

Research Article

A Grid-Based Motion Planning Approach for Coherent Groups

Yue-wen Fu,¹ Meng Li,² Jia-hong Liang,¹ and Xiao-qian Hu¹

¹College of Information System and Management, National University of Defense Technology, Changsha, Hunan 410073, China

²Academy of PLA, Hefei 230031, China

Correspondence should be addressed to Meng Li; claymore@nudt.edu.cn

Received 19 October 2014; Accepted 11 December 2014; Published 29 December 2014

Academic Editor: Roque J. Salterén

Copyright © 2014 Yue-wen Fu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a novel motion planning approach for coherent groups with constant area, and it integrates C-L method into the probabilistic roadmap algorithm with sampling on the medial axis (MAPRM). In the preprocessing phase, the group is discretized into a grid-set which represents the configuration of the group. Then, a number of samples are generated on workspace by medial axis technique. These samples are extended into group's configuration nodes of the roadmap using an extending strategy. Also, the group's deformation degree relative to the desired shape is introduced to improve the evaluation function. It gives users more flexibility to determine the respective weights of the group's deformation degree and its distance to the goal in the query phase. After that, a novel local planner is constructed to connect any two neighbor configurations by using C-L method and the improved evaluation function. Experiments show that our approach is able to find paths for the coherent group efficiently and keep its area invariant when moving toward the goal.

1. Introduction

Human crowd is a fascinating social phenomenon which has been constantly investigated by experts from various areas. Crowd simulation can be considered as a multiagent system [1, 2]. And it is enjoying considerable success in numerous applied domains, most notably in evacuation scenarios where simulated crowd behaviors can help to improve the safety of interior building designs [3, 4]. In crowd simulation, groups are the most important entities that are worth intensive studies. Groups can be defined as semipermanent collections of individuals sharing navigation goals who attempt to maintain spatial cohesion with each other. But these groups may split up when moving toward the goal. By contrast, coherent groups are a class of groups which maintain permanent gathering and would not split up [5].

At present, coherent groups have gained tremendous momentum in military and safety training applications. This type of coherent groups can be seen as deformable objects with area conservation. And investigating the motion planning problem for these deformable objects has an important meaning in computer animation, crowd simulation, and group behavior generating. For example, we need to simulate

the behavior of a squad of soldiers which have a desired formation with a constant area on the whole and constant space between any two soldiers when marching toward the goal. The formation of the soldier group can reflect their tactical strategy that can provide different functions for distinct physical effects, which could be a key factor in defeating the enemy in a battle. In addition, it can be used to simulate different motions of the wild congregated animals in computer animations to create special visual effects, for example, a crowd of migratory geese that herd to the south in perfect V formation. However, most current motion planning approaches cannot control the area of the group as a constant easily, and how to plan paths for these deformable objects with area conservation is still a challenging problem in these fields.

As one of the motion planning methods, C-L method proposed by Chang and Li [6] provides an excellent approach to planning the path of deformable objects and maintaining its size conservation when moving toward the goal. Unfortunately, the C-L method is computationally expensive because it did not use the environment's information efficiently. Fortunately, a probabilistic roadmap planner using the medial axis technique (MAPRM) which generates samples on the

medial axis of the workspace has an additional benefit in this situation to fully take advantage of the environment's information [7, 8].

Consequently, in this paper, a novel approach will be described which is used to plan a path to connect the initial and goal configuration of the coherent group by combining MAPRM algorithm and C-L method for size conservation. Rather than planning the motion for individual entities, the goal of this paper is planning the motion for the group as a whole, in which the group is modeled as a deformable grid-set with a constant area and planned for this particular grid-set.

This paper is organized as follows. The previous works related to group motion planning are introduced in Section 2. And the framework of our approach, including the flowchart of the local planner, is proposed in Section 3. In Section 4, we first describe the sampling mechanism on medial axis briefly and discretization approach for coherent group. Then we describe how to generate configuration nodes of the roadmap by extending the position samples on medial axis. Section 5 shows the detailed designing process of the local planner. Experiments and results are given in Section 6. Finally, we draw some conclusions and outline future work in Section 7.

2. Related Work

In the virtual environment the most common approach to simulating group movement is to use flocking which was introduced by Reynolds [9]. Bayazit et al. [10] combined flocking with the probabilistic roadmap algorithm (PRM) to guide the group's motion toward the goal. In their works, the roadmap provided a convenient abstract representation of global information in complex environments and associated rules with roadmap nodes which enables local customization of behaviors. Li and Chou [11] proposed a novel centralized planning approach that moves the robots in groups formed dynamically with a sphere-tree structure. Li et al. [12] combined basic flocking algorithm with leader-follower model to simulate group's motion. At the same time, in order to plan the path of the group, they tried to model the shape of the group's region by estimating the longitudinal depth of the group. However the above methods cannot ensure the group's coherence when moving toward the goal and do not consider the groups as deformable objects.

Guibas et al. [13] proposed a probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach (MAPRM). In their method, the medial axis of the workspace was computed at the preprocessing stage and the configurations of the flexible objects were fitted at random points along the medial axis. However, this method is only suitable for planning the path for the objects like sheets of metal or plastic flexible pipes which only can be bent. But the coherent group can deform to arbitrary shapes.

Overmars et al. [14–16] introduced a new approach to motion planning for coherent groups of entities. In their methods, the group was represented as a deformable shape of sufficient volume, such as deformable rounded rectangle or hinged rectangle. Then, an extension of the probabilistic roadmap method was used to plan the motion for this

deformable shape. In Kamphuis's method, the designers must take care to guarantee that the area of the deformable shape remained the same during its motion. However, the area of the group cannot be controlled as a constant easily in their method and the shape of the group cannot be simplified into rectangle in the complex environment.

Because of the defects of the above methods, Chang and Li [6] proposed a novel method to investigate the motion planning problem for deformable objects with size conservation which is named as C-L method in this paper. In C-L method, the deformable object was firstly discretized into a series of small grids. Together, these grids formed a grid-set that represented the configuration of the deformable object. Then, C-L method planned paths for the whole grid-set to generate the paths of the deformable objects. So, these grids became the smallest units to be operated by C-L method in which a new configuration was generated according to the current information of the environment and the configuration of the grid-set. At the same time, the deformation degree of the object which represented the difference between the current configuration of the object and the desired configuration was introduced to improve the evaluation function of the searching algorithm. Later, Chang and Li [17, 18] made the crowd conform to a specific shape while avoiding collisions with other agents in the crowd or with obstacles by combining C-L method with fuzzy controller. The C-L method was used to generate the global motion of the group, while the fuzzy controller was used to move the agents in the group to conform to a desired shape.

Chen et al. presented a modelling and simulation of extra-large-scale crowd evacuation using massively parallel computing technology [19]. Each individual in the scenario is modeled as an adaptable and autonomous agent driven by a weight-based decision-making mechanism. The simulation is intended to characterize the individuals' adaptable behaviors. Their approach adopts GPGPU to successfully sustain massively parallel modeling and simulation of a huge evacuation scenario consisting of hundreds of thousands of individuals. Their approaches for handling big scientific data with GPGPU are also applicable for dynamic data driven applications (DDAS) of online simulation studies [20–22].

