Mathematical Analysis of Cylinder Pose Estimation for UR3 Robotic Pick-and-Place

6DOF Pose Estimation Pipeline

June 12, 2025

Contents

1 Introduction

This document provides a comprehensive mathematical analysis of the cylinder pose estimation pipeline designed for UR3 robotic pick-and-place operations. The system estimates 6DOF poses of cylindrical objects (6mm radius, 10mm height) using RGB-D camera data in a top-down viewing configuration.

2 Pipeline Overview

The complete pipeline consists of eight main mathematical steps:

1. YOLO Object Detection
2. 3D Point Cloud Extraction via Pinhole Camera Model
3. 2D Circle Fitting using Circumcenter Method
4. Point-to-Circle Distance Calculation
5. RANSAC-based Robust Circle Fitting
6. UR3-Constrained Cylinder Fitting
7. 6DOF Pose Estimation
8. Multi-Component Confidence Assessment

3 Step 1: YOLO Object Detection

3.1 Mathematical Formulation

Input: RGB image $I(u, v) \in \mathbb{R}^{H \times W \times 3}$ where (u, v) are pixel coordinates.

Output: Bounding box coordinates $\mathcal{B} = [x_{min}, y_{min}, x_{max}, y_{max}]$ and confidence score $c \in [0, 1]$.

Process: Deep neural network inference:

$$(\mathcal{B}, c) = \text{YOLO}(I(u, v)) \tag{1}$$

The YOLO network performs convolutional feature extraction and object localization through learned parameters. The mathematical details involve complex non-linear transformations that are not explicitly formulated here.

4 Step 2: 3D Point Cloud Extraction

4.1 Pinhole Camera Model

The fundamental transformation from 2D pixel coordinates to 3D world coordinates follows the pinhole camera model.

4.1.1 Camera Intrinsics

Let the camera intrinsic parameters be:

$$f_x, f_y : \text{focal lengths in pixels} \tag{2}$$

$$c_x, c_y : \text{principal point coordinates} \tag{3}$$

4.1.2 Back-Projection Mathematics

Given:

- Depth image $D(u, v)$ in millimeters
- Pixel coordinates (u, v) within bounding box $\mathcal{B}$

Transformation equations:

$$Z = \frac{D(u, v)}{1000.0} \quad \text{(convert mm to meters)} \tag{4}$$

$$X = \frac{(u - c_x) \times Z}{f_x} \tag{5}$$

$$Y = \frac{(v - c_y) \times Z}{f_y} \tag{6}$$

4.1.3 Mathematical Derivation

The perspective projection model relates 3D points to 2D pixels:

$$u = f_x \cdot \frac{X}{Z} + c_x \tag{7}$$

$$v = f_y \cdot \frac{Y}{Z} + c_y \tag{8}$$

Inverting these equations yields the back-projection formulas:

$$X = (u - c_x) \cdot \frac{Z}{f_x} \tag{9}$$

$$Y = (v - c_y) \cdot \frac{Z}{f_y} \tag{10}$$

Output: Point cloud $\mathcal{P} = \{P_i = [X_i, Y_i, Z_i] \mid i = 1, \ldots, N\}$ where N is the number of valid depth pixels.

5 Step 3: 2D Circle Fitting - Circumcenter Method

5.1 Three-Point Circle Fitting

Given three non-collinear 2D points, we can uniquely determine a circle using the circumcenter calculation.

5.1.1 Input

Three 2D points: $A = [a_x, a_y]$, $B = [b_x, b_y]$, $C = [c_x, c_y]$

5.1.2 Collinearity Check

First, check if points are collinear using the determinant:

$$d = 2 \times (a_x(b_y - c_y) + b_x(c_y - a_y) + c_x(a_y - b_y)) \tag{11}$$

If $|d| < \epsilon$ (typically $\epsilon = 10^{-10}$), the points are collinear and no unique circle exists.

