Project Proposal: Vision-Based Robotic Affordance Detection and Feature Identification

1. Introduction & Motivation

For robots to interact effectively with the physical world, they must possess more than simple object classification capabilities. While identifying an object as a "mug" or a "tool" is useful for inventory, it is insufficient for physical manipulation. To grasp, insert, or sort objects, a robot must identify specific **affordances**—functional geometric features such as handles, graspable rims, flat suction surfaces, or insertion holes.

This project proposes the development of a **Vision-Based Feature Identification System**. By utilizing a high-fidelity physics simulation to generate synthetic RGB-D (Red, Green, Blue + Depth) data, the system will bridge the gap between visual perception and robotic action. The system will autonomously detect meaningful geometric features on various rigid bodies and calculate their 3D spatial coordinates for potential robotic interaction.

2. Problem Statement

Traditional robotic manipulation relies heavily on hard-coded coordinates or known object poses. If an object is unstructured or randomly placed, standard "blind" robots fail. Current Computer Vision systems often stop at 2D Bounding Boxes (detecting *where* an object is in an image), failing to provide the specific *grasp points* required for interaction.

This project addresses the need for a perception system that can extract specific geometric features (holes, handles, surfaces) from a monocular camera feed and translate them into actionable 3D world (x, y, z) coordinates relative to a robot's base.

3. Project Objectives

1. **Simulation Environment:** To design a physics-based simulation environment containing a robotic workspace and a set of test objects (industrial parts and household items).
2. **Synthetic Data Acquisition:** To implement a virtual RGB-D camera system capable of streaming synchronized color and depth data.
3. **Feature Extraction:** To develop computer vision algorithms capable of identifying specific geometric features:
    - **Apertures (Holes)** for insertion tasks.
    - **Planar Surfaces** for suction/picking tasks.
    - **Protrusions (Handles)** for grasping tasks.
4. **3D Coordinate Mapping:** To implement mathematical transformation logic that converts 2D pixel coordinates from the vision system into 3D world coordinates.

4. Proposed Methodology

The project will follow a four-stage pipeline, implemented entirely within a simulated environment to ensure safety, repeatability, and precise ground-truth validation.

Phase 1: Environment Setup (Simulation Layer)

I will utilize **PyBullet**, a physics engine used widely in machine learning and robotics research.

- **Workspace:** A tabletop environment will be simulated.
- **Objects:** Three distinct object classes will be modeled to represent common manipulation challenges:
    - *Class A (Washer/Nut):* Represents insertion tasks requiring center-point detection.
    - *Class B (Box/Container):* Represents logistics tasks requiring surface segmentation.
    - *Class C (Mug/Tool):* Represents complex geometry requiring handle/edge detection.

Phase 2: Perception System (Data Layer)

A virtual camera will be calibrated and positioned within the simulation.

- **RGB Extraction:** Capturing high-resolution color data for feature segmentation.
- **Depth Buffer Processing:** Extracting the Z-buffer data and linearizing it to obtain true Euclidean distance values for every pixel in the scene.

Phase 3: Algorithm Implementation (Processing Layer)

The core logic will utilize **OpenCV** and **NumPy** for geometric analysis without relying on heavy "black box" deep learning models, ensuring the logic is explainable and computationally efficient.

- **Aperture Detection:** Utilization of Hough Circle Transforms to identify circular geometries and calculate centroids.
- **Surface Detection:** Contour analysis and area thresholding to identify the largest planar surface for suction targets.
- **Handle/Edge Detection:** Canny Edge Detection coupled with morphological operations to isolate graspable protrusions.

Phase 4: Coordinate Transformation (Action Layer)

The system will perform a "Pixel-to-World" projection.

1. Identify the Feature (u, v) in the 2D image.
2. Sample the Depth at (u, v)
3. Apply the Camera Intrinsic Matrix to project the point into 3D space.
4. Apply the Camera Extrinsic Matrix (Rotation/Translation) to map the point to the World Coordinate System.

5. System Architecture

The proposed system consists of the following modular blocks:

1. **Input Module:** Synthetic Camera Stream (PyBullet).
2. **Preprocessing Module:** Noise reduction, Grayscale conversion, Region of Interest (ROI) masking.
3. **Feature Logic Module:** Specific algorithms for Hole, Surface, and Handle detection.
4. **Math Module:** Matrix transformations for 2D $\rightarrow$ 3D mapping.
5. **Visualization Module:** A graphical overlay drawing "Target Vectors" (approach axes) on the live video feed to demonstrate successful identification.

6. Tools and Technologies

- **Programming Language:** Python 3.13
- **Simulation Engine:** PyBullet (Physics and Rendering)
- **Computer Vision Library:** OpenCV (cv2)
- **Numerical Computation:** NumPy (Matrix operations)
- **Visualization:** Matplotlib / PyBullet Debug GUI

7. Expected Deliverables

By the conclusion of this project, the following will be delivered:

1. A functional Python-based simulation script.
2. A demonstration video showing real-time detection of features as objects are moved within the simulation.
3. A technical report detailing the algorithmic approach, mathematical coordinate transformations, and accuracy analysis.