Egocentric Affordance Fields in Pedestrian Steering

Mubbasir Kapadia[*] Shawn Singh[†] Billy Hewlett[‡] Petros Faloutsos[§]
University of California, Los Angeles

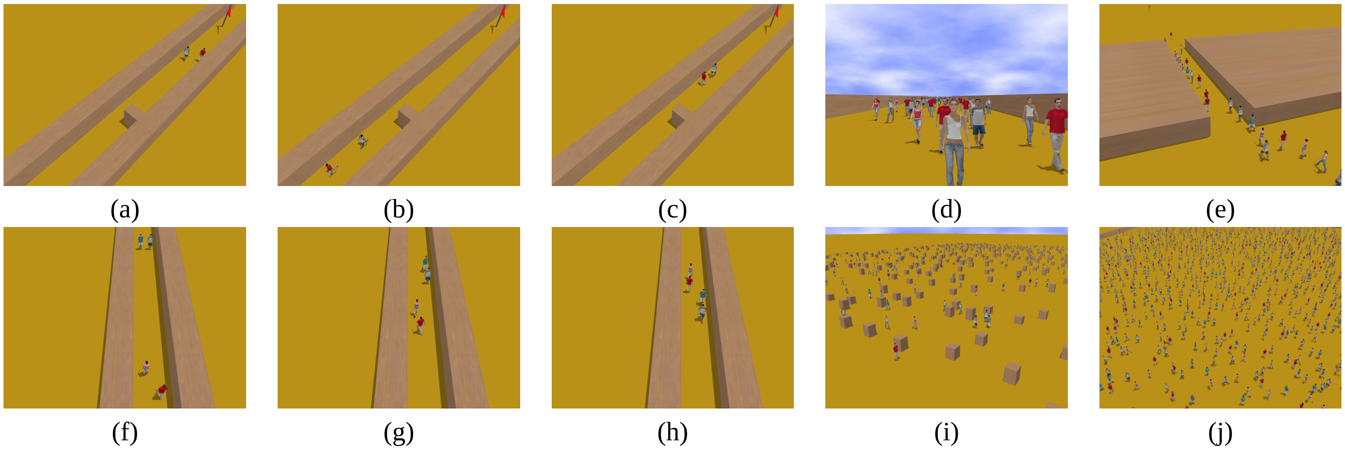

Figure 1: Figures (a)-(c): Demonstration of overtaking. Figure (d): Agents walking through a hallway. Figure (e): Queue formation as agents enter narrow passageway. Figures (f)-(h): Steering in confined environments. Figure (i): Forest. Figure (j): 5000 agent simulation.

Abstract

In this paper we propose a general framework for local path-planning and steering that can be easily extended to perform high-level behaviors. Our framework is based on the concept of *affordances* – the possible ways an agent can interact with its environment. Each agent perceives the environment through a set of vector and scalar fields that are represented in the agent's local space. This egocentric property allows us to efficiently compute a local space-time plan. We then use these perception fields to compute a "goodness" measure for every possible action, known as an affordance field. The action that has the optimal value in the affordance field is the agent's steering decision. Using our framework, we demonstrate completely autonomous virtual pedestrians that perform steering and path planning in unknown environments along with the emergence of cognitive responses to never seen before situations.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Intelligent Agents

Keywords: affordance, egocentric, steering, space-time planning

1 Introduction

Research in the area of pedestrian simulation has seen a dramatic rise in recent years. With the potential for this work being realized in a wide variety of areas, ranging from urban planning, training simulations and games, life-like steering motions for each individual have become critical for a truly immersive and realistic experience.

There are two major components involved in pedestrian navigation: path planning and a steering mechanism. They are often tackled as separate problems that need to be interfaced for a fully functional navigation system. Approaches such as A* [Hart et al. 1968] and potential fields [Warren 1990; Shimoda et al. 2005] are popular planning methods for pedestrian simulations. Given a target location and an obstacle laden environment, these techniques compute a global path to the target. Then, a steering mechanism (e.g. [Loscos et al. 2003; Pelechano et al. 2007]) tries to follow the planned path while avoiding dynamic objects.

However, an important feature of realistic steering is missing in traditional approaches: humans constantly compute a local short-term space-time plan to steer through their immediate environment which includes dynamic objects and other agents. This short-term plan is essential for *natural* steering in crowded environemnts, as well as for resolving deadlock situations, for example two people arriving at a doorway from opposite directions. Most agent-based

[*] email: mubbasir@cs.ucla.edu
[†] email: shawnsin@cs.ucla.edu
[‡] email: billyh@cs.ucla.edu
[§] email: pfal@cs.ucla.edu

approaches do not perform space-time planning. Recent works with space-time predictions (e.g. [Paris et al. 2007]) use an explicit time dimension and simple linear prediction. Continuum dynamics approaches (e.g., [Treuille et al. 2006]) do not model individual agent interactions that would use such short-term plans. Field-based approaches (e.g. [Shimoda et al. 2005]) effectively compute plans on the required scale, but do not take into account time and require storing large high-resolution fields for the entire environment.

This paper presents a novel technique that bridges the gap between steering and planning by using information fields in an egocentric fashion, where the agent is centered about the origin at all times. Our approach is based on the concept of *affordances* – the ways in which an agent can interact with its environment [Gibson 1977]. Affordances have been applied to agent systems [Michael and Chrysanthou 2003; Turner and Penn 2002], but not explicitly for steering. We define *goodness* to be a measure of how appropriate the associated affordance (associated action) will be. An *affordance field* is a scalar field that has a goodness measure associated with every affordance in the space of all possible actions. A final decision is the affordance associated with the optimal goodness measure.

This paper presents a novel egocentric fields steering framework with four main contributions:

- We represent the fields in an egocentric local-space, as opposed to a global world-space. This allows us to implicitly account for time when planning, and removes the resolution problem of global field-based approaches.
- We propose the concept of affordance fields as a powerful way to combine sensory information, giving more meaningful data than simple potential fields.
- Our discretized model has variable resolution, where information accuracy decreases with increase in distance from origin. This avoids wasteful computation and storage cost further away, where a plan would be re-computed sooner than it is used.
- Our approach performs short-term planning, accounting for dynamic objects, making it possible to steer naturally in challenging agent-agent interactions, such as deadlocks. Demonstration of these abilities can be seen in the accompanying supplementary video.

