3D POSE ESTIMATE OF AN OBJECT IN THE ENVIRONMENT

Presented by Kamalnath and Ahilesh

INTRODUCTION

Markerless Object Pose Estimation Using YOLO and Depth Sensors in ROS2

Problem Statement:

- Object pose estimation essential for robotic manipulation
- Traditional methods require markers attached to objects
- Need for flexible, markerless solutions in dynamic environments

Our Approach:

- Combines YOLOv5 for detection with depth sensing for 3D information
- Implemented in ROS2 for robotic integration
- Works in both RGB-only and RGB-D camera setups

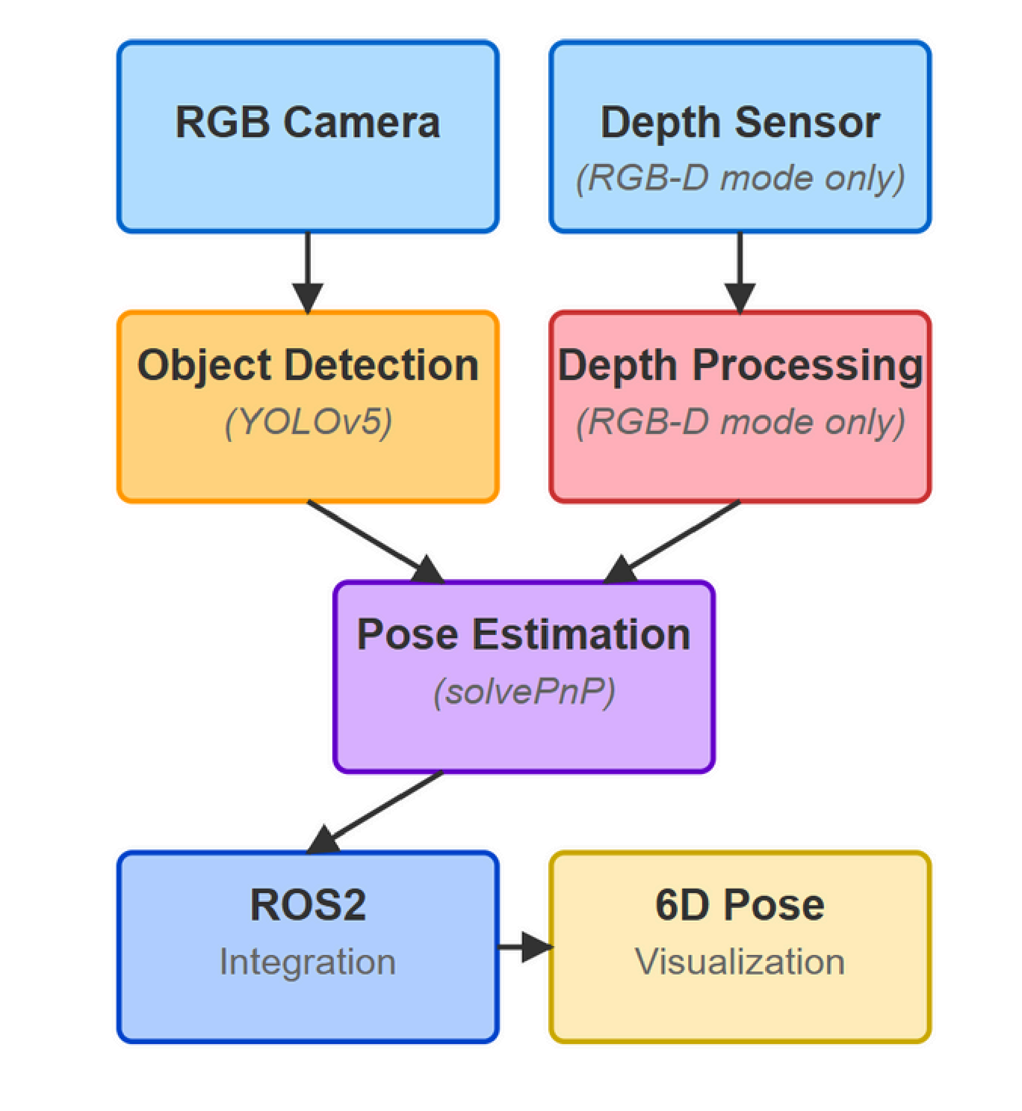
SYSTEM ARCHITECTURE

Modular Pipeline:

- Image Acquisition Module: Captures RGB and depth data
- Object Detection Module: Uses YOLOv5 to identify objects
- Depth Processing Module: Extracts reliable depth information
- Pose Estimation Module: Computes 6D object poses
- ROS2 Integration: Publishes poses and visualization

Key Innovations:

- Statistical depth filtering for improved reliability
- Class-specific 3D model generation
- Unified workflow for both RGB and RGB-D modes



IMPLEMENTATION METHODS

Object Detection:

- YOLOv5 trained on custom dataset
- 200 annotated images with various bg
- Transfer learning from pre-trained COCO model

Depth Integration:

- Region-based depth analysis for reliable measurements
- Handling invalid readings (zeros, NaNs)
- Model scaling based on measured depth

Pose Estimation Algorithm:

- Define 3D object based on object class
- Establish 2D-3D point correspondences
- Solve PnP problem using OpenCV
- Convert results to ROS2 coordinate frames

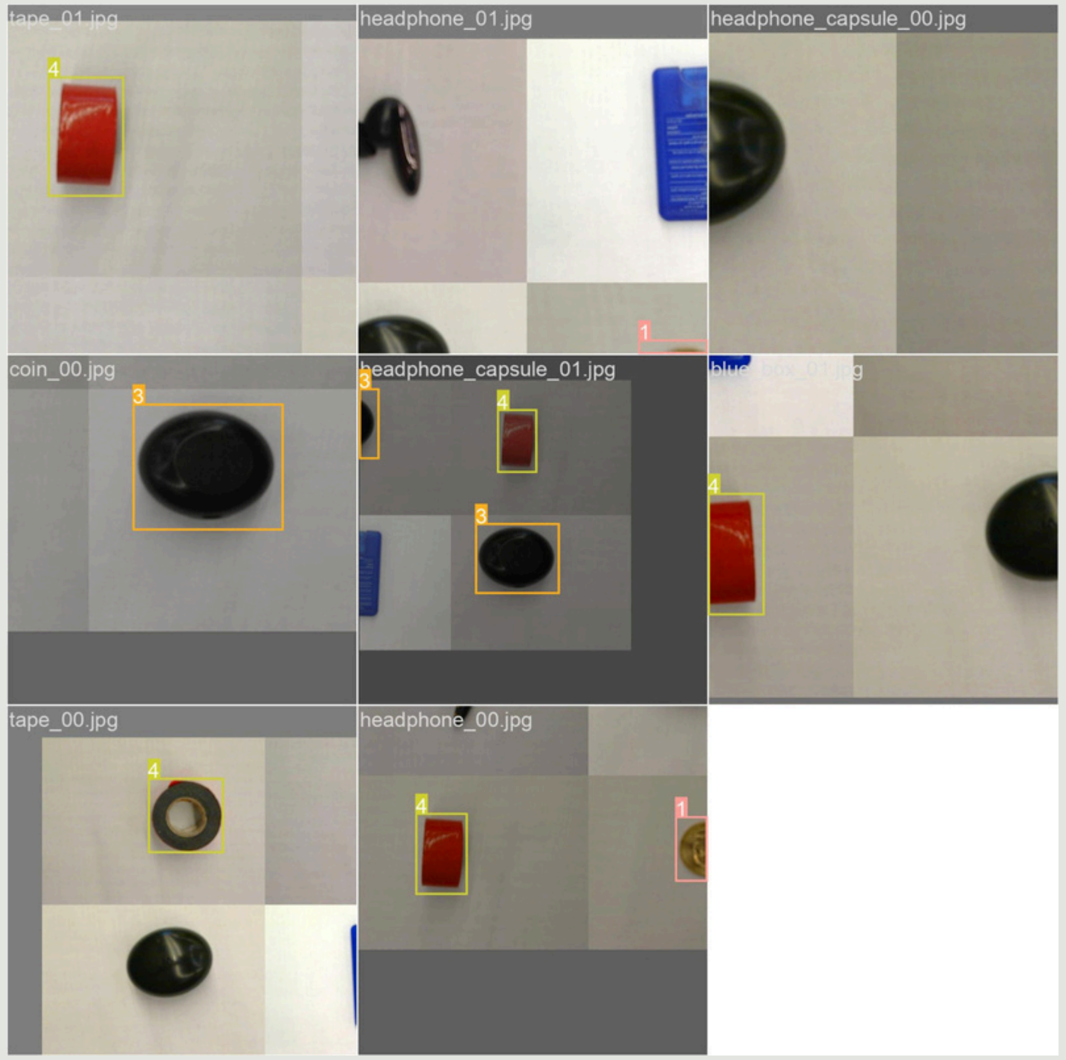
