therefore the research field of crowds should be discussed in an interdisciplinary way. However most state-of-the-art computer models dealing with crowd animation cover either only graphical, physical or sociological foci. A plausible visual environment have to extend the level of pure geometrical models. Realistic simulation of living entities (e.g. plants, animal, human) does not only belong to the transformation of physical laws into virtual worlds but first and foremost on the interface between computer science, biology and psychology. Simulation of fundamental life mechanisms also includes natural processes (e.g. birth, death, growth, natural selection, evolution).

The areas that discuss problems in Crowd Simulation can be divided into following cathegories. According to the question they try to answer:

- How to create a heterogenous, naturally looking crowd?
- How to make an effective and realistic animation?
- How to make crowd behave naturally?
- How to include a crowd in an environment?
- How to render a crowd?

Following the questions above, there is a better way to outline different topics separately. A coarse prospect about crowd animation deals with:

- Individual behavior
- Group behavior
- Rendering and appearance
- Interaction within groups
- Performance

The list above could not be seen as complete. There are many areas which overlaps in several kinds. For example achieving a plausible physical impression and realistic graphical rendering, both appearance and behaviour must be taken into account.

Like in the movie Titanic [OCD+02] there is a high demand going on from realism of behavior to realism of visualisation. Also going from individual behavior to collective behavior. These high level-of-detail leads to non-realtime application because of the very high complexity in the selected areas. For safety aspects very dangerous situations are rendered using virtual crowds, in order to reduce the risks of stuntmen and actors getting hurt.

The following subsections present a short overview of the special cases mentioned above:

© ICGA/TU Wien SS2009.

1.1. Individual behavior

A single character can be visualised using triangle meshes. A common approach is to use a content creation tool like Maya, Cinema4D or Blender. With this tools either low-poly or high-poly models can be created. The finished models have to be extended with texture mapping and bump mapping techniques to achieve a more realistic look. For the animation process there are different methods available. For single character animation, skeletal animation in conjunction with keyframe animation is widely used. Hartmen and Benes [HB06] refer to Maya modelled birds which are animated using keyframes. The main challenge using both approaches together is that the animation process must be done manually for all parts of the skeleton for the whole keyframes cycle. To achieve a more realistic movement in a shorter period of time motion capturing is widely used nowadays. The walking cycle animation is extracted from movies or recorded via computer vision. For motion capturing there are two well-established techniques. In addition to optical markers which are placed on joints and fingertips there is a non-optical technique where sensors and actuators are attached to a body-suit. The recorded marker positions have to be mapped to joint angles which are stored in databases. These data are being used in the animation of characters. The optical technique was used in Lord of the Rings - The two towers as seen in figure 1 to animate the Gollum more realistic.

Figure 1: *Motion capturing for Gollum. Image courtesy of Scott Remington [Rem03].*

1.2. Group behavior

The main focus within this subsection covers the intelligence of the crowd behavior model. It is used in situations with low demand for graphical quality but strong emphasises on sociological and safety aspects. Realism of behavior is widely used in simulations of real-life scenarios in virtual-reality applications. An important aspect to be dealt with later in this paper is the architectural planning for daily life purposes. To simulate the behavior of large groups in emergency or panic situations it is less important having good looking characters than checking the safety and robustness of the architectural plannings in case of fire or evacuation processes.

There is a tradeoff between simple characters and complex characters within any crowd. While simple characters are much easier to establish and also more efficient to evaluate more complex characters offer a more realistic crowd behavior is much harder to integrate and maintain within the whole collective.

Basic approaches in crowd animation relate mainly on particle systems. Within the simplest case all elements will move together in one direction at a constant delta time. These results are far away from reality and will look like a military review. Any actor in a large group is like as two peas in a pod. Particle systems can easily be used for animal crowd animation. Within an animal formation like a flock or a herd, every individual animal can be represented by one particle. Every single animal acts similar to its neighbors.

A much more complicated topic deals with different states of each actor and how one character could influence the other or the others. There is hardly no external leading force necessary for the flock behavior, for example when huge crowds of people are set to claustrophobic situations.

The behavior of animal and human crowds differs from each other.

For animal herds it is important to have a close proximity between the individual members of the group while they are changing direction or velocity. Simultaneously the group has to avoid collisions both with other animals and obstacles of the environment as Courty et al. depict in their paper [CM05]. To simulate a flock of birds it is important to know which species should be simulated. In the case of simulating migratory birds there is important to recognize the V-formation of the birds within a large compound. This kind of formation reduces the energy every bird has to spend during their long-term flight [HB06]. Also the leader of the formation will be exchanged circular to gain an almost equivalent energy distribution within the whole flock.

The realistic behavior of large human crowds remains a big challenge. A human spectator can easily recognize slightly unnatural movements of simulated humanoid characters. To achieve a high level of realism also gestures and interaction between individual characters are very important [MMKI02]. Both verbal and non-verbal communication are key features of the interaction with other humans.

There is a big difference between crowd activities in quite tight and more loose environments. In very tight environments like a concert or a football match people could be squashed together so an ordinary collision avoidance does not work optimal. Furthermore people bump against each other or converse to each other in uncoordinate manner. If the crowd is more loose some people may stand close together maybe talking to each other or holding hands. A key

© ICGA/TU Wien SS2009.

feature is a more chaotic behavior to act more random and look just more lifelike.

1.3. Rendering and appearance

Both in computer games and movies the realism of good looking crowds is much more important than a physical plausible environment.

For environmental planning there is are growing expectations for added realism. Especially for architectural plannings and designs of cultural heritage there is a huge demand for getting some kind of "life" within the visualisation of environments. Within urban environments like architectural pre-visualisation it looks much more natural if some individual humans are present than looking on a deserted atmosphere well known from ego-shooters like Fallout3 [Sof09] in figure 2.

Figure 2: *Screenshot from Fallout3: Shows a ghosted atmosphere. Image courtesy of [Sof09]*

Especially in VR (Virtual Reality) applications there is an increasing need for realistic immersion. In case of VR rehabilitation projects (e.g. crowd phobia, performance anxiety) it is very important to achieve a plausible immersive environment where the patient feels some kind of reality. The movie industry also focusses on using high-quality graphics for producing large-scale crowds for reduced costs and flexibility both in animated films (e.g. Madagascar [Ker05]) and real films (e.g. Lord of the rings trilogy [Rem03]).

1.4. Interaction within groups

By using crowd animation in realtime applications many restrictions must be considered. The main challenge to be dealt with is interactivity. The players character has to interact with the environmental depending on user input. If there are non-player characters (NPC) they are not controlled by human. But mostly they have to be controlled depending on user interaction. They have to react immediately to the current game state. These constraints are expensive for calculating in real-time.

Thus, in state-of-the-art games with detailed characters they act very individual and do not appear in large crowds, but mostly only few at a time with high-quality model details, animation and visualisation.

Figure 3: *Screenshot from Fifa Soccer 09: Shows audience as billboards. Image courtesy of [Spo09]*

For an increasing number of characters the frame-rate in real-time systems (e.g. FIFA Soccer, NHL Hockey) drop to an awkward level. The main focus is on the high rendering quality of the stadium and the football players. In those games the audience is mostly represented via billboards or textures as seen in image 3. In this case itt's possible to display a large crowd but animation of the crowd is rather impossible.

Unlike realtime applications, in the field of non-realtime applications it only can be dealt with global control scenarios or pre-scripted local events at runtime. The behavior of the whole crowd and there individual characters is scripted precisely in advantage. On the contrary to realtime applications there is no interactivity possible. It exclusively acts as an iterative full-run process like described by Braun et al. in [BBM05].

1.5. Performance

Because of both - limited computational power and the complexity of the overall mapping from real life to their virtual abstraction - there is no all-in-one device for every purpose possible.

Rachel McDonnel depicts in her Dissertation [McD06] that the game industry is the primary market for realtime crowd simulations. Therefore there is a huge demand for simplification techniques both in motion and geometry. Any modern graphics card is limited on the number of polygons which can be processed per second. Reducing the number of polygons per crowd is no good choice for displaying a collective group in a realistic way. A better approach is to generate several levels-of-detail (LOD) depending on how far is the distance between viewer and the object of interest. Objects closer to the viewer are rendered with more complexity

© ICGA/TU Wien SS2009.

than objects not in the main focus of the viewer. This could be achieved with progressive meshes to prevent the meshes from "popping" [SW08] between several levels-of-detail.

