






Formation-Aware Planning and Navigation with Corridor Shortest Path Maps

Ritesh Sharma,¹  Tomer Weiss²  and Marcelo Kallmann¹ 

¹University of California, Merced, Merced, CA, USA
rsharma39@ucmerced.edu, drkallmann@gmail.com

²New Jersey Institute of Technology, Newark, NJ, USA
tommer.weiss@njit.edu

Abstract

The need to plan motions for agents with variable shape constraints such as under different formations appears in several virtual and real-world applications of autonomous agents. In this work, we focus on planning and execution of formation-aware paths for a group of agents traversing a cluttered environment. The proposed planning framework addresses the trade-off between being able to enforce a preferable formation when traversing the corridors of the environment, versus accepting to switch to alternative formations requiring less clearance in order to utilize narrower corridors that can lead to a shorter overall path to the final destination. At the planning stage, this trade-off is addressed with a multi-layer graph annotated with per-layer navigation costs and formation transition costs, where each layer represents one formation together with its specific clearance requirement. At the navigation stage, we introduce Corridor Shortest Path Maps (CSPMs), which produce a vector field for guiding agents along the solution corridor, ensuring unobstructed in-formation navigation in cluttered environments, as well as group motion along lengthwise-optimal paths in the solution corridor. We also present examples of how our multi-layer planning framework can be applied to other types of multi-modal planning problems.

Keywords: global path planning, multi-modal planning, group formation, multi-agent navigation

CCS Concepts: • Computing methodologies → Artificial intelligence; Planning and scheduling; Multi-agent planning; • Computing methodologies → Computer graphics; Animation

1. Introduction

Maintaining agent formations during navigation is a common need in computer animation, simulation and games. Example cases range from groups of agents forming complex shapes for special effects in movies to formations of soldiers in strategy war games.

Maintaining a formation during navigation requires several aspects to be considered, from formation transition and maintenance, to environment adaptation, and adaptation to other interacting agents perturbing or passing through the formation. In this paper, we focus on a particular aspect that has received little attention in previous work, which is the problem of planning and exe-

cuting a formation-aware navigation plan in arbitrarily cluttered environments. In such situations, suitable global planning and plan execution strategies are necessary.

At the global planning level, while agents moving in formation should be able to maintain their desired formation, the clearance required by such a formation may lead to a path that is too long, or even non-existent. Therefore, we take into account alternative formations that require less clearance in order to search for solution paths which can exploit narrower corridors.

We propose a navigation planner that addresses the trade-offs involved when deciding between paths that have sufficient clearance for a desired formation versus switching to alternative paths with less clearance, and thus requiring formation changes. At the planning level, this problem is addressed in two main stages: First, a multi-layer skeleton graph representing the free space is computed

Marcelo Kallmann: Now affiliated with Amazon Robotics. This work was initiated before joining Amazon and is not related to Amazon work.

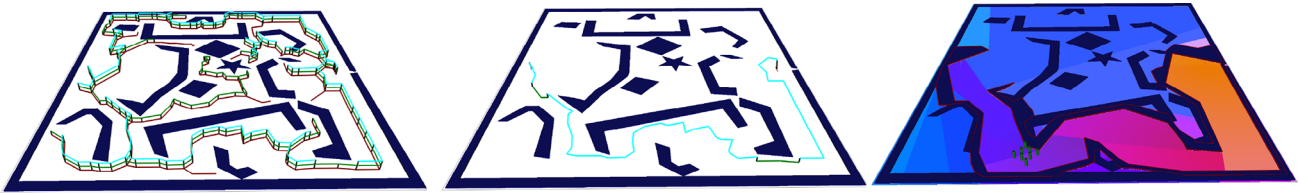


Figure 1: Formation-aware navigation example for three possible formations. Left: A multi-layer graph is first computed where each layer models clearance and transition requirements for its respective formation. Center: Our formation-aware planner extracts a solution from the graph taking into account path length, required per-formation clearance, transition costs and formation preferences. Right: At the navigation stage, we introduce Corridor Shortest Path Maps (CSPMs) for ensuring unobstructed in-formation agent navigation in the solution corridor.

where each layer observes the specific clearance requirements of one possible formation. The graph is annotated with specific per-layer navigation costs and formation transition costs. Given such graph, we apply a planner in order to search and identify formation transition points minimizing an overall cost function encoding the modelled trade-offs. At the navigation stage, we introduce CSPMs, a novel approach for path following. CSPMs produce a vector field that guides agents along lengthwise-optimal paths in the solution corridor while minimizing conflicts among the formation maintenance constraints and ensuring unobstructed navigation. See Figure 1 for an overview of the main stages of our overall method.

With respect to local behaviour, in this work, we explore agent formations maintained with Position-Based Dynamics (PBD) constraints [MHHR07, BKCW14, BMM17], which support generic formation shapes. Such constraints are simple yet powerful, allowing robust formation control in conjunction with other generic fine-grain artistic or PBD crowd-oriented motion constraints.

We perform several evaluation experiments to demonstrate our formation-aware planning and navigation method, both in qualitative and quantitative ways. Our experiments demonstrate the ability to compute customized formation-aware global plans in cluttered environments (see Figures 14 and 15), and the ability to robustly ensure that formations are not disrupted in challenging environments during navigation (see Figure 19).

Overall, our work proposes new approaches for global formation-aware planning and the maintenance of formations during navigation. Our method is not specific to a particular type of agent or motion model, and can be applied to produce different types of animations. For instance, we have simulated one example where agents acting as bodyguards are protecting an individual while switching between protective formations with different clearance requirements in order to move across the environment. Each formation places agents around the individual such that complete visibility around the group can be achieved (see Figure 16). As another example, agents in formation can be utilized in computer animation for generating a diverse array of special effects. To demonstrate this aspect, we have simulated a scenario where agents switch between formations composing CGF letters while traversing a cluttered environment (see Figure 17). Numerous additional scenarios can benefit from formation-aware planning and navigation, for example: family members moving together while adjusting to a cluttered environment, or even adjustable robotic formations for the transportation of large objects, where each formation places robots

at different grasping points in order to accommodate the available clearance across the environment.

2. Related Work

Agents and crowds [PAKB16] are prevalent in many applications, such as visual effects, entertainment, education and defence, to name a few. Our work focuses on multi-agent planning and navigation under formation constraints in highly cluttered environments. We review below the relevant previous work in these areas.

Path planning: Multi-agent navigation often depends on path planning algorithms, for which there are numerous approaches, from discrete search methods on grid-based environments [HKLS10, GKB14] to navigation meshes [Ger10, OP13, Kal14, PF16, RP20].

A navigation mesh typically sub-divides the virtual environment into triangle-shaped regions, where an agent navigates using graph search algorithms such as A* [HNR68]. Grid-based methods divide the environment into square-like cells of variable size, which naturally provide a graph-like data structure that can be used by various search algorithms. Adaptive grids used to construct navigation meshes for path planning have also been proposed [AG13].

While path planning methods have addressed several variations of the problem, no specific considerations have been made for planning paths for agents that can be in multiple formations. Previous work has also not explored robust in-formation navigation when executing solution paths in complex cluttered environments. Recent work has instead addressed simplified versions of the problem, such as abstracting multi-agent formations as one large agent [KG15], and considering local plans with velocity-obstacles for collision avoidance [WVM09]. Overall, the approach is to let formations deform in order to accommodate to the environment using reactive collision avoidance. While this is a valid approach, the global path planning stage does not consider preferences or other formations, and the path following phase does not offer any optimality criterion.

In contrast, our work addresses formation-related constraints at the global planning level with a multi-layer graph capturing per-formation clearance requirements and costs modelling formation preferences. Each layer in our graph represents the passable corridors for one given formation, such that layer transitions represent possible formation transitions. While previous work on modelling clearance has extensively explored the use

of the Medial Axis [Ger10] for path planning, also considering dynamic obstacles [GO07], our contribution is not on how to determine clearance but in using clearance values to build a multi-layer graph in order to search for paths taking into account formation transitions according to user-defined parameters. Our work also does not address automatically detecting navigable areas from images [DSG21]; however, once those areas are detected, then our proposed multi-layer graph can be constructed on them.

Our work introduces Corridor Shortest Path Maps (CSPMs) for achieving robust corridor navigation along shortest paths when executing formation-aware plans (Section 5). CSPMs apply the traditional concept of Shortest Path Maps (SPMs) [FK20, SFK20] to a solution corridor, after closing and isolating it from the rest of the environment, such that the solution can be reliably executed by the group of agents. See Figure 1 for an example. CSPMs provide a vector field that guides the agents along shortest paths, which is a unique SPM property when comparing to other common types of vector fields such as for collision avoidance [GO07] or user-controlled behaviour [CvTZ*22].

