Motion Planning for Coherent Groups of Entities

Arno Kamphuis

Mark H. Overmars

institute of information and computing sciences, utrecht university

technical report UU-CS-2003-036

www.cs.uu.nl

Motion Planning for Coherent Groups of Entities

Arno Kamphuis Mark H. Overmars

November 2003

Abstract

Motion planning for multiple entities is a challenging problem in today's virtual environments. In this paper we develop an innovative approach to motion planning for groups of entities, where coherence is taken into account.

We model the group by a deformable shape. Next, we use the Probabilistic Roadmap Method to plan the (global) motion of the shape. For this, we develop new sampling techniques and local planners. A new approach, called Group Potential Fields, is also introduced as a means to determine the local motions of the entities inside the deformable shape. Global and local motions are then combined to find the required paths for the entities.

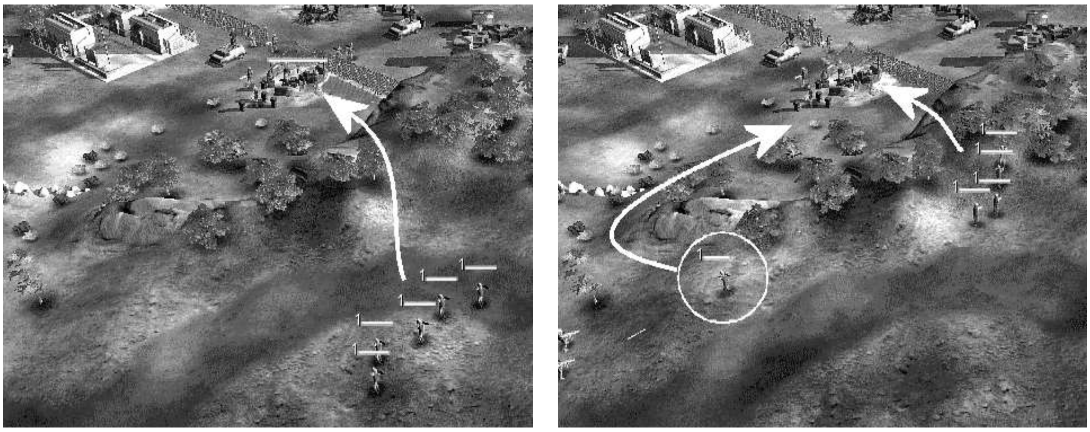
Experiments show that the approach is able to find paths for groups of varying sizes, after limited preprocessing time, where the groups stay coherent.

1 Introduction

Virtual environments are often populated with moving entities. Games in particular, but also other virtual environment applications, contain (very) large numbers of entities moving around. The entities 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.

In current applications, the quality of such group behavior is in general not very good, the primary reason being the need to compute the paths in real-time. As a result, most techniques

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



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

(b) The group splits up, with the unwanted situation of loosing troops.

use ad-hoc and local approaches, in which the entities all have a similar goal, but 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 Figure 1. Here, one individual decides to take a different route. This is highly unwanted behavior and in this particular example it leads to immediate death.

1.1 Previous work

Motion planning has 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 entities.

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 [12]. His *boids*-model described the behavior of the entities in a group using only local rules for the individual entities. Later, Reynolds extended the technique to include autonomous reactive behavior [13]. The idea is that entities steer themselves in such a way that they avoid collisions with other entities and the environment, while at the same moment they try to align themselves to other entities and to stay close to the other entities. 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 entities a goal they will move toward the goal together. The big drawback of this approach is that the entities 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 Figure 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 [15]. Different entities try to find a path through the grid while avoiding collisions with each other. This easily leads to entities getting stuck in ways that can only be resolved by rather unnatural motions (or cheating). 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 [11] is a potentially useful technique. It defines potential force fields between entities 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.

In the robotics community, one of the dominating techniques is the probabilistic roadmap approach (PRM) [7, 8]. Efficient probabilistic (centralized) techniques for multiple entities have been developed [16, 18]. They treat the different entities together as one large robotic system. Unfortunately, each entity 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 entities. When n gets larger the running time becomes too large. Also, these approaches require that the number and type of entities 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 entities [9]. Still though, these methods fail when the number of entities grows and the resulting motion is not coherent.

Recently, Bayazit, Lien and Amato [1] have combined the PRM approach with flocking techniques. The entities 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 a new approach that allows dynamic structuring of the entities/robots such that the centralized planning of the motions is greatly improved. Again, this approach lacks the ability of guaranteeing coherence.

1.2 Our contribution

The current techniques for planning the motions of groups of entities, although useful in a number of application, lack the concept of coherence. In this paper, we propose a novel approach to the planning of group motions where coherence is guaranteed.

Rather than planning the motion for individual entities, we plan the motion for the group as a whole. To avoid excessive computing time, we model the group with a deformable shape and plan the motion for this particular shape. We keep the entities inside the shape using a new group potential field technique. Combining the motion of the shape with the internal motion leads to the guaranteed coherent group motion. Experiments show that after limited preprocessing, the technique can solve motion queries in almost real-time.

The new approach we introduce consists of the following steps:

1. Model a group using a deformable shape of sufficient volume.
2. Construct a roadmap of possible motions for the deformable shape using an extension of the probabilistic roadmap approach.
3. To answer a motion query use the roadmap to compute a collision free path for the deformable shape.
4. Plan the motion of the individual entities using a group potential field technique such that they stay inside the deforming shape.
5. Combine the two motions into the final motion for the entities.

