

Finding Paths for Coherent Groups using Clearance

Arno Kamphuis

Mark H. Overmars

institute of information and computing sciences, utrecht university

technical report UU-CS-2004-062

www.cs.uu.nl

Finding Paths for Coherent Groups using Clearance

Arno Kamphuis Mark H. Overmars

December 2, 2004

Abstract

Virtual environment are often populated with moving units and the paths for these units should be planned. When multiple units need to exhibit coherent behavior in a cluttered environment, current techniques often fail, i.e. the resulting paths for the units in the group lack the coherence required. In this paper, we propose a novel approach to motion planning for coherent groups of units.

The method presented uses a path for a single unit, called the backbone path, which can be generated by any motion planner. This backbone path is extended to a corridor using the clearance along the path. The units can move freely inside this corridor. By limiting the width of this corridor, and the extent along the corridor where the units can move to, the approach guarantees coherence of the group.

Experiments show that the generated paths exhibit group coherence as required, like passing on the same side of obstacles and waiting for fellow group mates to catch up. Performance measurements show that the approach is capable of generating the paths in real-time. In our implementation, the method requires just a few percent of the processor time for groups consisting of up to 100 units.

1 Introduction

Virtual environments are often populated with moving units. Games in particular, but also other virtual environment applications, contain (very) large numbers of units moving around. The units should often behave as a coherent group rather than as individuals. For example, in safety training applications one needs to simulate the behavior of a crowd and in games one often needs to simulate the behavior of whole armies.

Current applications solve the problem of path finding on the unit level, i.e. they plan the motion of individual units, using techniques like flocking to keep the units together. However, in cluttered environments this often leads to non-coherent groups. There is no guarantee that the units will stay together, albeit that 'staying together' is not well defined. Even though the units all have a similar goal, they try to reach this goal without real coherence. This results in groups splitting up and taking different paths to the goal, for example as in Fig. 1. Here, one individual decides to take a different route. This is highly undesirable behavior and in this particular example, it leads to immediate death.



(a) A group of five characters should attack the site pointed to by the arrow.



(b) The group inappropriately splits up, losing some of its troops.

Figure 1: One of the problems with the current techniques for motion planning for multiple units is that the group splits up to reach the goal. This scene was taken from Command and Conquer: Generals from EA Games.

1.1 Previous work

Motion planning and path finding have been studied extensively, both in the virtual environment and game community, and in the robotics community. In this paper, we restrict ourselves to motion planning approaches to compute the simultaneous motion of multiple units. This spans the work done in robotics, flocking, and crowd and pedestrian/traffic flow simulations.

In the virtual environments community the most common approach to simulating group movement is to use flocking. The concept of flocking was introduced by Reynolds [15]. His *boids*-model described the behavior of the units in a group using only local rules for the individual units. Later, Reynolds extended the technique to include autonomous reactive behavior [16]. The idea is that units steer themselves in such a way that they avoid collisions with other units and the environment, while at the same moment they try to align themselves with other units and try to stay close to the other units. In open areas this leads to rather natural group behavior as can be observed in flocks of birds or schools of fish. When we also give the units a goal they will move toward the goal together. The big drawback of this approach is that the units act based on local information which easily gets them stuck in cluttered environments. Also, the combined steering behavior can easily lead to the group breaking up, as in Fig. 1.

Another widely used technique is grid searching in which the environment is divided into a grid that can be searched for a free path using A* like approaches [17]. Different units try to find a path through the grid while avoiding collisions with each other. This easily leads to units getting stuck in ways that can only be resolved by rather unnatural motions (or cheating like penetrating the walls). This problem gets even harder when non-holonomic constraints (like the fact that cars have a bounded turning radius) must be taken into account.

The social potential field technique [14] defines potential force fields between units of the group.

Desired behavior is then created by defining the correct force fields. However, the same problem as in flocking arises because only local information is taken into account.

Kamphuis and Overmars [6] developed a method for planning the motion of a coherent group of units using a multiphase algorithm. First, a path is planned for a deformable rectangle, representing the group shape. Second, the internal motion of the units inside this deformable rectangle is calculated using social potential fields. Third, the global and local paths are combined to give the total motion of the units. Although the technique guarantees coherence, it lacks completeness. The approach also generates unnatural behavior when a group enters or leaves a narrow passage.

One of the dominating techniques in robotics is the probabilistic roadmap approach (PRM) [7,8]. Efficient probabilistic (centralized) techniques for multiple units have been developed [18,20]. They treat the different units together as one large robotic system. Unfortunately, each unit has two degrees of freedom (assuming it is defined by its position on a floor surface) so the total robotic system has $2n$ degrees of freedom when there are n units. When n gets larger the running time becomes too large. Also, these approaches require that the number and type of units are known beforehand which is not a realistic assumption for the applications we have in mind. To overcome this problem, decentralized techniques, like path coordination, have been developed, enabling the planning of motion for a larger number of units [9]. Still though, these methods fail when the number of units grows and the resulting motion is not coherent.

Bayazit, Lien and Amato [2] have combined the PRM approach with flocking techniques. The units use the roadmap created by PRM to guide their motion toward the goal while they use flocking to act as a group and avoid local collisions. While this indeed leads to better goal finding abilities, groups still split up easily.

Li and Chou [10] developed an approach that allows dynamic structuring of the units such that the centralized planning of the motions is greatly improved. Again, this approach lacks the ability of guaranteeing coherence.

Crowd simulation also investigates the movement of large numbers of units in a virtual environment. This research area has received vast amounts of attention over the last few years, such as [11,19]. Although related to our research, the area has a different goal. The global idea behind crowd simulation is to have virtual units behave in a natural way, interacting with each other, based on (social) rules. The emergent behavior of the units is then studied. Our research in contrast is focused on computing high-quality paths directly.

Related to crowd simulation is pedestrian and traffic flow simulation. Helbing uses forces related to physics to simulate the behavior of pedestrians and traffic [4,5]. These are related to the social potential fields mentioned above, and have similar drawback for our setting.

Often formations are a good way of maintaining coherence. Balch and Hybinette used social potentials to generate scalable formations for mobile robots [1]. They devise a number of forces (potentials) to steer the robots, for example an obstacle avoidance and a formation keeping force. Formations are also very important in games. For example, troops should move in formation over a battlefield. Pottinger describes how these paths can be generated and implemented in two articles [12,13].