Maintaining multi-agent formations: Managing realistic formations of agents has been a topic of interest in recent years, with some inspiration coming from real-world group behaviour observed in nature [Rey87], and from potential applications in simulation and robotics [OPA15]. Representative agent-based group simulation systems have addressed the coordinated group behaviour of flocks of birds or schools of fish with user-specified static or dynamic shape constraints [XJY*08], and the simulation of visually plausible flying insects by extending a field-based method with external forces, achieving flock morphing with pre-defined shape constraints [CLT*19]. Research efforts have also focused on visualizing crowd formation for artistic purposes, such as sketching [GD11a, GD11b], synthesis [JCP*10] and morphing between formations [ZZC*14].

Interactive formation control is one particular topic of interest. One approach is to represent formation constraints as a deformable mesh [HSK13]. The approach re-assigns agent formation roles in order to successfully maintain the formation shape after a disturbance; however, the approach is based on interactive user feedback and does not target achieving autonomous agents with collision avoidance behaviours. Additional work has focused on autonomy and has simulated formations for physically embodied agents, such as humans and robotic agents [KO11, XWY*15, RCB*17]. These works demonstrate several techniques for maintaining formations that can deform in order to accommodate a range of interactions. Other works have proposed to use a velocity-based approach to model agent formations where agents dynamically choose between user-defined formations that best avoid collisions [KG15].

Additional representative work has developed a least effort principle for coherent group navigation with inter-group and intra-group maintenance techniques [HPNM16], and a comprehensive framework for crowd simulation including global navigation, local collision avoidance and group formation control [LCH*22]. Crowd behaviour in emergencies has also been investigated considering group dynamics [KGM18, OAYG10].

While previous work has made impressive progress in our understanding of techniques to maintain agent formations, it is still not straightforward to integrate reliable formation maintenance in conjunction with navigation in cluttered environments, where narrow passages or needle-like obstacles may interfere with constraints in unrecoverable ways, as illustrated in Figure 19. Our work focuses on addressing these difficulties.

With respect to maintaining formation constraints, one important aspect to be considered is the interaction between agents in formation and other agents interfering with the formation. Our work demonstrates that PBD can be used to provide a suitable framework for addressing such situations. PBD is widely utilized for artistic effects and physics-based simulation purposes [BMM17], with a recent application in crowd simulation [WLJT19], and group dynamics [W23]. We demonstrate that PBD is also effective for collision-avoidance in multi-agent formation scenarios.

Main contributions: While previous work has demonstrated several approaches for maintaining group formations, we focus on robust multi-formation planning and navigation in cluttered environments. Our work proposes two main contributions. First, we address the new problem of formation-aware global path planning by proposing a multi-layer graph construction and search methodology that can be applied to generic clearance-based multi-modal problems¹. Second, we introduce the concept of CSPMs to achieve unobstructed in-formation group navigation in cluttered environments. The main benefit of our planning approach is to consider during global planning the tradeoff between computing short solutions versus prioritizing user-defined preferred formations, taking into account formation clearance requirements. We also show that PBD can represent a simple yet suitable mechanism for maintaining agent formations.

3. Background

This section provides a quick overview of SPMs for completeness purposes.

A SPM is a cell decomposition structure computed with respect to ‘sources’, which are arbitrary segments or points representing goal locations in our scenarios. A single-source SPM computed for a particular planar environment encodes globally optimal Euclidean shortest paths to reach the source from *all* points in that environment. The cells of an SPM are regions sharing the same sequence of parent vertices on the shortest path to the source point. Figure 2-right illustrates the cell decomposition of one SPM and one path extracted from it.

We use SPMs to facilitate determination of corridors with a given clearance, and to produce a vector field on the solution corridor in order to robustly guide groups of agents in-formation. We nickname the latter CSPMs (Section 5).

We use the available SPM implementation in the SIG toolkit², which computes SPMs entirely in a GPU buffer [FK20]. Since every pixel in the GPU buffer stores its shortest path’s parent point

¹Problems involving changing modes, like shapes or motion models.

²<https://bitbucket.org/mkallmann/sig/wiki/Home>

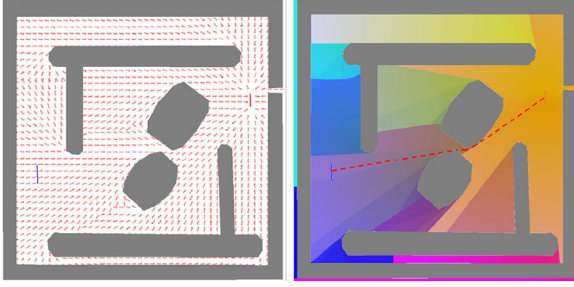


Figure 2: Left: Vector field used to guide agents along their shortest paths to the goal point. Right: The SPM generating the vector field has as source point the midpoint of the red vertical segment illustrated on the right side of the environment. The shortest path following the vector field to it is illustrated with the dashed line. Additional information on SPMs can be found in previous work [FK20].

Algorithm 1. SPM-based path extraction

```

1:  $\mathcal{T} \leftarrow \text{DelaunayTriangulation}(\mathcal{D}, O)$ ;
2:  $S \leftarrow \text{ExtractSkeleton}(\mathcal{T})$ ;
3:  $\mathcal{M}[] \leftarrow \text{ComputeSPMs}(\mathcal{D}, O, \text{src}, \text{snk}, \text{cl}_{\text{array}})$ ;
4:  $\mathcal{G} \leftarrow \text{CreateMultiGraph}(S, \mathcal{M})$ ;
5:  $\Pi \leftarrow \text{ShortestPath}(\mathcal{G})$ ;
6: return  $\Pi$ ;

```

information, agents can retrieve their next heading direction in constant time. The SPM, therefore, efficiently provides a vector field which represents an optimal control policy to guide all agents in an environment toward the goal point (see Figure 2).

4. Formation-Aware Path Planning

Our overall formation-aware path planning approach is outlined in Algorithm 1. Given an environment cluttered with polygonal obstacles $O = \{O_1, O_2, \dots, O_n\}$ and enclosed by a polygonal domain \mathcal{D} , the environment is first triangulated using a Conforming Delaunay Triangulation (DT) [HL92] in order for a skeleton graph of the environment to be built. While other triangulation schemes could be used, our Conforming DT is built by progressively adding midpoints on the boundary of the obstacles until all edges of the obstacles are represented in the triangulation, and until a desired skeleton resolution is met. These additional points, often called Steiner points, improve the obtained space discretization. By relying on a conforming version, rather than on a *Constrained Delaunay Triangulation*, we obtain a finer skeleton graph better describing possible transition points. The skeleton graph S is extracted as the adjacency graph of the free triangles, which is formed by connecting the mid points of all the edges of the free triangles and as well as the centre points of the free triangles that do not have an obstacle edge. Figure 3 illustrates the skeleton computation process for one example environment.

Next, we detect the limit clearance of the free corridors. Although other methodologies are possible, we use the SPM toolkit to represent collapsed corridors in the frame buffer. One SPM per

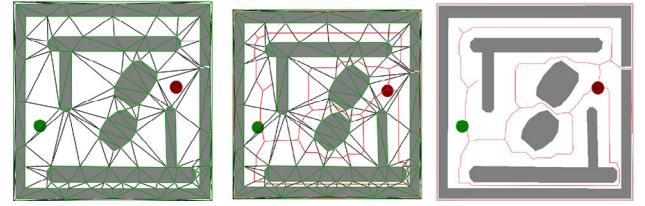


Figure 3: Skeleton extraction from an input environment: triangulation (left), adjacency graph connecting centres of adjacent free triangles (centre) and the final skeleton graph (right).

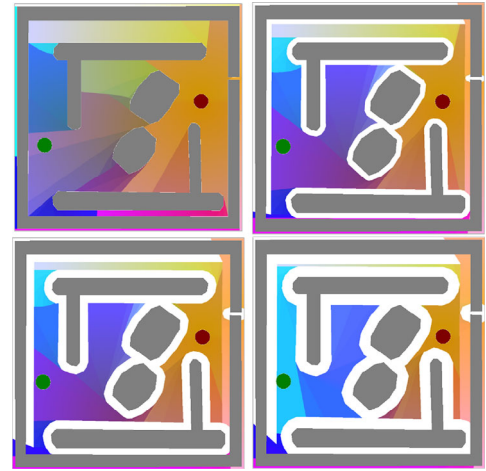


Figure 4: The top-left image shows an SPM with respect to the input obstacles without inflation. The other images show SPMs with respect to the same obstacles inflated by the different clearance values required by three given formations.

formation type is computed, where each one considers a clearance value cl_i corresponding to the needed obstacle clearance for ensuring that its respective formation can be maintained. In order to consider a clearance value cl_i , the respective SPM is built after inflating the input obstacles O by cl_i . See Figure 4 for an example.