Planning paths for deformable objects or flexible objects has gained various applications in industrial automation, computer generated animation, and virtual environments where the deformable properties of objects need to be considered for the creation of realistic motion. In this paper, we restrict ourselves to motion planning approaches to compute paths for a special flexible object which is the coherent group with area conservation. The group's contour changes in different environments; however the group's area must keep constant when moving along the planned path.

3. Planning Framework

In general, this study was inspired by the discretization technique in C-L method [17, 18], medial axis technique [7], and Kamphuis and Overmars' work [14, 16]. The goal of our approach is twofold. On one hand we should make the group

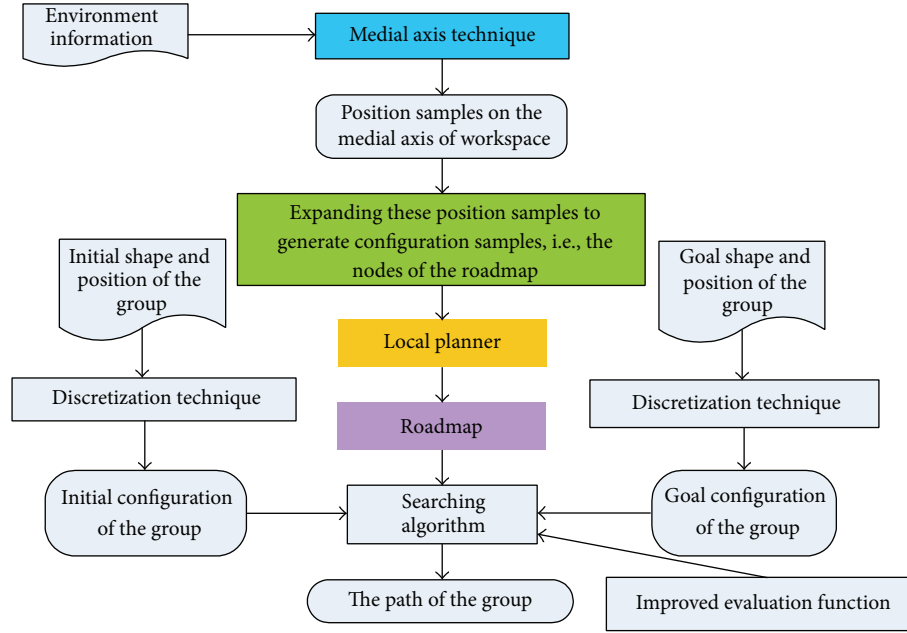


FIGURE 1: Flowchart of our approach.

move toward the goal with optimal evaluation function; on the other hand the group can be changed into any shape according to the environment and the group's area must be kept constant in this process. In this paper, the evaluation function is the weighted sum of the distance to the goal and the difference between the group's configuration and the desired configuration. Optimal evaluation function means that the value of evaluation function is the smallest at every run step. Our approach consists of the following steps.

- (1) Specify the initial, goal, and desired shape of the group and discretize these shapes into grid-sets which, respectively, represent the group's initial, goal, and desired configurations.
- (2) Generate a series of position samples on the medial axis of the workspace and extend these position samples into group's configuration samples which will be seen as the nodes of the roadmap.
- (3) Design the local planner based on C-L method and the improved evaluation function. Then, construct a roadmap of possible motions for the deformable configuration of the group between any two neighbor nodes using this local planner.
- (4) To answer a motion query, use the roadmap to compute a collision-free path for the deformable configuration.

The flowchart of our approach is illustrated in Figure 1 and the detailed process of the local planner is illustrated in Figure 2 in which we combine C-L method and improved evaluation function. Our local planner has two advantages: one is that the configuration nodes extended from position samples can provide guidance for C-L method and the other is that C-L method can ensure the constant of the group's area.

4. Generate Configuration Nodes

4.1. Medial Axis Sampling Technique and Discretization of the Group. Firstly, the medial axis technique for sampling strives to generate samples on the medial axis of the workspace [17, 18]. This approach has been used previously in motion planning for deformable objects [13]. The technique is as follows: generate a uniformly random point in workspace (or 2D configuration space in [18]). According to whether this point lies in free space or not, compute different retraction directions and then retract this point to medial axis. Repeat this process until a series of position samples on medial axis of the workspace are generated. These position samples are represented as $(x_{w,i}, y_{w,i})$ $i = 1, \dots, N$ where N represents the total number of the position samples. The fact that medial axis technique provides the largest clearance from obstacles for the group is advantageous. Secondly, the group's shape is discretized into a grid-set, denoted by $C_{\text{set}} = \{G_1(x_1, y_1, \theta_1), G_2(x_2, y_2, \theta_2), \dots, G_M(x_M, y_M, \theta_M)\}$, in which $G_i(x_i, y_i, \theta_i)$ $i = 1, \dots, M$ represents the information of the i th grid and M represents the total number of the grids. Obviously, C_{set} is the configuration of the coherent group. In the grid-set, there are two types of grids: inner grids and boundary grids. The inner grids are the ones whose neighbor grids are also the grids of C_{set} , while this is not true for the boundary grids. Moreover, we suppose that the group is compact; that is, no grid is isolated and no hole exists in C_{set} and the directions of all the grids are the same. An isolated grid means that there is no common border or vertex between it and any other ones. Figure 3 graphically illustrates the compact group versus incompact ones. Obviously, the compact property ensures the group's coherence.

4.2. The Extending Strategy of Generating Configuration Nodes. The goal of this section is to extend the 2D position

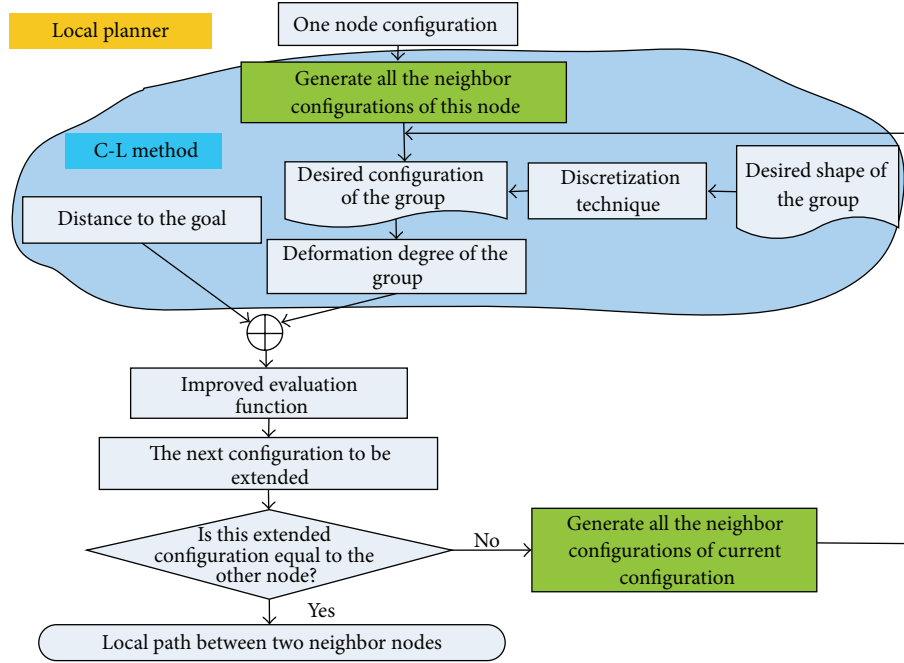


FIGURE 2: Flowchart of the local planner.

