GraphNav Map Viewer

Interactive 3D visualization tool for Boston Dynamics SPOT GraphNav maps using Plotly.

Features

- **3D Interactive Visualization**: Pan, zoom, and rotate the map in your browser
- **Waypoint Inspection**: Hover over waypoints to see ID, name, short code, position, and fiducial count
- **Delivery Location Highlighting**: Highlight specific waypoints (green, larger markers)
- **Point Cloud Display**: Visualize feature clouds captured at each waypoint
- **Fiducial Markers**: Show AprilTag positions as orange diamonds
- **Edge Connections**: Display navigation edges between waypoints
- **Layer Toggle**: Dropdown menu to show/hide different map elements
- **HTML Export**: Save standalone HTML files for sharing

Installation

```
# Install viewer dependencies
uv sync --extra viewer
```

Usage

Basic Usage

```
# View map in browser
uv run python -m src.map_viewer maps/map_catacombs_01/
```

With Anchoring (Recommended)

Use the `-a` flag to enable seed frame anchoring for accurate positioning:

```
uv run python -m src.map_viewer maps/map_catacombs_01/ -a
```

Highlight Delivery Locations

Highlight specific waypoints by their short codes (from `WAYPOINTS` dict):

```
uv run python -m src.map_viewer maps/map_catacombs_01/ -a --highlight al tv oh cw
```

Short codes: - `al` - Aula (assembly hall) - `tv` - Triangle - `oh` - Hauswart (caretaker's room) - `cw` - Turnhalle (gymnasium)

Show Point Clouds

Display feature clouds captured at each waypoint (sampled to 50k points for performance):

```
uv run python -m src.map_viewer maps/map_catacombs_01/ -a --show-point-clouds
```

Show All Waypoint Labels

By default, only delivery locations show labels. To show all waypoint labels:

```
uv run python -m src.map_viewer maps/map_catacombs_01/ -a --show-labels
```

Export to HTML

Save a standalone HTML file that can be opened without Python:

```
uv run python -m src.map_viewer maps/map_catacombs_01/ -a --export map.html
```

Full Example

```
uv run python -m src.map_viewer maps/map_catacombs_01/ \
    -a \
    --highlight al tv oh cw \
    --show-point-clouds \
    --title "Kanti Glarus Delivery Map"
```

CLI Options

Option	Description
`path`	Path to GraphNav map directory (required)
`-a, --anchoring`	Use seed frame anchoring if available
`--highlight CODE...`	Waypoint short codes or names to highlight
`--show-labels`	Show labels on all waypoints
`--show-point-clouds`	Show point cloud data
`--no-edges`	Hide edge connections
`--no-fiducials`	Hide fiducial markers
`--title TEXT`	Custom title for the visualization
`--export FILE`	Export to HTML file instead of opening browser
`-v, --verbose`	Enable verbose logging

Interactive Controls

In Browser

- **Rotate**: Click and drag
- **Pan**: Right-click and drag (or Shift + click and drag)
- **Zoom**: Scroll wheel
- **Reset View**: Double-click

Legend

- **Single-click** on legend item: Toggle visibility
- **Double-click** on legend item: Isolate (hide all others)

Dropdown Menu

Use the dropdown in the top-left to quickly toggle visibility: - All Visible / All Hidden - Only [Layer Name] - Show only one layer - Hide [Layer Name] - Hide specific layer

Module Structure

```
src/map_viewer/
    __init__.py      # Package exports
    __main__.py      # Entry point for python -m
    cli.py           # Command-line interface
    loader.py        # Load map from protobuf files
    transformer.py   # Coordinate transforms (SE3Pose -> world positions)
    viewer.py        # Plotly visualization
```

Architecture

Data Flow

1. **loader.py**: Reads GraphNav protobuf files (graph, waypoints, snapshots, anchors)
2. **transformer.py**: Computes world positions using anchoring or BFS traversal
3. **viewer.py**: Creates Plotly 3D figure with interactive traces

Key Functions

- `load_map(path)` - Load map data from directory
- `compute_waypoint_positions(map_data)` - Get (x, y, z) for all waypoints
- `compute_point_clouds(map_data)` - Extract and transform point cloud data
- `create_figure(map_data, ...)` - Build interactive Plotly figure

Comparison with SDK Viewer

This viewer is an alternative to the official `graph_nav_view_map` example from the SPOT SDK:

Feature	SDK Viewer (VTK)	This Viewer (Plotly)
Dependencies	VTK (heavy, complex install)	Plotly (lightweight)
Platform	Desktop only	Browser-based
Interactivity	Basic	Rich (hover, zoom, toggle)
Export	None	Standalone HTML
Point Clouds	Yes	Yes (sampled)
Waypoint Info	Axes only	Hover with metadata