Università di Pisa

Department of Computer Science

Master Degree in Computer Science

Artificial Intelligence

A Data Collection System for Ground-Truth Mass Distribution Learning

Candidate

Michele Morisco

Supervisors

Gianpaolo Palma

Daniela Giorgi

A.Y. 2025-2026

Contents

List of Figures

List of Tables

Abstract

Extended Reality, including Virtual Reality, Augmented Reality, and Mixed Reality, enables immersive interaction but still lacks realistic physical behavior in virtual objects, such as mass and weight. In particular, current virtual environments cannot largely represent and simulate intrinsic physical properties such as mass distribution and weight, which are essential for achieving truly immersive and physically consistent experiences. This thesis, developed to assist with some objectives of the EU-funded Social and Human-centered XR project, proposes a system for estimating mass distribution to generate high-quality ground-truth data for learning-based applications. The proposed system integrates both hardware and software components to enable scalable, accurate, and repeatable data acquisition. The hardware setup consists of a multi-camera acquisition box with four calibrated cameras, a passive hand-shaped gripper equipped with force sensors, and a modular 3D-printed object with configurable internal mass distributions. These components are coordinated through a central control unit. The software framework is divided into two modules: a synchronized data acquisition module and a computer vision module. The following performs camera calibration, object pose estimation, and gripper fingertip tracking using fiducial markers, producing precise affine transformation matrices. The system is designed to automate data acquisition for estimating inertial properties, with a focus on accuracy and scalability. Experimental results demonstrate an acceptable baseline of the proposed approach in estimating object and gripper poses and highlight the system's potential to support learning-based models that incorporate physical properties into virtual environments. This work contributes a solid baseline for data acquisition in XR applications and lays the basis for future research on embedding realistic physical behavior into virtual objects.

Abstract

La realtà estesa, che comprende realtà virtuale, realtà aumentata e realtà mista, consente un'interazione immersiva, ma resta carente di comportamenti fisici realistici negli oggetti virtuali, come la massa e il peso. In particolare, gli ambienti virtuali attuali non sono in grado di rappresentare e simulare in modo esaustivo le proprietà fisiche intrinseche quali la distribuzione della massa e il peso, elementi essenziali per ottenere esperienze veramente immersive e fisicamente coerenti. Questa tesi, sviluppata per supportare gli obiettivi del progetto europeo Social and Human-centered XR, propone un sistema per la stima della distribuzione della massa al fine di generare dati di riferimento di alta qualità per applicazioni di apprendimento. Il sistema proposto integra componenti hardware e software per consentire un'acquisizione di dati scalabile, accurata e ripetibile. La configurazione hardware è composta da una scatola per l'acquisizione multicamera con quattro camere calibrate, una pinza a forma di mano dotata di sensori di forza e un oggetto modulare stampato in 3D per le distribuzioni di massa interne configurabili. Questi componenti sono coordinati tramite un'unità di controllo. Il framework software è suddiviso in due moduli: un modulo di acquisizione dati sincronizzato e un modulo di computer vision. Quest'ultimo esegue la calibrazione delle camere, la stima della posa dell'oggetto e delle dita della pinza utilizzando i fiducial marker, producendo matrici di trasformazione affine precise. Il sistema è progettato per automatizzare l'acquisizione dei dati per la stima delle proprietà inerziali, con particolare attenzione all'accuratezza e alla scalabilità. I risultati sperimentali dimostrano una base di riferimento accettabile per l'approccio proposto alla stima delle pose dell'oggetto e della pinza ed evidenziano il potenziale del sistema per supportare modelli basati sull'apprendimento che incorporano proprietà fisiche in ambienti virtuali. Questo lavoro fornisce una solida base di partenza per l'acquisizione dei dati nelle applicazioni XR e pone le basi per future ricerche sull'integrazione di comportamenti fisici realistici in oggetti virtuali.

Chapter 1

Introduction

In the present scenario, Extended Reality (XR) is an umbrella term including a spectrum of immersive technologies that integrate real and virtual environments through computer-generated content and interactive systems. XR includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), each offering distinct yet complementary approaches to user interaction. Together, these technologies introduce novel paradigms for engaging with data, entertainment, and commercial applications. Over the years, XR has demonstrated significant progress and has achieved notable outcomes across various domains. However, despite these advancements, several technical, economic, and usability challenges remain that must be addressed to enable widespread adoption. XR has limitations, including device resource constraints, poor scalability, and high cost. In addition, it lacks convincing interaction between physical and virtual elements.
In recent years, the latter has not had many works in the literature; the current state of virtual objects is primarily rigid and geometric, allowing only basic manipulations. With the progress of AI, we have achieved

important results in visual realism, such as shape manipulation and high-quality rendering, but there is still an absence of interactive properties such as weight and flexibility. To achieve realistic behaviors, real-world physical properties must be extracted, learned, and embedded in virtual objects. [15]
The Social and Human-centered XR (SUN)[17] project is a European Union-funded project by the Horizon Europe funding programme. SUN aims to overcome these limitations in XR development by offering scalable, cost-effective solutions. This project aims to minimize limitations in the interaction between physical and virtual objects by replicating an object's physical and semantic properties from the real world to a virtual one and assigning the same behavior to a virtual element.
This work has assisted the activities of the project and some of its objectives. In particular, in this thesis, we propose the design and development of a system for mass distribution estimation, aimed at collecting high-quality ground truth data for learning-based applications. The system is structured around two primary components: hardware and software.
From a hardware perspective, the system is designed to ensure controlled, repeatable, and well-labeled data acquisition. It comprises three main elements: (i) a multi-camera image acquisition box equipped with four calibrated cameras, (ii) a passive hand-shaped gripper integrated with force sensors to enable controlled object manipulation, and (iii) a modular 3D printed object with configurable internal mass distribution. These components are coordinated through a central Control Unit (CU), which hosts the software toolkit and manages the overall system operation.
The software component is organized into two main modules. The first module is responsible for synchronized data acquisition within the image acquisition box. The second module performs computer vision tasks, including: (i) estimation of camera intrinsic and extrinsic parameters using ArUco markers placed within the acquisition environment, (ii) object pose estimation based on markers located on the object surface, and (iii) estimation of the gripper fingertip poses using ArUco markers mounted

on the gripper's fingertips.
During the acquisition phase, image data are collected and subsequently processed by the second module. The system performs camera calibration and pose estimation, after which it estimates both the pose of the modular object and the gripper fingertips, generating the corresponding affine transformation matrices.
The proposed system aims to enable fast, scalable, automatic data acquisition for learning-based applications. The primary objective of this thesis is to design and implement the system and to evaluate the accuracy of the estimated parameters, to minimize errors in the resulting transformation matrices.
The thesis is organized as follows. Chapter 2 describes the theoretical background to provide the necessary foundation for addressing the concepts explored in this thesis. Section 2.1 explores the fundamental concepts related to the inertial properties of a rigid body, while Section 2.2 discusses perspective camera models and lens distortions. Chapter 3 reviews the state of the art in inertial property estimation and in fiducial markers. Sections 3.1 and 3.2 present existing methods for estimating inertial parameters, while Section 3.3 analyzes available ground-truth datasets, and Section 3.4 introduces fiducial marker-based approaches, which form the basis for the proposed system.
Chapter 4 describes the hardware components of the system. Section 4.1 presents the image-based acquisition box, which defines a controlled workspace for data collection. Section 4.2 introduces the modular 3D-printed object, designed to allow multiple internal mass configurations while preserving the same external geometry. Section 4.3 describes the passive hand-shaped gripper used to manipulate and hold the object consistently and repeatably.
Chapter 5 details the software toolkit developed to manage the hardware system and process the collected data. Section 5.1 describes the data acquisition module, while Section 5.2 presents the computer vision module. In particular, Sections 5.2.1, 5.2.2, and 5.2.3 describe the algo-

rithms used for camera calibration, camera pose estimation, object pose estimation, and fingertip pose estimation, respectively.
Chapter 6 presents and discusses the experimental results, including both visualizations and quantitative evaluations. Section 6.1 reports the results related to camera calibration accuracy. Section 6.2 presents the results on camera pose estimation accuracy. Section 6.3 describes the accuracy of 3D-printed object pose estimation. Finally, Section 6.4 focuses on evaluating the pose estimation of the gripper's fingertips.
In the end, Chapter 7 concludes the thesis by summarizing the main contributions and outlining possible directions for future work.

Chapter 2

Background

Since the proposed system relies on visual sensing and physical manipulation to estimate an object's position in three-dimensional space, accurate estimation and tracking of these properties depend critically on reliable visual perception. This perception is typically modeled using the perspective camera model, which introduces geometric distortions in the projected image. Consequently, accurate camera modeling and calibration are required to compensate for these distortions and to ensure precise spatial measurements.

This chapter provides the theoretical background relevant to the development of the proposed system.

In particular, it focuses on the inertial properties of rigid bodies and the perspective camera model. Section 2.1 introduces the fundamental concepts of a rigid body's inertial properties, with a focus on key theoretical definitions. Section 2.2 reviews the perspective camera model and discusses common issues, including lens distortions.

This background provides the necessary foundation for understanding and addressing the concepts explored in this work.

2.1 Inertial properties of a rigid body

The inertial properties of a rigid body characterize its resistance to changes in its state of motion when subjected to external forces and torques. Specifically, they quantify the body's opposition to both translational and rotational accelerations.
The key inertial properties of a rigid body include its mass, the position of its center of mass (CoM), and its inertia tensor. The mass m of a rigid body is a scalar quantity that represents the total amount of matter in the body and controls its resistance to translational acceleration. Together with the CoM and the inertia tensor, it provides a complete description of how the body responds dynamically to applied forces and moments.
In classical mechanics, it is described through Newton's second law

$$F = ma \tag{2.1}$$

For a body with continuous mass distribution and density $\rho(\mathbf{r})$, we have

$$m = \int_V \rho(\mathbf{r})\, dV, \tag{2.2}$$

where V denotes the volume of the body and $\mathbf{r}$ is the position vector of a mass element relative to the chosen reference point.
The CoM is the point at which the total mass of the body may be considered to be concentrated for the purpose of analyzing its translational motion. The following formula describes CoM:

$$r_{CoM} = \frac{1}{m} \int_V \mathbf{r}\rho(\mathbf{r})\, dV \tag{2.3}$$

When external forces act on a rigid body, its translational motion is equivalent to that of a point mass located at the CoM. The inertia tensor describes the distribution of the body's mass with respect to a reference point and determines its resistance to rotational acceleration. For a rigid body expressed in a given reference frame, the inertial tensor $\mathbf{I} \in \mathbb{R}^{3\times3}$ is defined by

$$\mathbf{I} = \int_V \rho(\mathbf{r}) \left(\|\mathbf{r}\|^2 \mathbf{I}_3 - \mathbf{r}\mathbf{r}^\top \right) dV, \tag{2.4}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix.

2.2 Perspective camera model

A perspective camera models the image formation process as a projection of three-dimensional world points onto a two-dimensional image plane. In this model, shown in Figure 2.1, each point in the 3D scene is projected onto the image plane through a single point, known as the camera center, according to the principles of projective geometry. In practice, pinhole camera models may show deviations from this ideal projection due to optical imperfections in the imaging system. The most common forms of such deviations are radial distortion and tangential distortion, shown in Figure 2.2, which result in nonlinear deformations of the projected image.

Radial distortion causes straight lines in the scene to appear curved in the image, with the effect becoming more pronounced toward the image boundaries. It is commonly modeled using the coefficients k_1, k_2, and k_3

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.5}$$

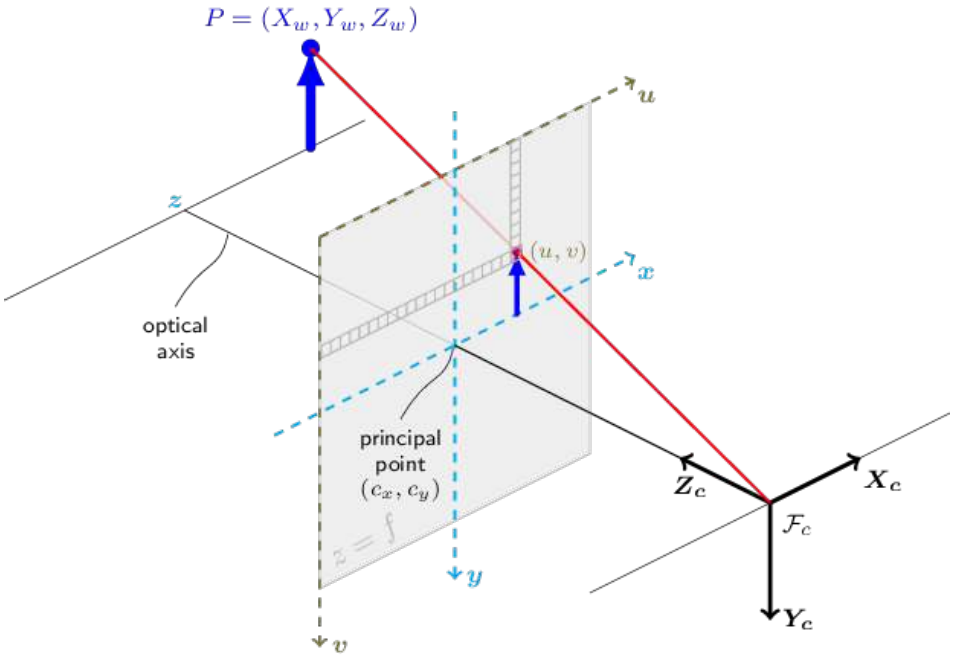

Figure 2.1: Figure that illustrates the pinhole camera model. Image from [12].

