Transformation of Physical Environments through Integrated Sensor Fusion and Computer Vision for Interactive Digital Recreation

Jesús Alejandro Gómez Bautista

Monterrey, N.L., México

Introduction

This poster presents a framework for capturing, detecting, and reconstructing 3D objects using Azure Kinect. Inspired by [1], it integrates depth sensing, YOLO-based detection, and simplified reconstruction to enable real-time virtual simulations, providing a foundation for teleoperated robotic interaction.

Hypotesis

The system integrates advanced computer vision and object detection, using YOLO and Azure Kinect to bridge physical and virtual environments. This enables accurate object identification, localization, and manipulation, with dynamic adaptability to simulation changes.

Experiment Design

The experiments progress through four phases: data collection and model training, real-world evaluation, object reconstruction, and integration with a simulated Tiago robotic arm. Inspired by Gustavo De Los Rios Alatorre's thesis, the reconstruction leverages canonical model deformation and 3D pose estimation. To meet time constraints, the focus was narrowed to a minimum viable demonstration, simplifying the reconstruction process.

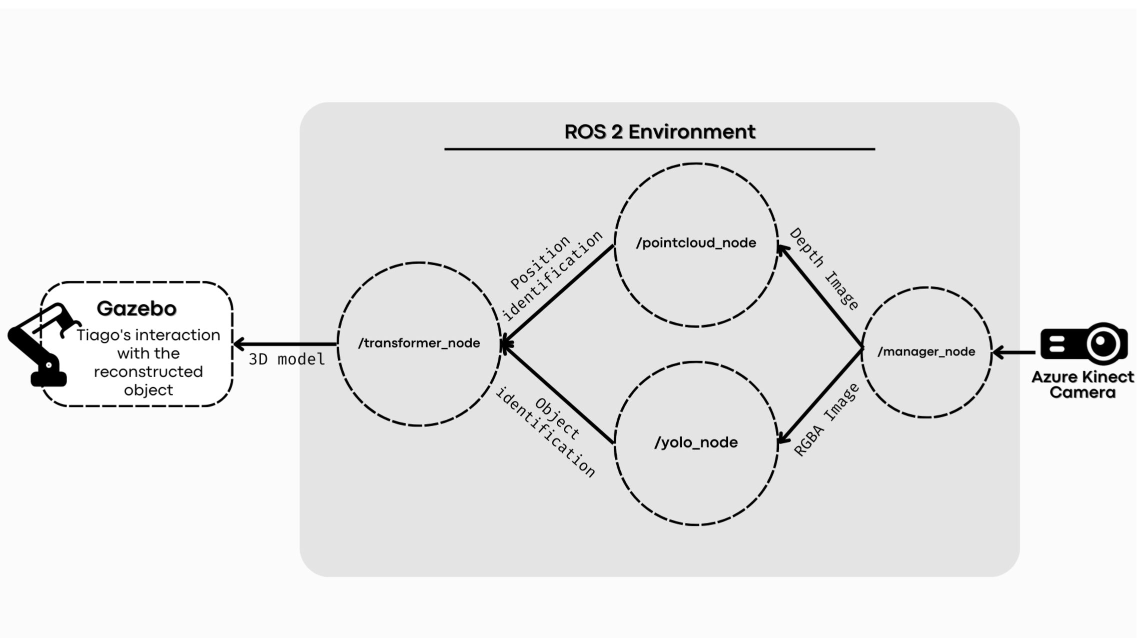

Figure 1: Experiment elements, nodes, and relations

PHASE I: Collection

In this initial phase, the environment is represented using depth and color data captured by the Azure Kinect. A dedicated 'manager_node' handles the camera feed and publishes the relevant topics for the other nodes.

1. Depth and Color Image Publishing:
   - The 'manager_node' initializes the Azure Kinect camera and streams both depth and RGB images.
   - Depth images are published for the 'env_node' to generate a point cloud, while RGB images are published for the 'yolo_node' to perform object detection.
2. Synchronization and Broadcasting:
   - The 'manager_node' ensures synchronized publishing of depth and color data, providing a consistent feed for the other nodes.
   - It broadcasts camera transforms to position the Kinect in the simulated environment.