Per-clearance SPMs are then stored in an array \mathcal{M} . For each entry in \mathcal{M} , a copy of the base skeleton S is created, however removing the edges and vertices which do not lie in a valid part of the respective SPM. This means that edges passing by corridors without enough clearance for the respective formation are removed. The shortest path information stored in the per-layer SPMs can be then discarded since these SPMs are used for the convenience of detection of collapsed corridors, given the fast queries achieved from their frame buffer grid-based representation.

Per-layer skeletons are then connected to each other by adding vertical edges connecting the existing nodes in each pair of adjacent layers, in order to form the final multi-layer skeleton graph. Each layer in the final graph represents the possible corridors for the group of agents to traverse in one given formation. Figure 5 shows one multi-layer graph example.

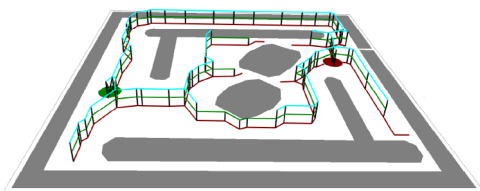


Figure 5: Multi-layer graph where layers represent valid corridors according to the clearance requirements of three different group formations.

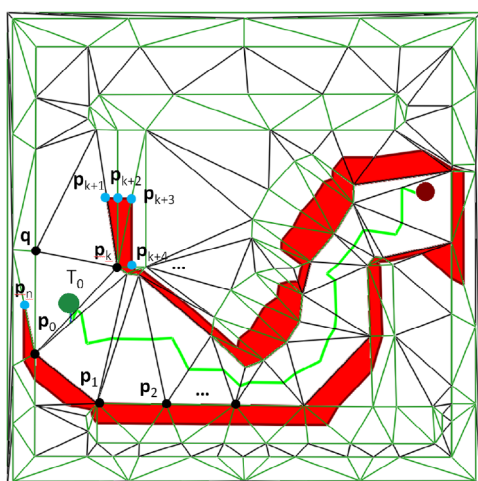


Figure 6: The construction of the corridor polygon P consists of extracting its boundary points. See text for details.

Two types of costs are then assigned to the final graph: A transition cost is assigned per vertical edge to indicate the relative preference from switching between the two connected formations, and an energy cost which is assigned per layer edge and multiplied by the length of each edge in the respective layer. The energy costs capture the cost to traverse each edge on the formation associated with the edge. With these costs assigned, the final graph \mathcal{G} is obtained. The final multi-transition path Π is then extracted from \mathcal{G} as the minimum-cost path connecting a given pair of nodes in \mathcal{G} .

5. Construction of Corridor SPMs

Given a global multi-layer solution path, we then process its respective corridor for robust navigation. The traversed corridor in the environment is defined by all the triangles intersecting with the path. Let corridor C be defined by the union of the triangles $\{T_0, \dots, T_n\}$, which are the triangles intersecting with path Π (Figure 6). We then proceed to construct the proposed Corridor SPM (CSPM) relative to Π .

First, all the edges, which are on the boundary of C , are marked as boundary edges. A boundary edge is any edge of a triangle T_i , $i \in \{0, \dots, n\}$, which is not shared with any other triangle in C .

Next, the corridor entrance is identified by edge $e = (p_0, p_k)$. Edge e is the non-boundary edge of triangle T_0 , which is the tri-

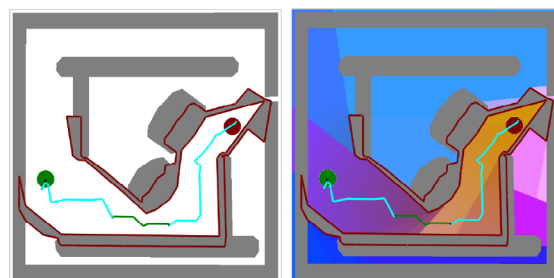


Figure 7: Extracted corridor boundary (left) and corresponding CSPM (right).

angle containing the initial point of Π , as shown in Figure 6, where $T_0 = \{p_0, p_k, q\}$.

Now the interior border of the corridor can be retrieved. This is performed by traversing all boundary corridor edges, in counter-clockwise order, starting from p_0 . Such traversal is efficiently performed with the use of a half-edge type of data structure that maintains constant-time adjacency relations of our triangulation. The outer boundary will end at vertex p_k , which is the first vertex after p_0 that is adjacent to T_0 . (See Figure 6.)

The exterior boundary of the corridor is now constructed as follows. Starting from edge $\{p_k, q\}$, the midpoints of all edges rotating around p_k in clockwise order are taken but while the edges do not reach the corridor boundary. These midpoints are points $\{p_{k+1}, p_{k+2}, p_{k+3}, p_{k+4}\}$ in Figure 6. The last edge visited is in a triangle adjacent to the corridor boundary. That edge is switched to the other edge of the triangle which is not adjacent to the corridor, and the rotation and mid-point collection continues until reaching again T_0 , when the last mid-point p_n is collected.

The sequence of points $\{p_0, p_1, \dots, p_k, p_{k+1}, \dots, p_n\}$ defines a U-shaped simple polygon P which delimits the solution corridor while leaving only one entrance to it. The CSPM can be then built by constructing an SPM in a clean environment containing only polygon P as obstacle, as shown in Figure 7. Once the CSPM is built, the underlying vector field (Figure 8) is used to guide agents anywhere in the environment, being able to provide directions directly to each agent in order to achieve unobstructed corridor traversal along the solution corridor to the goal.

6. Solution Execution in CSPMs

Given the minimum cost path Π computed from the multi-layered graph \mathcal{G} , the agents will then need to traverse the solution corridor while maintaining formation constraints. Our method directs agents to traverse the solution corridor along shortest paths encoded in the respective CSPM vector field.

6.1. Formation-aware solution corridor traversal

Our proposed approach targets the common desired situation of navigating inside the solution corridor in the shortest possible way. Here the intent is to have the group of agents to follow shortest

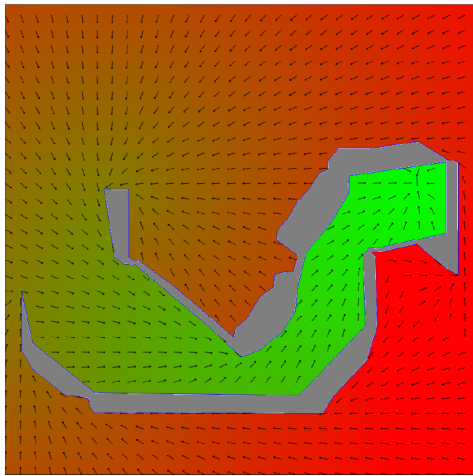


Figure 8: Vector field along with distance field encoded by the built CSPM.

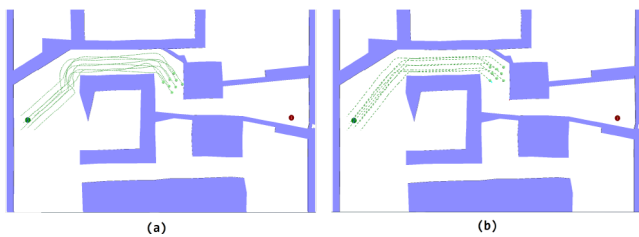


Figure 9: Trajectories of agents moving according to per-agent CSPM vector field query (a) and according to the proposed formation-aware CSPM query (b).

paths, while maintaining the current group formation, and performing formation adjustments as encoded in the solution path Π . We use CSPMs for this purpose.

CSPMs provide length-optimal corridor traversal and also guarantee unobstructed execution of solution paths, even for narrow unwanted passages where agents may get trapped with traditional path following methods. Figure 19 illustrates one such scenario where simply following a shortest path with added clearance in cluttered spaces is not sufficient.

Navigation along a CSPM may, however, lead to multi-agent navigation conflicts. For example, if one agent selfishly follows its shortest path, other agents in the group may be forced to collide with corners of the solution corridor, leading to unnatural behaviour and conflicting constraints for collision avoidance and group formation maintenance. Figure 9a demonstrates such unnatural behaviour. The obtained group motion is unnatural given the conflicting constraints and improving the results by tuning the formation stiffness and collision reaction forces becomes a challenge.