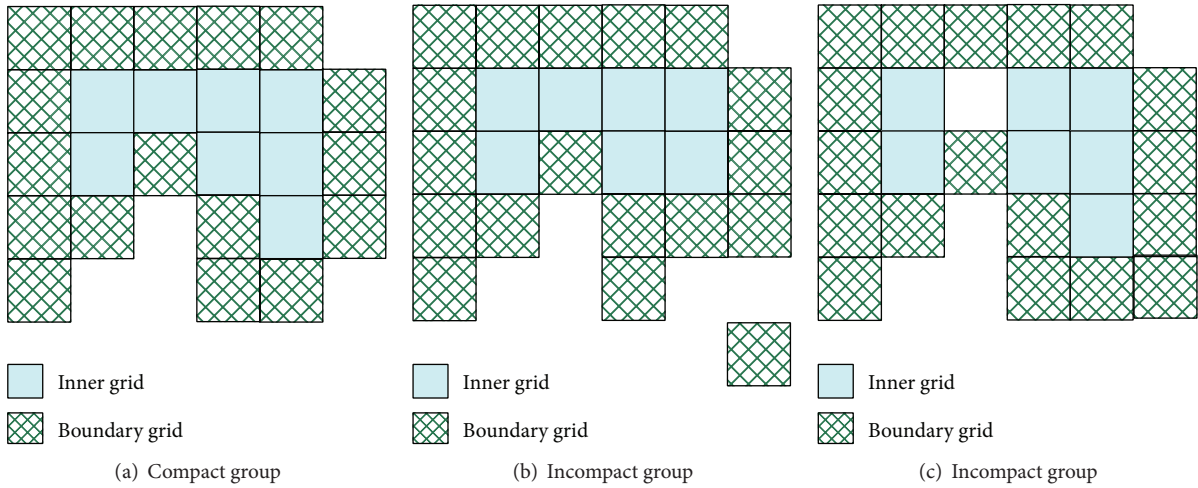


FIGURE 3: Compact group versus incompact ones.

samples into group's configuration samples which are M dimensions. The extending strategy is divided into two steps.

Step 1. Find the center grid of C_{set} and then place it on the position samples. The center grid of C_{set} is represented as $G_c(x_c, y_c, \theta_c)$ whose information can be computed by

$$\begin{aligned}
 & (x_c, y_c, \theta_c) \\
 & = \arg \min_{(x_j, y_j, \theta_j) \in G_{\text{set}}} \left(\sqrt{\left(\frac{\sum_{i=1}^M x_i}{M} - x_j \right)^2 + \left(\frac{\sum_{i=1}^M y_i}{M} - y_j \right)^2} \right). \quad (1)
 \end{aligned}$$

So, at every position sample, we make $x_c = x_{w,i}$, $y_c = y_{w,i}$, and θ_c equal to the tangent direction of the medial axis at $(x_{w,i}, y_{w,i})$.

Step 2. After placing the center grid on $(x_{w,i}, y_{w,i})$, we should decide the positions of the other $M - 1$ grids according to the environment around $(x_{w,i}, y_{w,i})$, the desired configuration C_P , and the compact property of the group. In this process, the desired configuration is regarded as a template to provide guidance for generating configuration samples.

At first, we place a duplicate \bar{C}_P of C_P on $(x_{w,i}, y_{w,i})$ by making its center grid coincide with $(x_{w,i}, y_{w,i})$ and its direction aligns with the tangent direction of medial axis;

Input. The desired Configuration C_p , Obstacle space C_{obs} , $(x_{w,i}, y_{w,i}) \quad i \in \{1, \dots, N\}$

Output. Configuration_Samples

$C_i^{sampling} = \{G_1^{sampling}(x_1, y_1, \theta_1), G_2^{sampling}(x_2, y_2, \theta_2), \dots, G_M^{sampling}(x_M, y_M, \theta_M)\} \quad i \in \{1, \dots, N\}$

/ Step 1–9 are used to place a copy of C_p on $(x_{w,i}, y_{w,i})$ and move the grids which collide with C_{obs} to the free space */*

- (1) Clone \tilde{C}_p from C_p and ensure that the center grid of \tilde{C}_p coincides with $(x_{w,i}, y_{w,i})$ and the direction of \tilde{C}_p is aligned with the tangent direction of medial axis at $(x_{w,i}, y_{w,i})$
- (2) **for** $j = 1, \dots, M$
- (3) **if** $\tilde{G}_j(x_j, y_j, \theta_j)$ collides with C_{obs}
- (4) find the boundary grid of \tilde{C}_p which is represented as $\tilde{G}_{j,boundary}^{nearest}$ nearest to $(x_{w,i}, y_{w,i})$ and collision-free with C_{obs}
- (5) find the neighbor grid of $\tilde{G}_{j,boundary}^{nearest}$ which is represented as $\tilde{G}_{j,boundary}^{nearest_neighbor}$ nearest to $(x_{w,i}, y_{w,i})$ and collision-free with C_{obs}
- (6) move $\tilde{G}_j(x_j, y_j, \theta_j)$ to $\tilde{G}_{j,boundary}^{nearest_neighbor}$
- (7) **end if**
- (8) **update** \tilde{C}_p
- (9) **end for**

/ Step 10–22 are used to iteratively adjust the center of \tilde{C}_p to $(x_{w,i}, y_{w,i})$ */*