We will assume that the entities move on a ground surface or 2D-submanifold, like a terrain. We model the entities as discs, although this can easily be extended. Our results show that the approach is feasible and that, after limited preprocessing time, coherent group motions can be computed efficiently.

1.3 Outline of the paper

This paper is organized as follows: We first define the notions of group and group shape in Section 2. In Section 3 we describe how the the probabilistic roadmap method is used for planning the motion of the shape. Section 4 shows the results of a number of experiments. Section 5 describes the group potential field technique we use to keep the entities inside the shape. Experiments and results of the group potential field technique are given in Section 6. Finally, we draw some conclusions and outline future work in Section 7.

2 Groups

Groups in general are a very subjective thing. That is, humans have the ability to 'feel' if a collection of entities behaves like a group, and the perception of a group is thereby influenced. A computer or algorithm lacks this ability. To be able to create a technique that is able to plan the motion of a group, and of which the resulting motion is perceived by humans as being coherent, we first need to define the notions of group and coherence.

2.1 Group processes and psychology

In psychology, a large diversity of meanings exist associated with the word 'group'. Some psychology theorists see *the experience of fate* as the main base for groups, others look at *social structures*, while again others look at *face-to-face interaction* as being the fundamental base of groups. A number of definitions of a group has been proposed, albeit mainly very subjective ones. For example: '*A group is two or more interdependent individuals who influence one another through social interaction.*' or '*A group is a social system involving regular interaction among members and a common group identity*.' [3].

2.2 Computer science view

In social sciences literature, a well-known definition for a group is as follows [14]:

Definition 2.1 (Group)**.** *A group (or crowd) is a collection of individuals in the same physical environment sharing a common goal.*

This definition lacks in describing the coherence of the group, i.e. it does not mention that entities should have other entities in their proximity. To overcome this problem, we would like to define the so-called *group shape*, which is the shape of a region such that all entities are inside that region. To guarantee that the entities have close proximity to other entities, this region should be simply connected. Furthermore, the region should be relatively fat [2] and have a bounded area. However, in general, the number of parameters needed to describe this shape is of the same order as the number of entities, $O(n)$. To improve the computability of the problem, we would like to reduce this to $O(1)$. This results in the following definition:

Definition 2.2 (S_α-coherent group)**.** *Let S be the class of shapes that can be described with $O(1)$ parameters. Let α be a constant, depended on the number of entities. Let S_α be the subset of S for which the area of the shapes is upper-bounded by α. A group is S_α-coherent if there is an element in S_α such that all entities of the group are contained in that shape.*

In this paper, we will refer to a group as being coherent when the group is S_α-coherent.

By choosing different classes S we will obtain different group behavior. In most of this paper we choose S to be the class of arbitrary oriented rounded rectangles. The variables needed for describing this shape (of constant volume) are: 1) the position, 2) the orientation and 3) the aspect ratio. The rounded corners are chosen because, internally, the entities will never reach the corners of the shape. The rounded corners also give the shape more room to maneuver in the virtual environment. Experiments show that rounded rectangles give enough flexibility to solve many problems. In some situations though we will use an extension, allowing for hinged rectangles (see below).

2.3 Global motion vs. Local motion

As stated, the motion of a group is composed of two motions, first the global group motion and second the local motion inside the group. The global motion of the group is described by the movement of the group shape. However, there is a choice to be made, namely: which variables for describing the group shape should be used to describe the global motion, and which should be included in the local motion?

Suppose we have a group of entities, all inside a deformable rounded box. The positions of the entities inside can be described in a local coordinate frame with its origin in the center of the box, and oriented along the sides of the box. The orientation of the box is thus included in the global motion of the group (the orientation of the box has no effect on the local motion of the entities). The other option is to describe the positions of the entities in a local coordinate frame which is aligned with the coordinate system of the world, and with its origin at the center of the box. The local motion of the entities is thus influenced by the orientation of the box. So the choice of incorporating the orientation of the box into the local or global motion is of great importance. We choose to include the orientation into the local motion of the entities because this makes the combination of the local and global motion the easiest. From the perspective of the entities, the borders of the box seem to move around them.

3 Planning the Motion of the Group Shape

A widely used technique for motion planning in virtual environments is the Probabilistic Roadmap Planner (PRM) approach. This approach consists of two phases, a preprocessing or learning phase and a query phase. During the preprocessing phase, a roadmap graph is constructed, while in the query phase this graph is searched for a path using Dijkstra's shortest path algorithm.

Most of the work of the PRM approach is done in the preprocessing phase. This phase generates random configurations, called samples, and tries to connect these using a deterministic local planner. The free samples are added to the roadmap and, when a local path between two samples is found, an edge is added. After sufficient time, this roadmap adequately covers the space, and the preprocessing phase ends. The generated roadmap can then be queried to find a desired path.

3.1 Configuration space

For the PRM approach to work, we need to define the configuration space. The configuration space is the space of all possible placements of the object for which we want to plan a path. To be able to describe the placement of the object, we need a collection of parameters, called degrees of freedom. The object we are interested in is a deformable box. This deformable box is situated in a 2D world, it can rotate, and its width can change (the volume of our box is constant, so the length depends on the width). This means that our configuration space is a four-dimensional space, two for the position, one for the orientation and one for the width of box.

