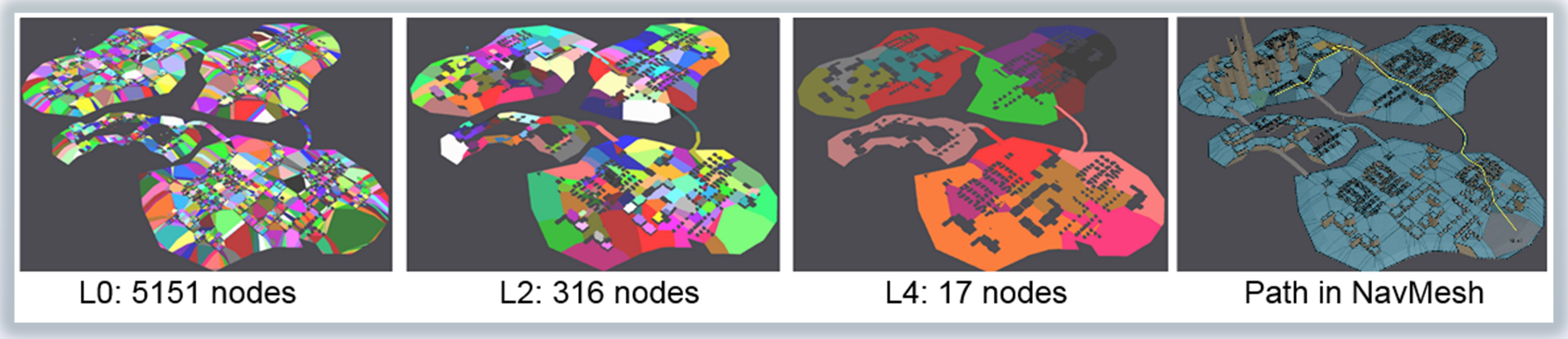
Hierarchical Path-Finding for Navigation Meshes (HNA*)

C. Fuentes, N. Pelechano
Universitat Politècnica de Catalunya

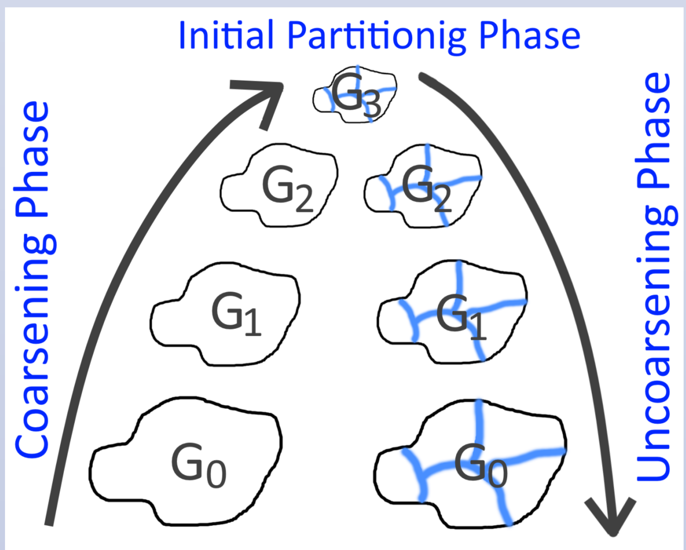


In this poster we present a method to create a hierarchical representation based on a multilevel k-way partitioning algorithm (MLkP), annotated with sub-paths that can be accessed online by our Hierarchical NavMesh Path-finding algorithm (HNA*). The algorithm greatly benefits from searching in graphs with a much smaller number of cells, thus performing up to 7.7 times faster than traditional A* over the initial NavMesh