This paper is organized as follows: Section 2 provides a brief overview of the current state of the art in the field of crowd simulation. We present our egocentric information representation, with implicit time dependency and discuss the concept of affordance in Section 3. Section 4 describes the discretization of the egocentric fields and discusses our method of implementing steering and implicit path planning in dynamically changing environments. We then evaluate our approach using a suite of test cases and we illustrate ease of integration with higher level behaviors by demonstrating group behaviors (Section 5). Finally, Section 7 discusses future work.

2 Related Work

Since the seminal work of [Reynolds 1987; Reynolds 1999], there has been a growing interest in crowd simulation with a wide array of techniques being tested and implemented. [Clements and Hughes 2004] and [Treuille et al. 2006] utilize fluid dynamics in producing high density crowd simulations. [Millan and Rudomin 2006] render large crowds by using existing programmable graphics hardware. These approaches generate smooth crowd locomotion at the expense of losing detailed behavior of each individual. [Sung et al. 2004] adopts an agent model where behaviors are added depending on the environment situation. Rule based systems [Loscos et al. 2003; Lamarche and Donikian 2004] limit the steering functionality to conditions that have been foreseen and do not react well to irregularities in the environment that inevitably arise in a high density crowd simulation. Example based approaches [Lerner et al. 2007] use video segments of real pedestrian crowds as input to a learning system that generates natural looking crowd behaviors. [Boulic 2008] offers the ability of reaching a goal with a prescribed direction by extending the funneling behavior. [Funge et al. 1999; Shao and Terzopoulos 2005; Yu and Terzopoulos 2007] focus on the cognitive abilities of autonomous pedestrians.

There are two popular methods to path planning in crowd simulation. The A* algorithm and its derivatives [Hart et al. 1968; Hart et al. 1972; Dechter and Pearl 1985; Trovato and Dorst 2002] tend to produce non-realistic routes and require smoothing techniques in addition to a steering mechanism for dynamic collision avoidance. The approach of potential fields [Warren 1989; Warren 1990; Shimoda et al. 2005] generates a global field for the entire landscape where the potential gradient is contingent upon the presence of obstacles and distance to goal. Since a change in target or environment requires significant re-computation, these navigation methods are generally confined to systems with non-changing goals and static environments. A solution is proposed by [Surasmith 2002] in which path-finding data is pre-computed and stored in a connectivity table which offsets this issue at the cost of a considerable memory overhead. [Li et al. 2003] uses a segregated local and global planner to perform path planning in a layered environment. The work of [Takeuchi et al. 1992] employs the notion of attractiveness of objects for path planning and its use in human animation.

Space-time models (e.g., [Kant and Zucker 1988; Tsubouchi et al. 1995; Shapiro et al. 2007]) combine space and time into a single construct by representing space as three dimensions and time as the fourth dimension. Space time planning exploits the inherent advantage of having information in time to predict collisions in the future. These models improve steering behaviors at the cost of an additional dimension which greatly increases the search space, incurring a considerable overhead.

There has been previous work in the realm of egocentric based navigation [Altun and Koku 2005; Chao and Dyer 1999; Fleming 2005]. These techniques use egocentric maps for simple static obstacle avoidance but do not address the issue of larger environments where the goal falls outside the local egocentric map. Moreover, this egocentric model does not address space-time planning for dynamic object avoidance or real-time performance.

In most crowd simulation approaches, the underlying steering framework offers parameters that can be used by a higher lever framework to implement group behaviors. For example, in force based approaches, properly designed attractive forces can keep a group of people together or make them follow a leader. Our framework allows group behaviors to be implemented by simply setting intermediate dynamic or static goals for the agents.

Comparison to previous work. The most similar works to ours are potential field methods [Warren 1989], affordance methods [Michael and Chrysanthou 2003; Turner and Penn 2002], and continuum crowds [Treuille et al. 2006]. The key difference between these works and ours is that our affordance fields are represented in an *egocentric* manner, giving us several benefits: (1) we are no longer bound to the resolution and scaling problems associated with global fields and continuum methods. (2) We can dynamically scale the resolution of our egocentric fields to get the highest resolution possible for the scale we need. (3) The computational cost of our approach is not dependent on the complexity of the environment. (4) Our approach naturally supports efficient space-time planning, which is difficult to integrate into global fields and con-

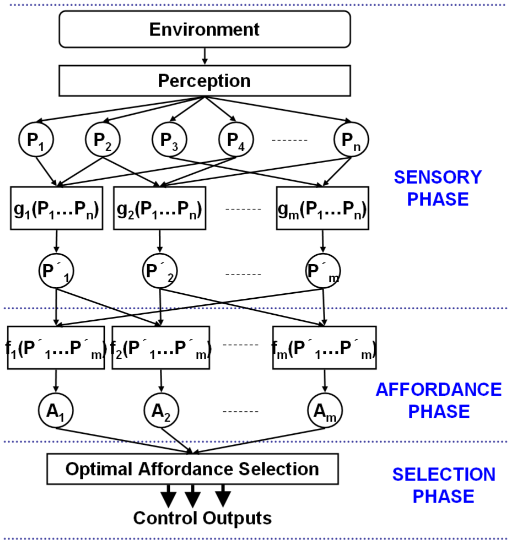

Figure 2: Overview of proposed framework and data flow using egocentric fields. P = Perception Fields, A = Affordance Fields.