3.2 Sampling

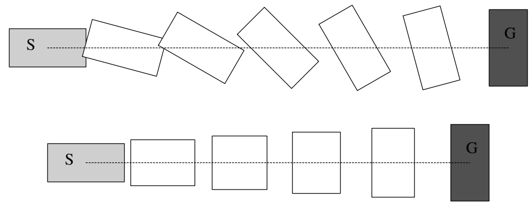
The simplest way of generating new configurations is to uniform randomly pick a value for all the degrees of freedom. However, more advanced techniques are expected to perform better [4, 6]. Early experiments showed that the use of local geometric information to align samples was helpful. This is mainly due to the fact, that for (narrow) passage, not only the position and orientation should be correctly chosen, but also the width of the box. This means that, in configuration space, the passage become narrower and harder to find. By making use of local geometric information this can be improved. The following sampling strategies have been studied.

- **Medial Axis (MA)** The medial axis technique for sampling strives to generate samples on the medial axis of the workspace [6]. This approach has been used previously in motion planning for deformable objects [5]. The technique has an additional benefit in our situation, since we can align our samples with the medial axis. The technique is as follows: uniform randomly generate a point in the workspace (2D). If this point lies in free space, retract it to the medial axis. This retraction method implicitly generates the normal of the medial axis in the retracted point, because we have the two points in workspace that lie closest to the retraction point. Using this information, we can determine the orientation and width of the box, resulting in a new sample (this sample need not be in free space).

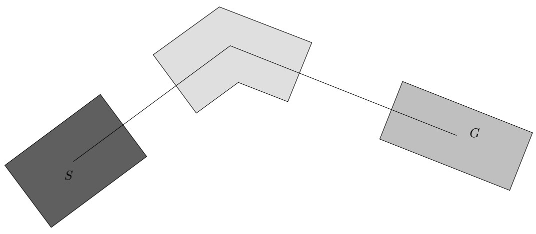
- **Medial Axis using Penetration Depth (MAP)** The standard medial axis method ignores the points chosen in the interior of obstacles. Using the penetration depth we can push the point out of the obstacles, and then retract it to the medial axis [6].

- **Pushing (PUS)** This novel sampling technique picks a uniform random configuration. If the center of the box lies in the free space, and the box itself does not, the penetration depth is used to select the orientation (perpendicular to the penetration direction) and to calculate the width of the box (the distance from the center to the border of the obstacle projected onto the penetration direction). This way, the sample is aligned with an obstacle close by. However, the sample might be in collision with other obstacles.

- **Alignment (ALI)** This new technique tries to find a configuration by uniform randomly picking a point and, if this point is in free space, sweeping a line, of which the length is proportional to the volume of the box, around this point. Using this sweep line we can detect orientations where the line 'touches' the obstacles. These orientations are then used to find the orientation and width of the configuration.



(c) The choice the deformable box local planner has to make depends on the rotations and deformation of the local path. The box can either rotate or deform.



(d) A typical path generated by the hinged box local planner, from the start configuration S to the goal configuration G. Intermediate configurations on the path, except one on the via point, are not depicted for clarity.

Figure 2: The two local planner in our approach.

- **Closest (CLO)** The last new approach works as follows: First, pick a uniform random point in free space, then find the closest point on the obstacles, and finally orient the configurations perpendicular to the direction from the center to the closest point. The distance to the obstacle is taken as the width of the box (maximized to the square root of the volume).

Below we will compare these techniques.

3.3 Local planning

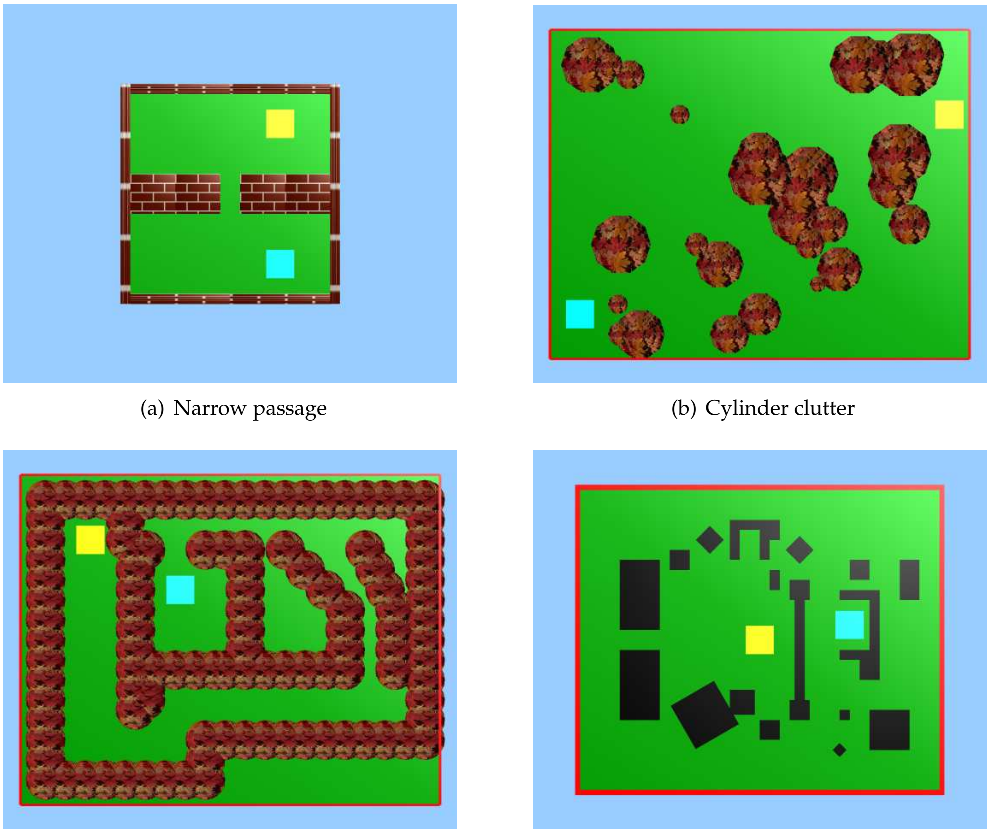
Local planning is the process of finding a (simple) path from one configuration to another configuration. The straight line local planner is often used. In this technique, the local path is a straight line in configuration space. However, in our case, experiments show that often, while a solution does exist, no solution is found using this planner. Furthermore, the resulting motion, when found, can lead to unwanted behavior for the box, like excessive rotations and deformations. We propose two local planners to circumvent this problem. The first local planner is an adjustment of the straight line planner, while the second uses a via-point and hinging of the box.

