UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

MONOCULAR CAMERA-BASED AUTONOMOUS SIDEWALK NAVIGATION USING EFFICIENT MACHINE-LEARNING-DRIVEN IMAGE SEGMENTATION AND IMAGE-SPACE PATH PLANNING

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

LKHANAAJAV MIJIDDORJ

Norman, Oklahoma

2026

MONOCULAR CAMERA-BASED AUTONOMOUS SIDEWALK NAVIGATION USING EFFICIENT MACHINE-LEARNING-DRIVEN IMAGE SEGMENTATION AND IMAGE-SPACE PATH PLANNING

A THESIS APPROVED FOR THE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Binbin Weng, Chair

Dr. Bin Xu

Dr. Samuel Cheng

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Binbin Weng, for his guidance, support, and encouragement throughout this research. His expertise in embedded systems and computer vision was invaluable in shaping this work.

I am grateful to the members of my thesis committee, Dr. Bin Xu and Dr. Samuel Cheng, for their insightful feedback and constructive suggestions.

I would also like to thank my collaborators—Tyler Beringer, Bilguunzaya Mijiddorj, Yan Yang, and Alex N. Ho—for their contributions to data collection, system testing, and many productive discussions that improved the quality of this work.

Finally, I thank my family for their unwavering support and encouragement during my graduate studies at the University of Oklahoma.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARM	Advanced RISC Machine (processor architecture)
BEV	Bird's-Eye View
COCO	Common Objects in Context (dataset)
CPU	Central Processing Unit
DT	Distance Transform
DWA	Dynamic Window Approach
EDT	Euclidean Distance Transform
EMA	Exponential Moving Average
FPS	Frames Per Second
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GT	Ground Truth
HUD	Heads-Up Display
IoU	Intersection over Union
IPM	Inverse Perspective Mapping
NMEA	National Marine Electronics Association
ONNX	Open Neural Network Exchange
RGB	Red, Green, Blue (color space)
ROS	Robot Operating System

ABSTRACT

This thesis studies sidewalk navigation for a small electric scooter using a deliberately simple sensor setup: one forward-facing commodity RGB camera per run, CPU-only processing, no LiDAR, no stereo pair, and no multi-camera rig. The video data were collected with phone/action-camera hardware, including an iPhone camera and a GoPro-style camera, so the work is framed around low-cost monocular sensing rather than specialized robotics sensors. The pipeline was developed through four design iterations, beginning with BEV skeleton-graph construction and ending with template-approval planning. The evaluation compares ten path-planning methods from geometric, cost-field, topological, and template-based families in both bird's-eye view (BEV) and image-space coordinates. For segmentation, a compact SegFormer-B0 student (3.7M parameters) is trained from OneFormer Swin-L pseudo-labels and then fine-tuned on human-corrected masks, reaching IoU 0.964 at 14.0 ms. Held-out IoU is 0.974, higher than the training-video IoU of 0.963.

For midpoint planning, the image-space planner obtains a lateral center error of 14.5 px with a runtime of 3.0 ms per frame on 448 hand-annotated campus sidewalk frames across eight video sequences. This is the best accuracy/runtime result among the ten planners. It also ranks significantly higher than the other methods (Friedman $p < 10^{-61}$; all nine pairwise comparisons between midpoint and the other planners are significant after Bonferroni correction; 32 of the 45 total pairwise comparisons are significant).

Although the midpoint planner has the best accuracy/runtime tradeoff among the tested planners, it is still intent-agnostic: it follows the visible corridor center and cannot perform commanded turns. For this reason, the final system keeps BEV for metric-scale turn compliance. The GPS-conditioned waypoint-turn planner checks the proposed turn against the BEV corridor using a simple containment safety guard.

The most costly part of BEV processing for turn validation is not the projection step, but

the search for a path. Path extraction through skeleton search followed by Dijkstra takes 540–690 ms per frame. By comparison, the path-acceptance test used in template approval takes only $3 - -5$ ms per frame. By using fixed-arc verification and family-reuse hysteresis in place of the noise-sensitive skeleton graph, the template-approval planner reduces mean heading angle by 40.6% and halves the total number of path-source switches compared with the skeleton baseline.

The image-space midpoint path can still contain noticeable heading-angle variation (4.5 deg/frame on average and 18.5 deg at the 95th percentile), which motivates the final temporal path-smoothing stage. On the 1800-frame accepted run, the method achieves 100% template acceptance with mean segmentation intersection-over-union (IoU) of 0.954 while maintaining 9.97 FPS.

In summary, the final dual-domain system uses the image-plane midpoint planner for forward motion, BEV for metric-scale turn-compliance checking and corridor extraction, and temporal smoothing for path stability. All results are evaluated offline from daytime sidewalk video in a single university-campus location. Future work includes testing in other locations, times of day, and weather conditions, as well as online closed-loop deployment on the robot.

CHAPTER 1

INTRODUCTION

1.1 Motivation

The motivating platform is a battery-powered scooter moving on campus sidewalks. In the collected video, the path may be concrete or asphalt, bordered by grass, buildings, retaining walls, curbs, or shadows. The scene has no lane markings, curb-paint cues, or high-definition map. The assumed deployment sensor is one forward-facing commodity RGB camera, such as the iPhone and GoPro-style cameras used for data collection, paired with low-power CPU hardware. The practical question is how much sidewalk perception and path planning can be done from that single monocular view.

The same perception problem appears in last-mile delivery robots, assistive mobility scooters, and low-speed campus vehicles [1]. These platforms need local navigation commands from hardware that fits on a small vehicle and runs from a limited battery. A monocular RGB camera is attractive because it is inexpensive, compact, and low power. The trade-off is that the system loses the direct depth measurements available from LiDAR, stereo, or RGB-D sensors.

Several vision-navigation approaches are difficult to use under this constraint. End-to-end policies can map images directly to steering commands, but they are hard to inspect and often assume GPU-class inference [2]. Learned BEV reconstruction methods such as Lift-Splat-Shoot [3] and BEVFormer [4] are strong in structured driving settings, but they generally assume multi-camera inputs and larger compute budgets [5]. Classical homography-based BEV is lighter. In this thesis, however, the slow step is often not projection but path discovery after projection: skeleton extraction, Dijkstra search, and distance-transform ridge tracing can take hundreds of milliseconds per frame. This motivates

using BEV selectively rather than treating it as the default planning domain.

1.2 Problem Statement

In this thesis, I consider a real-time sidewalk-following use case with a single forward-facing camera. In one frame, a drivable path must be chosen from the traversable region and passed to a downstream controller as a set of waypoints. Two research questions emerged during my experimentation. For straight-ahead following, is warping the mask to BEV useful, or is an image-space representation adequate? For situations where a BEV representation is helpful, such as turn geometry and corridor validation, can it be accommodated without compromising the frame rate?

To answer these questions, I designed a narrow experimental evaluation using recorded campus sidewalk video sequences. This design choice isolates the performance of my perception and planning system from closed-loop scooter dynamics, which would introduce additional complexity. The reported runtime is measured on a CPU. I evaluated the system on six video sequences containing more than 22 000 frames in total, and 448 ground-truth masks were hand-annotated. Since I did not test in closed loop, the results in this thesis are not intended as a safety claim. They represent evidence for my claims about the capabilities of the perception and planning system.

1.3 Approach Overview

To achieve sidewalk following, I designed the system in five stages. First, a trained SegFormer-B0 [6] teacher–student network produces a sidewalk mask. Second, the mask can optionally be warped to BEV using a homography matrix. Third, connected-component filtering and basic morphology clean the mask. Fourth, a path planner uses either the BEV sidewalk mask or the original image-space mask to choose a path from the traversable region. Finally, a temporal smoothing stage reduces frame-to-frame planning jitter, as shown in

Figure 3.1.

I iterated on this system four times, while keeping each version backward-compatible with the previous one. In this thesis, I describe the development from iteration to iteration. The iterations trace an arc away from path discovery and toward path verification:

- **Iteration 1:** A BEV skeleton-graph planner with a SegFormer-B2 teacher, discovering paths through BEV pixels.
- **Iteration 2:** A BEV distance-transform planner with enhanced mask morphology, finding ridge paths through the sidewalk area.
- **Iteration 3:** Image-space midpoint and distance-transform planners with a OneFormer Swin-L teacher, removing the dependency on BEV pixels for straight-ahead path following.
- **Iteration 4:** A template-approval planner that uses a BEV corridor to check the validity of GPS-conditioned waypoint turns. In other words, the system checks whether the waypoint-turn template falls within the corridor.

The system described here is closest to a classical corridor-following system and least similar to end-to-end learning. End-to-end systems can produce strong benchmark results, but they are hard to debug in the real world [2, 7]. Dense monocular learned BEV reconstruction can improve scene understanding, but it remains challenging for low-power devices [5]. Classical corridor-following planners can run faster than both of these approaches in structured settings such as agricultural crop rows, but sidewalks have more complex topology, with branches, intersections, and turning commands [8, 9]. My main concern in this thesis is the empirical claim: a fair comparison of BEV-domain planning and image-space planning on monocular sidewalk masks.

1.4 Contributions

This thesis makes the following contributions:

1. A systematic comparison of ten path-planning methods spanning four algorithm families on 448 hand-annotated frames across eight video sequences, providing statistically significant evidence (Friedman $p < 10^{-61}$) that image-space midpoint planning achieves 14.5 px lateral error at 3.0 ms per frame for straight-ahead following. The evaluation further demonstrates that midpoint planning is intent-agnostic—it cannot execute commanded turns—motivating the dual-domain architecture that retains BEV for turn compliance (Chapter 4).

2. A template-approval planning architecture that replaces costly skeleton-graph construction with efficient verification of pre-computed arc templates against distance-transform corridors, achieving 69.5 ms mean pipeline time (Table 4.13) while eliminating slow per-frame graph construction. The BEV domain is retained for corridor extraction and turn validation rather than being discarded (Chapter 3).

3. A semi-supervised segmentation training recipe using a high-capacity OneFormer Swin-L teacher [10] to generate pseudo-labels for a compact SegFormer-B0 student [6], achieving IoU of 0.934 at 14.6 ms on CPU, further improved to IoU of 0.964 through fine-tuning on 228 human-corrected masks. The fine-tuned model generalizes well: held-out video IoU (0.974) exceeds training-video IoU (0.963), confirming no overfitting (Chapter 4).

4. A multi-scale offline evaluation protocol that validates findings at four complementary scales—448 hand-annotated frames across eight video sequences for planner accuracy and segmentation quality, intent-conditioned evaluation on 17 930 annotated video frames for turn compliance, temporal stability analysis measuring heading jitter and path consistency, and an 1800-frame accepted run for sustained system validation. A train/test split confirms generalization: videos used for fine-tuning (VID_017–020) are evaluated separately from held-out videos (IMG_1878, IMG_1921, IMG_1922, IMG_1924) (Chapter 4).

5. A GPS-conditioned waypoint-turn planner with a containment safety guard that veri-

(a) Scooter platform with mounted camera.

(b) Example first-person view from the camera.

Figure 1.1: Hardware platform and camera view. Data were collected with one forward-facing commodity RGB camera per run, including iPhone and GoPro-style camera hardware.

fies committed paths against the BEV corridor before execution, validated with a 0% containment failure rate (Chapter 3, Chapter 4).

1.5 Thesis Organization

Chapter 2 reviews prior work and identifies research gaps. Chapter 3 describes the complete system design. Chapter 4 presents experimental results. Chapter 5 discusses implications, limitations, and threats to validity. Chapter 6 summarizes contributions and outlines future work. Figure 1.1 shows the hardware platform.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter reviews work directly connected to the pipeline: sidewalk perception platforms, BEV projection and local planning, distance-transform and skeleton path extraction, and semi-supervised training for embedded segmentation. The review is scoped to monocular sidewalk following on scooter-scale hardware.

2.1 Sidewalk Perception and Navigation

Sidewalk robots operate under a tighter sensor budget than road vehicles. A car can carry LiDAR, radar, multiple cameras, and GPU hardware; a scooter-scale robot often cannot. Starship Technologies has deployed delivery robots using cameras, ultrasonic sensors, and pre-mapped routes [1]. Viteri et al. [7] use RGB-D sensing for sidewalk following, and Machkour et al. [11] demonstrate monocular navigation in indoor corridors. These systems show that small-platform navigation is feasible, but they do not isolate the coordinate-domain question tested in this thesis: whether a monocular sidewalk mask is better planned in BEV or image space.

Semantic segmentation supplies the traversable-region mask used by the planners. Segmentation architectures range from FCN [12] and DeepLabV3+ [13] to lightweight networks such as BiSeNetV2 [14] and DDRNet [15]. SegFormer [6] combines a hierarchical Vision Transformer encoder [16] with a lightweight MLP decoder, with variants from B0 (3.7M parameters) to B5 (84.7M). The data problem is more limiting than the model choice: CityScapes [17] is vehicle-mounted street data, RUGD [18] is off-road terrain, and Mapillary Vistas [19] differs from a low-mounted scooter viewpoint. This motivates the teacher–student training setup in Section 3.2.1.

2.2 BEV Projection and Path Planning

The utility of BEV representations derives from their expression of local geometry in a top-down frame. Classical Inverse Perspective Mapping (IPM) [20] utilizes a planar homography to project image pixels onto the ground plane [21, 22]. However, this technique relies on the idealization of a planar surface and static camera extrinsics, which is only approximately satisfied on sidewalks. Learned BEV approaches, including Lift-Splat-Shoot [3], BEVFormer [4], and Focus on BEV [5], loosen certain of these assumptions, yet they typically require inputs from multiple cameras or GPU-class compute resources.