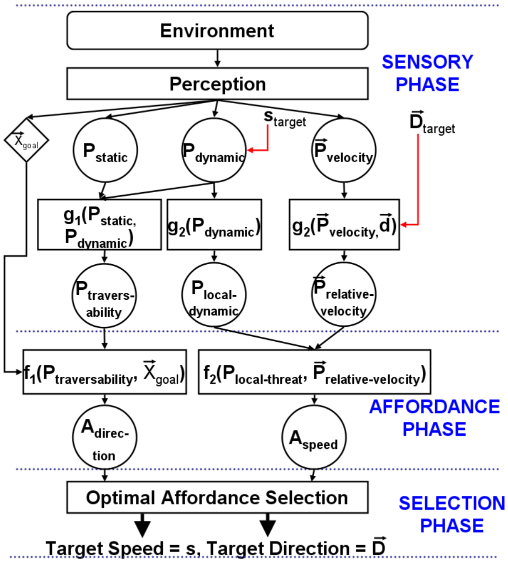

Figure 3: Data flow diagram for steering using egocentric fields.

tinuum methods. To our knowledge, our work is the first to present an egocentric fields approach to model affordances for steering.

3 Perception and Affordance Fields

In this section, we present a novel theoretical basis of information representation for steering. An overview of our proposed framework is illustrated in Figure 2 . There are two key elements in our model. First, the data is represented using *egocentric fields*, where the origin is always the center of focus. Second, we use the concept of *affordance*, described below, which is a convenient and powerful way to combine all sensory information. The model consists of three phases, which are described below.

3.1 Sensory Phase – Egocentric Perception Fields

An *egocentric perception field*, $P(\vec{X})$, is a vector or scalar field that quantifies a property of the environment. For example, a *traversability field* quantifies how easy it is to occupy a location in space – high traversability value implies that it is easy for an agent to occupy that location, while a low traversability value implies that an agent would not be able to occupy that location, perhaps because another object already occupies that location. Other examples of perception fields are: velocity information of nearby objects and planned trajectories of other agents. Egocentric perception fields can be computed from sensors for a robot or by querying the environment data structures, e.g., in a virtual world.

Time is naturally taken into account in this model, because of the egocentric representation. The agent is always located at the origin, and therefore the shortest distance between any point and the origin is directly proportional to the time it would take to reach that point. We use this property to efficiently predict collisions and plan in the *space-time* domain without requiring an explicit additional dimension in the system.

Perception fields are combined to provide a more intuitive representation of sensory information which can then be used to compute the affordance fields,

$$P'(\vec{X}) = g(P_1, P_2, P_3, ...P_n, goal) \tag{1}$$

where $g(\cdot)$ is a function of one or more perception fields. For example, a linear combination of static and dynamic perceptual information provides information of traversability in the environment.

3.2 Affordance Phase – Affordance Fields

The concept of affordance was introduced by Gibson in 1954 [Greeno 1994]. In our context, affordances describe the various ways that an agent can interact with its environment. Specifically, we define an affordance as a possible steering action that an agent could perform at a given point in time.

An *affordance field*, $A(q)$, quantifies the relative "strength" of all affordances, based on the desired goals of the agent. An affordance value, $A(q_i)$ for a particular action q_i is an evaluation of how much this action would help advance the agent towards accomplishing its goals. It is computed as a function of perception fields:

$$A(q) = f(P'_1, P'_2, P'_3, ...P'_m) \tag{2}$$

where the function $f(\cdot)$ is defined so that $A(q)$ provides a numeric value indicating the strength of a particular affordance. The specific functions that we use for our implementation of steering are described in Section 4.

3.3 Selection Phase

The final output of the system is the affordance (action) q_i associated with the optimal value $A(q_i)$. Optimality is defined by maximizing or minimizing $f(\cdot)$. Example of output decisions are target speed and desired direction of an agent.

So far our discussion has been in continuous space. Section 4 presents a discretization of these fields and applies this discrete model to steering.

4 Discrete Egocentric Fields for Steering

This section describes the specific egocentric fields used for steering, and the steering algorithm itself. A discretization of egocentric fields is described in Section 4.1. Sensory information, such as traversability, dynamic threats and velocity of neighboring agents, is represented using egocentric perception fields (Section 4.2). Affordance fields are computed as a function of these perception fields which provide the relative strength of all possible steering decisions, based on the goal(s) of the agent (Section 4.3). The final output decisions, in the form of target speed, s_{target} and target direction, $\vec{D}_{\text{target}}$ are used for locomotion of agent (Section 4.4). The data-flow diagram for steering is illustrated in Figure 3.

4.1 Discretization of Egocentric Fields

We implement a discretization of the model developed in Section 3 as a connectionist architecture that uses nodes arranged in concentric circles to maintain egocentric spatial information. At all times, the central node represents the current position of the agent. Each node perceives information corresponding to its "spatial awareness" in the environment.

Discrete Egocentric Fields comprise the following structural components, shown in Figure 4:

- **Root:** The *root* represents the current position of the agent and is the origin of the egocentric fields.
- **Layers:** The egocentric map is segregated into layers, denoted by the layer number l, where each layer comprises a fixed number of nodes that store the information of an area of the environment. The number of nodes per layer n and the number of layers m are the two user-defined parameters.
- **Layer Radius:** The distance from the root to the l^{th} layer is known as the *layer radius*, denoted as $r_{\text{layer}}(l)$.
- **Node Radius:** The radius of the area associated with node of the l^{th} layer is termed as the *node radius*, $r_{\text{node}}(l)$. The node radius increases for layers further from the root. As a result, the spatial area covered by the node increases with increase in distance from the root, giving rise to variable resolution.
- **Inter-node weight:** The *inter-node weight*, $w(l)$, determines the area between two adjacent nodes in successive layers. It allows us to dynamically scale the coverage of the environment, for a constant memory cost.
- **Node Information:** Each node contains its location, connectivity to neighboring nodes, and values of the perception and affordance fields for its given location.

Variable Resolution and Dynamic Scaling

Our discrete egocentric perception fields are modeled with variable resolution where the accuracy of information is high near the root and decreases further from the origin. The layer radius is thus a determining factor of the relative importance of the information. The information storage per unit area is dense close to the origin, and density decreases further from the root.