For motion techniques there is also some level-of-detail possible as O'Sullivan wrote about in [OCD+02]. Characters next to the viewer need very detailed motion and have to succeed plausible behavior interacting with others. For objects farther away from the center of interest could change their activities random and at different intervals for a believably natural behavior.

The rest of the article is organized as follows. In section 2, we provide the related work to the topics of crowd animation. Section 3 deals with local behavior of crowds and there is also a focus on sociological aspects of individual members of a crowd. Chapter 4 gives an overview over path planning algorithms for global crowd behavior.

2. Related Work

There have been many papers published on Crowd animation and simulation topics. In this work we would like to mention only few, important ones from our perspective. When simulating a crowd the following pipeline is proposed:

- Placement - defining a positions of individuals.
- Geometry of individuals - creating a geometry for participants in a crowd, usually in heterogeneous way.
- Animation (high-level) - overall movement of a crowd using particles or defining a behavior.
- Animation (low-level) - animating individual gestures.

2.1. Generation of crowd

When the distributed models are created, they need to be placed in a space. This step is called layout creation. Positions of individuals can be defined as in [TYK+09] for example for a dance performance performance where choreographer specifies positions of artists. Another examples are individuals placed in some object or on some object [Bez01]. Example here are mass scenes where members of a group are distributed on the terrain or in the hall.

Firstly, when the crowd is going to be simulated it has to be created. This is not a trivial step, because crowd should have heterogeneous individuals. A crowd, that is generated only from the copies of the same individual, is not looking natural especially considering crowd of humans. Usually individuals do not look, move or behave the same way.

It is costly to model and animate each individual separately, therefore some automatic tools need to be developed. One of the possibilities of creating scenes involving thousands of animated individuals is *Crowdbrush* [UHT04]. In this interactive tool authors implemented a brush option. With this brush user can add, remove or modify the individuals and easily create the crowd. The area to be modified can be selected as a point or area. Results from the Crowdbrush are shown in Fig. 4.

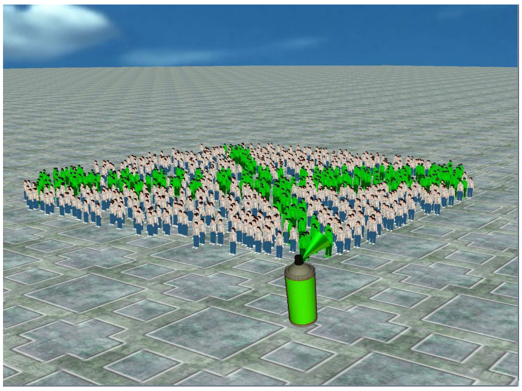

Figure 4: *Color brush with uniform operator. Courtesy of [UHT04].*

An interesting area is crowd of human population. For creating such crowd, the individuals are defined as *somatotypes*. To define such criteria, the anthropometry is used. Three factors are considered endomorphic, mesomorphic and ectomorphic. Endomorphic factor is calculated by measurement of skinfolds, the mesomorphic factor by body dimensions and ectomorphic by weight and height.

The techniques mentioned above are automated, or semi-automated, but in both databases of models need to be used. These models are then automatically processed and final result is given. This automation reduces possibility of change by an artist. When creating a crowd the final output models have to be suitable for animation [TM07].

2.2. Animation

After crowd is generated and heterogeneous individuals are created, the next step is animation of this crowd. This step includes problems with an animation of a crowd in an environment. In this step we consider entities as particle systems only with physical forces, without any social forces involved. Particle systems consider crowds as whole with global view on movement [Ree83]. Therefore path planning and collision avoidance need to be considered. Collision detection should be computed not only with obstacles, but also with another individuals in a crowd. Defining motion manually for each individual is inefficient in the large crowd, therefore also here some automations are investigated.

In [KLLT08] a *motion graph* is created in order to interactively manipulate motion of the crowd. The graph is created in a way, that each vertex represents the location of an individual at a sampled frame, edges represent moving trajectories and neighborhood formations as shown in Fig. 5. Moreover once a graph is created a user can manipulate ver-

© ICGA/TU Wien SS2009.

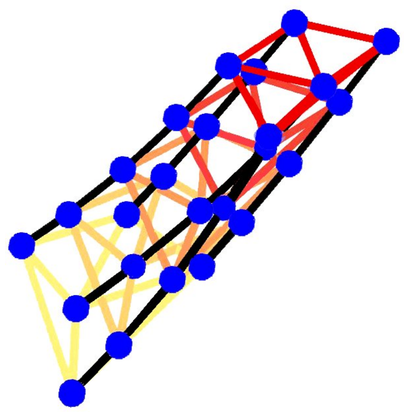

Figure 5: *A group motion graph constructed from the clip. Vertices (blue dots) are connected in two sets of edges, formation edges (colored edges in the figure) and motion edges (black edges). Courtesy of [LCF05].*

tices and by them also the whole group. Therefore graphs can be stitched together and longer motion can be achieved.

Furthermore, using this motion graphs also efficient collision detection can be calculated similarly to [SKG05]. The key positions of individuals and poses are defined and then the optimal path is found. At first, probabilistic roadmaps are created for navigation and path planner. Afterward the randomized search algorithm is created for refinement.

Path planning for groups is the main topic of a recent paper by [TYK+09]. Here the path is calculated for the individuals in a crowd from two given keyframe formations, such as dancers or artists. This work is explained in more detail in the next sections.

Another method is *procedural animation* widely used for the motion of particles. This motion is based on physical or mathematical description of the movement [Ree83].

In many situations simple animation could be insufficient, considering the needs of film industry. A specified position introducing the crowd or specified path of one individual in the crowd involves *constrained animation*. It allows definition of constraints such as exact position of agents in the time, or shapes of group layout [AMC03].

2.3. Behavioral animation

Behavioral animation considers members of the crowd as agents with defined rules and this allows creating of complex behaviors [Rey87]. Depending on which behavioral model is chosen, the path can be planned and the efficiency of algorithms can be achieved. Behavior is a result from social and psychological interaction between entities. These behavioral models depend on the specified task. In many cases the works are multidisciplinary, such as [CKFL05], where social behavioral of animals is studied. For this study the cooperation with ethologists was necessary. In some works the behavior of animal species is studied and the crowd simulation is created afterward. Another example are simulations of evacuation drills, where human behavior in stressful situations has to be considered, as in [Nyg07]. In this approaches cooperation with psychologists is necessary.

The most important work in this field is [Rey87]. Here members of the group are considered as individuals, autonomous agents with specified goal. The boids (bird-like agents) have three local rules of behavior:

- Flocking - boids tend to stay near each other, to stay in the flock.
- Collision avoidance - boids tend to have a distance to other boids to avoid collisions.
- Velocity matching - boids tend to match the velocity with other boids in the flock in order to stay together.

This model can be used not only for birds, but also for other animal species.

Furthermore, derived from this approach is [HB06] where not only this three features are considered, but the *change of leadership* is added. Usually the flock is following the leader, not really considering the other members separately. In this approach boid can separate from the crowd and violate the rule of flocking. Others follow to satisfy the rule of flocking in the smaller group that separates.

[HB94] considers not only the needs of an agent in the group but also problems of individual, such as stability, individual velocity and position in the group. This approach can be used for human crowds as well as for animals in a flock. These systems are called *System with significant dynamics*.

2.4. Integration to environments

Important part of crowd simulation is integration of the crowd in the virtual environment, in order to get more believable results. Environments considered include buildings for emergency procedure, or buildings for special purposes, like museums [TM97]. Another examples are pedestrian simulations where cars usually stays at the road and people on the pavement [TCP06]. This kind of the environment is called *Populated Environments* [TM07]. and should allow cooperation with the crowd to avoid collision with obstacles.

2.5. Rendering

The problem is how to render the animation and simulation of the crowd. In the real-time rendering the main problems are efficient computation time, data storing and fast access to the stored data. Offline rendering usually has to fulfill special needs of director and artists. Therefore, animation should be flexible and easy to manipulate.

© ICGA/TU Wien SS2009.

Figure 6: *Group of lemurs, where they are positioned and rotated in a way, that they look at the king Julian. Courtesy of DreamWorks Animation SKG [Ker05]*

Common in the real-time, the modern approach is to pre-compute animation and save it to database, or even to texture, where coordinates can stand for the key-frames [TM07]. Also level of detail is a challenging problem not only for models, but also for animation. Model level of detail is well studied not only for crowds [OCV+02]. . Animation level of detail can be solved by allowing more random animation for the parts of a crowd that are further away [OCV+02]. Another approach is *motion level-of-detail*, which means modification of model animation in a way that skeleton is simplified. This improvement lowers the costs for computation of skeletal animation.