To overcome such unnatural behaviour, we introduce a formation-aware CSPM vector field querying strategy, which is based on the next CSPM generator vertex on each side of the corridor. In this approach, the size of the entire group is considered. At every

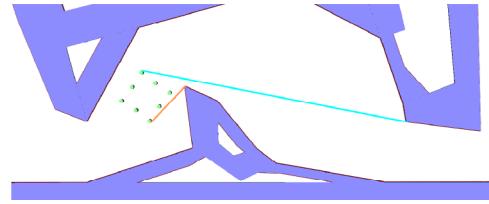


Figure 10: The proposed group-aware CSPM traversal computes the two tangent segments (in orange and cyan) from the extremes of the group to their closest CSPM vertex generators (next vertices along their shortest paths). Then, the direction used for obstacle avoidance is the underlying direction of the shortest segment (in the shown example, the orange one).

time step, the extreme tangent points of the group, with respect to the movement direction of the centre of the group, are computed as shown in Figure 10. Next, from each tangent point, the closest CSPM vertex generator on the respective corridor side is selected. A vertex generator is a corridor vertex that is convex and thus will be a corner tangent to a shortest path. The line segments from the group tangent points and their respective closest generators are then computed. The direction vector along the shortest of the two line segments provides the agents with the next velocity vector direction. All agents in the group will then follow the same direction vector without conflicts. This approach prevents undesired group rotations along the corners since it is predicting the next corners to be avoided and proactively aiming to overcome those corners, resulting in smooth trajectories. See Figure 9b.

6.2. Enforcement of formation constraints

In the context of this paper, a formation is defined as the spatial arrangement of multiple agents nearby each other. Formations are target spatial arrangements to be used when the group of agents is moving through challenging environments. These formations can be defined for facilitating efficient navigation, for demonstrating agent coordination, and for maintaining specific properties between the agents. Formations are enforced by maintaining a consistent and suitable separation between the agents, considering the available space and clearance requirements. A transition from one formation to another is defined as a quick re-arrangement of the positions of the agents in order to reach the spatial arrangement of the new target formation. We employ PBD-based constraints in order to keep the group under the current formation along the solution corridor.

Algorithm 2 summarizes the main loop of our overall navigation control procedure including formation and collision avoidance constraints. It follows the position-based crowd simulation loop described in previous work [WLJT19]. For the scope of this work, we only discuss our modifications to the original algorithm.

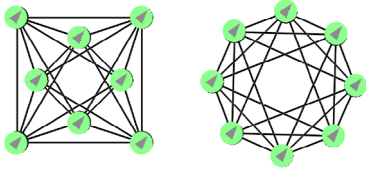
Our algorithm differs mainly at three steps: the computation of the per-agent target velocity, the introduction of formation constraints and the obstacle avoidance. At the beginning of the simulation loop, each agent i extracts the formation-aware velocity vector v_i^{sppm} from the CSPM being used for navigation, as explained in the previous

Algorithm 2. PBD crowds formation aware navigation loop

```

1: for each agent  $i$  do
2:   Retrieve  $v_i^{spm}$ 
3:   Calculate  $v_i^b$  from  $v_i^{spm}$  and  $v_i$ 
4:    $x_i^p \leftarrow x_i + \Delta t v_i^b$ 
5: for each agent  $i$  do
6:   Solve formation constraints (Sec. 6.2)
7: while iteration < maximum stability iterations do
8:   for each agent  $i$  do
9:     Position correction for short range interactions
10:  while iteration < maximum long-range iterations do
11:    for each agent  $i$  do
12:      Position correction for long range interactions
13:  for each agent  $i$  do
14:    Clamp velocity vector
15:    Position correction for obstacle collision (Sec. 6.3)
16:     $x_i \leftarrow x_i^p$ 

```

**Figure 11:** These formations are maintained by connecting each agent with its six neighbour agents with distance constraints.

section. Next, the blended velocity vector v_i^b is computed by blending v_i^{spm} and v_i , which is the current velocity of the agent. Next, the predicted position of the agent, x_i^p , is determined by moving the current position of the agent by a Δt time step in the direction of the blended velocity v_i^b .

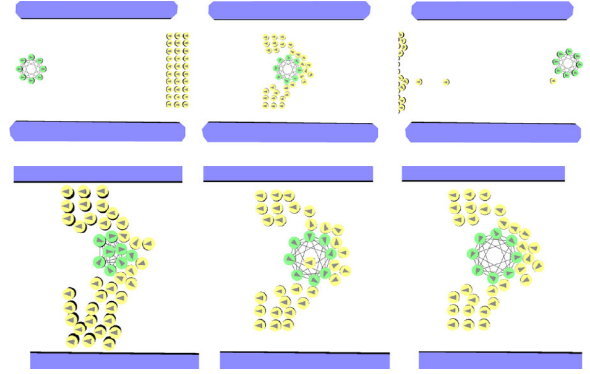
To preserve formations, we use the same approach as for simulating elastic materials with constraints [MHHR07]. Specifically, formations are maintained as distance constraints needed to enforce the shape of each defined formation. For each agent in a formation, we enforce distance constraints to maintain the original information distances to its k closest neighbours in the group. In our examples, we have used $k = 6$. As illustrated in Figure 11, depending on the formation, an appropriate number of constraints is selected in order to achieve sufficient maintenance of the original shape. Each distance constraint between agent i and j is enforced with

$$C(x_i, x_j) = \|x_{ij}\| - d = 0, \quad (1)$$

where $x_{ij} = x_i - x_j$, $x_i, x_j \in \mathbb{R}^2$ denotes the position of disc-like agents i and j , and d is a user-defined formation distance. To satisfy the constraint, PBD's positional correction for agent i is then

$$\Delta x_i = -k \frac{w_i}{w_i + w_j} C(x_i, x_j) \frac{x_{ij}}{\|x_{ij}\|}, \quad (2)$$

where $k \in [0, 1]$ is the formation constraint stiffness, and w_i and w_j are the inverse masses of the agent i and j , respectively. Formation stiffness controls the rigidity of the formation as it moves. Figure 12 illustrates the effect of different stiffness values on a circular forma-

**Figure 12:** With high stiffness, a group of agents perfectly maintains its circular formation while avoiding independent agents moving in its opposite direction (top sequence). Different behaviours can be achieved when avoiding independent agents with low, medium and high stiffness values (bottom sequence, in left-to-right order).

tion navigating through independent agents. In cases where a corridor has low clearance or is overly crowded, it may be desirable to lower the stiffness value such that the formation can achieve higher deformation in order to successfully overcome the adversarial conditions. Stiffness values can be adjusted by the user or determined automatically. For more details on simulating crowds with PBD, please refer to Refs. [WLJT19, MHHR07].

Formation constraints for a group of agents are encoded in an adjacency matrix M_f . Each entry d_{ij} of M_f defines the distance to be enforced between agent i and agent j in the formation. If a pair of agents does not have a distance constraint, the respective matrix entry is assigned as -1 . For each formation specified to the planner, a formation constraint matrix M_f is initialized with its fixed values at the beginning of the simulation.

Whenever the group of agents reaches a formation transition boundary along the solution path, the current formation matrix M_f switches to the next one, and the new formation constraints are automatically addressed, resulting in the agents gradually adjusting their positions into the new formation. If desired, additional formation morphing control can be achieved by choosing a customized blending function to gradually transform the values of a current formation matrix to the values of the next formation matrix.

6.3. Obstacle avoidance

A naive multi-agent collision avoidance does not readily apply for agents in formation, since any positional correction that affects one agent will in turn disrupt the group's formation distances. Thus, we extend the collision avoidance behaviour to achieve formation preservation by applying collision reactions to all agents of the group.

As illustrated in Figure 13, consider that an agent at time step t_i is at position x and at the next time step t_{i+1} it will be at point p . To avoid the collision, first the penetration depth of the agent is computed with the distance from the point of collision q and p . Next, the agent is moved in the normal direction $q\vec{p}$ by the amount of the

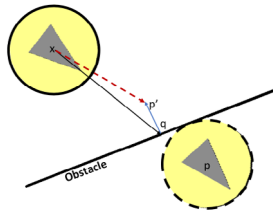


Figure 13: The same reaction vector \vec{q}' for obstacle avoidance is applied to all the agents in the group formation.

penetration depth. Then, the collision-free predicted position p' for the agent is determined by the sum of \vec{q}' and \vec{x}_q . To keep the formation, the penetration information is computed only for the first agent colliding with the obstacle, and then the same resulting reaction correction vector \vec{q}' is applied to all the agents in the group.

