Social Lane-Based Cognitive Model for Simulating Pedestrian Flow in Games

David J. Howden
Centre for Information Technology Research
Swinburne University of Technology
Melbourne, Australia
Email: http://www.davidhowden.org/contact

Stephen Y. Chen
School of Information Technology
York University
Toronto, Canada
Email: sychen@yorku.ca

Abstract—In computer game software, the implementation of simulated urban crowds is widespread. Representation of pedestrians in computer games has, to date, lagged behind what has been shown possible in academic studies and simulation software. Primary reasons for this are the strict CPU budgets that game AI has to function under, and algorithm implementation which is frequently complex for even a baseline approach (i.e. before performance optimisations). This paper presents a novel lane-based approach to pedestrian AI which combines the verisimilitude of AI based on behavioural studies, with the intuitive implementation and minimal computation cost required for games development. To achieve this, each agent filters its surroundings into virtual lanes, and then sidesteps or continues forward as dictated by social convention.

Keywords—Lane, pedestrian, game, embodiment, social, crowd, artificial life, simulation.

I. INTRODUCTION

The scene when standing back and observing the crowds flowing through the streets of a busy city is one of fluid chaos. When focus is brought to bear on any particular individual within the crowd however, despite a seemingly endless procession of imminent collisions, their stride never breaks and personal space remains inviolate. It is as if by tacit agreement the masses part to make way for this individual's benefit. Continue watching and patterns begin to emerge, unspoken rules which seem to guide interaction. This wordless coordination and aggregate motion of strangers, who only perceive each other for the briefest moments of time, is informally codified as social norms and has been the topic of numerous behavioural studies. As with any rule based system, simulation is possible.

In this work, we address the simulation of social rules to guide the movement of pedestrians on crowded footpaths specifically for use in computer games. The interactions of a crowd of pedestrians on crowded paths are complex, with numerous potential collisions to be avoided in any given moment. However, in real life, collisions are a rarity and crowd movement is relatively free flowing. The difficulty of producing this behaviour in computer games when crowd density is high is aggravated by the severe CPU budget that game AI must operate under.

In addition to budget constraints, crowd movement needs to be both efficient, in terms of avoiding congestion, but believable. This is a dual problem of creating responses which are plausible, while avoiding any behavioural artefacts which break the player's immersion by appearing unintelligent. In the context of simulating human crowds, verisimilitude (the appearance of being real) requires that actions are logical, without appearing too logical. As an example, logical behaviour is for pedestrians to walk on the same side of a footpath, while improving efficiency by maintaining single file would appear too logical. The aforementioned budget constraints are especially poignant when it comes to maintaining the illusion of intelligent planning. Actually performing predictive planning at an individual level is often prohibitively expensive.

To resolve these problems, we propose a novel new approach for pedestrian movement simulation which leverages sociological studies and the concept of lane-based movement. Our solution abstracts the environment and all agents within it to a series of virtual lanes with a limited number of states, and then reacts using a decision tree. The output of this process is simply to continue forward, or, depending on the circumstances, to sidestep left or right. The primary benefits of this algorithm over existing methods for pedestrian simulation are the appearance of predictive behaviour without actually performing prediction, the use of explicitly coded social rules, and a linear scaling of computational cost with very low overheads.

In the next section, contemporary approaches to crowd simulation are presented, with a focus on their applicability to implementation in games. Section III describes the algorithm's behavioural trees and environmental representation, then Section IV discusses the algorithm's required implementation framework and parameters. Performance of the algorithm is discussed in Section V in absolute and relative terms, followed by discussions and conclusions.