DATASET & TRAINING

Custom Dataset:

- 5 object classes: blue box, coin, tape, earphone, earphone capsule
- 80% training / 20% validation split
- Multiple angles and lighting conditions

Training Results:

- 100 epochs of YOLOv5 training
- mAP@0.5: 0.746 across all classes
- Best performance: blue box (0.995 precision)
- Challenges with small objects (coin: 0.497 precision)



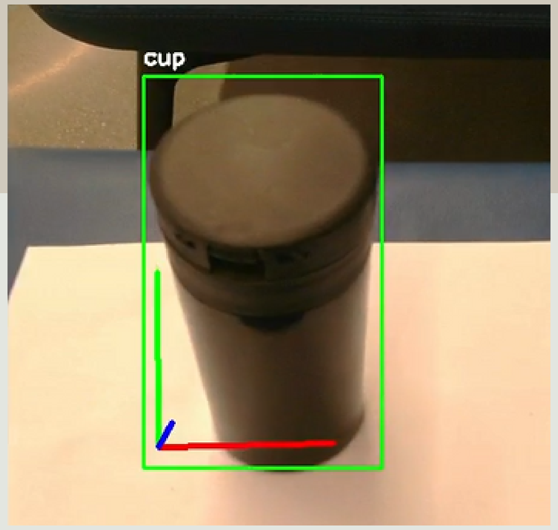
RESULTS: RGB VS. RGB-D

RGB-Only Mode:

- 15-20 Hz operation on standard hardware
- Good x-y position accuracy
- Limited depth (z-axis) accuracy
- 5-15% distance error

RGB-D Mode:

- 8-12 Hz operation (depth processing overhead)
- Superior position accuracy (<2cm error at 1m)
- Handles scale ambiguity
- Challenges with reflective surfaces




```
[DETECTED] CUP
  Position (x, y, z): [   0.069207     -0.1023    -0.54017]
  Orientation (Rodrigues): [ -0.0025659     -3.2109     0.15657]
  Orientation (Quaternion): [ 0.00079763     0.99815   -0.048673    0.036557]
```