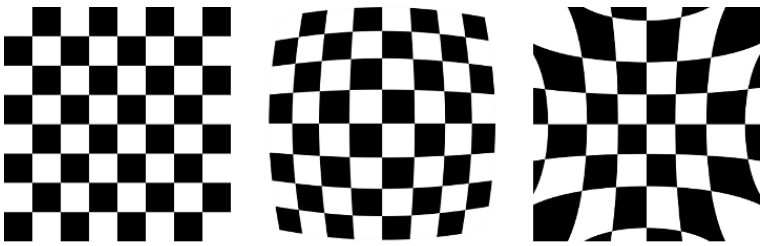

Figure 2.2: Two common types of radial distortion. No distortion (Left), negative radial distortion (Center), and positive radial distortion (Right). Image from [12].

$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.6}$$

where r is the Euclidean distance of a pixel from the distortion center, defined as $r^2 = x^2 + y^2$.
Tangential distortion, on the other hand, arises from misalignment between the lens elements and the image plane. It is typically modeled using parameters p_1 and p_2

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \tag{2.7}$$

$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \tag{2.8}$$

In our case, GoPro cameras are affected primarily by radial distortion. To compensate for this effect and obtain undistorted images, the distortion coefficients must be estimated together with the camera's intrinsic and extrinsic parameters.
The intrinsic parameters describe the camera's internal characteristics and define the mapping from three-dimensional points in the camera coordinate system to two-dimensional pixel coordinates on the image plane. These parameters are independent of the camera's position and orientation within the scene and are commonly represented by the following matrix:

$$\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f_x and f_y are the focal lengths expressed in pixel units along the horizontal and vertical axes, c_x and c_y denote the principal point, typically close to the image center, and s is the skew parameter.
The extrinsic parameters describe the position and orientation of the camera with respect to a reference world coordinate system. They define the rigid transformation that maps three-dimensional points from the world coordinate frame to the camera coordinate frame and consist of:

$$\begin{bmatrix} R & t \end{bmatrix}$$

with rotation matrix $R \in \mathbb{R}^{3\times3}$ and translation vector $t \in \mathbb{R}^3$.

Chapter 3

Related work

Most studies in the field of XR concerning the realism of virtual objects are still evolving, as this remains a relatively recent research topic.
The study [7] describes how virtual objects must provide sensory feedback that aligns with user expectations, combining sensory stimuli such as touch, sound, and vision to enhance immersion. To integrate these properties, a scene graph, a hierarchical data structure that represents objects, users, and their interactions, is used for robotic manipulation, as an example. Currently, vision-based tracking struggles with subtle hand and finger movements. Another limitation of many existing systems is that they primarily react to collisions, rather than adapting their responses according to the specific characteristics of different interactions. Furthermore, interactions should produce responses consistent with the physical and material properties of the objects involved. In particular, virtual objects should support articulated motion, interactions with the surrounding environment, and physically meaningful properties such as the CoM and the inertia tensor. Estimating these physical properties is essential for enhancing the realism of XR simulations. Particularly,

accurate knowledge of an object's mass and its distribution enables the simulation of weight perception and dynamic behavior, thereby improving physical interactions between virtual objects and their environment. In this context, the survey presented in [11] reviews the principal methods for estimating an object's inertial parameters, categorizing them into three main approaches, as shown in Figure 3.1. The first category discussed in the survey focuses on purely vision-based methods, which rely exclusively on visual information to infer inertial properties. While these approaches can be computationally efficient, they typically require simplifying assumptions, such as uniform density. Consequently, their accuracy may degrade when such assumptions do not hold, particularly in cases involving non-uniform or unknown mass distributions. The second type of method described in the survey is the exploratory method, which uses physical interaction with the object, such as pushing or tilting the object, to measure responses and estimate parameters using physical laws. These methods are more accurate, but are often limited to controlled environments.

The last type mentioned in the paper is the Fixed-Object method, which involves attaching or fixing the object to a robot to estimate parameters from joint motion and forces. These methods are highly accurate but limited to situations where the object can be grasped. In fact, for the latter, we have several studies on inferring these properties using different techniques such as finite element analysis (FEA), motion capture, and robotic manipulation.

As we have said previously, the proposed system relies on both visual sensing and physical manipulation to estimate an object's position in three-dimensional space.

To further enhance estimation robustness in visual perception, it is advantageous to exploit distinctive, easily detectable feature points on the object's surface. In this context, integrating cameras with fiducial markers provides a reliable solution. Fiducial marker-based approaches enable robust object detection and pose estimation by establishing consistent

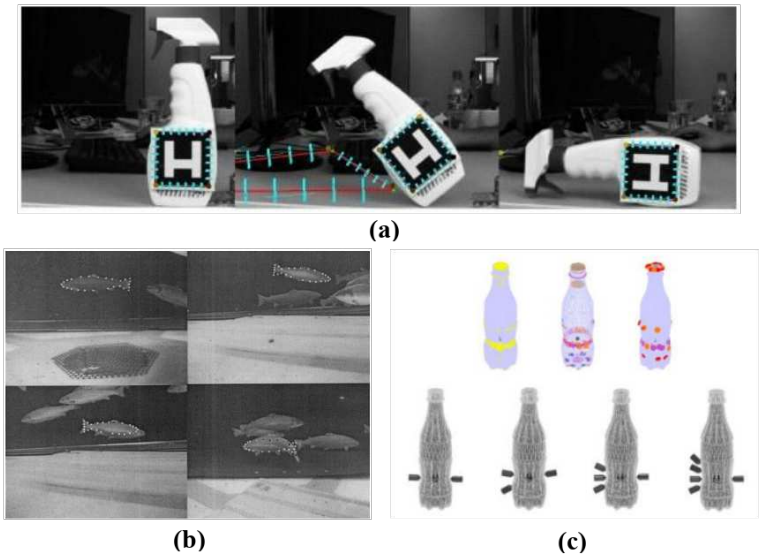

Figure 3.1: Examples of the main methods for estimating an object's inertial parameters. (a) Exploratory method: the mass and CoM are estimated by physically interacting with the object, for example, by striking it. (b) Purely visual method: inertial properties are inferred from visual features extracted from video sequences, such as a fish's observed shape and motion. (c) Fixed-object method: known inertial parameters are exploited to ensure stable grasping and manipulation. Images from [11].

correspondences between the real and virtual domains. These techniques are widely adopted to improve tracking accuracy and repeatability, particularly in scenarios where purely vision-based methods are prone to ambiguity or drift.
In particular, this work adopts ArUco markers, which provide an efficient and flexible framework for generating customizable fiducial markers tailored to the system requirements. Their automatic identification and robustness to noise make them especially suitable for accurate pose estimation in controlled experimental setups.

3.1 Inertial properties estimation using visual methods

As mentioned above, there are several primary methods for estimating inertial parameters using purely visual methods. Current developments in computational capabilities and image-processing methodologies have enabled the establishment of quantitative relationships between visual descriptors and inertial properties, based on empirical data and statistical modeling. The study [18] proposed a single-camera, multi-view image-processing approach for estimating both the volume and mass of axi-symmetric fruits by classifying fruit shapes into spherical, ellipsoidal, and paraboloidal categories and applying the corresponding analytical volume models. The method captures five views with a single camera, improving the accuracy while maintaining low hardware complexity.

The authors established strong correlations between estimated volume and actual mass using regression models. For CoM estimation, more recent approaches leverage machine learning and simulation-based techniques to infer it from visual and interactive data, significantly improving accuracy and robustness. For example, in [9], the authors proposed a deep learning–based pig mass estimation framework designed explicitly for unconstrained environments. The approach integrates instance seg-

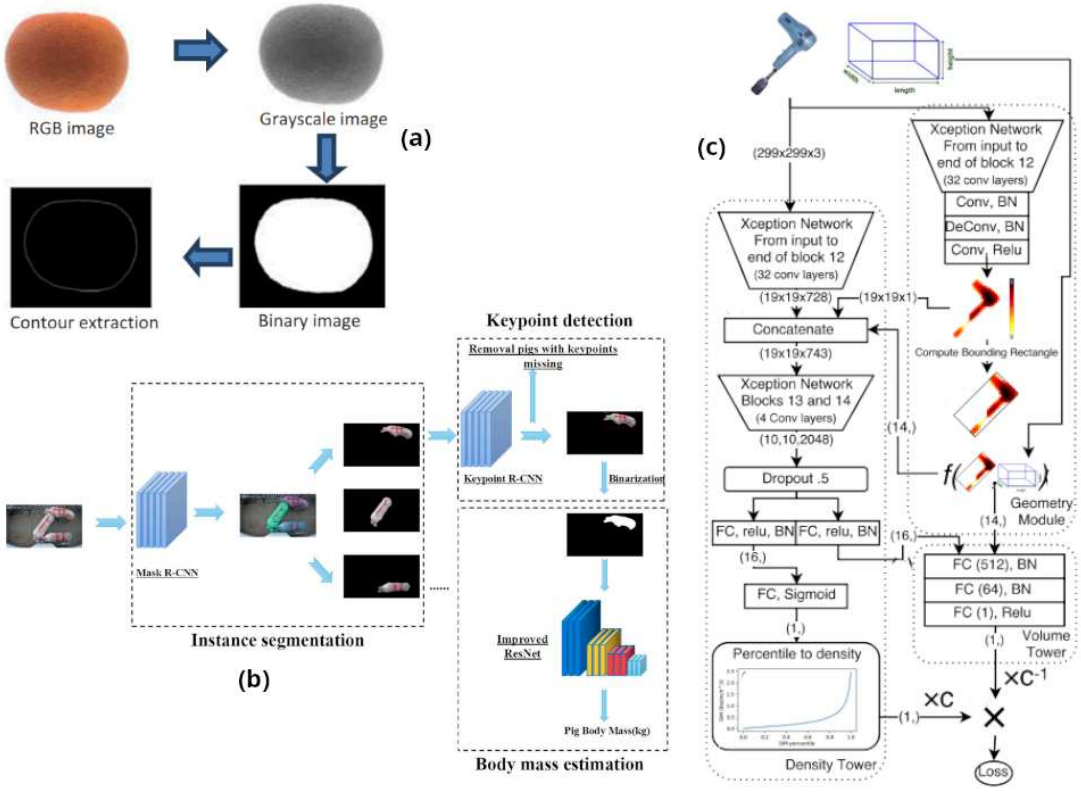

Figure 3.2: (a) The process, in steps, to extract the contour of the fruit from the RGB image. From the paper [18]. (b) In the structure of the Pig Mass Estimation Model based on a Deep Learning approach from the study [9], pigs are segmented from images. (c) Overview of the architecture for estimating the mass of an object from images described in the study [14].

mentation (Mask R-CNN), keypoint detection (Keypoint R-CNN), and a customized ResNet-based regression network to isolate individual pigs, filter out incomplete or occluded samples, and estimate body mass in real time.

In recent research on deep learning-based approaches, [14] introduced the image-to-mass problem, demonstrating that object mass can be learned from visual appearance and geometric hints when paired with large-scale ground-truth data. By collecting online product listings that include images, dimensions, and weights, the authors constructed a sizable dataset and demonstrated that combining learned density cues with geometry-aware volume estimation significantly improves mass prediction. The work illustrates both the feasibility and the challenges of learning from large-scale image data, particularly regarding data noise, labeling inaccuracies, and dataset bias inherent in web-sourced data. This framework is especially applicable to objects characterized by organic or complex geometries, such as biological entities and intricate industrial components. The Figure 3.2 provides an overview of the cited works discussed in this section.

3.2 Inertial properties estimation by manipulation

Early research on robotic perception of physical properties focused on direct probing and force-based interaction, in which robots estimate material characteristics such as mass, stiffness, or friction through contact sensing. The [8] study introduced a conceptual framework in which manipulation is used not for task execution per se but as a deliberate act of perception, demonstrating that material properties could be inferred through controlled robotic probing and laying the foundation for interaction-based perception. Through physical probing and by sensing the resulting forces, displacements, and acoustic responses, robots could

infer intrinsic material characteristics.

Consequently, accurate modeling of object mass distribution is crucial for robotic grasping and manipulation, especially when interacting with unknown or irregular objects. While traditional vision-based methods have focused on geometry or surface features, they often lack physical understanding, such as internal mass distribution, leading to unstable grasps or inefficient manipulation strategies. In response, recent research has increasingly turned to multi-modal sensing and data-driven approaches that incorporate tactile, force, and visual feedback to infer object dynamics and improve grasp planning.

In the study [3], the authors developed a system that integrates tactile and visual sensors with slip-detection mechanisms to identify unstable grasps and replan accordingly. Their key innovation lies in the introduction of a re-grasp planner that estimates the object's CoM based on tactile and torque data collected during the lifting phase, deploying both Support Vector Machines (SVMs) and Long Short-Term Memory (LSTM) networks for slip classification, and using an LSTM-based model for learning a re-grasp policy conditioned on multi-sensor input.

The concept of haptic vision, introduced by study [16], represents a hybrid paradigm that combines active vision with minimal, well-designed physical contact. Instead of relying on full grasping or dense tactile sensing, haptic vision uses vision to estimate object geometry and posture, then plans simple interactions, such as pushing, to extract informative object behavior. The approach exploits object symmetry and frictional contact to induce straight-line sliding motion, allowing mass to be estimated from measured contact forces under known friction coefficients, reducing manipulation complexity while maintaining reasonable accuracy.

Figure 3.3 illustrates the key points of the works discussed in this section.