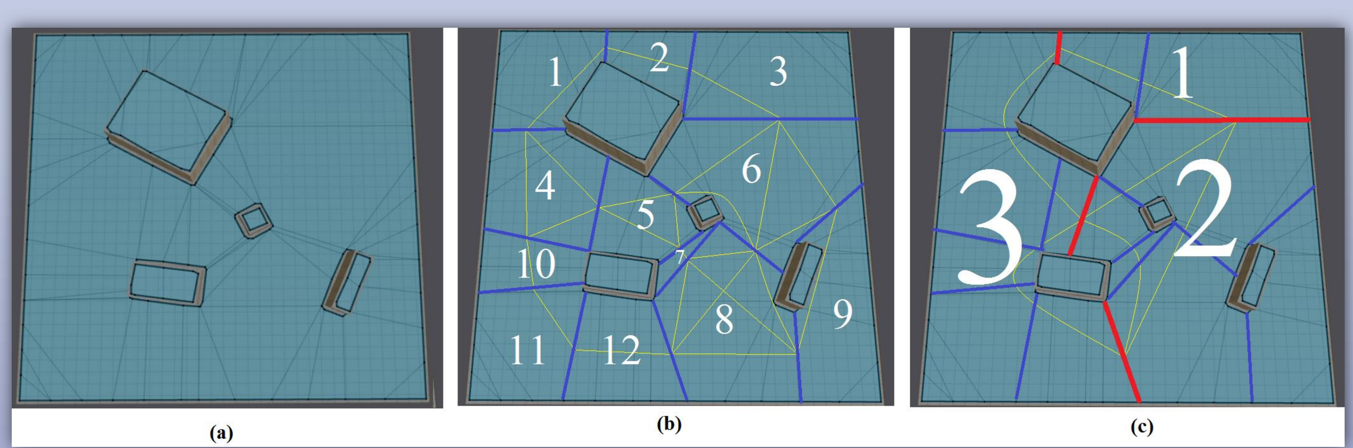
Hierarchical Representation

Based on: multilevel k-way partitioning algorithm (MLkP)

Graph partition with:
- ✓ good balance of cells
- ✓ small number of edges between partitions



- ▸ Intra-edges: store optimal paths between portal edges
- ▸ Inter-edges: connect nodes of the partition

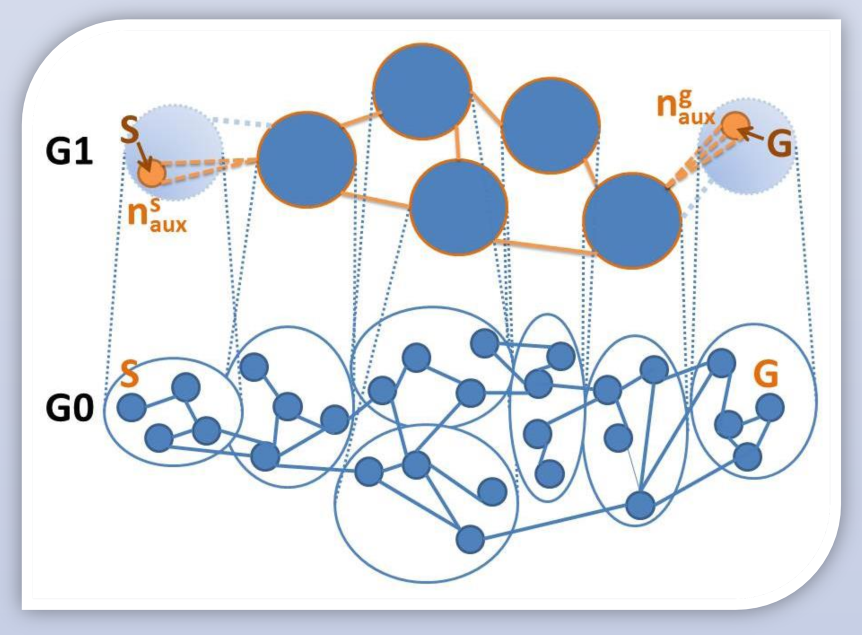


Hierarchical subdivision of a simple map, with μ=5 and 3 levels. Red lines in (c) represent inter-edges and yellow lines in (b) and (c) represent intra-edges. Partitions are shown with black (a), blue (b) and red (c) separation lines respectively. Level 0=76 nodes (a), Level 1=12 nodes (b), Level 2=3 nodes (c).

Path Finding

HNA* search:

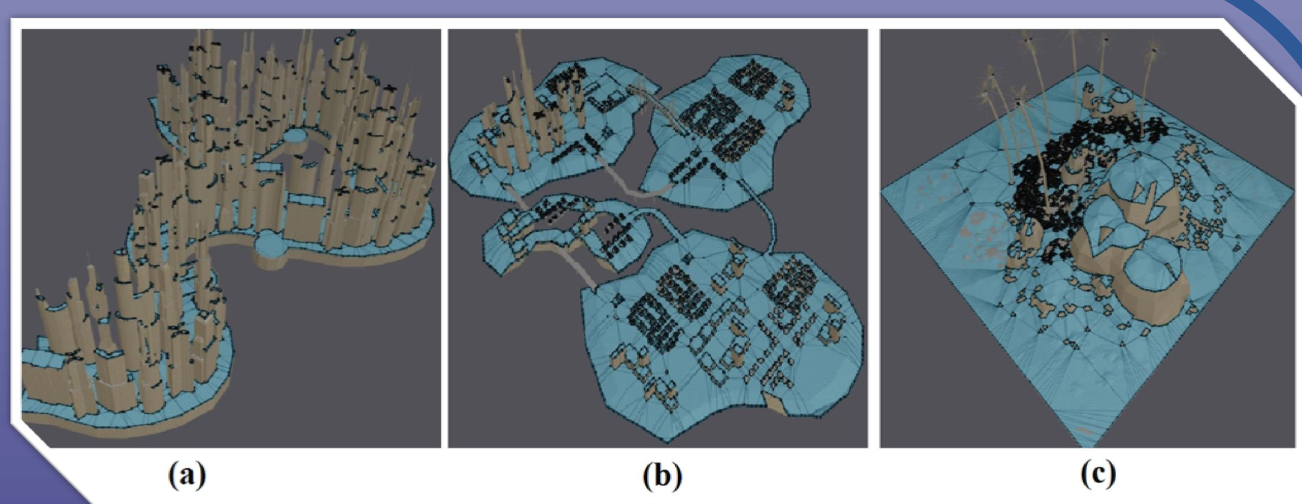
1. Insert and connect start (S) and goal (G)
2. Search path at the highest level
3. Extract intra-edges
4. Delete temporal nodes



