Openpose with ROS & GAZEBO simulation

2020.4.20-5.11 WE-02 Sprint

AI Software Team

Ismael El Houas Ghouddana

Table of contents

- Introduction
- Gazebo and Animated worlds
- TRT Pose in Gazebo
- ROS implementation on Jetson Xavier
  - Opencv Openpose VGG19
  - NVIDIA TRT Pose estimation
  - Improve Pose estimation performance
  - Openpose Service and Client
- Conclusions
- References

Introduction

- Goal:
  - R&D development process using Gazebo
  - Gazebo simulation as a development tool for the AI team.
  - Speed up Drone simulation Targeting to ~30 fps
  - Openpose VGG19 & TRT Pose solutions developed in Gazebo implement it with ROS on Jetson Xavier

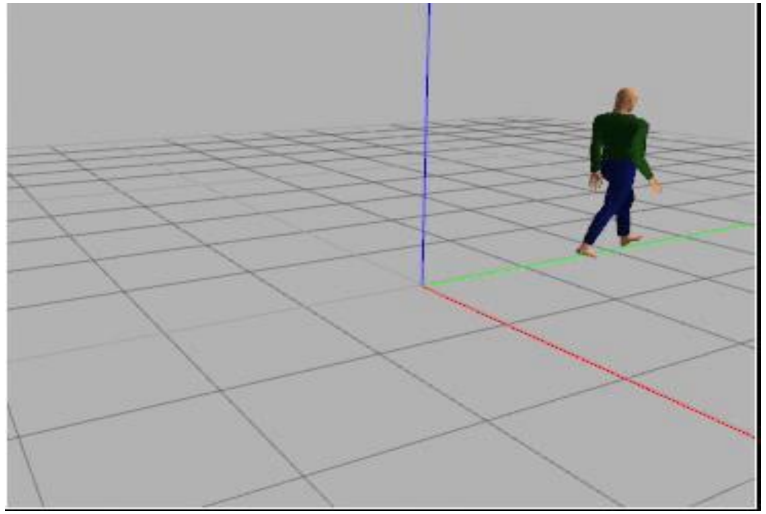
Gazebo and Animated worlds

- Robot simulator tool in complex indoor and outdoor environments
- Complete toolbox for any Roboticist:
  - Test algorithms
  - Design robots
  - AI systems training
  - ROS compatibity
- Benefits
  - Availability (24/7)
  - Productivity
  - Faster R&D -> Production
  - Complex cases
  - No real damage
  - Innovation

Gazebo and Animated worlds

- Animated worlds for AI team R&D development

```
….
<script>
    <trajectory id="0" type="walking">
     <waypoint>
      <time>0</time>
      <pose>0 2 0 0 0 -1.57</pose>
     </waypoint>
….
```



- Huge number of combinations:
  - 1 target walking
  - 2 people crossing
  - …
- How to automate? -> WORK SMARTER NOT HARDER
- **Embedded Ruby** to create dynamic animated worlds
  - Compile erb file to generate world **-> erb -T 1 walk.world.erb > walk.world**

Drone simulation with TRT Pose estimation

- Last sprint development of ROS package with TRT pose estimation
- No test it in real situation
- Use Tao's drone simulation for solution development
- Speed up Pose estimation from Openpose VGG19
  - 3fps -> 30 fps
- Lower pick up rate