1.2 Overall problem setting and approach

In our problem setting we are given a virtual environment in which a group of units must move from a given start to a given goal position. The units must avoid collisions with the environment and with each other, and should stay together as one group. The units are modeled as discs (or cylinders) and are assumed to move on a plane or terrain.

First, our approach computes a backbone path for a single unit. This path defines the homotopic class used by all units. Two paths P_0 and P_1 are said to be in the same homotopic class only if P_0 can be continuously distorted in P_1 without intersecting the obstacles. Next, a corridor is defined around the backbone path and all units will stay in this corridor. Units move through the corridor using attraction points on the backbone path. By limiting the distance between the attraction points for the different units coherence of the group is guaranteed.

The approach is capable of generating these paths in real-time, thus making it possible for the technique to be used in virtual environments, and especially games.

1.3 Outline of the paper

This paper is organized as follows. The global approach is further detailed in Section 2. Section 3 discusses what requirements are needed for coherent group motion and how our approach meets these requirements. Experiments to show the usability and performance of the approach are discussed in Section 4. Finally, some conclusions are presented in Section 5.

2 Group Path Planning Approach

This section describes the problem setting in more detail, and gives an overview of the approach taken to solve it.

2.1 Problem setting

We are given a geometric description of the virtual environment, either 2D or 3D, in which a group of units must move. In case of a 3D environment, the units are assumed to move on a plane or terrain, resulting in a 2D motion planning problem. The units are represented as a circle (cylinder in 3D) of radius r . Then we can describe the configuration of a unit with two parameters (degrees of freedom), namely (x, y) in world coordinates. This position of unit i is referred to as U_i . The position is dependent on the time, thus at a fixed time t the position of the unit is given by $U_i(t)$. We denote the individual configuration space of unit i as C_i . The configuration space of the group is denoted as $C = C_1 \otimes C_2 \otimes \dots \otimes C_n \subset R^{2n}$.

The problem is characterized as follows. The units $i \in [1, n]$ need to reach a goal area from a start area without colliding with the environment and with each other, while at the same time they must stay together as a coherent group. The goal is reached when all units are within the goal area. The relative position of the units in the start and goal area is not relevant. Coherence in this problem setting is

defined informally as: the units will each traverse a path that is in the same homotopic class as all the paths of the other units, and the group members will stay close together.

2.2 Global solution

The technique we propose is a two-phase approach consisting of the following phases:

1. Plan a path for a single unit in the correct homotopic class.
2. Plan the motion of the units in the group using this path.

We will refer to the path found in the first phase of the approach as the *backbone path* of the group. Every entity should individually be able to follow this path. However, in an ideal situation, the units in the group will not traverse the path exactly. The units will feather out in the neighborhood of the backbone path. The second phase of the approach will allow for this, while at the same time keeping the paths of the units in the same homotopic class as the backbone path. In other words, the group uses the backbone path as a guide, but traverses it in a coherent manner.

Because the second phase of approach is the major novelty, we will mainly focus on that phase in this paper. The first phase has been studied extensively in the past and we will only briefly treat it here.

2.3 First phase: backbone path

The first phase of the approach consists of finding the backbone path. Since every unit should be able to traverse the path, the clearance on the path should be bounded by a minimum value, namely the radius of the enclosing circle/cylinder. The backbone path can thus be defined as follows:

Definition 2.1 (Backbone path). *A backbone path $\Pi : [0, 1] \rightarrow R^2$ is a path in the 2D workspace, where the clearance at every point on the path is at least the radius of the enclosing circle/cylinder of the units.*

Although finding a path with a minimum clearance of the radius of the enclosing circle is required, we prefer a larger clearance. In the second phase, a larger clearance along the backbone path leads to more coherent behavior.

As mentioned in the introduction one of the most widely used techniques for motion planning is the Probabilistic Roadmap Method (PRM). There are a number of approaches to find a path using the PRM for a point with a minimum clearance. Geraerts and Overmars give an overview of these techniques in [3].

Although the PRM is a good method for finding the backbone path, other techniques could be used as well. For example, it would suffice to calculate the medial axis (Voronoi diagram) of the environment and search this to find a path with enough clearance. Waypoint graphs, used often in computer games, are also very useful for finding the backbone paths.

Again, the larger the clearance around the path, the more coherent the group can be. Hence, we prefer paths with a high clearance. On the other hand, high-clearance paths might be long, resulting in a long motion to the goal. The precise choice depends on the application and will not be studied here. So for the rest of the paper we assume that a backbone path is given.

In the rest of this paper we assume the backbone path is given, as well as the clearance on every point on the backbone path. The second phase of the approach keeps the entities inside this clearance around the backbone path meaning that the obstacles can be discarded in that phase.

2.4 Second phase: group movement

The second phase of the approach uses the backbone path of the first phase. It extends the backbone path to form a corridor in which the units can move freely. Inside this corridor, the unit paths are generated using social potential fields. By choosing a force such that the units are attracted to certain points on the path, the units are constrained in their movement and stay in a coherent group. The next section will enunciate this phase.

3 Path Finding for the Group

The result from the first phase of the approach is a backbone path. However, it is not trivial to see how the backbone path can be used to plan the paths for the units in the group. A leader following approach could be used here, where the leader follows the backbone path and the followers the leader. But, this approach does not guarantee that all units will stay together as a coherent group. This means that we have to use a more elaborate strategy. In this section we introduce the requirements needed to plan the paths for a coherent group. This is followed by a description how these requirements are to be met.

3.1 Requirements

Before diving into the details of the method, let us try to provide some intuition. This intuition will result in two requirements that the approach must meet.

When looking at a coherent group of people in a real environment, we see that the group stays together in two ways. First, the lateral dispersion of the group is limited. Humans in a group will not spread out too far, i.e. only a few people will walk side-by-side, depending on the group size and the available space in the environment. Second, this group contains different people, where some will walk at a higher pace than others. The fastest persons in the group will wait for the slower people when for example a corner is taken, such that the slower persons will not loose track of the group. Thus, the longitudinal dispersion is also limited.

So, when planning the paths for a group of units in a virtual environment these two dispersion, lateral and longitudinal, should be restricted. The next sections will provide the details on how our approach meets these two requirements.