In regards to agent-agent interactions, we utilize the short range constraints proposed by Weiss *et al.* [WLJT19]. PBD constraints also handle collision avoidance with dynamic obstacles such as independent agents by iteratively correcting positions in the PBD simulation loop.

7. Results and Discussion

Our multi-layer graph construction and search procedure ensure that solution paths result in formation transitions that adapt to the available clearance in the environment as the group of agents traverses a solution corridor.

Figure 21 illustrates a group of agents traversing a solution path. The group starts with the largest formation and eventually changes to a smallest formation when passing through the narrowest passage (Figure 21c). As soon as the corridor becomes wider, the formation switches back to the preferred larger formations. Figure 22 shows an additional example animation obtained.

Figure 20 demonstrates a group of agents traversing through a solution corridor while avoiding adversarial agents. It can be seen that the PBD constraints are able to keep the group in formation and at the same time allow some deformation to occur for the incoming agents to pass through. The rigidity of the group formation can be controlled by varying the stiffness parameter. The higher the stiffness parameter is set, the more rigid the formation will be.

7.1. User-controllable solution extraction

To evaluate our multi-layer path search, we run our planning algorithm on two different scenes as shown in Figures 14 and 15. We show that user-defined costs can successfully lead to different types of possible solutions. In both scenes, the formation switch cost S_c in the graph is kept the same while the energy cost E_c is changed in order to obtain paths of different lengths and costs. E_c represents the cost per metre to move in each formation and is applied as a multiplier to graph edge lengths during the multi-layer path search.

Figures 14 and 15 illustrate the different paths extracted according to the user-defined parameters. A collection of different corri-

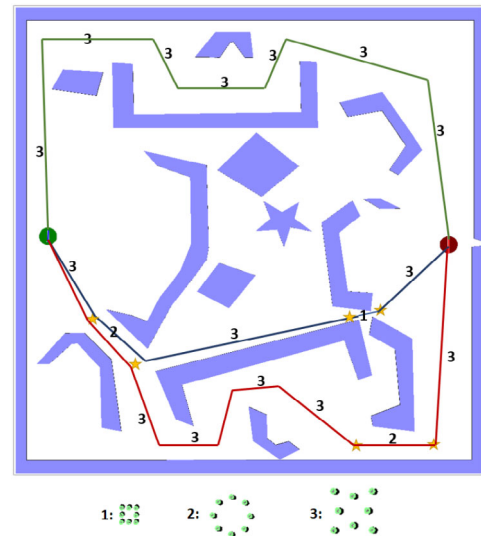


Figure 14: Three different path solutions between source (green) and destination (red) can be obtained with different sets of parameters. The numbers nearby path sections represent the selected formation on each sector, where the larger the number, the more clearance is required for the selected formation. The transition points are marked with a star shape in yellow.

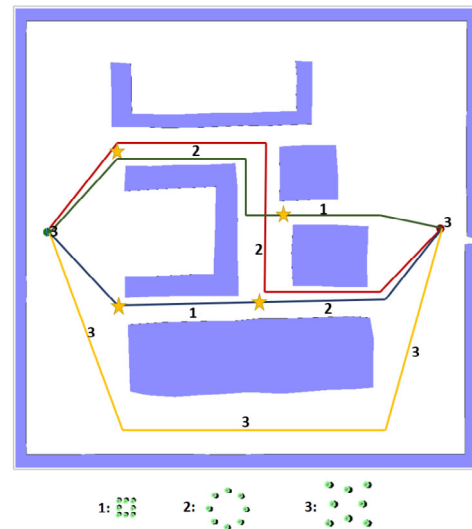


Figure 15: Four different path solutions between source (green) and destination (red) can be obtained with different sets of parameters. The numbers nearby path sections represent the selected formation on each sector, where the larger the number, the more clearance is required for the selected formation. The transition points are marked with a star shape in yellow.

dors exploring the trade-offs between clearance and length in the solutions can be clearly observed.

Table 1 quantitatively summarizes our analysis of the two scenes. For both Scenes 1 and 2, we varied E_c to illustrate the distinct paths that can be extracted. The table also presents total costs T_c and path

Table 1: Influence of parameters on path result. Cl : clearances required for each considered formation, E_c : cost per metre to move in each formation (a multiplier to graph edge lengths in the cost function), T_c : solution total cost, PL : solution path length in metres.

Scene	Cl	Path	Colour	E_c	T_c	PL (m)
1 (Figure 14)	8,14,20	a	Red	50,5,2	2946.3	1373.6
		b	Blue	50,5,35	5306.87	896.326
		c	Green	40,28,4	1539.66	1539.66
2 (Figure 15)	16,20,24	a	Red	20,9,28	8710.8	843.626
		b	Blue	10,19,28	7789.22	778.922
		c	Green	15,10,5	5649.75	846.335
		d	Yellow	20,20,5	5878.54	1167.71

lengths PL . The smaller the value of E_c for a particular Cl clearance value, the higher the priority of the formation requiring Cl for a group of agents to navigate in the environment. This can be observed from the PL and T_c values shown in the table.

In both Scenes 1 and 2, the multi-layer path search addresses the trade-offs between path length along feasible corridors and minimizing the number of formation changes.

In Scene 1, while path b is the shortest of all the obtained solution paths, the group has to undergo multiple formation changes and has to squeeze into the smallest formation to pass a narrow gap, which may not be a desirable behaviour in some scenarios, as for example in the case of a group of rescuers carrying an injured person from a fire-affected site. Paths a and c in Scene 1 do not represent the best solution in terms of length but require less formation changes and do not need to shrink to the smallest formation. Path a or c will be selected depending on the relative importance of number of formation transitions and path length.

Similar trade-offs can be observed in Scene 2, where path b is considered to be the best choice in terms of length alone, and the other paths are obtained depending on the energy costs associated to each layer as listed in Table 1.

The presented results demonstrate the ability of our multi-layer graph to explore meaningful combinations of corridors in order to extract a multi-layer solution with formation transition points according to user-specified parameters.

7.2. Application examples

Our formation planning approach can be applied to numerous applications and we highlight a few specific examples below.

One possible application of our formation-aware planning is to orchestrate the movement of a group of agents which need to be in strategically designed formations. This scenario appears in the case of a team of bodyguards protecting an individual while moving. As illustrated in Figure 16, we have simulated green agents acting as guards strategically adjusting their positions to adapt to various formations as they expertly navigate through the environment while maintaining full visibility coverage of the surrounding

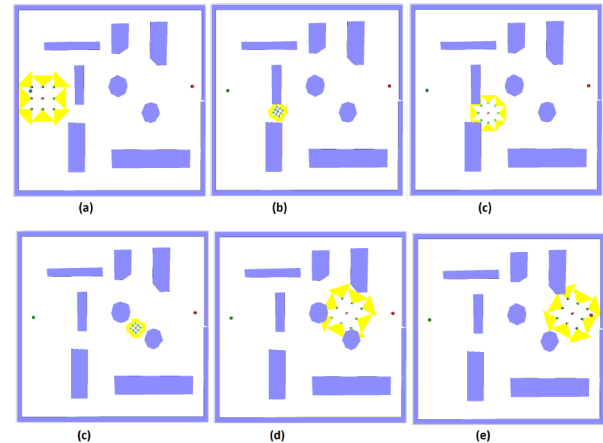


Figure 16: Example demonstrating agents in formations surrounding an individual (in red) for protection. Yellow visibility sectors are illustrated in order to show that the selected formations, each with a different clearance requirement, maintain visibility coverage around the group. While the preferred formation requires the highest clearance, when needed the group switches to smaller formations in order to reach the goal location.

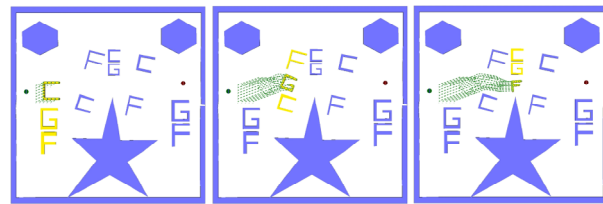


Figure 17: Example demonstrating our method for the creative animation of CGF letters, modelled as group formations, moving while taking different sizes and switching between themselves in order to form the acronym at different locations of the cluttered environment.

space. The red agent in the centre represents the individual being protected.

