# Group Agent-based Steering for the Realistic Corner Turning and Group Movement of Pedestrians in a Crowd Simulation

Francisco Arturo Rojas KAIST frojas@kaist.ac.kr

Jin Hyoung Park KAIST, KIOST jin.h.park@kaist.ac.kr Hyun Seung Yang KAIST hsyang@kaist.ac.kr

### Abstract

Crowd simulation realism has progressed a great deal in the past decade; however, much of this progress only deals with the movement of individuals rather than group movement. We propose group agent-based steering as an alternative to assigning a member of the group as the leader for which other members must follow. Our new scheme keeps pedestrians moving around essentially shoulder to shoulder in a small group as in real life, leaving it up to the individual members for performing collision avoidance while keeping up with the group as a whole. We show how this scheme can be useful also for coherent corner turning. Furthermore, we also propose a way to handle traffic congestion by groups of agents corner turning through the use of group agent path decision making with the help of triggers. Experiments show that congestion is reduced and collision avoidance handling is simplified because there is more room for maneuvering.

**Keywords:** crowd simulation, group steering, corner turning, congestion avoidance

### **1. Introduction**

Many pedestrian crowd simulations show solitary movement of individuals as if everyone preferred to move around as loners [1][2][3][4][5]. In real life, most people like to associate themselves with others while going to places often. Thus, people do walk together shoulder-to-shoulder in pairs, triplets, and small groups if they can. Furthermore, in the circumstance of oncoming obstacles, members could temporarily separate and rejoin after maneuvering past the obstacles.

Another commonly seen trend in crowd simulations is the straight path destinations assigned for pedestrians with very little or nonexistent focus on corner turning. There are also simulations with maze-like corridors in which pedestrians can go around corners, but more often than not, they hug the corners, potentially causing congestion at the corners. This is avoidable in broad passages.

In this paper we propose a group agent steering approach for handling groups of people moving essentially shoulder-to-shoulder most of the time; the approach works wells for corner turning coherently as a group. The trigger-based corner turning path selection allows for more maneuvering space for individual collision avoidance such that pedestrians can avoid causing jams and clogging intersections in dense situations.

### 2. Related Work



*Figure 1 : Line-of-sight tests improve path [6]* 

On the topic of corner turning, different methods exist. When an optimal path is solved in a graph, it could still be visually displeasing and even unrealistic [6] as shown by the original path in Figure 1 where an agent follows waypoint to waypoint in straight lines. Rather than doing this, Bezier curve or Bspline [7] approaches could be used, but there are other simpler ways as well. Line-of-sight testing can be used to reduce the number of waypoints an agent must visit by skipping to the farthest seen waypoint after arriving at each waypoint [8]. By lifting the restriction that waypoints must be visited, the heading of the character can be changed as soon as a farther waypoint on the path is visible [6], resulting in the agent hugging corners and walls. Although this results in more natural movement when corner turning, when many agents do the same coming from different directions at the same time, it can cause traffic. Although unavoidable in narrow passageways, the problem of traffic arising from hugging the walls and corners while turning is addressed in this work.

A noticeable drawback of some previous works such as in [9] is the handling of group behavior in which there must be a group leader agent that leads the other following agents. In many cases, the group leader agent never yields its leadership to another, and there could also be the case that the leader may have to backtrack that could result in unnatural movement of the followers. Our approach needs not one of the pedestrians to be a leader; what matters is that the group is going together in the general direction as in a tourist group, and the leader or leaders can easily change based on the dynamics of the group at a given time.

### 3. Group Movement

#### 3.1 Group Agent

Group Agent with 5 Slots



Figure 2 : Group agent in wall (line) formation

In this work, a group agent is an invisible leader agent composed of five equally fixed pedestrian-wide size slots arranged in line formation as shown in Figure 2, which moves independently from the visible pedestrians, all of which are agents that follow their assigned slots in this group agent by one-to-one correspondence. The group agent's job is primary path planning for the group as a whole; it can move on the searched path and cannot collide with static nor dynamic obstacles on the path, although static obstacles as part of the environment can influence its path. For the sake of controlled corner turning discussed later, the group agent is primarily a waypoint agent that navigates through and from one waypoint graph to another. The pedestrians on the other hand are grid graph or navigation mesh agents and can collide with both static and dynamic obstacles, and so they perform collision avoidance individually while following their assigned slots in the group agent. Because of obstacles that may make pedestrians fall behind during their maneuvers following the group while agent, the individuals must adjust their speed to eventually catch up with the group agent and get back into formation.