Accuracy is dependent on the layer radius, node radius, and the inter-node weight, which in turn are determined by user-defined parameters: number of nodes per layer, n, and number of layers, m. Appendix A demonstrates how the structural components are determined from these two user-defined parameters.

To dynamically scale the field at runtime, we use inter-node weights, $w[l]$, to scale the field with respect to the distance of goal, D. Let D' be the radius of the outermost layer. If $D > D'$, the goal lies outside of the field. In this case, we iteratively increase $w(l)$ to scale the field until the goal lies inside the field. Thus, the scaled node radius $r'_{\text{node}}(l)$ is,

$$r'_{\text{node}}(l) = r_{\text{node}}(l) \times w(l) \qquad (3)$$

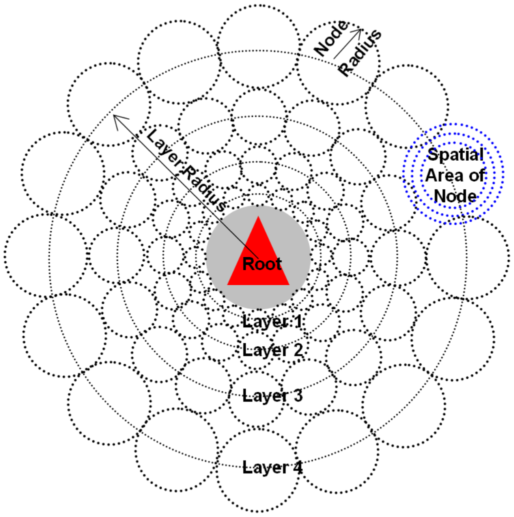

Figure 4: Structure of Egocentric Fields with variable resolution information representation.

4.2 Sensory Phase – Perception Fields

Our steering uses the following egocentric perception fields:

- **Static Field**: The static field, $P_{\text{static}}(\vec{X})$, represents the configuration of the obstacles in the environment surrounding the agent. A high numeric value of the field for a given location indicates that the location is free from static obstacles, whereas a low value indicates that it cannot be traversed.
- **Dynamic Field**: The dynamic field, $P_{\text{dynamic}}(\vec{X})$ represents the current state of all dynamic objects, by providing the predicted positions of neighboring agents at various points of time. A high numeric value indicates that the likelihood of a dynamic threat at the given location is minimal, whereas a low value indicates a high probability of a dynamic threat.
- **Velocity Field**: The velocity field, $\vec{P}_{\text{velocity}}(\vec{X})$ is a vector field that provides the direction and speed magnitude of neighboring agents.
- **Traversability Field**: $P_{\text{traversability}}(\vec{X})$, the traversability field, is a combination of P_{static} and P_{dynamic}:

$$P_{\text{traversability}}(\vec{X}) = P_{\text{static}}(\vec{X}) + P_{\text{dynamic}}(\vec{X}) \qquad (4)$$

- **Local Dynamic Field**: The dynamic field, $P_{\text{dynamic}}(\vec{X})$ is subject to a kernel function that considers the regions in which dynamic threats are most imminent. The resulting fields are known as local dynamic fields, denoted by $P_{\text{local-dynamic}}$.

$$P_{\text{local-dynamic}}(\vec{X}) = K(\vec{X}) \cdot P_{\text{dynamic}}(\vec{X}) \qquad (5)$$

Our current implementation uses a simple step function which only considers the information in the first $m/2$ layers of the dynamic field, where m is the total number of layers :

$$K(\vec{X}) = \begin{cases} 1 \text{ if } |\vec{X}| < r_{\text{layer}}(m/2) \\ 0 \text{ otherwise} \end{cases} \qquad (6)$$

- **Relative Velocity Field**: The relative velocity field, $\vec{P}_{\text{relative-velocity}}(\vec{X})$ provides the relative velocity of neighboring agents with respect to the agents velocity.

$$\vec{P}_{\text{relative-velocity}}(\vec{X}) = \vec{P}_{\text{velocity}}(\vec{X}) - (s_{\text{target}} \times \vec{D}_{\text{target}}) \quad (7)$$

where s_{target} is the speed with which the agent is traveling and $\vec{D}_{\text{target}}$ is its current direction of motion.

Implicit Space-Time Planning

Previous approaches implement space-time planning by representing space in two or three spatial dimensions with time as an additional dimension. This incurs a considerable processing overhead which becomes intractable in large crowd simulations. Our approach naturally supports efficient space-time planning by using egocentric fields. Due to the egocentric nature of our data representation, we can represent time implicitly as the distance from the origin to any point of interest, effectively reducing the dimensions in space-time planning by one. The remainder of the section describes how we apply this concept to perform efficient space-time planning and threat prediction simultaneously.

There exists a mapping between time and a particular layer of an agents egocentric field, as time is directly proportional to the distance from the origin of the field. Let $t[l]$ be the time taken by the agent to travel a distance $r_{\text{layer}}(l)$, for a particular layer, l. P_{dynamic} and $\vec{P}_{\text{velocity}}$ are computed by considering this *time-level associativity* and the neighbors, N, surrounding an agent . If the difference in the time taken by the neighbor $N(i)$ in travelling a distance $r_{\text{layer}}(j)$ and the time taken by the agent to travel a distance $r_{\text{layer}}(k)$ is below a certain threshold, ϵ, then a dynamic threat is predicted at that instance of time and space. P_{dynamic} of the agent reflects a potential threat at the predicted position of $N(i)$, at time $t[k]$. $\vec{P}_{\text{velocity}}$ stores the current velocity of the potential threats at that point in space. Once these fields are computed, $P_{\text{local-dynamic}}$ and $\vec{P}_{\text{relative-velocity}}$ are determined using Equations 5 and 7.