3.2 Corridor

Along the backbone path $\Pi(s)$, we can define circles centered at $\Pi(s)$ for all $s \in [0, 1]$. Let $\rho_c(s)$ denote these circles. The radius of these circles is the minimum of the real clearance in point $\Pi(s)$

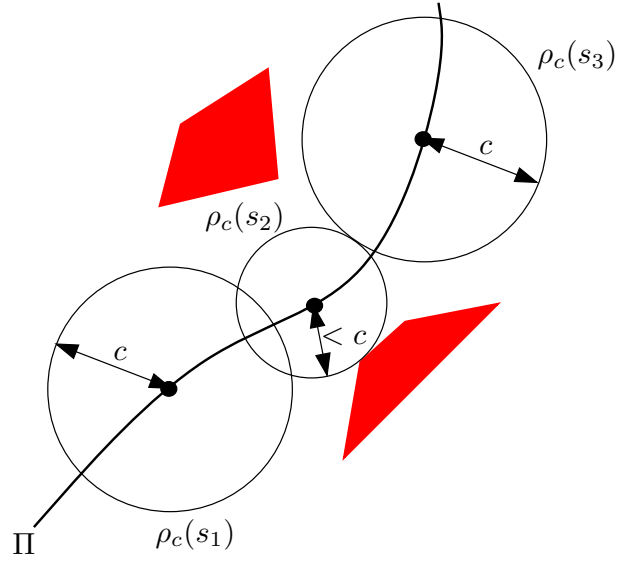
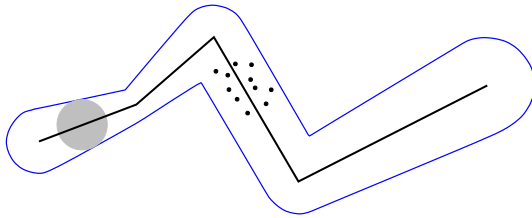
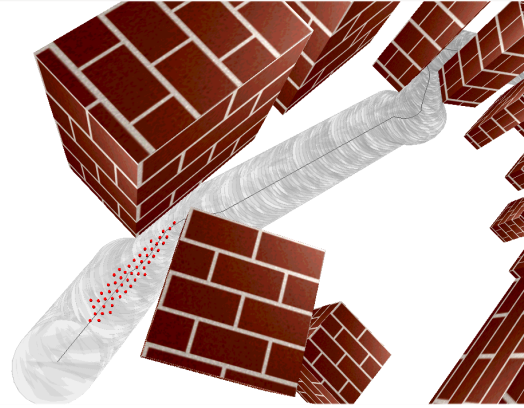


Figure 2: The first and third circle, $\rho_c(s_1)$ and $\rho_c(s_3)$, have a radius of c , since their clearance in the environment is larger than c . However, the clearance of point $\Pi(s_2)$ is clearly smaller than c , resulting in a smaller radius for $\rho_c(s_2)$.



(a) The corridor is formed by the union of all circle $\rho_c(s)$ along path Π . The units must move inside this corridor without colliding with the environment, and satisfying the requirement of lateral dispersion. In this figure, a group of units is depicted as black dots.



(b) A corridor visualized in the developed software.

Figure 3: The corridor. Both a schematic depiction and a real generated corridor.

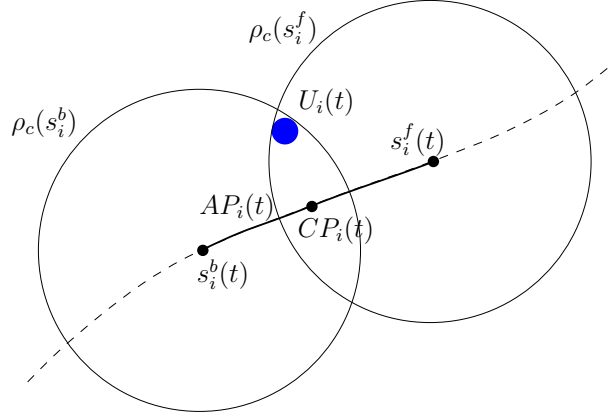


Figure 4: The attraction points at time t for a unit i that is on position $U_i(t)$, denoted by the solid line between $s_i^b(t)$ and $s_i^f(t)$.

and a chosen constant c . Fig. 2 depicts three of these circles, $\rho_c(s_1)$, $\rho_c(s_2)$ and $\rho_c(s_3)$. The first circle ($\rho_c(s_1)$) has radius c since the clearance on that point on the path is larger than c . The second circle has a smaller radius than c , due to the smaller clearance around the point $\Pi(s_2)$. The last circle has again the full radius c . In subsequent discussions this constant c is used to upper bound the lateral dispersion of the group.

The union of circles $\rho_c(s)$ over all $s \in [0, 1]$ forms an area that we call *the corridor*, along the path Π (see Fig. 3). As all circles are collision free, the corridor is also collision free. This corridor forms a passageway through which the units will move. When the units stay inside this corridor, their movement is always collision free (with respect to the environment).

3.3 Lateral dispersion

We define the lateral dispersion as follows:

Definition 3.1 (Lateral dispersion). *The lateral dispersion $D_{lat}(t)$ at time t is the maximum distance of any unit U_i to its closest point on the backbone path Π :*

$$D_{lat}(t) = \max_i \left\{ \min_{s \in [0,1]} \|\Pi(s) - U_i(t)\| \right\} \quad (1)$$

The corridor forms a free passageway through which the units should move. This means that we want to keep the units inside the corridor. This requirement can be reformulated in terms of attraction points, defined as follows:

Definition 3.2 (Attraction point). *An attraction point for unit i at time t is a point s on the backbone path such that unit i is inside the circle $\rho_c(s)$. For every unit, there is a set of possible attraction points on the*

backbone path (see Fig. 4):

$$AP_i(t) = \{s : s \in [0, 1] : U_i(t) \in \rho_c(s)\} \quad (2)$$

A unit is within the corridor, if and only if the set $AP_i(t)$ is not empty. So we will make sure that $AP_i(t)$ never becomes empty. This lead to the required limited lateral dispersion.

Nevertheless, how does the corridor relate to lateral dispersion? The following theorem does exactly that.

Theorem 3.1 (Limited lateral dispersion). *If for all i and all t $AP_i(t)$ is not empty, the lateral dispersion $D_{lat}(t)$ is upper bounded by c .*

Proof It is clear that the closest point on the backbone path to a unit i

$$CP_i(t) = \arg \min_{s \in [0, 1]} \|\Pi(s) - U_i(t)\| \quad (3)$$

is always contained in $AP_i(t)$. Since this closest point is the center of a $\rho_c(s)$ that contains U_i , it follows that

$$D_{lat}(t) \leq c \quad (4)$$

This means that by varying the value c we can limit the lateral dispersion. This is the first step in getting to coherent paths. By selecting a larger value for c the group will be allowed to feather out more, while a smaller value c means that the group will be more narrow. If we would take $c = r$, the radius of the enclosing circle of the units, the group will move in a single file, which in general is undesirable. A larger value for c would be more appropriate.