- **Deformable box** The deformable box local planner is an adapted straight line planner. Instead of taking the direct straight line between the start and goal configurations, this local planner has the choice between two paths. The first path is the straight line, the second is the path where the box rotates $\frac{\pi}{2} - \alpha$ (where α is the rotation angle of the straight line planner) in the opposite direction and deforms to reach the correct goal configuration (Figure 2(c)). The local planner chooses the shortest of the two paths. This is dependent on the metric used. By using different metrics, the characteristics of the generated local paths is changed. The metric could favor rotations over deformations or visa-versa.

- **Hinged box** The second local planner, called the hinged box, calculates a via-point, which is the intersection point of the two lines through the centers of the start and goal configuration in the directions of the longest edge. The box then moves over the two line segments (start to via, via to goal). The box is hinged at the via point. Figure 2(d) shows a typical path generated by this local planner, where an intermediate configuration is depicted at the via point. Care must be taken to guarantee that the area of the hinged box stays the same during the motion.

4 Experiments and Results

We will analyze the effects of the different sampling techniques and local planners. The aim is to see if our technique is able to find paths, and if so, what the running times are for building the roadmap.



(a) Narrow passage (b) Cylinder clutter

(c) Tree path (d) City scape

Figure 3: Scenes used in the experiments. The start and goal area are depicted as light-gray squares.

All techniques were implemented using Microsoft Visual C++ using the Solid collision detection package [17]. All experiments were run on a Pentium IV 2.40 Ghz with 1 GB internal memory. The reported durations are all in seconds, and averaged over 200 runs.

We have selected four scenes for which we build a roadmap. We measure the time it takes to build this roadmap. The scenes are all specified in 3D, and the motion planning takes place in a plane in the scene. The scenes used were as follows:

- **Narrow passage** This scene is chosen to see if the technique functions well in scenes with a narrow passage. There is only one path from one side to the other (Figure 3(a)). It is expected that the medial axis approaches outperform the other techniques, while uniform sampling will be the worst technique. In this scene, the hinged local planner is expected to perform better than the deformable box local planner.

- **Cylinder clutter** The purpose of this scene is to show that the techniques work for round obstacles. The scene is rather open, and allows for multiple path around the cylinders to go to the same location (Figure 3(b)). Due to the large open space, the uniform sampling approach is expected to achieve the same performance as the other sampling techniques.

- **Tree path** Virtual environments often contain parts of scenes where there is only one path which the group can follow, although the entities can individually take shortcuts. This scene contains a collection of trees (Figure 3(c)). These trees are situated such, that when, for instance, flocking is used, the entities can move in between the trees. However, coherent

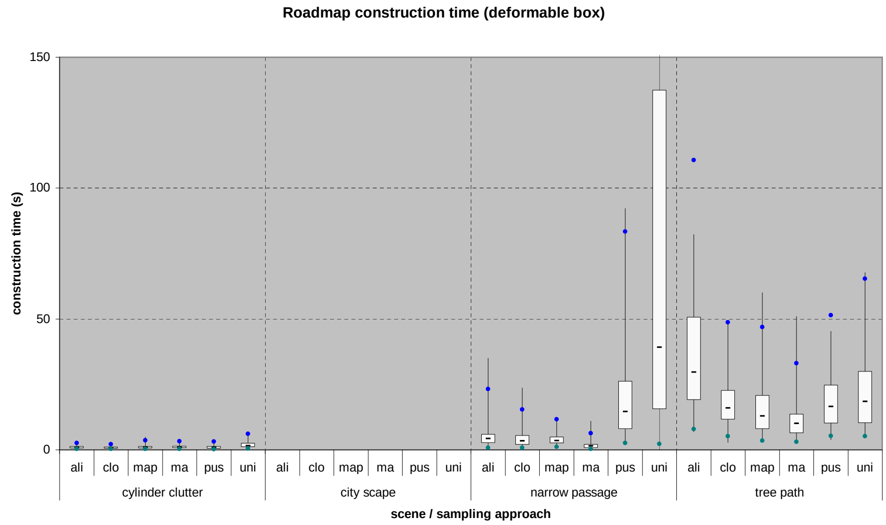


Figure 4: A box plot of the construction times for the deformable box local planner.

group movement should restrict this, and allow the group to go the only path available around the trees.

- **City scape** Many games and virtual environments contain some city elements or alike. Our approach should work in these types of scenes. Therefore, we modeled a small part of a city in this scene (Figure 3(d)). In this scene, the hinged local planner is expected to perform well, due to the sharp turns and narrow passages that need to be taken. Again, standard flocking could result in entities taking different paths in between the buildings.