The more subtle challenge involves the computational load associated with BEV. Although a monocular camera yields a narrow trapezoidal observation region, many path-finding strategies continue to perform an exhaustive search across the BEV grid at every frame. Skeletonization, Dijkstra search, and distance-transform ridge extraction can collectively dominate the allowable time budget. This thesis decouples BEV discovery from BEV verification, estimating the discovery cost at 540–927 ms per frame and the cost of verifying fixed templates against the corridor at 3–5 ms (Section 4.3).

In terms of local planning methods, approaches encompass graph search [23, 24], potential fields [25], the Dynamic Window Approach (DWA) [26], trajectory optimization [27], and end-to-end learned policies [2, 28]. The task of agricultural row following [8, 9] shares visual similarities because it follows a corridor-shaped region, although sidewalks contain more complex structures such as curved paths, offshoots, curb cuts, and discontinuous edges. A significant body of prior research has conducted planning in BEV or world coordinates from the outset; Chapter 4 examines this assumption through benchmarking ten methods across BEV and image-space domains.

2.3 Distance Transform and Skeleton Methods

The Euclidean Distance Transform (EDT) [29] calculates the distance to the closest boundary for each pixel within an area, yielding a clearance map whose ridge follows the corridor centerline. In 2D binary mask planning, the EDT ridge can be extracted by means of Dijkstra search on a cost field $c(x,y) = 1/(d(x,y)+\varepsilon)^{\alpha}$, thus steering a robot toward the corridor's widest central regions. This thesis explores a distinct application: utilizing the distance-transform corridor as a verification layer against which to evaluate pre-computed template arcs, transforming the role of the EDT from a discovery engine to a verification layer (Chapter 3).

Morphological skeletonization [30, 31] condenses a binary mask into its medial axis to enable graph-based path enumeration. Nonetheless, this technique is sensitive to noise; minute mask protrusions result in spurious branches requiring additional heuristic pruning. The complete pipeline—thinning, pruning, graph construction, and Dijkstra search—is quite costly. None of the existing literature reviewed here compares skeleton-graph construction and template verification on sidewalk image data; the present thesis provides this comparison (Chapter 4).

2.4 Semi-Supervised Learning and Embedded Deployment

Knowledge distillation [32] facilitates the transfer of capabilities from a large teacher model to a smaller student model by utilizing the teacher's soft probabilities to train the student, rather than relying only on hard labels. Some semi-supervised variants employ the teacher model to generate pseudo-labels for unlabeled data [33]. OneFormer [10] consolidates semantic, instance, and panoptic segmentation through task-specific joint training; trained on ADE20K [34], COCO [35], and CityScapes [17], it can produce high-quality pseudo-labels across diverse visual classes using a Swin-L backbone [36]. I do not know of

prior work using OneFormer Swin-L as the teacher and SegFormer-B0 as the student for binary sidewalk segmentation; this thesis implements this configuration and achieves an IoU of 0.964 at 14.0 ms on CPU after fine-tuning on 228 manually corrected mask images (Chapter 4).

Deployment on embedded platforms such as the Raspberry Pi 4 and Rock 5B demands deliberate consideration of model selection and pipeline design. A 10 Hz (100 ms) processing loop is adequate for vehicles operating at 1–2 m/s [37, 38]. Although frameworks such as ONNX Runtime [39] and techniques such as model quantization may further reduce latency, few sidewalk applications have evaluated a full perception-to-planning pipeline on CPU. This thesis presents such an end-to-end CPU-based system evaluation, with an image-space pipeline capable of delivering 50 FPS on CPU (Chapter 4).

2.5 Summary and Research Gap

This thesis identifies four research gaps:

1. **The absence of a systematic comparison between BEV and image space for sidewalk navigation.** This work addresses this through a ten-method, two-domain benchmarking effort (Chapter 4).
2. **A lack of cost-benefit analysis of BEV discovery versus BEV verification.** The cost-benefit study in Chapter 4 identifies a difference of several orders of magnitude in cost between BEV discovery and BEV verification, and shows that the latter still holds value for metric-scale turn-radius geometry.
3. **A failure to apply the OneFormer-to-SegFormer distillation recipe to sidewalk segmentation.** This thesis provides such a recipe, achieving 0.964 IoU at 14.0 ms on CPU.
4. **A paucity of end-to-end CPU pipeline benchmarks for sidewalk applications.** This thesis presents system-level profiling and benchmarking showing that the recom-

mended image-space pipeline, from segmentation to planning, achieves 50 FPS on CPU.

These research gaps motivated the system design in Chapter 3.

CHAPTER 3

SYSTEM DESIGN

The system changed through four design iterations. Version 1 used BEV skeleton-graph planning and exposed topology, but it was unstable when masks changed. Version 2 used a BEV distance-transform ridge and reduced some skeleton artifacts, but the graph search remained too slow. Version 3 moved straight-ahead following into image space. Version 4 kept image-space following and returned to BEV only for corridor verification and commanded turns. The final system therefore uses image space for fast straight following and BEV for metric turn geometry.

3.1 Hardware Platform

The hardware constraint is the starting point for the design. Each run uses one forward-facing monocular RGB camera and CPU-only processing. Without depth, stereo, LiDAR, or IMU input in the evaluated pipeline, three-dimensional reasoning comes from calibrated image geometry, specifically a planar homography.

The experimental platform is an electric scooter with a commodity RGB camera mounted to the handlebar stem at approximately 0.8 m above ground level and angled about 10° downward (Figure 1.1). The dataset includes video captured with an iPhone camera and GoPro-style action-camera hardware; each sequence is treated as a monocular forward-view stream. The target deployment class is a Raspberry Pi 4 (Cortex-A72, 8 GB RAM) or Rock 5B single-board computer. Development and evaluation were performed on a workstation (Intel Core i7-12700K, 32 GB RAM, NVIDIA RTX 3060), but reported runtimes use CPU-only inference with the GPU disabled. Raspberry Pi latencies are expected to be 2–3$\times$ higher because of lower clock speed and memory bandwidth.

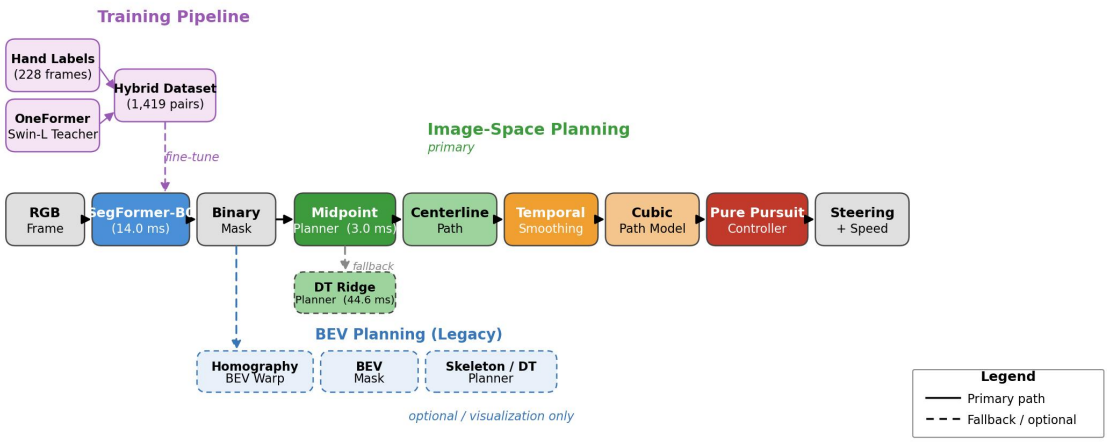


Figure 3.1: Sidewalk navigation pipeline.

These constraints set the segmentation model size and the planner runtime budget.

3.2 Segmentation Module and Supervision Strategy

The 10 Hz inference target constrains how large the segmentation model can be. In the CPU testing environment, SegFormer-B0 [6] has 3.7M parameters and executes in less than 50 ms, allowing time for mask post-processing and path planning. Consequently, larger configurations such as B2 and B5 cannot be supported without sacrificing the overall pipeline frame rate.

3.2.1 Teacher–Student Supervision

Since the annotated sidewalk dataset is relatively small, this thesis adopts a teacher–student supervision strategy. Two teacher models were tested: a SegFormer-B2 model trained on a small set of manually annotated frames, and OneFormer Swin-L [10] trained on ADE20K. The OneFormer teacher generates pseudo-labels for 1,419 unlabeled frames drawn from 10 different campus video clips. Connected-component filtering removes noise from the teacher predictions. A confidence threshold of $\tau = 0.60$ is then applied, and manual ground truth is prioritized over teacher pseudo-labels wherever both are available.

3.2.2 Limitations of Teacher Pseudo-Labels

The teacher's pseudo-labels consistently exhibit systematic errors, primarily because its pre-training distribution assumes a different camera viewpoint than the scooter-mounted sensor. The teacher model was largely trained on vehicle-borne imagery captured from approximately 1.5–2.0 m above the ground, whereas the scooter camera operates from a height of roughly 0.8 m. While evaluating errors during the annotation of 228 frames from VID_017 through VID_020, three prominent issue classes were identified:

- **Unreachable surface segmentation:** The teacher detects any surface resembling a drivable road, such as public roads located behind barriers or parallel pathways situated at different elevations, even when those surfaces cannot be physically traversed by the scooter.
- **Shadow and sunlight failures:** Under strong shadows, the teacher either fails to detect sidewalk or incorrectly classifies grass. Direct sunlight also causes the model to confuse overexposed sidewalk regions with adjacent surfaces.
- **Seasonal appearance variation:** Brown, dormant grass looks similar to concrete, and the teacher has trouble distinguishing these textures at their shared boundaries.

3.2.3 Human-Corrected Fine-Tuning

In the fine-tuning stage, a stricter definition is enforced: the model predicts only sidewalk traversable by the ego vehicle. Starting from the teacher model checkpoint, training runs for 8 epochs using a learning rate of 2×10^{-5} with cosine annealing and a batch size of 4. The model is trained using a combined cross-entropy and Dice loss function, and IoU increases from 0.934 to 0.964 while precision increases from 0.951 to 0.981 (Table 4.2). The most notable improvement is the reduction of over-segmentation on unreachable pavement. The loss function is defined as $\mathscr{L} = \mathscr{L}_{\text{CE}} + \mathscr{L}_{\text{Dice}}$, with added weighting for sidewalk pixels.

Table 3.1 summarizes the complete training stages.

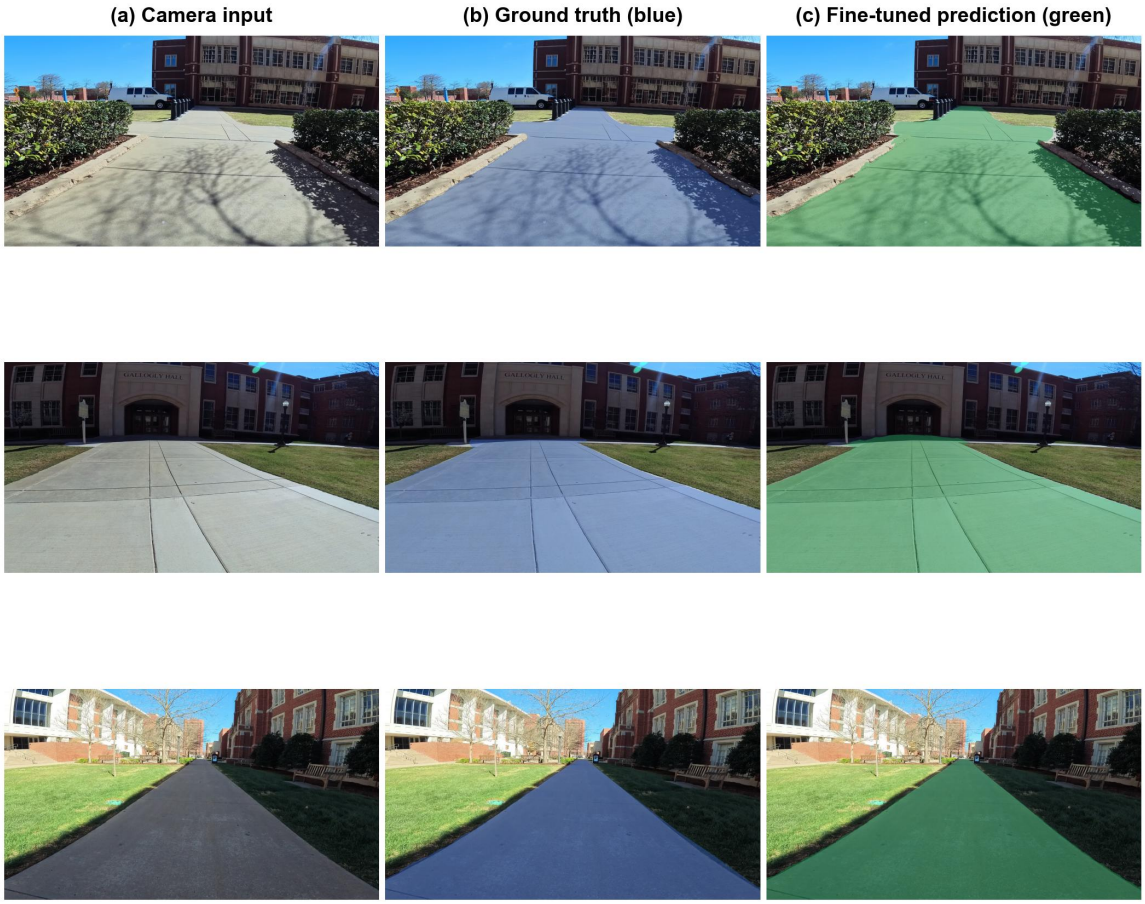


Figure 3.2: Segmentation examples from training videos.