3.4 Longitudinal dispersion

The second requirement to accomplish coherence is the limitation of longitudinal dispersion. For this we need also a measure. As we already defined the attraction points in the previous section, we can use these. First we need to define the minimum and maximum attraction point.

Definition 3.3 (Minimum and maximum attraction points). *The minimum attraction point $s_i^b(t)$ for a unit i at time t is defined as:*

$$s_i^b(t) = \min AP_i(t) \quad (5)$$

The maximum attraction point $s_i^f(t)$ for a unit i at time t is defined as:

$$s_i^f(t) = \max AP_i(t) \quad (6)$$

These points are depicted in Fig. 4. Superscripts b and f depict back and front, respectively.

We can also define the minimum and maximum group attraction points. Using these, we will define a region, i.e. a part of the corridor, in which all units must remain. This region, called the *group region*, can be used to define the longitudinal dispersion.

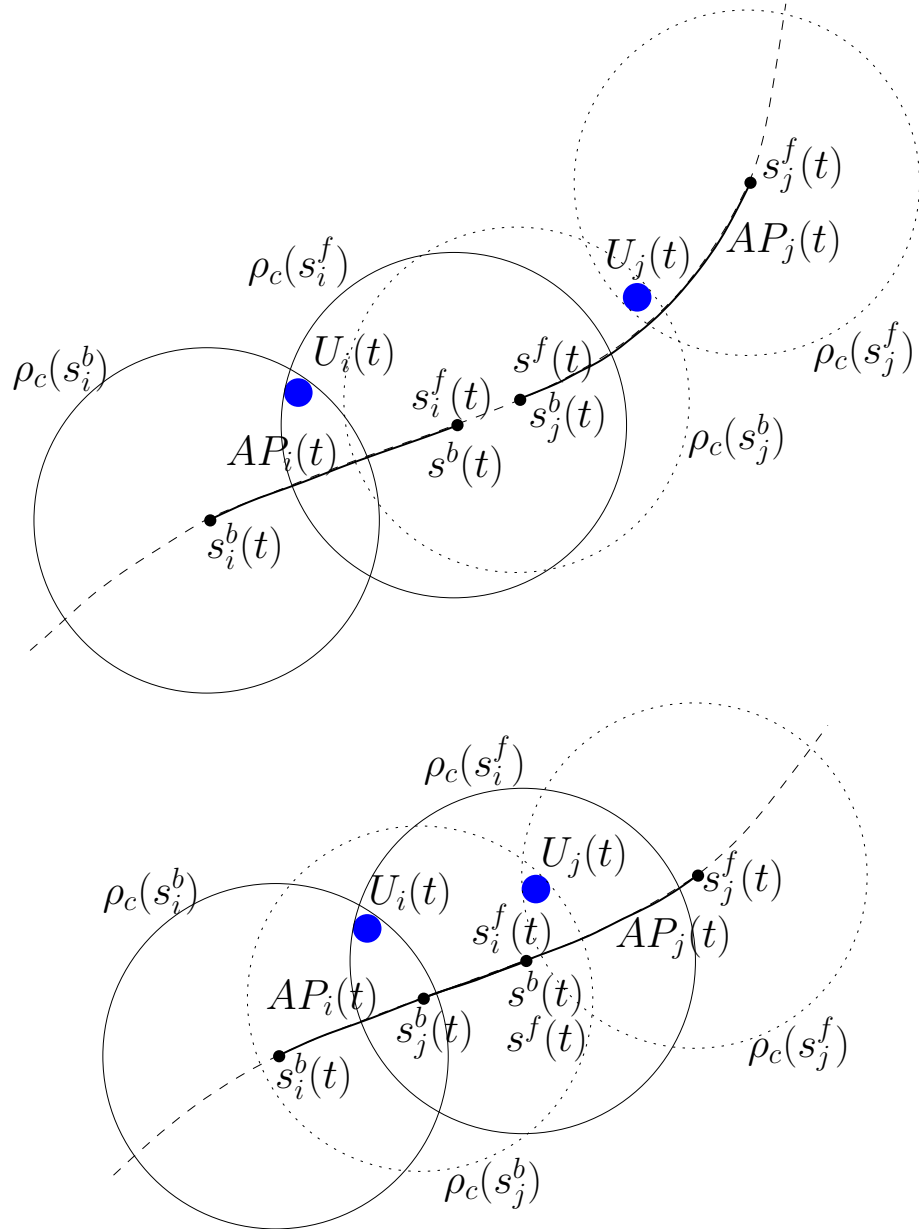


Figure 5: The construction of the minimum and maximum group attraction points. In the upper image the maximum and minimum are not the same, whereas in the lower image they are.

Definition 3.4 (Minimum and maximum group attraction points). *The minimum group attraction point $s^b(t)$ at time t is defined as being the minimum over all maximum attraction points:*

$$s^b(t) = \min_i \{s_i^f(t)\} \quad (7)$$

The maximum group attraction point $s^f(t)$ at time t is defined as being the maximum over all minimum attraction points, but not smaller than the minimum group attraction point $s^b(t)$:

$$s^f(t) = \max \left(\max_i \{s_i^b(t)\}, s^b(t) \right) \quad (8)$$

This definition might at first sight be counter-intuitive. Our goal though is to keep the distance between $s^b(t)$ and $s^f(t)$ along the path as small as possible, while still guaranteeing that each unit has an attraction point between them.

The construction of the minimum and maximum group attraction points is depicted in Fig. 5 for two units i and j . The upper diagram shows the situation where the minimum of the maximum attraction points is smaller than the maximum of the minimum attraction points. In the lower diagram the maximum group attraction point is the same as the minimum group attraction point.

The distance between the minimum group attraction point $s^b(t)$ and the maximum group attraction point $s^f(t)$ along the backbone path can be used as a measure for the longitudinal dispersion, i.e.:

Definition 3.5 (Longitudinal dispersion). *The longitudinal dispersion is the distance between the minimum and maximum group attraction point along the backbone path:*

$$D_{long}(t) = dist_{\Pi} (s^b(t), s^f(t)) \quad (9)$$

where $dist_{\Pi}$ denotes the distance along the path Π .