4.1 Compare sampling techniques

To compare the different sampling methods, we ran experiments for every scene and technique. The local planner used is the deformable box. The results can be found in Table 1. A boxplot of the roadmap construction times is depicted in Figure 4. The extreme values of the 95% confidence interval are also marked with points.

As expected, the uniform sampling technique performs poorly in every situation. The variance of the uniform sampling is larger than that of the other sampling strategies, especially in the narrow passage scene. In the city scape scene, using the deformable box local planner, no sampling strategy is able to find a path within 10 minutes of running time. Less than 5 of the total of 1200 runs resulted in a solution, far too low for any statistics.

Narrow passages are best solved by the medial axis sampling technique. The reason that the medial axis using the penetration depth does not perform better is that the samples generated

	UNI	MA	MAP	PUS	ALI	CLO
Cylinder clutter	1.1	0.8	0.7	**0.5**	0.7	0.6
City scape	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Narrow passage	15.6	**0.8**	2.6	8.0	2.7	2.1
Tree path	10.3	**6.5**	8.0	10.1	19.1	11.6

Table 1: Results from the comparison experiment of sampling techniques. Running time for the completion of the roadmap building. Times are reported in seconds to complete.

Figure 5: A box plot of the construction times for the hinged box local planner.

are not situated close to the entrance of the narrow passage, as opposed to the general medial axis technique. These samples are helpful in finding the narrow passage when the deformable box local planner is used.

Overall, no sampling technique is best in all situations. When using the deformable box local planner, the uniform sampling strategy should be avoided. Overall, the sampling strategy expected to perform best in general situations is the medial axis strategy. This sampling strategy is able to find the solution reasonably fast in these scenes (except the city scape).

4.2 Compare local planner techniques

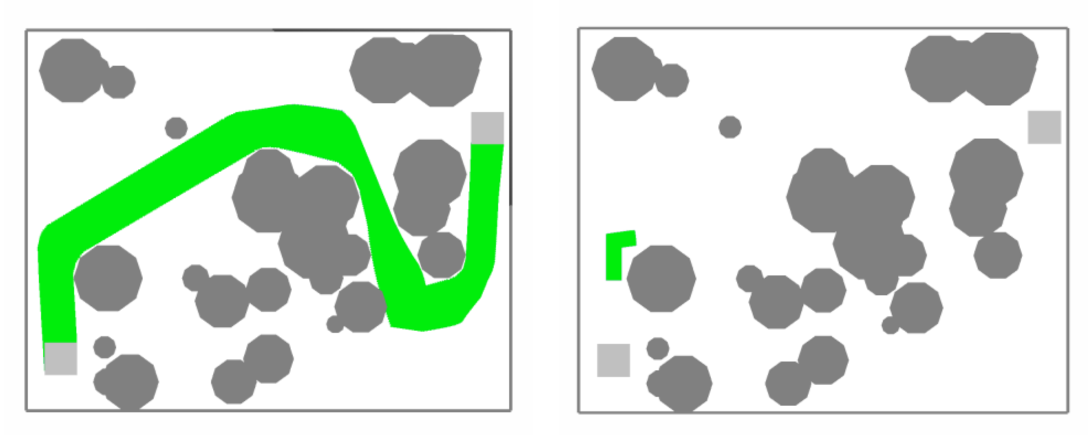
To compare the different local planner methods, we ran the same experiments with the hinged local planner. The results can be found in Table 2. A boxplot of the construction times is given in Figure 5. The extreme values of the 95% confidence interval are also marked with points.

First thing to notice is that all sampling method are able to solve the city scape problem when the hinged box local planner is used. Furthermore, the construction times are rather low. This can be attributed to the local planner being able to find paths around corners and through narrow passages. This is also noticeable in the construction times for the narrow passage scene. In the narrow passage scene, the hinged box local planner reduces the construction times for all sampling techniques when compared to the deformable box local planner.

It is clear from the boxplot that the uniform sampling become very competitive when the hinged box local planner is used. Both the construction time and the variance is low. Remarkable is that both medial axis strategy perform poorly in the tree path scene. A possible reason for this

	UNI	MA	MAP	PUS	ALI	CLO
Cylinder clutter	**0.5**	1.7	1.6	1.2	1.0	0.9
City scape	3.4	2.3	2.6	4.8	2.2	**1.4**
Narrow passage	0.9	0.3	**0.1**	2.4	0.3	0.4
Tree path	**9.4**	11.7	13.4	23.0	15.2	10.8

Table 2: Results from the comparison experiment of sampling techniques. Running time for the completion of the roadmap building. Times are reported in seconds to complete.



(a) An example of a sweep path of a possible motion of the deformable shape, where the deformable box local planner is used.

(b) An example of a intermediate configuration on a path using the hinged box local planner.

Figure 6: Examples of possible results of the two local planners

is that the samples generated by the strategies hinder the local planner in finding a path quickly around corners, as is needed in this scene.

Overall, the best performance is achieved by the hinged box local planner. An additional advantage of this local planner is that the relatively simple uniform sampling technique can be used, making it easier to implement.

To give an indication of the possible results from the techniques, Figure 6 depicts to examples. The first example is the sweep path of the shape where the deformable box is used as a local planner (Figure 6(a)). The second is an intermediate configuration along the path planned with the hinged local planner (Figure 6(b)).

