



Automated Chess Board Setup: Model-Based Manipulation of Chess Pieces

Sunni Magan
Amy Mendiola
Nicolas Torres

Background

Objective: Develop a system in which a robotic manipulator can autonomously set up a chess board.

Goal: To accurately identify each chess piece and place it in the correct position on the board.

This involves detecting, pose estimation, grasping, and moving the pieces based on their type (king, rook, bishop, etc.)

Environment

Robot: FETCH

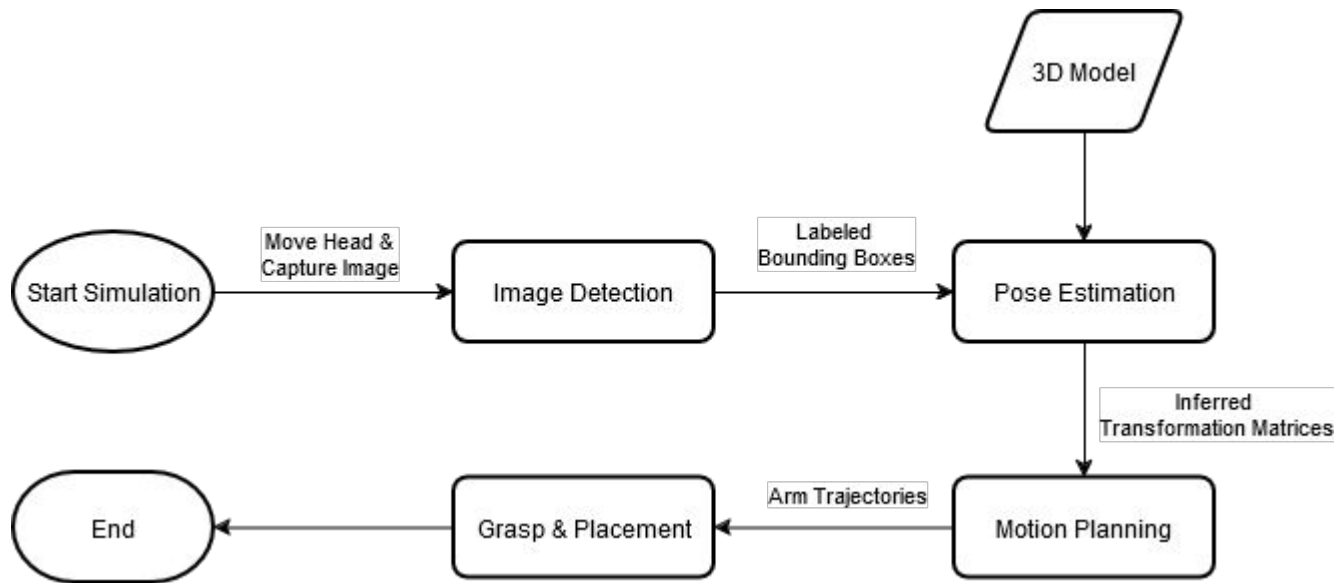
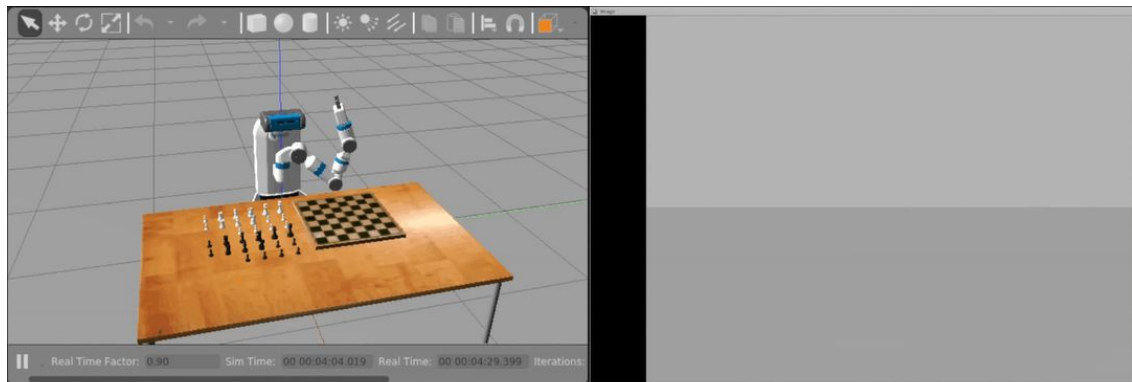
Simulation: Gazebo

Image Detection: OpenCV

Pose Estimation: [MegaPose](#)

Motion Planning: MoveIt

Workflow



Computer Vision

OpenCV:

Input:

RGB image and depth image.