5.1.3 Circumcenter Calculation

For non-collinear points, the circumcenter coordinates are:

$$u_x = \frac{(a_x^2 + a_y^2)(b_y - c_y) + (b_x^2 + b_y^2)(c_y - a_y) + (c_x^2 + c_y^2)(a_y - b_y)}{d} \tag{12}$$

$$u_y = \frac{(a_x^2 + a_y^2)(c_x - b_x) + (b_x^2 + b_y^2)(a_x - c_x) + (c_x^2 + c_y^2)(b_x - a_x)}{d} \tag{13}$$

5.1.4 Radius Calculation

The circle radius is the distance from the circumcenter to any of the three points:

$$r = \sqrt{(a_x - u_x)^2 + (a_y - u_y)^2} \tag{14}$$

5.1.5 Verification

Verify that all three points are equidistant from the center:

$$r_1 = \sqrt{(a_x - u_x)^2 + (a_y - u_y)^2} \tag{15}$$

$$r_2 = \sqrt{(b_x - u_x)^2 + (b_y - u_y)^2} \tag{16}$$

$$r_3 = \sqrt{(c_x - u_x)^2 + (c_y - u_y)^2} \tag{17}$$

The maximum error should be within tolerance:

$$\max(|r_1 - r_2|, |r_2 - r_3|, |r_1 - r_3|) \leq \tau \tag{18}$$

where $\tau = 0.001$ m (1mm tolerance).

6 Step 4: Point-to-Circle Distance

6.1 Distance Metric

For a given point $P = [p_x, p_y]$ and circle with center $[u_x, u_y]$ and radius r, the distance from the point to the circle circumference is:

$$d_{circumference}(P) = \left| \sqrt{(p_x - u_x)^2 + (p_y - u_y)^2} - r \right| \tag{19}$$

This metric measures how far a point deviates from the ideal circle, which is crucial for the RANSAC inlier determination.

7 Step 5: RANSAC Circle Fitting

7.1 Algorithm Description

RANSAC (Random Sample Consensus) provides robust fitting in the presence of outliers.

7.2 Mathematical Conditions

7.2.1 Inlier Determination

A point P_i is classified as an inlier if:

$$\left| \sqrt{(p_{i,x} - u_x)^2 + (p_{i,y} - u_y)^2} - r \right| \leq \epsilon \tag{20}$$

where $\epsilon = 0.0005$ m (0.5mm tolerance for UR3 precision requirements).

Algorithm 1 RANSAC 2D Circle Fitting

Require: Point set $\mathcal{P}_{2D} = \{P_i \mid i = 1, \ldots, N\}$
Require: Maximum iterations K_{max}, inlier threshold ϵ, minimum inliers N_{min}
Ensure: Best circle parameters (u_x^*, u_y^*, r^*) and inlier mask

$best\_count \leftarrow 0$
$best\_params \leftarrow \emptyset$
for $k = 1$ **to** K_{max} **do**
Randomly sample 3 points $\{P_{i_1}, P_{i_2}, P_{i_3}\}$
$(u_x, u_y, r) \leftarrow$ CircumcenterFit$(P_{i_1}, P_{i_2}, P_{i_3})$
if fit is valid **then**
Apply radius filter: $\frac{|r - r_{expected}|}{r_{expected}} \leq 0.5$
if radius filter passes **then**
$inliers \leftarrow 0$
for each point $P_j \in \mathcal{P}_{2D}$ **do**
if $d_{circumference}(P_j) \leq \epsilon$ **then**
$inliers \leftarrow inliers + 1$
end if
end for
if $inliers > best\_count$ **then**
$best\_count \leftarrow inliers$
$best\_params \leftarrow (u_x, u_y, r)$
end if
end if
end if
end for
return $best\_params$ if $best\_count \geq N_{min}$, else failure

7.2.2 Radius Filtering

To reject obviously incorrect circles, we apply a radius filter:

$$\frac{|r_{fitted} - r_{expected}|}{r_{expected}} \leq 0.5 \tag{21}$$

where $r_{expected} = 0.006$ m (6mm cylinder radius).

8 Step 6: UR3-Constrained Cylinder Fitting

8.1 Constraint Application

The UR3 robotic setup provides strong geometric constraints that we exploit for improved fitting accuracy.

