# Retail Imager

*Flexible Retail-Object Recognition System Utilizing 3D (without Machine Learning)*

**Matthew Boyd, Vivian Pazmany, Chenhui "Elvis" Zhu - Fall 2019**

BOSTON UNIVERSITY

## INTRODUCTION

Utilize 3D characteristics and 2D image features to quickly and automatically identify individual retail items in a cluttered box of products. Rather than require workers to specifically orient products for automated identification and sorting (i.e., barcodes showing), or exhaustively training machine learning models, our system is able to identify products regardless of orientation and to add new products with minimal setup time.

## GOAL

Improve efficiency in the picking and sorting of pick-up/delivery orders for retail stores.

### Why not use Machine Learning?

- Data labeling
- Resource requirements
- Training time
- **Overkill for a known product list**

## SYSTEM COMPONENTS

Software

- Depth Capture: **Intel RealSense SDK**
- 3D Calculations: **Point Cloud Library (PCL)**
- 2D Feature Extraction: **OpenCV, SIFT (SIFT)**
- Image Matching: **OpenCV, FLANN (FLANN)**
- Product Database: **MySQL**
- GUI: **NodeJS, Socket.IO, Bootstrap**

Hardware

Intel RealSense D415 Depth Camera, Windows 10 Laptop

## CHALLENGES

- **Image Match:** FLANN in lieu of unusable VDMS db
- **Square items:** Oriented Bounding Box may rotate 45°
- **Object tilt in z-plane**: less accurate image extraction
- **Object shapes:** designed for packaging with straight lines and flat surfaces (i.e., squares and rectangles)
- **2D Image Quality:** low image quality from Intel camera complicates identification

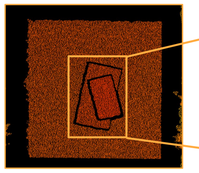
## CAPTURE

START

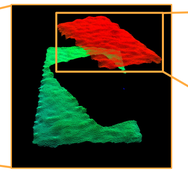
Image Scene — D415

Generate Point Cloud — PCL

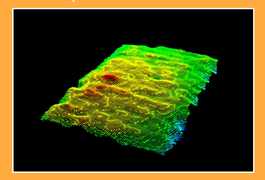
Filter Out Background — PCL



Extract Top-most Plane — PCL



**Result** Top-most Plane



Elapsed Time: **4.81 sec.**

## CHARACTERIZE

**Results** [128x128] Feature Array, Package Length (in mm)


Elapsed Time: **5.92 sec.**

Extract Image Features — SIFT

Crop Product Image from Scene — OpenCV

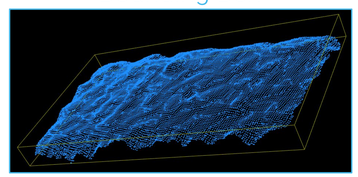
Calculate Dimensions & Corner Points — C++

$(x_1, y_1, z_1)$ $(x_2, y_2, z_2)$

mm x mm

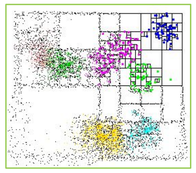
$(x_3, y_3, z_3)$ $(x_4, y_4, z_4)$

Generate Oriented Bounding Box — PCL



## IDENTIFY

ANN Search for Image Features in Index — FLANN



Send Product ID of Top Match to GUI — Python

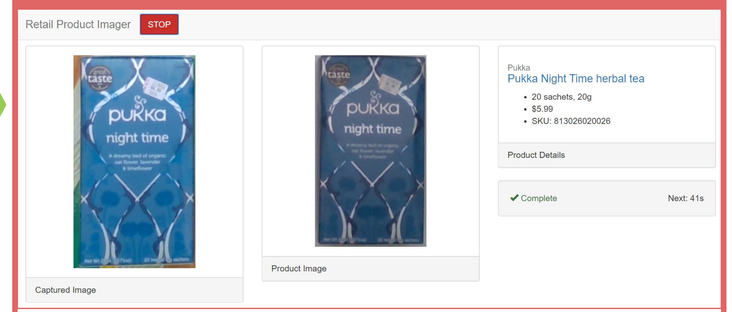
{id:5}

Query for Product Information

Elapsed Time: **7.61 sec.**

## RESULTS

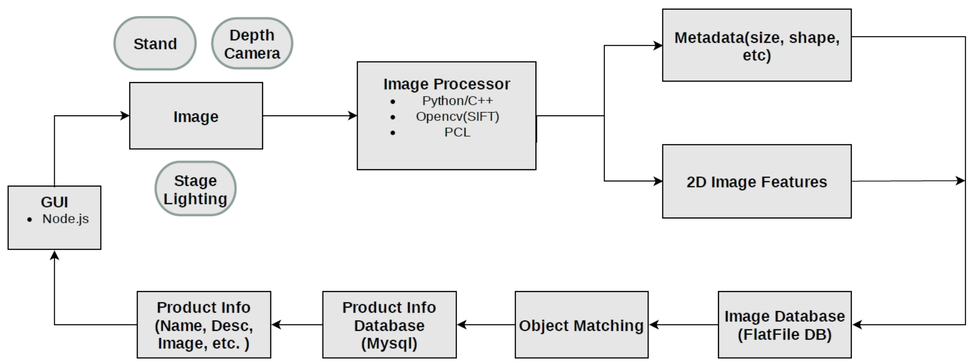
✓ **Match** in **7.61** total seconds!




**4900+ images** in index • **<3ms** for ANN search

**100% match** rate given our constraints

## SYSTEM DESIGN




## NEXT STEPS

- Robotic arms to sort products
- High-resolution camera for 2D image capture
- Additional packaging shapes (e.g., curved, irregular, etc.)
- Alternate features (e.g., weight, color, barcode, etc.)
- Automatically add new products (2D & 3D features)