- (10) compute the center of \tilde{C}_p which is represented as $\tilde{G}_c(x_c, y_c, \theta_c)$
- (11) **if** the distance between $\tilde{G}_c(x_c, y_c, \theta_c)$ and $(x_{w,i}, y_{w,i})$ is bigger than a small positive value
- (12) **if** $\{\tilde{G}_1(x_1, y_1, \theta_1) + \nu, \tilde{G}_2(x_2, y_2, \theta_2) + \nu, \dots, \tilde{G}_M(x_M, y_M, \theta_M) + \nu\}$ don't collide with C_{obs}
- (13) move $\{\tilde{G}_1(x_1, y_1, \theta_1), \tilde{G}_2(x_2, y_2, \theta_2), \dots, \tilde{G}_M(x_M, y_M, \theta_M)\}$ to $\{\tilde{G}_1(x_1, y_1, \theta_1) + \nu, \tilde{G}_2(x_2, y_2, \theta_2) + \nu, \dots, \tilde{G}_M(x_M, y_M, \theta_M) + \nu\}$ in which ν is the vector from $\tilde{G}_c(x_c, y_c, \theta_c)$ to $(x_{w,i}, y_{w,i})$
- (14) **return** $C_i^{sampling} = \{\tilde{G}_1(x_1, y_1, \theta_1) + \nu, \tilde{G}_2(x_2, y_2, \theta_2) + \nu, \dots, \tilde{G}_M(x_M, y_M, \theta_M) + \nu\}$
- (15) **else** find the farthest boundary grid from $\tilde{G}_c(x_c, y_c, \theta_c)$ which is denoted as $\tilde{G}_{farthest}^\alpha(x, y, \theta)$ satisfying the condition that $(\tilde{G}_{farthest}^\alpha(x, y, \theta) - \tilde{G}_c(x_c, y_c, \theta_c)) \cdot \nu < 0$
- (16) find the farthest boundary grid from $\tilde{G}_c(x_c, y_c, \theta_c)$ which is denoted as $\tilde{G}_{farthest}^\beta(x, y, \theta)$ satisfying the condition that $(\tilde{G}_{farthest}^\beta(x, y, \theta) - \tilde{G}_c(x_c, y_c, \theta_c)) \cdot \nu > 0$
- (17) find the neighbor grid of $\tilde{G}_{farthest}^\beta(x, y, \theta)$ which is farthest to $\tilde{G}_c(x_c, y_c, \theta_c)$ and is represented as $\tilde{G}_{farthest}^{\beta, neighbor}(x, y, \theta)$
- (18) move $\tilde{G}_{farthest}^\alpha(x, y, \theta)$ to $\tilde{G}_{farthest}^{\beta, neighbor}(x, y, \theta)$
- (19) **update** \tilde{C}_p
- (20) **go to** Step 10 until the distance between $\tilde{G}_c(x_c, y_c, \theta_c)$ and $(x_{w,i}, y_{w,i})$ is small enough
- (21) **end if**
- (22) **end if**

ALGORITHM 1: The extending strategy of generating configuration samples.

then we move the grids in \tilde{C}_p which have collided with the obstacles to free space according to the compact property and the distance to $(x_{w,i}, y_{w,i})$. However, as these collision grids move to free space, the center of \tilde{C}_p may change. Then we should adjust the center of \tilde{C}_p so that it coincides with $(x_{w,i}, y_{w,i})$ once again. This is done by adjusting the boundary grids and the whole position of \tilde{C}_p iteratively. Using C_p as the template to guide the generation of configuration samples, it can reduce the differences between the desired configuration and the generated configuration. The detailed algorithm for generating the configuration samples is illustrated in Algorithm 1 and Figure 4 gives graphic depictions of the results computed by Algorithm 1.

It is important to note that the medial axis is prone to having small “twigs” in the obstacle boundaries. These are of

no use during the planning stage and are discarded if their length is less than a specified threshold [13].

5. Local Planner

5.1. The Node Neighbors and Shape Differences. Based on the medial axis technique and Algorithm 1, N configuration samples $C_1^{sampling}, C_2^{sampling}, \dots, C_N^{sampling}$ have been generated which will be regarded as the nodes of the roadmap. The role of the local planner is to connect any two neighbor configuration samples and then construct the roadmap. Suppose $C_q^{sampling} = \{G_{q,1}^{sampling}(x_1, y_1, \theta_1), \dots, G_{q,M}^{sampling}(x_M, y_M, \theta_M)\}$ and $C_w^{sampling} = \{G_{w,1}^{sampling}(x_1, y_1, \theta_1), \dots, G_{w,M}^{sampling}(x_M, y_M, \theta_M)\}$ are two neighbour configuration

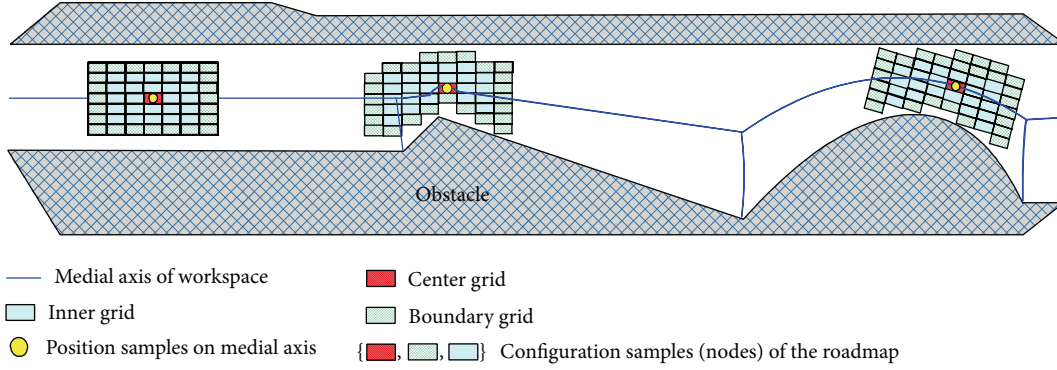


FIGURE 4: Generated configuration nodes of the roadmap.

samples. Making the judgment that C_q^{sampling} is the neighbor node of C_w^{sampling} is based on the two conditions in

$$\sqrt{(x_{c,w}^{\text{sampling}} - x_{c,q}^{\text{sampling}})^2 + (y_{c,w}^{\text{sampling}} - y_{c,q}^{\text{sampling}})^2} \leq \max_distance, \quad (2)$$

$$S_{\text{shape}}(C_q^{\text{sampling}}, C_w^{\text{sampling}}) \leq \max_deformation,$$

where $(x_{c,q}^{\text{sampling}}, y_{c,q}^{\text{sampling}})$ and $(x_{c,w}^{\text{sampling}}, y_{c,w}^{\text{sampling}})$ represent the center grid of C_q^{sampling} and C_w^{sampling} , respectively. $\max_distance$ and $\max_deformation$ are the predefined thresholds. $S_{\text{shape}}(C_q^{\text{sampling}}, C_w^{\text{sampling}})$ represents the shape difference between C_q^{sampling} and C_w^{sampling} .

The two conditions in (2) represent two criteria deciding which configuration samples we should try to connect using the local planner. One of the criteria we use is the distance of the center grid in the workspace between C_q^{sampling} and C_w^{sampling} . If they are too far apart, the chance of finding a free path between them is small. The second criterion is the shape difference between them. If they are bigger than a predefined threshold, the chance of finding a path between them is also small. $S_{\text{shape}}(C_q^{\text{sampling}}, C_w^{\text{sampling}})$ is formulated in

$$S_{\text{shape}}(C_q^{\text{sampling}}, C_w^{\text{sampling}}) = \sum_{j=1}^M r(G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j), C_w^{\text{sampling}}). \quad (3)$$

In formulation (3), $r(G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j), C_w^{\text{sampling}})$ is used to compute the minimum distance from the grid $G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j)$ to C_w^{sampling} and is formulated in

$$r(G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j), C_w^{\text{sampling}}) = \begin{cases} 0 & \text{if } G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j) \text{ is in } C_w^{\text{sampling}} \\ \min \left\{ \text{distance}(G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j), G_{w,k}^{\text{sampling}}(x_k, y_k, \theta_k)) : \right. \\ \quad \left. \forall G_{w,k}^{\text{sampling}}(x_k, y_k, \theta_k) \in C_w^{\text{sampling}} \right\} & \\ \text{if } G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j) \text{ is outside } C_w^{\text{sampling}}. & \end{cases} \quad (4)$$

Formulation (3) denotes that $S_{\text{shape}}(C_w^{\text{sampling}}, C_q^{\text{sampling}})$ is the sum of all the distances from M grids in C_q^{sampling} to C_w^{sampling} . The bigger the $S_{\text{shape}}(C_w^{\text{sampling}}, C_q^{\text{sampling}})$ is, the larger the shape difference between C_q^{sampling} and C_w^{sampling} is. Obviously, (3) can also be used to compute the deformation degree of a configuration relative to C_P .