4.3 Affordance Phase – Affordance Fields

Affordance fields for direction and speed of locomotion are defined as follows:

Direction Affordance Fields

Direction Affordance Fields, $A_{\text{direction}}(\theta)$, quantify the relative strengths of all possible directions, θ in which an agent can steer. A pedestrian in a crowd bases its direction of travel on the presence of static objects in the environment as well as other pedestrians (dynamic objects). For instance, a slow moving pedestrian in front would mandate a direction change in order to perform an overtaking maneuver. This is an emergent behavior in our model by using the following formulation:

$$A_{\text{direction}}(\theta) = f_1(P_{\text{traversability}}, \vec{X}_{\text{goal}}) \tag{8}$$

where f_1 is an iterative process on P_{spatial}. The process starts by adding a strong goodness measure at the goal position and then propagating this value in all directions. The propagation at each point in space is affected by the spatial perception fields. For example, if there is an untraversable object between the agent and its goal, the value of goodness will not propagate through the object. Instead the goodness measure will eventually reach the agent by propagating around the object.

Given a 3D location $\vec{X_1}$, which is initally set to $\vec{X}_{\text{goal}}$, a set of points at an infinitely small displacement of Δr in all directions around $\vec{X_1}$ can be represented by the following function, $a(\vec{X})$:

$$a(\vec{X}) = \{\vec{X_2} : |\vec{X} - \vec{X_1}|^2 = \Delta r^2, \vec{X} \in R^3\} \tag{9}$$

The goodness measure propagates from point $\vec{X_1}$ to point $\vec{X_2}$ according to the following recurrence:

$$A_{\text{spatial}}(\vec{X_2}) = (A_{\text{spatial}}(\vec{X_1}) - P_{\text{traversability}}(\vec{X_2})) \times \alpha \tag{10}$$

where $\alpha \in (0, 1)$ is the rate of decay. The end result of this process is a path of high goodness from $\vec{X}_{\text{origin}}$ to $\vec{X}_{\text{goal}}$ that represents the path that must be traversed to reach the goal.

The spatial affordance field, $A_{\text{spatial}}(\vec{X})$ provides goodness values for all points in space. However, we require the goodness measure for all possible directions which serve as our steering choices. The goodness values for all points in space immediately surrounding us serve as the values for direction affordance, $A_{\text{direction}}(\theta)$.

Speed Affordance Fields

The speed of an agent is based on the velocity of neighboring agents that are perceived as threats. The goodness of speed is governed by the the presence of threats along the previously predicted path to the goal. The speed affordance field, $A_{\text{speed}}(s)$ provides the relative strength of all speed affordances.

$$A_{\text{speed}}(s) = \arg\min_{\vec{X}}(\vec{X}_{\text{i}} + t \times P_{\text{relative-velocity}}(\vec{X})) \tag{11}$$

where $\vec{X}_{\text{i}} \in \{\vec{X} : P_{\text{local-dynamic}} > 0\}$

4.4 Selection Phase – Optimal Affordance Selection

Once the affordance fields are computed, the final step is to select the speed and direction having optimal goodness values. The target speed, s_{target} is the speed for which, $A_{\text{speed}}(s)$ is maximized and is computed as follows:

$$s_{\text{target}} = \arg\max_{\text{s}} A_{\text{speed}}(s) \tag{12}$$

The target direction, $\vec{D}_{\text{target}}$ is estimated by rotating the current direction by an angle, θ_{target} which is computed as follows:

$$\theta_{\text{target}} = \arg\max_{\theta} A_{\text{direction}}(\theta) \tag{13}$$

In the discretized implementation of our model, selecting the direction having maximum goodness produces *vibrations* which arise due to the fact that the goodness of adjacent directions may oscillate over key frames. We offset this undesirable effect by performing quadratic interpolation over a window of directions whose cumulative goodness is maximized. Let $Y(\theta) = A\theta^2 + B\theta + C$ be a quadratic equation that maps goodness values to angular displacements. The value of θ for which $Y(\theta)$ is maximized is simply given by $\theta_{\text{target}} = -B/2A$.

4.5 The Steering Algorithm

The complete algorithm for steering using egocentric fields is outlined below:

- Determine goal position of an agent, $\vec{X}_{\text{goal}}$.
- Initialize node weights, $w(l) = 1, \forall l$. If necessary, iteratively increase $w(l)$ to scale the field until the goal lies inside the field.
- Populate the static perception field, P_{static}.

