GROUP MEMBERS

Syed Darain Hyder Kazmi
Muhammad Uzair Warraich
Asadullah Baig

EXPLORING THE **MYCOBOT 280 PI** 3D VISION KIT

Product study for Intro to Computer Vision Study of Elephant Robotics' Artificial Intelligence 3D Kit focusing on MyCobot 280 Pi with 3D vision; authorship and course metadata included.

Why Study the **MyCobot Kit?**

Integrates **vision** and manipulation for teaching, research, and prototyping

1. **Combines computer vision and robot manipulation in one integrated kit**
   Single platform for sensing and actuation

2. **Real example covering camera models, calibration, depth sensing, object detection and 3D localization, coordinate transforms, and robot kinematics**
   Hands-on coverage of listed course topics

3. **Designed for education, research, and rapid prototyping**
   Flexible for labs, papers, and demos

4. **Demonstrations include AI-driven pick-and-place, sorting, stacking**
   Practical tasks showing end-to-end flow

5. **Our goal: explain hardware, software, and vision algorithms used in the kit and connect each part to core computer vision concepts**
   Teach theory through system components

What Is the MyCobot 280 Pi 3D Vision Kit?

Complete desktop robotic vision platform from Elephant Robotics

Hardware

- MyCobot 280 Pi 6-DOF collaborative robot arm
- 3D RGB-D camera (color and depth)
- End-effectors: parallel gripper and suction cup
- Workstation accessories: mounts, fixtures, calibration board
- Software environment and sample projects for AI and 3D vision

Capabilities

- 2D and 3D object recognition and localization
- Vision-guided grasping and pick-and-place
- Demos of classical computer vision and deep-learning-based AI

Kit Variants and Our Focus

Raspberry Pi based MyCobot 280 Pi 3D Vision Kit

Elephant Robotics offers multiple controller versions: MyCobot 280 Pi, MyCobot 280 Jetson Nano, M5Stack, Arduino