The relationship between image resolution and runtime performance is detailed in the next section.

3.3 Resolution Trade-Off Analysis

The selected input resolution was determined through empirical testing. Lower resolutions failed to preserve narrow sidewalk boundaries, while higher resolutions exceeded the available computational budget.

SegFormer-B0 was benchmarked on CPU at the resolutions detailed in Table 3.2. Configurations at 320×180 resolution and below were discarded because they lacked sufficient boundary detail. Configurations at 960×540 and higher were rejected for exceeding the

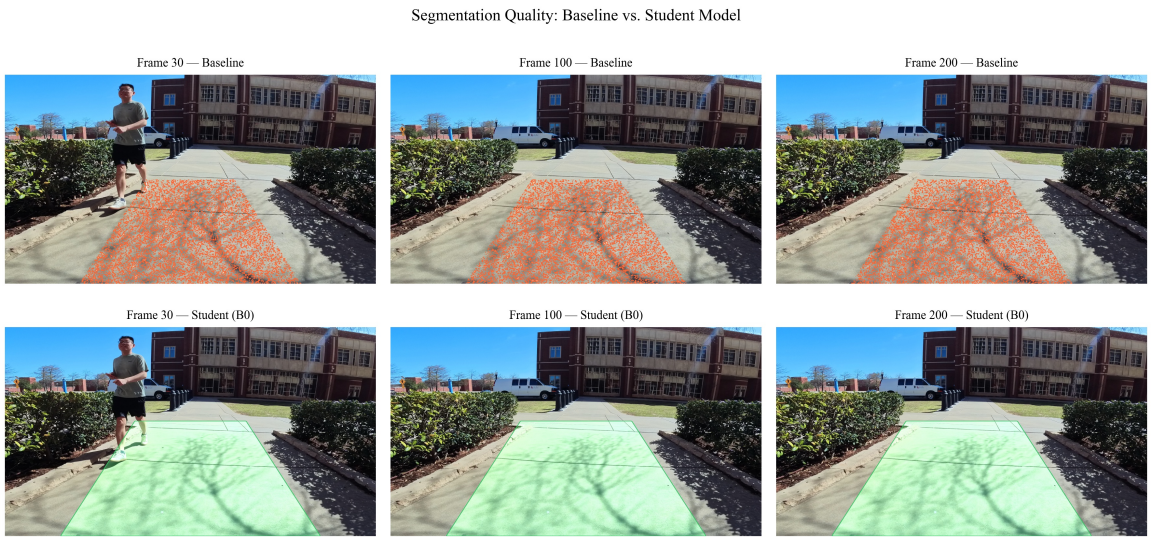


Figure 3.3: Baseline and OneFormer-trained segmentation outputs.

Table 3.1: Segmentation training progression. SegFormer-B0 is unchanged; the changes are the teacher, loss, and added frames. Val IoU uses the validation split, while Table 4.2 reports the separate 228-frame hand-labeled evaluation.

Stage	Teacher	Training Frames	Val IoU	Latency (ms)
1	SegFormer-B2 (300 hand-labeled)	699	0.880	18.9
2	OneFormer Swin-L (ADE20K)	699	0.944	11.7
3	OneFormer Swin-L + boundary loss	699	0.948	11.7
4	OneFormer Swin-L + expanded data	1,419	**0.959**	**11.7**

50 ms latency constraint. The selected input is 640×360 because it retains usable boundary detail while keeping latency under 50 ms.

The next step is the BEV projection performed by the turn and verification modules.

3.4 Bird's-Eye View Projection and Mask Refinement

Metric verification remains necessary for turn decisions. Using a four-point planar homography ($\mathbf{p}' \sim H\mathbf{p}$, where H is derived from four known ground-plane correspondences), the mask is projected onto a 6m × 11m grid in bird's-eye view. This target grid resolution is 360×660 pixels (60 px/m). The warp process takes approximately 0.9 ms on CPU. Given the assumptions of planar ground and a fixed camera pose, this projection is applied only

Table 3.2: SegFormer-B0 CPU latency by input resolution.

Resolution	Latency (ms)	FPS
160×90	19.84	50.40
320×180	25.35	39.45
480×270	42.05	23.78
640×360	**46.00**	**21.74**
768×432	93.55	10.69
960×540	152.45	6.56
1280×720	352.40	2.84

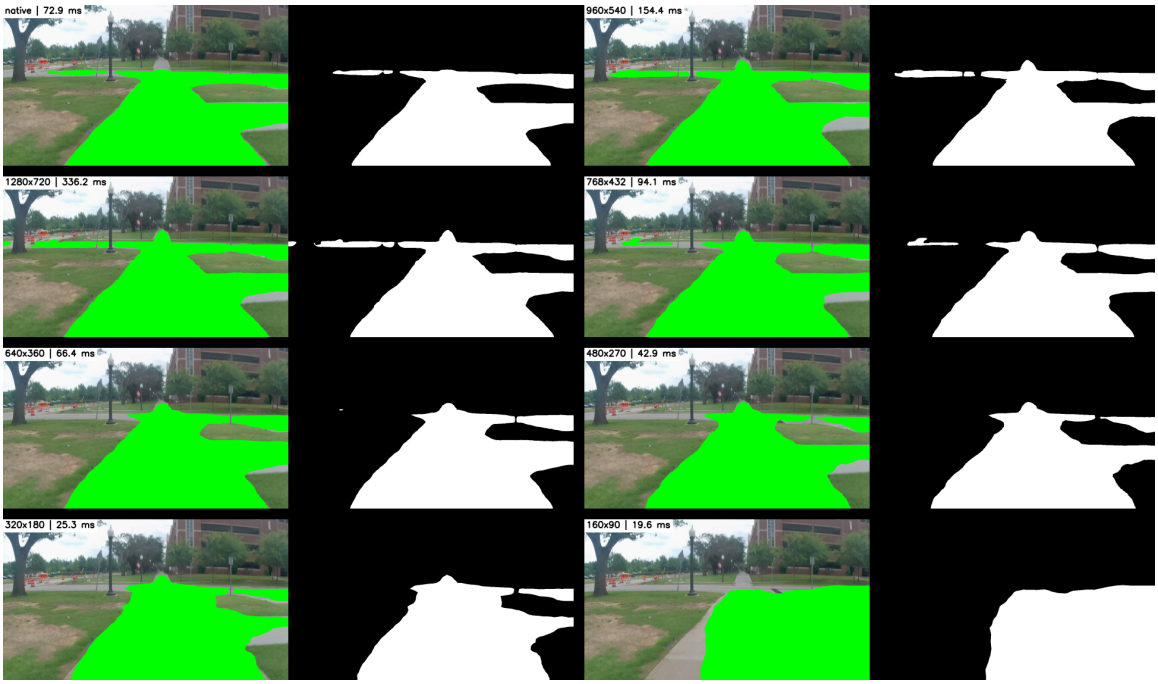


Figure 3.4: SegFormer-B0 resolution sweep.

where precise metric geometry is required: corridor verification and turn-geometry planning.

Noise in the far distance increases after the perspective warp. To address this, the pipeline retains only the ego-connected component after masking, optionally applying morphological closing and opening and filling holes. Boundary smoothing and distance-transform-based component selection are also optional. The subsequent planners consume either this BEV mask or the corrected 2D image mask as input.

3.5 Path Planning Methods

Ten path planning methods spanning two geometric domains and four algorithm families are compared. The central lesson of the design progression is that the planning *domain*—BEV versus image-space—matters as much as the planning *algorithm*. Table 3.3 provides a compact overview; the subsections below describe each method.

Table 3.3: Ten path planning methods.

Method	Domain	Family	Strategy	Complexity
Midpoint	Image	Geometric	Per-row boundary averaging	$O(H \cdot W)$
Weighted Centroid	Image	Geometric	Area-weighted center of mass	$O(H \cdot W)$
Image DT	Image	Cost-field	EDT ridge + drift constraint	$O(H \cdot W)$
Image DT-Ridge	Image	Cost-field	EDT ridge tracing (image)	$O(H \cdot W)$
BEV DT-Ridge	BEV	Cost-field	EDT ridge tracing (BEV)	$O(N \log N)$
Skeleton-Graph	BEV	Topological	Graph search (Dijkstra)	$O(N \log N)$
Skeleton Hybrid	Image	Topological	Coarse skeleton + DT refine	$O(H \cdot W)$
Template-Approval	BEV	Template	Propose-and-verify (7 arcs)	$O(K \cdot L)$
Dynamic Window	Image	Sampling	Velocity-space sampling	$O(K \cdot T)$
Potential Field	Image	Sampling	Attractive/repulsive forces	$O(H \cdot W)$

3.5.1 BEV Skeleton-Graph Planner

The Guo–Hall parallel thinning algorithm [31] reduces the BEV mask to a 1-pixel-wide skeleton. An undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is constructed from skeleton pixels with 8-neighbor edges, and Dijkstra's algorithm [40] enumerates candidate paths scored by curvature and lateral shift. This method exposes junction topology but proved unstable in practice: minor mask perturbations create or destroy branches, causing path jumps between frames. At 734 ms per frame (Table 4.6), it consumes the entire real-time budget.

3.5.2 BEV Distance-Transform Ridge Planner

This method computes the EDT of the BEV mask and traces the ridge of maximum clearance via Dijkstra on a cost field $c(x,y) = 1/(d(x,y) + \varepsilon)^{\alpha}$ ($\varepsilon = 0.5$, $\alpha = 1.5$), smoothed with

a Savitzky–Golay filter [41]. The ridge tracks the corridor center more consistently than skeletonization but inherits the full cost of BEV-domain graph search.

3.5.3 BEV Template-Approval Planner

A bank of seven pre-computed arc templates (straight plus gentle, medium, and sharp arcs in each direction) is scored against the distance-transform corridor [42]. Each template is evaluated by containment within the corridor, and the highest-scoring template passing a confidence gate becomes the planned path. Family-reuse hysteresis suppresses frame-to-frame switching. This approach eliminates noise-sensitive graph construction, reducing mean heading error by 40.6% and halving path-source switches compared to the skeleton-graph baseline (Table 4.9). This is the planner used in the 1,800-frame accepted run.

3.5.4 Image-Space Midpoint Planner

For each row y of the image-plane mask, the left and right sidewalk boundaries are identified and the midpoint $x_{\text{mid}}(y) = (x_L(y) + x_R(y))/2$ is computed. A connected-component filter ensures only the ego-connected region is used, and per-row midpoints are smoothed with a Savitzky–Golay filter. This method achieves 14.5 px lateral center error at 3.0 ms—the lowest error and fastest runtime among all ten methods—a 186$\times$ speedup over BEV skeleton-graph planning. The central finding: for straight-ahead path extraction, the segmentation mask already encodes sufficient geometric information without BEV transformation.

3.5.5 Remaining Planners

Image-Space DT. The EDT is computed on the image-plane mask, and the maximum-clearance ridge is traced with a per-row lateral drift constraint. More robust than midpoint on irregular masks but slower (129.9 ms).

Image-Space DT-Ridge. Traces the global EDT ridge without drift constraint, allowing unconstrained ridge following (99.4 px error, 44.6 ms).

Weighted Centroid. Area-weighted center of mass per row, biased toward the corridor interior (51.1 px error, 2.4 ms—second-best accuracy).

Dynamic Window Approach. Samples admissible velocity pairs [26] and scores trajectories by heading alignment, clearance, and progress (77.0 px error, 3.6 ms). Exhibits the lowest heading jitter (2.1 deg/frame).

Potential Field. Attractive/repulsive forces [25] from goal and boundaries (154.4 px error, 2.8 ms). Suffers local-minimum sensitivity; paths on only 92% of frames.

Skeleton Hybrid. Coarse image-space medial-axis extraction refined by local EDT ridge snapping (82.1 px error, 8.1 ms—much faster than BEV skeleton-graph).

3.5.6 GPS-Conditioned Waypoint-Turn Planner

Turn maneuvers require metric-scale arc geometry that only BEV coordinates can provide. When a GPS waypoint triggers a junction maneuver, the planner operates in five stages: (1) scanning a forward decision band (2 m to 7 m) for asymmetric road openings on the commanded side; (2) clustering supported rows and selecting the candidate turn apex; (3) target gating with acquisition/sustain hysteresis (thresholds 0.40/0.25); (4) fitting a constant-curvature circular arc from ego through the apex, parameterized by arc angle a and turn-side sign s as $x = R\sin(a)$ and $y = sR(1-\cos(a))$; and (5) containment gating requiring $\geq$60% of path points inside the corridor and $\geq$70% within the near-field (1.2 m).