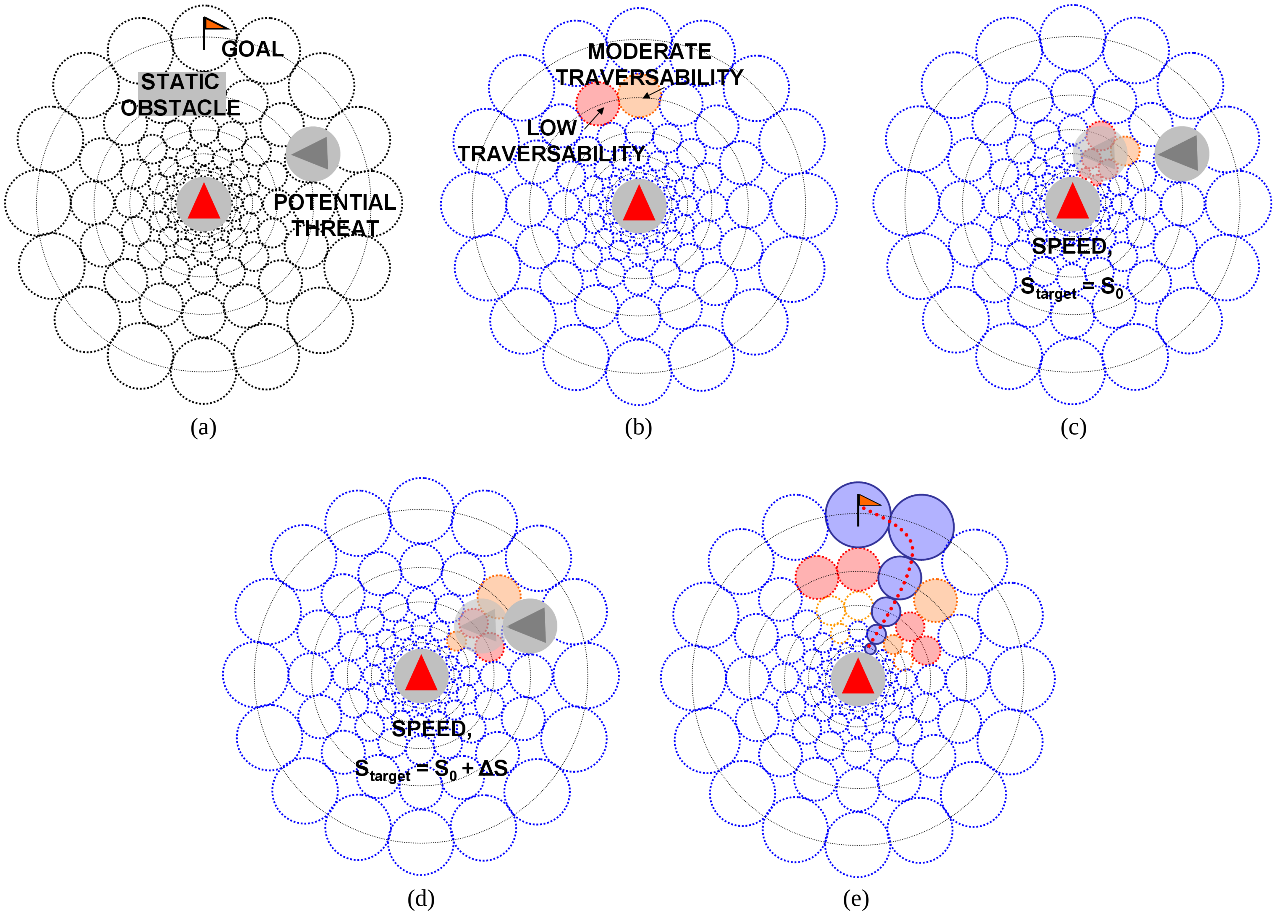

Figure 5: The Steering Algorithm: (a) The current state of the environment. (b) Static Perception Field indicating low traversability at position of obstacle. (c) Dynamic Perception Field for velocity, $s_{\text{target}} = s_0$. (d) Dynamic Perception Field for velocity, $s_{\text{target}} = s_0 + \Delta V$ (e) Affordance Field indicating path of high goodness to goal.

- Estimate time-layer associativity, $t[i] = 1$ to m, the number of layers for all i. $t[i]$ is the predicted time taken by the agent to travel a distance $r_{\text{layer}}[i]$, corresponding to layer i, at the current speed.
- Populate dynamic threat perception field, P_{dynamic}, at the current speed.
- Populate local dynamic threat perception field, $P_{\text{local-dynamic}}$ using Equation 5.
- Populate velocity perception field, $\vec{P}_{\text{velocity}}$.
- Populate relative velocity perception field, $\vec{P}_{\text{relative-velocity}}$ using Equation 7.
- Generate speed affordance fields, $A_{\text{speed}}(s) = f_2(P_{\text{local-threat}}, \vec{P}_{\text{relative-velocity}}))$.
- Compute new value of target speed, s_{target} which maximizes the goodness of the speed affordance. Refer to Equation 12.
- Re-estimate time-layer associativity, dynamic fields and velocity fields at new target speed.
- Generate direction affordance fields, $A_{\text{direction}}(\theta) = f_1(P_{\text{static}}, P_{\text{dynamic}})$.
- Compute target direction, $\vec{D}_{\text{target}}$ (Section 4.4).

The above mentioned steps are executed at every time step for each agent. s_{target} and $\vec{D}_{\text{target}}$ are the steering decisions made by our system.

5 Evaluation

We evaluate our system by testing it against a suite of scenarios ([Singh et al. 2008]) that are frequently encountered in pedestrian crowds. These scenarios range from basic cases, testing the ability of agents to handle oncoming and crossing threats, to large scale cases which stress test our system in the presence of a large number of agents. Section 5.1 outlines the suite of cases that we tested our system with. Section 5.2 discusses the performance of our system and presents interesting results. Finally, Section 5.3 demonstrates a variety of group behaviors, illustrating ease of integration of our system with higher level behaviors.

5.1 Scenarios

Similar direction: Agents traveling in similar directions, with slightly differing goals.
Crossing threats: Agents crossing paths, at various angles, in the presence of obstacles.
Oncoming threats: Agents traveling in opposite directions, with a potential for head-on collisions, with obstacles in the way.
Curves: Agents having to travel along a curved path to avoid obstacles.
Group-interactions: Agents traveling in groups, with other agents cutting across.
Squeeze: 2-4 agents, passing through a narrow hallway, with same

or opposite directions (Figure 1).
Doorway: Agents having to pass through a narrow doorway. (Figure 1 (f)-(h))
Overtake: An agent, encountering a slower moving agent in front, while traveling through a narrow passageway. (Figure 1 (a)-(c))
Confusion: Agents traveling in opposite directions, arriving at the same place, at approximately the same time.
Random: A large number of agents, with random goals. (Figure 1 (j))
Forest: A large number of agents, with random goals, in an obstacle laden environment. (Figure 1 (i))
Urban: A large number of agents, with random goals, in an environment with large obstacles, resembling large buildings.

5.2 Discussion

Here, we discuss our results of our framework on the scenarios described above (Section 5.1). These and additional results are shown in the supplementary video.

- *Local Agent Interactions.* Agents steer naturally around each other, with and without obstacles. This is shown in all scenarios, particularly the Crossing, Oncoming, Confusion, and Curves scenarios.

- *Human-like Behaviors.* Natural reactions are also captured by our framework. For example in the Surprise scenarios (with sharp turns), agents do not see each other until the collision is imminent. In such cases, behavior is affected by each individual's visual field. Macro-scale crowd simulations with global knowledge cannot model this. Our framework models this individuality successfully.

- *Implicit Space-time Planning.* The importance of space-time planning is demonstrated in doorway, overtake, confusion, and squeeze (narrow passage) scenarios. Comparisons of behaviors with and without implicit space-time for the 3-way confusion and overtaking scenarios show that the natural, anticipatory behaviors are a result of our implicit space- time planning. In general, we observe that space-time planning is essential for complicated interactions involving 3 or more agents.

