Hierarchical Semantic Path-Planning in 3D Scene Graphs

Evangelos Psomiadis and Panagiotis Tsiotras

***Abstract*— This paper addresses the problem of robot navigation in mixed geometric and semantic 3D environments. Given a hierarchical representation of the environment, the objective is to navigate from a start position to a goal while minimizing computational cost and satisfying task-specific safety constraints described by semantics. We introduce Hierarchical Class-ordered A* (HCOA*), an algorithm that leverages the hierarchical structure of 3D Scene Graphs (3DSGs) for efficient and safe path-planning. We use a total order over the semantic classes and prove completeness of the algorithm. To incorporate safety constraints on the upper-layers of the hierarchical environment, we propose two methods for higher-layer node classification based on the semantics of the lowest layer: a Graph Neural Network-based method and a Majority-Class method. We validate our approach on the uHumans2 3DSG dataset [1], demonstrating that HCOA* reduces node expansions by 25% and computational time by 16% compared to the state-of-the-art baseline, while finding the optimal path and effectively avoiding unsafe objects and rooms.**

I. Introduction

As robotic sensing technologies advance, enabling robots to perceive vast and diverse information, two fundamental questions arise: *What information from this extensive data stream is most important for a given task, and how can the robot effectively utilize this information for decision-making?* Hierarchical semantic environment representations, such as 3D Scene Graphs (3DSG) [2]–[4], provide rich and structured abstractions that mirror human way of thinking, facilitating the selection and organization of information.

Previous research in hierarchical path-planning has primarily addressed the first question [5]–[8]. In [5] the authors introduce Hierarchical Path-Finding A*, a hierarchical A* variant for grid-based maps. Their approach partitions the map into clusters with designated entrance points, which are used for high-level path-planning. Similarly, the authors in [6] propose a hierarchical graph search algorithm for graphs with edge weights represented as intervals.

Semantic path-planning has focused on the second question by incorporating semantics into the decision-making process. Safety constraints can be expressed as relationships between different semantic categories and implicitly enforced by the path-planning algorithms. In [9] the authors introduce a weighted function that combines the edge cost and the semantic class to determine the optimal path. However, computing this weighted function requires global graph properties, which can be computationally demanding. To address this limitation, [10] and [11] propose Class-ordered A*/LPA*, two extensions of traditional A* [12] that efficiently incorporate semantics, based on a total order over the available classes. Building on [6] and [10], we develop a hierarchical class-ordered path-planning algorithm.

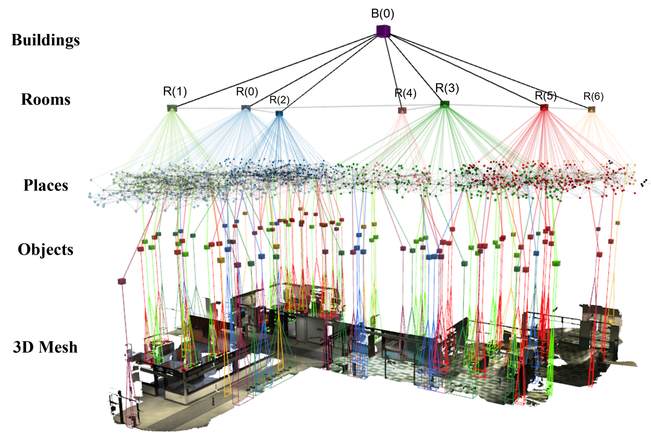

Fig. 1: 3D Scene Graph generated from the uHumans2 office scene dataset [1] using Hydra [3]. The graph comprises five layers, as shown in the figure. Room nodes are denoted as R(·), while building nodes are represented as B(·).

Navigation within 3DSGs has also been explored in prior work. In [13], the authors address Task and Motion Planning in 3DSGs using a three-level hierarchical planner. Their approach focuses on optimizing task planning, while path-planning operates on the unpruned 3DSG. Finally, learning-based methods have also been leveraged to facilitate task execution in a 3DSG. In [14] and [4], the authors propose the Neural Tree. Although effective in classifying higher-layer nodes in 3DSGs, this approach requires a tree decomposition of the input graph, which increases computational time.

Main Contributions: Our approach integrates task semantics directly into the path-planning process within a unified algorithm. Our key contributions are as follows:

1) We introduce Hierarchical Class-ordered A*, a novel hierarchical semantic path-planning algorithm for navigation in 3DSGs.
2) We propose two methods for node classification on the higher layers of the hierarchy based on underlying semantic classes: a GNN-based method and a majority-class method.
3) We validate our approach on a publicly available 3DSG dataset.