By taking the union of all circles between $s^b(t)$ and $s^f(t)$ an region, called the group region, is formed that includes all units. This region is contained inside the corridor, since both the corridor and this region are the unions of subsets of the same circles (see Fig. 6).

Definition 3.6 (Group region). *The group region is the union of all circles $\rho_c(s)$ between $s^b(t)$ and $s^f(t)$:*

$$GR(t) = \bigcup_{s \in [s^b(t), s^f(t)]} \rho_c(s) \quad (10)$$

As long as this group region is not too large, the longitudinal dispersion is also bounded. So we want to enforce that the area of the group region remains bounded by some value A (A must be chosen such that all units will fit in the region and have some spare to move around). To enforce this, we limit the actual attraction points for the units. Let $s^b(t)$ be the minimum group attraction points at some time t , which in effect indicates where the back of the group is located. We limit the position of the units by defining the maximum allowed attraction point as follows (see Fig. 6):

Definition 3.7 (Maximum allowed attraction point).

$$s_a^f(t) = \arg \max_{s \in [s^b(t), 1]} \left\{ \bigcup_{u \in [s^b(t), s]} \rho_c(u) \leq A \right\} \quad (11)$$

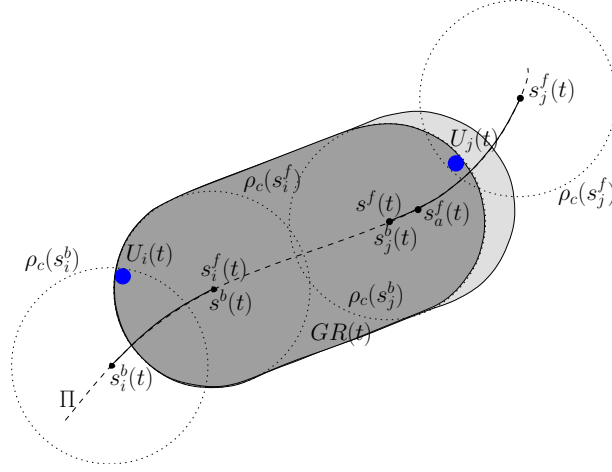


Figure 6: The group region is denoted by $GR(t)$ and depicted as the dark gray area. This region contains all units and the area of this region is smaller than a constant A . The maximum allowed attraction point $s_a^f(t)$ is determined using the light gray region, of which the area is exactly A .

We will ensure that the units move in such a way that all units have an attraction point inside the interval $[s^b(t), s_a^f(t)]$. This way the longitudinal dispersion is limited.

We will limit the lateral dispersion by means of the maximum corridor width c , while limiting the longitudinal dispersion by setting the maximum area of the group region A . With these two constants, c and A , we can influence the coherence of the group. Increasing the maximum area of the group region A allows the group to spread more along the backbone path. Increasing the maximum corridor width c makes the group wider, but decreases the longitudinal dispersion if the area A is kept constant.

3.5 Social Forces

To generate the paths inside the corridor, we are going to use an approach based on social (potential) force fields. As discussed in the introduction, this technique is used both in virtual environments and in pedestrian/traffic flow simulation. The approach uses forces to drive the units toward a position further on their path. There are a number of forces needed for generating the paths. First, the units should not collide with each other, requiring a repulsive force between units. Second, the units should not move outside the corridor, thus a repulsive force from the boundary of the corridor inward onto the units is required. Third, the units should 'feel' the urge to go forward, so a driving force forward is needed. The second and third force can be combined as is shown below.

Collision avoidance force. The collision avoidance force between units is dependent on the distance between the units. The closer the units are, the higher the repulsive force. This force is not Newtonian, i.e. the force that one unit experiences is not the same as the force the other unit experiences. The reason for this is that one unit might be outside the region of influence of the other. We model this using a cone of influence for every unit, or *visibility cone*. This visibility cone is a cone in the direction

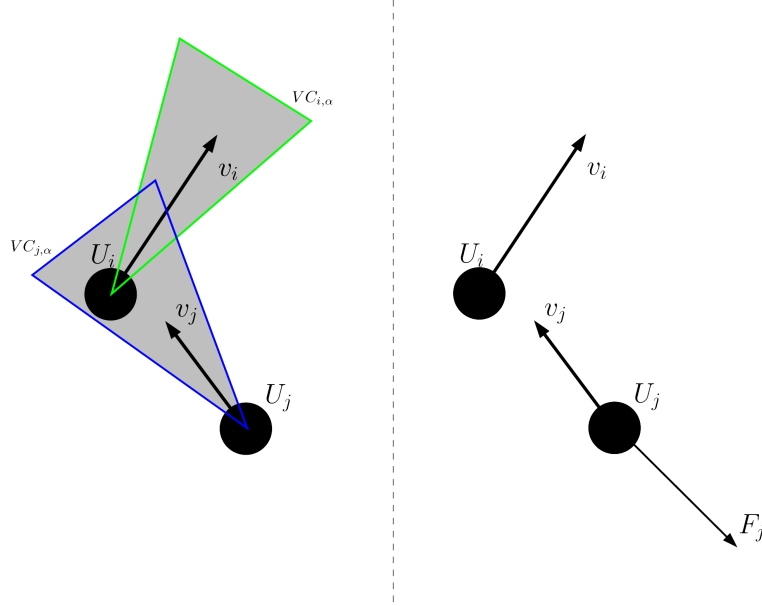


Figure 7: Unit U_i is inside the visibility cone of unit U_j resulting in a force on U_j by U_i , unit U_j wants to avoid unit U_i . Conversely, unit U_i will not ‘feel’ the presence of unit U_j , since U_i can not see U_j .

of movement, with a top angle α . Whenever a unit A is within this cone of some other unit B, B will be pushed away from A (see Fig. 7).

Corridor force and driving force. The corridor force and the driving force are related. In fact, they can be regarded as the same force. The attraction points play an important role in this. By choosing the actual points of attraction as far as possible on the path Π , i.e. the minimum of $s_i^f(t)$ and $s_a^f(t)$, the units are forced forward if this is possible, and hold back if the longitudinal dispersion requires this. However, the units are always kept inside the corridor (the force is always pointed inward).

Time integration of the forces and position updating. The result from the method outlined above is a system of positions, velocities and forces. In fact, it is just a particle system. This particle system can be solved using any time integration scheme, for example Euler Forward, Runge Kutta or Verlet, resulting in positions and velocities for all units over time. However, for the forces to be calculated correctly in every step of the numerical integration, the actual points of attraction should also be updated.