8.1.1 Geometric Constraints

- Camera orientation: perpendicular to work plane (top-down view)
- Cylinder orientation: always vertical with axis $\vec{a} = [0, 0, -1]$
- Known dimensions: radius $r = 0.006$ m, height $h = 0.01$ m

8.2 Mathematical Process

8.2.1 Step 1: 3D to 2D Projection

Project 3D points to 2D by dropping the Z-coordinate:

$$\mathcal{P}_{2D} = \{[P_{i,x}, P_{i,y}] \mid P_i = [P_{i,x}, P_{i,y}, P_{i,z}] \in \mathcal{P}_{3D}\} \tag{22}$$

8.2.2 Step 2: 2D Circle Fitting

Apply RANSAC circle fitting to $\mathcal{P}_{2D}$ to obtain:

$$(u_x, u_y, r_{2D}, \mathcal{M}_{inliers}) \tag{23}$$

where $\mathcal{M}_{inliers}$ is the boolean inlier mask.

8.2.3 Step 3: Z-Position Analysis

For inlier points, analyze Z-coordinates to determine cylinder positioning:

$$\mathcal{Z}_{inliers} = \{P_{i,z} \mid \mathcal{M}_{inliers}[i] = \text{true}\} \tag{24}$$

$$Z_{top} = \text{median}(\mathcal{Z}_{inliers}) \tag{25}$$

$$Z_{bottom} = Z_{top} + h \tag{26}$$

$$Z_{center} = \frac{Z_{top} + Z_{bottom}}{2} \tag{27}$$

The median is used instead of mean to be robust against outliers.

8.2.4 Step 4: 3D Cylinder Reconstruction

Construct the final 3D cylinder parameters:

$$\text{Center} = [u_x, u_y, Z_{center}] \tag{28}$$
$$\text{Top Center} = [u_x, u_y, Z_{top}] \tag{29}$$
$$\text{Bottom Center} = [u_x, u_y, Z_{bottom}] \tag{30}$$
$$\text{Axis} = [0, 0, -1] \tag{31}$$
$$\text{Radius} = r_{2D} \tag{32}$$
$$\text{Height} = h \tag{33}$$

9 Step 7: 6DOF Pose Estimation

9.1 Position Estimation

The 3D position of the cylinder in the camera coordinate frame is simply the cylinder center:

$$\vec{t} = [u_x, u_y, Z_{center}]^T \tag{34}$$

9.2 Rotation Estimation

For the UR3 top-down grasping scenario, we use a canonical rotation matrix since the exact rotational orientation around the cylinder axis is not critical for grasping.

9.2.1 Rotation Matrix

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{35}$$

This matrix represents:

- X-axis: $[1, 0, 0]$ (unchanged)
- Y-axis: $[0, 1, 0]$ (unchanged)
- Z-axis: $[0, 0, -1]$ (points toward camera along cylinder axis)

9.2.2 Euler Angles

For the canonical orientation, the Euler angles (XYZ convention) are:

$$[\phi, \theta, \psi] = [0, 0, 0] \text{ radians} \tag{36}$$

9.3 6DOF Pose Representation

The complete 6DOF pose is represented as:

$$\mathcal{T} = \begin{bmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & u_x \\ 0 & 1 & 0 & u_y \\ 0 & 0 & -1 & Z_{center} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{37}$$

10 Step 8: Multi-Component Confidence Assessment

10.1 Confidence Components

The pose confidence is computed as a weighted combination of four components, prioritizing position accuracy over geometric fitting accuracy.

10.1.1 Component 1: Inlier Ratio (50% weight)

$$S_{inlier} = \frac{N_{inliers}}{N_{total}} \tag{38}$$

where $N_{inliers}$ is the number of points classified as inliers and N_{total} is the total number of points.