3.5.7 Turn Containment Safety Guard

Every committed control path is evaluated against the BEV drivable mask before execution, measuring path-outside ratio, path-outside count, and minimum boundary clearance. The first 4 samples near the ego are excluded due to BEV bottom-edge noise. Paths failing containment trigger hold behavior with speed reduction, ensuring unsafe paths never reach the controller. Overnight validation achieved a 0% containment failure rate.

3.6 Temporal Smoothing

Frame-by-frame inference produces unavoidable jitter. Three EMA filters are applied: (1) **path smoothing** on cubic polynomial coefficients with adaptive α (0.35–0.85) and topology-change reset; (2) **heading smoothing** with circular EMA handling the $\pm 180^\circ$ wraparound, resetting on jumps exceeding 45°; and (3) **mask smoothing** with a conservative-blend policy ($\alpha = 0.65$, consistency threshold $c_{\text{thresh}} = 0.20$) that reduces blending when consecutive-frame IoU drops sharply.

3.7 Obstacle Detection, GPS Navigation, and Control

Obstacle detection. Obstacle detection uses YOLOv8-nano [43, 44] with a confidence threshold of $\tau_{\text{det}} = 0.35$. Distance is estimated from the bounding-box foot position using a pinhole approximation:

$$d \approx \frac{h_c}{\tan\left(\phi_v \left(\frac{y_{\text{foot}}}{H} - 0.5\right)\right)},$$

where $h_c = 0.8\,\text{m}$. Obstacles within $1.0\,\text{m}$ trigger a full stop; obstacles within $3.0\,\text{m}$ reduce the speed to $0.3\,\text{m/s}$.

GPS navigation. GPS serves as an intent-conditioning signal, not a localization source. A low-cost GNSS module (u-blox NEO-series, 1 Hz, 2.5 m CEP) provides waypoint intent:

bearing difference below 12° yields straight intent, 12–40° yields left/right, above 40° yields sharp turn. The heading command blends vision ($w_v = 0.7$) and GPS ($w_g = 0.3$); when no fix is available, w_g falls to zero.

Steering and speed. A pure-pursuit-inspired control law [45] converts the heading angle to discrete commands: straight ($|\theta| < 12°$, $v = 1.5$ m/s), left/right (12–40°, linearly interpolated), or sharp turn ($> 40°$, $v = 0.4$ m/s). Commands are transmitted via USB-serial with a 500 ms hardware watchdog.

3.8 Safety Mechanisms

The default response to uncertainty is to slow or stop [46]. Non-negotiable gates include: full stop with no valid path for N consecutive frames; immediate stop within 1.0 m of obstacles; GPS loss fallback to vision-only control; manual override via hardware button; and speed cap (0.2 m/s) when segmentation IoU drops below threshold for multiple consecutive frames.

3.9 Evaluation Metrics

The pipeline is evaluated offline using five metrics: (1) **Segmentation IoU** between predicted and ground-truth masks; (2) **Lateral center error** (LCE $= \frac{1}{N}\sum |x_{p_i} - c(y_i)|$, in pixels); (3) **Inside-GT ratio**—fraction of path points within the ground-truth mask; (4) **Temporal stability**—frame-to-frame mask IoU, with frames below 0.80 flagged as unstable; and (5) **Runtime** in milliseconds on CPU.

3.10 Software Architecture

The system is implemented in Python 3.10+ using OpenCV, PyTorch, NumPy, and SciPy. A centralized configuration module (`config.py`) serves as the single source of truth. Each

research improvement is gated by a boolean flag, enabling exact reproduction of any prior iteration. Key reproducibility artifacts include the SegFormer-B0 checkpoints, BEV calibration matrix, 448 hand-annotated masks, and the centralized configuration.

CHAPTER 4

EXPERIMENTAL EVALUATION

This chapter presents experimental results organized around three themes: (1) segmentation quality and generalization, (2) the ten-method planner comparison with BEV cost–benefit analysis, and (3) system-level validation including the template planner, turn planner, and sustained operation. The experimental setup is described first.

4.1 Experimental Setup

All experiments use campus video collected at the University of Oklahoma during daytime hours (9 AM–4 PM) under clear to partly cloudy conditions. Eleven video sequences were recorded; six totaling 22 679 frames are used for full-video evaluation. A total of 448 hand-annotated frames across eight video sequences (VID_017–020, IMG_1878, IMG_1921, IMG_1922, IMG_1924) serves as the primary benchmark. Of these, 228 (VID_017–020) were used for fine-tuning and 220 (IMG_1878, IMG_1921, IMG_1922, IMG_1924) are held out. All annotations were produced by a single annotator (noted as a threat to validity in Section 5.4). Runtime is mean per-frame wall time on CPU (Intel Core i7-12700K, GPU disabled). Table 4.1 summarizes the dataset.

For the planner comparison, each method receives the same cleaned binary mask; BEV methods receive the homography-warped version. BEV planners run in stateless mode (fresh instance per frame). The fine-tuned model is trained on 228 frames from VID_017–020 with a 20% validation hold-out; the reported IoU (0.964) is the validation-set metric.

Table 4.1: Video dataset summary.

Video ID	Approx. Frames	Conditions
IMG_1876	∼2,000	Straight, mild shadows
IMG_1877	∼2,500	Curves, tree canopy
IMG_1878	∼3,000	T-junctions, surface changes
IMG_1921	∼2,500	Wide path, open area
IMG_1922	∼4,000	Mixed conditions
IMG_1924	∼5,000	Diverse, pedestrian traffic
IMG_1931	∼3,500	Narrow paths, tight turns
VID_017–020	∼4,000 each	New campus routes, March 2026

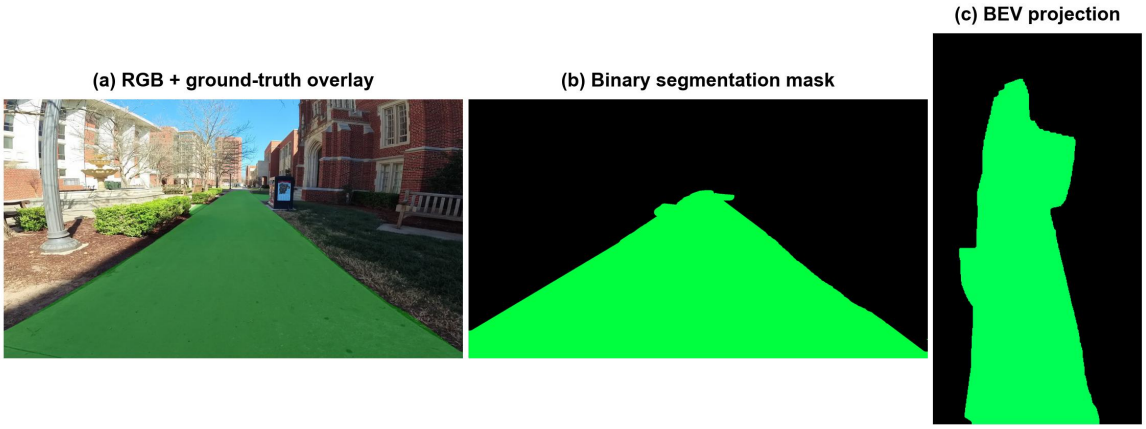


Figure 4.1: Segmentation pipeline stages.

4.2 Segmentation Quality and Generalization

Figure 4.1 illustrates the segmentation pipeline stages from RGB input through BEV projection. Table 4.2 compares the three-stage segmentation progression on 228 hand-annotated frames. Switching from a SegFormer-B2 teacher to OneFormer Swin-L raised IoU from 0.926 to 0.934; fine-tuning on human-corrected masks raised IoU to 0.964, driven by precision gains (0.951 $\rightarrow$ 0.981) that confirm suppression of background over-segmentation. The student architecture (SegFormer-B0, 3.7M parameters) remained unchanged throughout all three stages; only the teacher, training data, and recipe varied. Per-video results (Table 4.3) confirm consistency across sequences.

A full-video replay across six sequences (22 679 frames) confirms generalization: the

Table 4.2: Segmentation quality on 228 hand-annotated frames.

Model	IoU	Prec.	Recall	F1	ms
Baseline (B2 teacher)	0.926	0.946	0.978	0.960	15.6
Teacher-trained (OneFormer)	0.934	0.951	0.982	0.965	14.6
Fine-tuned (hand-annotated)	**0.964**	**0.981**	**0.982**	**0.982**	**14.0**

Table 4.3: Per-video segmentation IoU for the fine-tuned model (228 frames).

Video	IoU	Precision	Recall
VID_017	0.971	0.982	0.989
VID_018	0.960	0.978	0.982
VID_019	0.962	0.988	0.974
VID_020	0.963	0.978	0.984
All	**0.964**	**0.981**	**0.982**

candidate model reduces temporal instability by 77% (1.46% $\rightarrow$ 0.33%) and increases template success rate by 5.6 percentage points (Table 4.4, Figure 4.4).

Table 4.4: Full-video replay metrics.

Metric	Baseline	Candidate	Δ
Mean Seg IoU	0.909	0.925	+0.016
Unstable Rate [%]	1.46	0.33	-1.13 pp
Template Success [%]	73.7	79.3	+5.6 pp
Fallback Rate [%]	19.0	14.3	-4.7 pp

To verify that fine-tuning is not memorization, both models were evaluated on sequences *never used* during fine-tuning (IMG_1878, IMG_1921, IMG_1922). The fine-tuned model produces tighter masks with less background over-segmentation (Figure 4.5), and held-out IoU (0.974) exceeds training-set IoU (0.963), confirming generalization.

Figure 4.6 illustrates the complete perception-to-planning pipeline on three diverse campus scenes. Each row shows (a) the raw camera input, (b) the fine-tuned SegFormer prediction overlaid on the frame, and (c) planned navigation paths computed from the predicted mask, with a BEV inset showing the projected corridor.

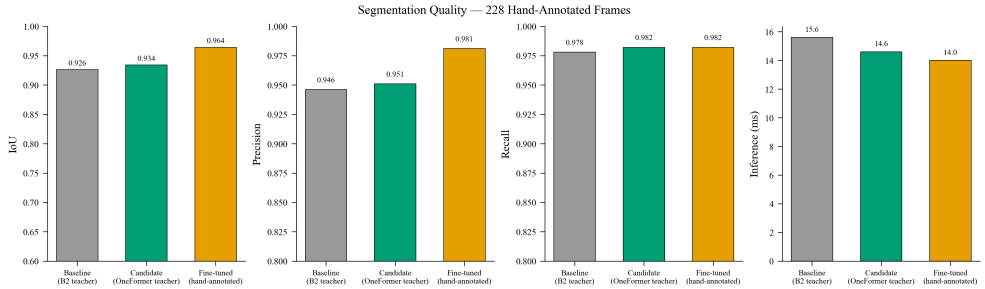


Figure 4.2: Segmentation quality comparison.

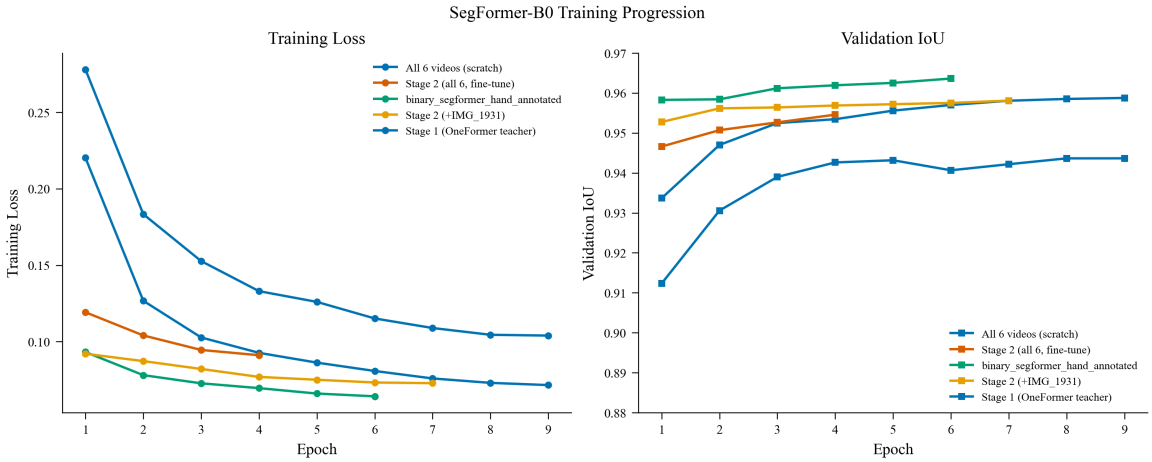


Figure 4.3: SegFormer-B0 training curves.

4.3 Ten-Method Planner Comparison and BEV Cost–Benefit

Table 4.5 illustrates how the four design iterations produced the ten planners being evaluated. These planners are assessed on 448 frames manually annotated with masks representing the oracle. The ranking is statistically significant (Friedman test, $\chi^2 = 311.1$, $p = 1.17 \times 10^{-61}$, $N = 71$ common frames). Of the 45 pairwise Wilcoxon signed-rank tests with Bonferroni correction, 32 remain significant, and the comparisons with midpoint are always significant. The common-frame count indicates how many frames each planner found a path in. Since the BEV DT-ridge planner could produce paths on only 16% of frames, it has a lower common-frame count than the other planners. When that planner is removed, the Friedman statistic for the remaining nine-method comparison is $\chi^2 = 1513.7$, $p < 10^{-318}$, $N = 410$.