II. RELATED WORK

Previous work on the problem of pedestrian simulation has predominantly utilised force-based approaches. In a force-based approach, the movement of pedestrians is guided by the sum of force vectors such as repulsion from peers and attraction to waypoints. In more complex implementations, this behaviour can be augmented through creation of cognitive models, such as fitness landscapes where attraction is to areas of low density. In such approaches, repulsive forces serve to model social norms such as maintenance of personal space and not charging headlong into people.
These approaches are almost universally grounded, explicitly or otherwise, in the seminal crowd animation work of Craig Reynolds in 1987.\cite{Reynolds1987} Reynolds group movement simulation, known as Boids (a pun on bird-like-objects, and the word birds pronounced with a Boston accent), introduced the concept of guiding movement through the interaction of multiple applied forces (i.e. steering). A canonical steering algorithm for groups of animals uses three forces: attraction to the groups centre of mass, repulsion from adjacent peers and obstacles, and a force aligning each agent in the same direction as its neighbours. In Boids a flock of birds using this steering behaviour navigates through 3D space using reactive, deterministic logic while producing the appearance of randomness and planning.

Reynolds extended this work in 1999 with Steering Behaviours For Autonomous Characters.\cite{Reynolds1999} While the original Boids had the agent’s velocities directly modified by applied forces, this new work separated steering from the character’s locomotion. In the process, it enabled steering behaviours to be used for higher level goals such as “get from here to there while avoiding obstacles, follow this corridor, join that group of characters”.\cite{Reynolds1999} The decoupling of force and movement is achieved by using forces to calculate a desired velocity, then leaving it to the agent to attempt to match it. High level behaviours are chiefly achieved by selecting various points of attraction. To produce a “following” behaviour, an attraction point is placed on the agent being followed. To produce “wandering” behaviour, an attraction point is placed at randomly in a radius in front of the agent.

In the context of game AI, steering behaviours are appealing for two key reasons. Firstly they are general purpose, meaning at a minimum they will work adequately in almost any scenario involving agents that need to navigate an environment. Secondly, they are intuitive and easy to implement: this frees up time to tweak parameters to produce specific desired results. The drawback of steering is that computational overhead does not scale linearly, becoming problematic with large groups. In addition, steering works best when applied to groups such as schools of fish, and other environments with relatively unrestricted movement and limited vision. An urban environment is constricted in nature and steering behaviours can struggle to model social interactions specific to pedestrian movement, such as not overtaking if it would cause an incoming pedestrian inconvenience. In regards to overhead, steering behaviours were designed to be updated every time step, with a growth rate of $O(n^2)$ due to a need to iterate through every other agent in to determine neighbourhoods (the subset of closest agents), and then calculate various forces. Through use of various spatial optimisation techniques, this can be parallelised and reduced to $O(n)$, though the update frequency remains the same and the algorithms complexity with regards to implementation increases.\cite{Narain2010}

Steering behaviours work best in open environments, and difficulties exist in extending them to operate in constrained and ordered scenarios (such as pedestrian interaction). Crowd behaviour modelled as a collection of trajectories often results in the inability to anticipate future collisions from distant agents, primarily due to the limiting nature of using repulsion as a means of separation. Repulsion which is too strong, or operating at too great a distance, leads to agents which have difficulty reaching goals or even travelling in a straight line. Alternatively, repulsion which is too weak or close range results in paths which are too straight, right until last moment collision avoidance is mandated.\cite{Narain2010} While tweaking force parameters is sufficient to produce intelligent looking behaviour from a distance, this facade can still break down when individuals inside the population are observed.

There are a number of approaches which are still essentially steering based that can increase performance by encoding navigation information through the environment rather than at the agent level.\cite{Kallmann2010} These approaches often have good scalability as individual cognitive processes are kept to a minimum, but the resulting behaviours are often quite simplistic. An example of such approach is work by Mankyu Sung et al.,\cite{Sung2008} which extends the Smart Object work by Marcelo Kallmann et al.,\cite{Kallmann2008} where behaviours such as using road crossings or entering rooms are encoded into the environment itself. When the agent senses an interactive zone, there is a probabilistic chance that the agent will enter a state dictated by the object itself. The default behaviours is essentially random wandering, or weighted wander towards a distant goal. While interesting, this approach is more suited to groups which mill about and occupy a space rather than pedestrians purposefully travelling a shared pathway.\cite{Kallmann2010}