Non-real-time applications have to show visually attractive scenes usually with a lot more members of the group. These members usually perform specified actions and be in the scene at specified times. The main problems are with the motion itself, because these individuals in a crowd should perform similar action, but not looking as a copies. *High-level behavior* approach was introduced to deal with such animation, where behavior defines motion in more natural way with preserving another part of a group. This behavior is applied on the extra layer of animation on top of existing animated cycles [Ker05]. For example the animation of a crowd of lemurs in a scene where king Julian speaks to them as can be seen in figure 6. The problem in this scene is the rotation of the heads of lemur. When king start to speak, natural is that most of the heads suddenly turn his direction, whereas body stays the way it was. Some artifacts suddenly appeared with the older methods, where only one joint is affected when head moves. With high-level behavior approach the result is more natural.

On the other hand, also human crowds are used in the film industry. The most famous from recent years are crowds in The Lord of the Rings movie. The mass scenes were here created with Massive software. Motion capture was used here for the animation key frames. Then *action tree* was created by the choreographers. To each individual is then assigned set of rules depending on the fight strategy (like offensive or strike).

Although in non-real-time rendering there are no such problems as memory, or computational costs that makes simulation harder for the real-time rendering, the visual part is important. In most of the movies that use visual effects, the virtual characters have to be rendered as well. Therefore, it is not simple to animate them, even when motion capture is used [Rem03].

3. Local crowd behavior

This chapter provides information about local crowd behavior. The section 3.1 provides general information of crowds action in normal and panic situations. The section 3.2 characterizes different crowds and their acting in normal life. The next section 3.3 shows the underlying technique of cellular automata to simulate evacuation processes. And last but not least the section 3.4 shows the sociological aspects of individuals within crowd behavior especially in stressful human conditions.

3.1. Crowd Situations

The simulation of crowds is separated into two main groups according to [CD06].

- Normal situations
- Panic situations

The normal situation could be seen as a model-based flow simulation. People feel a strong apathy moving in the different direction of the main flow in the streets. They also do not want to go a long way round but they probably will not take the shortest path to reach their goal. The fastest and less crowded way is the most popular decision. Many people want to reach a desired walking speed which takes at least energy and is most comfortable. The comfortability is another main part. People do not want to touch each other or getting touched from strangers. Therefore they keep a certain distance to other people and obstacles on their way. This distance could be reduced if people are in hurry or the crowd is getting denser.

People in stressful situations are often referred to be in panic. But panic can not be described as easily because of the irrationality. It is hard to understand in case of non-life-threatening situations like: Entering an underground train or getting good seats at a football match. But mostly life-threatening panic situation start as normal situations and get out of control like seen in figure 7. Major crowd disasters happens mainly with an increasing size of crowds. There is an incomplete list of such fatalities in the work of Helbing et al. [HFMV02]. Panic situations are mostly caused by humans in conjunction with natural disasters like earthquakes or fire disasters. Panic means a collective hysteria which results more often in a great number of victims especially if

© ICGA/TU Wien SS2009.

it happens in a huge crowd of people. Unlike normal situations, panic situations can not be compared to a fluid flow simulation because there is no equal distribution and the following steps can hardly being predicted. If there is no leader within an evacuation process, people tend to show a mixture of herd behavior and individual behavior as seen in figure 8. If there is a leader two possible evacuation methods are possible: "follow-direction method" and the "follow-me method". In the first one, the leader shouts out instructions to the evacuees, in the second the leader tells a few of the nearest evacuees to follow him. In an uncoordinated scenario instead of trying to equally distribute to the two exit points people tend to follow other individuals through a bottleneck which becomes very uncoordinated. But on the other hand some people are trying to escape on the less crowded exit due to individual behavior. People showing herd behavior will start pushing others and interacting physical. This results in people falling or getting injured which causes a much slower escape due to obstacles on their way out. Also the evacuation speed decreases with the density of people are willing to exit the building quickly. These drastic scenarios are well known from concerts, religious gatherings or sporting events.

Figure 7: *Football fans in Sheffield. Clogging makes it difficult passing the open door. Image courtesy of Dirk Helbing et al.*

3.2. Characterization of crowds

The space of individual characters within a crowd in urban environments is limited by obstacles like walls and fences. Crowds can be seen as a set of groups composed of a set of individuals with more than just singular behavior.

By considering high-level behavior the organisation of form of groups could be separated into the following list [CD06]:

- **Flocking**
  The whole group together moves towards the same goal at the same speed
- **Following**
  Individuals within a group follow their group leader

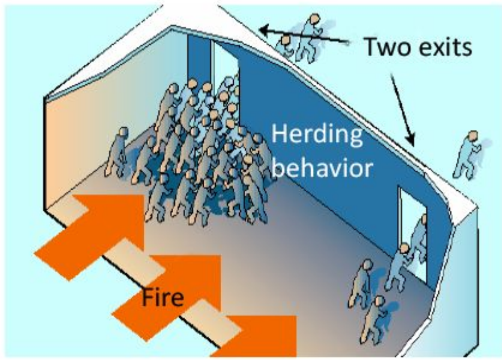

Figure 8: *Helbing Model for Panic Situations. Image courtesy of Dirk Helbing et al.*

- **Goal changing**
  Individuals change initial goal to a convenient goal
- **Attraction in normal situations**
  Individuals are attracted to an attraction point
- **Attraction in panic situations**
  Individuals are attracted to an exit point
- **Static obstacle avoidance**
  Possible obstacle events are determined using mathematical equations. The direction of the agent is changed before reaching the obstacle depending on the distance between an agent and the bounding of the static obstacle.
- **Collision avoidance**
  Most common and most important behavior of animals and human beings. Collisions cannot always be avoided. The possible ways of avoiding collisions are altering direction, speed or self behavior.

Figure 9 shows the architecture of the evacuation simulation which is divided into two main steps:

- Firstly the environment and normal situations of crowds are set up. There is normal behavior and crowd movement until a panic event occurs.
- Within the second step a panic event occurs. The user chooses one of the available strategies and the evacuation time will be calculated.

In normal situations there is general crowd behavior and individual behavior viewable. But in panic situations the concept of groups is getting lost. Within this special case each agent acts different from the rest of the group [CD06].

There must be a psychological aspect taken into account. Humans are not aware of the internal structure of a building and therefore do not know fundamental paths for a quick escape. The rising stress level in case of emergency reduces the full functioning of senses. Due to decreasing awareness it is much harder to orient quickly in rooms or within houses. The next section brings an overview over human behavior in case of fire emergency.

© ICGA/TU Wien SS2009.

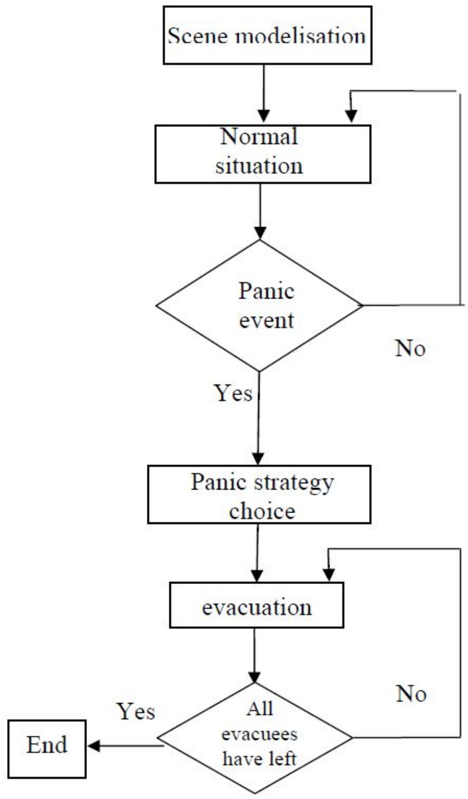

Figure 9: *System Architecture for Evacuation.*

3.3. Cellular Automaton

A cellular automaton models the transition from physical dynamics to discretized environments. A grid of cells represents the environment. Each cell has two states, mostly called "dead" and "alive", and a set of neighbours relative to the specified cell. At the initial state ($t = 0$) a state for each cell is assigned. In advancing state ($t = t + 1$) a new generation is created according to some fixed rules according to both the state of the cells and the states of the adjacent neighborhood.

A simple set of evacuation rules could consist of the following parts:

- Move closer to the exit
- Move farther from the exit
- Move to a cell with the same distance from the exit
- Do not move

This is a simple approach of traditional evacuation simulators. Everyone is treated as an uniform bit with the same behavior. Within the simplest evacuation process people tend to move directly towards the exit.