10.1.2 Component 2: Position Stability (30% weight)

Measure how consistently inlier points agree on the center position:

XY Position Consistency:

$$\bar{P}_{XY} = \frac{1}{N_{inliers}} \sum_{i \in \text{inliers}} [P_{i,x}, P_{i,y}] \tag{39}$$

$$\sigma_{XY} = \text{std}\left(\{\|[P_{i,x}, P_{i,y}] - \bar{P}_{XY}\|_2 \mid i \in \text{inliers}\}\right) \tag{40}$$

$$S_{XY} = e^{-\sigma_{XY}/0.001} \tag{41}$$

Z Position Consistency:

$$\sigma_Z = \text{std}\left(\{P_{i,z} \mid i \in \text{inliers}\}\right) \tag{42}$$

$$S_Z = e^{-\sigma_Z/0.002} \tag{43}$$

Combined Position Stability:

$$S_{position} = \frac{S_{XY} + S_Z}{2} \tag{44}$$

The exponential decay function rewards tight clustering of inlier points around the estimated center.

10.1.3 Component 3: Point Density (15% weight)

$$S_{density} = \min\left(1.0, \frac{N_{inliers}}{50}\right) \tag{45}$$

This component saturates at 50 inliers, reflecting diminishing returns from additional points.

10.1.4 Component 4: Radius Reasonableness (5% weight)

Provide gentle penalty for unreasonable radius estimates:

$$\epsilon_{radius} = \frac{|r_{fitted} - r_{expected}|}{r_{expected}} \tag{46}$$

$$S_{radius} = \begin{cases} 1.0 & \text{if } \epsilon_{radius} < 1.0 \\ \max(0.3, 1.0 - \epsilon_{radius}) & \text{otherwise} \end{cases} \tag{47}$$

10.2 Final Confidence Score

The overall confidence is computed as:

$$\boxed{C = 0.50 \cdot S_{inlier} + 0.30 \cdot S_{position} + 0.15 \cdot S_{density} + 0.05 \cdot S_{radius}} \tag{48}$$

The weights reflect the importance hierarchy for UR3 grasping applications:

1. Inlier consensus (most important)
2. Position stability (critical for grasping accuracy)
3. Point density (more data = better estimate)
4. Radius check (sanity check only)

11 Final Output

11.1 6DOF Pose Format

The complete pipeline outputs a 6DOF pose in the camera coordinate frame:

$$\begin{aligned}
\text{Position} &: [X, Y, Z] \text{ in meters} && (49)\\
\text{Rotation (Euler)} &: [R_x, R_y, R_z] \text{ in radians (XYZ convention)} && (50)\\
\text{Rotation (Matrix)} &: R \in SO(3) && (51)\\
\text{Confidence} &: C \in [0, 1] && (52)
\end{aligned}$$

11.2 Coordinate Frame

The pose is expressed in the camera coordinate frame where:

- X-axis: points right in the image
- Y-axis: points down in the image
- Z-axis: points away from camera into the scene

12 Mathematical Properties and Guarantees

12.1 Robustness Properties

- **Outlier Robustness**: RANSAC can handle up to 50% outliers
- **Noise Tolerance**: Sub-millimeter tolerance ($\epsilon = 0.5$ mm)
- **Geometric Consistency**: Constraints ensure physically plausible solutions

12.2 Accuracy Expectations

Under ideal conditions with the UR3 setup:

- **Position Accuracy**: ± 1 mm in X,Y; ± 2 mm in Z
- **Rotation Accuracy**: Canonical orientation (rotation around Z-axis not critical)
- **Processing Time**: < 100 ms per frame
- **Success Rate**: $> 95\%$ for well-lit, unoccluded cylinders

13 Conclusion

This mathematical framework provides a robust and accurate method for 6DOF cylinder pose estimation specifically optimized for UR3 robotic pick-and-place operations. The constrained approach leverages domain knowledge to achieve superior performance compared to general-purpose 3D fitting methods.

Key innovations include:

1. UR3-specific geometric constraints
2. Position-prioritized confidence scoring
3. Robust RANSAC implementation with radius filtering
4. Multi-component confidence assessment

The mathematical rigor ensures reliable performance in real-world robotic applications while maintaining computational efficiency suitable for real-time operation.