- *Crowd Behaviors.* The video shows several bottleneck and densely crowded scenarios where agents cooperatively wait or steer around each other. Note that many previous approaches steer unnaturally into each other and into obstacles, relying on collision resolution and greedy steering to progress the agents. We do not explicitly prevent agents from colliding and overlapping; collision avoidance is purely a result of our steering algorithm.

Visualizing Results. Robustly animating the agent's locomotion is orthogonal to the focus of this paper. The animations in the supplementary video are produced by retrofitting walking and idling animations to the resulting agent paths. Agents use a walking animation when going faster than some user-specified speed threshold, and agents use an idle animation when below this threshold. We also scale the animation speed with the speed of the pedestrian. Artifacts in the video (unnatural large steps or abrupt transitions between animations when agents move slowly) are a result of this simple animation technique, not because of the steering algorithm.

Performance and Memory Consideration. Each agent has associated with it a set of fields that serve as its memory repository. We observe that a field with 8 layers and 16 nodes per layer is sufficient to perform effective steering in a virtual environment. It takes 2.5 - 3 KB of memory per agent to store the information of these fields. This is about 3 MB of memory per 1000 agents, which we believe to be a manageable overhead that increases linearly with increase in crowd density. The stress test cases, discussed earlier, are evaluated by varying the number of agents from 100 to 5000 on a 2.66 GHz Core 2 processor (on a single thread). The time of update per agent for each of these scenarios is outlined in Figure 6. We observe that the computation time linearly increases with increase in number of agents, and is independentof the complexity of the environment.

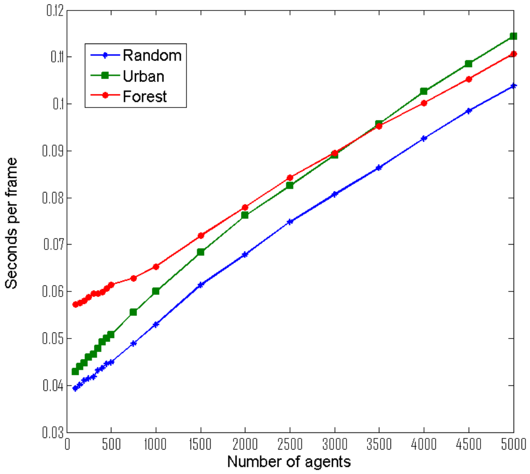

Figure 6: Time of update per agent for stress cases. Number of nodes per layer, n = 16. Number of layers, m = 8.

5.3 Group Behaviors

By adding a simple high-level layer which decides intermediate goals for each agent, our steering framework can perform common group behaviors. The intermediate goals can be dynamic (e.g. other agents) or static (e.g. location in space). To implement the group behaviors demonstrated in the video, agents automatically choose a dynamic goal to be the closest agent in front of itself. The following examples of group behaviors are demonstrated in the video:

Lane Formation and Queueing: When several agents are given the same goal, agents with no-one in front simply steer towards the goal. Agents with others in front begin to follow the agents immediately in front. The strictness of the lane is defined by a 'comfort distance' between agents. As the agents near the goal, they 'queue' up politely. (Figure 1 (e))
Snake Motion: We demonstrate snake-like motion by having a leader weaving around a set of obstacles, and each previous agent follows the next one.
Group persistence: Persistence of groups is demonstrated in the Oncoming-Groups scenario. Agents perceive the oncoming group as a single entity, because of the variable resolution fields. As a result, the two groups steer around each other.

Additional behaviors can be implemented by varying parameters (e.g., desired speed) or adding fields to the framework. For example, an additional field could be defined based on social constraints, such as "prefer to stay on the sidewalk" or "avoid a scary group of people". Aggressive and polite behaviors, such as agents being pushy or patient, can be modeled as exerting untraversability onto the environment. For example, a pushy person would exert untraversability of his future position onto other agent's fields, while a timid person would exert untraversability of other agents onto his own field.

6 Conclusion and Future Work

In this paper, we present a view-point dependent model of information representation, known as egocentric fields, having implicit time dependancy. Next, we introduce the concepts of affordance fields which quantify the relative strengths of all possible actions. An action is selected which has optimal affordance value. We present a discretized implementation of the above mentioned model and illustrate its use in crowd simulation. Some of the features of our system include variable resolution, dynamic scaling, and implicit path planning within the bounds of the field. Finally, we evaluate our system against a wide variety of test cases, including higher-level group behaviors.

The main focus of our work is in the realm of pedestrian simulation which is largely based in two dimensions. As illustrated in Section 3, our generic model is inherently in three dimensions, representing space with an implicit time dimension. In the future, we aim to implement a fully functional discrete implementation of a variable resolution 3-D model of the environment that may find use in a wide variety of applications not limited to steering.

Our current model performs implicit path planning only within the confines of area covered by the field. Also, information detail is lost at locations farther away, which may be essential for an agent to recognize narrow passageways that are located at a distance. We propose a hierarchy of fields as an extension to our model, whereby each node can be equipped with a sub-egocentric field of the area it encompasses. This would increase the resolution of areas along the path that are further away.

7 Acknowledgements

We wish to thank the anonymous reviewers for their comments. The work in this paper was partially supported by NSF grant No. CCF-0429983. We thank Intel Corp., Microsoft Corp., and AMD/ATI Corp.for their generous support through equipment and software grants.

A Derivation of User-defined Parameters

The layer-radius $r_{\text{layer}}(l)$, node-radius $r_{\text{node}}(l)$, and inter-node weight $w(l)$ all contribute to the variable resolution and the dynamic scaling of the egocentric fields. These structural components are dependent on the layer number l and the number of nodes per layer n. Note that n is a user-defined parameter.