Algorithm Online HNA*

procedure ONLINESEARCH(S, G, l)
  //step 1. Insert and connect nodes S and G at level l
  $n^s_l \leftarrow getNode(S, l)$
  $n^g_l \leftarrow getNode(G, l)$
  if $l = 0$ **then**
    $path \leftarrow findPath(n^s_l, S, n^g_l, G, 0)$
    return $path$
  $n^s_{aux} \leftarrow linkStartToGraph(S, n^s_l)$
  $n^g_{aux} \leftarrow linkGoalToGraph(G, n^g_l)$
  //step 2. Path-finding between S and G at level l:
  $tempPath \leftarrow findHNA^*Path(n^s_{aux}, S, n^g_{aux}, G, l)$
  //step 3. Extract sub-paths:
  for $subpath \in temPath$ **do**
    $path \leftarrow getIntraEdges(subpath, l-1)$
  //step 4. Delete S and G:
  $deleteTempNode(n^s_{aux})$
  $deleteTempNode(n^g_{aux})$
  return $path$

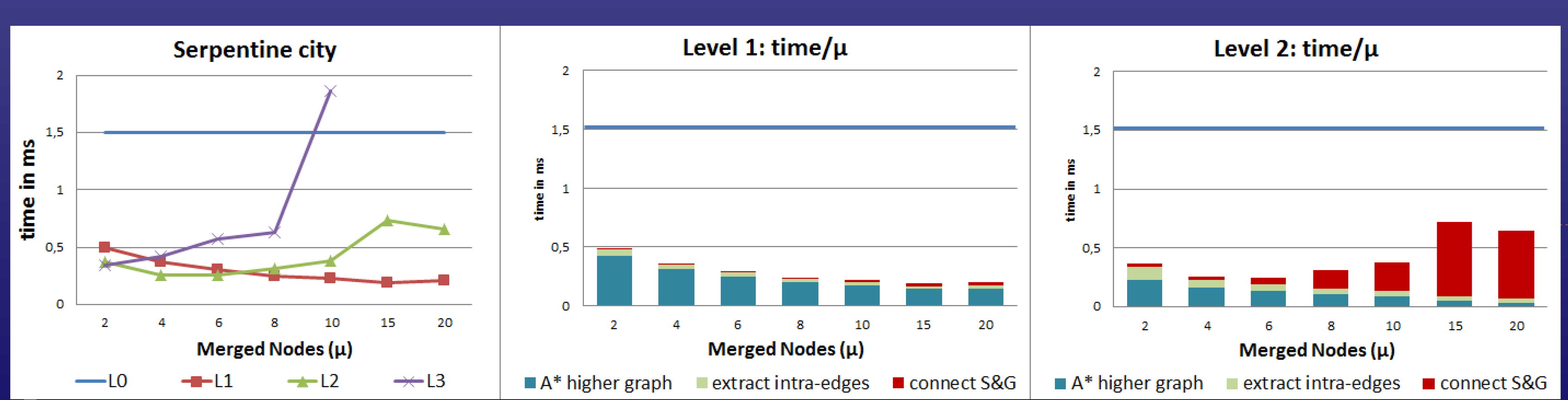
Results:



nodes = (a) 3908, (b) 5515, (c) 12666

The average cost of calculating several paths using HNA* in NavMeshes of different sizes has been computed with an intel core i7-4770 CPU@3.5Gz, 16GB RAM.
We have used up to three levels for the hierarchy and increasing values of merged polygons between levels (μ). For the example NavMeshes we obtained the following speed ups:
(a) 7.7x for L1 and μ = [15; 20], (b) 3.9x for L1 and μ = 15, and (c) 4.0x for L2 and μ = 6.
The current bottleneck is the cost of connecting S and G using A* which can escalate as the partition size increases.

a)



b)