Topology-based Global Crowd Control

Adam Barnett
University of Edinburgh
A.Barnett-1@sms.ed.ac.uk

Myung Geol Choi
University of Edinburgh
mchoi@staffmail.ed.ac.uk

Taku Komura
University of Edinburgh
tkomura@ed.ac.uk

Abstract

In this paper we propose a method to determine the flow of large crowds of agents in a scene such that it is filled to capacity with smoothly moving agents. Our approach provides a focus on coordinated and cooperative motion across the entire crowd. This is done with a view to providing control which animators can use to quickly and easily populate and fill a scene, in addition to other previous methods. We solve this global planning problem by first finding the topology of the scene using a Reeb graph which is computed from a harmonic field of the environment. The max-flow can then be calculated across this graph providing paths which the agents can follow throughout the space. We demonstrate the effectiveness of our system in creating smooth motion through comparison to another recent method.

Keywords: crowd coordination, topology

1 Introduction

Controlling agents in a crowded environment is a complex problem with many dimensions. Congestion is a common issue and its presence may cause the crowd to become undesirably slow and inefficient. Removing such congestion when it is not required is non-trivial, especially as most current methods use greedy agents focussed primarily on their own goals. Our method produces control at a higher level which, while still compatible with previous methods, also automatically distributes the crowd to avoid the formation of congestion.

In this paper we propose a form of global control over the motion of agents through a given scene. We simplify this problem by converting the planning into a max-flow problem at the topological level, thus finding the maximum number of agents that can fit through the graph. Given an environment with a set of start and end points for the agents, we find the topology of the scene using a Reeb graph which in turn is computed from a harmonic field whose source and sink points are set to the start and end points respectively. The capacities of each edge in the Reeb graph are then also computed using the harmonic field and once this is done the maximum flow can be computed over this graph, this tells us how many agents it will take to fill the scene to capacity and where we should send them to produce this result.

We demonstrate our method on a couple of different scenarios, including one with over 70,000 agents. We also provide a comparison to the Continuum Crowds method, [1], which demonstrates the smoothness of the motion provided on a number of scenes.

Contribution We provide a method for efficiently and smoothly sending the maximal number of agents from one location to another in a scene.

2 Related Work

The most common type of controller used for crowd control is agent based, such as in [2]. This type of method generally focusses on local avoidance and behaviours, with any global plans computed purely in geometric space with no consideration of other agents. There are also many methods which use a field based approach,

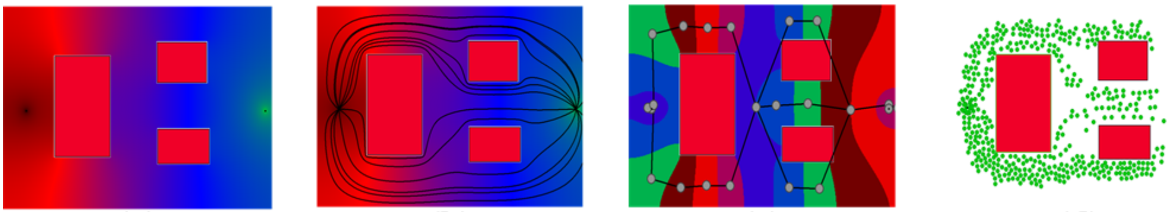
(a) (b) (c) (d)

Figure 1: The different sections of the system. They are: (a) the harmonic field, (b) the guide lines, (c) the Reeb Graph and finally (d) these are combined to provide routes for a set of agents.

similar to the harmonic field in our method, for instance [3]. Here the global plans for agents utilise the same potential field, but with again no consideration for the other agents in the scene.

Perhaps the most comparable paper in terms of global planning is [1]. Here the global plans for agents are created with a field based method which uses similar global knowledge of the situation of other agents such that a pathway will be avoided if blocked by other agents, even if the current agent cannot see that blockage. However in this case the plan produced is not global over the entire course of the motion, that is, agents will still choose paths even in the case that they are going to be blocked soon. Our method produces plans not just over the entire current state of the system, but for future states as well.