The node radius is proportional to the circumference of the l^{th} layer. Thus,

$$r_{\text{layer}}(l) = r_{\text{node}}(l) \times \frac{n}{\pi}. \tag{14}$$

The first layer is at an offset of $r_{\text{node}}(0)$, outside the agent. The agent is modeled as a circle, with radius r_{agent}.

$$r_{\text{layer}}(0) = r_{\text{node}}(0) + r_{\text{agent}}. \tag{15}$$

From 14 and 15, the initial condition of $r_{\text{node}}(0)$ is

$$r_{\text{node}}(0) = \frac{\pi}{n-\pi} \times r_{\text{agent}}. \tag{16}$$

The node radius of the next layer increases to ensure the coverage of adjacent nodes following the geometry shown in Figure 4,

$$r_{\text{node}}(l+1) = \frac{n+\pi}{n-\pi} \times r_{\text{node}}(l). \tag{17}$$

Equations 14, 16 and 17 provide a method of estimating the node radii and the layer radii respectively. The user can thus specify the number of nodes per layer, n and the number of layers, m to control the size and resolution of the map.

References

ALTUN, K., AND KOKU, A. 2005. Evaluation of egocentric navigation methods. *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, 396–401.

BOULIC, R. 2008. Relaxed steering towards oriented region goals. In *Motion in Games, First International Workshop*, 176–187.

CHAO, G., AND DYER, M. 1999. Concentric spatial maps for neural network based navigation. *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470) 1*, 144–149 vol.1.

CLEMENTS, R. R., AND HUGHES, R. L. 2004. Mathematical modelling of a mediaeval battle: the battle of agincourt, 1415. *Math. Comput. Simul. 64*, 2, 259–269.

DECHTER, R., AND PEARL, J. 1985. Generalized best-first search strategies and the optimality af a*. *J. ACM 32*, 3, 505–536.

FLEMING, P. 2005. *Implementing a robust 3 dimensional egocentric navigation system.* Master's thesis, Graduate School of Vanderbilt University.

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 29–38.

GIBSON, J. J. 1977. *The Theory of Affordances. In Perceiving, Acting, and Knowing.*

GREENO, J. G. 1994. Gibson's affordances. *Psychological Review*, 336–342.

HART, P., NILSSON, N., AND RAPHAEL, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on 4*, 2, 100–107.

HART, P. E., NILSSON, N. J., AND RAPHAEL, B. 1972. Correction to "a formal basis for the heuristic determination of minimum cost paths". *SIGART Bull.*, 37, 28–29.

KANT, K., AND ZUCKER, S. W. 1988. Planning collision-free trajectories in time-varying environments: a two-level hierarchy. *The Visual Computer 3*, 5, 304–313.

LAMARCHE, F., AND DONIKIAN, S. 2004. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. In *Computer Graphics Forum 23.*

LERNER, A., CHRYSANTHOU, Y., AND LISCHINSKI, D. 2007. Crowds by example. *Computer Graphics Forum 26*, 3, 655–664.

LI, T.-Y., CHEN, P.-F., AND HUANG, P.-Z. 2003. Motion planning for humanoid walking in a layered environment. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on 3*, 3421–3427 vol.3.

LOSCOS, C., MARCHAL, D., AND MEYER, A. 2003. Intuitive crowd behavior in dense urban environments using local laws. *Theory and Practice of Computer Graphics, 2003. Proceedings*, 122–129.

MICHAEL, D., AND CHRYSANTHOU, Y. 2003. Automatic high level avatar guidance based on affordance of movement. In *Eurographics 2003*, Eurographics Association.

MILLAN, E., AND RUDOMIN, I. 2006. Impostors and pseudo-instancing for gpu crowd rendering. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, ACM, New York, NY, USA, 49–55.

PARIS, S., PETTRÉ, J., AND DONIKIAN, S. 2007. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *EUROGRAPHICS 2007*, vol. 26, 665–674.

PELECHANO, N., ALLBECK, J. M., AND BADLER, N. I. 2007. Controlling individual agents in high-density crowd simulation. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 99–108.

REYNOLDS, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 25–34.

REYNOLDS, C., 1999. Steering behaviors for autonomous characters.

SHAO, W., AND TERZOPOULOS, D. 2005. Autonomous pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, 19–28.

SHAPIRO, A., KALLMANN, M., AND FALOUTSOS, P. 2007. Interactive motion correction and object manipulation. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, ACM, New York, NY, USA, 137–144.

SHIMODA, S., KURODA, Y., AND IAGNEMMA, K. 2005. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2828–2833.

SINGH, S., NAIK, M., KAPADIA, M., FALOUTSOS, P., AND REINMAN, G. 2008. Watch out! a framework for evaluating steering behaviors. In *Motion in Games, First International Workshop*, 200–209.

SUNG, M., GLEICHER, M., AND CHENNEY, S. 2004. Scalable behaviors for crowd simulation. *Computer Graphics Forum 23*, 3, 519–528.

SURASMITH, S. 2002. Preprocessed solution for open terrain navigation. In *AI Game Programming Wisdom*, 161–170.

TAKEUCHI, R., UNUMA, M., AND AMAKAWA, K. 1992. Path planning and its application to human animation system. 163–175.

TREUILLE, A., COOPER, S., AND POPOVIĆ, Z. 2006. Continuum crowds. *ACM Trans. Graph. 25*, 3, 1160–1168.

TROVATO, K. I., AND DORST, L. 2002. Differential a*. *IEEE Trans. on Knowl. and Data Eng. 14*, 6, 1218–1229.

TSUBOUCHI, T., KURAMOCHI, S., AND ARIMOTO, S. 1995. Iterated forecast and planning algorithm to steer and drive a mobile robot in the presence of multiple moving objects. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems-Volume 2*, IEEE Computer Society, Washington, DC, USA, 2033.

TURNER, A., AND PENN, A. 2002. Encoding natural movement as an agent-based system: an investigation to human pedestrian behaviour in the built environment.

WARREN, C. 1989. Global path planning using artificial potential fields. *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 316–321 vol.1.

WARREN, C. 1990. Multiple robot path coordination using artificial potential fields. *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 500–505 vol.1.

YU, Q., AND TERZOPOULOS, D. 2007. A decision network framework for the behavioral animation of virtual humans. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 119–128.