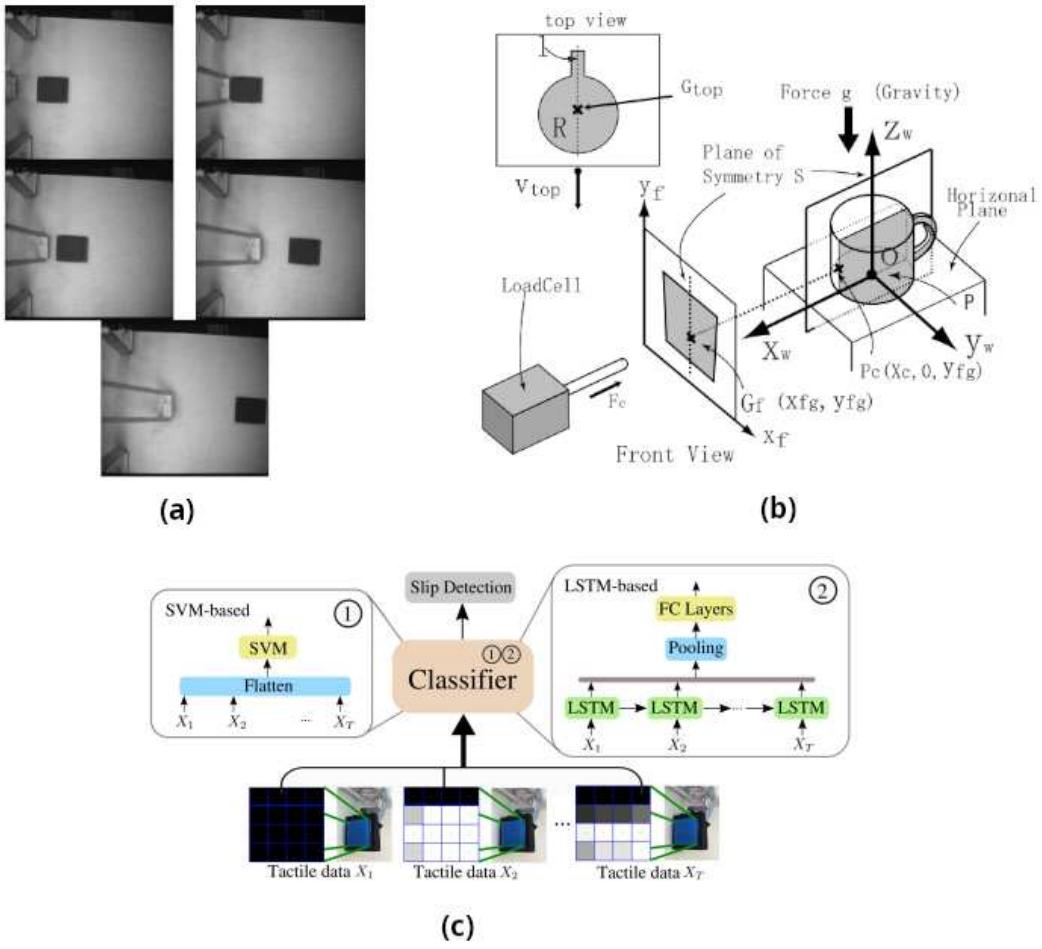

Figure 3.3: (a) A sequence of images of the pendulum making contact and of the object sliding, shown in the study [8]. (b) Contact point and contact force estimation of the haptic vision system is present in the paper [16]. (c) The slip detection model system introduced in the study [3]. It is processed with two classifiers: an SVM and an LSTM model to predict the slip.

3.3 Ground-truth physical properties

As we have seen, more recent approaches taking into account machine learning and simulation-based strategies for CoM estimation rely on the availability of high-quality ground-truth data. Several benchmark datasets and systems provide important insights into how reliable ground-truth can be acquired in complex interaction scenarios, integrating synthetic and real data sets.

In terms of ground-truth datasets, the study [5] introduced one of the first large-scale, real-data benchmarks that capture hand–object interactions with precise 3D hand-pose annotations. The data collection system combines an RGB-D camera with a magnetic motion capture setup, in which six 6-DoF magnetic sensors attached to the fingertips and wrist are used with inverse kinematics to infer full 21-joint hand poses. This hybrid sensing approach addresses a key limitation of vision-only systems, severe self-occlusions caused by object manipulation, which significantly degrade annotation quality when relying solely on RGB or depth data. Beyond hand pose, the dataset also includes 6D object pose annotations and 3D object meshes for a subset of objects.

The paper [2] introduces a large-scale data collection and annotation framework for robotic grasping in cluttered scenes. Although originally designed for grasp pose learning, GraspNet is highly relevant to mass distribution learning due to its hybrid real–analytic data generation pipeline, physics-based ground-truth labeling, and scalable scene-level annotation strategy.

The authors propose a two-stage annotation pipeline that combines real RGB-D data acquisition with analytic computation of physical feasibility. Real scenes are captured using calibrated RGB-D sensors mounted on a robotic arm, ensuring visual realism and accurate geometry. The study [10] introduced HandData, a large-scale, first-person dataset capturing human reach-to-grasp interactions with real objects using non-visual proximity sensing and kinematic measurements, shown in Figure

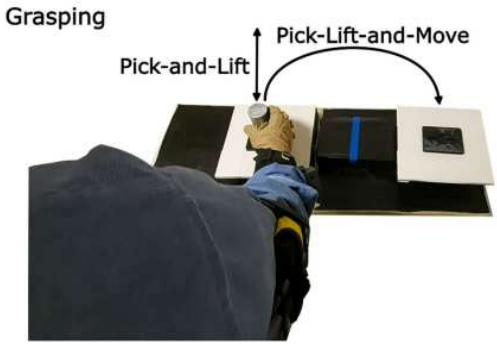

Figure 3.4: Image from the study [10] that shows the data acquisition protocol. In the grasping scenarios, the user interacts with the object.

3.4.

This study describes a different approach, unlike traditional computer vision techniques. The following method avoids RGB and RGB-D cameras and instead tracks finger joint angles, wrist, and forearm kinematics using radar and time-of-flight (ToF) proximity sensors mounted on a custom glove to record the object's proximity before grasping. The authors have created a dataset that considers different scenarios, such as picking and lifting objects or moving them between two platforms, and records grasping motion.

In this work, we will consider a computer vision-based approach that provides a full 3D hand-pose structure during object grasping using RGB cameras, which helps a system estimate physical properties such as the CoM and inertial tensor.

3.4 Fiducial markers

Fiducial markers are artificially designed visual patterns introduced into a scene to provide reliable reference points for computer vision systems. Their primary function is to provide robust detection, identification, localization, and pose estimation by exploiting prior knowledge of the marker's appearance and geometry. The use of fiducial markers guarantees easy detection and unique identification, rather than relying on the extraction of natural features from arbitrary scenes.
Markers establish reliable correspondences between 2D image points and known 3D locations, enabling accurate estimation of camera pose via geometric transformations.
The study [4] compares two fiducial marker systems, ARTag and ARToolkit. Both of them employ planar square markers with a distinct black border and an encoded interior pattern. The pose estimation is achieved by detecting the four marker corners in a frame, which define a homography between the marker and the image plane.
Previous systems relied on predefined marker dictionaries, which either restricted the number of usable markers or resulted in small inter-marker distances, increasing confusion in noisy settings. Indeed, one main drawback is the limited error correction, making them sensitive to blur, lightning changes, and partial occlusion.
The [6] paper introduces a fiducial marker system that works to improve the limitations we discussed above. The ArUco system employs an automatic method for generating configurable marker dictionaries, formulated as an optimization problem that maximizes the inter-marker Hamming distance to reduce confusion between markers and the number of bit transitions, improving detectability and robustness to environmental conditions.

ArUco uses a robust marker detection and identification pipeline, as illustrated in Figure 3.5. The process includes adaptive thresholding, contour extraction, polygonal approximation, and binary code extraction.

Figure 3.5: ArUco, the fiducial markers system to provide robust detection, identification, and pose estimation. Each marker has a pattern; for example, the markers with IDs 200 and 201 are represented by the patterns shown in the figure, respectively.

Marker identification is performed by comparing detected codes against the generated dictionary, accounting for all four possible rotations and avoiding redundant checksum bits, since it uses dictionary-based distance analysis for error detection and correction.
Another study [13] analyzes the accuracy of 3D pose estimation with ArUco markers using intrinsic and extrinsic camera parameters, rather than relying solely on physical experiments. The results show that pose accuracy degrades significantly as the marker's projected area in the image decreases, and that large tilt angles reduce both detection probability and estimation robustness. So, the study demonstrates that ArUco-based camera pose estimation accuracy strongly depends on imaging geometry, particularly marker size in pixels and orientation relative to the camera.
The camera pose relative to the marker is a 3D transformation from the marker coordinate system to the camera coordinate system, where the Perspective-n-Point (PnP) function estimates the object pose given a set of object points, taking into account the camera intrinsic matrix and the previously calculated distortion coefficients.
The points X_w expressed in the world frame are projected into the image plane $[u, v]$ using the perspective projection model Π and the camera

intrinsic parameters matrix A

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A\Pi^{c}T_{w} \begin{bmatrix} X_{w} \\ Y_{w} \\ Z_{w} \\ 1 \end{bmatrix} \tag{3.1}$$

The estimated pose is the rotation and the translation vectors that allow transforming a 3D point expressed in the world frame into the camera frame:

$$\begin{bmatrix} X_{c} \\ Y_{c} \\ Z_{c} \\ 1 \end{bmatrix} = {}^{c}T_{w} \begin{bmatrix} X_{w} \\ Y_{w} \\ Z_{w} \\ 1 \end{bmatrix} \tag{3.2}$$

Chapter 4

Hardware Architecture of the System

Learning the ground-truth mass distribution requires accurate, repeatable data acquired under controlled conditions. In this context, we consider a hybrid approach from the literature that we have analyzed above to provide a reliable system for collecting data under advantageous conditions.
The hybrid approach consists of the visual and fixed-object methods, combining the benefits of both strategies. Indeed, the purely visual approach enables pose estimation from multi-view observations, while the fixed-object approach yields stable, repeatable measurements under controlled conditions. So, the purpose of the system is to enable the systematic collection of visual and pose data during object manipulation experiments, which are used in the subsequent learning process.

The data collection system developed for learning ground-truth mass distributions is designed to provide controlled, repeatable, and well-labeled data. It includes three main hardware components: (i) a multi-

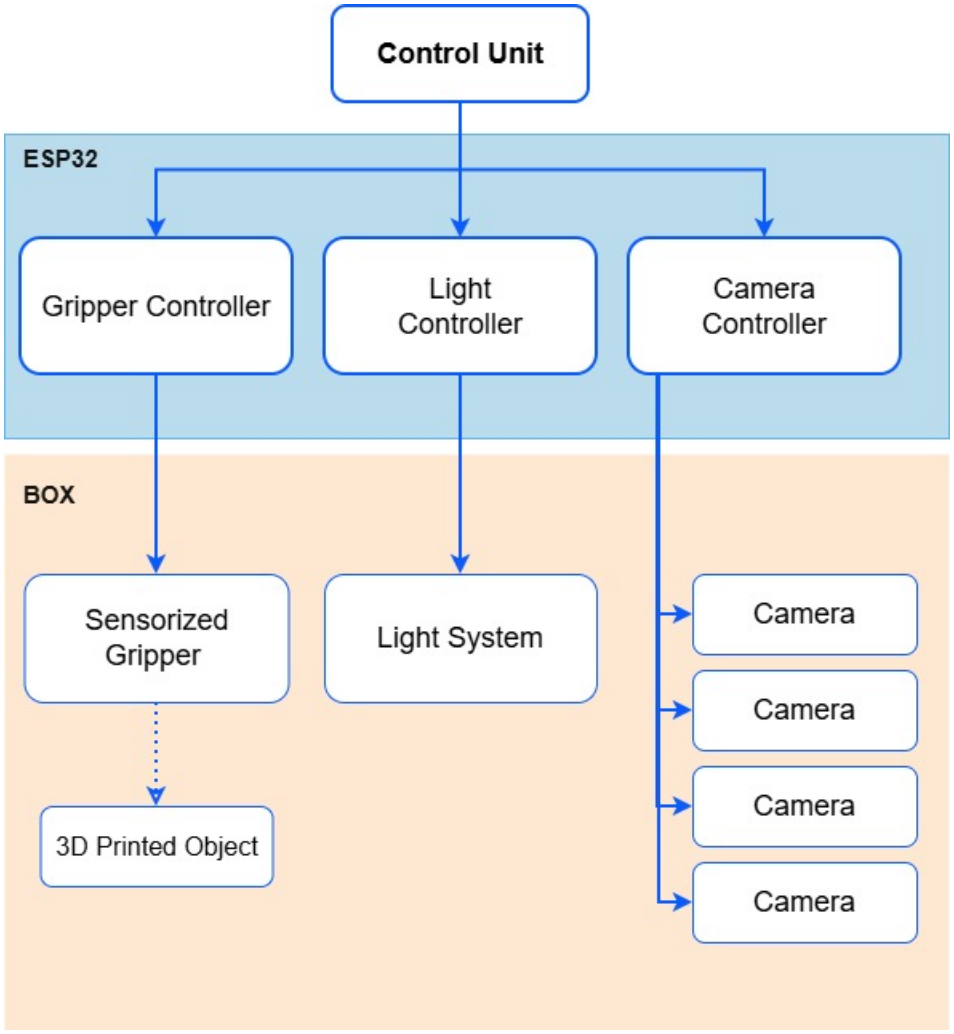

Figure 4.1: Overview of the system architecture, highlighting the hardware components and their interconnections.

camera image-based acquisition box, (ii) a passive hand-shaped gripper, and (iii) a 3D printed modular object.
Figure 4.1 illustrates the system architecture. The acquisition box is equipped with four cameras and a dimmable lighting system. The gripper is used for controlled object manipulation and interacts with the 3D printed object, whose internal mass distribution can be changed.
The cameras, LED lights, and the sensorized hand-shaped gripper are connected to an ESP32 microcontroller. The CU communicates with the microcontroller via the serial port, sending commands to control the overall system.
In Section 4.1, the multi-camera image-based acquisition box and its lighting system are presented. Section 4.2 introduces the 3D-printed modular object, while Section 4.3 describes the hand-shaped gripper.