Accuracy and runtime are listed in Table 4.6, while Table 4.7 shows how changing the

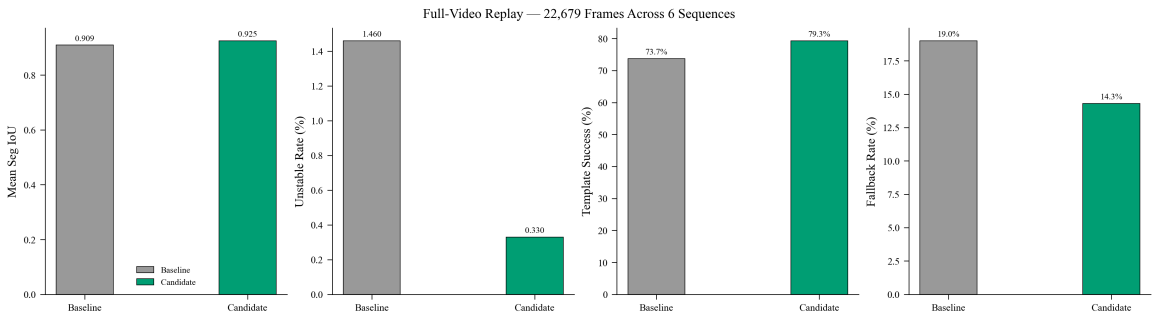


Figure 4.4: Full-video replay comparison.

Table 4.5: Design iteration progression.

Iter.	Planner	Domain	Key Change	FPS
v1	Skeleton-Graph	BEV	Guo–Hall + Dijkstra	9.1
v2	DT Ridge	BEV	EDT ridge replaces skeleton	$\sim$1
v3	Midpoint / DT	Image-space	BEV bypass	59.2
v4	Template Arc	BEV corridor	Propose-and-verify	9.97

segmentation model changes midpoint planning accuracy.

The accuracy/speed tradeoff is visualized in Figure 4.8. Midpoint is in the Pareto-optimal corner, with the best accuracy and fastest runtime. Four planners fit the real-time budget of 100 ms: midpoint, weighted centroid, dynamic window, and potential field. Midpoint is superior on accuracy within this set, while weighted centroid has the next-best accuracy at 51.1 px error.

Image-space midpoint achieves the lowest lateral center error, 14.5 px (95% CI [13.5, 15.4]), in the fastest runtime, 3.0 ms. When the segmentation model is fine-tuned, midpoint error is reduced from 35.6 px to 18.3 px (Table 4.7), an improvement of 49% toward the oracle bound. As discussed earlier, the midpoint planner is intent-agnostic, so it cannot perform the turns it is commanded to make; this is why the system adopts a dual-domain architecture. Of the other planners, weighted centroid (51.1 px error, 2.4 ms runtime) and DWA (77.0 px error, 3.6 ms runtime) are attractive options within the real-time constraints. Figure 4.9 summarizes these results, and Figure 4.10 shows the BEV skeleton-graph pipeline.

Table 4.6: Ten-planner oracle-mask comparison.

Family	Planner	Path [%]	Inside-GT	Err. [px]	ms
Geometric (boundary-based)					
	Midpoint	100	0.99	**14.5**	**3.0**
	Weighted Centroid	100	0.99	51.1	2.4
Cost-field (distance-transform)					
	Image DT	100	1.00	87.0	129.9
	Image DT-Ridge	100	1.00	99.4	44.6
	BEV DT-Ridge	16	0.80	48.0	174.2
Topological (skeleton/graph)					
	BEV Skeleton-Graph	100	0.96	84.1	734
	Skeleton Hybrid	100	0.99	82.1	8.1
Template and sampling					
	BEV Template-Approval	100	0.98	84.8	259
	Dynamic Window	100	0.99	77.0	3.6
	Potential Field	92	0.91	154.4	2.8

Table 4.7: Segmentation model effect on midpoint planning.

Segmentation Model	Inside-GT	Center Err. [px]	ms
Baseline (old model)	0.987	39.4	3.0
Teacher-trained	0.968	35.6	3.0
Fine-tuned	**0.985**	**18.3**	3.0
Oracle (ground truth)	0.994	14.5	3.0

Segmentation Error Cascade

If the planners are given predicted masks rather than oracle masks, the midpoint planner degrades by only 3.8 px (14.5 px $\rightarrow$ 18.3 px), suggesting a robust pipeline. The held-out segmentation IoU is 0.974 for the evaluations in this chapter, which is higher than the 0.963 measured on the training set. The midpoint lateral error is also lower on the held-out video set, at 14.0 px, than on the training videos, at 22.4 px. This is attributed to differences in video geometry rather than overfitting to the training set. Figure 4.11 summarizes the error cascade and evidence for generalization, and Figure 4.12 visualizes this effect on three

frames.

Intent-Conditioned Evaluation

There are 17 930 frames with intent labels. While midpoint remains reasonably accurate in terms of position (23–28 px center error), it succeeds on only a fraction of commanded turns (33–46%; Figure 4.14). This is exactly what one would expect, since it does not know where the user wishes to go and simply follows the center of the visible corridor. The waypoint-turn planner in Section 4.5.2 addresses this problem.

Temporal Stability

The DWA exhibits the least heading jitter, only 2.1 deg/frame. Midpoint exhibits the second lowest, at 4.5 deg/frame on average ($p_{95} = 18.5$ deg), which is why the system adds a temporal smoother. Right turns destabilize every planner, relative to straight motion, by 2–3×.

4.3.1 BEV Cost–Benefit Analysis

The BEV masks achieve approximately 33% fill, enough to produce a viable corridor and turn shape to evaluate. The computational bottleneck is path finding; it requires 540–927 ms per frame. The path verification step requires only 3–5 ms (Table 4.8). The BEV skeleton-graph planner still achieves an error of 82.1 px with oracle masks, compared to 14.5 px for the image-space midpoint planner, implying that the coordinate transform itself is error-prone.

Table 4.8: BEV discovery and verification cost.

Iteration	BEV Role	Planner Cost	BEV Value
v1 (Skeleton-graph)	Path discovery	734 ms	Metric paths (slow)
v2 (DT-ridge corridor)	Path discovery	926.8 ms	Metric paths (slower)
v3 (Image-space)	None	2.2 ms	—
v4 (Template-approval)	Verification	3 −−5 ms	Corridor + turn geometry

The dual-domain architecture addresses this problem. Most planning is performed in image space for straight-ahead driving. BEV is used when the system needs to verify corridor occupancy or turn geometry, since these tasks require metric-scale reasoning.

4.4 Template-Approval vs. Skeleton Discovery

To isolate the effect of the two proposed planners, the template planner is compared with the skeleton-graph planner on a short, calibrated clip of 220 frames (VID_017 with detections disabled). As Table 4.9 shows, the template planner reduces mean heading error by 40.6% and halves the number of path-source switches. The fixed arc bank, combined with family-reuse hysteresis, produces a smoother control output than searching for a new graph path at every frame. The lower mean speed of 1.067 vs. 1.500 m/s in this test corresponds to slowing in the middle of a turn to account for lower confidence compared with straight-line driving. It is a deliberate tradeoff.

Table 4.9: Template-approval versus skeleton-graph baseline.

Metric	Graph Baseline	Template-Approval
Mean $\lvert\theta\rvert$ [deg]	1.190	**0.707**
p_{95} $\lvert\theta\rvert$ [deg]	3.724	**3.508**
Path-source switches	36	**18**
Template rate [%]	—	62.3
Graph rate [%]	75.0	5.9
Fallback rate [%]	25.0	31.8
Mean speed [m/s]	1.500	1.067

4.5 System Validation

4.5.1 Runtime Analysis

Table 4.10 compares per-module runtimes. The planner dominates the difference: image-space midpoint costs 3.0 ms vs. 734 ms for BEV skeleton-graph, a 186× speedup.

Table 4.10: Per-module runtime comparison.

Module	BEV Pipeline	Image-Space	Note
SegFormer Inference	14.0 ms	14.0 ms	Shared (fine-tuned)
Mask Refinement	8.5 ms	3.0 ms	Img skips BEV cleanup
BEV Projection	0.9 ms	—	
BEV Cleanup	14.6 ms	—	
Planner	734 ms	3.0 ms	186× speedup
Total	726 ms	20.0 ms	
FPS	1.4	**50.0**	

Table 4.11 shows profiled runtimes under different system configurations, illustrating the impact of detection mode and predictive frame reuse on throughput.

Table 4.11: System-level runtime.

Configuration	FPS	Seg (ms)	Det (ms)	BEV (ms)
No detection, predictor on	25.3	20.3	—	11.0
GPU detection, predictor on	20.1	17.0	12.2	13.4
CPU detection, predictor on	12.2	18.8	39.0	14.6
No detection, predictor off	8.9	75.3	—	29.5

4.5.2 Waypoint-Turn Planner Validation

The turn planner is tested in a replay of 300 frames from VID_017, where the scooter's right-turn intent was commanded for frames 20–30. Instead of relying on GPS bearing comparisons to infer intent as in the deployed system, I explicitly set intent in this evaluation for the video frames at which actual turns took place when collecting data. This allows for testing of the end-to-end pipeline (corridor scan, arc fitting, containment gating, and template-to-turn-to-template transitions) on video data, and in real turn geometry. Table 4.12 indicates a failure rate of 0% for turn containment. The planner turns for frames 24–25 (support for the corridor is higher than the 0.40 acquisition gate) and maintains hold behavior on surrounding frames; otherwise, the scooter follows a template. Final validation with real-time turns that are triggered through GPS is still left to future work.

Table 4.12: Waypoint-turn planner evaluation.

Metric	Value
Total frames	300
Template rate [%]	96.3 (289/300)
Turn containment failure rate	0.0%
Min boundary clearance [px]	4.775
Mean FPS	12.86

4.5.3 1800-Frame Accepted Run

Table 4.13 shows evaluation results for the entire VID_017 video (stride 4, detection off). The template planner has an evaluation success rate of 100% over 1800 frames, which is a much better result compared to the 62.3% template rate in the 220-frame evaluation (Table 4.9). It is believed to be caused by a better segmentation model (IoU 0.954) and temporal downsampling by a factor of 4. The mean FPS of 9.97 meets the real-time constraint of 10 Hz, indicating that the system can be deployed on CPU hardware.

Table 4.13: Accepted run on VID_017.

Metric	Value
Frames processed	1800
Mean loop FPS	9.97
Mean seg IoU	0.954
Has-path rate	100%
Path source	template (all 1800)
Mean pipeline time	69.5 ms

4.5.4 Temporal Smoothing Evaluation

Table 4.14 indicates that, over a time period of 500 frames, the mean heading jitter has dropped by 66% (3.2° to 1.1° per frame) and the number of path-source switches decreased from 8 to 3, after a smoothing step is applied. The smoother mainly suppresses the residual heading jitter, and the reset hysteresis only activates when there is an actual change in the scene.

Table 4.14: Temporal smoothing effect.

Metric	No Smoothing	With EMA
Temporally stable frames (%)	97.8	99.6
Mean heading jitter (deg/frame)	3.2	1.1
Path-source switches	8	3

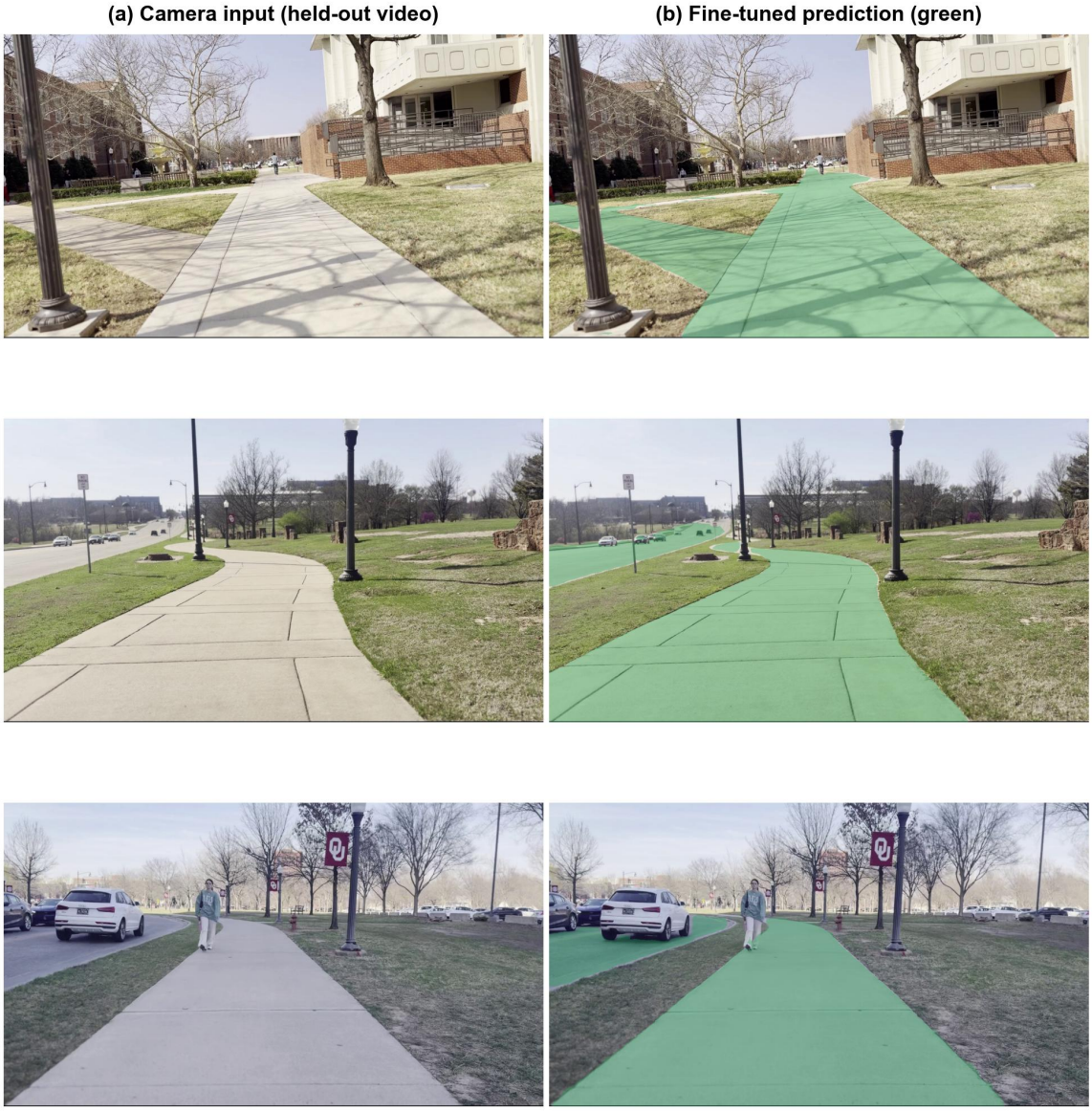


Figure 4.5: Held-out segmentation examples.