A novel approach by Rahul Narain et al. is to simulate crowds using fluid dynamics.\cite{Narain2010} Aimed at the massive crowds as could be found at music events or conventions, it is observed that major collision avoidance algorithms are a computational bottleneck when density increases. The use of fluid mechanics is an attempt to sidestep this issue while still maintaining plausible movement. Performance scales well, however noted limitations of this approach are the inability to deal with head on collisions until agents are already adjacent, lack of integration of social conventions, and behavioural artefacts when dealing with corners.

Interesting work has also been done in the combination of grid based movement planning and vector based collision avoidance.\cite{Musse2010} This approach combines spatial optimisations to increase performance with long range planning for goal assignment. Algorithm initialisation begins with overlaying the environment with a grid of cells, and then determining which cells contain static objects. This initial grid is used for mid-range collision avoidance by projecting a fan of lines to determine clear paths. Next, relevant cells are checked for mobile targets (e.g. other agents), and steering behaviours are performed to avoid collision. Finally, a path is planned through the grid between the agent and a goal. Actual navigation behaviours are quite complex, but they do result in interesting emergent behaviours such as queuing to pass through narrow corridors and dynamic lane formation. This approach is quite sophisticated by design, and is aimed towards simulation and artificial life. The sophistication of this algorithm, however, renders it largely unsuitable for computer games except as inspiration. Even without rendering, 100 agents is enough to bring performance down to 60 frames per second at 100% CPU usage.\cite{Musse2010}

The combination of goals and desires with pedestrian movement is a common theme in artificial life AI. Musse\cite{Musse2010} takes this theme to the level that agents have different walk animations depending on their emotional state. In his work pedestrians are modelled as groups which experience desires...
such as attraction to entertainment which can trigger events such as “partying”, and also sympathetic reactions such as fear and panic. As the focus is largely on the emotional state of the crowd, actual pedestrian movement is a straightforward steering implementation with all the benefits and drawbacks that entails.

A promising recent approach utilises potential fields,[13] an AI technique more commonly seen in tactical AIs for strategy games and swarm robotics. Potential fields (also known as influence maps) are a cognitive model constructed by overlaying the environment with a grid. Cells in the grid containing points of attraction are assigned positive values, which are propagated outwards in decreasing magnitude to create a gradient. Providing values are only propagated to passable terrain, gradient ascent from any point in the map will bring an agent to the origin cell. When combined with additions such as repulsion points, evaporation, moving attraction points, etc., more complex behaviour can be engineered.

Treuille et al.[13] use potential fields to combine global, goal based movement with partial local collision avoidance. Similar to the fluid dynamic model, the population flows through the environment, rather than reacting as individuals, allowing for good scalability. It however displays the same artefacts as other vector based methods which stem from local interactions. Agents which are pushed into collision courses spin and slide off each other, and agents which have been pushed into close formation tend to stay in these arrangements even once there is room to spread out.1 Finally, as path planning is optimal and globally calculated, the movement of crowds can appear too logical. In reality, individuals acting independently and rationally according to self-interest will often act in a way detrimental to the group. Treuille has stated that this last issue may be resolved in future work by limiting visibility.

A recent and fruitful branch of research in pedestrian simulation has been in the modelling of social factors.[14] Representing social conventions in control techniques has mutual advantages in both apparent realism and efficiency of movement. The focus has been on elements such as pedestrians walking in pairs,[15] including group cohesion in choke points,[16] and emergent lane forming. Lamarche’s work [17] is inspired by psychological studies on pedestrian behaviour, specifically that by Goffman [18] in classifying interactions into rules.