Another application is provided in the area of special effects where words or acronyms can be simulated with group formations in order to achieve a desired animation result. Such scenarios are, for instance, applicable to simulating opening ceremonies of events, where individuals may form words with group formations in order to captivate the audience. In the provided example, formations are used to simulate letters moving across a cluttered environment, for instance, for a possible use in a brand advertising animation. As illustrated in Figure 17, agents correctly converge to create the acronym **CGF** (Computer Graphics Forum) a few times during their seamless navigation through the environment. The simulation of formations can generate captivating and expressive elements in various domains, such as in computer games and special effects for sporting events, enabling creative communication ways for animating messages and also for other types of artistic expression.

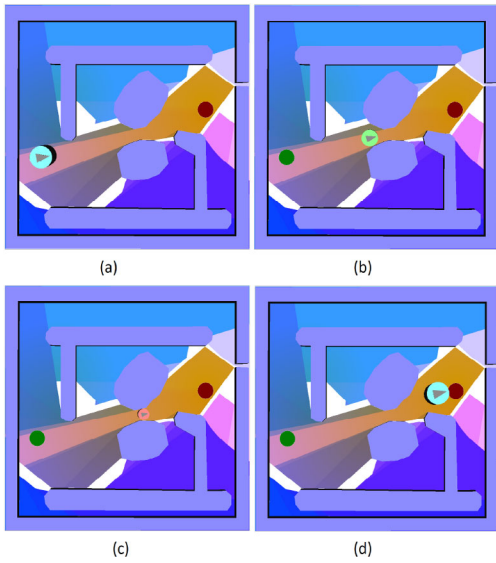


Figure 18: Example of a multi-modal object moving through a solution corridor obtained by our multi-layer planning method. The colour of each cylinder represents that a different object size can be accommodated. For instance, the different sizes can indicate a specific type of transportation that can be used. Each transportation mode can be associated with its specific energy cost and mode-switching costs can model the cost of switching between two types of transportation.

7.2.1. Generic multi-modal planning problems

The proposed multi-layer graph can also be used to solve different types of multi-modal planning problems. Since the multi-layer graph encodes a different clearance constraint in each layer, it can be used for modelling generic multi-modal planning problems where each mode imposes a specific clearance constraint during navigation.

For example, Figure 18 illustrates a problem where an object can switch between shapes requiring different clearance amounts in order to navigate through the environment. These different clearance constraints can represent a shape-shifting object or can also be considered as switching between different vehicles for transportation. With the proposed multi-layer graph, one can easily model such types of problems and compute the most cost-effective multi-modal solution for a planning query. A typical application in this area is planning paths for a shape-shifting robot which changes its shape according to the available clearance.

7.3. Comparing the CSPM formation-aware navigation with regular path following

In order to show the effectiveness of our CSPM approach for solution execution, we evaluate our method against a regular path following approach. While traditional path following is simpler and sufficient when an individual agent follows a path, for a group traversing a cluttered environment several unwanted difficulties can

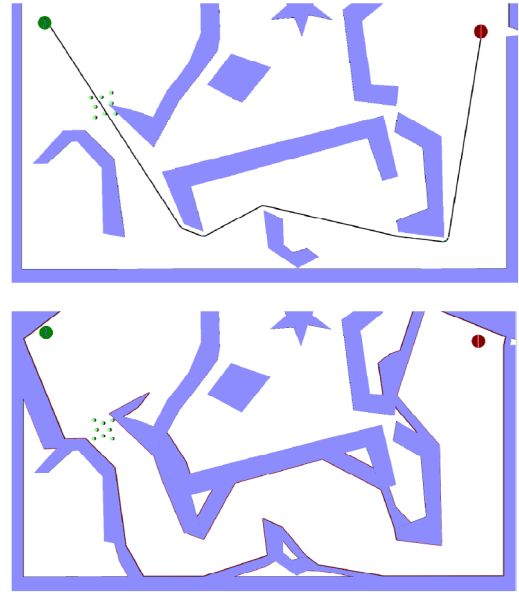


Figure 19: Regular path following approaches for groups of agents may lead to agents trapped by sharp obstacle corners (top), while CSPMs eliminate such problems by closing the solution corridor from unwanted passages (bottom).

arise. Figure 19 illustrates one particular difficulty often encountered in our tested environments.

Figure 19-top illustrates agents trapped in a sharp obstacle corner. Such a situation happens in narrow spaces where an agent gets pushed to the side by the other agents, during formation maintenance, and ends up getting trapped on the other side of an obstacle. With the proposed CSPMs, since the solution corridor is isolated and closed from unwanted corridor entrances, such disturbances are removed, as shown in Figure 19-bottom.

7.4. Implementation and performance

Our implementation was executed on an Intel Core i7 2.90-GHz computer, with 32 GB of RAM and an NVIDIA GeForce RTX 2060 graphics card. We have set the Δt step to 1/2 s and the agent size to 2.5 m. Our examples have used a stiffness value of 0.005, except for the example shown in Figure 20, where the lower value of 0.000056 was used. The blending coefficient for computing the blended velocity vector v_i^b between the retrieved SPM vector v_i^{spm} and the current velocity vector v_i was kept at 0.0385 for all the experiments. The number of stability iterations is 1 for position corrections due to short range interactions. In most of our experiments, we have not used long range interactions except in the scenario shown in Figure 20 where the group has to avoid incoming adversarial agents. The iteration for correcting position for long range collision avoidance is set to 1.

Table 2 shows the computational times for the four scenes shown in Figures 1, 7, 21 and 22. Labels T_G , T_P and T_C represent, respectively, the time (in seconds) to build multi-layer skeleton graphs, extract paths, and to compute CSPMs. The multi-layer

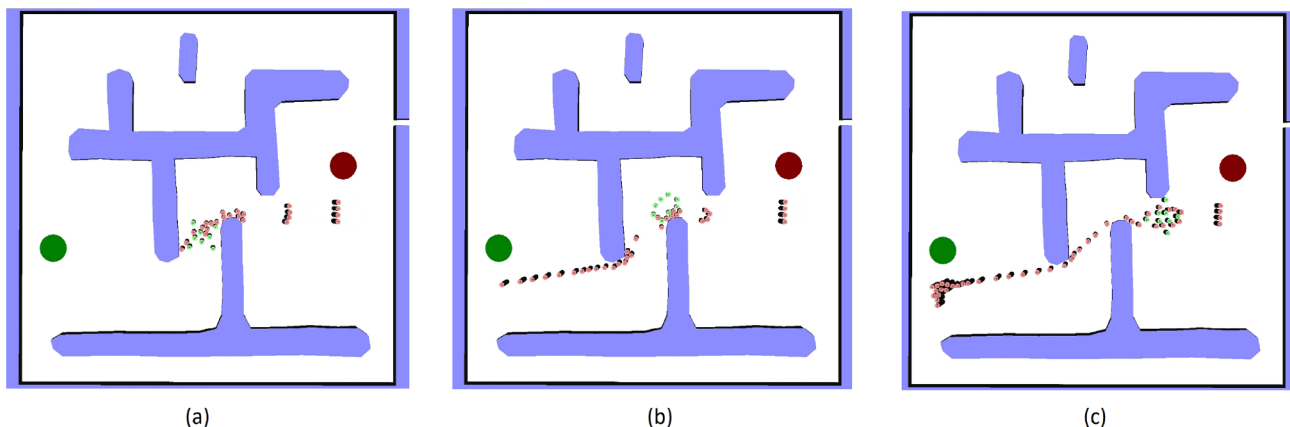


Figure 20: A group of agents (in green) switching between formations while facing adversarial agents (in red) along narrow passages.