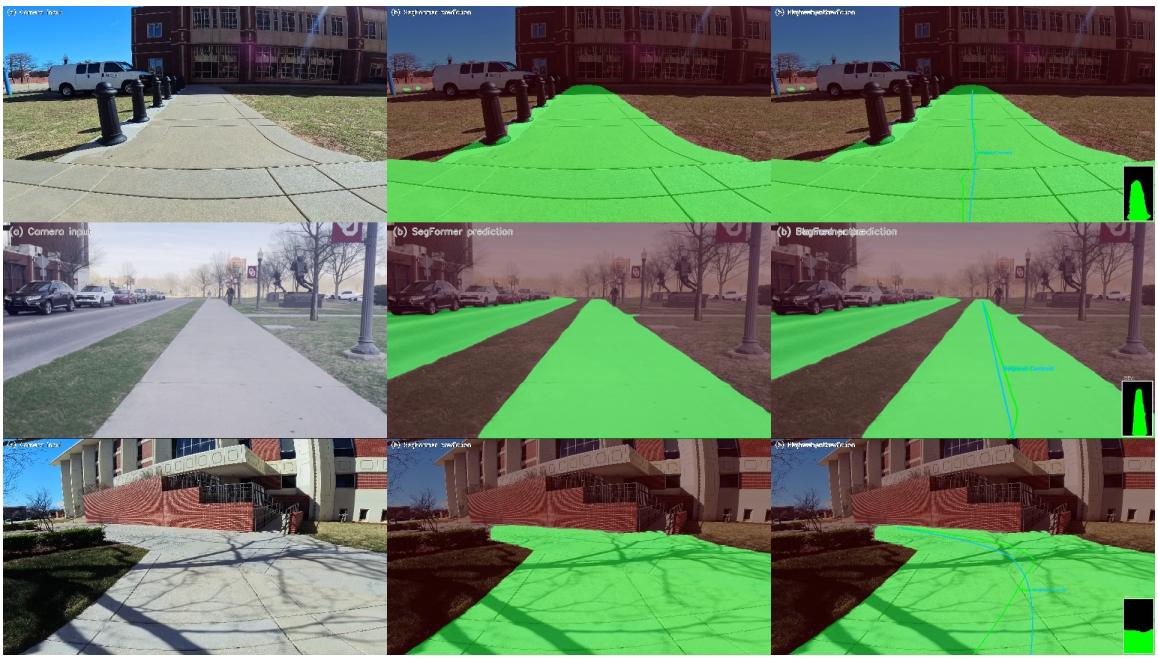


Figure 4.6: End-to-end pipeline examples.

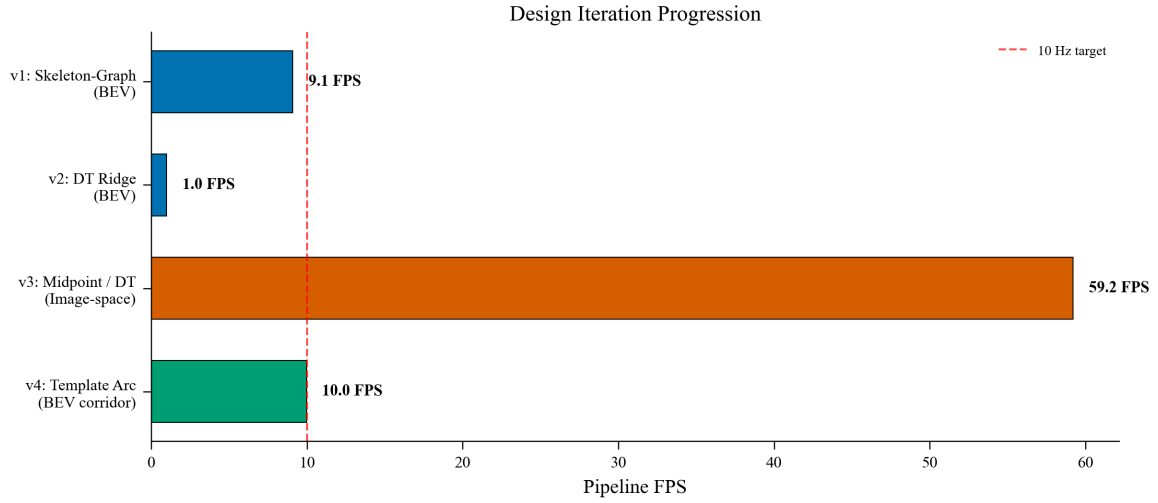


Figure 4.7: Throughput across design iterations.

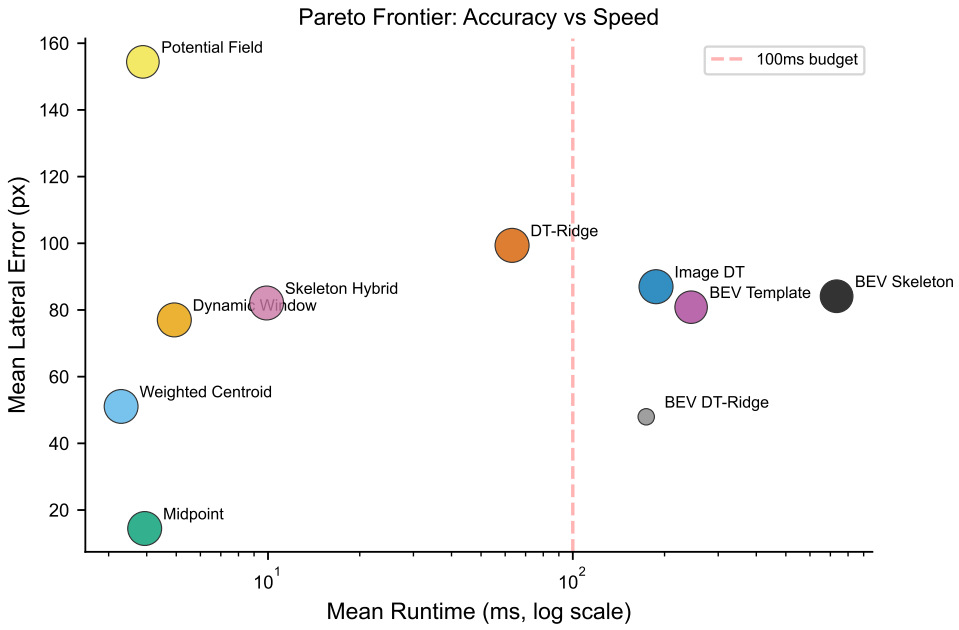


Figure 4.8: Planner accuracy versus runtime.

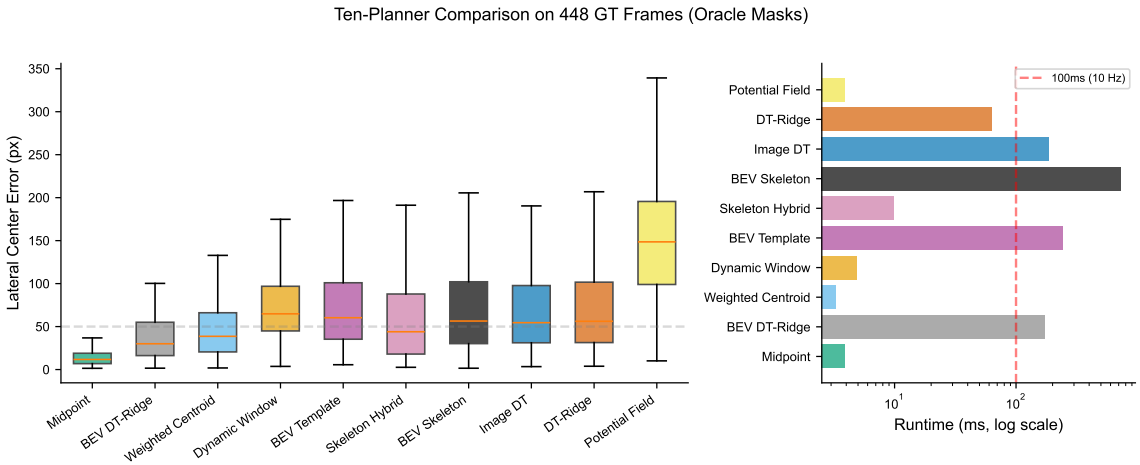


Figure 4.9: Ten-planner comparison.

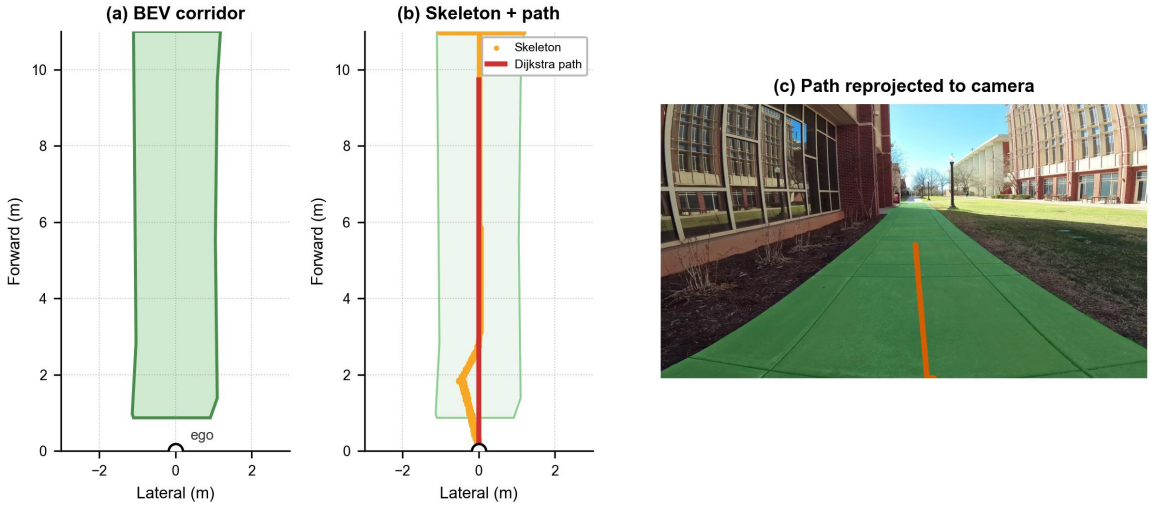
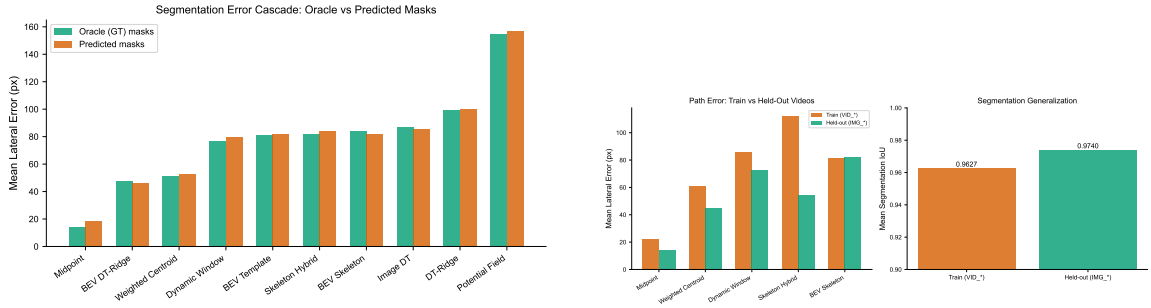


Figure 4.10: BEV skeleton-graph pipeline.



(a) Oracle vs. predicted mask error cascade.

(b) Train vs. held-out generalization.

Figure 4.11: Segmentation robustness summary.