Method:

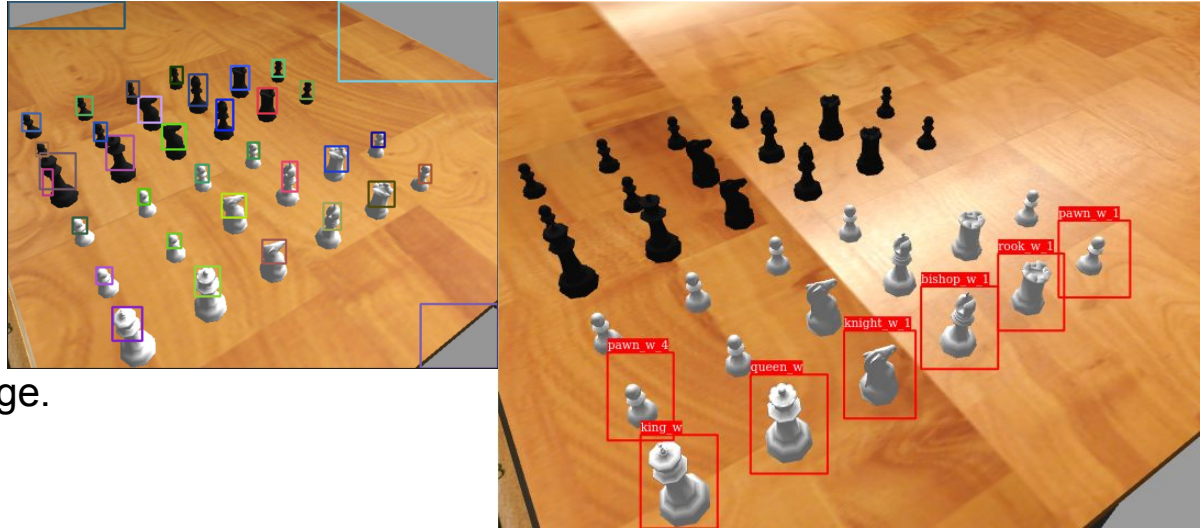
Preprocessing: Convert depth image to grayscale, blur it, and detect edges using Canny Edge Detection.

Contour Detection: Find contours, approximate them as polygons, and compute bounding boxes.

Bounding Box Adjustment: Extend bounding boxes and check image boundaries.

Visualization: Draw bounding boxes with random colors on the RGB image.

Output: Display processed RGB image with contours and bounding boxes.



Pose Estimation

MegaPose:

Input:

An RGB (or RGB-D) image, a region of interest, and a CAD model.

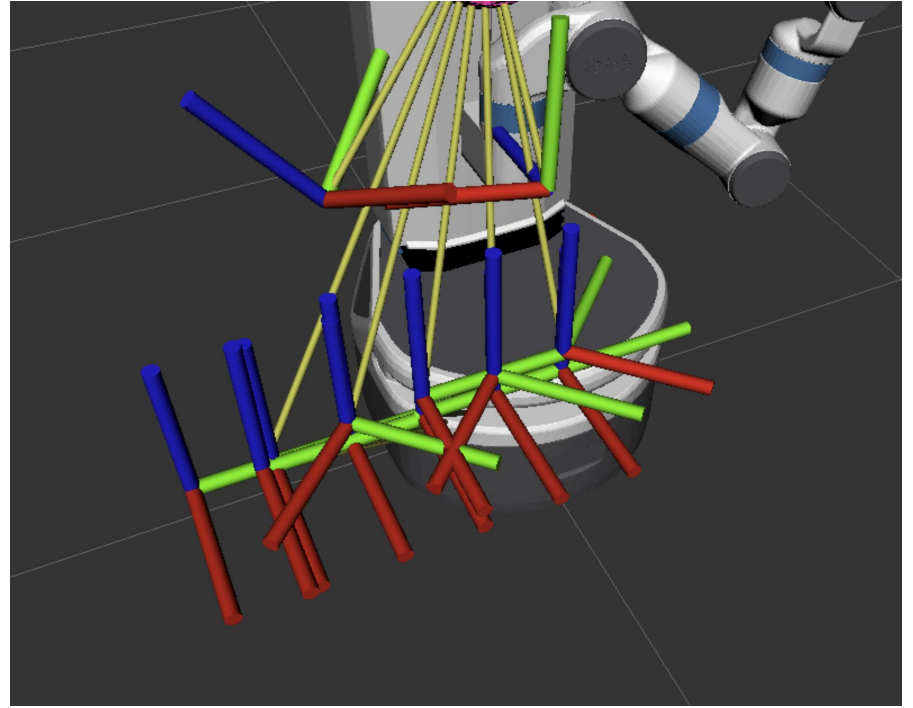
Pipeline:

1. Detect objects in 2D.
2. Generate coarse pose hypotheses using multiple synthetic renderings.
3. Refine the best hypothesis through iterative rendering and comparison.