In pedestrial simulation one pedestrian is mapped to one

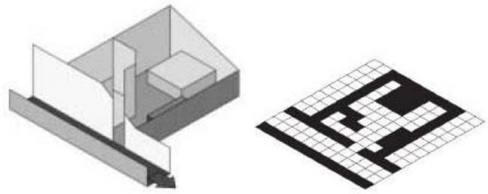

Figure 10: *Space representation in a cellular automata model. Image courtesy of Li et al. [LTS04]*

cell at a time. The local density and velocity are taken into account how people will move between the cells of the underlying grid. Popular examples for cellular automaton are the Nagel-Schreckenberg model [NS92] for traffic simulation or the "Game of Life" [Gar70] from John Conway. The cellular automata not only models the moving parts. Figure 10 shows a representation of an urban environment. Fixed obstacles and walls are shown in black while the white areas can be occupied by pedestrians.

After specifying the fixed parts of a cellular automata the virtual agents in figure 11 are specified for path searching algorithms. Individual or small groups of occupants have individual behavior in normal situations. But in evacuation situations they grow to a very large crowd in centers of evacuation.

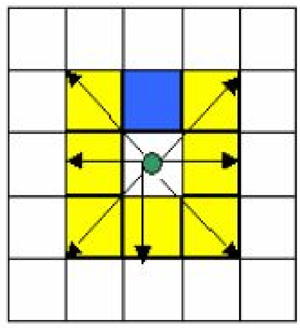

Figure 11: *A* grid with virtual agents position (dot) and its eight neighboring cells (shaded). Image courtesy of Li et al. [LTS04]*

In real-life when a fire siren starts, groups will follow mostly trained evacuation procedures and routes to exit the building very quickly. A computer system acts differently. There are different exit paths for each group of agents possible. During the evacuation process the states of the path planning algorithm is updated due to local constraints, like Li et al. specified in [LTS04]. In this example an embedded A* path planning algorithm is responsible for the path finding process. An A* search algorithm (pronounced "A star")

© ICGA/TU Wien SS2009.

is a best-first, graph-based search algorithm finding the least-cost path from the initial node to one or more possible goal nodes. Figure 11 shows a grid system with an agent and its eight neighbors. The arrows show the search direction. The blue cell shows that a cell was deactivated an though it is invalid. The blue cell is equivalent to a wall or an obstacle.

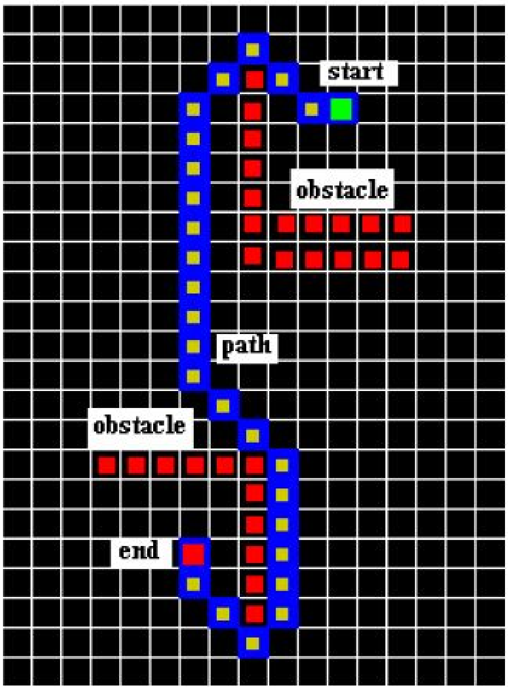

Figure 12: *An A* search example. Image courtesy of Li et al. [LTS04]*

The A* algorithm improves the classic shortest path algorithm by Dijkstra [APR99]. The algorithm of Dijkstra searches all directions from a given starting point to all other points in a grid and determines the least distance from the starting point to all the others sorted in increasing order of their distances. The A* algorithm adds a cost function which is used in a heuristic estimation of the least cost for the whole path movement. Figure 12 shows an example of the A* search algorithm.

The environment is represented by a grid mesh with the same dimension of the building in the environment. There must be a trade-off taken into account between modelling as much as necessary and as much as possible. The red cells in figure 12 show the obstacles like walls. These cells are deactivated from the searching algorithm. Therefore the intelligence of autonomous agents is restricted.

3.4. Sociological Aspects

Social interactions must be taken into account including what they are saying or doing. If there is a leader who is crying about a dangerous situation the evacuee is more likely to change the shortest evacuation route to a much more safe one. Cellular automata models are very helpful in simulating evacuation processes. But knowing the limitations is an essential part of understanding the work of cellular automaton.

Helbing et al. [HFV00] introduced a model for the phenomenon of escape panics. The simulation model is based on a generalized force model. In this model a mixture of socio-psychological and physical forces influences each agent of the crowd:

- N_i: Single agent
- m_i: Mass of a single agent
- v_i: Velocity of a single agent
- τ_i: Certain time of actual velocity v_i
- v_i^0: Certain desired speed
- e_i^0: Certain direction of a single agent

The overall parameters for the model are W for the walls, j for as an index for other agents which is used for the modelling of the interaction forces f_{ij} and f_{iW}. The change of velocity in time t is given by the following acceleration equation:

$$m_i \frac{d\mathbf{v}_i}{dt} = F_i^{(H)} = m_i \frac{v_i^0(t)e_i^0(t) - v_i(t)}{\tau_i} + \sum_{j(\neq i)} f_{ij} + \sum_W f_{iW}$$

Due to the loss of group behavior in stressful situations Braun et al. [BBM05] extended the cellular automaton model in order to deal with different individuals. The population of the agents will be composed of individuals with different attributes. The crowd is described with the following attributes depending on agent i:

- Id_i:
  Identifier of an agent
- $IdFamily_i$:
  Identifier of a family. A family represents a group of agents knowing each other. This is useful for group identification during the simulation.
- M_i:
  Mobility level of the agent represented by a value within the interval $[0,1]$. This value emulates the ability of moving without help.
- A_i:
  Altruism level represented by an interval $[0,1]$. It represents the intention of agents helping each other. For simplicity altruism is only distributed in the same family. Agents with high altruism tend to help dependent agents of the same family.
- e_i:
  Hazardous events represent the accidents causing an evacuation, such as fire, smoke, explosion, etc.

The dynamic equation from Helbing et al. [HFV00] can therefore be rewritten as follows:

© ICGA/TU Wien SS2009.

$$m_i \frac{d\mathbf{v}_i}{dt} = F_i^{(H)} + \sum_{j \neq i} f a_{ij} + \sum_e f_{ie}$$

where $F_i^{(H)}$ is the resulting force of agent i introduced by Helbing et al.,

$$\sum_{j \neq i} f a_{ij}$$

is the resulting force due to the altruism force and

$$\sum_e f_{ie}$$

is the resulting force between agent i and the hazardous effect e.

An evacuation simulation is a kind of social simulation. Any social behavior strongly influences the observations in real-life. In traditional simulations there is no evacuation leader specified. Mostly fire guards or security guards provide the competence to lead people into the right direction because they have the ability to navigate through a building and know parts of it. They can not assume each individual has an overall knowledge of the environment they are in. If the agents are knowing the environment inside out there is a possibility to guide the escapee to alternate routes when known pathes seems to be blocked. As Pelechano and Malkawi [PM07] described, the effectiveness of remembering the whole number of exit paths can be reduced due to stress in the panic situation. The search for an appropriate exit can be much more chaotic and there is a possibility to explore the same exit path more than once as can be seen in figure 13.

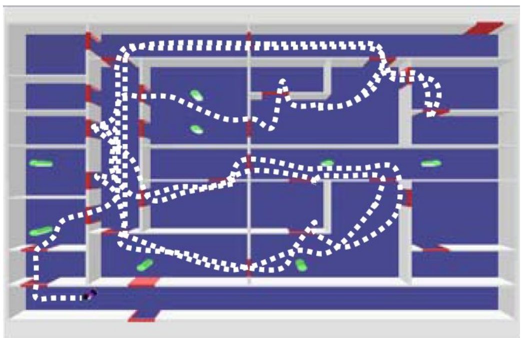

Figure 13: *Chaotic disorientation of agents due to stress. Image courtesy of Pelechano et al. [PM07]*

There has to be a global view for evacuation simulation. [CD07] describes three global elements for evacuation behavior:

- Less encumbered exit
- The nearest exit
- Guided exit

The evacuees will notice the available exits and will choose the exit where is no clogging at the moment and where least people waiting for leaving. This exit will be the goad no matter if it is the nearest or the farthest exit available. But he does not know the status of the exit. Maybe the exit is blocked or has a dead end. In iterating over other possible exits the evacuee may waste time but he can inform other people about wrong exits and they can change their evacuation plannings.