The work was supported by ARL award DCIST CRA W911NF-17-2-0181 and ONR award N00014-23-1-2304.

E. Psomiadis, and P. Tsiotras are with the D. Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: {epsomiadis3, tsiotras}@gatech.edu

II. Preliminaries

Let $G = (V, E, \mathcal{K})$ be a 3D Scene Graph (3DSG), as in Figure 1, with n layers (disregarding mesh and objects layers), where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, and $\mathcal{K} = 1, \dots K$ is a set of semantic classes ordered by decreasing priority. We denote the set of layers as L, where $\ell = 0$ is the places layer and $\ell = n$ is the highest layer (root). Each layer ℓ forms a connected weighted subgraph $G^l = (V^\ell, E^\ell, \mathcal{K})$, with an associated weight function $w : E^\ell \to \mathbb{R}^+$. We assume that each node in layer $\ell - 1$, for $\ell = 1, \dots n$, is connected to exactly one node in layer ℓ, which we refer to as its parent node. More generally, we define an ancestor as a node's parent or any higher-layer predecessor in the hierarchy. We define the projection function $p : V \times L \to V$ that maps each node to its corresponding ancestor in layer ℓ.

Each node in $\ell = 0$ is assigned a semantic class based on perception data and safety constraints using the function $\phi_V^0 : V^0 \to \mathcal{K}$. We extend the labeling function ϕ_V^0 to layers $\ell \neq 0$. Details on the computation of ϕ_V^ℓ for $\ell \neq 0$ will be given in Section V. Additionally, we introduce the function $\phi_E^\ell : E^\ell \to \mathcal{K}$, which assigns semantic classes to the edges of G. Specifically, we define the edge classification function for an edge $e = (v, u)$ as $\phi_E^\ell(e) = \max(\phi_V^\ell(v), \phi_V^\ell(u))$, thereby overestimating the edge class.

Let $\Pi(v_s^\ell, v_g^\ell)$ denote the set of all acyclic paths in G^ℓ from $v_s^\ell \in V^\ell$ to $v_g^\ell \in V^\ell$, and let Π_k be the subset of Π in which the least favorable edge class is exactly k. Formally,

$$\Pi_k = \{\pi : N(\pi, k) > 0 \text{ and } N(\pi, k') = 0, \forall k' > k\}, \quad (1)$$

where $N(\pi, k) = |\{e \in \pi : \phi_E^\ell(e) = k\}|$. We further define $\Pi_k^i \subseteq \Pi_k$ as $\Pi_k^i = \{\pi \in \Pi_k : N(\pi, k) = i\}$. To enable a consistent comparison of paths across different classes, we impose a total order such that $k < \ell \Rightarrow \Pi_k^i \prec \Pi_\ell^j$ for all i, j and $i < j \Rightarrow \Pi_k^i \prec \Pi_k^j$. This order ensures that any two paths with the same start and goal nodes can be compared.

III. Problem Formulation

Consider a mobile robot tasked with navigating a complex 3D environment where certain regions have lower traversal priority. The robot is provided with a 3DSG of the environment, where the semantic classes of the nodes in layer $\ell = 0$ are assigned based on perception and tailored to the specific task and safety constraints (e.g. avoid certain areas and objects). Let $v_s \in V^0$ and $v_g \in V^0$ denote the robot's starting and goal nodes, respectively. The objective is to determine the shortest path within layer $\ell = 0$ while minimizing traversal through the least favorable edges. Formally,

$$\pi^*(v_s, v_g) = \underset{\pi \in \Pi^*(v_s, v_g)}{\arg\min} \sum_{e \in \pi} w(e), \quad (2a)$$

$$\Pi^* = \min_{i \in \mathbb{N}} \min_{k \in \mathcal{K}} \Pi_k^i, \quad (2b)$$

where Π^* is the set of paths obtained by minimizing over all possible classes k and number of least favorable class i.

However, due to computational constraints, the robot seeks to avoid a full graph search over G^0, as its structure can be large, potentially rendering the search intractable.