Goffman describes the four combined principles present in pedestrian collision avoidance: externalisation, scanning, minimisation of interaction, and security region.[18] Externalisation is the act of projecting your intentions for the purpose of allowing others to predict your trajectory without having to actually interact. Scanning is the technique of gathering the externalised information within a selective, local proximity. Minimisation of adjustment is technique by which pedestrians alter their trajectories in advance of it being necessary, with the purpose again of minimising interaction. The security region is defined as an oval around a pedestrian with width equal to the socially acceptable gap to pass a person or follow a wall, and a length equal to the distance that an oncoming pedestrian

1Local interactions are observable in the author’s presentation, available at: http://www.davidhowden.org/cec2013

should be reacted to. This continuous non-verbal communication between pedestrians, marked by apparent indifference, is known as civil inattention. Civil inattention is a process by which strangers demonstrate they are aware of the other’s presence, through passing eye contact or subtle modification of trajectory, without imposing on their privacy.[18]

With the group of algorithms which models social factors, the verisimilitude of pedestrian behaviour increases dramatically. In contrast, performance of these algorithms is often either low, or unstated. To be successful as a game AI, algorithms require both computational efficiency as well as believability. In addition, algorithms should be intuitive to understand and simple to implement, as evidenced by the dominance of steering behaviour inspired algorithms. The algorithm presented in this paper fulfils these three criteria with the introduction of a new technique, lane-based behaviour. A simple rule-based decision tree is combined with individual cognitive models to produce behaviour which is trivial to compute and still compliant with expected social norms.

III. Algorithm Framework

The proposed algorithm is primarily based on two concepts. Firstly, pedestrians perceive crowded footpaths as a series of lanes which can be either clear or blocked by another pedestrian. These lanes are virtual (centred on the individual’s current position) rather than a property of the footpath itself. The simulated pedestrian’s world exists only as a forward lane leading to its next waypoint, a lane to the left, and one to the right. When a pedestrian’s front lane is clear, they continue forward and no further action is required. If the front lane is blocked, then a decision tree is used to determine which lane (if any) to move into, creating a new “front” lane.

Fundamentally, the decision tree evaluates lanes as being blocked by obstructive pedestrians (pedestrians currently moving slower than the evaluating agent), and oncoming pedestrians. Oncoming pedestrians are reacted to by moving to the right, in accordance with road-law based social convention. Obstructive pedestrians (referred to as obstacles) are reacted to by moving into a clear lane if available, or slowing down if not. This fundamental behaviour is shown in Fig. 1.

Within this framework, externalisation (making others aware of your intentions) is handled under the principle of
embodiment. Embodiment is a swarm intelligence concept where an agent’s physical location in the environment communicates information. As the agents are controlled by the same deterministic algorithm, their reactions are predictable based on their relative position and heading.

Scanning, the collection of externalised information, is the process by which lane information is gathered, performed through a succession of up to three short range ray casts. Scanning occurs within the parameters defined by the security region which is effectively rectangular, though of the same dimensions as the Goffman model,[18] as illustrated in Fig. 2. The left and right bounds of this rectangle are still representative of the acceptable passing or following distance.

IV. IMPLEMENTATION DETAILS

The presented decision tree will result in relatively competent pedestrian movement, however to make it appear natural to a human observer an element of randomness is required. Often algorithms will implement the random wander steering behaviour to achieve this, however this almost invariably leads to unrealistic meandering walking which looks, at best, drunk. To produce natural looking pedestrian flow, pedestrians should subtly drift towards the middle of the footpath. This, along with the virtual lanes being centred on the individual pedestrian rather than the footpath itself, produces a staggered crowd and the appearance of heterogeneous behaviour.