5.2. Improved Evaluation Function. In C-L method, an improved evaluation function is proposed which is worthy of being used for reference and is formulated in

$$f(C_{\text{set}}) = \alpha \cdot h(C_{\text{set}}) + \beta \cdot S_{\text{shape}}(C_{\text{set}}, C_P). \quad (5)$$

In formulation (5), $h(C_{\text{set}})$ represents the estimated distance from C_{set} to the goal. $S_{\text{shape}}(C_{\text{set}}, C_P)$ represents the deformation degree of C_{set} relative to C_P . α and β , respectively, represent the weight of the distance to the goal and the weight of deformation degree. They also satisfy the restriction of $\alpha + \beta = 1$.

Different allocations of α and β produce diverse evaluation function values which will have a great effect on the configuration to be extended at the next run step. Therefore, α and β reflect the priorities between maintaining similarity to C_P and the distance to the goal. When we increase α , the next extended configuration will be prone to the one which has smaller distance to the goal. Conversely, the configuration which has a smaller shape difference with C_P will be preferred.

5.3. Local Planner and Roadmap Generation. After establishing the evaluation function and the neighboring relationship between C_q^{sampling} and C_w^{sampling} , the local planner can be constructed, the process of which is illustrated in Algorithm 2.

In Algorithm 2, L represents a list whose elements are all the neighbor configurations of the current ones C_Q updated by **Dequeue**(Q), that is, the first element in Q . Moreover, the neighbor configuration of C_Q is generated by moving its boundary grids on the basis of the compact property or rotating C_Q in one run step.

It must be noted that, in Section 5.1, the expression “node neighbors” means the configuration nodes that satisfy the condition in (2). In order to generate the neighbor

```

Input.  $C_q^{\text{sampling}}, C_w^{\text{sampling}}$ , The desired Configuration  $C_P$ , Obstacle space  $C_{\text{obs}}, \alpha, \beta$ 
Output. Local path  $\mathbf{P}$  which connects  $C_q^{\text{sampling}}$  and  $C_w^{\text{sampling}}$ 
(1) Initialize a priority queue  $Q$ 
(2) SUCCESS = false;
(3) Insert  $C_q^{\text{sampling}}$  into  $Q$ 
(4) while  $Q$  is not empty and SUCCESS = false do
(5)    $C_Q = \{G_{Q,1}(x_1, y_1, \theta_1), G_{Q,2}(x_2, y_2, \theta_2), \dots, G_{Q,M}(x_M, y_M, \theta_M)\} = \text{Dequeue}(Q)$ 
(6)   for every neighbor configuration  $C_Q^{\text{neighbor}}$  in  $L = \text{Generate\_neighbor\_configuration}(C_Q)$ 
(7)     if  $C_Q^{\text{neighbor}}$  is collision-free with  $C_{\text{obs}}$ 
(8)       Insert  $C_Q^{\text{neighbor}}$  into  $Q$  according to the improved evaluation function  $f(C_Q^{\text{neighbor}})$ 
(9)     end if
(10)    if  $C_Q^{\text{neighbor}} = C_w^{\text{sampling}}$ 
(11)      SUCCESS = true
(12)    end if
(13)  end for
(14) end while
(15) if SUCCESS = true
(16)   return the constructed path  $\mathbf{P}$  by tracing the configuration from  $C_w^{\text{sampling}}$  to  $C_q^{\text{sampling}}$ 
(17) else return failure
(18) end if

```

ALGORITHM 2: The local planner connecting any two neighbor configurations.

configurations of C_Q , a local coordinate system is defined with its origin coinciding with the center grid of C_Q and x -axis aligning with the direction of C_Q . There are two ways to generate the neighbor configurations of C_Q in one run step. One is moving the boundary grids in the direction of x -axis or y -axis with a distance unit x_{step} and y_{step} according to the compact property; the other is rotating C_Q with a predefined angle unit θ_{step} . The detailed process is illustrated in Algorithm 3.

Compared with previous researchers' works, Algorithm 3 has the advantage of keeping the group's area constant in the planning. After the above procedures, the roadmap is constructed.

5.4. Querying. When performing a query, we connect the start and goal configurations to the roadmap and determine the shortest path in the roadmap between them. The querying procedure in our approach is the same as the one in classic PRM algorithm. The difference is that, in every node of the roadmap, we make use of the improved evaluation function once again. So the evaluation function at the i th node of the roadmap can be represented as

$$f(C_i^{\text{sampling}}) = \alpha \cdot h(C_i^{\text{sampling}}) + \beta \cdot S_{\text{shape}}(C_i^{\text{sampling}}, C_P), \quad i \in [1, N]. \quad (6)$$

By allocating different values of α and β , we can influence whether we prefer a path that is short but has lots of deformation relative to C_P or a path that is longer but with less deformation.

Complexity Analysis. The computational cost of our approach comes from the above three stages. In the stage of generating

configuration nodes, we generate N position samples by the medial axis technique and extend each position sample into group's configuration samples in M dimensions. In the second stage, the local planner is to connect any two neighbor configuration samples. We use the $S_{\text{shape}}(C_q^{\text{sampling}}, C_w^{\text{sampling}})$ formulation for M times to compute the minimum distance from the grid $G_{q,j}^{\text{sampling}}(x_j, y_j, \theta_j)$ to C_w^{sampling} and use Algorithm 3 to generate the neighbor configurations of C_Q for M times at most. In the third stage, the evaluation function in PRM algorithm at every node of the roadmap is called M times. Then the complexity of our planner is $O(N \cdot M + N \cdot N \cdot M \cdot M + N \cdot M) < O(N^2M + N^2M^2)$ which is equivalent to $O(N^2M^2)$.

Through the analysis of the corresponding computational complexity, the computational cost of the classic C-L method is vast; however, the configuration nodes extended from position samples in the first stage can provide guidance for the local planner and reduce the complexity of our approach.

6. Results

In this section we will show some experiment results of the approach described above. The experiment environment is in 3D, and we use two experiments to test our approach. The experiment environments are illustrated in Figure 5 in which there are four corridors A, B, C, and D, respectively. The widths of them are 2.3 m, 3.7 m, 2.3 m, and 0.8 m. In experiments 1 and 2, the desired shape of the group is a square which size is 3.5 m \times 3.5 m. And we discretize the group into 25 grids with the size of 0.7 m \times 0.7 m. The initial and goal configurations are placed in Figure 5. x_{step} and y_{step} are set to the size of the grid, that is, 0.7 m. θ_{step} is set to 5°.