List of available controllers for the AI / 3D Vision kit

Presentation focus: MyCobot 280 Pi 3D Vision Kit

Primary kit covered in this presentation

Emphasis: Raspberry Pi based control and 3D RGB-D camera for depth perception

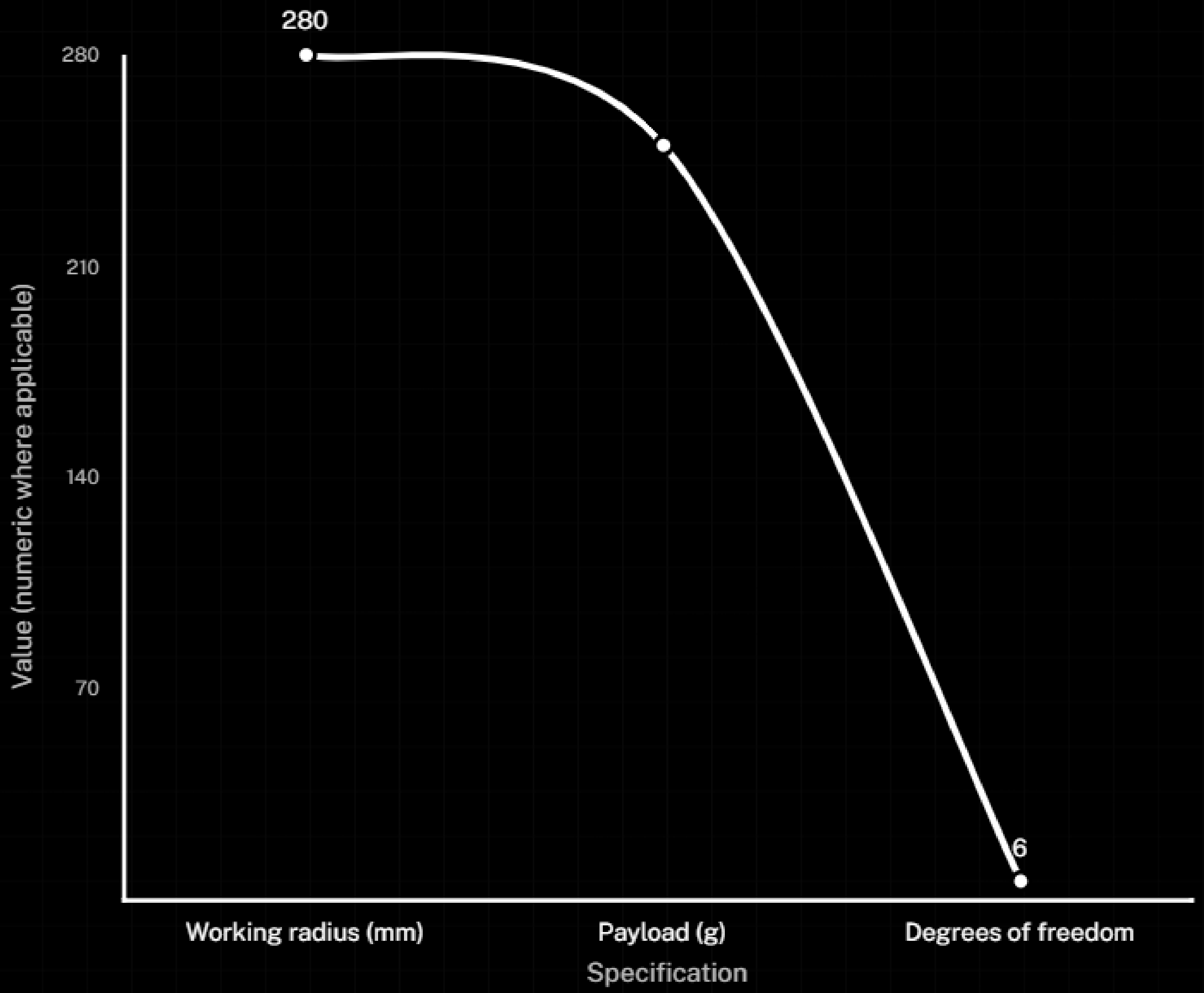
Key hardware highlights and sensing capability

Includes sample computer vision and AI tasks from Elephant Robotics documentation

Prebuilt CV and AI examples provided with the kit

MyCobot 280 Pi: Key Specifications

Compact 6-DOF arm with Raspberry Pi 4B control



3D RGB-D Camera in the Kit

Integrated color and per-pixel depth for vision and 3D sensing

1

Sensor type: RGB-D camera (integrated color and depth)

Provides synchronized RGB and per-pixel depth

2

Image properties: HD color; lower-resolution depth

RGB up to HD; depth lower but time-synced

3

Working range: tens of centimeters to a few meters

Operates from close range up to several meters

4

Primary roles: object detection, 3D position estimation, point cloud creation

RGB for detection; depth for 3D location and scene cloud

5

Mounting options: eye-to-hand above workspace or fixed on/near robot

Either overhead (static) or robot-mounted with fixed offset

Other Physical Components in the 3D Vision Kit

Essential hardware for reliable demos, calibration, and training

End-effectors: parallel gripper and suction cup

Parallel gripper for general objects; suction cup for flat, smooth surfaces

Mechanical fixtures: camera mount and calibration board

Camera mount/stand, robot base plate or bracket, checkerboard for calibration

Workspace accessories: marked surface and training objects

Marked work surface for demos; blocks, shapes, cards for training

Power and connectivity: adapters and USB cables

Power adapters for robot and compute unit; USB cables for camera to Pi

Software and Development Environment

Linux on Raspberry Pi 4B; AI options on Jetson variants

Operating system: **Linux** on Raspberry Pi 4B

Lightweight runtime for MyCobot control and vision

Core libraries: **Python 3**, OpenCV, NumPy, SciPy

Classic computer vision and linear algebra on Pi

ROS / ROS2: modular robot and vision integration (optional)

Use for scalable, message-driven architectures

Deep learning: PyTorch and TensorFlow supported on Jetson

AI detection requires more powerful variants like Jetson Nano

Elephant Robotics resources: official SDKs and demos

MyCobot control, vision-guided picking, object recognition

System Architecture: Perception to Motion

Layered hardware and software stack with clear data flow

1

Hardware: Sensing

3D RGB-D camera captures images and depth for environment perception.