4.1 Image-based acquisition box

The image-based acquisition device consists of a rigid rectangular plywood box that defines a controlled workspace for data collection. The use of this enclosure serves two primary purposes. First, it reduces the influence of external factors, such as background clutter, which could otherwise introduce unwanted variability into the visual data. Second, it provides a fixed geometric reference that facilitates camera calibration and synchronization.

Each panel of the acquisition box is covered with a pattern of 16 ArUco markers, each assigned a unique identifier, as illustrated in Figure 4.2. This configuration facilitates the estimation of both intrinsic and extrinsic camera parameters during calibration. The use of distinct marker IDs ensures unambiguous identification of each panel; for example, marker IDs 64–79 are assigned to the bottom panel.
For each marker, the corresponding three-dimensional position is defined in the box's reference frame and annotated in a CSV file. These positions are determined during the design phase and physically enforced during

Figure 4.2: ArUco configuration applied for each box side: in the top picture, it is the concept of the configuration, using that order of ID ArUco in the panel. The picture below shows the box with the markers affixed in the little groove corresponding to its 3D position.

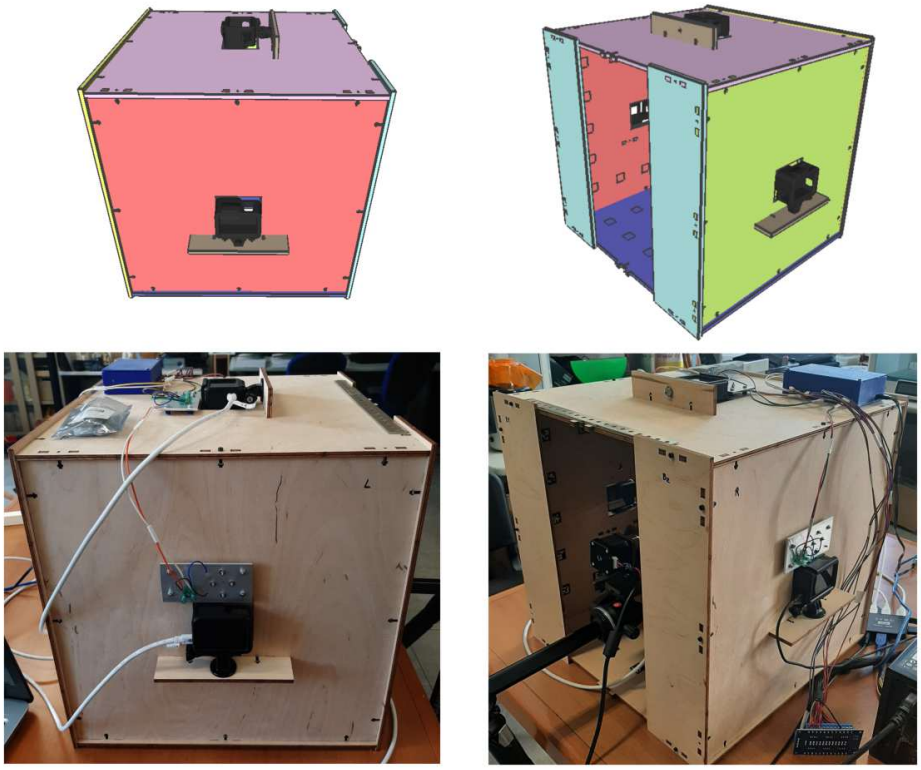

Figure 4.3: The image-based acquisition box, used for a controlled environment, has four RGB cameras mounted on the lateral faces to take photos on each side.

assembly. In particular, shallow grooves are engraved on each panel to ensure precise and consistent placement of the markers within the box. Finally, four RGB cameras are mounted on one side of the box – namely, the left, right, top, and front positions – as illustrated in Figure 4.3. Each camera is positioned at the same height and oriented toward the center of the enclosure. This configuration enables the simultaneous acquisition of four views from different viewpoints, providing comprehensive visual coverage of the workspace. Such coverage is particularly important when the 3D object undergoes rotations or complex motions, as it significantly reduces the likelihood of occlusions.

All cameras are synchronized and calibrated with respect to a common reference frame defined by the enclosure. This calibration ensures consistent multi-view observations and supports tasks such as motion tracking, pose estimation, and multi-view analysis.

The rear panel of the box includes an opening through which the gripper holding the object is introduced into the workspace. To ensure consistent illumination during data acquisition, each panel is equipped with high-power LED lights, providing uniform, stable lighting throughout the enclosure.

4.2 Modular 3D printed object

The test object used in this work is a small rigid parallelepiped manufactured by 3D printing, with multiple internal cavities and fixed external dimensions. The external geometry of the object remains unchanged across all experiments, a key design requirement to isolate the effects of internal mass distribution on the object's motion. The object is completed with a cover, resulting in an enclosed structure. Its design, shown in Figure 4.4, is optimized for straightforward fabrication using a Fused Deposition Modeling (FDM) 3D printer. Internally, the object is subdivided into 36 small cavities, each capable of accommodating a removable bar made of different materials, such as iron. The cavities are arranged

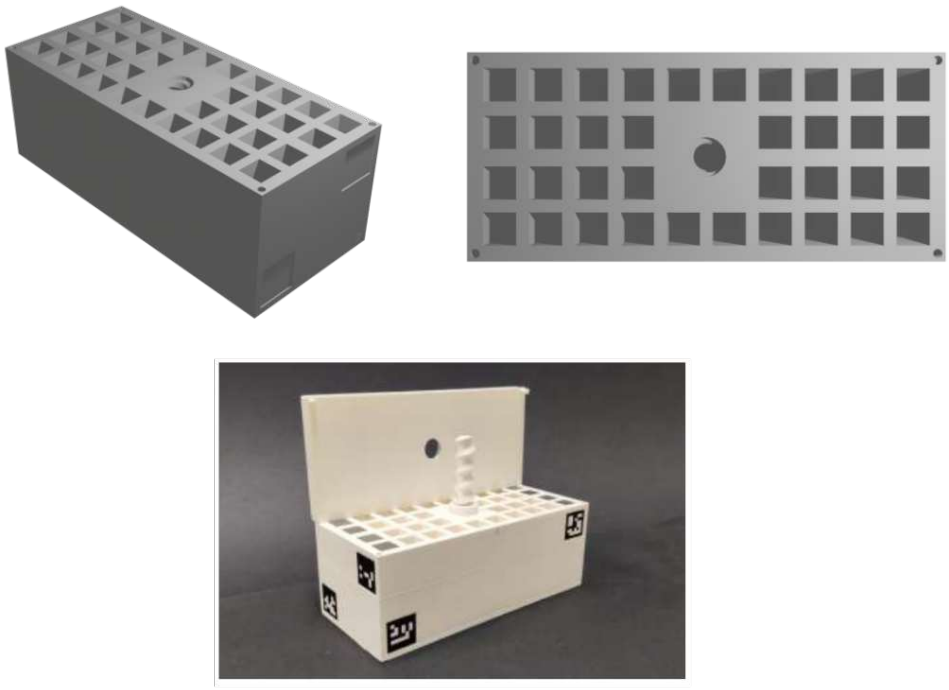

Figure 4.4: The 3D model of the modular test object. As shown, the box has 36 small cavities and two fiducial markers on each lateral side.

in a regular pattern within the object, and their positions are known with respect to the object's reference frame. By inserting bars into different cavities, a wide range of internal mass configurations can be achieved with the same physical object.
This internal modularity allows the object's CoM to be shifted in a controlled, repeatable manner. Importantly, since the external shape and visual appearance of the object remain unchanged, any variations observed in the object's motion can be directly attributed to changes in mass distribution rather than to geometric or visual cues.
Additionally, two ArUco markers are placed on each face of the object to increase the likelihood that a sufficient subset of markers remains visible at all times, even during rotations or partial occlusions caused by the gripper's fingers. The markers on the object surface enable robust estimation of the object's pose within the reference frame of the acquisition box.
To achieve this, each marker is associated with a predefined three-dimensional position on the object surface, as described in Section 4.1. This correspondence between 2D image observations and known 3D marker locations enables accurate pose estimation via standard PnP formulations, supporting reliable data acquisition and robust motion analysis.

4.3 Passive hand-shaped gripper

To manipulate and hold the test object consistently and repeatably, a passive gripper – designed as a low-cost approximation of a human hand – is used throughout the data collection process. The hand-shaped gripper was developed in collaboration with Sant'Anna School of Advanced Studies and enables precise control of the object's pose and motion while minimizing variability introduced by direct human interaction. An overview of the gripper holding the object within the experimental environment is shown in Figure 4.5. The gripper consists of a hybrid structure combining laser-cut components and 3D-printed parts. It features a spring-loaded

Figure 4.5: The hand-shaped gripper used for the experiments holds the test object. Each cube-like shape fingertips has different fiducial markers on each side.

thumb opposing three fixed fingers.
Each finger is equipped with a dedicated force-sensitive resistor (FSR) sensor (Alpha MF01A-N-221-A01, force range up to 10 N) to measure the contact forces generated by the object's mass distribution. The sensors are covered with a 2-mm-thick silicone layer, which improves force distribution and increases static friction at the contact interface. Each fingertip is designed with a cube-like geometry to accommodate a set of ArUco markers, enabling accurate tracking of finger motion and contact dynamics. As with the acquisition box and the object, each marker is associated with a unique three-dimensional position defined during the fingertip design phase. These correspondences between known 3D marker locations and their 2D projections in the image are used to estimate the pose of each fingertip with respect to both the box and object reference frames.
This formulation allows for precise estimation of the contact point between the fingertip sensor and the object surface. This design choice is particularly relevant given the four-camera setup described in the previous subsection. As discussed in Section 4.2, placing markers on all sides of the fingertip ensures that a subset of markers remains visible at all times, thereby reducing the impact of partial occlusions caused by the object or by other fingers.
The gripper has a single actuated degree of freedom and is driven by a stepper motor, allowing rotation about the pronation-supination axis. Embedded electronics, including an ESP32 carrier board and a DRV4988 stepper driver, are integrated to control the actuation and to acquire measurements from the fingertip sensors.

Chapter 5

Software Architecture of the System

This chapter describes the software toolkit developed for the data collection system. The toolkit is implemented in C++ and uses external libraries, including OpenCV.

As illustrated in Figure 5.1, the software is composed of two main modules. The first module is responsible for data acquisition within the image-based acquisition box. In particular, it controls the internal lighting system and manages camera image capture in two operating modes. In the first mode, images are acquired for camera calibration and pose estimation; in this case, images are captured simultaneously from the four cameras. The second mode is designed for dynamic data acquisition, in which image capture is synchronized with the gripper's rotation. During this process, the software drives the gripper with controlled acceleration and acquires images at each rotation step.

The second module handles the computer vision tasks. These include estimating the cameras' intrinsic and extrinsic parameters using ArUco markers placed inside the enclosure, estimating the object pose using markers on its surface, and estimating the fingertip poses using ArUco

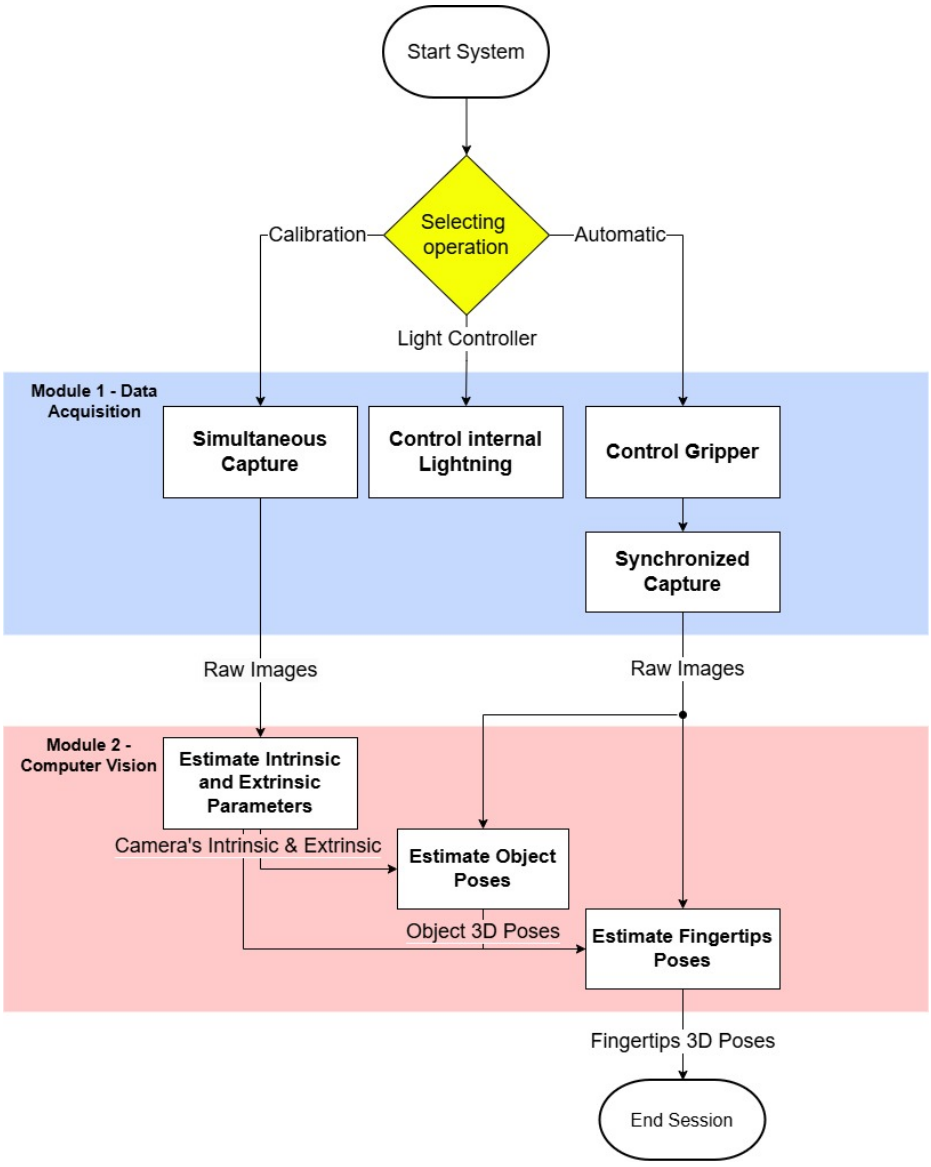

Figure 5.1: Diagram of the software toolkit architecture, showing the logical workflow and the interaction between the Data Acquisition and Computer Vision modules.

markers mounted on their cube-like structures.
The two modules operate in a coordinated manner to ensure consistent association between visual data and manipulation actions. Together, they form a unified software tool that is operated through a command-line interface (CLI).