5 Planning the Internal Group Motion

In the previous section, we outlined how to compute the global motion for the group using a deformable rounded rectangle. In this section, we will describe our approach to keep the entities inside the shape and to generate the local motions. For now we assume that the deformable shape stays at the same position and only rotates and deforms. Later, we will combine the motion of the entities with the translational motion of the shape (see Section 2.3).

The approach given in this section does not depend on the fact that we use a rounded rectangle. With only minor modifications it can be extended to a multifariousness of bounding shapes.

Since there are no obstacles inside the shape, we can use a potential field method without the risk of running into local minima. We will use a new technique we call *group potential fields*, which is related to the technique known as social potential fields [11], but takes the bounding shape into account. Our technique uses potential force field functions between the entities of the group, as well as having a force field that repulses the entities away from the boundary (Figure 7).

5.1 Group potential fields

In the group potential field model, every entity is viewed as a point particle (numbered from 1 to n) with the positions $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n$ at a certain time. We model the entities as discs. These discs form a required clearance cl_i around each point particle i. We can define the standard Euclidean distance between two entities as $r_{ij} = \|\vec{x}_i - \vec{x}_j\|$. The force between two entities is defined as

$$\vec{F_{ij}}(\vec{x}_i, \vec{x}_j) = \left(-\frac{c_{1,ij}}{(z_{ij})^{\sigma_{1,ij}}} + \frac{c_{2,ij}}{(z_{ij})^{\sigma_{2,ij}}}\right)\left(\frac{\vec{x}_i - \vec{x}_j}{r_{ij}}\right) \tag{1}$$

where $z_{ij} = (r_{ij} - (cl_i + cl_j))$ and the constants $c_{1,ij}, c_{2,ij}, \sigma_{1,ij}, \sigma_{2,ij}$ can be chosen, depending on the desired behavior. Since the repulsive force should always act on a closer range than the attractive force, the following should always hold: $c_{1,ij} < c_{2,ij}$, $\sigma_{1,ij} < \sigma_{2,ij}$, $c_{1,ij} > 0$ and $\sigma_{1,ij} > 1$. For symmetry, we take $c_{ij} = c_{ji}$ and $\sigma_{ij} = \sigma_{ji}$. Actually, in our implementations we take the constants independent of i and j, so all entities are treated in the same way. However, this can be changed if different entities play a different role.

In addition to the entity-entity forces, the entities also experience a force from the border of the shape. This force is dependent on the distance to the border as follows:

$$\vec{F_{b,i}}(\vec{x}_i, \mathbf{B}) = \left(\frac{c_{3,i}}{d_{\perp}^{\sigma_{3,i}}(\vec{x}_i)}\right)\vec{n}_{b_i} \tag{2}$$

where $d_{\perp}(\vec{x}_i)$ is the perpendicular distance of the entity to the border, $c_{3,i}, \sigma_{3,i}$ are arbitrary coefficients such that $c_{3,i} > 0$ and $\sigma_{3,i} > 1$ and $\vec{n}_{b_i}$ is the normal of unit length pointing from the border toward $\vec{x}_i$.

The total force on an entity i becomes

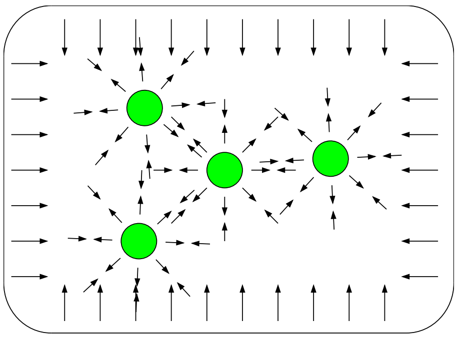
$$\vec{F_i} = \vec{F_{b,i}} + \sum_{i \neq j} \vec{F_{ij}} \tag{3}$$

Given the force function on the entities, we can use the laws of physics to calculate the positions, velocities and accelerations. From Newtons Second Law of Physics, we have $F = ma$. Hence, from the forces on the entities we can determine their acceleration. Using any standard integration scheme like Runge Kutta [19], we can determine the velocities and positions in every time step. This results in individual paths for the entities inside the deformable shape.

5.2 Temporal distortion

During the deformation and rotation of the shape, the entities move around inside the shape. Therefore, the entities will eventually come close to the border and be pushed away. Rapid deformations and rotations will result in large accelerations of the entities, exceeding the maximum allowed velocities and acceleration, resulting in entities crossing the border of the shape or in unnatural motions. Clearly, we want to limit the speed of deformation in such a way that the entities move naturally and stay inside the shape. So, given the deformations of the group shape, we want to determine a speed function such that, when deforming with that speed, the entities

Figure 7: Group potential fields, such that the entity-entity force is chosen such that the force is repulsive if the entities are close together, and attractive if they are far apart.



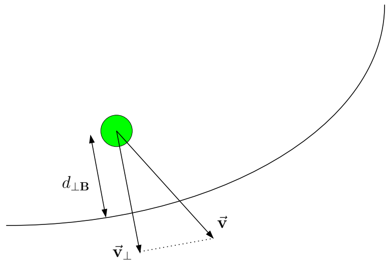


Figure 8: The construction of definition 5.1. The velocity vector is decomposed in a direction perpendicular to the border and one parallel to the border. The perpendicular direction is decreased by the repulsive force of the border. If this is not possible the entity is too close to the border.

