NEOGEN: Near Optimal Generator of Navigation Meshes for 3D Multi-Layered Environments.

R. Oliva, N. Pelechano

Universitat Politcnica de Catalunya

Abstract

In this paper we introduce a novel automatic method for generating near optimal navigation meshes from a 3D multi-layered virtual environment. Firstly, a GPU voxelization of the entire scene is calculated in order to identify and extract the different walkable layers. Secondly, a high resolution render is performed with a fragment shader to obtain the 2D floor plan of each layer. Finally, a convex decomposition of each layer is calculated and layers are linked in order to create a Navigation Mesh of the scene. Results show that our method is not only faster than previous work, but also creates more accurate NavMeshes since it respects the original shape of the static geometry. It also provides a significantly lower number of cells and avoids ill-conditioned cells and T-Joints between portals that could lead to unnatural character navigation.

Keywords: Cell-and-Portal Graphs, path planning.

1. Introduction

Character navigation in complex scenes is commonly performed by having a Navigation Mesh (NavMesh), which encodes a convex decomposition of the scene. The NavMesh is represented with a Cell-and-Portal Graph (CPG), where cells are convex regions and portals are the edges shared by convex regions that a character can traverse. Path finding algorithms such as A* are then used over those NavMeshes.

Although NavMeshes are widely used in complex applications such as video games and virtual simulations, there are limited applications to automatically generate a NavMesh suitable for path planning. Either the user needs to manually refine semi-automatically generated NavMeshes, or create them manually from scratch, which is extremely time consuming and a source of errors. There is therefore a need for automatic methods to generate CPGs from any given 3D environment with minimum user input required.

Previous work is either not fully automatic, cannot handle any geometry, and/or provides CPGs with far too many cells or ill-conditioned cells (a cell where vertices are practically collinear which occurs when any of the internal angles is close to 0, or when the ratio Area/Perimeter of the polygon is close to 0). An over-segmented partition has two main problems. Firstly, the performance of the path finding algorithm directly depends on the dimensions of the generated graph, so the fewer cells we have, the faster this step will be. Secondly, depending on the underlying local movement algorithm being used, an over-segmented partition may end up with characters walking in zig-zags through a long convex space as they are forced to go through unnecessary portals, or in portals so close together that they add too many unnecessary nearby attractors and therefore complexity when trying to achieve natural looking local movement.

The architecture presented in this paper is novel, as it overcomes all limitations described above and presents an entire pipeline to go automatically from a 3D multilayer environment given as a polygon soup, to the final navigation mesh which adjusts tightly to the original geometry.

The main contribution of this paper is a novel GPU based method to generate a CPG for a given 3D scene (with slopes, steps and other obstacles). The algorithm starts by performing a GPU voxelization of the geometry to classify the different layers and calculate a *cutting shape*, *CS*. The *CS* is a depth filter used by the fragment shader to flatten each layers' geometry into a 2D high resolution texture encoding the depth map of each layer. Then each layer is encoded as a single simple polygon with holes which is the required input for the NavMesh generator [1]. The NavMesh generator provides a convex decomposition with a number of cells close to the optimal value, since for most cases it only needs to create one portal per notch of the polygon with holes (see proof in the appendix). The resulting CPGs are further optimized with a novel convexity relaxation technique which further reduces the number of cells. Finally all the layers' CPG are automatically linked together to obtain the final CPG of the entire scene. Figure 1 shows the flow of the algorithm.

To the best of our knowledge this is the first fully automatic architecture that can provide an accurate navigation mesh, with an almost optimal number of cells, ready for path planning from just a polygon soup (with degeneracies such as intersections and holes). The presented system is efficient enough to allow the scene modeler to make changes and observe the impact on the final navigation mesh at interactive rates and is of high importance to the fields of video games, robotics, movies and character simulation.