مستندسازی مراحل انجام پروژه

پروژه: راه‌اندازی دوربین TurtleBot3 در ROS2

مقدمه

در این سند، مراحل انجام‌شده در پروژه به‌صورت گام‌به‌گام و منظم مستندسازی شده است. هدف این مستند، ارائه یک راهنمای شفاف از مسیر صحیحی است که طی آن شبیه‌سازی ربات TurtleBot3 اجرا شده و تصویر دوربین ربات با موفقیت دریافت و نمایش داده شده است.

۱ فعال‌سازی محیط ROS2

در ابتدای هر جلسه کاری، محیط ROS2 و workspace پروژه فعال شد:

```
source /opt/ros/humble/setup.bash
source ~/ros2_ws/install/setup.bash
```

نتیجه: محیط ROS2 Humble و workspace پروژه با موفقیت فعال شد.

۲ انتخاب مدل ربات

برای اطمینان از وجود سنسور دوربین، مدل مناسب ربات انتخاب شد:

```
export TURTLEBOT3_MODEL=waffle_pi
```

نتیجه: مدل waffle_pi به‌عنوان ربات پیش‌فرض تنظیم شد.

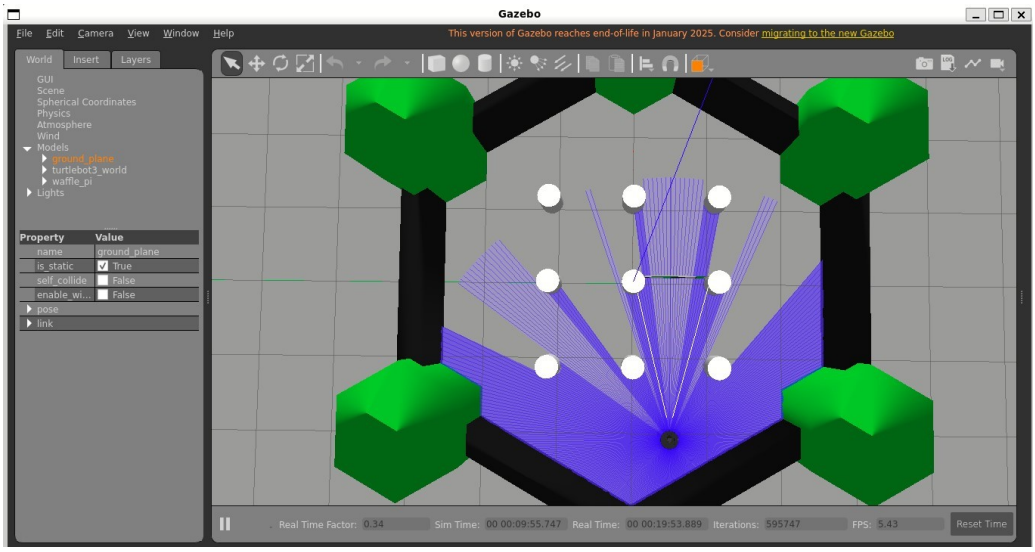
۳ اجرای شبیه‌ساز Gazebo

شبیه‌ساز Gazebo به‌همراه ربات اجرا شد:

```
ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

نتایج:

- اجرای موفق Gazebo
- ظاهر شدن ربات در محیط
- فعال شدن سنسورها (دوربین، لیزر، IMU)



شکل ۱: محیط شبیه‌سازی Gazebo و ربات TurtleBot3

۴ بررسی تاپیک تصویر دوربین

برای بررسی انتشار تصویر:

```
ros2 topic list | grep image
```

تاپیک خروجی:

```
/camera/image_raw
```

۵ بررسی نرخ انتشار تصویر

```
ros2 topic hz /camera/image_raw
```

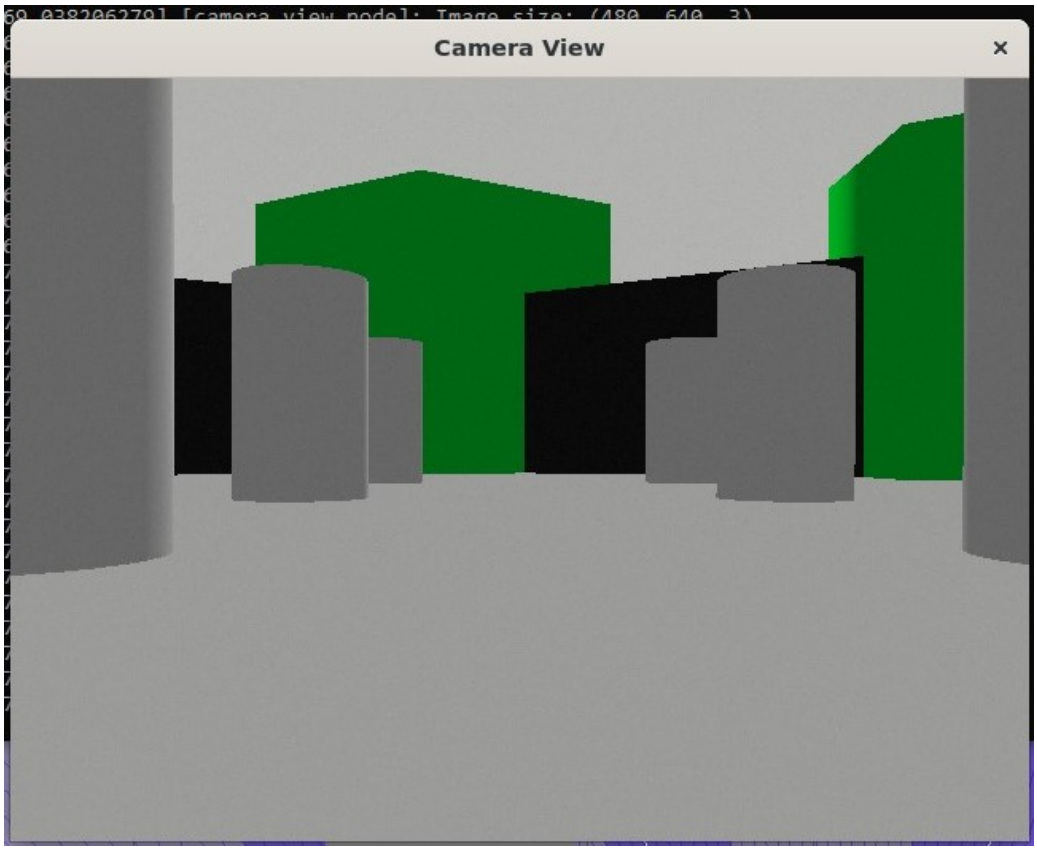
نتیجه: تصویر با نرخ حدود 30Hz منتشر می‌شود.

۶ اجرای نود نمایش تصویر

```
ros2 run my_camera_pkg camera_view_node
```

نتایج:

- باز شدن پنجره OpenCV
- نمایش زنده تصویر دوربین



شکل ۲: نشان دادن تصویر دریافتی از دوربین ربات TurtleBot3

۷ مسیر کامل داده تصویر

Gazebo

↓

TurtleBot3 (waffle_pi)

↓

Camera Sensor

↓

/camera/image_raw

↓

camera_view_node

↓

OpenCV Window

جمع‌بندی

در این پروژه، شبیه‌سازی TurtleBot3 با موفقیت انجام شد و داده تصویر دوربین از طریق ROS2 دریافت و در یک نود سفارشی نمایش داده شد.

مراحل بعدی:

- تشخیص فلش یا علامت در تصویر
- کنترل حرکت ربات
- طراحی سناریوی کامل ناوبری