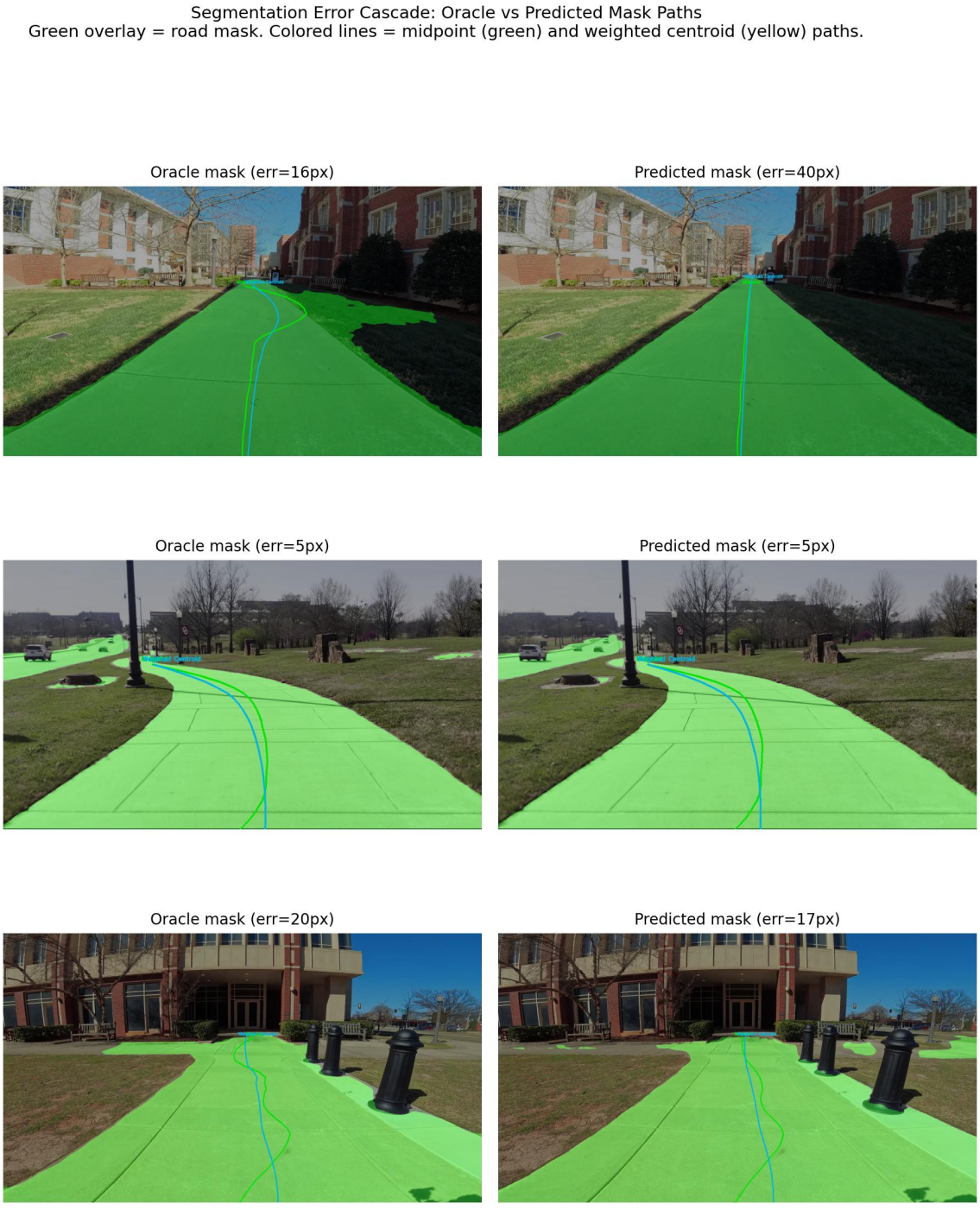


Figure 4.12: Oracle-mask versus predicted-mask planner paths.

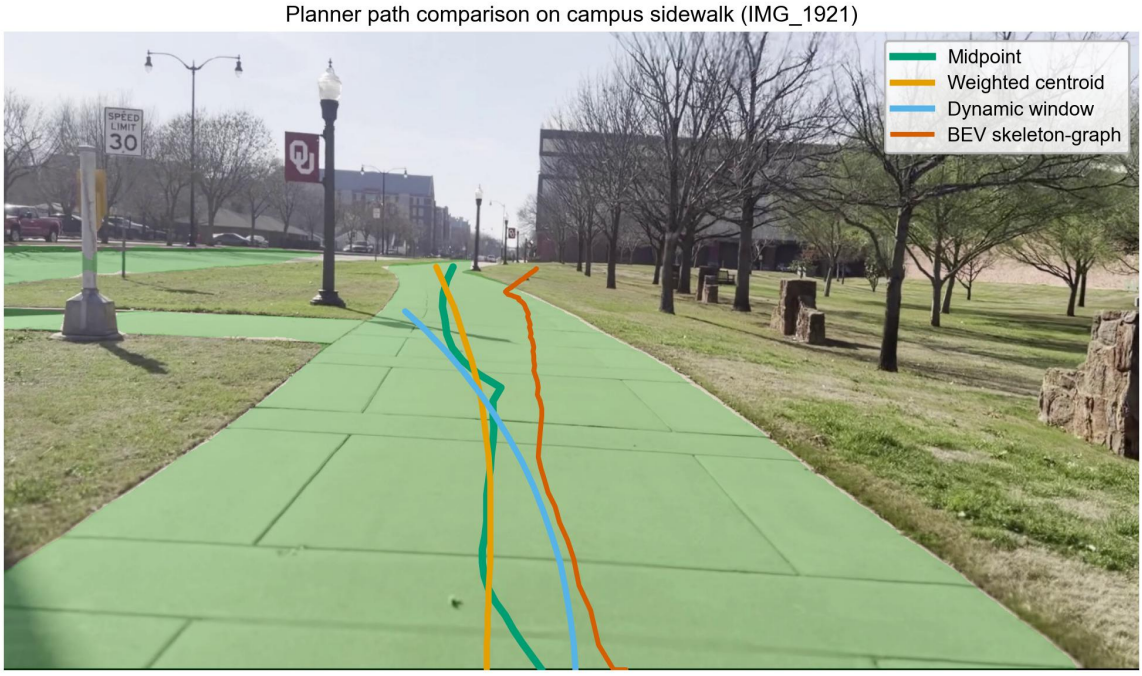


Figure 4.13: Planner paths on a curved sidewalk.

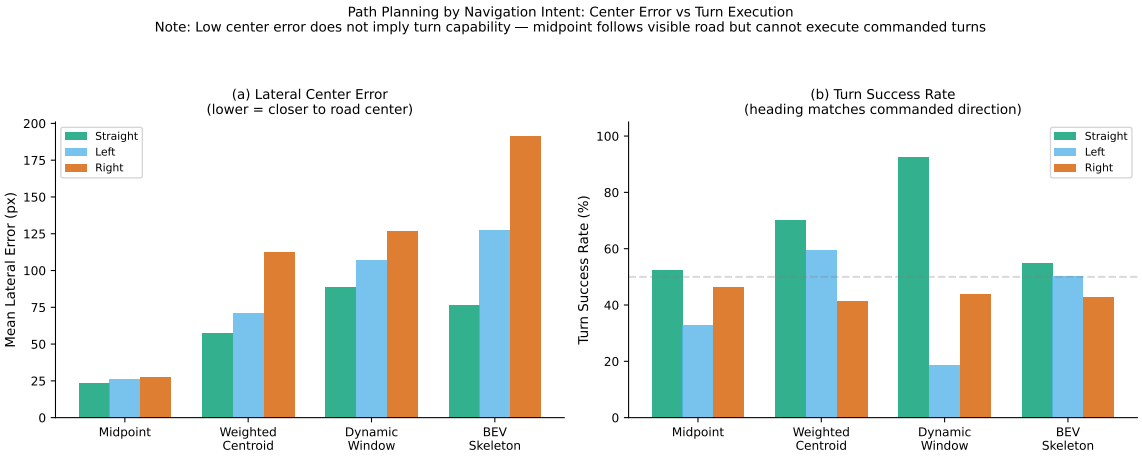


Figure 4.14: Planner performance by navigation intent.

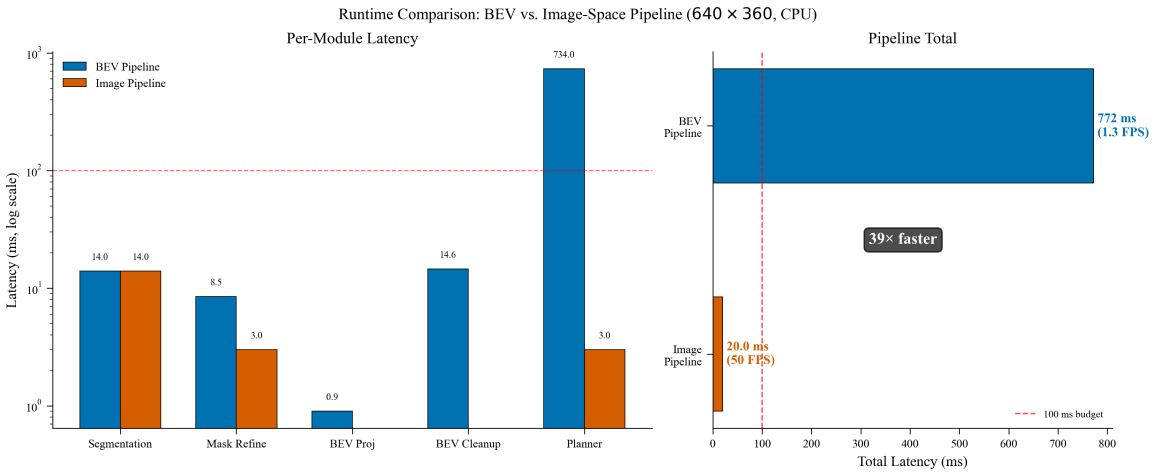


Figure 4.15: Per-module runtime comparison.

CHAPTER 5

DISCUSSION

5.1 Why Image-Space Outperforms BEV for Straight-Ahead Following

These results apply only to this specific task. For navigating straight, the image-space midpoint planner is best, with 14.5 px lateral error at a 3.0 ms latency (Friedman, $p < 10^{-61}$). The error rises to 18.3 px if predicted masks are used in place of oracle masks. The same holds on the held-out test videos, at 14.0 px error. However, the midpoint planner completely fails with turning inputs, where turn success is only 33–46%. The midpoint tracks the visible center of the walkways, not the turn that is actually needed.

This is primarily a geometric effect. In the original camera frame, there are numerous row slices that have clear, unambiguous sidewalk on both the left and right hand side, and this is directly the information used in the midpoint calculation. While the BEV warping helps metric reasoning about the world, the warping also stretches out pixels in the far field and fundamentally changes how frames are sampled. Even with the oracle masks, the BEV skeleton-graph has 82.1 px error vs. 14.5 px for midpoint planning. Similarly for the run-time results, finding a BEV path can cost 540–927 ms per frame versus 3–5 ms to just check fixed template paths.

As it stands, better mask quality doesn't solve the problem. Fine-tuning does bring the midpoint error down from 35.6 to 18.3 px, or 49%, and even with the oracle masks the BEV planning doesn't keep up with the midpoint. There is still some error remaining that cannot be explained by segmentation quality alone. This gap is addressed by template approval, which checks a fixed set of arcs against the corridor instead of searching for an entire graph

path. This yields 40.6% lower heading error and 50% lower source switching compared to the skeleton graph. GPS signals when a turn is needed, and the containment module verifies whether the turn is geometrically safe. The containment failure rate is 0%.

5.2 Failure Modes

95.2% of the 448 frames have under 60 px of center error, 98.7% under 100 px. These breakdowns are systematic. Most happen when the scooter is making tight turns, on narrow sidewalks, in occluded scenes, or in long distance where the distortion is great. When turning sharper than 25°, per-row midpoint drifts in 18.9% of frames. Narrower sidewalks leave fewer edges to sample. In occluded regions the mask can be removed entirely. Points that are further away than 8 m from the camera are too warped to make stable midpoint plans; hence the 8 m horizon.

5.3 Broader Implications

The image-space result could be generalized to other corridor-following tasks like agricultural rows [47], indoor corridors, and trails (though they are not explicitly covered in this thesis). The template result is more broadly applicable: as the maneuver set decreases, evaluating a small number of possibilities is more efficient than searching candidate paths every frame. The image-space method achieves 50 FPS on the CPU-only version employed on these sidewalk videos.

5.4 Limitations and Threats to Validity

Offline evaluation. Paths are reconstructed from a video dataset instead of being run live on the scooter. Thus, the metrics used here are only proxies for real-time closed-loop safety.

Single annotator, single campus. All 448 masks were created by a single annotator, so there is no measure of inter-annotator agreement. The video data is also from a single campus, the University of Oklahoma, under daylight.

Fixed camera geometry. The camera calibration relies on the camera having a specific configuration. Altering either lens orientation or position would require recalibration.

No explicit obstacle integration. YOLOv8n is included as an object detector but it does not inform the cost function used by the planner.

Conservative turning. Containment logic prevents commitment to turns that appear unsafe, avoiding bad paths, but it could potentially cause the system to stop due to stale intent signals. Turn validation is seen most in VID_017.