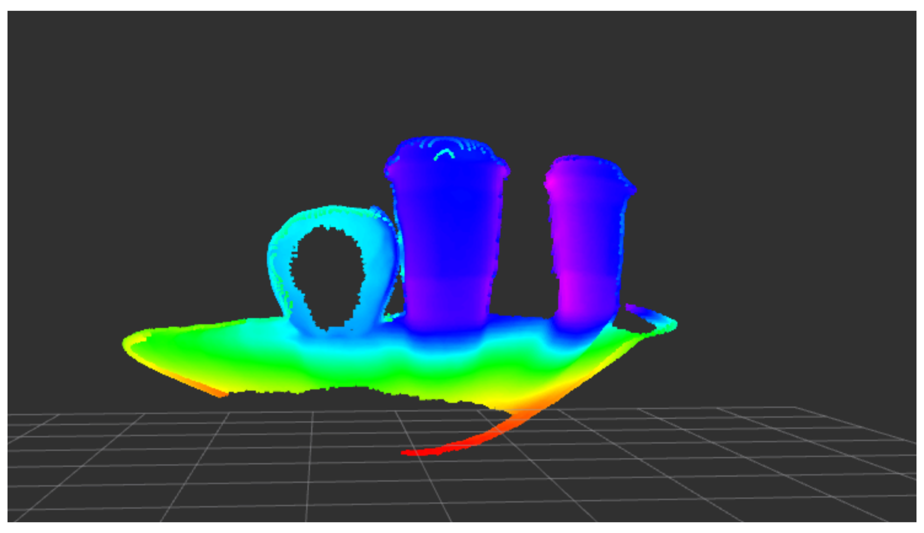

Figure 2: Operation of the point cloud node

PHASE II: Identification

The 'yolo_node' detects and classifies individual objects using the color image stream from the 'manager_node'.

1. Real-Time Object Detection:
   - The 'yolo_node' uses a YOLOv8 model trained on specific object classes (e.g., cans, coffee, apples) to detect and classify objects.
   - Bounding boxes are generated for each detected object in the RGB image.
2. Publishing Detection Results:
   - The node publishes annotated RGB images with bounding boxes and text files containing object detection results.

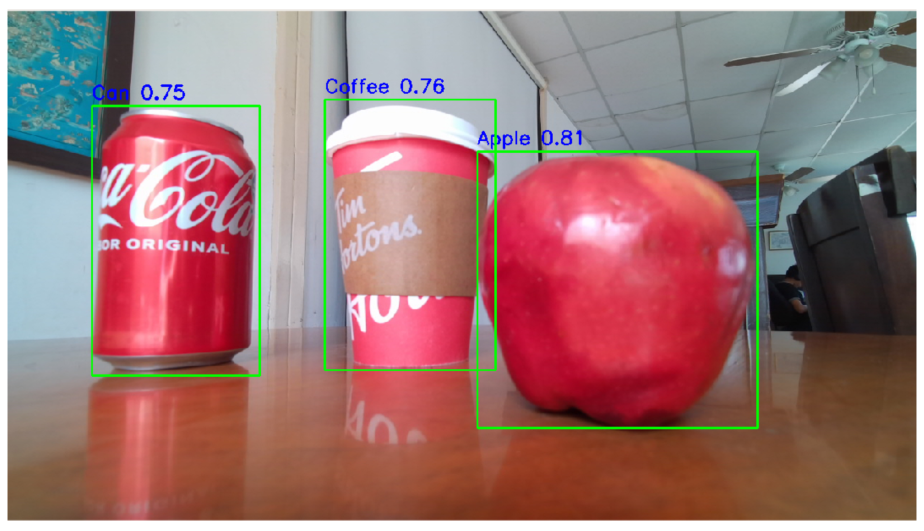

Figure 3: Operation of the YOLO node

PHASE III: Reconstruction

The 'transformer_node' performs 3D object reconstruction using canonical models and pose estimation.

1. Canonical Model Alignment:
   - The node uses canonical STL models for each object class (e.g., cans, coffee, apples).
   - Detected objects from the 'yolo_node' and segmented point clouds from the 'env_node' are used to align and deform these models.
2. Pose Estimation:
   - The reconstructed models are positioned and oriented using pose estimation to match their location in the real world.
3. Publishing Reconstructed Models:
   - The reconstructed 3D models are published to the ROS environment for interaction in simulation.