Updating the actual points of attraction. To let the group advance along the backbone path, the actual points of attraction need to be updated in every time step after calculating the new positions U_i of the units. These actual points of attraction are updated according to Algorithm 1. Here the actual points of attraction are denoted by $p_i^a(t)$ In this algorithm, the time progresses with steps of δt . Let $t' = t - \delta t$ denote the previous time step.

Formations As described in [1], social force fields can be used to generate formations among the units. To accomplish this in our approach, an extra force should be incorporated that makes the units

Algorithm 1 Update actual points of attraction

```
1: for  $i = 0$  to  $n$  do
2:    $p_i^a(t) \leftarrow \max_{s \in [s_i(t'), 1]} \{ \forall s'' \in [s_i(t'), s] : U_i(t) \in \rho(s'') \}$ 
3: end for
4:  $s^b(t) \leftarrow \min_i \{ p_i^a(t) \}$ 
5:  $s_a^f(t) \leftarrow \max_{s \in [s^b(t), 1]} \left\{ \text{area}(\bigcup_{s'' \in [s^b(t), s]} \rho(s'')) < A \right\}$ 
6: for  $i = 0$  to  $n$  do
7:   if  $p_i^a(t) > s_a^f(t)$  then
8:      $p_i^a(t) \leftarrow s_a^f(t)$ 
9:   end if
10: end for
```

keep the formation.

3.6 Resulting paths

As mentioned, the choice of maximum corridor width c and group area A has effect on the coherent behavior of the group. To test this, we implemented the system and ran various experiments. In Fig. 8, a typical results from varying the group width c is depicted. Here, the first image shows a group of 50 units with a large width. The group stays close together, but is wide spread. The second image shows the same group, but this time with a smaller group width. The third image show the same group, but now with a very small group width. Since the corridor is now very small the units move right after each other.

The resulting paths exhibit coherent group behavior, as demonstrated in Fig. 8. Besides that, there also appears real coherent behavior like waiting for other units that lag behind. Fig. 8(d) shows a screenshot of a group passing a corner in the environment from top to bottom. Some units lag behind in this corner. Units in front wait for these units before continuing along the path. If the group area was chosen larger, the units in front would not have waited.

4 Performance

We have implemented the presented approach to test its effectiveness and applicability. We used Microsoft Visual Studio .Net 2002 with the C++ programming language. All experiments were run on a Pentium IV 2.4 Ghz computer with 1 Gb RAM running Microsoft Windows XP.

The nature of the approach is such that the complexity of the environment is only of influence when finding the backbone path. However, this is a general path planning problem for which many experiments have already been conducted (see [3]). So here we concentrate on the second phase in which the only things that matter are the clearance and the group size. We used the cluttered environment depicted in Fig. 9. This is a 3D environment where the units move over a polygonal terrain. The obstacles are boxes and cylinders. The radius of the units is 1. All distances are expressed in this unit radius. The test environment is 1200 wide and 940 long, consisting of polyhedral obstacles.

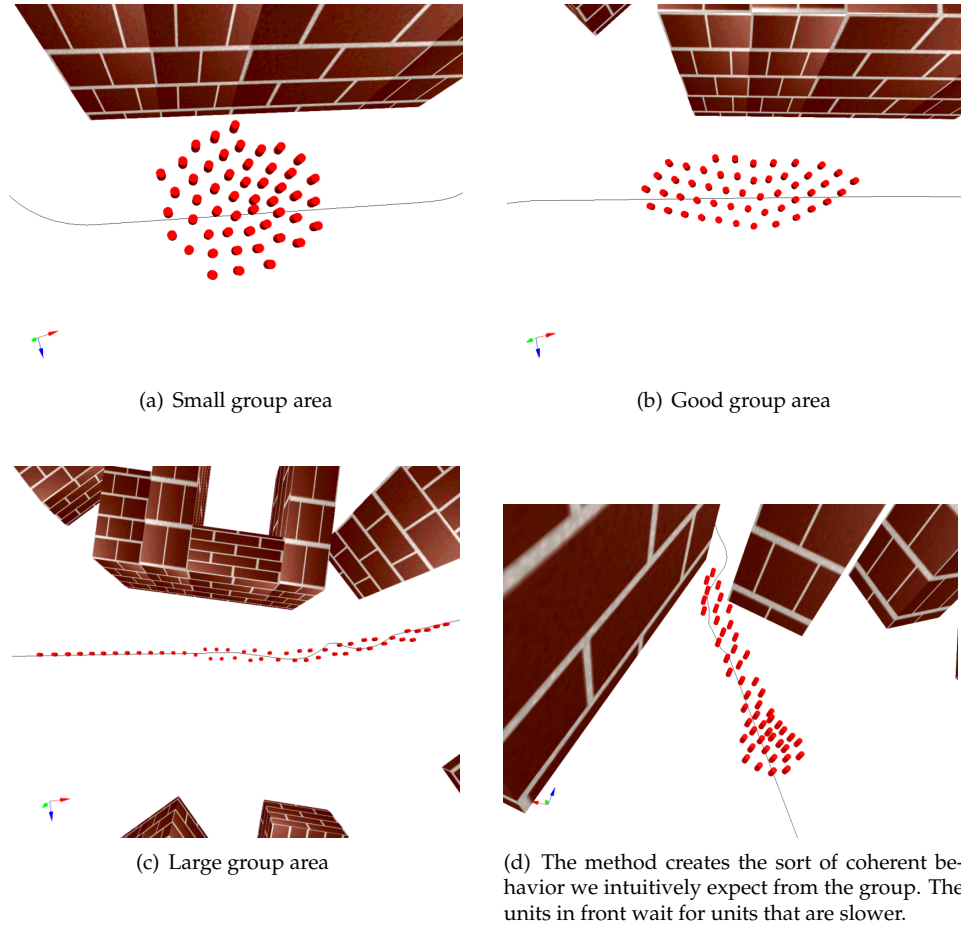


Figure 8: Group behavior for different choices of width c (fig. a-c), as well as the resulting behavior (fig. d)