This appearance of heterogeneous behaviour, as well as for the the important consideration of load balancing the AI CPU budget, is greatly aided by using relatively large update intervals with pedestrian behaviour. The exact period will be dependant on relative walking speeds used in-game; as a rule of thumb, a one second mean has been found to work well. By making the exact period to the next update random, the artefact of adjacent pedestrians consecutively reacting at the same time to stimulus can be avoided. This implementation detail is important as it enables behaviour which appears intelligent and imbues dumb-reactive AI with glimpses of personality. Examples of this are pedestrians that appear to stride confidently down the middle of a path forcing others out of their way, pedestrians that appear to be in a rush but can not quite get past a slow walker, or pedestrians that appear to be not paying attention to where they are walking who then swerve at the last moment to avoid a collision.

For navigation, the algorithm requires only a basic navigation mesh. Pedestrian movement can be as simple as moving towards a waypoint, then choosing a new one at random that does not double back. In games, as opposed to strict simulation, pedestrians only need to exist as long as they are in view, which is typically very brief so the requirement for cogent and logical long term planning is unnecessary.

For a population of virtual lanes by raycasting to be effective, pedestrian hitboxes are require to have a minimum width. To prevent pedestrians moving undetected between an agent’s forward-lane ray cast, and side-lane ray cast, hitboxes should be no narrower than the distance between these two lines. Intuitively, this hitbox size corresponds to the socially acceptable gap when passing.

Finally the use of ray casts should be optimised so they are only sent out as required by the decision tree, rather than populating all three lanes at once. The middle lane is always required, but if it is clear no others are needed. The left and right lanes can then be calculated as required.

V. PERFORMANCE

In this section, algorithm performance is measured both theoretically and through an implementation in Unity3D.[19] Unity3D is a relatively new engine which has been gaining popularity as a prototyping and development platform for small to medium size projects, and has recently gained substantial attention after a number of professional game studios chose it as the engine for their high profile Kickstarter campaigns.

In practice, algorithm performance is a function of update frequency and ray casts required. Outside of the ray casts, the only computation required is a trivial quantity of arithmetic and conditional evaluations. Assuming a staggered random update with a mean of one second, computations required are between one to three ray casts per agent per second. The minimum assumes that either the agent’s forward lane is clear, or the agent is on the far right of a footpath and the front lane is obstructed by an oncoming pedestrian. The maximum only occurs when an agent is not adjacent to any wall, its front lane is blocked by an obstructing (i.e. not oncoming) pedestrian, and its left lane is also blocked.

---

2 A navigation mesh (navmesh) is a group of nodes overlaid on an environment used for pathfinding. Typically they are arranged as a graph of waypoints with no intervening static objects, reducing the computation required for collision avoidance.

3 Kickstarter is a “crowd funding” website, where companies solicit donations based on design concepts.
Fig. 3. Emergent lane formation: simulation screenshot of the Unity3D prototype created for evaluating algorithm performance. Pedestrians are spawned at the ends of each footpath and travel to opposing waypoints.

The number of ray casts required (and thus, performance) is dependant on pedestrian density. Fig. 4 shows a graph of ray cast usage for various frequencies of pedestrians. The experiment was run first by simulating a single stream of pedestrians (overtaking only), and then by having two colliding streams on the same path, shown in Fig. 3. It can be seen that even with two streams with pedestrians being spawned at one second intervals, average ray cast usage is still only slightly over two per evaluation.

Fig. 5 shows the algorithm’s performance when operating at a full three ray casts per agent update, while Fig. 6 shows just the algorithm’s cost (in an unoptimised Unity3D implementation). Even when forcing maximum ray cast usage it can be see that until 1750 pedestrians are present the algorithm’s cost is nearly indistinguishable from background runtime fluctuations. Also of note is that until this point (1750 agents) performance scales linearly.

After this point, limitations in the Unity3D engine manifest. Unity3D is known to have issues with running a large number of physics objects or colliders, and this manifests in the performance graphs. This artefact of performance degrading at a rate disproportional to the increase in complexity can be seen more clearly in Fig. 7, where the program is run without the AI, but with uncapped object updates.