The nearest exit uses quite the same strategy as the less encumbered exit does. But in this approach only the nearest exit will be chosen.

By using a guided evacuation model the leader chooses the exit. It can be an autonomous agent with a global view of the environment who can lead all the groups directly to the designated exits. Furthermore they can combine the two strategies discussed before.

The changes of human behavior in stressful situations is hard to transfer into a formal model because of the quite unknown change in human mind when stressful situations occur. Especially psychologists and sociologists have to study the behavior of individuals in panic situations. The simulation of evacuation scenarios is quite impossible so real cases of emergency have to be analyzed.

The next section 4 deals with a global view of crowd simulation. Information about path planning and the interpolation of group movement and formation creation are given there.

4. Path Planning

Path planning is a topic that includes generation of layout for further group movement, interpolation between keyframes and finding the path avoiding the obstacles. This topic still can be better investigated. We refer in this paper to the formation as some shape created with the positioning of individuals in the crowd in some keyframes. Formations can be defined as a shape the particles create or by the rules they follow.

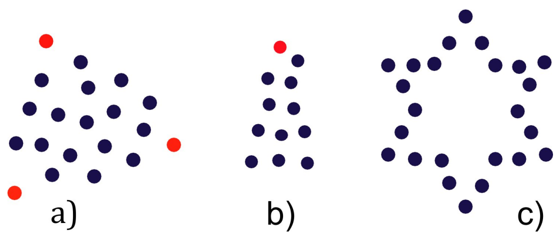

Figure 14: *Differences between the motion based on shepherding (a), following the leader (b) and creation of formation (c). (a) Group of shepherds is in red, group of other animals is in blue. (b) Leader is in red and followers are in blue. (c) Each participant is equal to other and together they create visually attractive formation.*

We refer to a formation as a meaningful shape such as artistic layouts in mass performances. The need for creating formations is based on observation of real situations. Computer graphics often wants to provide simulation or approximation of real processes.

These processes can be divided into:

- **Shepherding**
  Motion of one group here is controlled by another group [LRMA05]. Shepherds are agents that influence the movement of a flock, where flock is a group of agents that move in coordinated manner and respond to the external factors as is shown in Fig. 14a. Goal of the shepherds is to steer the flock toward specified position. Formations and path planning here is affected by collision in the flock and heading toward the position.
- **Following the leader**
  Motion of the whole group depends on a few individuals that are leaders as is shown in Fig. 14b. Leaders are informed agents and affect whole group [CKFL05]. Animation with the leadership is based on the observation of the groups of animals. Some species were chosen and similarities were found. From the movement point of view, similarly to the previous approach, group moves together to gather the defined goal. No special visual formations are created.
- **Creating a formation**
  Previous approaches provides solutions for group motion based on agents, but also can be emulated with particle systems, where the description of forces between individuals allow control of a crowd movement. Each individual is considered as agent with defined behavior. Either is it a leader, a shepherd or a follower. Another method is considering a crowd as whole. Particle systems is a method that allows such a definition, where individuals are considered as particles. Takahashi et al. [TYK+09] bring control of movement over whole group and the particles can create visually attractive formations as is shown in Fig. 14c. When considering the group as a whole not each separate individual.

The difference between agent-based systems and particle systems is manipulation with a crowd. Where in agent-based approach individual simulation is considered, in particle system approach the whole group is considered. Individual approach allows defining more specific behavior of individual, in particle systems the simulation is usually described with physical forces. In [Rey99] the *seek* and *pursuit* behaviors are described. Although these are behavioral methods the goal is same as in [TYK+09]. The individual tries to reach defined static position (seek) or moving target (pursuit). Distinct from these behaviors, the *path following* sweeps object along path, usually defined by spline curve. This method use less agent based approach and can be used with particles.

4.1. Constrained animation

Constrained animation defines restrictions on the animation of a group. This can be either a specific layout with obstacles, specific position of the group or movement of individual. Obstacles are a problem in almost any realistic group animation or simulation because they are common in real world. On the other hand, positioning members of the group or special definition for individual movement are more common for artistic reasons in the film industry.

The simplest way of creating constrained animation is to manually modify the path if it is possible. The easiest way of creating such simulation is to manually define key-frames and positions for the parts of animation [TYK+09]. This is not a trivial problem and in methods that do not deal with collisions directly it is left for the future work or is adjusted manually.

Another methods such as [AMC03] deal with constrained animation directly. Key-frame based methods allow full control over the animation, that artists like to have, but it is time-consuming and the defined paths for some individuals do not allow them to cooperate with the rest of crowd. Cooperation or influence is a natural ability of humans and animals, therefore it is often required and some randomness in the movement looks less mechanical [AMC03].

Splines in crowd simulation are used to calculate the movement of participants. To calculate splines, the key-points are created. These keypoints are abstracted from the keyframes. Then smooth interpolation between these keyframes is obtained with splines. Splines can be also successfully used also with the constraints [RBB97]. Avoiding obstacles or collision detection is not fully automatic, but with the right settings of *hard constraints* and *soft constraints* obstacles can be avoided. Hard constraints in this context have to be fully satisfied and soft ones have to be satisfied as much as possible, but not fully.

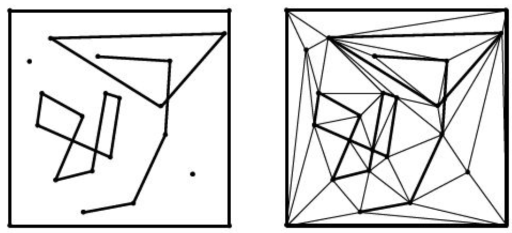

Figure 15: *Constraints in layout (left) Delaunay triangulation with preserving the constraints (right) [KBT03]*

The problem of dynamic constrained path planning is discussed in [KBT03]. Here the problem is solved with creating a constrained delaunay triangulation (CDT), which is similar to delaunay triangulation, but respects constraints as is shown in Fig. 15. Once the CDT graph is created, the

© ICGA/TU Wien SS2009.

two points are defined (p_1, p_2) and the shortest path between them is going to be found. These points are located in triangles of the triangulation. Then over the adjacency graph the path of triangles is created. Let P be a polygon defined with these path. Finally the shortest path from p_1 to p_2 is inside the polygon P. The shortest path is found that respects constraints. Because the CDT gets dynamically changed, this approach can be used also for moving constraints.

4.2. Particle systems

Fluids, water, smoke and other physically well described objects in computer graphics are hard to model with polygonal surfaces or curved surfaces. Particle systems firstly introduced in [Ree83] are easy way of describing such phenomena. Particle systems describe an object as a cloud of primitive particles that defines volume instead of surface. Particles are very simple, primitive objects, that can interact with environment and among each other.

Particle system has advantage in the computational time, because number of particles can vary, animation is physical based. Level of detail can be easily computed with lowering the number of particles when we do not consider avoiding with other participants in a crowd. The movement and description of particles is usually mathematical, therefore it was widely used for physical processes, as motion of fluids [Ree83] or airbag [BCN97]. Also group motion of pedestrians can be described with particle system basis [BCN97]. Furthermore particles can be associated with more complex geometry and description of behavior and create more complex simulation.

In the elementary idea of particle systems for fluid dynamics, the particles have a property of lifetime. For each frame following is calculated [Ree83]:

- new particles are created and assigned individual features and initial settings
- any particles that have existed past their prescribed lifetime are extinguished
- other particles are moved and transformed according to their attributes
- image is rendered

Difference between particle systems use in crowd simulation and fuzzy objects is in creating and destroying the particles. Where in traditional approach particles are created and destroyed, there is a change in the number [Ree83]. In the approach used for crowd simulation the number is not changing [TYK+09]. Other features of particle systems can be used also when considering crowd simulation. In fluid dynamics particles are usually partially transparent and can blend together to create more realistic motion, but in crowd simulation particles are usually solid and moreover have to calculate collision instead of blending. Similar conditions are applied in the furry objects, where are particles also widely used [LPFH01]. There are some similarities in the distribution of particles in the generation step between crowd simulation and furry objects as can be seen in Fig. 16.