Algorithm 1 Hierarchical Class-ordered A* (HCOA*)

Input: G, v_s, v_g, $h(v)$
Output: π^0

for all $\ell = \ell_n, \dots, \ell_0$ **do**
  $v_s^\ell \leftarrow p(v_s, \ell)$; $v_g^\ell \leftarrow p(v_g, \ell)$
  $g(v_s^\ell) \leftarrow 0$; $\theta(v_s^\ell) \leftarrow 0 \cdot \mathbf{1}_K$; $f(v_s^\ell) \leftarrow h(v_s^\ell)$
  $g(v^\ell) \leftarrow \infty$; $\theta(v^\ell) \leftarrow \infty \cdot \mathbf{1}_K$, $\forall v^\ell \in G \setminus \{v_s^\ell\}$
  PredecessorMap $\leftarrow \varnothing$ ▷ Track path reconstruction
  $Q \leftarrow \{v_s^\ell\}$ ▷ Initialize priority queue
  while Q is not empty **do**
    $v \leftarrow$ PopNode(Q); $Q \leftarrow Q \setminus \{v\}$
    if $v = v_g^\ell$ **then**
      $\pi^\ell \leftarrow$ Path(PredecessorMap, v_g)
      Break
    end if
    for all $u \in$ neighbors$(v, G^{\ell'})$ **do**
      $\theta(v, u) \leftarrow$ Semantics$(\phi_V^\ell(v), G^{0'}(u))$
      if $\big(\theta(v) + \theta(v, u) \prec \theta(u)\big)$ **or**
      $\big(\theta(v) + \theta(v, u) = \theta(u)$ **and**
      $g(v) + w(v, u) < w(u)\big)$ **then**
        $Q \leftarrow Q \cup \{u\}$
        PredecessorMap$[u] \leftarrow v$
        $\theta(u) \leftarrow \theta(v) + \theta(v, u)$
        $g(u) \leftarrow g(v) + w(v, u)$
        $f(u) \leftarrow g(u) + h(u)$
      end if
    end for
  end while
  for all $\ell' < \ell$ **do**
    $G^{\ell'} \leftarrow \{v \in G^{\ell'} : p(v, \ell) \in \pi^\ell\}$
  end for
end for
return π^0

A. Problem Statement

We propose a hierarchical semantic path-planning algorithm for robot navigation. The algorithm operates top-down across the layers of the 3DSG, iteratively computing the optimal semantic path at each layer while pruning nodes that are not included in the selected path. Additionally, we introduce two methods for semantic class prediction of higher-layer nodes: a Graph Neural Network (GNN)-based approach and a majority-class method.

IV. Path-Planning

A. Hierarchical Class-ordered A*

We introduce Hierarchical Class-ordered A* (HCOA*), a hierarchical semantic path-planning algorithm for 3DSGs. The algorithm begins by finding a path in layer $\ell = n - 1$ and then refining it by recursively applying the same process at the lower levels of the hierarchy. At each layer, the algorithm utilizes Class-ordered A* (COA*) [10], which finds the shortest path while minimizing the number of least favorable edges through lexicographic comparison.

HCOA* is presented in Algorithm 1. We use the function $h : V \to \mathbb{R}$ to denote an admissible heuristic function,

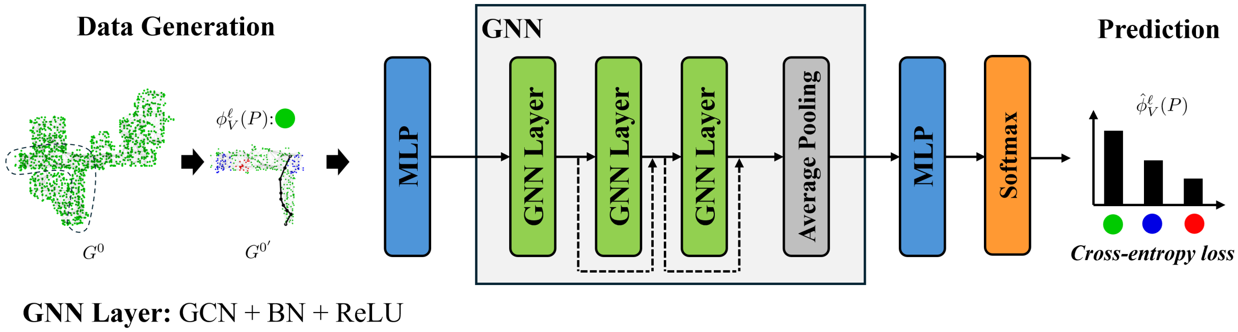

Fig. 2: Proposed GNN architecture for node $P \in V^\ell$ ($\ell \neq 0$) classification, utilizing the semantic classes (green, blue, red) of nodes in $\ell = 0$.

similar to standard A*. Lines 2–6 initialize the variables. Lines 7–24 executes a simplified version of COA* [10]. Notably, Line 14 computes the semantic class of the edge (v, u) based on the semantic classes of nodes v and u. Additionally, $G^{0'}(u)$ is the induced subgraph of node u in layer $\ell = 0$ given by $G^{0'}(u) = \{u' \in V^0 : p(u', \ell) = u\}$. Finally, Lines 25–27 refine the path by pruning nodes that do not share an ancestor with the paths in the higher layers.