On a practical level, a Unity3D game rendering this many pedestrians will be bottlenecked by the renderer before this becomes a real issue. With modern games built using engines with a significantly higher degree of optimisation such as the Grand Theft Auto, Assassin’s Creed, or Sleeping Dogs, it would be unusual for even a hundred pedestrians to be rendered simultaneously. In usual-case scenarios the algorithm’s cost is essentially negligible, leaving perceived realism, and to a lesser extent implementation difficulty, as the key metrics.

VI. DISCUSSION AND FUTURE WORK

With AI for games, the most important consideration is not creating behaviours which appear intelligent, but avoiding behaviours which break immersion by appearing unintelligent. A classic example of this is the “good enough” AI that can handle most situations reasonably, with a collection of hard coded patches for specific scenarios which have been observed to be undesirable. This is at odds with most academic AI which puts a high value on general-purpose AIs which can handle a wide range of dynamic scenarios intelligently (preferably optimally), treating unrealistic artefacts in specific scenarios.
The main advantages of lane based pedestrian movement is verisimilitude without the expense of a complex decision making process. As mentioned previously, verisimilitude, the appearance of being true or real, is the second most valued aspect of a game AI, immediately after staying within processor budget. Verisimilitude is difficult to quantify, as it is an inherently subjective property. Perhaps the best method of evaluation is simple observation, which is possible via auxiliary videos available online.[20]

This behaviour is the result of the following rules and phenomenon which are a combination of explicitly encoded and emergent:

- Formation of lanes in heavy traffic
- Social norms for passing
- Social norms for overtaking
- Preservation of personal space in a natural manner (i.e. not repulsion based)
- Asymmetric reactions between homogeneous agents

The first three items set up the framework for agent behaviour, while the last two are what gives life to the scene. These last two are a result of the combination of lanes being a literal construct relative to the individual agent’s position, and updates being staggered leading to reactions occurring at different times, or only by one party. This is a fundamentally different proposition to adding randomness to velocity-based steering techniques. In adding randomness to the timing of the decision making, rather than the outcome of the decision, behaviour remains deterministic, which is beneficial for minimizing complexity. Predictable responses are what allows for collisions to be avoided without explicit prediction code and with minimal agent reactions.

A very common artefact of contemporary pedestrian AI is for agents to walk straight into each other, and then execute emergency collision avoidance to deal with it. This can becomes almost comical when more than two agents are
involved, as it involves agents spinning on the spot looking for a way past.

Limitations of this approach are that as it is specifically designed for metropolitan footpaths, behaviour is optimal when paths are straight and clear. Any obstacles or turns not marked by a waypoint are treated the same as slow moving pedestrians, in that they’ll be overtaken if there is a clear lane, or the pedestrian will slow to their speed (i.e. stationary) if not. In irregular environments, or environments without a navmesh, a traditional steering algorithm would be more appropriate.

Future work will include additional rules to handle an increased number of environments. Firstly, and most importantly, the algorithm requires rules for handling intersections. It is hypothesised that this will require modification to hitbox size and shape, and the categorization of perpendicular moving pedestrians as ‘oncoming’. An increase in agent heterogeneity will also be investigated through the addition of agent parameters such as distraction or confidence. In addition the range or likelihood that trajectory modification occurs at could be effectively randomised while still keeping the behaviour of individuals consistent. Handling of heterogeneous agents would also encompass the player’s character, or other pedestrians not bound by the game’s normal rules.

VII. Conclusion

This paper has described the novel new approach of lane-based behaviour for simulation pedestrian movement in computer games. Linear scaling and minimal computational overhead have been demonstrated in addition to simplicity of implementation. The approach is based on sociological studies, and it believably models social convention based decision making in crowds. It has been shown that the abstraction of a decision tree without stochastic elements which outputs the required virtual lane to turn into is sufficient to give the appearance of complex heterogeneous agents.

REFERENCES