---

will stay inside the shape. Formally we define this problem as follows:

Problem 5.1 (Temporal distortion function)**.** *Given a deformation function* $\Pi : [0, 1] \to R^n$ *of the shape, find a* **temporal distortion function** $T : [0, t] \to [0, 1]$ *(where t is the duration of the motion), such that all entities will be able to stay inside the shape, for the path* $\Pi' = \Pi \circ T : [0, t] \to R^n$ *of the deformable shape.*

To solve problem 5.1, every time step we check the maximum time-derivative (slope) of the temporal function. This is done as described in Algorithm 1.

Algorithm 1 The algorithm to determine whether an entity is close to the border.

```
while s < 1 do
    if no entity close to border then
        use the identity function (e.g. ∂T/∂s = 1) for the temporal function
    else
        use the zero function (e.g. ∂T/∂s = ∞) for the temporal function
    end if
end while
```

When entities get close to the border, we temporarily stop the deformation until the repulsive border potential has pushed them away again. To be able to do this, we need to define what near to the border means. The following definition specifies a measure for closeness to the border.

Definition 5.1 (Near the border)**.** *An entity is near the border if the following equation holds:*

$$v_{i,\perp} < \sqrt{2\, d_{\perp}(\vec{x}_i)\, a_{max}} \tag{4}$$

where $v_{\perp}$ *is the perpendicular velocity of the entity with respect to the border,* $d_{\perp}(\vec{x}_i)$ *is the perpendicular distance of the entity to the border and* a_{max} *is the maximum acceleration of the entity.*

This definition stems from the fact that when an entity is moving toward the border of the shape it is decelerated in the perpendicular direction. This deceleration should be large enough, such that the entity could reach complete stand still (in the perpendicular direction) before the border of the shape is reached, and the entity is accelerated inward again (see Figure 8).

Clearly, the larger we make the volume of the shape, the smaller the chance that entities get near to the boundary and the faster we can deform the shape. But, with a larger shape it will be more difficult if not impossible to find a path. So we carefully need to balance these aspects.

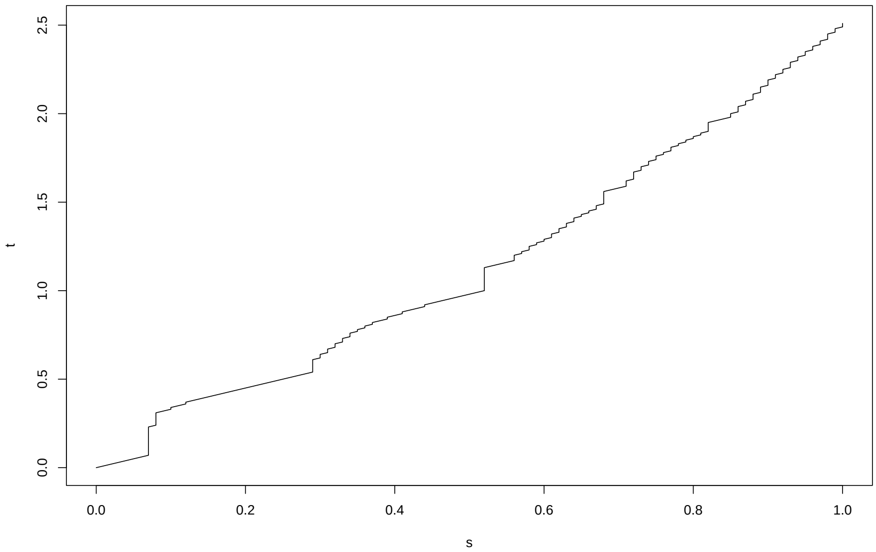


Figure 9: An example of the temporal distortion function resulting from a typical simulation run.

6 Experiments and Results

The same hard- and software is used as in section 4. We ran a number of simulations of the group potential field technique. From these simulations a typical example of the resulting temporal distortion function is given in Figure 9. The vertical segments correspond to the situations where an entity is close to the boundary.

We also combined the global motion for the group shape with the internal motion from the group potential field. In all situation the technique was able to find a path for all entities. Figure 10 shows the path of a group of sheep in the cylinder clutter scene. The left sequence of images is the result from our approach. The right sequence of images is the result from a simulation where flocking is used. Clearly visible is the coherent group behavior in our approach, and the lack of coherence in the flocking approach.

Local path generation and combining the local and global paths are done in real-time.

7 Conclusion and future work

From the results, it can be concluded that the technique is able to plan the paths of a large number of entities. The technique enforces group coherence, which is a highly wanted behavior.

In our experiments with the PRM technique, the chosen sampling method is not of high importance, while a good choice of local planning is paramount. Clearly, the hinged box local planner is the planner of choice. It is able to solve all problems, in relatively short construction times, while also enabling the use of the uniform random sampling technique.

One of the main advances of our approach is that the technique guarantees coherence. However, this is achieved at a cost. The approach is not complete anymore. There are situations where there is a path for the group, but it is not found by our approach due to the choice of group shape. Future research into different group shapes is needed to make the approach complete.