Input. $C_Q = \{G_{Q,1}(x_1, y_1, \theta_1), G_{Q,2}(x_2, y_2, \theta_2), \dots, G_{Q,M}(x_M, y_M, \theta_M)\}$, Obstacle space C_{obs} , x_{step} , y_{step} , θ_{step}

Output. A neighbor list L of C_Q

- (1) Initialize a queue Q , Initialize a list L
- (2) Clone C'_Q from C_Q
- (3) **if** all grids in boundary grid of C'_Q have no collision-free neighbors
- (4) **return null**
- (5) **end if**
- /* Step 6-8 are used to rotate C_Q */*
- (6) **if** $\{G_{Q,1}(x_1, y_1, \theta_1 \pm \theta_{step}), G_{Q,2}(x_2, y_2, \theta_2 \pm \theta_{step}), \dots, G_{Q,M}(x_M, y_M, \theta_M \pm \theta_{step})\}$ is collision-free
- (7) add $\{G_{Q,1}(x_1, y_1, \theta_1 \pm \theta_{step}), G_{Q,2}(x_2, y_2, \theta_2 \pm \theta_{step}), \dots, G_{Q,M}(x_M, y_M, \theta_M \pm \theta_{step})\}$ into L
- (8) **end if**
- /* Step 9-23 are used to moving the boundary grids of C_Q according to the compact property */*
- (9) let $C_Q(k)$ represent the k th boundary grid of C_Q , $num(C_Q)$ represents the total number of boundary grid of C_Q
- (10) **for** ($k = 1; k \leq num(C_Q); k++$)
- (11) **if** $C_Q(k)$ have collision-free neighbors
- (12) let μ be the vector from $C_Q(k)$ to one of its free neighbors with step size of x_{step} or y_{step}
- (13) add $C_Q(k) + \mu$ to C'_Q and add all other grids in C_Q except $C_Q(k)$ into a queue Q according to the distance to $C_Q(k)$
- (14) **while** Q is not empty
- (15) $Q_k = \text{Dequeue}(Q)$
- (16) **if** $Q_k + \mu$ is collision-free, add $Q_k + \mu$ to C'_Q
- (17) **else** add $Q_k + \mu'$ to C'_Q , where μ' is the new direction which ensures the grid-set satisfying the compact property
- (18) **end if**
- (19) **end while**
- (20) add C'_Q into L
- (21) **end if**
- (22) **end for**
- (23) **return** L

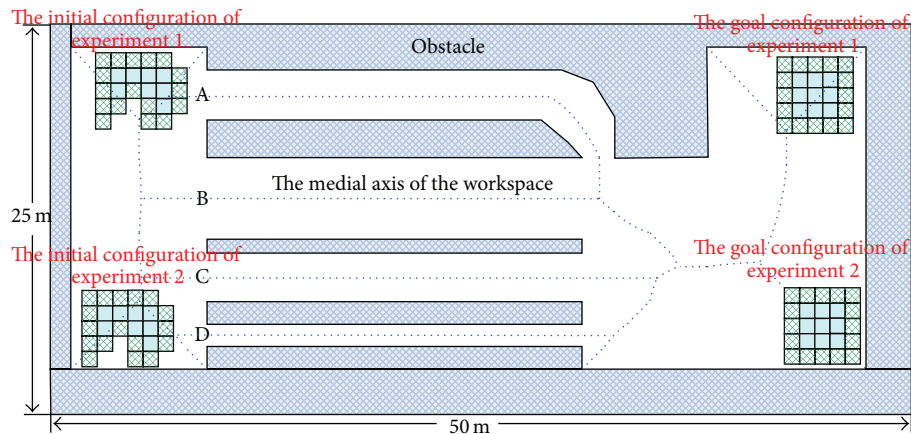
ALGORITHM 3: Generating all the neighbors of C_Q .

FIGURE 5: Generating the neighbors of one configuration.

Firstly, the medial axis technique is used to generate 100 position samples. And 12 position samples are discarded because they are too close to the obstacle boundaries. Then we use Algorithm 1 to extend the remaining ones into 88 configuration samples, part of which are illustrated in Figure 6. In experiments 1 and 2, we set $\alpha = 0.9$, $\beta = 0.1$ and $\alpha = 0.3$, $\beta = 0.7$, respectively. Experiment 1 and experiment

2 were implemented using Microsoft Visual C++ using the Solid collision detection package [23] and run on an Intel Core i5-2400 CPU @ 3.10 Ghz with 4 GB internal memory. The results are depicted in Figures 7 and 8.

Figure 7 depicts the shape difference between the group and the desired configuration at every step of our planning approach. Figure 8 depicts the distance between the group

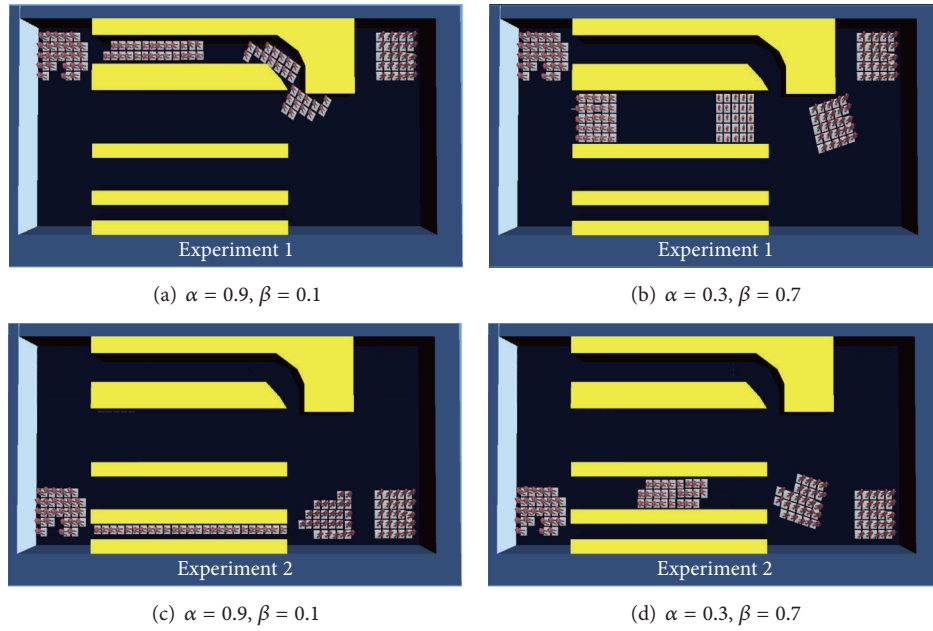


FIGURE 6: Some representative configuration samples.