2

Hardware: Compute

Raspberry Pi runs perception and control software and orchestrates processing.

3

Hardware: Actuation

MyCobot servos execute joint commands and operate the gripper.

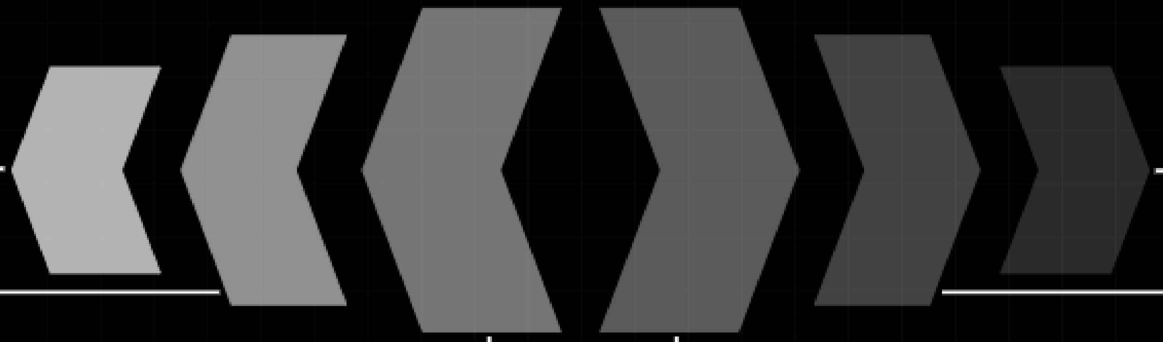
4

Software: Perception

Image acquisition, CV and AI detection/segmentation, and 3D localization.

Coordinate Frames in the Kit

How camera, base, end-effector, and world frames connect



Camera frame {Cam}

Vision outputs object pose T_cam^obj in the camera coordinate system.

Robot base {Base}

Reference for planning; known transform T_base^cam maps camera to base.

End-effector {EE}

Computed pose T_base^ee from forward kinematics for control and grasping.

World / Table {World}

Global/table reference used for scene context and measurements.

Key transforms

$T\_base^{cam,\ T\_base}ee$, T_cam^obj preserved exactly as given.

Control flow

Detect object: obtain $T\_cam^{obj;\ compute}$ $^{T\_base}obj = T\_base^{cam\ *}$ $^{T\_cam}obj$; plan T_base^{ee} $^{relative\ to\ T\_base}obj$ to grasp.

Camera Model and Depth Geometry

Pinhole projection and back-projection for 3D reconstruction

$$u = f_x \frac{X}{Z} + c_x, \quad v = f_y \frac{Y}{Z} + c_y$$

- (u, v) – pixel coordinates
- (X, Y, Z) – 3D point in $\{Cam\}$ frame
- f_x, f_y – focal lengths in pixels
- c_x, c_y – principal point (optical center)

- **Depth image**:
    - For each pixel (u, v), sensor returns a depth value Z

Calibration Procedures for Accurate 3D Perception

Intrinsic camera and extrinsic hand–eye steps to minimize 3D errors

1

Intrinsic camera calibration

Use checkerboard images and OpenCV calibrateCamera to estimate camera matrix K and distortion parameters from multiple images

2

Capture calibration pattern at poses

Place checkerboard at several known robot poses and capture synchronized images and robot pose data

3

Solve hand–eye (extrinsic) calibration

Compute transformation T_{base}^{cam} by aligning robot poses with camera observations to determine camera to robot base transform

4

Validate and refine

Assess 3D position errors and repeat captures or refine calibrations to reduce grasping errors and misalignment

