RViz Setup Guide for Enhanced ORB Tracker

Step-by-Step RViz Configuration

1. Launch RViz

```bash
# Open RViz
rviz2

# Or if you want to save/load configurations
rviz2 -d your_config.rviz
```

2. Basic Setup

1. **Set Fixed Frame**: In the Global Options panel, set `Fixed Frame` to `map`
2. **Add Grid** (optional):
   - Click `Add` → `By display type` → `Grid`
   - Set Grid properties:
     - Reference Frame: `map`
     - Cell Count: 20
     - Cell Size: 1.0

3. Add PointCloud2 Visualization

1. Click `Add` → `By display type` → `PointCloud2`
2. Configure PointCloud2:
   - **Topic**: `/feature_cloud`
   - **Size (Pixels)**: 5-10
   - **Style**: Points
   - **Color Transformer**: RGB8
   - **Alpha**: 0.8
   - **Decay Time**: 0.0 (for real-time updates)

4. Add MarkerArray for Bounding Boxes

1. Click `Add` → `By display type` → `MarkerArray`
2. Configure MarkerArray:
   - **Topic**: `/tracking_markers`
   - **Marker Array**: Enable all namespaces
   - Leave other settings as default

5. Add Robot Pose (Optional)

1. Click `Add` → `By display type` → `PoseStamped`
2. Configure PoseStamped:
   - **Topic**: `/robot_pose`
   - **Shape**: Arrow
   - **Arrow Length**: 0.5
   - **Arrow Radius**: 0.1
   - **Color**: Blue (0, 0, 255)

6. Optimize View Settings

1. **Camera View**:
   - Set view to `Third Person Follower` or `Orbit`
   - Adjust distance to see both point cloud and bounding boxes clearly
2. **Background Color**:
   - In Global Options, set Background Color to dark (0, 0, 0) for better contrast

7. Save Configuration

1. File → Save Config As → `orb_tracker_visualization.rviz`
2. Next time, load with: `rviz2 -d orb_tracker_visualization.rviz`

Running the Complete System

Terminal 1: Launch your simulation/camera

```bash
# Your existing camera/simulation launch command
ros2 launch your_package your_simulation.launch.py
```

Terminal 2: Run the Enhanced ORB Tracker

```bash
# Navigate to your workspace
cd ~/auv_ws
source install/setup.bash

# Run the tracker node
ros2 run your_package_name tracker_node.py
```

Terminal 3: Launch RViz with configuration

```bash
# Launch RViz
rviz2 -d orb_tracker_visualization.rviz
```

Expected Visualization

In RViz you should see:

1. **Green/Colored Points**: Feature points from the point cloud
2. **Red Semi-transparent Boxes**: Bounding boxes around tracked features
3. **Blue Arrow** (optional): Robot pose and orientation

In CV2 Window you should see:

1. **Side-by-side display**: Original frame (left) and processed frame (right)
2. **Linear Velocity**: Current speed in m/s
3. **Roll, Pitch, Yaw**: Orientation angles in degrees
4. **Feature Coordinates**: First 5 tracked feature point coordinates
5. **Green Bounding Boxes**: Around tracked features

Troubleshooting

If PointCloud2 is not visible:

1. Check if `/feature_cloud` topic is being published: `ros2 topic echo /feature_cloud`
2. Verify the Fixed Frame is set to `map`
3. Try changing Color Transformer to `Intensity` or `AxisColor`

If MarkerArray is not visible:

1. Check if `/tracking_markers` topic is being published: `ros2 topic echo /tracking_markers`
2. Verify all namespaces are enabled in MarkerArray display

If velocity is still showing 0:

1. Ensure ORB-SLAM3 is properly tracking (green features should be visible)
2. Move the camera/robot to generate motion
3. Check that poses are being published: `ros2 topic echo /robot_pose`

Performance Tips:

1. Reduce PointCloud2 size if visualization is slow
2. Set appropriate Decay Time for smooth visualization
3. Limit the number of displayed feature coordinates in the code

Topics Being Published:

- `/robot_pose` - Current camera/robot pose
- `/robot_velocity` - Linear velocity information
- `/feature_cloud` - 3D point cloud of tracked features
- `/tracking_markers` - Bounding box markers
- `/debug_image` - Processed image for debugging

Use `ros2 topic list` to verify all topics are being published correctly.