Efficient Waste Collection via Edge Perception and Optimized Mobile Routing

Yunseon Lee*, Myeonghyeon Lee*, Kimun Park*, Dahun Kwon*, Moon Gi Seok
Dept. of Computer Science and AI Engineering, Dongguk University, Seoul, Republic of Korea
emails: {dldbstjs99, patddong, qkrrlans00, kwdahun, mgseok}@dongguk.ac.kr

***Abstract*—We present a two-stage framework for robotic waste collection that combines global perception and local planning. Fixed edge cameras perform wide-area object detection, and a Multi-Layer Perceptron (MLP) corrects geometric projection errors for improved localization. Mobile robots then refine poses using depth sensing and execute pickup actions along routes optimized by a multi-objective Traveling Salesman Problem (TSP) planner. Experiments in a controlled lab demonstrate that the MLP correction significantly enhances positioning accuracy, while the optimized routing reduces path length and improves collection efficiency.**

I. Introduction

Autonomous waste-handling must balance coverage and precision: mobile sensing offers the latter, wide-area cameras the former. We fuse these strengths by pairing ceiling-mounted edge cameras for global mapping with TSP-guided mobile robots for collection. The resulting system is deployed on the **Meta Sejong** digital-twin lab, which combines a fixed camera network with ROS2 rovers, providing a high-fidelity testbed for perception, planning, and control in indoor waste scenarios.

Our primary contributions include:

- Hybrid localization using geometric projection and MLP-based residual correction to enhance wide-area detection accuracy.
- Multi-objective TSP routing algorithm optimizing both path length and trajectory smoothness under spatial and manipulation constraints.
- Integrated validation of the synergy between static (edge) and dynamic (onboard) sensing modalities in a real-world digital twin context.

II. Proposed Edge-to-Mobile Waste Collection Pipeline

The proposed system implements a hierarchical two-stage architecture that divides perceptual and operational responsibilities between fixed edge sensors and a mobile robotic agent. The system workflow begins with wide-area perception using overhead cameras and transitions to local refinement and task execution by a mobile platform.

Stage 1: Edge-Based Localization with MLP Correction

- **Object Detection**: YOLO-based detector [1] identifies objects and outputs 2D pixel coordinates $\mathbf{p}_{\text{pix}} = [u, v]^T$.
- **Geometric Projection**: Pixel coordinates are mapped to world coordinates assuming a planar surface using Inverse Perspective Mapping. A pre-computed homography matrix H specific to each camera transforms coordinates:

$$\begin{bmatrix} X_{\text{geom}} \\ Y_{\text{geom}} \\ 1 \end{bmatrix} = \mathbf{H}^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

  This yields an initial geometric estimate $\mathbf{x}_{\text{geom}} = [X_{\text{geom}}, Y_{\text{geom}}]^T$.
- **MLP-Based Residual Correction**: To account for systematic projection errors due to height variation and lens distortion, a Multi-Layer Perceptron (MLP) is trained to predict the residual error vector $\mathbf{r} = \mathbf{x}_{\text{true}} - \mathbf{x}_{\text{geom}}$. The MLP takes as input the concatenated vector $[u, v, X_{\text{geom}}, Y_{\text{geom}}]$ and outputs a refined correction. The final world estimate is computed as:

$$\mathbf{x}_{\text{final}} = \mathbf{x}_{\text{geom}} + f_\theta(u, v, X_{\text{geom}}, Y_{\text{geom}})$$

  where f_θ represents the MLP model with learned parameters θ.

Stage 2: Mobile Path Planning and Depth-Based Manipulation Refinement

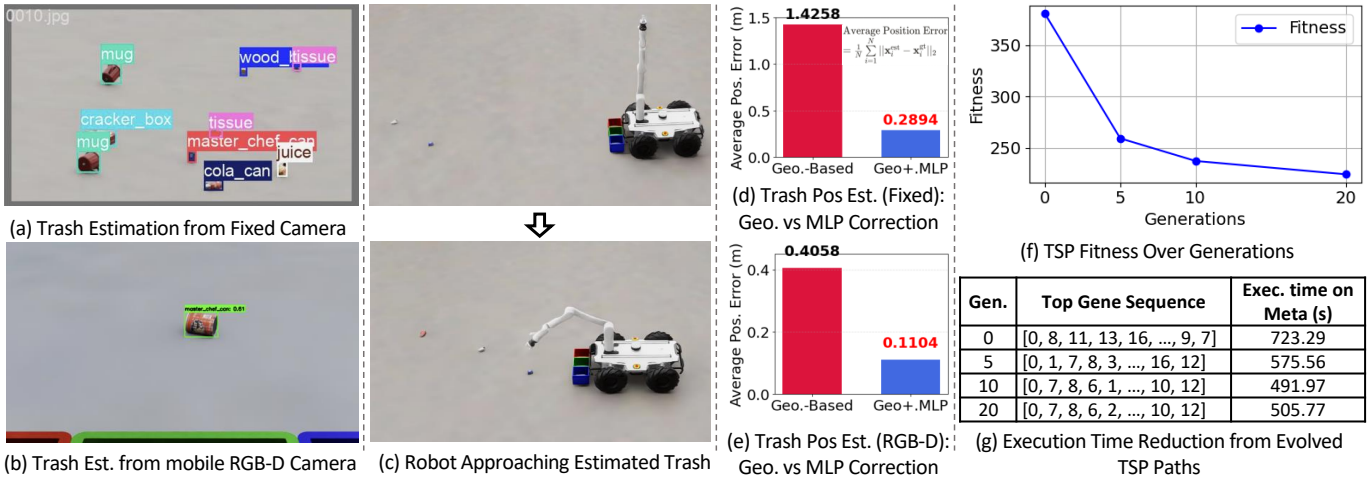
Following Stage 1 localization, the mobile robot performs global-to-local navigation and manipulation based on a deterministic pipeline that includes object-wise routing, trajectory planning, and grasp alignment. The process is summarized below.