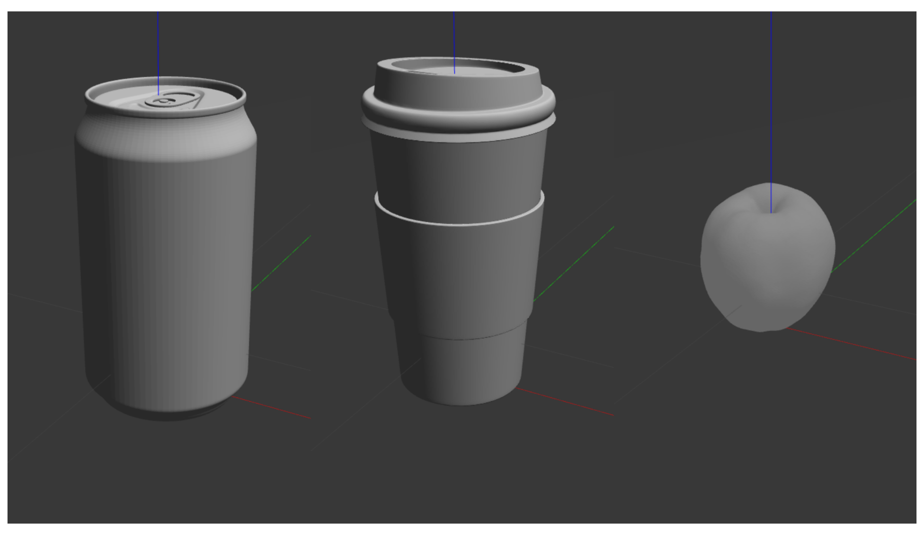

Figure 4: Operation of the transformer node

PHASE IV: Simulation

This phase focuses on integrating the reconstructed objects into a simulation environment, allowing interaction with the Tiago robot [2].

1. Simulated Environment:
   - Reconstructed objects are imported into Gazebo for simulation.
   - The Tiago robot interacts with these models based on user commands.
2. Feedback and Iterative Refinement:
   - Feedback from the simulation is used to refine the object reconstruction and grasping processes.

Results Interpretation

The project has made substantial progress, achieving 98% functionality. The YOLO detection node effectively identifies cans, coffee containers, and apples, while Gazebo integrates virtual models of real objects. Core objectives like object detection, spawning, and interaction are complete, though grasping is still under refinement. The system bridges physical and virtual environments with improved precision and speed. While advanced alignment was omitted for simplicity, key goals of point cloud generation and interaction were successfully met, providing a strong basis for future enhancements.

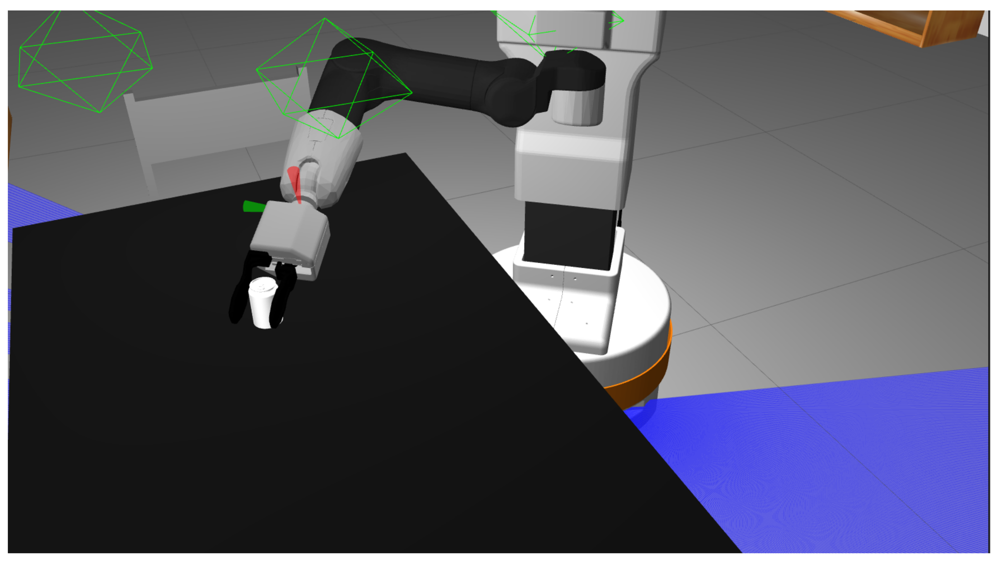

Figure 5: Final performance of the demonstration

Conclusion

This work showcases the digitalization of physical objects using Azure Kinect for robotic simulation. Future efforts will focus on improving interaction capabilities and integrating automated commands for better realism and usability in robotic applications.

References

[1] G. de los Ríos Alvarez, "3-d detection tracking for semi-deformable objects," Master's thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey, Monterrey, Nuevo León, 2024.

[2] F. Martín, *A Concise Introduction to Robot Programming with ROS2*. CRC Press, 2023.

Downloads

PAPER REPOSITORY