5.1 Data acquisition module

In the CLI, the data acquisition module is run using the acquire command, which opens an interactive menu for data capture. Upon execution, the software attempts to connect to the microcontrollers that manage the acquisition box. If the connection is successful, the user can select among the following operations:

- Capturing a single image or starting/stopping video recording;
- Adjust the light intensity;
- Manually rotate the gripper; and
- Perform automatic acquisition during gripper rotation.

The automatic acquisition mode employs a dedicated routine to capture images of the object in multiple poses. In this mode, the gripper is rotated incrementally, and an image is acquired at each rotation step. The rotation is controlled by the micro-gripper firmware, which manages both the gripper actuation and the acquisition of fingertip force measurements. This approach emulates video frame acquisition while providing greater control over the angular resolution, as images are captured at predefined rotational positions.
During each iteration of the acquisition loop, images are captured simultaneously from all four GoPro cameras using the micro-Gopro firmware, which handles camera synchronization.

The routine requires two input parameters: the number of rotation steps and the maximum angular velocity, specified in the range [0,500] steps per second. For each step, the software sends a command to the ESP32 microcontroller via serial communication to rotate the gripper. Once the target angle is reached, a trigger command is executed to simultaneously acquire images from all cameras. After completing all rotation steps – 20 different poses in the experiments presented in this work – the gripper receives the command to return to its initial position.

5.2 Computer vision module

After the images are acquired by the data acquisition module, the software toolkit processes them as input to perform the CV tasks. The CV module is responsible for computing the cameras' intrinsic and extrinsic parameters, which are required for camera calibration and, consequently, for estimating each camera's position.
Once the camera poses are determined, the toolkit can estimate both the object poses and the poses of all the fingertips. This enables the construction of a digital twin of the test object and the hand-shaped gripper within the experimental environment.
The CV tasks are described in detail in the following sections. Specifically, Section 5.2.1 addresses camera calibration and camera pose estimation; Section 5.2.2 describes the algorithm used to estimate the test object's pose; and Section 5.2.3 presents the algorithm for estimating the fingertips' pose.

5.2.1 Camera calibration and camera pose

The first task performed by the CV module is camera calibration, as its results are fundamental for all subsequent vision-based operations. The calibration process takes as input a set of multi-view images of a

calibration pattern, in this case a chessboard, from which known correspondences between three-dimensional object points and their two-dimensional image projections are established.
Camera calibration is performed using the cv::calibrateCamera() function provided by OpenCV. This procedure estimates the camera intrinsic parameters—such as focal length, principal point, and skew—as well as the lens distortion coefficients. These parameters are essential for accurately modeling the imaging process and compensating for geometric distortions, thereby enabling reliable metric measurements and pose estimation in subsequent stages.

Object points correspond to the three-dimensional coordinates of real-world reference features, expressed in the calibration pattern's coordinate system. These points are defined a priori based on the known geometry of the pattern. In this work, they are obtained by computing the coordinates of the chessboard corners, given the board dimensions and the square size.
Image points, in contrast, represent the two-dimensional pixel coordinates of the same physical features as observed in the captured images. These points are automatically extracted from the acquired data using feature detection algorithms. By providing object and image points computed to cv::calibrateCamera() as parameters, the algorithm solves a nonlinear parameter estimation problem, minimizing the reprojection error defined as

$$\sum_i \|\mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i}\|^2 \tag{5.1}$$

where $\mathbf{x_i}$ is the detected image point and $\hat{\mathbf{x}}_\mathbf{i}$ is the projected object point. The calibration function estimates the camera matrix, distortion coefficients, and a set of rotation and translation vectors, which are subsequently stored in an XML file for later use. The camera matrix is a 3×3 matrix that encodes the intrinsic parameters, including the focal

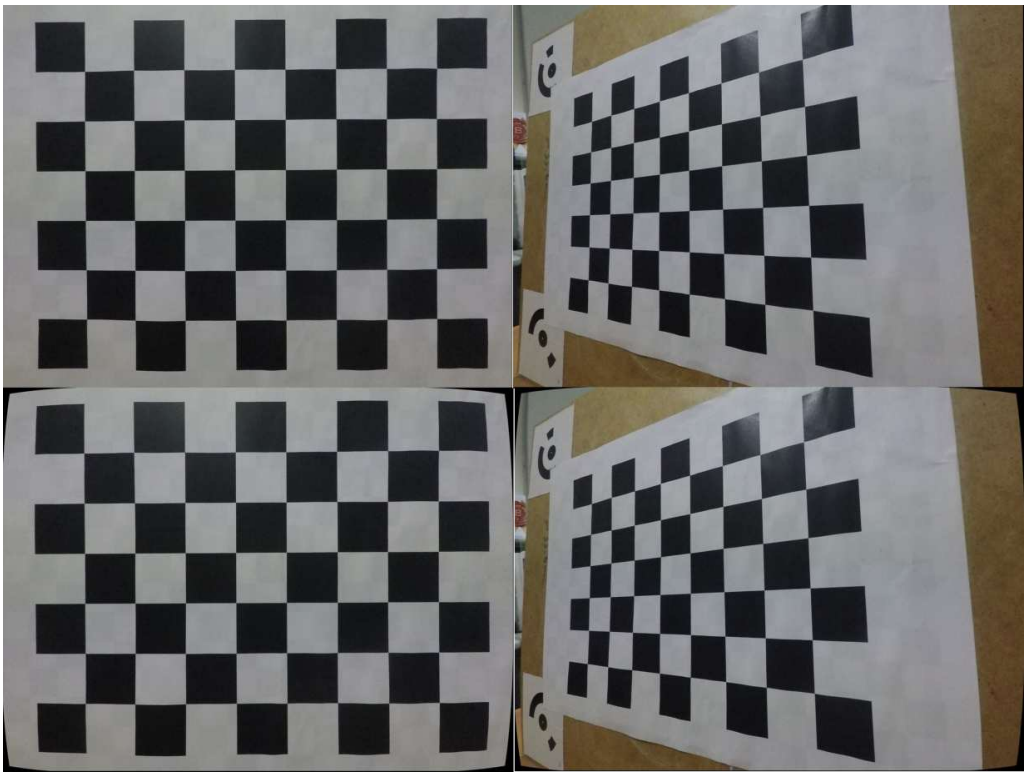

Figure 5.2: The two images in the top row depict the chessboard patterns used as input to the calibration process within the toolkit. The images in the bottom row illustrate the corresponding undistorted results obtained after the calibration phase, highlighting the correction of lens-induced distortions.

lengths and the principal point coordinates. The distortion coefficients, typically represented as a vector with five or more elements, model the optical distortions introduced by the camera lens. An example of undistorted images, after the calibration phase, is shown in Figure 5.2.
Once the intrinsic parameters have been estimated, the camera pose can be determined. To this end, ArUco markers, introduced in Section 3.4, are employed. The toolkit leverages the predefined marker pattern of the image acquisition box, shown in Figure 5.3, to estimate the pose of each of the four cameras with respect to the box reference frame. Figure 5.4 shows the detected marker, including its identifier. The pose of the camera relative to a marker is represented as a three-dimensional rigid transformation from the marker coordinate system to the camera coordinate system, defined by a rotation and a translation. The OpenCV function cv::solvePnP() estimates this transformation using all detected markers in the image. This is achieved by solving a PnP problem, defined as:

$$x = K \left[R|t\right] X \tag{5.2}$$

where K is the intrinsic camera matrix, $[R|t]$ is the extrinsic parameters with the rotation R and the translation t, X denotes the three-dimensional coordinates of the marker corners, and x represents their corresponding two-dimensional image projections. Given these known quantities, the algorithm computes the rotation vector (*rvec*) and translation vector (*tvec*), which together define the camera's extrinsic parameters relative to the marker.
The three-dimensional coordinates of the marker corners are measured in advance and stored in a CSV file, with each marker ID associated with its corresponding corner positions expressed in the acquisition box's reference frame.

Figure 5.3: The four images used as input for camera pose estimation are shown. In clockwise order, they correspond to the views captured by the left, right, top, and front cameras, respectively.

Figure 5.4: The four images shown are the outputs of the camera pose estimation process based on detected ArUco markers. In clockwise order, they correspond to the views captured by the top camera and its close-up, right camera and its close-up, respectively.

5.2.2 Object's pose estimation

After estimating the intrinsic and extrinsic camera parameters, the pose of the tester object can be computed for each acquisition step.
As described in Section 5.1, the software toolkit operates in an automatic mode, capturing multiple images at each rotation step from all camera viewpoints. Using these synchronized observations, the object pose is estimated in the reference frame of the image-based acquisition box for each of the 20 rotation steps, as shown in Figure 5.5.
The object pose is obtained by solving the following optimization problem:

$$\min_{t_x t_y t_z \alpha \beta \gamma} \sum \|p - KCO(t_x, t_y, t_z, \alpha, \beta, \gamma)P\|^2 \tag{5.3}$$

where:

- p is the 2D image coordinate of a detected marker,
- P is the corresponding 3D marker coordinate in the object reference frame,
- K is the camera intrisic matrix,
- C is the camera extrinsic matrix,
- t_x, t_y, t_z define the translation vector of the object,
- α, β, γ are the Euler angles representing object rotation,
- $O(t_x, t_y, t_z, \alpha, \beta, \gamma)$ is the homogeneous transformation matrix describing the object pose in the acquisition box reference system. This transformation represents the unknown variable to be estimated.

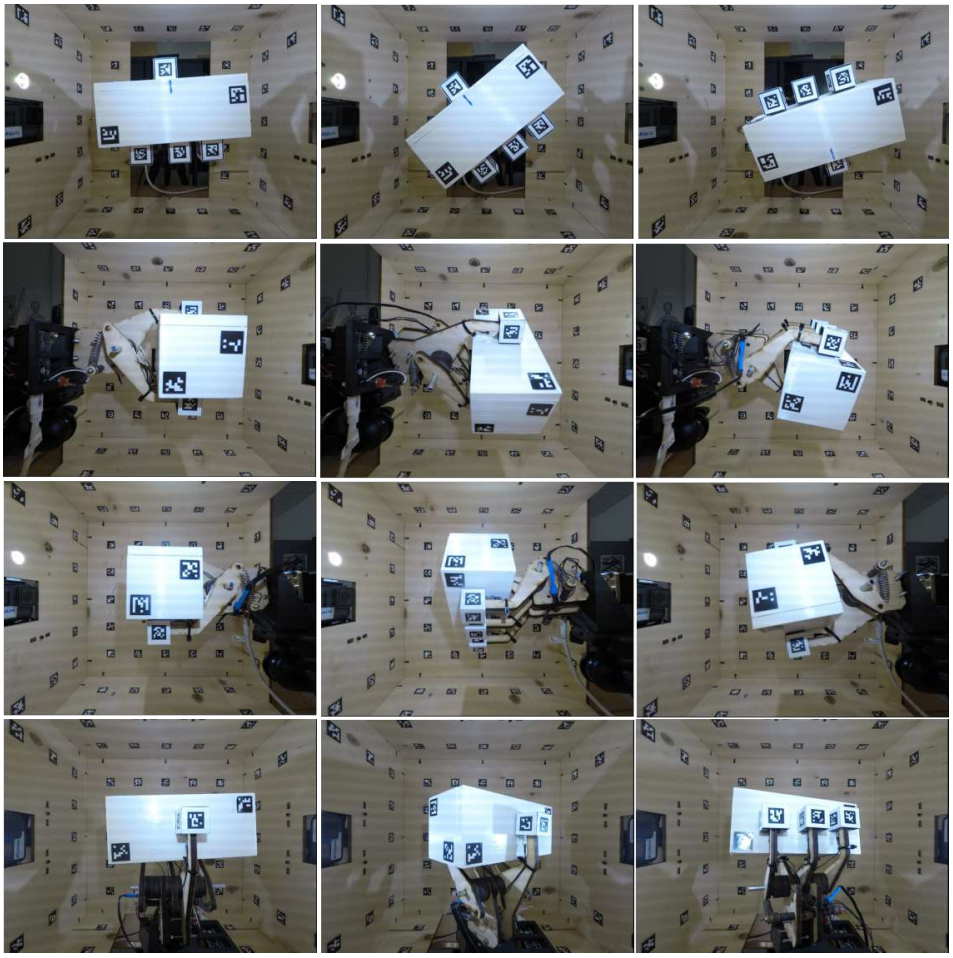

Figure 5.5: Some images shown in this Figure represent the input used for object and fingertips pose estimation. The object is rotated by the gripper.