3 Harmonic Fields and Guide Lines

The harmonic field is central to the method as it provides both a representation of both the shape of the available space, which is used by the Reeb graph, and the paths through this space which agents will eventually follow. The harmonic field is discretised across a triangular mesh on the space and it is computed using the method described in [4]. A harmonic field computed for one start point and one end point can be seen in Figure 1 part (a).

As well as using the harmonic field as a representation of the space, a number of guidelines are also calculated by following the gradient of this field from the start points to the end points. These are created by sampling the space evenly in a $20X20$ grid and following each of these points forwards and back along the gradients to the start and end points. An example of the guidelines can be seen in Figure 1 part (b).

4 Computing the Reeb Graph

Computing the Reeb graph provides the topology of the space which the agents will be travelling through. In order to compute it we use the method described in [5], however we fix the resolution at which we calculate it to 128. That is, the harmonic field is divided into 128 different levels, with every contiguous set of triangles at that level taken as a node in the Reeb graph. These nodes are then connected up to any regions which they border to create the edges of the graph. This provides an abstract representation of the paths which agents could potentially take from the start to the goal points and an example can be seen in Figure 1 part (c).

However our system needs to compute the maximum flow over this graph and currently there is no information about the number of agents who can potentially move through each edge. In order to compute these capacities the harmonic field is again used. For the space between every two connected Reeb nodes the capacity lines are drawn, these run perpendicular to the gradient of the harmonic field. These lines are ideal to provide a representation of the agents who can fit through each area of the scene, as they specifically run perpendicular to the direction in which we expect the agents to be travelling at any point. Several of them are drawn between every connected pair of nodes, by following this perpendicular gradient in both directions until they reach an obstacle or the edge of the space, the shortest of those drawn is then taken as the available capacity of that edge.

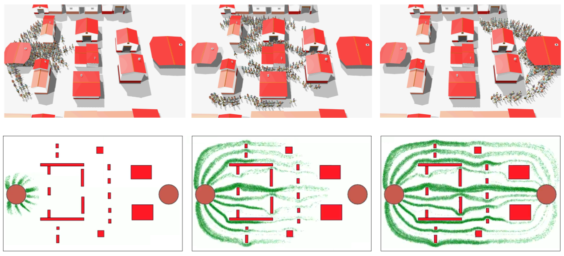

Figure 2: (Top) Small Town Example with 324 agents. (Bottom) Large Scale Example with 70,000 agents.

5 Maximum flow

The Reeb graph now has capacities for each edge and so it is possible to compute the maximum flow over this graph. This is done using the method described in [6]. This will give us the maximum capacity of the scene, allowing us to fill it up, but it also tells us how many agents will be moving along each edge in this case, allowing us to direct them through the space.

Having computed these routes for the agents they are then directed through the space at run-time by giving them guidelines, as described in Section 3. Each group of agents is given a set of guidelines which follow a set of Reeb sections allowing them to fill a particular route through the graph. They then use these guidelines as a set of waypoints which they follow using flocking to control their local movements. An example of the produced motion can be seen in Figure 1.

6 Experimental Results and Evaluation

We simulated different scenes to demonstrate our system, they are presented in Section 6.1. In Section 6.2, our system is compared to the Continuum Crowds method [1].

6.1 *Scenarios*

In the following scenarios we fill up the scene to its maximum capacity as found by our system. All experiments were carried out on an Intel core duo 3.00GHz machine.

Small Town Here 324 agents are moving from a start point on left side of the town towards a goal point on the right side of the town (top row in Figure 2). The total pre-computation time needed by our system is 3.79 seconds.

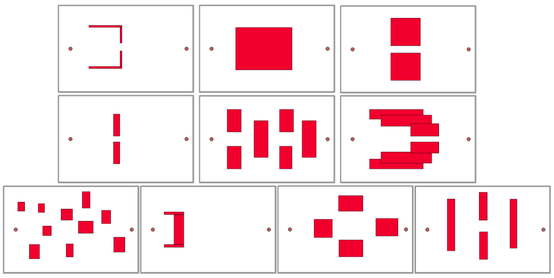

Figure 3: The experimental cases on which we compared our system to Continuum Crowds.