Performance

- Translation Error
 - CAD Model
- Rotational Error
 - Symmetry of pieces
- Pawns
 - CAD Model



Motion Planning

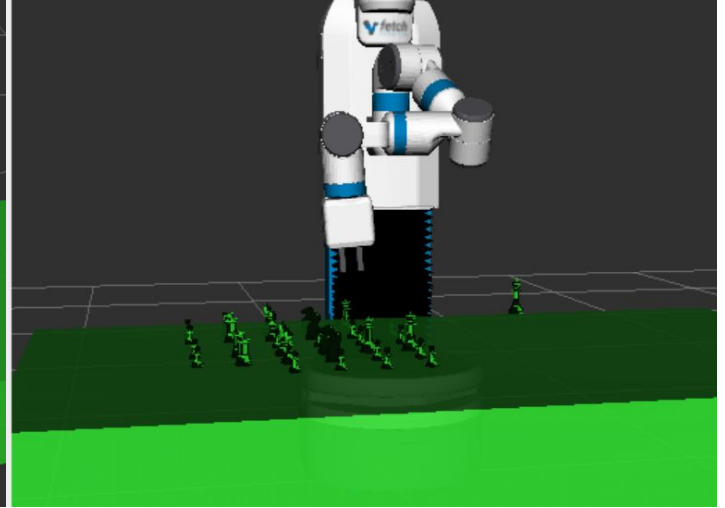
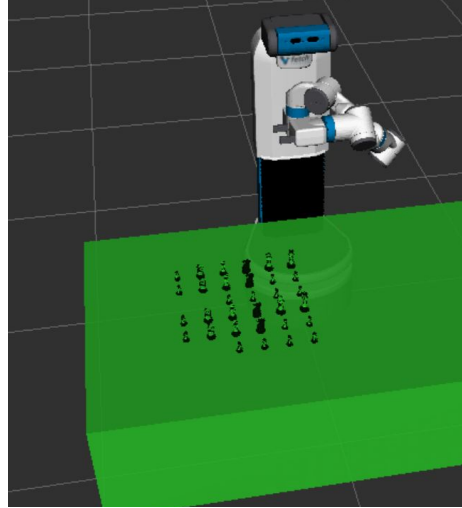
MoveIt!

Input:

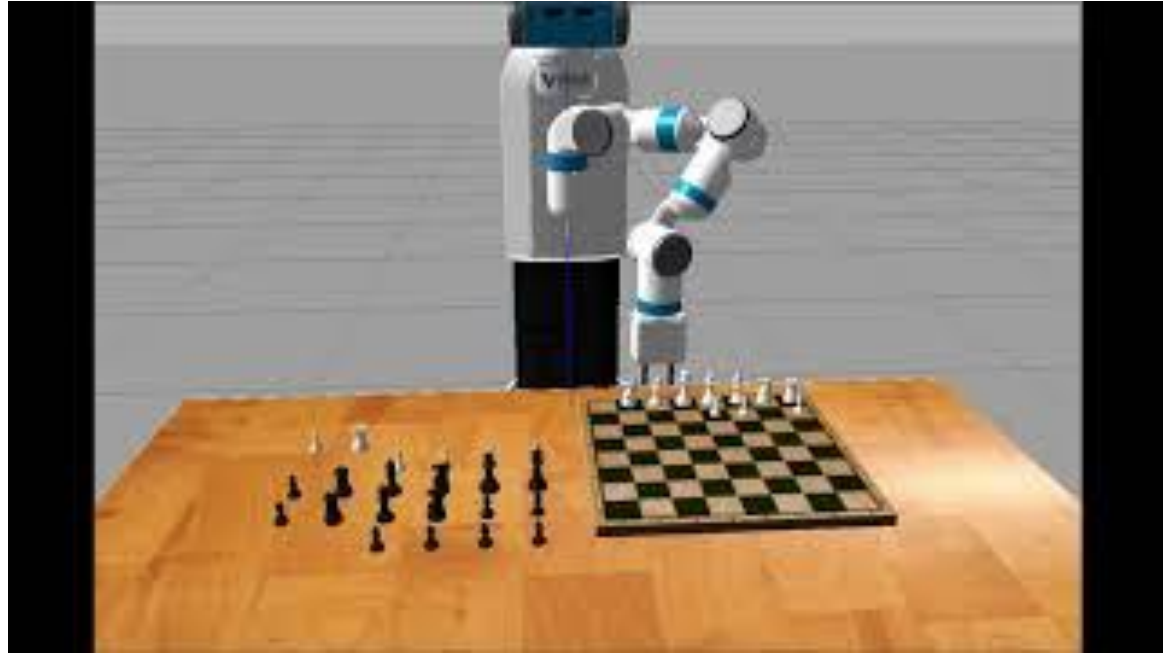
- Piece pose estimation

Design:

- Collision Box/Mesh for table and pieces
- All pieces gripped from above
- For each piece:
 - Remove collision mesh for current piece
 - Map and Execute Pick and Place of the current piece (using approach pose and pick/place pose)
 - Update pieces collision location



Demo



Questions?