This work makes use of only one type of group formation that is commonly seen in normal pedestrian settings, which is wall (line) formation. Other formations exist such as those that can be found in tactical situations like in sports games or first-person shooters, a few examples being V-formation, defensive circle, and two abreast in cover [10]. The wall formation in use, however, will not necessarily mean that pedestrians will move together shoulder to shoulder in the simulation, but it is the desired end behavior of a small group after maneuvering around people, obstacles, and corners, just like in real life. The flexibility of pedestrians catching up to the group agent allows for them to be in different positions, to rejoin after being separated, and to even follow each other at times, all in response to the conditions of each pedestrian's surroundings. Thus the group formation dynamically changes in implicit ways, but it eventually goes back to its default formation if the maneuvers by group members cease temporarily when no obstacles are on the path nearby. The slot positions of the group agent are statically placed and do not move so as to explicitly change the group's formation. Because of this, the simulation does not place emphasis on a single pedestrian leader, but rather on the general direction of the group, such that leaders emerge at different times or are not necessarily present. The leader in real life, if there is one, is not always ahead of the others. If a leader radically changes direction or is falling way behind, the followers do not have to wait for the leader to pass them and follow, instead they are already heading in the desired direction, like how tourists move with the tour guide when being shown around.

#### 3.2 Adjusting Pedestrian Velocities

It is highly likely that pedestrians will need to perform collision avoidance while following the assigned slots of the group agent. The group agent acts independently and does not wait for the pedestrians to finish maneuvering. Rather, it is each pedestrian's responsibility to keep up with the group agent at all times. Thus, each pedestrian must adjust its velocity based on the distance to the group agent.



Figure 3 : Speed action FSM for pedestrian

The finite state machine (FSM) in Figure 3 illustrates the speed action to take based on the distance in meters between the pedestrian and the assigned slot position in the group agent. The speed of the group agent matches that of the defined pedestrian walk speed; thus the slower states saunter and dally mean that the pedestrian should move more slowly accordingly than the group agent itself upon approach in order to avoid getting too close and sometimes even accidently passing the assigned slot if going in the same direction as the group agent. Adjusting speed is a gradual process that takes time between the states, and so random target speeds are selected from a range at each speed action state. The current speed of the pedestrian controls a speed blending parameter in animation blending between walking and running animation clips using a blend tree in Unity's Mecanim feature.

#### 3.3 Corner Turning



Figure 4 : Waypoint graphs and triggers setup

As shown by Figure 4, congestion-free corner turning is achieved by a series of carefully placed independent waypoint graphs at every intersection going to all possible turning directions and with large triggers aiding in the decision making regarding which waypoint graph path to take before turning. The placement of waypoint nodes for turning is up to the world designer but there should be enough nodes to allow for gradual smooth turning regardless of the shape and type of intersection, the extreme case being a U-turn onto another path (such as an area partly divided by a wall). At each path near the intersection, a large path-wide trigger is placed such that it can only detect the group movement coming toward the intersection (the trigger's forward vector faces toward the intersection for logic handling purposes). In the event of a trigger, a decision is made about which waypoint graph the group agent should take to make a turn in the desired direction; the decision making algorithm is as follows.

