A Generic Multi-Level Framework for Agent Navigation

Wouter van Toll* Utrecht University
Norman Jaklin† Utrecht University
Roland Geraerts‡ Utrecht University

1 Introduction

Path planning and crowd simulation are important tasks in simulations and games. Virtual characters (agents) need to autonomously find a path from their current position to a goal position. This is usually solved by performing A* search on a grid or a *navigation mesh*. However, strictly traversing a path is often not sufficient: agents need to avoid each other or react to changes in the environment. Multiple *levels* of planning are needed to achieve efficiency and realism. Many crowd simulation frameworks do not have this flexibility. We propose a generic five-level hierarchy for solving agent navigation problems, based on the work of Jaklin et al. [2013].

2 A Multi-level Planning Hierarchy

Figure 1 shows an overview of the hierarchy. Using a single agent as an example, the levels can be summarized as follows:

- *High-level planning* (5) translates semantic actions (e.g. 'go home') to geometric queries ('find a path from s to g').
- *Global planning* (4) computes an *indicative route*, i.e. a path from s to g that should be roughly followed. For instance, this can be a shortest route, a route that avoids dense areas, or a route that stays in attractive regions.
- The three lower levels update the agent in every step of the simulation loop. *Path following* (3) lets the agent choose a *preferred velocity* such that it follows the indicative route in a smooth manner. Note that a velocity is a 2D vector that encodes both *speed* and *direction*.
- *Local movement* (2) computes an actual velocity that is close to the preferred velocity but deals with the agent's current situation, e.g. by avoiding collisions with other agents or coordinating with the local crowd flow. The simulation then applies the chosen velocity through time integration.
- Finally, *animation* (1) handles the visual movement of the agent, down to its 3D skeleton representation. The visualization framerate is often much higher than the crowd simulation framerate, which typically has fixed time steps of $0.1s$.

By treating each level separately, we subdivide a complex problem into manageable subproblems. Lower levels can communicate with higher ones, e.g. to let local observations trigger global re-planning.

3 Environment Representation

A representation of the environment should provide the flexibility required by multi-level crowd simulation. Traditional graphs lead to fixed and unnatural paths, and regular grids either lack precision or are expensive to store and query. By contrast, a *navigation mesh* efficiently subdivides the walkable space into polygonal areas.

Our navigation mesh, the Explicit Corridor Map (ECM), is based the medial axis. It is a sparse graph that allows fast route planning, it can be constructed quickly and without grid-like approximations, it

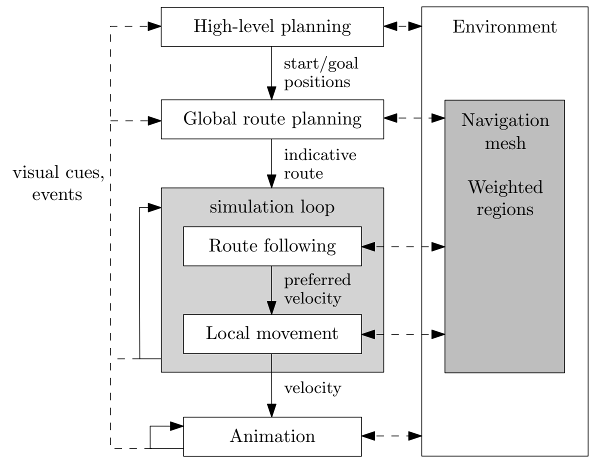

Figure 1: *A five-level hierarchy for (multi-)agent navigation.*

stores clearance information to allow path planning for disk-shaped agents of any size, it can produce a variety of indicative routes, it supports efficient nearest-obstacle queries, it can be dynamically updated in real-time, and it is well-defined for multi-layered 3D environments such as stadiums and multi-storey buildings. In short, the ECM is an efficient and generic basis for agent navigation.

4 Software

We have created a modular ECM-based crowd simulation software package for global planning, path following, and local movement. It runs on multiple platforms, it supports multi-threading, and it can be linked to other programs as an external library. Experiments show that our software can efficiently compute the ECM for large environments. Operations such as computing indicative routes, constructing visibility polygons, and inserting or removing dynamic obstacles can be performed within milliseconds. Without collision avoidance, we can simulate over $100,000$ agents in real-time on one thread. With collision avoidance and 8 parallel threads, we can simulate over $10,000$ agents in real-time. The software has a small memory footprint that supports over 1 million agents.

Our software has been used by companies to simulate various real-life events, such as the *Grand Départ* of the Tour de France in Utrecht in 2015, and the King's Day festivities in Amsterdam in 2013. A remaining challenge for crowd simulation research is *validating* the results by comparing them to real-world observations.

Acknowledgments

This research has been supported by the COMMIT project (`http://www.commit-nl.nl/`).

References

JAKLIN, N., TOLL, W., AND GERAERTS, R. 2013. Way to go - a framework for multi-level planning in games. In *Proceedings of the 3rd International Planning in Games Workshop*, 11–14.

*e-mail:W.G.vanToll@uu.nl
†e-mail:N.S.Jaklin@uu.nl
‡e-mail:R.J.Geraerts@uu.nl