Vision Pipeline for the MyCobot 3D Kit

From synchronized RGB and depth capture to pick and place execution

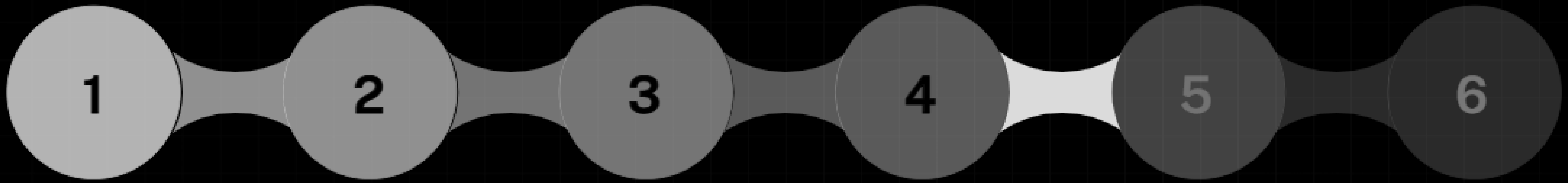


Image acquisition

Synchronized RGB and depth capture for aligned visual and range data

Preprocessing

Color space conversion and depth noise filtering to clean inputs

Object detection and recognition

Detect objects using classical computer vision or AI models

3D localization

Read depth and back-project pixels to 3D coordinates

Coordinate transformation

Transform points using T_base^{cam} to robot base frame

Robot control

Compute end-effector pose and execute pick and place

Classical 2D Detection Methods

Practical techniques for color and contour detection

1. **Color-based segmentation: convert RGB to HSV and use inRange**
   Segment color ranges to produce a binary mask

2. **Morphological operations: erosion and dilation**
   Remove noise and close small holes in masks

3. **Contour analysis: use findContours to extract outlines**
   Compute area, shape, centroid, and bounding rectangle

4. **Applications: demos for object picking and sorting**
   Sort by color or basic shape for simple automation

AI-Based Object Detection: robust detection and 3D localization

Pre-trained deep learning models, frameworks, and 2D to 3D integration

1

Model types: CNNs for generic or custom object detection

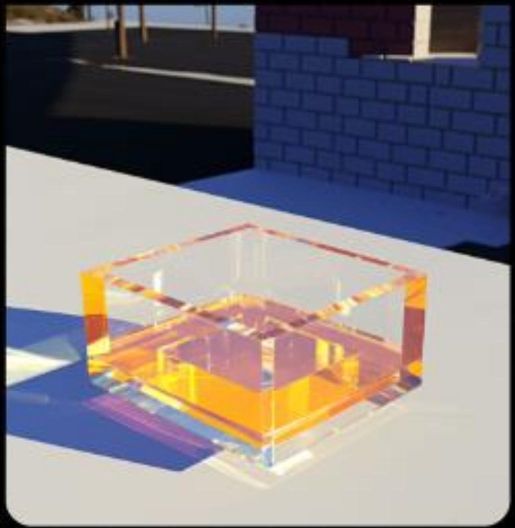
Run pre-trained deep learning models for detection and recognition

2

Frameworks: PyTorch, TensorFlow, ONNX Runtime

Supports common deep learning runtimes for inference

3



Advantages: robust to lighting and complex backgrounds

Detects objects without distinct colors or simple shapes

4

2D to 3D integration: 2D bounding boxes plus depth to compute 3D positions

Depth adds position and, with extra processing, possible orientation

From 2D Centroid to 3D Object Position

Depth sampling, back-projection, and transform to robot base

1

2D centroid

Use detected 2D region centroid (u_c, v_c) from bounding box or detector

2

Depth sampling

Read $Z_c = Z(u_c, v_c)$ or average a small neighborhood

3

Back-projection

$X_c = (u_c - c_x) Z_c / f_x$, $Y_c = (v_c - c_y) Z_c / f_y$, keep Z_c

4

Homogeneous point

Form $p_{cam} = X_c, Y_c, Z_c, 1^T$

5

Transform to base

Compute $p_{base} = T_{base}^{cam} \cdot p_{cam}$ to obtain object position in robot frame

6

Tooling

Example programs in the kit automate these computations

Robot Kinematics and Motion Control

6-DOF kinematics, IK usage, and MyCobot motion sequence

1 **6-DOF kinematics** give full 3D position and orientation control of the gripper