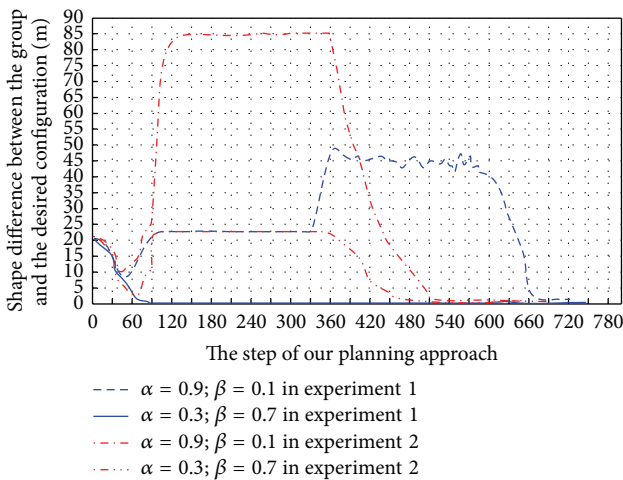


FIGURE 7: The shape difference at every step.

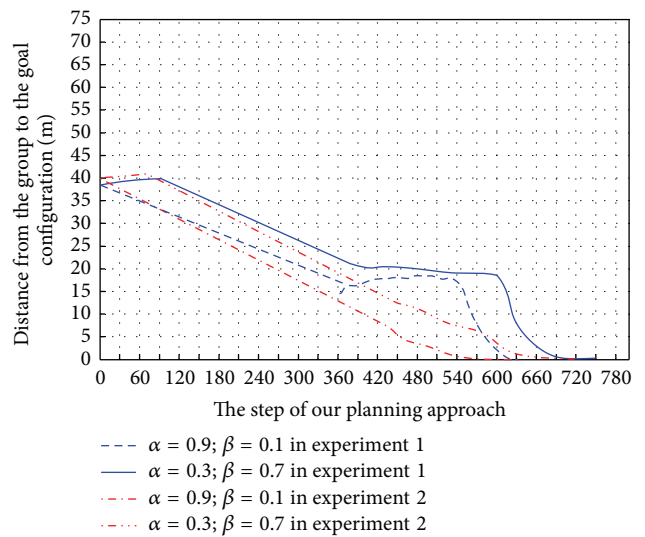


FIGURE 8: The distance between the group and the goal at every step.

and the goal at every step. The experiment results show that when $\alpha = 0.9, \beta = 0.1$, our approach selects the path across corridor A in experiment 1 and the path across corridor D in experiment 2. When $\alpha = 0.3, \beta = 0.7$, our approach selects the path across corridor B in experiment 1 and the path across corridor C in experiment 2. These two paths are longer but have less deformation than the paths across corridors A and D. In Figure 7 we can see that because the initial configuration is different from the desired configuration, the group tries to make its shape approach the desired shape in the beginning of the planning steps. So, the shape difference declines before entering the corridors. When the group enters the corridors, the deformation begins. In the worst case (corridor D), the maximum shape difference equals 85 m. In

the best case (corridor B), because its width is bigger than the size of the desired shape, the shape difference remains zero after reducing to zero. Therefore, increasing β will make the group less deformed when passing through narrow corridors. However, this will lead to the increase of the distance from the group to the goal in some planning steps, as can be seen in Figure 8.

Moreover, the computational cost of C-L method is compared with our approach in the query phase in experiments 1 and 2. The results are depicted in Figure 9.

In Figure 9 we can see that our approach has a better performance than C-L method. This is because the

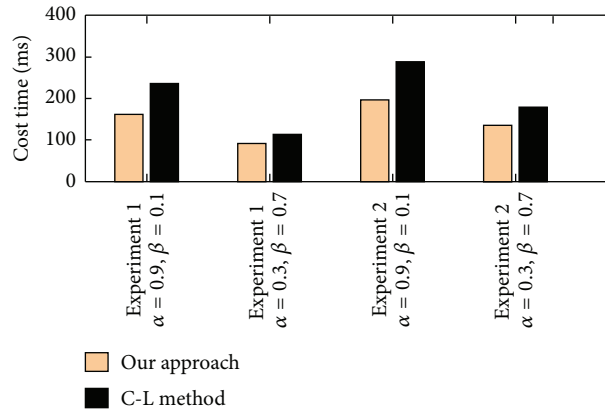


FIGURE 9: The consumed time of our approach versus C-L method in the query phase.

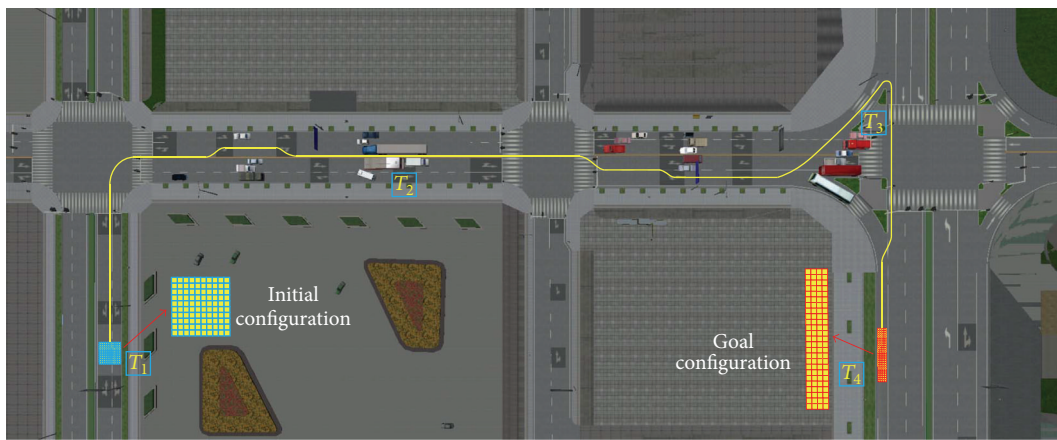


FIGURE 10: The virtual environment of the experiment.

configuration samples which are regarded as nodes of the roadmap provide guidance in planning the group's path. Further, the computational time of our approach mainly comes from the collision detection of the boundary grids. So if there are two paths whose lengths are the same, the one which has more deformations will need more computational time.

Furthermore, in order to prove the effectiveness of our approach, we perform an experiment in a more complicated environment with more grids. The scenario can be depicted in Figure 10: there are some motionless vehicles which can be regarded as parts of obstacles, and the coherent group consists of a hundred virtual individuals.

Both the proposed approach and C-L method are used to plan a path connecting the initial and goal configurations of the coherent group. And the visualization simulation results of our approach in 3D space are illustrated in Figure 11. The experimental results show that the coherent group can find a path to reach the destination in the goal configuration and be collision-free with the obstacles.

Moreover, the computational cost of our approach is compared with C-L method in this experiment, and the

comparison result is 627 s to 703 s, which means that our approach has a better performance than C-L method.