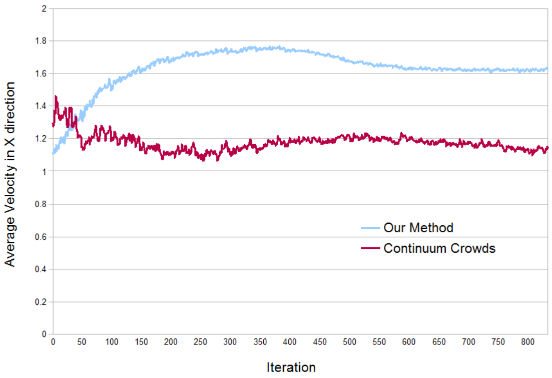

Figure 4: The average motion in the x direction for all agents in all comparisons for our system and Continuum Crowds.

Large Scale Crowd To demonstrate the scaling of our sytem, we conducted a simulation with an extremely large crowd of 70,000 agents (bottom row of Figure 2). The total pre-computation time of this scene was 18.28 seconds and this increase is caused largely by the increased number of obstacles. As the plans produced by our system exist primarily at the topological level this same scene could be applied at a larger or smaller scale (with larger or smaller agents) at no additional computational cost.

6.2 *Comparison*

In order to demonstrate the efficacy of the coordination produced by our system we compared it to an implementation of the Continuum Crowds method. We quantitatively compared the performance of the two controllers using the environments shown in Figure 3. In each case there is

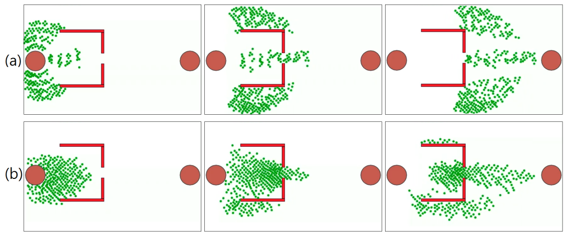

Figure 5: The results of running our system (a) and Continuum crowds (b) on the first example from Figure 3.

a single start point on the left and a single goal on the right. An example run for each system can be seen in Figure 5, here it can be seen that our system plans for the bottleneck in the centre, whereas Continuum crowds initially moves all of the agents towards it.

Figure 4 shows the result of this comparison across all examples. This graph is the average speed in the x direction (that is, the most direct path from the start to the goal) of every agent across every one of the 10 test cases. As can be seen, Continuum Crowds control invariably takes a greedy approach initially, with the agents moving directly to the goal causing a very fast initial x velocity but leading to a blockage and a much slower equilibrium. Our method provides a much faster equilibrium speed, this demonstrates that the agents are efficiently moving around the obstructions in a coordinated way to get towards the goal.

7 Discussions and Future Work

Previous methods of crowd control have tended to focus on localised realism and allowed the global situation to be an emergent process. Our system institutes control at this global level and as a result is capable of providing coordination for large crowds should the scene require it. It is also, because of its high level approach, completely compatible with most of the previous localised methods, as the guideline based paths could easily be followed by any such controller allowing the creation of different features or realism. Although scenes similar to those presented in this paper could be synthesised by other methods, they would require additional tedious tuning by the animator. Specifically, they would need to be carefully initialised to avoid overfilling the scene and all agents would need some form of assigned routes to avoid them all becoming congested in any bottlenecks. This process is all taken care of automatically by our system. In the future we will attempt to optimise the computation such that the entire planning could be recalculated during runtime, allowing agents to take account of congested areas and plan accordingly.

Acknowledgements

This work is partially supported by grants from EPSRC (EP/H012338/1) and EU FP7 TOMSY.

References

[1] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Trans. Graph.*, 25(3):1160–1168, July 2006.

[2] J. van den Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin. Interactive navigation of multiple agents in crowded environments. I3D ’08, pages 139–147.

[3] R. Silveira, L. Fischer, J. Ferreira, E. Prestes, and L. Nedel. Path-planning for rts games based on potential fields. MIG’10, pages 410–421.

[4] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des.*, 22(5):392–423, 2005.

[5] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. SIGGRAPH ’01, pages 203–212. ACM, 2001.

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.