



¹Nokia
Bell Labs



²Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
⁴BSc Computer Engineering [CE]



³TopoLynx
Ltd.

Semantic Navigation Agnostic to Map-Robot Variations

László Freund^{1,2,4}
author
BSc Student
Nokia Bell Labs

Márk Czimber^{2,3,4}
author
BSc Student
HUN-REN Sztaki

Domokos Kiss, PhD²
supervisor
Senior Lecturer

Dániel Hadházi²
supervisor
Scientific Assistant

Gábor Sörös, PhD¹
supervisor
Nokia Bell Labs
Senior Research Scientist

Kornél Czimber, PhD³
supervisor
University of Sopron
Associate Professor, Vice Dean

2024, Scientific Student Conference

1 Appendix

```
Task: Your task is to describe and label images taken inside a building.

Expected input:

- Image: An image captured inside a building.
- Global Context: A brief description of the building, including its layout, purpose,
                  or any relevant information that could help identify
                  different rooms or spaces.

Output:
Respond in the following JSON format:

{
  "label": "<room_or_space_name>",
  "description": "<brief_description_and_reasoning>"
}

label: Provide the most likely name or function of the room/space where the image was
       taken (e.g., "kitchen," "conference room," "hallway").

description: Give a brief description of the image and explain the reasoning behind
             your decision, citing features in the image and the global context
             that led you to that conclusion.

Use maximum 1 sentence for each cluster description, be as brief as possible.

---

Global Context:
```

Figure 1: Image description prompt: the LLM is asked to label and describe images while taking into account the global context of the scene.

The input to the model is a 3D scene described in a JSON format.

The JSON includes:

- Global Context: A brief description of the building, including its layout, purpose, or any relevant information that could help identify different rooms or spaces.
- Clusters: A list of clustered spaces representing rooms or other distinct parts of the building. Each cluster contains an "objects" field, listing the detected items present in the space, and an "image_descriptions" field, containing labels and descriptions for images taken in the cluster.

Note that image descriptions and guessed labels may be unreliable ,should not be taken for granted.

Example Input:

```
{
  "global_context": "description of the building (e.g., office, hospital)",
  "clusters": [
    {
      "cluster_id": 1,
      "objects": ["object1", "object2", ...],
      "image_descriptions": [
        {
          "label": "hallway",
          "description": "Description for the image ..."
        },
        ...
      ]
    },
    ...
  ]
}
```

Label each cluster based on the global context, the items within the cluster, and the image descriptions, while considering that the provided image descriptions may be inaccurate. Output the results as a JSON array of objects, where each object includes:

- "cluster_id": The ID of the cluster.
- "label": A concise label that best describes the cluster's purpose or function.
- "description": A brief explanation of why the label was chosen, considering the global context, objects, and image descriptions.

Example Output:

```
[
  {
    "cluster_id": 1,
    "label": "hallway",
    "description": "A long, narrow corridor with a white tiled floor and a few potted plants, likely a common area connecting different parts of the office."
  },
  ...
]
```

Make sure the labels are concise while accurately capturing the function of each cluster. Use maximum 1 sentence for each cluster description. Make sure to give your response in JSON format, based on the example output. Do not add any other text to the response, other than the JSON output.

Figure 2: Cluster labeling prompt: the LLM is asked to label and describe clusters based on image descriptions, predicted image labels, nearby objects, and the global context.

The input to the model is a 3D scene described in a JSON format. The JSON contains a global description of the scene, specified in the "global_context" field. The scene is represented as a list of clusters, with each entry in the JSON describing a distinct room or space within the scene using the following fields:

- "cluster_id": A unique identifier for the cluster.
- "label": A brief, sometimes approximate tag categorizing the cluster.
- "description": A brief description of the function or purpose of the cluster within the scene.
- "objects": A list of objects present in the cluster.

Each object in the JSON is described with the following fields:

- "id": A unique object identifier.
- "object_tag": A brief, sometimes approximate, tag categorizing the object.

Example Input:

```
{
  "global_context": "A modern office building with multiple rooms for meetings.",
  "clusters": [
    {
      "cluster_id": "cluster_001",
      "label": "Conference Room",
      "description": "Conference room equipped for meetings.",
      "objects": [
        { "id": "obj_001", "object_tag": "table" },
      ]
    },
  ]
}
```

Once you have parsed the JSON, including the global scene description, and are ready to answer questions about the scene, say "I'm ready."

The user will then ask questions, and the task is to answer various user queries about the 3D scene. For each question, respond with a JSON dictionary with the following fields:

- "inferred_query": Your interpretation of the user query in a concise form.
- "relevant_clusters": A list of relevant cluster IDs for the user query, with a maximum of 5 entries, an empty list if there are no relevant clusters in the scene.
- "relevant_objects": A list of relevant object IDs for the user query, with a maximum of 5 entries, an empty list if there are no relevant objects in the scene..
- "query_achievable": Whether the user-specified query is achievable using the clusters, objects, and descriptions provided.
- "final_relevant": A mixed list of cluster IDs and object IDs relevant to the user-specified task, ordered by relevance. This list should combine clusters and objects as necessary, starting with the most relevant.
- "explanation": A brief explanation of why the selected clusters and objects are relevant.
- "most_relevant_desc": A concise description of the most relevant cluster or object, limited to 10 words, without mentioning the location.

Example Output:

```
{
  "inferred_query": "Find where the projector is located",
  "relevant_clusters": ["room_001"],
  "relevant_objects": ["obj_003"],
  "query_achievable": true,
  "final_relevant": ["room_001", "obj_003"],
  "explanation": "The projector is located in Room 001, the conference room.",
  "most_relevant_desc": "Projector in the conference room."
}
```

Figure 3: Semantic Navigation prompt: the LLM is instructed to interpret the scene description and respond to queries regarding objects and clusters within it.

Hyperparameter configurations:

General:

```
"base_height": 1.5,  
"base_radius": 0.2,  
"connection_radius": 3.0,  
"too_close_threshold": 0.5
```

SkeletonFinder:

```
"frontier_creation_threshold": 0.3,  
"frontier_jump_threshold": 3.0,  
"frontier_split_threshold": 2.0,  
"max_expansion_ray_length": 2.5,  
"max_facets_grouped": 10.0,  
"max_height_diff": 1.5,  
"max_ray_length": 2.5,  
"min_flowback_creation_radius_threshold": 0.25,  
"min_flowback_creation_threshold": 1.0,  
"min_node_radius": 0.8,  
"raywalking_max_height_diff": 0.2,  
"resolution": 0.1,  
"robot_type": 1.0,  
"sampling_density": 100.0,  
"search_margin": 0.1,  
"start_x": 1.0,  
"start_y": 0.0,  
"start_z": 1.0,
```

Figure 4: The hyperparameter configuration used for all our experiments, unless otherwise specified.