Defines pose control in space

2 Forward kinematics: input joint angles q = q1,...,q6, output end-effector pose T_base^ee

Compute pose from joint angles

3 Inverse kinematics (IK): input desired T_base^ee, output joint angles q that realize the pose

Solve joint values for a target pose

4 MyCobot API handles IK; joint command: send_angles(q1,...,q6, speed)

Joint-space motion command

5 MyCobot task command: send_coords(x,y,z,rx,ry,rz, speed, mode)

Task-space motion with pose vector

6 Typical motion strategy: move to home, pre-grasp above target, descend to grasp, close gripper or suction, lift, move to destination, open gripper

Standard pick and place sequence

Internal Software Workflow for a 3D Picking Demo

Integrated perception, 3D geometry, and robot control

Initialization

Load camera intrinsics and T_base^cam; initialize robot to home pose

Perception Loop

Acquire RGB and depth frames; run object detection; compute 2D centroid and sample depth for each target

3D Point Computation

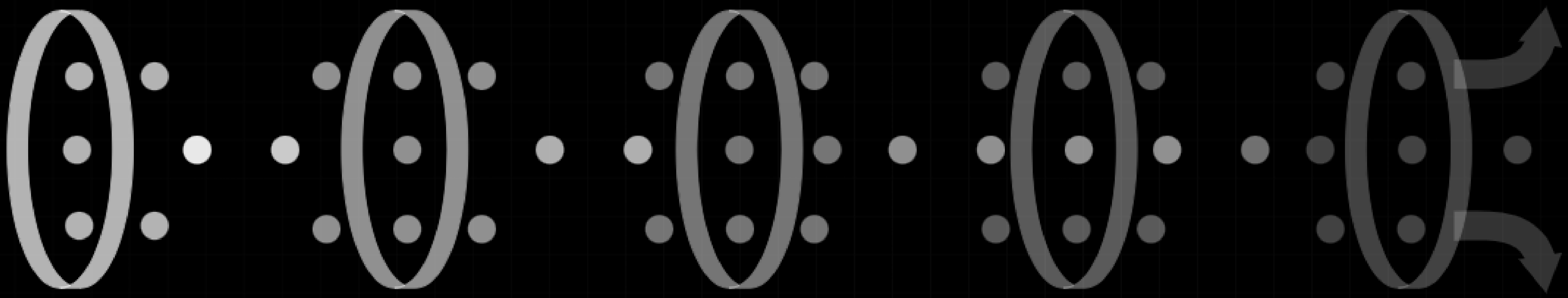
Transform sampled depth pixel to a 3D point in the Base frame

Planning and Motion

Generate pre-grasp and grasp poses, call IK, send joint commands, control gripper open and close

Repeat or Stop

Continue until all objects removed or user stops the program



Educational Value of the AI 3D Kit

Hands-on practice mapping to course learning outcomes

Hands-on **computer vision** fundamentals

Camera calibration, distortion, depth sensing, 3D reconstruction

Perception: segmentation, detection, recognition

Run detection and recognition neural networks on embedded platforms

Robot vision and control

Hand-eye coordination, homogeneous transformations, visual servoing

System-level integration

Combine perception, planning, and actuation in a closed loop

Practical exposure to theory and tasks

Bridges course concepts to real integration challenges

Summary & Key Takeaways

1

The MyCobot 280 Pi 3D Vision / AI 3D Kit is a compact platform integrating a 6-DOF collaborative robot arm, a 3D RGB-D camera, end-effectors, calibration tools, demo objects, and a software stack for CV + AI + robot control.

Complete hardware and software kit including 6-DOF arm and 3D RGB-D camera

2

It demonstrates how 2D images plus depth can guide 3D robot actions and how computer vision algorithms are used in real robotic systems.

Shows practical integration of 2D images and depth for robot actions

3

For Intro to CV, the kit is a complete case study moving from pixels → 3D geometry → robot motion.

End-to-end case study from images to robot motion


myCobo AI Kit
3D

THANKS 4 UR PATIENCE