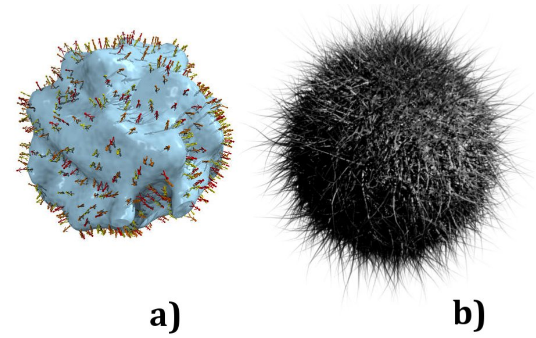

Figure 16: *Generation of the layout and distribution of particles for the crowd simulation (a) and for the furry object (b). In both cases distribution of particles is approximately uniform*
(a) Courtesy of Dusan Bezak (b) Courtesy of Blender Association

Using particle hierarchy as introduced in [Ree83] the particles can be themselves particle systems. Transformations applied on the parent particle then are applied on the whole system in this particle. This way the movement of the whole group can be either divided into independent subgroups or the whole group can be easily transformed.

Before 2006 almost all simulations of crowds were agent-based. It has the advantage that also in real world each individual is independent. Therefore, these agents can respond to situations based on the rules. Disadvantage is that it is hard to develop realistic motion and manipulate the crowd as whole. Approach discussed in [TCP06] defines motion synthesis model for large crowds without agent-based dynamics. Here the global planning provides also avoiding obstacles and other people by minimizing the energy. Particle systems are base of this approach.

Provides mathematical analysis for the group motion. Four hypotheses are provided that describe group motion. We refer to individuals in a group as people.

- **Goal**
  Each person has a goal. Either it is a geographic location or dynamic goal as following some moving object.
- **Speed**
  Each person moves the available speed.
- **Preferences**
  Each person has path preferences for a certain path, such as pedestrians usually walk on the sidewalks.
- **Path**
  When first three conditions are successfully fullfiled, per-

son chooses the path that provides the optimal combination of distance, time and discomfort.

In [BCN97] is shown that particle systems are powerful and can be used in various situations. Particles can interact among themselves, with other systems and also with obstacles. Firstly, these systems were designed for animation of fluid flow, but with some enhancement they can be used also for crowd simulation. A human flow simulation is a complex problem, because human behavior needs to be provided for better realism. Therefore some sort of decision making and reaction to the environment need to be possible.

Therefore these behaviors have to have mathematical description that can be implemented to the particle system. In [BCN97] such description is provided for reflex reactions and decisions. Reflex reactions concern immediate change in movement considering avoiding obstacles and other particles. Decisions are more complex problem, but a person with decision charge can be influenced in the same way as a particle with an electric charge. And that is a physical process that has mathematical description.

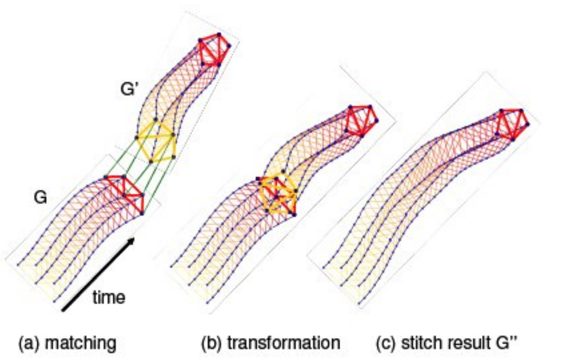

Figure 17: *Group motion stitching steps.*
(a) finding corresponding points (b) aligning of the graphs (c) smooth blending between the graphs
Courtesy of [KLLT08]

4.3. Motion Graphs

As was said in the introduction and related work section for the group animation motion graph can be used. In [KLLT08] the motion graph is enhanced to provide multiple motion clips in the same timeline. Either larger groups can be divided into smaller or group can be combined together and form larger formation.

In graph the edges are divided into formation and motion edges. Formation edge represents neighborhood relationship between vertices and can be useful to preserve adjacent relationship. However, these relations are usually not well defined. Delaunay triangulation can produce reasonable connectivity between individuals in keyframes. For [KLLT08] this approach is sufficient with the optional user adjustment. Also with this triangulation still local enhancement is automatically calculated to preserve the distortion of local features and shape. With this motion graph and formation edges the advanced formation planning can be provided.

Formation edges define movement in the space and motion edges define movement in the time. For spatial features the whole group needs to be considered to compute movement, but for temporal features only moving paths of individuals are considered. Spatial movement is calculated for the keyframe. Both these edges together define spatiotemporal movement of a person. The main problem is to stitch motion groups together with preobserving spatiotemporal features of all groups. To do this, following conditions need to be satisfied [KLLT08]:

1. establishing one-to-one mapping between first group and second group Fig. 17a
2. alignment of the clips Fig. 17b
3. smooth morphing from one group to another Fig. 17c

First point needs to be always satisfied when considering a simulation of a group that creates formations as keyframes. Such a group can be for example dancers. These dancers need to have defined spatial positions respect to a keyframe in a time. The movement from one position in a first keyframe to new position in a next frame can be automatically calculated.

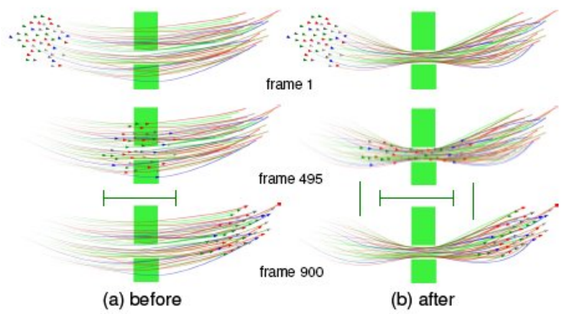

Figure 18: *Avoiding obstacles.*
Forcing a circular group (a) through a narrow opening such that the group must elongate to pass. In this specific example, penetrating points inside obstacles are pulled toward the opening. Resolving all collisions required 34733 iterations, which took about 163 seconds.
Courtesy of [KLLT08]

Collision detection is solved in post processing because only when linear trajectory of a point is inside the obstacle, this trajectory is adjusted by moving the deepest point from collision to the boundary. This process is iterative and continues until trajectory does not penetrate obstacle. Results shown in Fig. 18.

© ICGA/TU Wien SS2009.

There appear artifacts such as velocity change because of artifacts in motion edit and enhancements in animation. This is partially solved in [KLLT08], where the goal of warping step is to move closely to the original speed by defining time-warping problem as a least-squares optimization.

4.4. Spectral Based Group Formation Control

The article by Takahashi et al. [TYK+09] is the most recent one in the manual path planning field. Topics related to the article, main contributions and possible enhancements for future work are discussed in the next sections.

Group formations in the real world situations can be found in the mass performances or tactical sports such as soccer. This situations have inspired authors of the paper to provide a mathematical description of the motion. This model allows to compute the smooth and realistic movement of a group while respecting adjacency relationships. In mass performances relative positions between neighbors are usually kept to achieve visually pleasant movements.

Authors of the article divide methods of crows behavioral control in *microscopic* and *macroscopic*. In the microscopic approach, simulation of the group behavior is considered as local motion of individuals. Usually these individuals are agents with defined rules, as it is in agent-based approaches as in [Rey87]. These individuals suffer with the lack of properties defined by the whole group. On the other hand we would like to closely look at the more global techniques that are defined by the macroscopic approaches. We would like to consider a group of individuals as a particle system [Ree83] and with that obtain a definition of a group as a whole. With particles systems it is hard to define complicated behavior of the individuals, but the main goal of our paper is finding the model of a motion not behavioral model for the complex situations such as evacuation drill.

Firstly, to create smooth transitions between formations, these shapes need to be defined. There are many ways how to define formations, one of which is manual association of individuals with their spatiotemporal positions. Other way is to extract positions in the formation from the captured video as in [LCHL07] where they captured a performance from a plane. Usually a scenario and spatiotemporal correspondence is provided beforehand by an artist or a choreographer by defining the shapes or even position of individuals. Afterward automatic calculation of simulation is possible, but artists would like to have a possibility of further control. More automatic approaches are discussed in the section about possible future plans.

After the formations have been created, second step is to calculate a smooth motion. Pre-processing step is needed where the formation is analyzed and calculations are prepared for the next steps. Adjacency relationships are extracted from the formation with *Delaunay triangulation* and this graph is then used as a graph for adjacency relationship. In the final graph vertices represent individuals and edges are defined by the triangulation. Edges have positive weights, that are defined as a inverse of the distance between vertices. In order to respect the spectral-based structure of a group formation the *Laplacian matrix* is analyzed. This matrix is symmetric, positive and definite. Therefore the eigenvalues are nonnegative and eigenvectors define *orthonormal basis* called *eigenbasis*. These structures allow spectral decomposition of the graph and with these decomposition calculation of an interpolation respecting adjacency relationships is possible.