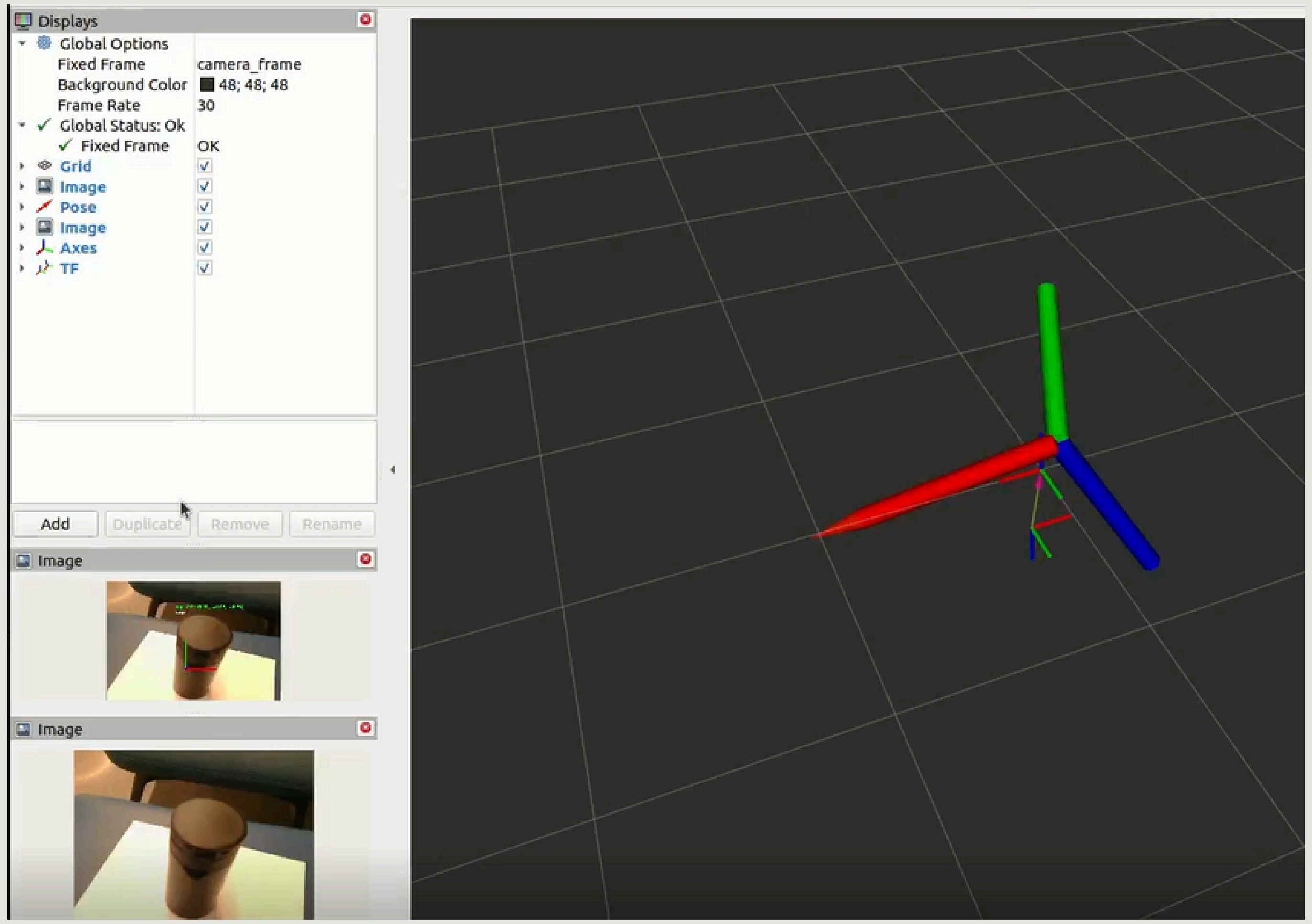
ROS2 INTEGRATION & VISUALIZATION

ROS2 Components:

- Camera interface nodes
- Pose estimation nodes
- TF frame broadcasting
- RViz visualization

Published Topics:

- Object pose messages (geometry_msgs/PoseStamped)
- Visualization markers
- Annotated images with pose axes



LIMITATIONS & CHALLENGES

Current Limitations:

- Background complexity affects detection
- Limited viewpoint generalization
- Fixed object dimensions assumption
- Depth sensor issues with reflective materials

Technical Challenges Encountered:

- Time synchronization between RGB and depth
- Handling invalid depth readings
- Pose stability for symmetrical objects
- Real-time performance trade-offs

CONCLUSION & FUTURE WORK

Key Contributions:

- Practical markerless pose estimation system
- Flexible RGB/RGB-D operation modes
- Complete ROS2 integration
- Demonstrated viability on standard hardware

Future Directions:

- Learning from limited data (few-shot learning)
- Instance segmentation for improved boundaries
- Temporal filtering for pose stability
- Robotic manipulation integration for validation
- Multi-view fusion for improved accuracy