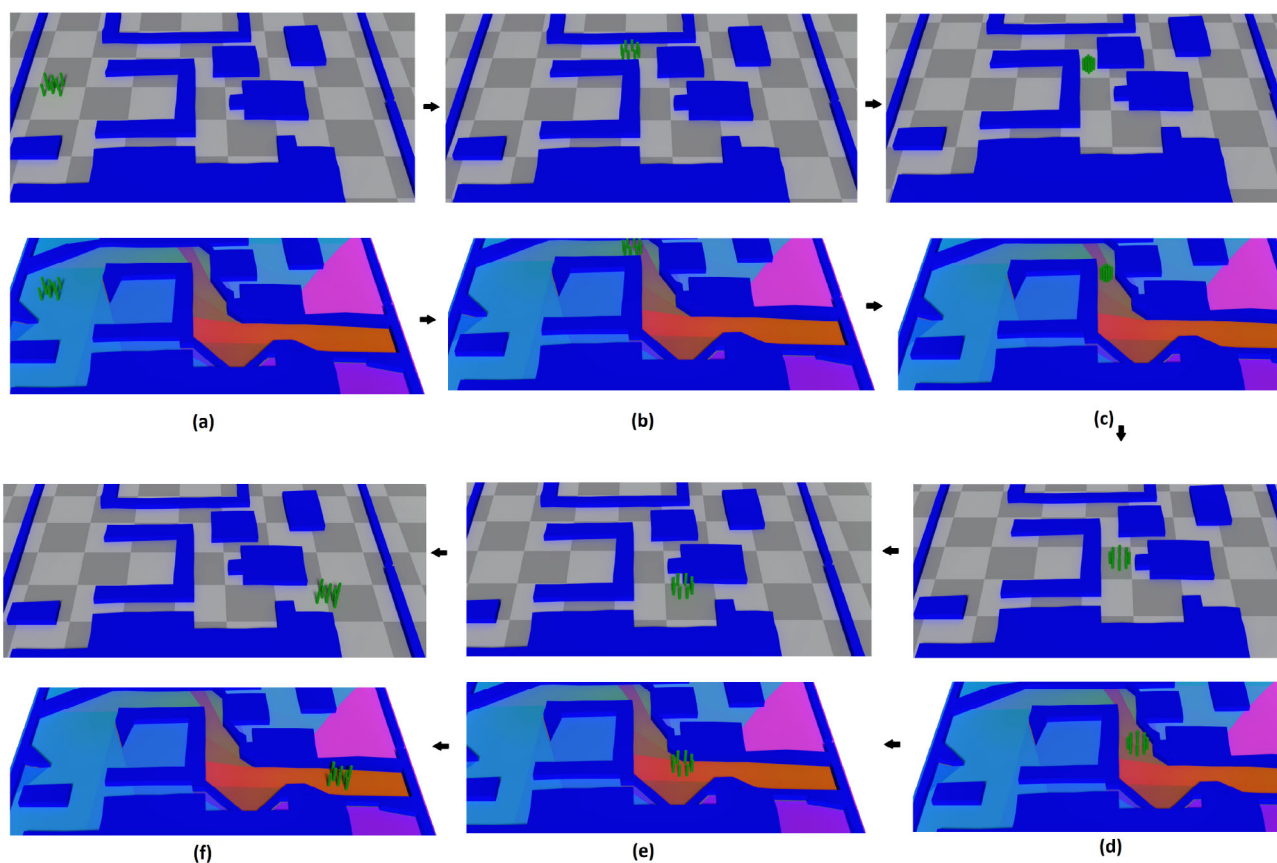


Figure 21: Example sequence (in clockwise order) illustrating required formation changes for traversing a solution corridor. The bottom row shows the closed solution corridor and the coloured cells of the respective CSPM.

graph construction depends on the SPM computations which in turn depend on the number of the vertices $V_{R_{obs}}$ of all the obstacles in the scene. $V_{C_{obs}}$ is the number of vertices of the constructed corridor obstacle. More details on the time complexity of computing SPMs can be found in the text of Farias and Kallmann [FK20].

The computational time for computing CSPMs clearly depends on the number of vertices of all the obstacles in the environment. In our implementation, depending on the complexity of the environment, our proposed method can be used for interactive navigation, as illustrated in Figure 23. The accompanying video to this paper presents a screen capture of the interactive animation obtained.

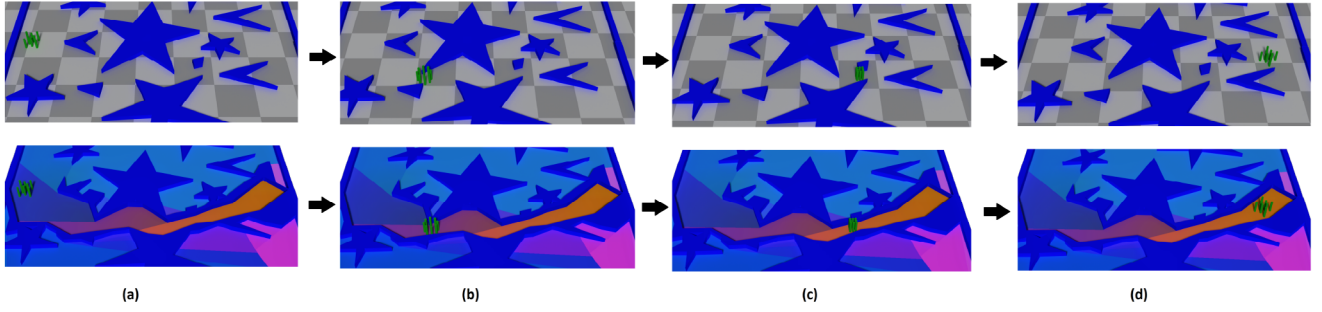


Figure 22: Additional example illustrating two required formation changes for traversing a solution corridor. The bottom row shows the closed solution corridor and the coloured cells of the respective CSPM.

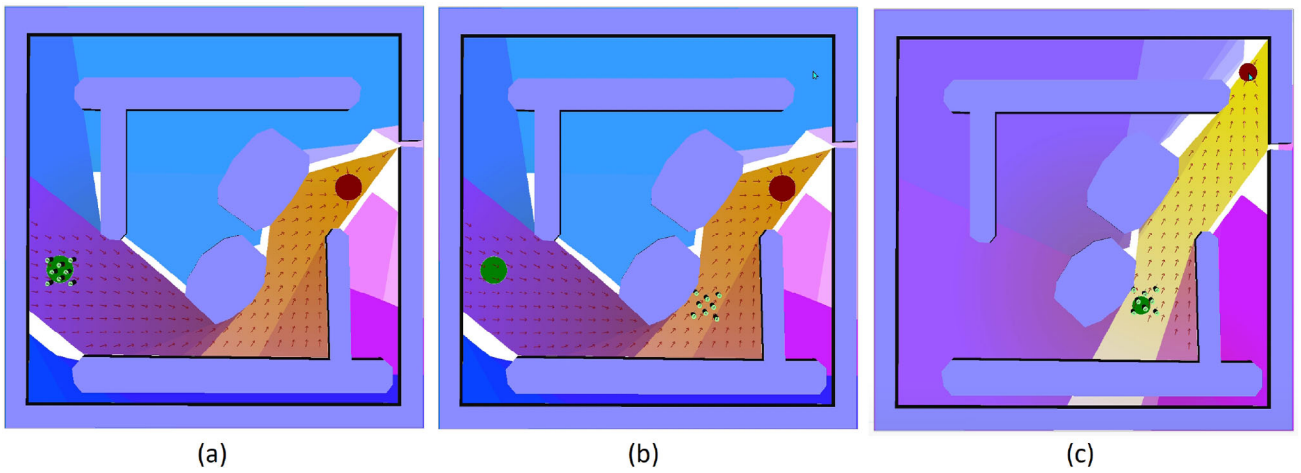


Figure 23: Illustration of our method being used interactively. While the group of agents start moving from the green disk position (a) to the first red disk position (b), a new position for the red disk goal is given with a mouse click (c) and a new CSPM is quickly computed and used to guide the agents towards the new goal without interruptions.

Table 2: Computational time (in seconds) for the different stages of planning. $V_{R_{obs}}$: total number of vertices on all obstacles in the given scene, T_G : multi-layer graph computation time, V_G : number of nodes in the multi-layer graph, T_P : path computation time, $V_{C_{obs}}$: number of vertices from the obtained corridor obstacle and T_C : corridor SPM computation time.

Scene	$V_{R_{obs}}$	$T_G(s)$	V_G	$T_P(s)$	$V_{C_{obs}}$	$T_C(s)$
Figure 1	126	4.35	1270	0.00043	121	0.51
Figure 7	60	1.59	754	0.00021	72	0.29
Figure 21	163	3.91	1367	0.00042	127	0.49
Figure 22	116	3.70	1203	0.00048	111	0.39

8. Limitations and Future Work

We have not integrated in this work any realistic motion models intended to simulate one particular type of agent. The speed modulation observed in the accompanying video merely reflects how our simulation advances according to the given time step and enforced constraints. Another limitation is that we represent the clearance needed for each formation by employing the popular simplification

of inflating obstacles, which is equivalent to computing clearances for disk shapes capturing the space needed by each formation. We have left the integration of a more precise clearance representation for generic types of shapes as future work. Finally, our multi-layer skeleton graph construction relies on the triangulation of the environment, and the number of triangles affects the number of skeleton vertices and segments used to represent clearance information. As part of our future work, we intend to investigate a triangulation refinement method which can lead to a skeleton approximation of the medial axis, therefore more accurately representing clearance information. There are several additional opportunities for future work. A clear main area for further development is the integration of agent-specific movement models, for instance, to address groups of vehicles or groups of people with realistic kinematic and dynamic constraints.