ROS implementation on Jetson Xavier

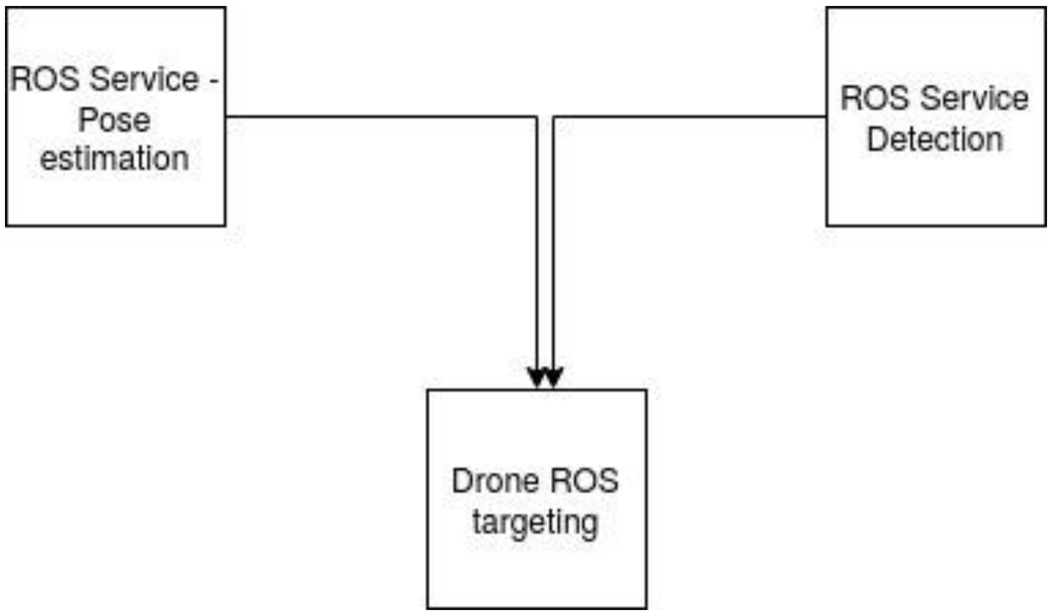
- Solutions developed in Gazebo need to be tested on production environment Jetson Xavier
- Rosbags of single target moving horizontally at 6, 10 and 12 m
- Results:
  - Openpose VGG19
    - ~3 fps
    - Big model (200MB)
    - Good pick up rate at 6 m < d
  - TRT Pose estimation (Resnet18)
    - ~30 fps
    - Small model (85 MB)
    - Pick up rate decreases at 6 m < d

ROS implementation on Jetson Xavier

Improve Pose estimation performance

- Resnet18 accuracy need to be improved at longer distances
  - Hypothesis -> TRT optimization with any other model (Microsoft Resnet50, 121...)
  - Problem -> Customized model from NVIDIA not all models have the same layers
  - Solution -> Improve resolution with same model
  - Results -> Accurate results until 10 m, pick up rate does not decrease

ROS implementation on Jetson Xavier



Openpose Service and Client

- Pack Openpose solutions as ROS services
  - TRT Pose estimation
  - Openpose VGG19
- Work together with Detection service (Harry)
- Input -> D435 ROS topic
- Output -> Keypoint ROS msgs
- Benefits:
  - Modular
  - Work on demand
  - Interchangeable

Conclusion

- Gazebo as a toolbox for all members of AI team
- Improves Productivity, Innovation and R&D to product process
- Need for a Host machine -> Laptops for members
- TRT pose and VGG19 show good results and possible future solutions
- Use on demand with ROS service
  - Combine with detection
  - Use in determined situations
- Improved tracking using pose features
- Drone targeting more robust in complex indoor and outdoor situations

References

- **Gazebo:** http://gazebosim.org/
- **Ruby**:
  - https://answers.gazebosim.org//question/14063/how-can-i-declare-a-variable-in-modelsdf-file/
  - http://gazebosim.org/tutorials?tut=kinematic_loop&cat=
  - https://en.wikipedia.org/wiki/ERuby#Different_implementation_tags_comparison
  - https://answers.gazebosim.org//question/8305/is-it-possible-to-use-variables-in-sdf-file/
- **Results videos**: https://console.aws.amazon.com/s3/buckets/wecorp-ai/videos/?region=eu-west-2&tab=overview
- **Learn Opencv**: https://github.com/spmallick/learnopencv/tree/master/OpenPose-Multi-Person)
- **Openpose:** https://github.com/CMU-Perceptual-Computing-Lab/openpose)
- **NVIDIA TRT Pose:** https://github.com/NVIDIA-AI-IOT/trt_pose
- **Pytorch:** https://forums.developer.nvidia.com/t/pytorch-for-jetson-nano-version-1-4-0-now-available/72048)
- **Torch2trt:** https://github.com/NVIDIA-AI-IOT/torch2trt)
- **Jetcam solve:** https://forums.developer.nvidia.com/t/hello-camera-displayed-nothing/81475/4)
- **ROS course:** https://rds.theconstructsim.com/r/039bdaae5b5c/ros_developers_live_class_79_using_nvidia_jetson_nano_with_ros/)