The approach proposed in this paper enables the planning of different sized groups without the need to replan the global path (of the deformable shape), as long as the shape is of sufficient

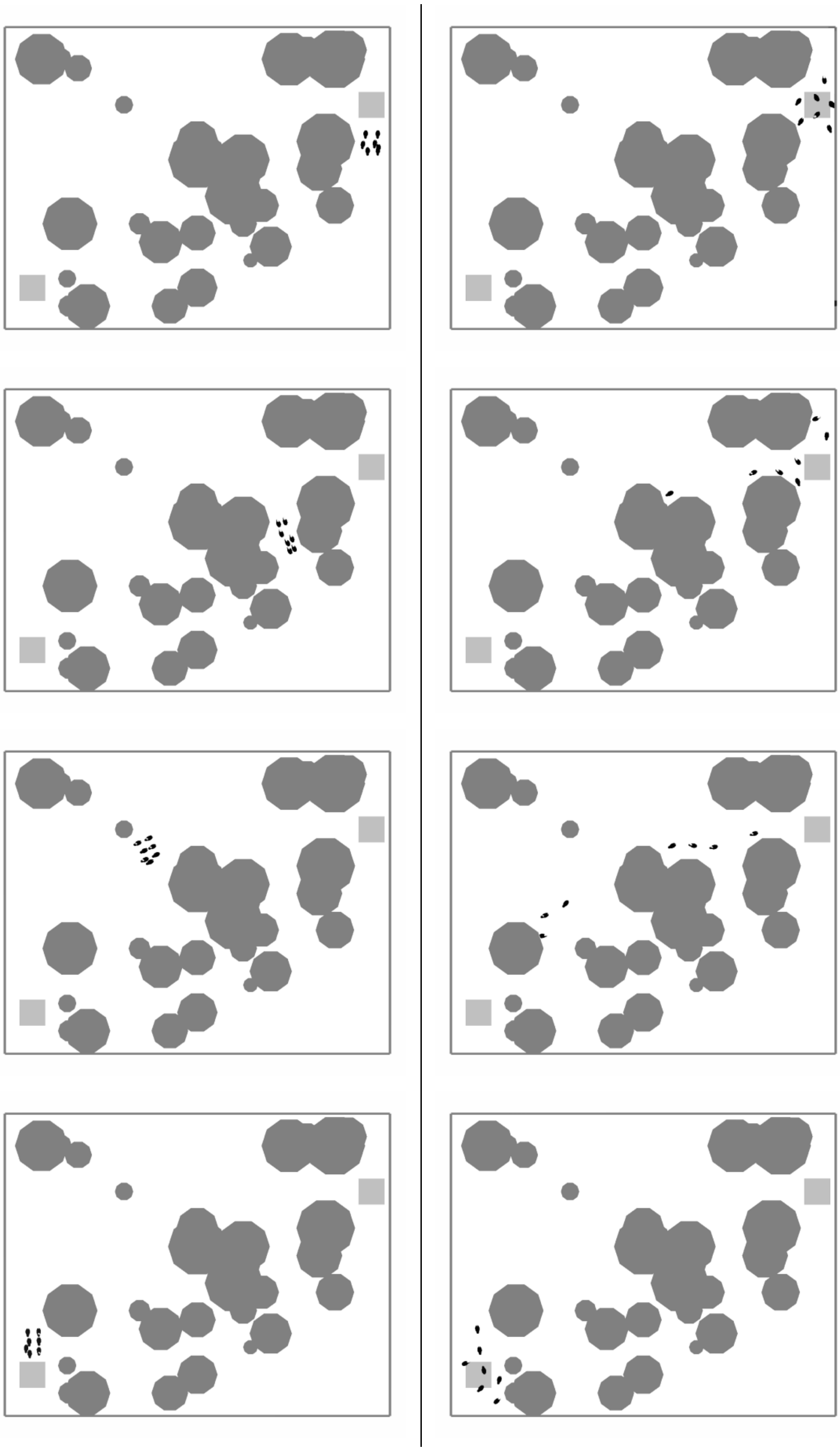


Figure 10: Resulting motion from both our approach (left, top to bottom) and flocking (right, top to bottom). Our approach leads to coherence, while flocking lacks this entirely (Textures have been omitted for clarity reasons).

volume. The complexity of the approach is also rather insensitive to the group size itself, since most of the planning is done independent of it.

Acknowledgment

The authors would like to thank Dennis Nieuwenhuisen for developing the Callisto collision and visualization toolkit. The authors also like to thank Judith Helgers for contributing in the research and development of this approach.

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] 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.

[2] Mark de Berg, Matthew J. Katz, A. Frank van der Stappen, and Jules Vleugels. Realistic input models for geometric algorithms. In *Symposium on Computational Geometry*, pages 294–303, 1997.

[3] D.R. Forsyth. *Group Dynamics*. Wadsworth Publishing Company, Belmont, CA 94002, USA, 3rd edition, 1998.

[4] R. Geraerts and M. Overmars. A comparative study of probabilistic roadmap planners. In *Algorithmic Foundations of Robotics V, Springer Tracts in Advanced Robotics 7*, pages 43–57. Springer-Verlag Berlin Heidelberg, 2004.

[5] L. Guibas, C. Holleman, and L. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *In Proc. IEEE Int. Conf. on Intelligent Robots*, 1999.

[6] C. Holleman and L. Kavraki. A framework for using the workspace medial axis in prm planners. In *International Conference on Robotics and Automation*, volume 2, pages 1408–1413, 2000.

[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] 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.

[12] C.W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[13] C.W. Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.

[14] M.E. Roloff. *Interpersonal Communication - The Social Exchange Approach.* SAGE Publications, v.6, London., 1981.

[15] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1994.

[16] 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.

[17] G. van den Bergen. *Collision Detection in Interactive 3D Environments*. Morgan Kaufmann, 2003.

[18] P. Švestka and M.H. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23:125–152, 1998.

[19] Eric W. Weisstein. Runge-kutta method.