This formulation corresponds to a nonlinear least-squares problem of the general form

$$\min_{\mathbf{x}} \sum_{i} (r_i(\mathbf{x}))^2 \tag{5.4}$$

where $r_i(\mathbf{x})$ are residual functions defined by the reprojection errors.
The optimization problem is solved using OpenCV's Levenberg–Marquardt (LevMar) nonlinear solver, cv::LMSolver. The solver relies on a user-defined callback class derived from cv::LMSolver::Callback, which computes residual errors for a given parameter vector and optionally provides the Jacobian matrix to improve convergence. The Jacobian matrix contains the first-order partial derivatives of the residual functions with respect to the model parameters and represents the local linearization of the nonlinear system. Then it is used to compute parameter updates by approximating the residual function via a first-order Taylor expansion.
In cv::LMSolver, the Jacobian can either be approximated numerically using finite differences, where each column of the Jacobian is estimated by perturbing one parameter at a time and observing the resulting change in residuals, in mathematical terms:

$$\frac{\partial f}{\partial p} \approx \frac{f(p+\epsilon) - f(p-\epsilon)}{2\epsilon} \tag{5.5}$$

where ϵ is the pertubation scalar.
As in the camera pose estimation procedure, the 3D marker coordinates are measured in advance in the test object reference frame and stored in a CSV file. During processing, these data are loaded into an ObjectData structure, which stores, for each marker:

- its 3D coordinates in object reference frame,

- its detected 2D image coordinates,
- the intrinsic and extrinsic parameters of the corresponding camera.

This structure is provided as input to the SolveObjectPose class, which derives from cv::LMSolver::Callback. Within its compute() method, the optimization problem defined in Eq.5.3 is evaluated. The residual error is computed using marker observations from all four camera views, enabling multi-view pose estimation and thereby improving robustness and accuracy.
After convergence, the estimated object pose for each rotation step is represented as a homogeneous affine transformation matrix. These matrices are stored in a YAML (YML) file and describe the object pose in the acquisition box reference frame at each rotation step.

5.2.3 Gripper's fingertips pose estimation

After obtaining the object poses, we can use the file created from the algorithm, described in the Section 5.2.2, to compute the fingertips poses following the formula:

$$\min_{t_x t_y t_z \alpha \beta \gamma} \sum \|p - KCOF(t_x, t_y, t_z, \alpha, \beta, \gamma)P\|^2 \tag{5.6}$$

where:

- p denotes the 2D image coordinates of a detected marker,
- P represents the corresponding 3D marker coordinates expressed in the fingertip reference frame,
- K is the camera intrisic matrix,
- C is the camera extrinsic matrix,

- O is the previously computed object registration matrix,
- t_x, t_y, t_z define the translation components of the fingertip pose,
- α, β, γ are the Euler angles describing the fingertip's orientation,
- $F(t_x, t_y, t_z, \alpha, \beta, \gamma)$ denotes the homogeneous registration matrix of the fingertip expressed in the object reference frame. This transformation represents the unknown variable to be estimated.

As discussed in Section 5.2.2, pose estimation is performed using a LevMar nonlinear least-squares solver. The same optimization framework is adopted to solve the problem defined in Eq. 5.6.

The 3D coordinates of the fingertip markers, pre-measured in the fingertip reference frame, are stored in a CSV file. During processing, these data are loaded into a FingerData structure, which contains, for each marker of the hand-shaped gripper:

- its 3D coordinates in the fingertip reference frame,
- its detected 2D image coordinates,
- the affine transformation matrix of the object,
- the intrinsic and extrinsic parameters of the corresponding camera view.

The EstimateHandPose() function iterates over all four fingertips. For each fingertip, the EstimateFingerPose() method processes the raw images together with the camera and object affine transformation matrices. Based on the detected marker IDs, the method identifies the corresponding finger and populates the FingerData structure with the required information.

Finally, the ComputeFingerPose() class, which derives from LMSolver::Callback, performs the nonlinear optimization using the LevMar solver to estimate

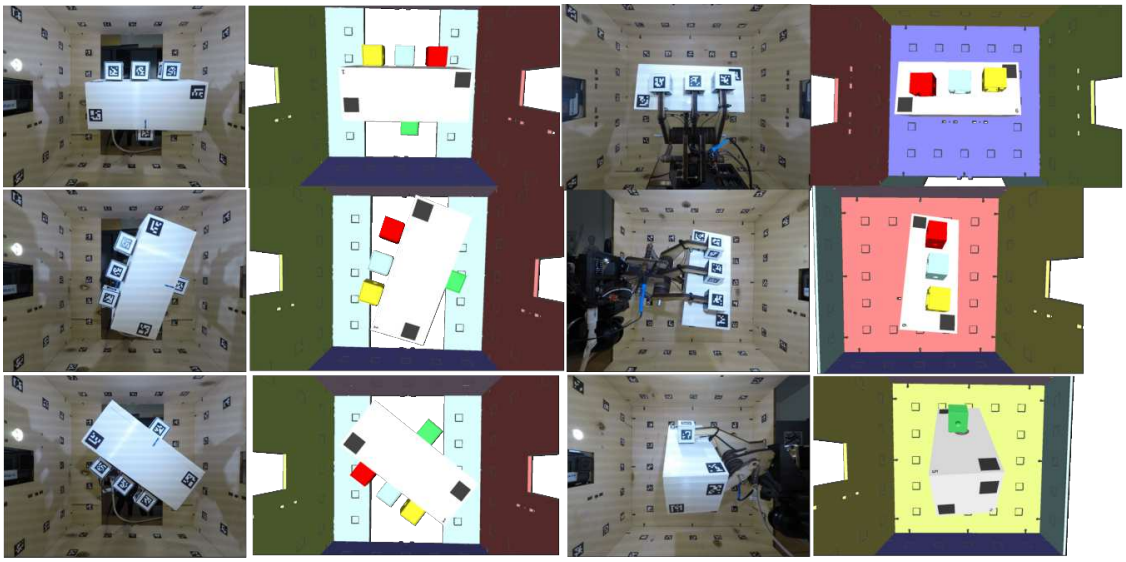

Figure 5.6: Comparison between the real image position and the 3D model with the estimated pose, previously computed.

the fingertip pose. As we have seen in the Section 5.2.2, the Jacobian matrix is computed using finite differences.
After convergence, the estimated fingertip poses for all captured frames are stored in a YML file and used to generate an MLP file, a MeshLab project. MeshLab [1] is an open source system for processing and editing 3D triangular meshes, which was key for our work. Figure 5.6 shows the 3D models positioned according to the coordinates estimated by the algorithm.

Chapter 6

Results

This chapter presents experimental results obtained with the proposed multi-camera data acquisition system for learning ground-truth mass distributions.
The objective of these experiments is to evaluate the geometric reliability of the system in terms of camera calibration accuracy and pose estimation consistency for the gripper fingertips, to minimize errors in the resulting transformation matrices.
The performance of the system was evaluated using intrinsic camera calibration, camera pose estimation relative to a common reference frame, and marker-based pose estimation of the object and the gripper fingertips during controlled rotational motion. The estimated poses were represented using rotation and translation vectors and subsequently converted into homogeneous transformation matrices for further analysis.
In the absence of an external ground-truth tracking system, the quality of the estimated poses was evaluated by analyzing the reprojection error. This metric indicates the geometric consistency of the calibration and pose estimation processes and enables assessment of the acquisition

system's stability and suitability for generating reliable pose data for learning tasks.
In the experiments, the dataset comprises multiple object positions and corresponding grasp poses acquired during data collection. As discussed in Section 5.1, each configuration was sampled over 20 discrete rotation steps to increase the amount of collected data. At each step, both the object and the gripper were rotated by 18°, resulting in a complete 360° rotation.
This chapter is organized as follows. Section 6.1 summarizes the camera calibration results. Section 6.2 reports the estimated camera poses with respect to the reference marker board. Section 6.3 presents the pose estimation results for the tester object. Finally, Section 6.4 evaluates the accuracy of the gripper fingertip pose estimation.

6.1 Camera calibration

Camera calibration was performed independently for each of the four cameras composing the multi-view acquisition setup. The calibration procedure estimates the intrinsic parameters and lens distortion coefficients required for accurate projection.
For each camera, a set of calibration images was acquired using a chessboard pattern, as described in Section 5.1. The pattern was observed from multiple viewpoints and orientations to ensure a robust estimation of the calibration parameters. Table 6.1 summarizes the number of detected chessboard corners and the corresponding reprojection error metrics for each camera, expressed in pixels. A lower reprojection error indicates better geometric consistency between the calibrated projection model and the observed image measurements.
During the calibration procedure, several calibration attempts were discarded due to unstable parameter estimation or excessively high reprojection errors. These issues were typically caused by insufficient coverage of the calibration pattern within the image plane or motion blur. Such

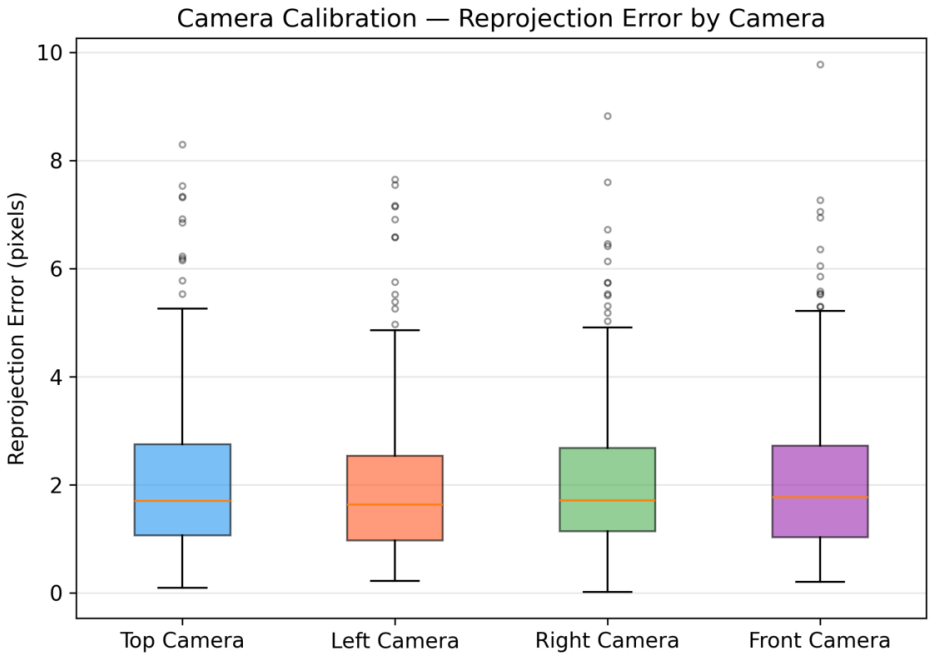

Figure 6.1: Comparison of the reprojection error across the four camera views. The plot summarizes the distribution of reprojection error across all measured points for each camera. The boxes represent the interquartile ranges, while the red lines indicate the median values. The whiskers extend to the most extreme non-outlier observations, and the points beyond them denote the outliers.

conditions degrade the accuracy of corner detection and, consequently, the estimation of intrinsic parameters.
Multiple calibration strategies were explored to improve the results. Initially, a ChArUco board was used and images were acquired from multiple viewpoints; however, the resulting reprojection errors were relatively high (> 3.5 pixels). Subsequently, calibration was performed using a standard chessboard pattern, which reduced the reprojection error to approximately (≈ 2.5 pixels). Finally, a larger chessboard pattern was employed under improved illumination conditions, yielding further improvements, although the reprojection error values remained relatively high.

These results are consistent with typical calibration performance for GoPro cameras, whose wide-angle lenses introduce significant distortion that can affect calibration accuracy. Figure 6.1 illustrates the distribution of the reprojection error across all measured points. The interquartile ranges (IQRs), represented by the boxes, are consistent across the four cameras. The median values, indicated by the red lines, are centered around 2px. Several outliers are also present, with errors of approximately 8–10px.
Figure 6.2 provides a complementary view by showing the frequency distribution of the reprojection errors. This representation indicates that no single camera exhibits significantly worse calibration performance com-

Calibration Results - Reprojection Error (px)				
Camera	Mean	Median	Std	N° Corners
Front Camera	2.0595	1.7677	1.3905	336
Right Camera	2.0843	1.7149	1.3897	288
Left Camera	1.9379	1.6316	1.3729	336
Top Camera	2.0765	1.7024	1.4087	336

Table 6.1: Reprojection error (mean, median, and standard deviation) for camera calibration and number of detected corners for each camera.

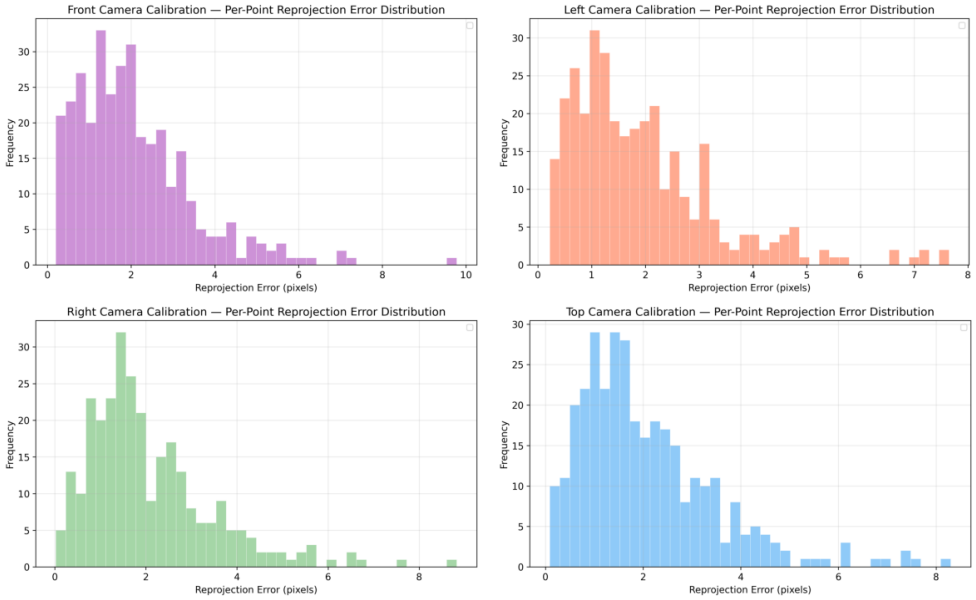

Figure 6.2: Comparison of the reprojection error frequencies across the four camera views. The plots indicate that most errors cluster between 0-3px, with a long right-tailed distribution extending toward higher error values.

pared to the others.