B. HCOA* Completeness

The following proposition establish the algorithm's completeness. We assume that the assumptions for completeness and optimality of COA* hold [10].

Proposition 1: Let G be an 3DSG and let nodes $v_s, v_g \in V^0$. HCOA* is *complete*, meaning that it is guaranteed to find a path $\pi^0 = \pi(v_s, v_g)$, whenever one exists.

Proof: Let π^0 exists. The structure of G ensures that for each layer $\ell \in L$, there exists a corresponding path $\pi^\ell = \pi(v_s^\ell, v_g^\ell)$, where $v_s^\ell = p(v_s, \ell)$ and $v_g^\ell = p(v_g, \ell)$.

At each subgraph G^ℓ, COA* is guaranteed to find the optimal path $\pi^{\ell *}$, if one exists [10]. Suppose, ad absurdum, that the graph pruning performed in Lines 25–27 results in $\Pi(v_s^{\ell-1}, v_g^{\ell-1}) = \varnothing$, preventing COA* from finding a solution in layer $\ell - 1$. This would imply that $\pi^{\ell *}$ is disconnected, contradicting the completeness of COA*. ■

V. Semantic Class Prediction

Let $\phi_V^\ell : V^\ell \to \mathcal{K}$ be the function that assigns semantic classes to nodes in the layers $\ell \neq 0$. Ideally, given an optimal path $\pi^*(v_s, v_g)$ computed in G^0, the semantic class of a node $P \in V^\ell$ is determined by $\phi_V^\ell(P) = \max_{u \in \pi'(P)} \phi_V^0(u)$, where $\pi'(P) = \pi^*(v_s, v_g) \cap G^{0'}$. This process follows a bottom-up approach, necessitating the computation of the optimal path in layer $\ell = 0$. Given that HCOA* works top-down, we seek an alternative method.

A. Graph Neural Network Method

We formulate this problem as a supervised node classification task and design a GNN to predict the semantic classes. We construct a dataset $\mathcal{D} = \{G^{0'}(P), \phi_V^\ell(P)\}$ by selecting nodes $P \in V^\ell$ for $\ell \neq 0$ and extracting their induced subgraphs $G^{0'}(P)$. We then assign semantic classes to the nodes in $G^{0'}(P)$ based on the given task. Next, we execute COA* on this graph and compute $\phi_V^\ell(P)$, where the starting location is a border node. A border node $v^* \in V^0$ of layer ℓ is defined as a node that shares an edge with another node from a different ancestor. Formally, there exists $e = (v^*, u^*)$ where $v^*, u^* \in V^0$ and $p(v^*, \ell) \neq p(u^*, \ell)$. Border nodes play a crucial role in the classification of $P \in V^\ell$.

The proposed model is illustrated in Figure 2. We use border nodes as input features for the network, which are concatenated with the semantic classes of the nodes in $G^{0'}$. A 2-layer MLP is used to preprocess the input. The GNN consists of three layers, each incorporating a Graph Convolutional Network (GCN) operator [15] for message passing, followed by batch normalization and a ReLU activation. To mitigate oversmoothing, we introduce skip connections between the GNN layers. Additionally, we apply average pooling to handle environments of varying sizes. The pooled representation is then processed through another 2-layer MLP, followed by a softmax function. Finally, we use cross-entropy as the loss function for node classification.

B. Majority-Class Method

The Majority-Class (MC) method computes the semantic class of higher-layer nodes by counting the occurrences of each semantic class among the nodes in the induced subgraph $G^{0'}(P)$ and selecting the most frequent one. That is,

$$\widehat{\phi}_V^\ell(P) = \arg\max_{k \in \mathcal{K}} N_V(G^{0'}(P), k), \tag{3}$$

where $N_V(G, k) = |\{v \in G : \phi_V^0(v) = k\}|$.