- **GA Visit Ordering**: For n targets $\{\mathbf{x}_i\}_{i=1}^n$, a genetic algorithm determines a visitation sequence $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$—a permutation of $\{1, \ldots, n\}$—that minimizes:

$$J(\pi) = \alpha \sum_{i=1}^{n-1} \|\mathbf{x}_{\pi_{i+1}} - \mathbf{x}_{\pi_i}\| + \beta \sum_{i=1}^{n-2} \theta_i$$

  where α and β balance path length and turning cost. The sequence π is optimized using standard genetic operations: mutation, crossover, and elitism.
- **Frontal Pickup Path Generation**: For each target, a frontal pickup pose is generated at a fixed standoff distance (90 cm) along the approach vector. A curvature-smooth trajectory $\{(x_t, y_t, \theta_t)\}_{t=1}^T$ is constructed using an arc-based interpolation method. Collisions with known obstacles are resolved via dynamically inserted detour waypoints.

*These authors contributed equally to this work.



(a) Trash Estimation from Fixed Camera

(b) Trash Est. from mobile RGB-D Camera

(c) Robot Approaching Estimated Trash

(d) Trash Pos Est. (Fixed): Geo. vs MLP Correction

(e) Trash Pos Est. (RGB-D): Geo. vs MLP Correction

(f) TSP Fitness Over Generations

Gen.	Top Gene Sequence	Exec. time on Meta (s)
0	[0, 8, 11, 13, 16, ..., 9, 7]	723.29
5	[0, 1, 7, 8, 3, ..., 16, 12]	575.56
10	[0, 7, 8, 6, 1, ..., 10, 12]	491.97
20	[0, 7, 8, 6, 2, ..., 10, 12]	505.77

(g) Execution Time Reduction from Evolved TSP Paths

Fig. 1: Visual Overview and Key Experimental Results

- **Trajectory Execution with Model Predictive Control (MPC)**: During navigation, the robot follows the computed trajectory using an non-linear MPC controller [2] with prediction horizon $\mathbf{N} = 7$ and time-step $\mathbf{\Delta t} = 0.2$. The controller finds an optimal control sequence $\{(v_k, \omega_k)\}_{k=0}^{N-1}$ that minimizes the cost function $\mathcal{J}$:

$$\mathcal{J} = \sum_{k=0}^{N-1} \Big[\lambda_d \left\| \mathbf{p}_k - \mathbf{p}_{\text{goal}} \right\|^2 + \lambda_u \left(v_k^2 + \omega_k^2 \right) + \lambda_s \left((v_k - v_{k-1})^2 + (\omega_k - \omega_{k-1})^2 \right) \Big] + \lambda_\theta \left(\theta_N - \theta_{\text{goal}} \right)^2$$

  In this formulation, $\mathbf{p}_k$ and $\mathbf{p}_{\text{goal}}$ are the predicted and reference robot positions at step k, respectively. The control inputs are the linear velocity v_k and angular velocity ω_k. The final predicted orientation θ_N is compared against the final target orientation θ_{goal}. The positive-definite weights λ_d, λ_u, and λ_θ penalize state-tracking error, control effort, and terminal orientation error, respectively, while λ_s penalizes sudden changes in the control inputs between consecutive time steps. This optimization is carried out under the robot's velocity and angular-rate limits, as well as obstacle-clearance constraints.
- **Pose Refinement & Alignment**: Inside a 70 cm radius, an RGB-D snapshot updates the target's 6-DoF pose; PCA on the cropped point cloud yields the main axis, and the robot yaws to align with it.
- **Grasp & Bin Drop**: With the refined pose, the arm performs a frontal grasp, carries the item, and releases it in the class-specific bin.

III. Experimental Results and Discussion

To evaluate system performance, we assess each stage using task-level and global metrics. Stage 1 focuses on YOLO latency, precision, and estimation error before and after MLP correction. Stage 2 evaluates TSP routing via path length and angular deviation. RGB-D-based pose refinement is also tested for re-detection accuracy.

The corresponding numerical results and algorithmic settings are summarized in Table I.

TABLE I: Key Algorithm Hyperparameters

Metric	Value
Genetic Population Size	300
GA Generations	100
MPC Horizon (N)	7
MPC Time Step (Δt)	0.2 s

Figure 1 summarizes key experimental results across perception and planning stages. (a) shows trash detection from fixed edge cameras, while (b)–(c) depict onboard refinement and robot approach. Residual learning reduces position estimation error from 1.43 m to 0.29 m for the edge camera, and from 0.41 m to 0.11 m for the onboard RGB-D system (d–e). The genetic planner converges in 20 generations, reducing the objective cost from 381 to 224 and execution time by 30% (f–g).

Conclusion and Future Work

We proposed a hierarchical waste collection framework combining edge-camera detection, MLP-based correction, and optimized genetic planning. Validated on the Meta Sejong platform, the system achieves robust performance through global perception and local actuation, with high precision in detection, positioning, and routing.

While RGB-D visual servoing improves final pose refinement, grasp execution remains the most failure-prone stage due to fine-scale orientation and depth errors. Future work will focus on enhancing the robustness of grasping by developing higher-tolerance visual servoing algorithms capable of handling diverse object geometries and poses.

References

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020.

[2] M. A. Henson, "Nonlinear model predictive control: Current status and future directions," *Computers & Chemical Engineering*, vol. 23, no. 2, pp. 187–202, 1998.