6.2 Camera pose estimation

For each camera, the rotation vector (*rvec*) and translation vector (*tvec*) were computed and subsequently converted into rotation matrices and camera position vectors expressed in the reference frame of the box marker pattern.

Table 6.2 reports the number of detected markers for each camera

Camera Pose Estimation — Reprojection Error (px)				
Camera	Mean	Median	Std	N° markers
Front Camera	3.0904	2.9917	1.4149	34
Right Camera	2.6933	2.5161	1.5249	38
Left Camera	2.8404	2.5767	1.6342	31
Top Camera	2.6471	2.3764	1.4583	31

Table 6.2: Reprojection error (mean, median, and standard deviation) for camera pose estimation and number of detected markers for each camera.

and the accuracy of the estimated camera poses, evaluated using the reprojection error associated with the marker-based PnP solution. This error is the average pixel distance between the detected marker corner locations and the projected positions of the corresponding 3D marker points, computed using the estimated camera pose.
Across all experimental trials, the cameras exhibited a consistent distribution; only the front camera showed an average pose-estimation reprojection error of approximately 3.1 pixels. Although this value is higher than the reprojection error obtained during intrinsic calibration, it should be noted that pose estimation errors are also influenced by marker-detection noise and image resolution.

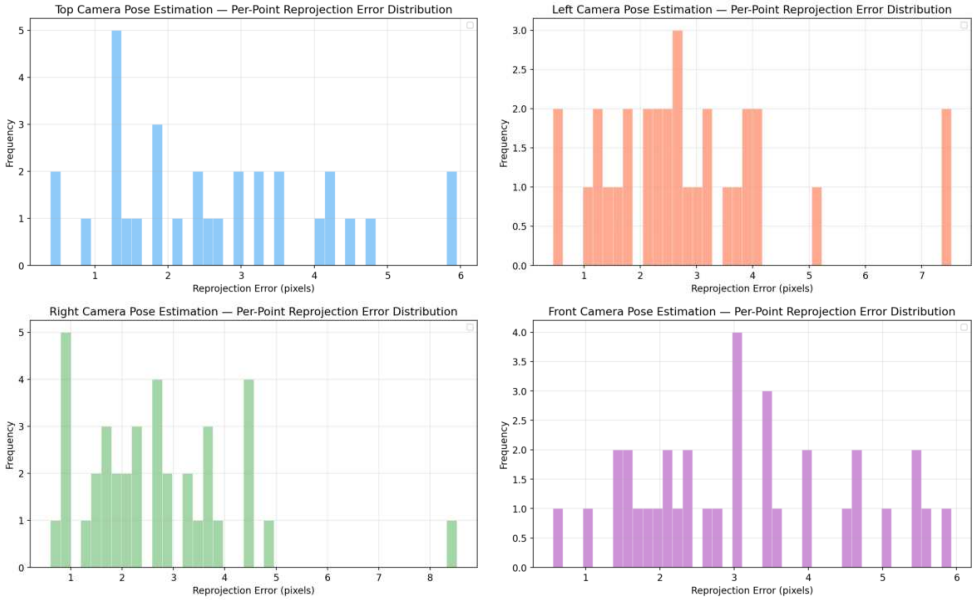

Figure 6.3: Comparison of the reprojection error frequencies across the four camera views. The right camera's distribution spans a wider range on the x-axis than the others, which show more tightly clustered reprojection errors with only a few low-error outliers.

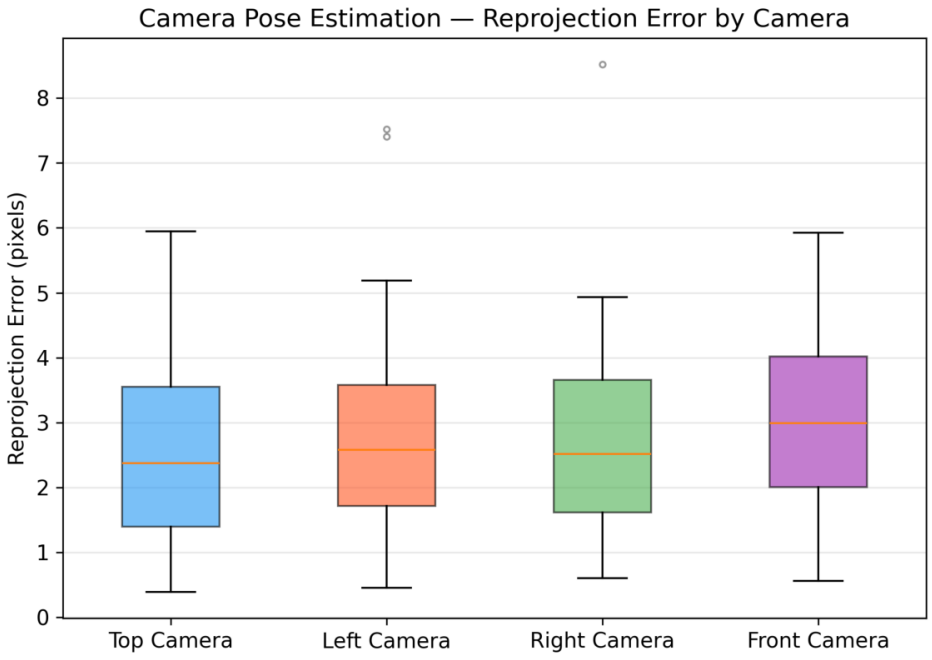

Figure 6.4: All boxes are nearly identical, showing a consistent error distribution. Additionally, the right camera shows a significantly larger outlier than the other views.

Figure 6.4 shows a relatively tight and consistent error distribution for all cameras, demonstrating uniformity. The IQRs are nearly identical across all views. In contrast, the plot summarizes a few outliers for the left and right cameras. Furthermore, Figure 6.3 indicates that the number of samples per camera is relatively small ($n \approx 31 - 38$), suggesting a limited number of detected feature points.

6.3 Object pose estimation

As discussed previously, the object's pose was estimated at each of the 20 rotation steps during the acquisition phase. Table 6.3 reports the reprojection error of the tester object for each camera view, along with the corresponding number of detected markers.

Figure 6.5 illustrates the distribution of the reprojection error direc-

Object Pose Estimation — Reprojection Error (px)				
Camera	Mean	Std	Median	N° markers
Front Camera	22.1856	12.887	18.9188	41
Left Camera	30.0312	16.1903	28.3995	41
Right Camera	26.4422	12.0578	25.1395	43
Top Camera	19.6012	10.7776	17.1841	39
Overall	24.6485	13.7279	23.875	164

Table 6.3: Reprojection error (mean, median, and standard deviation) for object pose estimation and number of detected markers for each camera.

tions. The mean error is approximately centered around $(0, 0)$, indicating a minimal systematic bias. The 1σ ellipse is nearly circular, suggesting isotropic error characteristics and high precision. Although the ellipse remains relatively compact, with a radius of approximately 20px, this

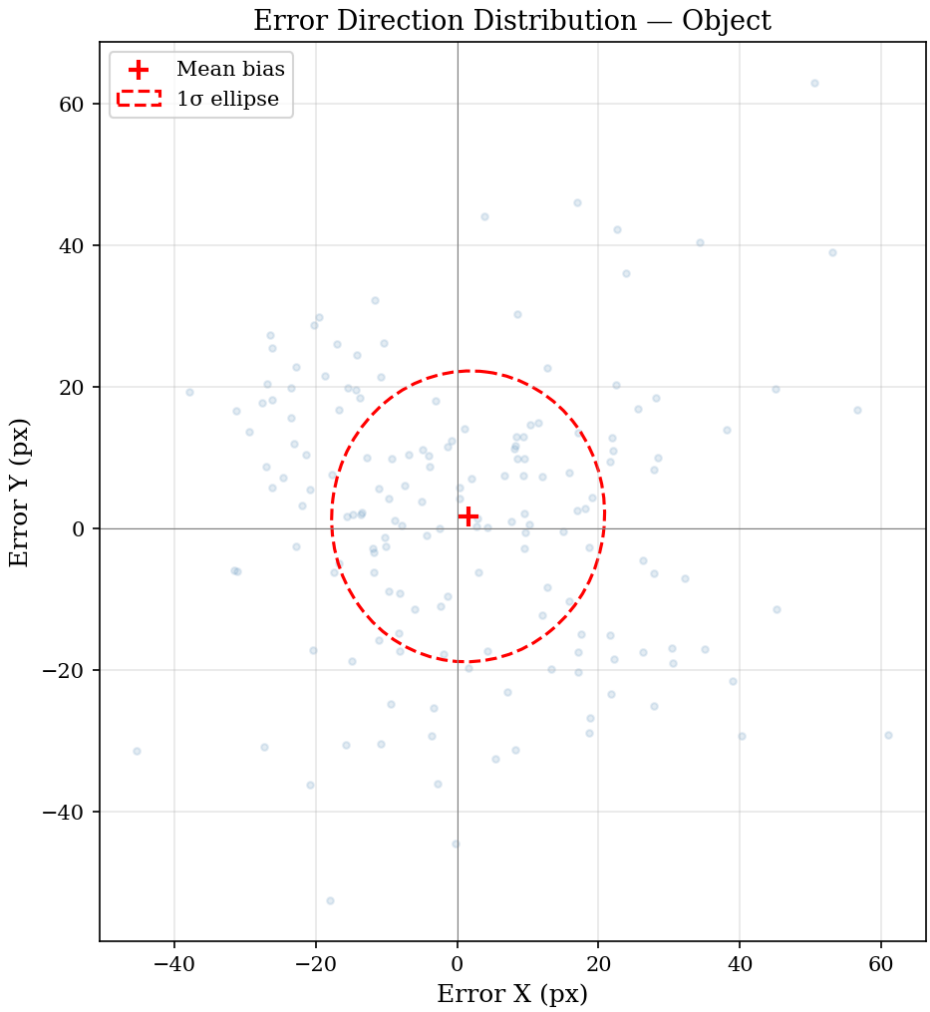

Figure 6.5: This plot summarizes the error decomposed in X and Y components separately, for each marker detection. The plot determines whether the system consistently produces a systematic error. Overall, the results indicate both high precision and good accuracy.

Figure 6.6: The front and top cameras outperform the others, achieving the lowest median reprojection errors. In contrast, the left camera demonstrates the worst performance, with a higher median, indicating greater variability and less consistent measurements.

value is still comparatively large.
Figure 6.6 shows that the front and top cameras achieve the best performance, with median reprojection errors below 20px. In addition, their IQRs are relatively small, indicating consistent error distributions.
In contrast, the left camera exhibits poorer performance, with a median error closer to 30px and a significant outlier of approximately 80px. The right camera exhibits intermediate behavior, with a median reprojection error of around 26 px.

6.4 Fingertips pose estimation

The gripper fingertips were estimated at each of the 20 rotation steps during the acquisition phase. For each step, the estimated rotation and translation vectors were converted into homogeneous 4×4 transformation matrices that represent the poses of the object and the fingertips in the reference coordinate system of the acquisition setup.

Table 6.4 reports the reprojection error of the gripper fingertips for each camera view, together with the corresponding number of detected markers. Table 6.5 presents the overall reprojection errors aggregated across the camera views, along with the total number of detected markers.
It can be observed that the thumb and middle fingers exhibit a lower number of detected markers compared to the other fingers. This behavior is expected, as these fingers are more frequently occluded by the object or by other fingers during the acquisition phase, reducing the visibility of their markers in some camera views. In particular, Figure 6.7 summarizes the distribution of the reprojection error directions. The results indicate a systematic bias for some fingers: the pinky finger shows a noticeable negative bias along the y-axis, while the thumb exhibits a negative bias along the x-axis. And generally, the other fingers show a positive bias in both axis. The 1σ confidence ellipses for the thumb and index fingers are larger, indicating greater uncertainty in the marker de-