VI. Simulations

We perform two sets of simulations on the publicly available uHumans2 office 3DSG, shown in Figure 1 and constructed using Hydra [3]. The places layer $\ell = 0$ is a subgraph where each node represents an obstacle-free location. The rooms layer $\ell = 1$ consist of nodes representing room centers. We consider a set of three semantic classes $\mathcal{K}$, ordered by decreasing priority: 1 (Green), 2 (Blue), 3 (Red). We utilize the objects layer to identify objects and assign semantic classes to the surrounding place nodes.

In the first section, we present the comparison between GNN and the MC approach. The second section demonstrates a path-planning scenario and compares HCOA*

TABLE I: Room Node Classification.

Metrics	GNN	MC
Training Time (min)	90	-
Validation Accuracy (%)	**67.64**	30.36
Test Accuracy (Rooms with 1-20 bn) (%)	**82.50**	54.25
Test Accuracy (Rooms with 21-30 bn) (%)	**65.00**	19.75
Test Accuracy (Rooms with 31-40 bn) (%)	**64.00**	19.25
Test Accuracy (Rooms with 41-50 bn) (%)	**59.00**	24.50

TABLE II: Path-Planning on uHumans2 Office Scene.

Metrics	HCOA*-GNN	HCOA*-MC	COA*
Expanded Nodes	**412**	**412**	549
Time (10^{-3}s)	38.8 $\pm$ 47.1	**4.3** $\pm$ **2.6**	5.1 $\pm$ 2.5
Optimal Path	✓	✓	✓

against COA*. HCOA*-GNN utilizes the best GNN model from the previous training phase for room inference. To evaluate the performance of room classification, we compute the accuracy, which measures the proportion of correctly classified rooms in each dataset. For the path-planning scenario, we compute the computational time of the algorithms along with the number of expanded nodes.

A. Room Node Classification

In this section, we present the results from predicting the semantic class of room nodes. To generate the dataset, we constructed 14,000 graphs by extracting the induced subgraphs from all rooms and randomly assigning semantic classes to nodes within a disk of randomly chosen center locations, repeating this process 2,000 times per room. Then, we ran COA* to determine the semantic class of the room, starting from a randomly selected border node. The dataset was split into 80% training, 10% validation, and 10% testing. The model was trained using the Adam optimizer [16] with a learning rate of 10^{-2}, over 1,600 epochs, a dropout rate of 0.2 and a batch size of 64. The GNN and MLP layers contain 32 neurons. Table I summarizes the results of room node classification. The results indicate that the GNN approach outperforms significantly the MC method across both the validation and test sets. Additionally, the training accuracy of the GNN is 67.94%.

B. Path-Planning on uHumans2 Office Scene

In this section, we demonstrate a path-planning scenario in the 3DSG shown in Figure 1. The starting node v_s is in R(2), while the goal v_g is in R(1). For the environment's semantic classes, we assign $\phi_V^0(v) = 2$, for all $v \in$ R(0) (e.g., R(0) represents a typically crowded area, which the robot should try to avoid). Additionally, we define $\phi_V^0(v) = 3$, for all $v \in C$, where C is the set of place nodes located within a disk centered around specific objects. Formally, $C = \{v \in G^0 : \|x(v) - x(o)\|_2 \leq r, \ \forall o \in O\}$, where $x(\cdot)$ denotes the location of a place node v or an object o, and O is a set of objects. In this scenario, we set $r = 3$m and define O as the set of all computers in the 3DSG. This means that the robot should avoid passing too close to computers for safety reasons. Figure 3(a) depicts the places layer of the environment along with the start and goal nodes.