9. Conclusion

In this work we address the problem of formation-aware path planning and navigation in cluttered environments. Our approach

consists of three main steps: multi-layer graph construction where each layer models formation-specific clearance requirements, global planning on the multi-layer graph and unobstructed navigation with CSPMs.

We demonstrate that our proposed multi-layer graph is capable of representing user-defined formation constraints and costs for specifying the types of obtained solutions. Additionally, the new concept of corridor SPMs guarantees unobstructed in-formation navigation. We also present several examples demonstrating the usefulness of being able to keep the agents in a formation together, and several results demonstrating convex and non-convex formation transitions, and stiffness control in face of adversarial agents.

Acknowledgements

This research was partially sponsored by the Army Research Office under Grant Number W911NF-17-1-0463. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

References

- [AG13] AKAYDIN A., GUDUKBAY U.: Adaptive grids: An image-based approach to generate navigation meshes. *Optical Engineering* 52, 2 (2013), 7002. <https://doi.org/10.1117/1.OE.52.2.027002>.
- [BKCW14] BENDER J., KOSCHIER D., CHARRIER P., WEBER D.: Position-based simulation of continuous materials. *Computers & Graphics* 44 (2014), 1–10.
- [BMM17] BENDER J., MÜLLER M., MACKLIN M.: A survey on position based dynamics, 2017. In *EG'17: Proceedings of the European Association for Computer Graphics: Tutorials* (Goslar, DEU, 2017), Eurographics Association, pp. 31. <https://doi.org/10.2312/egt.20171034>.
- [CLT*19] CHEN Q., LUO G., TONG Y., JIN X., DENG Z.: Shape-constrained flying insects animation. *Computer Animation and Virtual Worlds* 30, 3-4 (2019), e1902.
- [CvTZ*22] COLAS A., VAN TOLL W., ZIBREK K., HOYET L., OLIVIER A.-H., PETTRÉ J.: Interaction fields: Intuitive sketch-based steering behaviors for crowd simulation. *Computer Graphics Forum* 41, 2 (2022), 521–534.
- [DSG21] DOĞAN Y., SONLU S., GÜDÜKBAY U.: An augmented crowd simulation system using automatic determination of navigable areas. *Computers & Graphics* 95, (2021), 141–155.
- [FK20] FARIAS R., KALLMANN M.: Optimal path maps on the GPU. *IEEE Transactions on Visualization and Computer Graphics* 26, 9 (2020), 2863–2874.
- [GD11a] GU Q., DENG Z.: Formation sketching: An approach to stylize groups in crowd simulation. In *Graphics Interface*. Cite-seer (2011), pp. 1–8. <https://dl.acm.org/doi/10.5555/1992917.1992919>.
- [GD11b] GU Q., DENG Z.: Generating freestyle group formations in agent-based crowd simulations. *IEEE Computer Graphics and Applications* 33, 1 (2011), 20–31.
- [Ger10] GERAERTS R.: Planning short paths with clearance using explicit corridors. In *IEEE International Conference on Robotics and Automation* (2010), pp. 1997–2004.
- [GKB14] GARCIA F. M., KAPADIA M., BADLER N. I.: GPU-based dynamic search on adaptive resolution grids. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), 1631–1638.
- [GO07] GERAERTS R., OVERMARS M. H.: The corridor map method: Real-time high-quality path planning. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (2007), pp. 1023–1028. <https://doi.org/10.1109/ROBOT.2007.363119>.
- [HKLS10] HENDERSON M., KIDER J. T., LIKHACHEV M., SAFONOVA A.: High-dimensional planning on the GPU. In *2010 IEEE International Conference on Robotics and Automation* (2010), pp. 2515–2522. <https://doi.org/10.1109/ROBOT.2010.5509470>.
- [HL92] HANSEN A. J., LEVIN P. L.: On conforming delaunay mesh generation. *Advances in Engineering Software* 14, 2 (1992), 129–135.
- [HNR68] HART P. E., NILSSON N. J., RAPHAEL B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.
- [HPNM16] HE L., PAN J., NARANG S., MANOCHA D.: Dynamic group behaviors for interactive crowd simulation. In *SCA'16: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2016), Eurographics Association, pp. 139–147.
- [HSK13] HENRY J., SHUM H. P., KOMURA T.: Interactive formation control in complex environments. *IEEE Transactions on Visualization and Computer Graphics* 20, 2 (2013), 211–222.
- [JCP*10] JU E., CHOI M. G., PARK M., LEE J., LEE K. H., TAKAHASHI S.: Morphable crowds. *ACM Transactions on Graphics (TOG)* 29, 6 (2010), 1–10.
- [Kal14] KALLMANN M.: Dynamic and robust local clearance triangulations. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–17.
- [KG15] KARAMOZAS I., GUY S. J.: Prioritized group navigation with formation velocity obstacles. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), IEEE, pp. 5983–5989.
- [KGM18] KULLU K., GÜDÜKBAY U., MANOCHA D.: ACMICS: An agent communication model for interacting crowd simulation:

- JAAMAS track. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS* (2018), vol. 2, International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), pp. 1177–1179.
- [KO11] KARAMOUZAS I., OVERMARS M.: Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (2011), 394–406.
- [LCH*22] LIU P., CHAO Q., HUANG H., WANG Q., ZHAO Z., PENG Q., YIP M. K., LIU E. S., JIN X.: Velocity-based dynamic crowd simulation by data-driven optimization. *The Visual Computer* 38, 9-10 (2022), 3499–3512.
- [MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.
- [OAYG10] OĞUZ O., AKAYDIN A., YILMAZ T., GÜDÜKBAY U.: Emergency crowd simulation for outdoor environments. *Computers & Graphics* 34, 2 (2010), 136–144.
- [OP13] OLIVA R., PELECHANO N.: NEOGEN: Near optimal generator of navigation meshes for 3D multi-layered environments. *Computers & Graphics* 37, 5 (2013), 403–412.
- [OPA15] OH K.-K., PARK M.-C., AHN H.-S.: A survey of multi-agent formation control. *Automatica* 53 (2015), 424–440.
- [PAKB16] PELECHANO N., ALLBECK J. M., KAPADIA M., BADLER N. I.: *Simulating Heterogeneous Crowds with Interactive Behaviors*. CRC Press, Boca Raton, 2016.
- [PF16] PELECHANO N., FUENTES C.: Hierarchical path-finding for navigation meshes (HNA*). *Computers & Graphics* 59 (2016), 68–78.
- [RCB*17] REN Z., CHARALAMBOUS P., BRUNEAU J., PENG Q., PETTRÉ J.: Group modeling: A unified velocity-based approach. *Computer Graphics Forum* 36 (2017), 45–56.
- [Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (1987), pp. 25–34.
- [RP20] RAHMANI V., PELECHANO N.: Multi-agent parallel hierarchical path finding in navigation meshes (MA-HNA*). *Computers & Graphics* 86 (2020), 1–14.
- [SFK20] SHARMA R., FARIAS R., KALLMANN M.: Integrating Local Collision Avoidance with Shortest Path Maps. In *Eurographics 2020 - Posters*. T. Ritschel and G. Eilertsen (Eds.). The Eurographics Association (2020). <https://diglib.eg.org/handle/10.2312/egp20201037>.
- [W23] WEISS T.: Fast position-based multi-agent group dynamics. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6, 1 (2023), 1–15. <https://doi.org/10.1145/3585507>
- [WLJT19] WEISS T., LITTENEKER A., JIANG C., TERZOPOULOS D.: Position-based real-time simulation of large crowds. *Computers & Graphics* 78 (2019), 12–22.
- [WVM09] WILKIE D., VAN DEN BERG J., MANOCHA D.: Generalized velocity obstacles. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), IEEE, pp. 5573–5578.
- [XJY*08] XU J., JIN X., YU Y., SHEN T., ZHOU M.: Shape-constrained flock animation. *Computer Animation and Virtual Worlds* 19, 3-4 (2008), 319–330.
- [XWY*15] XU M., WU Y., YE Y., FARKAS I., JIANG H., DENG Z.: Collective crowd formation transform with mutual information-based runtime feedback. *Computer Graphics Forum* 34, 1 (2015), 60–73.
- [ZZC*14] ZHENG L., ZHAO J., CHENG Y., CHEN H., LIU X., WANG W.: Geometry-constrained crowd formation animation. *Computers & Graphics* 38 (2014), 268–276.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information