4-V	Ay-INTERSECTION-ON-TRIGGER-ENTER(pedestrian,id)	
i	n: rigidbody collider pedestrian, trigger identification id	
1: 0	rounAgent ← pedestrian.assignedGrounAgent	
2: if aroupAgent directionAssigned is true then		Direction already assigned
3	return	Do not repeat from group
4: if dot(trigger.forward.normalized, groupAgent.forward.normalized) >= 0.9		If same direction (near 1)
5:	switch id	
6:	case 1	Going north trigger
7:	switch groupAgent.direction	Desired turn direction or not
8:	case 1	Go west
9:	groupAgent.CW = CLOSEST-WAYPOINT(WaypointIdList[])	Starting nodes of graphs list
10:	break	
11:	case 2	⊳ Go north
12:	groupAgent.CW = NULL	Corner turn graph need not
13:	break	
14:		
15:	groupAgent.directionAssigned ← true	
16:		

Alg. 1 : Nearest Waypoint Graph Selection

The algorithm is initiated when a pedestrian member of the group agent enters the trigger zone (see Alg. 1). Subsequent triggers by other members are ignored, so only one decision is made by the first person to trigger since all members are following the same group agent and going in the same direction. The waypoint graph whose end node is closest to the group agent's position is chosen as the best route for turning. The group agent adjusts the path slightly on approach before following the chosen path. For the cross intersection, no waypoint graphs are needed for group agents desiring to go straight without turning. For other intersections, the waypoint graphs always determine the direction the group agent should go after the intersection when turning.

### 4. Experiment



Figure 5 : Group movement simulation

Corner turning in this simulation is set up such that if pedestrians fail to keep up with their group agents when they move out of bounds and relocate, they must run to wherever their group agents reappear in the world, and so the hugging of corners is permitted in their navigation mesh path planning because it seldom occurs when turning. The simulation balances wide corner turns and corner hugging, thus increasing the free flow of groups at intersections by reducing the pedestrian density and facilitating collision avoidance as a result of more maneuvering space. It can change user perspectives and is optionally viewable in an augmented reality app via Android and iOS.

## 5. Conclusion

We introduced a novel group agent steering approach that keeps agents coherently shoulder-to-shoulder even for corner turning while allowing flexibility for pedestrians to maneuver around obstacles independently. We used triggers for group turning path selection that helped reduce traffic at corners. Since currently simulation group members are static, in future work we shall introduce dynamic regrouping/merging of group members and more diverse situation-dependent group formations that imitate real life experiences.

# Acknowledgements

This research was supported by the IT R&D program of MKE/KEIT, (10039165, Development of learner-participatory and interactive 3D virtual learning contents technology).

# References

- [1] A. Sud et al. Real-time Navigation of Independent Agents Using Adaptive Roadmaps, ACM Symp on VR Software and Tech, Newport Beach, CA, USA, 2007
- [2] J. Berg et al. Interactive Navigation of Individual Agents in Crowded Env, Symp on Interactive 3D Graphics and Games, Redwood Shores, CA, USA, 2008
- [3] R. Narain et al. Aggregate Dynamics for Dense Crowd Sim, ACM T GRAPHIC, vol. 28, no. 5, pages 122:1-122:8, 2009
- [4] S. Guy et al. A Least-effort Approach to Crowd Simulation, ACM SIGGRAPH Symposium on Computer Animation, Madrid, Spain, 2010
- [5] S. Patil et al. Directing Crowd Sims Using Navigation Fields, IEEE T VIS COMPUT GR, Vol. 17, No. 2, pages 244-254, 2011
- [6] J. Smed and H. Hakonen. Algorithms and Networking for Computer Games, pages 108-109. John Wiley & Sons Ltd, Chichester, England, 2006
- [7] A. Watt. 3D Computer Graphics 3rd Ed. Addison-Wesley, Harlow, UK, 2000
- [8] G. Snook. Simplified 3D Movement and Pathfinding Using Navigation Meshes, M DeLoura, editor, Game Programming Gems, pages 288-304. Charles River Media, Hingham, MA, USA, 2000
- [9] C. Niederberger and M. Gross. Hierarchical and Heterogeneous Reactive Agents for Real-Time Apps. COMPUT GRAPH FORUM, Vol. 22, No. 3, 2003
- [10] I. Millington and J. Funge. AI for Games 2nd Ed., pages 144–171. Morgan Kaufmann, Burlington, MA, USA, 2009