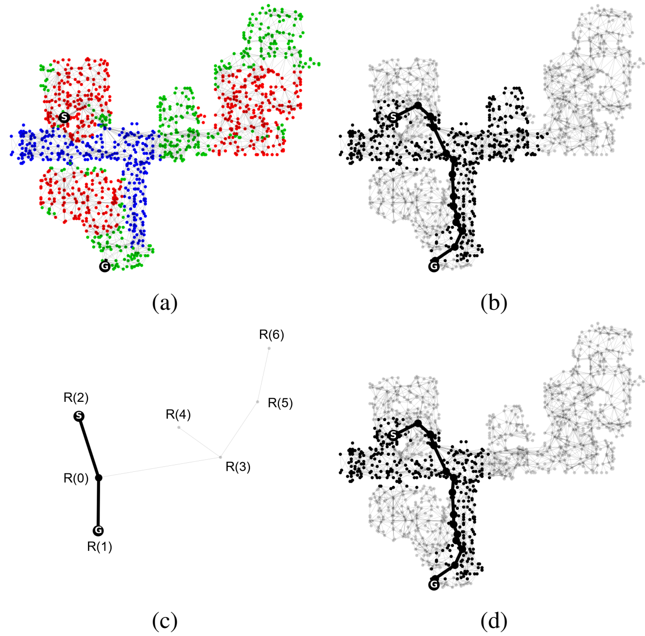

Fig. 3: (a) Places subgraph of the uHumans2 office scene (52m×45m) with the assigned semantic classes. The starting node is denoted by **S**, and the goal by **G**; (b-d) The path of each algorithm is shown in bold black. Expanded nodes are depicted in regular black, while unexpanded nodes in gray: (b) Path of COA* on the places; (c) Path of HCOA* on the rooms; (d) Path of HCOA* on the places.

The computed path of COA* in the places layer is shown in Figure 3(b). Figures 3(c)-(d) illustrate the results of HCOA*. Table II presents the results. HCOA* demonstrates a 25% reduction in expanded nodes compared to COA*. The computational time of the algorithms is calculated over 1,000 runs. We observe that HCOA*-MC achieves the best performance in terms of computational efficiency, reducing the execution time by 16% compared to COA*. However, HCOA*-GNN has the highest computational time, as GNN inference can be more time-consuming than graph search in small graphs (the entire places subgraph contains only 1,314 nodes). Finally, all three approaches successfully compute the optimal path in the places layer, as shown in Figure 3.

VII. Conclusions

In this paper, we addressed the problem of robot navigation in 3D geometric and semantic environments. We introduced Hierarchical Class-ordered A* (HCOA*), an algorithm that exploits a total order over semantic classes to guide the search process while significantly reducing computational effort and satisfying safety constraints. To classify higher-layer nodes, we proposed two methods: a Graph Neural Network-based method and a Majority-Class method. Through simulations on a 3D Scene Graph, we showed that HCOA* effectively reduces computational cost compared to a state-of-the-art method. Specifically, our results show that HCOA* finds the optimal path while reducing the number of expanded nodes by 25% and achieving a 16% reduction in computational time for a typical 3DSG with 1,314 nodes.

REFERENCES

[1] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: from SLAM to spatial perception with 3D dynamic Scene Graphs," *Intl. J. of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

[2] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3D Scene Graph: A structure for unified semantics, 3D space, and camera," in *IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct 27 - Nov 02 2019, pp. 5663–5672.

[3] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Robotics: Science and Systems (RSS)*, New York, NY, June 27 – July 1 2022.

[4] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *The International Journal of Robotics Research*, vol. 43, no. 10, pp. 1457–1505, Feb 2024.

[5] A. Botea, M. Müller, and J. Schaeffer, "Near optimal hierarchical path-finding (HPA*)," *Journal of Game Development*, vol. 1, pp. 1–30, Jan 2004.

[6] J. Fernandez and J. Gonzalez, "Hierarchical graph search for mobile robot path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, Leuven, Belgium, May 16-21 1998, pp. 656–661.

[7] C. W. Warren, "Fast path planning using modified A* method," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, Atlanta, GA, May 1993, pp. 662–667.

[8] R. Holte, C. Drummond, M. Perez, R. Zimmer, and A. Macdonald, "Searching with abstractions: A unifying framework and new high-performance algorithm," in *Canadian Conference on Artificial Intelligence*, Banff, Canada, Jan 1994, pp. 263–270.

[9] D. Wooden and M. Egerstedt, "On finding globally optimal paths through weighted colored graphs," in *IEEE Conference on Decision and Control (CDC)*, San Diego, CA, Dec 13-15 2006, pp. 1948–1953.

[10] J. Lim and P. Tsiotras, "A generalized A* algorithm for finding globally optimal paths in weighted colored graphs," Dec 2020, arXiv:2012.13057.

[11] J. Lim, O. Salzman, and P. Tsiotras, "Class-ordered LPA*: An incremental-search algorithm for weighted colored graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, Sep 27 - Oct 01 2021, pp. 6907–6913.

[12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[13] A. Ray, C. Bradley, L. Carlone, and N. Roy, "Task and motion planning in hierarchical 3D Scene Graphs," 2024, arXiv:2403.08094.

[14] R. Talak, S. Hu, L. Peng, and L. Carlone, "Neural trees for learning on graphs," in *Conference on Neural Information Processing Systems (NeurIPS)*, Dec 06-14 2021.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, Toulon, France, April 24-26 2017.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 07 - 09 2015.