7. Conclusions

We present a novel planning method for the coherent group whose area is constant when moving toward the goal. Our approach has two advantages: compared with the method which is based on classic PRM or its variants, our approach embeds the discretization way of C-L method in local planner to keep the area invariant in the planning. Compared with C-L method, the medial axis technique in the preprocessing phase can provide guidance for generating roadmaps, which decreases the blindness of the C-L method. Moreover, different weights of α and β can produce different paths which have diverse features. And the computational cost of our approach means that it is suitable for planning paths for the coherent group.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

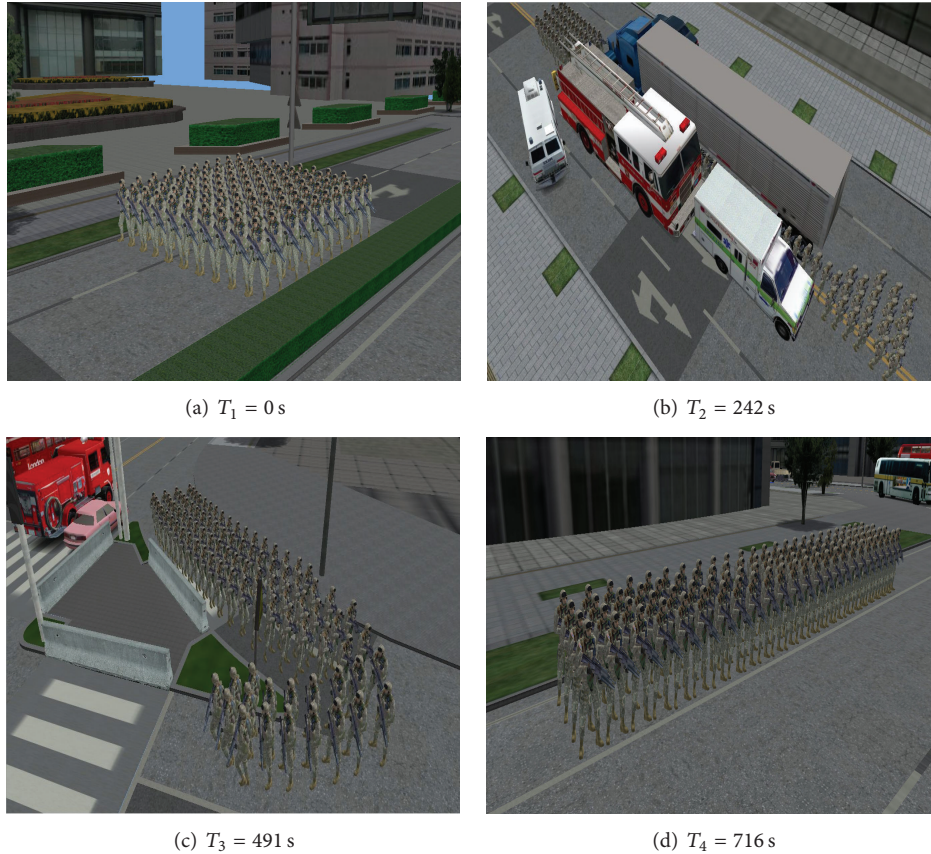


FIGURE 11: The configurations of the coherent group at different times.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China (Grants no. 61170160 and no. 61374185).

References

- [1] A. Lomuscio, W. Penczek, and H. Qu, "Partial order reductions for model checking temporal-epistemic logics over interleaved multi-agent systems," *Fundamenta Informaticae*, vol. 101, no. 1-2, pp. 71–90, 2010.
- [2] N. Wijermans, R. Jorna, and W. Jager, "CROSS: modelling crowd behaviour with social-cognitive agents," *Journal of Artificial Societies and Social Simulation*, vol. 16, no. 4, pp. 1–18, 2013.
- [3] D. Chen, L. Wang, X. Wu et al., "Hybrid modelling and simulation of huge crowd over a hierarchical Grid architecture," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1309–1317, 2013.
- [4] N. Magnenat-Thalmann and D. Thalmann, "Virtual humans: thirty years of research, what next?" *The Visual Computer*, vol. 21, no. 12, pp. 997–1015, 2005.
- [5] C. Peters and C. Ennis, "Modeling groups of plausible virtual pedestrians," *IEEE Computer Graphics and Applications*, vol. 29, no. 4, pp. 54–63, 2009.
- [6] J.-Y. Chang and T.-Y. Li, "Motion planning for reshapable objects with size conservation," in *Proceedings of National Computer Symposium*, 2005.
- [7] C. Holleman and L. E. Kavraki, "Framework for using the workspace medial axis in PRM planners," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, pp. 1408–1413, April 2000.
- [8] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the space," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1024–1031, May 1999.
- [9] C. W. Reynolds, "Flocks, herds, and schools: a distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.
- [10] 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*, pp. 95–111, Springer, Berlin, Germany, 2004.
- [11] T.-Y. Li and H.-C. Chou, "Motion planning for a crowd of robots," in *Proceedings Of the IEEE International Conference on Robotics and Automation*, pp. 4215–4221, Taipei, Taiwan, September 2003.
- [12] T. Y. Li, Y. J. Jeng, and S. I. Chang, "Simulating virtual human crowds with a leader-follower model," in *Proceedings of the Computer Animation Conference*, Seoul, Korea, 2001.
- [13] L. Guibas, C. Holleman, and L. Kavraki, "A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach," in *Proceedings of IEEE International Conference on Intelligent Robots*, 1999.
- [14] A. Kamphuis and M. H. Overmars, "Motion planning for coherent groups of entities," in *Proceedings of IEEE International*

- Conference on Robotics and Automation*, vol. 4, pp. 3815–3822, 2004.
- [15] R. Geraerts and M. H. Overmars, “Clearance based path optimization for motion planning,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2386–2392, 2004.
- [16] A. Kamphuis and M. H. Overmars, “Finding paths for coherent groups using clearance,” in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pp. 193–202, Grenoble, France, August 2004.
- [17] J.-Y. Chang and T.-Y. Li, “Simulating crowd motion with shape preference and fuzzy rules,” in *Proceedings of the 12th International Symposium on Artificial Life and Robotics (AROB '07)*, pp. 364–367, Oita, Japan, January 2007.
- [18] J. Y. Chang and T. Y. Li, “Simulating virtual crowd with fuzzy logic and motion planning for shape template,” in *Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems*, 2008.
- [19] D. Chen, L. Wang, A. Zomaya et al., “Parallel simulation of complex evacuation scenarios with adaptive agent models,” *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, p. 1, 2014.
- [20] D. Chen, X. Li, L. Wang et al., “Fast and scalable multi-way analysis of massive neural data,” *IEEE Transactions on Computers*, vol. PP, no. 99, p. 1, 2014.
- [21] D. Chen, X. Li, D. Cui, L. Wang, and D. Lu, “Global synchronization measurement of multivariate neural signals with massively parallel nonlinear interdependence analysis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 1, pp. 33–43, 2014.
- [22] D. Chen, D. Li, M. Xiong, H. Bao, and X. Li, “GPGPU-aided ensemble empirical-mode decomposition for EEG analysis during anesthesia,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 6, pp. 1417–1427, 2010.
- [23] G. van den Bergen, *Collision Detection in Interactive 3D Environments*, Morgan Kaufmann, Boston, Mass, USA, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