Afterward processing is divided into two steps. Firstly, interpolation is calculated from the first formation shape to a second with preserving adjacent relationships. Secondly, the calculation of geometric interpolation is provided. From the pre-processing step we have Laplacian matrices defined for the formations and interpolation is then calculated as a transformation of the eigenvectors from the source Laplacian matrix to the target Laplacian matrix. To keep spatial adjacency relationships, also keeping the orthogonality of the vectors is needed. Therefore, the most suitable transformation is rotation, because it satisfies orthogonality, while being simple. Source matrix is defined as:

$$S = (e_1^S, \ldots, e_n^S) \tag{1}$$

target matrix is defined as:

$$T = (e_1^T, \ldots, e_n^T) \tag{2}$$

where $e_k^{S,T} k = 1, \ldots, n$ are eigenvectors source and target respectively. Rotation matrix is derived from the source and target matrices as follows:

$$R = TS^{-1} \tag{3}$$

For the interpolation the matrix of eigenvectors at the time t is defined as $M(t) = (e_1(t), \ldots, e_n(t))$ and derived from rotation matrix as follows:

$$M(t) = R^t S \tag{4}$$

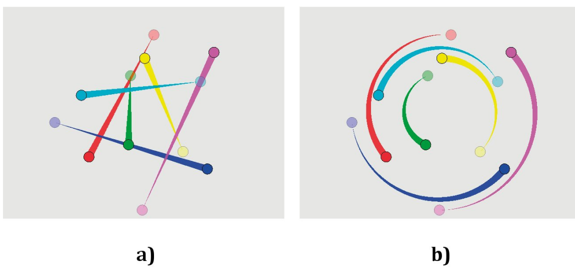

Figure 19: *Difference between linear interpolation (a) and polar interpolation (b). Courtesy of [TYK+09]*

The geometric interpolation is performed to fix the coefficient vectors at each time step. Since we have obtained eigenvectors at each time step from the previous equation, only coefficients are needed for the interpolation and final motion of the the group. These coefficient are obtained based on *polar interpolation*. The difference between linear interpolation (a) and polar interpolation (b) is shown in Fig. 19. If the two given formations are moderately different, firstly is necessary to find the principal axes using principal component analysis (PCA) and then rotate the target formation and principal axes to align with the source formation.

In the previous sections smooth motion between two formations is discussed, but the mass performances of group simulation usually consists of more than two formations. Therefore, also a solution for multiple formations interpolation is provided. In the [TYK+09] extended Catmull-Rom splines are used for the smooth transformation.

Unfortunately, avoiding obstacles and collision between members of the group is not a simple problem, when considering a group from a global point of view. Therefore also paper [TYK+09] considers environment better unconstrained and in the common mass performances or stage performances there are usually not many obstacles, even better the movement space can be considered as 2D. However, potential of this approach leaves enhancements to the future work such as collision detection and is discussed in the next section.

4.5. Possible enhancements

From our point of view, possible improvements for the approach described in [TYK+09] includes:

- better collision avoidance
- local adjustments
- more automated pre-processing step
- extension to the 3D space

Better collision avoidance
Avoiding obstacles is currently handled by social forces as is described in [HM95]. Another obstacle avoiding approach is described in [KLLT08]. It is also post processing, that means that firstly the path is obtained then collisions are resolved. If a part of the path collides with an obstacle the deepest penetrating point is move to the boundary. Moreover, for this solution the clip motion graphs need to be defined for formations. With the constraints deals also [KBT03] where the constraints are included in the creation of the Delaunay graph. If we have Delaunay graph created in such a way, that constraints are included, better avoidance can be provided. For the collision avoidance between the group members, weights in a graph represents the stability of the adjacency relationship and therefore help not only for the motion but also for the collision avoidance.

Local adjustments
Constrained animation requires definition of feature for some individuals. In this approach it means defining of the specific motion for the part of crowd. As is shown in the paper [TYK+09], hierarchical structure is provided that allows operating with the subgroups separately. Therefore when the only one individual is in this group artist can define position for this individual.

Also local adjustments can be viewed from the point of path movement. Now the path is calculated from keyframe to keyframe, but in the future also better mathematical manipulation of trajectories could be provided. Considering not only straight trajectories, but also waved.

More automated pre-processing step
Defining each formation manually is a very time consuming step, also extracting formations from a video and finding associated individuals from one formation in the next. Manual definition is necessary when artists or choreographers want to define positions of the individuals in each formations, or when the simulation has to be based on real performance. On the other hand when providing some visual stimuli by the shapes with group is sufficient, some automation in this step should be provided. These shapes can be also defined with some geometrical rules. If some popular shapes are stored in the database, they can be randomly chosen for the keyframes.

For some performances only shape of the formation is important, not the position of specific individuals. Even the positions that created the shape are not important. A way of obtaining these positions in the shape to create a formation from a crowd is also interesting. Artist defines only the shape with a scratch, or model, or even picture and layout distribution can be calculated. With this approach some pre-defined shapes can be stored and the final position of the individuals is calculated depend on the number of particles. This method is more independent in number of particles.

The position in the next formation can be obtained with calculating the smallest sum of the distances for all particles. Smallest distance can be obtained for each individual from the source formation to the position in target. The problem is, when this mapping is not bijective. Either for one source individual two position in target formation are equal, or the position in target formation best fits for more than one individual from source formation. This is left for further discussion. Also some semi-automated approaches can be used. Semi-automated are those when some parts of a keyframe are constrained by an artist and some can be automatically calculated.

Extension to the 3D space
In the paper some extensions to the 3D space are provided, but there are many different methods how this approach can be extended to 3D space:

- Position in formation lay in the same plane as in Fig. 20. This plane can be rotated, so the final formation is in a 3D space, but the interpolation is the same as proposed

© ICGA/TU Wien SS2009.

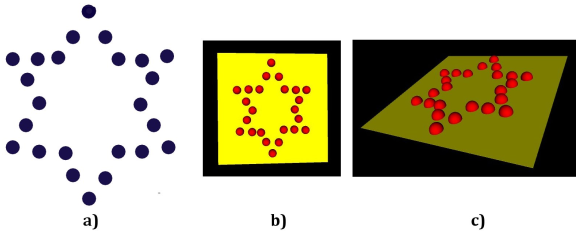

Figure 20: *Difference between formation in 2D and planar formation in 3D. (a) formation in 2D (b) formation in 3D, TopView (c) formation in 3D, rotated view*

Takahashi et al. [TYK+09]. The only difference is that three coordinates need to be considered. Example is free-fall parachutist. They usually create formations that lay in the plane while flying in the 3D space.

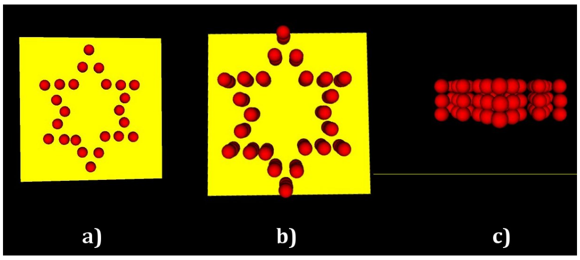

Figure 21: *Difference between planar formation in 3D and layered formation in 3D. (a) planar formation (b) layered formation, TopView (c) layered formation FrontView*

- Another type is a formation that can be divided in the subgroups, where all members of the subgroup lay in the plane as in Fig. 21. If the all formations can be divided in the similar layers, the interpolation can be obtained in the layers separately.

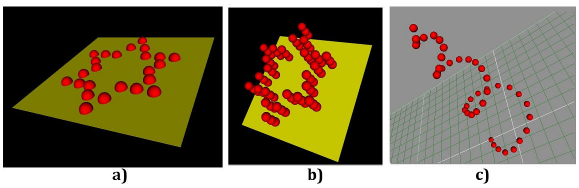

Figure 22: *Difference between planar formation in 3D (a) layered formation (b) and full 3D formation (c)*

- The last option is a full 3D formation where individuals are distributed in all three directions 22. For this option also movement needs to be considered in all three directions. Examples are flying birds, or exhibition of airplanes.