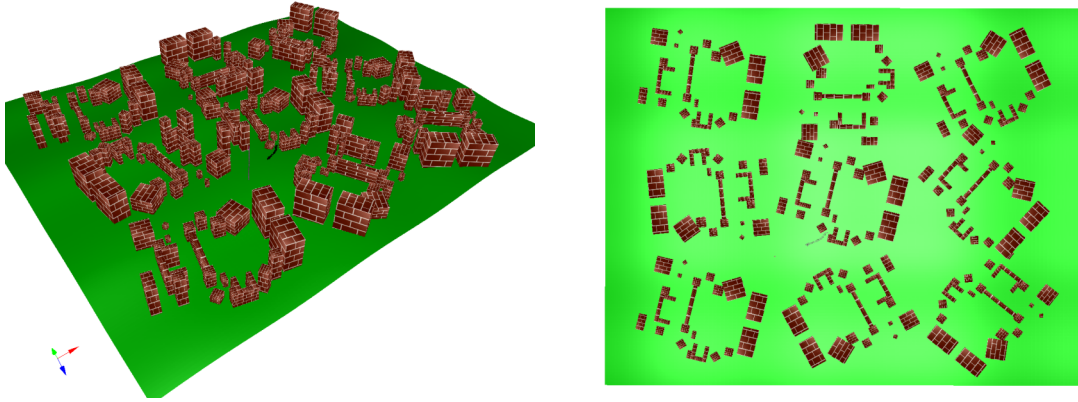


Figure 9: The virtual environment used for testing.

Beside the quality of the generated path, another important requirement for the approach is that it is fast, such that it can be used in interactive applications. Given a path for a single unit, the backbone path, the method should generate the paths of the units very quickly. We conducted a number of experiments. First, we tested how fast the method is in an absolute sense, i.e. how long does it take to produce a path on average. This is done for various group sizes. Second, we experimented with typical application settings to show that the technique is suitable for real-time applications.

4.1 Group size dependence

To test the performance of the technique we generated 400 paths from random start and goal position in the environment (Fig. 9). We measured the average time it takes to calculate one iteration for varying group sizes. An iteration is the updating of the positions, velocities and attraction points as described in Section 3.5. The size of the group was varied between 5 and 100 units per group. The results are depicted Figure 10. For a group size of 50 units the calculation time per iteration is 0.002 seconds.

Clearly noticeable is the quadratic dependence of the time on the group size. This is to be expected since the units are close together and are influenced by each other resulting in the quadratic dependence.

4.2 Usability in real-time applications

To test the usability in real-time application, we constructed a demo application. In this demo application, the units are allowed to move at a maximum speed of 15 per second (remember that the measure of distance is the unit radius). From experiments with this demo application we can conclude that, in order to produce paths reliably (that is without collisions), we are allowed to move 3 per iteration. Although the physics engine of a typical computer game runs at 50-60 Hz, the positions and velocities of the units in our demo application need to be updated at 5 iterations per second.

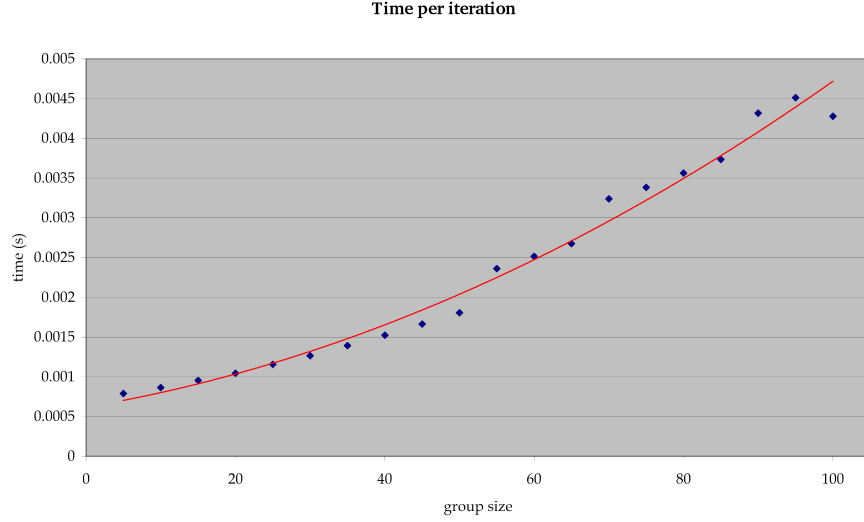


Figure 10: The time needed for updating the positions, velocities and attraction points, as a function of the group size.

group size	processor usage
20	0.5%
40	0.8%
60	1.2%
80	1.8%
100	2.4%

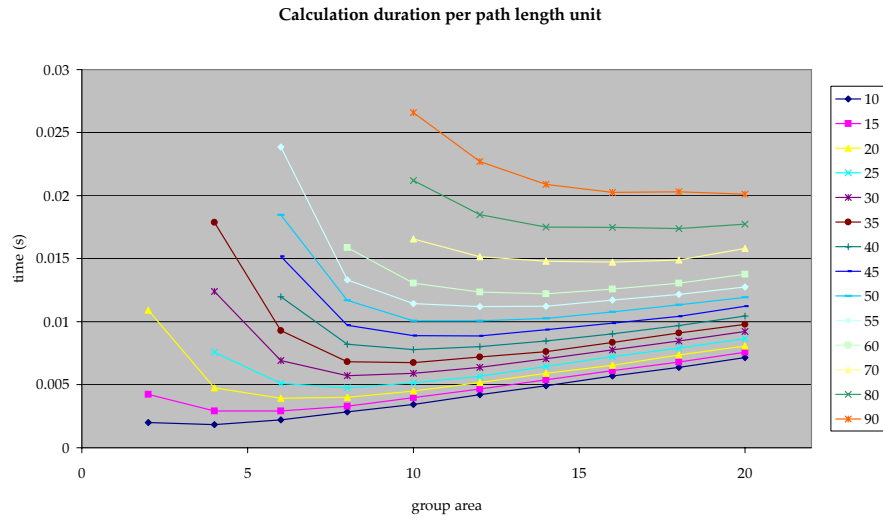
Table 1: Processor usage in the demo application.

Every iteration (pre-)calculates about 0.2 seconds of movement. The resulting processor usages for different group sizes are given in Table 1. For example, it takes 0.02 seconds to calculate a second of movement for a group of 100 units, i.e. only 2% of processor time.