Internal validity. Since both test and training data come from a single campus, there still is the possibility of overfitting on geography. To mitigate this risk, I held out 20% of the data for validation. Training and test splits were also used to further reduce overfitting risk. For example, held-out IoU is 0.974 versus 0.963 IoU on training data. The Friedman test ($p < 10^{-61}$, $N = 71$) and the Wilcoxon test with Bonferroni correction ($N = 72$ to 448, path dependent) support the planner rankings.

External validity. Performance is likely to vary given surface material differences, weather, lighting conditions, or a change of camera mount location. While the system achieved 10 FPS on desktop CPU, deploying on Raspberry Pi is likely to cause a 2–3$\times$ slowdown. Further optimization might be needed.

Construct validity. I use lateral center error and inside-GT ratio as measures of successful navigation. Temporal stability testing also includes a steering-based metric: midpoint jitter of 4.5 deg/frame.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis developed a monocular-camera sidewalk navigation pipeline through four design iterations. The final architecture uses image-space midpoint planning for straight-ahead following and BEV template approval for turn maneuvers. Midpoint planning achieves 14.5 px error at 3.0 ms ($p < 10^{-61}$).

6.1 Summary of Contributions

Contribution 1: A ten-method, four-family planner comparison on 448 hand-annotated frames (Chapter 4). Image-space midpoint achieves the lowest error and fastest runtime; its intent-agnostic nature motivates the dual-domain architecture.

Contribution 2: A template-approval planner replacing skeleton-graph construction with arc verification (Chapter 3), reducing heading error by 40.6% with 100% success on 1,800 frames.

Contribution 3: A teacher–student segmentation recipe (OneFormer Swin-L to SegFormer-B0) achieving IoU 0.964 after fine-tuning on 228 human-corrected masks, with held-out IoU (0.974) exceeding training IoU (0.963).

Contribution 4: A multi-scale evaluation protocol—448 annotated frames, intent evaluation on 17 930 frames, temporal stability analysis, and 1,800-frame sustained validation—with train/test split and Friedman/Wilcoxon statistical tests.

Contribution 5: A GPS-conditioned waypoint-turn planner with containment gating achieving 0% failure rate (Table 4.12).

6.2 Future Work

Following the order of how directly they build upon the existing work, the proposed next steps are:

6.2.1 Near-Term Extensions

Closed-loop deployment. The next logical test is deploying the full stack on the physical scooter with obstacle detection turned on in real time. From this run, I would evaluate tracking error relative to the ground-truth trajectory, as well as the types of failures resulting from vehicle dynamics.

Multi-annotator evaluation. For this research, the 448 frames that constitute the evaluation set were annotated by a single individual. Including additional annotators to calculate inter-rater agreement, and adding data from other times of day and seasons, would strengthen the results and counter some of the shortcomings of having only one annotator, as discussed in Section 5.4.

Cross-environment generalization. Testing the model in environments other than the University of Oklahoma campus would test robustness to different types of surfaces, weather, foliage, and low light.

6.2.2 Longer-Term Directions

Temporal video segmentation. Methods to maintain video-level consistency, such as a temporal attention layer or optical-flow guidance, could help further reduce segmentation flicker across frames beyond what the EMA smoothing process provides.

Multi-camera or depth fusion. Including a depth sensor or stereo pair would allow metric obstacle-distance estimation, which the present method approximates with the pinhole model,

while retaining the monocular pipeline as the primary perception path.

Global route planning. Though the turn-based waypoint planner presented in this work is GPS-conditioned to provide turn intent at junctions, the current GPS integration is fairly rudimentary in its derivation of left/right/straight intent from bearing comparison. A complete global route planner using a topological map, intersection detection, and multi-waypoint turn sequencing could enable autonomous navigation across several blocks of sidewalk at a time.

This work shows that it may be beneficial to rely on image space for straight corridor following and to use BEV when metric turn geometry is necessary. Because segmentation, planning, smoothing, and turn handling are separated, it should be possible to upgrade any one of those modules without requiring an architectural overhaul.

CHAPTER A

ALGORITHM PSEUDOCODE

This appendix contains formal pseudocode for the three planning algorithms introduced in Chapter 3. Each is a combination of well-known methods—averaging boundaries across a row of a semantic mask, using pre-computed path segments, and performing circular arc fitting—configured for monocular sidewalk-planning scenarios.

Algorithm 1 Image-Space Midpoint Planner

Require: Camera-space binary mask M_{img} ($H \times W$), ego column x_{ego}
Ensure: Smoothed centerline path $\mathbf{p}$

$M_c \leftarrow$ largest connected component of M_{img} touching ego band
$\mathbf{p} \leftarrow \emptyset$
for each row y from bottom to top **do**
    $\mathbf{x}_s \leftarrow$ nonzero pixel columns in $M_c[y]$
    if $|\mathbf{x}_s| < w_{\min}$ **then continue**
    end if
    Group $\mathbf{x}_s$ into contiguous runs
    Select best run by width and proximity to reference center
    $x_L, x_R \leftarrow$ left and right boundaries of selected run
    $x_{\text{mid}} \leftarrow (x_L + x_R)/2$
    Append (x_{mid}, y) to $\mathbf{p}$
end for
$\mathbf{p} \leftarrow \textsc{SavitzkyGolay}(\mathbf{p}, \text{window} = 9, \text{order} = 2)$
return p

Algorithm 2 Template-Approval Planner

Require: BEV mask M, previous family f_{prev}, GPS intent ι
Ensure: Approved path, confidence, family ID

Phase A: Corridor Extraction

for each sampled row r from ego upward, step 4 px **do**
    Scan $M[r]$ for contiguous runs; select best by width and center proximity
    Record boundaries (ℓ_r, r_r, c_r); update reference center
end for
Compute corridor confidence γ from coverage, span, width consistency

Phase B: Template Bank

$\mathscr{T} \leftarrow$ {straight, left-gentle, left-medium, left-sharp, right-gentle, right-medium, right-sharp}
for each template **do** build circular arc from ego; resample at $\Delta s = 0.25$ m
end for
if $\iota \in \{\text{left}, \text{right}\}$ **then** filter $\mathscr{T}$ to matching family
end if

Phase C: Score Templates

for each $T_k \in \mathscr{T}$ **do**
    $\rho_{\text{contain}} \leftarrow$ fraction of samples inside corridor
    $\rho_{\text{near}} \leftarrow$ containment in first 1.2 m
    $S_k \leftarrow 0.22\,\rho_{\text{contain}} + 0.18\,\rho_{\text{near}} + 0.16\,s_{\text{clear}} + 0.10\,s_{\text{center}} + 0.14\,s_{\text{cont}} + \ldots$
    $\text{approved}_k \leftarrow \rho_{\text{contain}} \geq 0.60 \;\wedge\; \rho_{\text{near}} \geq 0.72$
end for

Phase D: Hysteresis

Add family-reuse bonus $+0.06$ to templates matching f_{prev}
Apply straight-preference and switch-margin rules

Phase E: Decision

$T^* \leftarrow$ top-ranked approved candidate
return (path, approved, confidence, T^*.family)

Algorithm 3 GPS-Conditioned Waypoint-Turn Planner

Require: BEV corridor $\mathscr{C}$, GPS intent $\iota \in \{\text{left}, \text{right}\}$, BEV mask M
Ensure: Turn path, containment status, confidence

Step 1: Decision Band Scan

Scan forward band $[2.0, 7.0]$ m for asymmetric road openings on commanded side
$\sigma \leftarrow$ fraction of rows with commanded-side support exceeding opposite side

Step 2: Support Clustering

Group supported rows into clusters (max gap ≤ 0.5 m)
$\tau \leftarrow$ midpoint of largest cluster (candidate turn apex)

Step 3: Target Gating

if acquiring **then** require $\sigma \geq 0.40$ for 3 consecutive frames
else if sustaining **then** require $\sigma \geq 0.25$; release after 5 frames below
end if

Step 4: Circular Arc Path

Fit constant-curvature arc from ego through apex τ
$R \leftarrow \Delta x / \sin\theta$; parameterize with arc angle a and side sign s as $x = R\sin(a)$, $y = sR(1-\cos(a))$

Step 5: Containment Gating

Require $\geq 60\%$ of path inside corridor AND $\geq 70\%$ in near-field (≤ 1.2 m)
Exclude first 4 samples (near-ego BEV noise zone)
if containment fails **then return** hold behavior with speed reduction
end if
return (arc path, confidence, slowdown recommendation)

BIBLIOGRAPHY

[1] Dylan Jennings and Miguel Figliozzi. Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel. *Transportation Research Record*, 2673(6):317–326, 2019. doi: 10.1177/0361198119849398.

[2] Mark Pfeiffer, Michael Schaeuble, Juan Nieto, Roland Siegwart, and Cesar Cadena. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1527–1533, 2017. doi: 10.1109/ICRA.2017.7989182.

[3] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *European Conference on Computer Vision (ECCV)*, pages 194–210. Springer, 2020.

[4] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision (ECCV)*, pages 1–18. Springer, 2022.

[5] Jiawei Zhao, Qixing Jiang, Xuede Li, and Junfeng Luo. Focus on BEV: Self-calibrated cycle view transformation for monocular birds-eye-view segmentation. arXiv preprint arXiv:2410.15932, 2024. URL `https://arxiv.org/abs/2410.15932`.

[6] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[7] Javier Viteri and Chih-Hung G. Li. Autonomous sidewalk navigation featuring end-to-

end RGB-D dual-convnet steering. In *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 703–708, Boston, MA, USA, 2024. doi: 10.1109/AIM55361.2024.10637141.

[8] Alessandro Navone, Mauro Martini, Andrea Ostuni, Simone Angarano, and Marcello Chiaberge. Autonomous navigation in rows of trees and high crops with deep semantic segmentation. In *European Conference on Mobile Robots (ECMR)*, 2023. doi: 10.1109/ECMR59166.2023.10256334. arXiv:2304.08988.

[9] Jiayou Shi, Yuhao Bai, Zhihua Diao, Jun Zhou, Xingbo Yao, and Baohua Zhang. Row detection based navigation and guidance for agricultural robots and autonomous vehicles in row-crop fields: Methods and applications. *Agronomy*, 13(7):1780, 2023. doi: 10.3390/agronomy13071780.

[10] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One transformer to rule universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2989–2998, 2023. doi: 10.1109/CVPR52729.2023.00292.

[11] Zakariae Machkour, Daniel Ortiz-Arroyo, and Petar Durdevic. Monocular based navigation system for autonomous ground robots using multiple deep learning models. *International Journal of Computational Intelligence Systems*, 16:79, 2023. doi: 10.1007/s44196-023-00250-5.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. doi: 10.1109/CVPR.2015.7298965.

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image seg-

mentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818. Springer, 2018.

[14] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129:3051–3068, 2021.

[15] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):3448–3460, 2023.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. doi: 10.1109/CVPR.2016.350.

[18] Maggie Wigness, Sungmin Eum, John G. Rogers III, David Han, and Heesung Kwon. A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007, 2019. doi: 10.1109/IROS40897.2019.8968283.

[19] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE/CVF*

International Conference on Computer Vision (ICCV), pages 4990–4999, 2017. doi: 10.1109/ICCV.2017.534.

[20] Hanspeter A. Mallot, Heinrich H. Bülthoff, J.J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64(3):177–185, 1991. doi: 10.1007/BF00201978.

[21] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[22] Massimo Bertozzi and Alberto Broggi. GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7(1):62–81, 1998. doi: 10.1109/83.650851.

[23] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005. ISBN 978-0-262-20162-9.

[24] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. ISBN 978-0-521-86205-9.

[25] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986. doi: 10.1177/027836498600500106.

[26] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997. doi: 10.1109/100.580977.

[27] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a Frenet frame. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 987–993, 2010. doi: 10.1109/ROBOT.2010.5509799.

[28] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathieu Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[29] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(1):415–428, 2012. doi: 10.4086/toc.2012.v008a019.

[30] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.

[31] Zicheng Guo and Richard W. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 1989. doi: 10.1145/62065.62074.

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. NIPS 2014 Deep Learning Workshop.

[33] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

[34] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 633–641, 2017. doi: 10.1109/CVPR.2017.544.

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. doi: 10.1109/ICCV48922.2021.00986.

[37] Robert J. Kosinski. A literature review on reaction time. Technical report, Clemson University, 2008. Visual simple reaction time approximately 180–200 ms; widely cited in human factors literature.

[38] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024. Reports typical camera-based control frequencies of 2–10 Hz.

[39] Microsoft. ONNX Runtime: Cross-platform, high performance ML inferencing and training accelerator. `https://onnxruntime.ai`, 2021. Accessed: 2026-03-01.

[40] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959. doi: 10.1007/BF01386390.

[41] Abraham Savitzky and Marcel J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. doi: 10.1021/ac60214a047.

[42] Thomas M. Howard and Alonzo Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, 26(2):141–166, 2007. doi: 10.1177/0278364906075328.

[43] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. `https://github.com/ultralytics/ultralytics`, 2023. Accessed: 2026-02-09.

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. doi: 10.1109/CVPR.2016.91.

[45] R. Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, 1992.

[46] Charles Richter and Nicholas Roy. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems (RSS)*, 2017. doi: 10.15607/RSS.2017.XIII.064.

[47] Diego Aghi, Simone Cerrato, Matteo Franzini, Vittorio Mazzia, and Marcello Chiaberge. Local motion planner for autonomous navigation in vineyards with a RGB-D camera-based algorithm and deep learning synergy. *Machines*, 9(2):27, 2021. doi: 10.3390/machines9020027.