As is explained in the paper for the movement in the 3D space,s spherical interpolation instead of polar can be used. Another problem is pre-processing step where spatial adjacency is described as graph and this graph is obtained with the Delaunay triangulation. Also in 3D space Delaunay triangulation is possible and discussed in [CMS98].

References

[AMC03] Matt Anderson, Eric Mcdaniel, and Stephen Chenney. Constrained animation of flocks. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 286–297. Eurographics Association, 2003. 5, 11

[APR99] J. Aslam, K. Pelekhov, and D. Rus. A practical clustering algorithm for static and dynamic information organization. In *In Proceedings of the 1999 symposium on discrete algorithms*, Baltimore, MD, 1999. 9

[BBM05] Adriana Braun, Bardo E.J. Bodmann, and Soraia R. Musse. Simulating virtual crowds in emergency situations. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 244–252. ACM, 2005. 3, 9

[BCN97] Eric Bouvier, Eyal Cohen, and Laurent Najman. From crowd simulation to airbag deplyoment: particle systems, a new paradigm of simulation. *Journal of Electronic Imaging*, 6(1):94–107, January 1997. 12, 13

[Bez01] Dusan Bezak. Mass scenes rendering framework. Master's thesis, Commenius University, Bratislava, Slovakia, 2001. 4

[CD06] Foudil Cherif and Noureddine Djedi. A computer simulation model of emergency egress for space planners. *Georgian Electronic Scientific Journal*, 8(1):17–27, 2006. 6, 7

[CD07] Foudil Cherif and Noureddine Djedi. An autonomous and guided crowd in panic situations. *Journal of Computer Science and Technology (JCS&T)*, 7(2):134–140, 2007. 10

[CKFL05] Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, February 2005. 5, 11

[CM05] Nicolas Courty and Soraia R. Musse. Simulation of large crowds in emergency situations including gaseous phenomena. In *Computer Graphics International 2005*, pages 206–221, 2005. 2

[CMS98] P. Cignoni, C. Montani, and R. Scopigno. Dewall: a fast divide and conquer delaunay triangulation algorithm in ed. *Computer-Aided Design*, 30(5):333– 341, April 1998. 16

[Gar70] M. Gardner. The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, October 1970. 8

© ICGA/TU Wien SS2009.

[HB94] Jessica K. Hodginsm and David C. Brogan. Robot herds: Group behaviors for systems with significant dynamics. In *Proc. Artificial Life IV.*, pages 319–324. IEEE Computer Society, 1994. 5

[HB06] Christopher Hartman and Bedrich Benes. Autonomous boids. *Computer Animation and Virtual Worlds*, 17(3-4):199–206, July 2006. 2, 5

[HFMV02] Dirk Helbing, Illes J. Farkas, Peter Molnar, and Tamas Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. In *Pedestrian and Evacuation Dynamics*, pages 21–58. Springer, 2002. 6

[HFV00] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, September 2000. 9

[HM95] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995. 15

[KBT03] Marcelo Kallmann, Hanspeter Bieri, and Daniel Thalmann. Fully dynamic constrained delaunay triangulations. In *Geometric Modelling for Scientific Visualization*, pages 241–257. Springer-Verlag, Heidelberg, Germany, first edition, 2003. ISBN 3-540-40116-4. 11, 15

[Ker05] Laurent Kermel. Crowds in madagascar. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*. ACM, 2005. 3, 6

[KLLT08] Taesoo Kwon, Kang H. Lee, Jehee Lee, and Shigeo Takahashi. Group motion editing. In *SIGGRAPH '08 Proceedings*, pages 1–8. ACM, 2008. 4, 13, 14, 15

[LCF05] Yu-Chi Lai, Stephen Chenney, and Shaohua Fan. Group motion graphs. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 281–290. ACM Press, 2005. 5

[LCHL07] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: a data-driven approach to crowd simulation. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2007. 14

[LPFH01] Jerome Lengyel, Emil Praun, Adam Finkelstein, and Hugues Hoppe. Real-time fur over arbitrary surfaces. In *In Symposium on Interactive 3D Graphics*, pages 227–232. ACM Press, 2001. 12

[LRMA05] Jyh-Ming Lien, Samuel Rodriguez, Jean-Phillipe Malric, and Nancy M. Amato. Shepherding behaviors with multiple shepherds. In *Proc. of the 2005 IEEE International Conference on Robotics and Automation*, pages 3402–3407. IEEE, 2005. 11

[LTS04] Huibo Li, Wen Tang, and D. Simpson. Behaviour based motion simulation for fire evacuation procedures. In *Theory and Practice of Computer Graphics, 2004. Proceedings*, pages 112–118. IEEE, 2004. 8, 9

[McD06] Rachel McDonnell. *Realistic Crowd Animation: A Perceptual Approach*. PhD in crowdanimation, University of Dublin, Trinity College, 2006. 3

[MMKI02] Yohei Murakami, Kazuhisa Minami, Tomoyuki Kawasoe, and Toru Ishida. Multi-agent simulation for crisis management. *IEEE Workshop on Knowledge Media Networking (KMN'02*, 0:135, 2002. 2

[NS92] Kai Nagel and Michael Schreckenberg. A cellular automaton model for freeway traffic. In *Journal of Physics I France 2*, number 12 in 2, pages 2021–2229. JOP, 1992. 8

[Nyg07] Martin Nygren. Simulation of human behavior in stressful crowd situations. Master's thesis, Royal Institute of Technology, Stockholm, Sweden, 2007. 5

[OCD+02] Carol O'Sullivan, Justine Cassell, Simon Dobbyn, Christopher Peters, William Leeson, Thanh Giang, and John Dingliana. Crowd and group simulation with levels of detail for geometry, motion and behaviour. In *Third Irish Workshop on Computer Graphics*, 2002. 1, 4

[OCV+02] Carol O'Sullivan, Justine Cassell, Hannes Vilhjalmsson, John Dingaliana, Simon Dobbyn, Brian Mcnamee, Christopher Peters, and Thang Giang. Levels of detail for crowds and groups. *Computer Graphics Forum*, 21(4):733–741, 2002. 6

[PM07] Nuria Pelechano and Ali Malkawi. Comparison of crowd simulation for building evacuation and an alternative approach. *Building Simulation*, pages 1514–1521, 2007. 10

[RBB97] Ravi Ramamoorthi, Cindy Ball, and Alan H. Barr. Dynamic splines with constraints for animation. Technical report, California Institute of Technology, Pasadena, CA, USA, 1997. 11

[Ree83] William T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2:359–376, 1983. 4, 5, 12, 14

[Rem03] Scott Remington. Sparking life: notes on the performance capture sessions for the lord of the rings: the two towers. *SIGGRAPH Comput. Graph.*, 37(4):17–21, 2003. 2, 3, 6

[Rey87] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, July 1987. 5, 14

[Rey99] Craig W. Reynolds. Steering behaviors for autonomous characters, 1999. 11

[SKG05] Mankyu Sung, Lucas Kovar, and Michael Gleicher. Fast and accurate goal-directed motion synthesis for crowds. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 291–300, New York, NY, USA, 2005. ACM. 5

[Sof09] Bethesda Software. Fallout3 - official webpage. http://fallout.bethsoft.com/index.html, 2009 (accessed May 27th, 2009). 3

[Spo09] EA Sports. Fifa09 - official webpage. http://www.fifa09.ea.com, 2009 (accessed June 24th, 2009). 3

[SW08] Daniel Scherzer and Michael Wimmer. Frame sequential interpolation for discrete level-of-detail rendering. *Computer Graphics Forum (Proceedings EGSR 2008)*, 27(4):1175–1181, June 2008. 4

[TCP06] Adrien Treuille, Seth Cooper, and Zoran Popovi. Continuum crowds. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1160–1168. ACM Press, 2006. 5, 12

[TM97] Daniel Thalmann and Soraia R. Musse. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Workshop Computer Animation and Simulation of Eurographics*, pages 39–52, 1997. 5

[TM07] Daniel Thalmann and Soraia R. Musse. *Crowd Simulation*. Springer, 1 edition, October 2007. 4, 5, 6

[TYK+09] Shigeo Takahashi, Kenichi Yoshida, Taesoo Kwon, Kang Hoon H. Lee, Jehee Lee, and Sung Yong Y. Shin. Spectral-based group formation control. *Computer Graphics Forum*, 28:639–648, 2009. 4, 5, 11, 12, 14, 15, 16

[UHT04] Branislav Ulicny, Pablo Heras, and Daniel Thalmann. Crowdbrush: interactive authoring of real-time crowd scenes. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 243–252. ACM Press, 2004. 4

© ICGA/TU Wien SS2009.