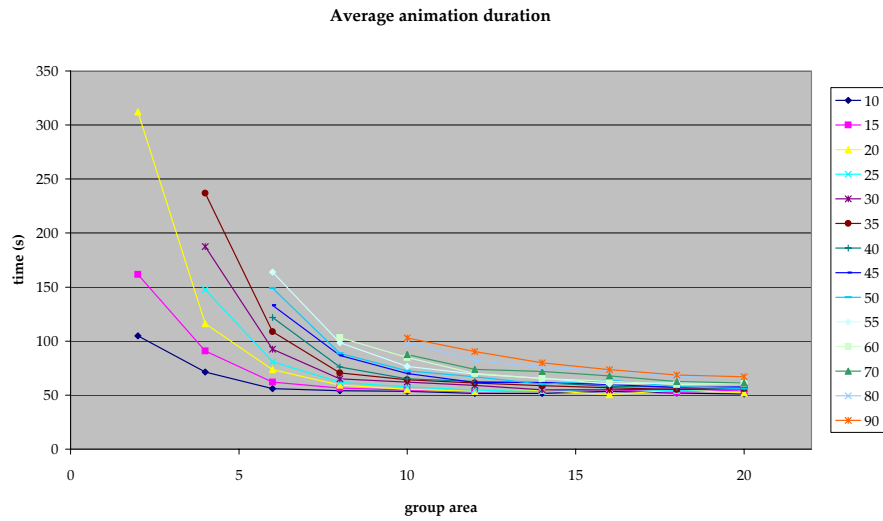
4.3 Group area dependence

The performance of the method is directly related to the group area since the area of the union of discs is to be calculated often. To test the dependence of the group area on the performance, we generated 400 random paths in the test scene (Fig. 9) for increasing group area (for different group sizes). The results are depicted in Fig. 11(a). The group area has also influence on the resulting animation duration. A smaller area makes the group more compact (more coherent) but hinders the speedy traversal of the path, since more units interfere with each other. This can be seen in the result depicted in Fig. 11(b).

From the results we can deduce that there is an optimal group area for every group size. When the



(a) The calculation duration per unit length depending on the group area. As the group gets bigger, the calculation duration increases as expected. For every group size there is an optimal group area. This is the group area such that the path can be calculated as fast as possible. There is also a minimum group area, since for a certain small area the units do not fit inside this area, thus immobilizing them.



(b) The average animation duration over 400 random paths as a function of the group area for different group sizes.

Figure 11: Results from the group area dependency experiments.

group area gets smaller than the optimal value, the units are hindered more in their traversal of the path, until at some point the units are not capable of traversing the path. The units are then pushed too close together due to the small group area. However, when the group area is larger than the optimal value, the units are already moving optimally along the path, so the extra group area only adds to the calculation duration.

Whenever the group area is large enough, i.e. larger than a minimum group area, the method is always able to find the paths for the units in the group. None of the numerous generated random paths ever resulted in paths not being found.

5 Conclusions

We presented a novel approach to finding paths for coherent groups of units. Experiments show that the method is capable of finding paths that exhibit realistic coherence. The approach is such that this coherence can be guaranteed. The level of coherence can be adjusted by two parameters, namely the corridor width and the group area. A requirement is that a path for a single unit can be easily found. However, there exist numerous ways of finding such paths and virtual environment applications will in general already have this functionality.

The performance of the approach is promising. The paths are generated in real-time, even for larger groups. Experiments show that the processor resources required are low, only 2.4% for groups consisting of up to 100 units. We expect that careful implementation could considerably lower this even further.

We are currently extending the approach in various ways. In the method presented no obstacles are allowed inside the corridor generated by the backbone path. This means that the group will evade even small obstacles, like for example a light-pole. To circumvent this, we are currently researching the possibilities of allowing small or insignificant obstacles inside the corridor.

Joining and splitting of groups, a common event in virtual environments, is also one of the research items currently under investigation. Furthermore, we study the planning of multiple groups.

6 Acknowledgments

The authors would like to thank Dennis Nieuwenhuisen for the development of the collision detection and visualization library Callisto and the roadmap manipulation library Atlas.

This research was supported by the Dutch Organization for Scientific Research (N.W.O.). This research was also supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2001-39250 (MOVIE - Motion Planning in Virtual Environments).

References

- [1] Tucker Balch and M. Hybinette. Social potentials for scalable multi-robot formations. In *IEEE International Conference on Robotics and Automation (ICRA '00)*, volume 1, pages 73 – 80, April 2000.
- [2] O.B. Bayazit, J.-M. Lien, and N.M. Amato. Better flocking behaviors using rule-based roadmaps. In *Algorithmic Foundations of Robotics V, Springer Tracts in Advanced Robotics 7*, pages 95–111. Springer-Verlag Berlin Heidelberg, 2004.
- [3] Roland Geraerts and Mark H. Overmars. Clearance based path optimization for motion planning. In *International Conference on Robotics and Automation (ICRA)*. IEEE Press, San Diego, CA, 2004.
- [4] D. Helbing. A mathematical model for the behavior of individuals in a social field. *Journal of Mathematical Sociology*, 19(3):189–219, 1994.
- [5] D. Helbing and P. Molnr. Social force model for pedestrian dynamics. *Physical Review*, 51:4282–4286, 1995.
- [6] Arno Kamphuis and Mark H. Overmars. Motion planning for coherent groups of entities. In *IEEE Int. Conf. on Robotics and Automation*. IEEE Press, San Diego, CA, 2004.
- [7] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *International Conference on Robotics and Automation (ICRA)*, pages 2138–2139. IEEE Press, San Diego, CA, 1994.
- [8] L. Kavraki, P. Švestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:556–580, 1996.
- [9] S. Leroy, J.-P. Laumond, and T. Simeon. Multiple path coordination for mobile robots: A geometric algorithm. In *International Joint Conference on Artificial Intelligence*, pages 1118–1123, 1999.
- [10] Tsai-Yen Li and Hsu-Chi Chou. Motion planning for a crowd of robots. In *International Conference on Robotics and Automation (ICRA)*. IEEE Press, San Diego, CA, 2003.
- [11] S. Raupp Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):152–164, 2001.
- [12] Dave Pottinger. Coordinated unit movement. *Game Developer*, pages 42–51, January 1999.
- [13] Dave Pottinger. Implementing coordinated movement. *Game Developer*, pages 48–58, February 1999.
- [14] J. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 431–459. A. K. Peters, Wellesley, MA, 1995., 1995.
- [15] C.W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

- [16] C.W. Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.
- [17] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1994.
- [18] G. Sanchez and J.C. Latombe. Using a prm planner to compare centralized and decoupled planning for multi-robot systems. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2112–2119, 2002.
- [19] B. Ulicny and D. Thalmann. Crowd simulation for interactive virtual environments and vrtraining systems. In *Eurographics Workshop on Animation and Simulation*, pages 163–170. Springer-Verlag, 2001.
- [20] P. Švestka and M.H. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23:125–152, 1998.