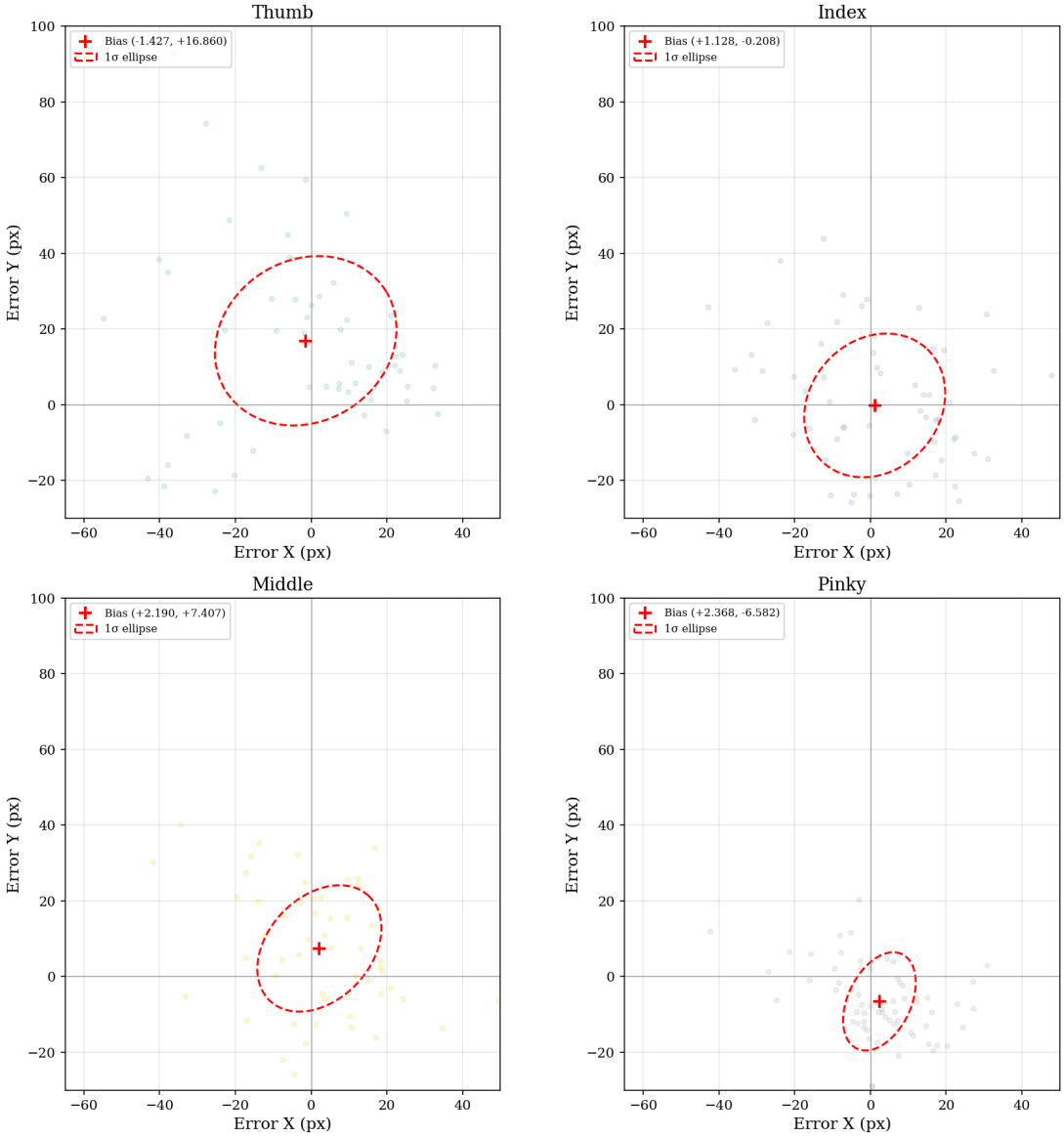

Figure 6.7: Comparison of the error direction scatter between all fingertips.

tection for these fingers. In contrast, the pinky finger shows the smallest ellipse, suggesting the most consistent and stable detection among all fingers.

Thumb - Reprojection error (px)				
Camera	Mean	Std	Median	N° markers
Front	25.2931	10.3963	25.1425	14
Left	30.148	17.6766	26.0567	14
Right	43.1064	25.1654	38.9082	13
Top	29.8648	10.0002	28.8696	14
Index - Reprojection error (px)				
Camera	Mean	Std	Median	N° markers
Front	23.5836	8.9916	23.8145	27
Left	22.7434	8.7647	21.167	12
Right	26.9238	11.787	29.9403	11
Top	25.2735	11.8133	24.1171	13
Middle - Reprojection error (px)				
Camera	Mean	Std	Median	N° markers
Front	17.6958	6.2468	17.6407	21
Left	21.6453	8.1332	20.8889	10
Right	27.7895	14.5019	26.8646	12
Top	23.7634	11.6312	23.1897	11
Pinky - Reprojection error (px)				
Camera	Mean	Std	Median	N° markers
Front	12.8489	5.7714	12.6963	27
Left	23.5444	5.3967	25.4577	12
Right	15.9963	10.2118	13.2546	13
Top	12.9538	7.4062	10.7325	14

Table 6.4: For each fingertip, the reprojection error (mean, median, and standard deviation) for its pose estimation and the number of detected markers for each camera.

Figure 6.8 further illustrates the distribution of reprojection error for each camera. It can be observed that for most fingers, the left and right cameras exhibit higher median errors and wider IQRs than the front camera. This suggests that these viewpoints experience greater difficulty in depth perception or feature detection. Finally, Figure 6.9 summarizes the tracking performance for each finger. The plot indicates that the thumb performs the worst, with the highest median error and several extreme outliers. In contrast, the pinky finger performs best, showing the lowest error values and the most consistent tracking results.

Cross-finger — Reprojection Error (px)				
Finger	Mean	Std	Median	N° markers
Thumb	31.903	18.0228	27.5497	55
Index	24.3555	10.2183	23.8145	63
Middle	21.9062	10.8175	19.3551	54
Pinky	15.4357	8.195	13.613	66

Table 6.5: Comparison of the overall reprojection error across all fingertips.

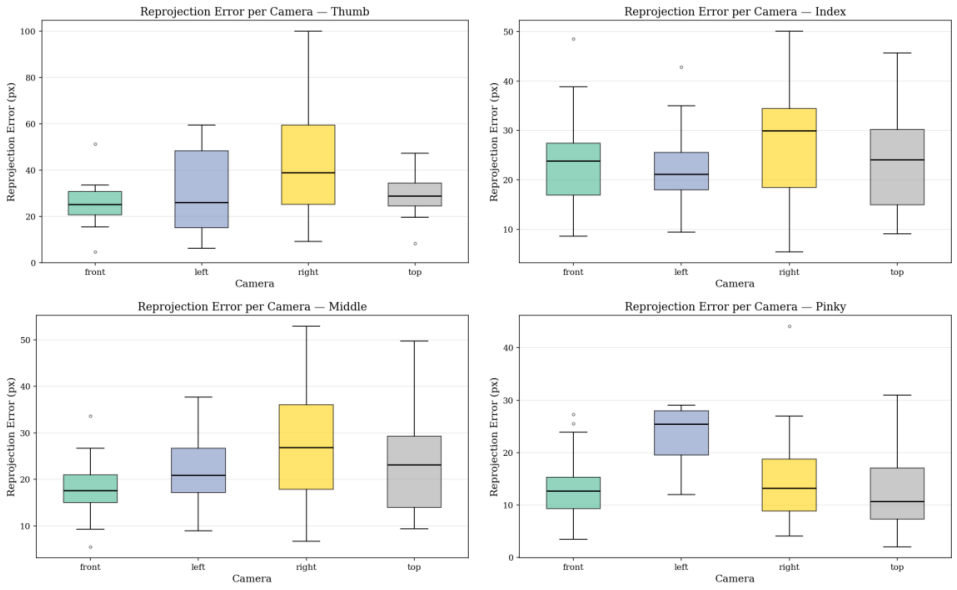

Figure 6.8: Comparison of reprojection error across the four camera views for each fingertip.

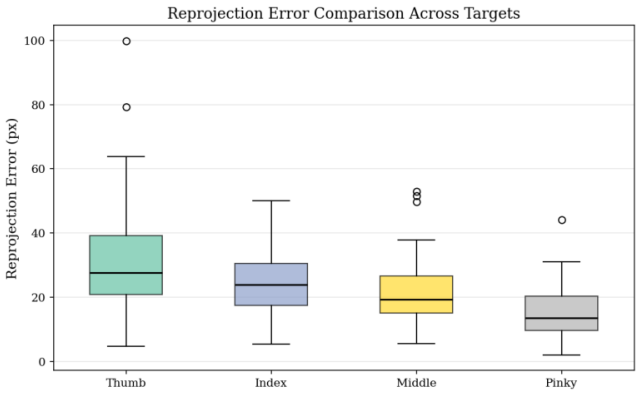

Figure 6.9: Plot of the overall reprojection error across all fingertips.

Chapter 7

Conclusions and Future Works

This thesis presented the design and development of a system for mass distribution estimation, aimed at collecting high-quality ground-truth data for learning-based applications. The motivation for this work is aligned with the objectives of the SUN project, which addresses the increasing demand for realistic physical behavior in XR applications. In such contexts, the lack of reliable real-world datasets often limits the fidelity of simulated interactions. In particular, this work contributes to the SUN project by developing a dataset designed to support its research activities.

The proposed system combines a calibrated vision pipeline with a structured and repeatable experimental setup, enabling consistent acquisition of interaction data. A key aspect of the system is the integration of camera calibration and camera pose estimation, which together establish a common spatial reference frame across all observations. The calibration process ensures stable intrinsic parameters and low reprojection error, while the camera pose estimation pipeline enables precise tracking of the manipulated object. This combination provides the geometric consis-

tency required for reconstructing object configurations and interaction dynamics.

In addition to the vision components, the system incorporates a modular test object and a hand-shaped gripper equipped with predefined fingertip contact points. The system relies on accurate pose estimation of both the object and the fingertips to capture the spatial relationships arising during manipulation.

The experimental results demonstrate that the proposed system provides a reliable baseline for modeling the spatial configuration of the acquisition setup. The intrinsic and extrinsic calibration achieved a high level of consistency across all cameras, ensuring a stable geometric reference for subsequent processing stages. During the pose estimation phase, the system maintained an acceptable level of accuracy for all cameras. However, the front camera exhibited slightly greater sensitivity to feature detection quality and potential extrinsic misalignments.

The most significant challenges were observed in fingertip pose estimation. In particular, the thumb consistently exhibited higher reprojection errors and a noticeable directional bias, likely due to more frequent occlusions and its wider range of motion during manipulation. In contrast, the pinky finger achieved the highest precision, suggesting more stable marker visibility across the multi-camera setup.

Despite these contributions, several limitations remain. The overall accuracy of the system is constrained by the sensing hardware, including camera resolution and noise characteristics typical of GoPro cameras, as well as by the precision of the calibration process. Small errors in calibration or mechanical alignment may propagate through the pipeline and affect pose estimation accuracy.

Future work may extend the system in several directions. Integrating additional sensing modalities, such as force or tactile sensors at the gripper fingertips, would provide richer information about object interactions and improve mass distribution estimation. Further improvements in calibration techniques, including continuous or online calibration, could en-

hance robustness and reduce sensitivity to environmental changes. Additionally, extending the modular object design to include different geometries could enable more manipulation strategies beyond fixed rotational sequences.

In conclusion, this work presents a framework for fast, scalable, automatic data acquisition for learning-based applications. Although limitations remain in terms of generalization and scalability, the proposed system establishes a solid foundation for future research toward more realistic and data-driven XR simulation systems.

Bibliography

[1] Paolo Cignoni et al. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[2] Hao-Shu Fang et al. "GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11441–11450. DOI: 10.1109/CVPR42600.2020.01146.

[3] Qian Feng et al. "Center-of-Mass-based Robust Grasp Planning for Unknown Objects Using Tactile-Visual Sensors". In: *CoRR* abs/2006.00906 (2020). arXiv: 2006.00906. URL: https://arxiv.org/abs/2006.00906.

[4] M. Fiala. "Comparing ARTag and ARToolkit Plus fiducial marker systems". In: *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*. 2005, 6 pp.-. DOI: 10.1109/HAVE.2005.1545669.

[5] Guillermo Garcia-Hernando et al. "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations". In: *CoRR* abs/1704.02463 (2017). arXiv: 1704.02463. URL: http://arxiv.org/abs/1704.02463.

[6] Sergio Garrido-Jurado et al. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. In: *Pattern Recognition* 47 (June 2014), pp. 2280–2292. DOI: 10.1016/j.patcog.2014.01.005.

[7] Dooyoung Kim et al. “Meta-Objects: Interactive and Multisensory Virtual Objects Learned from the Real World for Use in Augmented Reality”. In: *Human-Computer Interaction (cs.HC)* (2024). DOI: https://doi.org/10.48550/arXiv.2404.17179.

[8] Eric Krotkov. “Robotic Perception of Material.” In: Jan. 1995, pp. 88–95.

[9] Junbin Liu et al. “A Pig Mass Estimation Model Based on Deep Learning without Constraint”. In: *Animals* 13.8 (2023). ISSN: 2076-2615. DOI: 10.3390/ani13081376. URL: https://www.mdpi.com/2076-2615/13/8/1376.

[10] E. Mastinu et al. “HANDdata – first-person dataset including proximity and kinematics measurements from reach-to-grasp actions”. In: *www.nature.com/scientificdata* (2023). DOI: https://doi.org/10.1038/s41597-023-02313-w.

[11] Nikos Mavrakis and Rustam Stolkin. “Estimation and Exploitation of Objects’ Inertial Parameters in Robotic Grasping and Manipulation: A Survey”. In: *Surrey Space Centre, University of Surrey, United Kingdom* (2019). DOI: https://doi.org/10.48550/arXiv.1911.04397.

[12] *OpenCV - Camera Calibration and 3D Reconstruction*. URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html.

[13] Anton Poroykov et al. “Modeling ArUco Markers Images for Accuracy Analysis of Their 3D Pose Estimation”. In: *Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). Part 2* (Dec. 2020), short14–1. DOI: 10.51130/graphicon-2020-2-4-14.

[14] Trevor Standley et al. “image2mass: Estimating the Mass of an Object from Its Image”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. Ed. by Sergey Levine, Vincent Vanhoucke, and Ken Goldberg. Vol. 78. Proceedings of Machine Learning Research. PMLR, 13–15 Nov 2017, pp. 324–333. URL: https://proceedings.mlr.press/v78/standley17a.html.

[15] *SUN: Social and hUman ceNtered XR*. URL: https://www.sun-xr-project.eu/.

[16] S. Tanaka et al. “Active mass estimation with haptic vision”. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. 2004, 256–261 Vol.3. DOI: 10.1109/ICPR.2004.1334516.

[17] Claudio Vairo et al. *SUN: Social and hUman ceNtered XR - A Horizon Europe Project Paving the Way for the Widespread Adoption of Extended and Virtual Worlds*. CNR Edizioni, Nov. 2025. ISBN: 978-88-8080-801-5. DOI: 10.32079/ISTI-BOOK-2025/001. URL: https://www.edizioni.cnr.it.

[18] G. Vivek Venkatesh et al. “Estimation of Volume and Mass of Axi-Symmetric Fruits Using Image Processing Technique”. In: *International Journal of Food Properties* 18.3 (2015), pp. 608–626. DOI: 10.1080/10942912.2013.831444.