THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AEROSPACE ENGINEERING

COMPLEXITY-AWARE PATH PLANNING FOR ONLINE UAV-BASED 3D RECONSTRUCTION

ALEXANDRIA RHOADS
SPRING 2026

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Aerospace Engineering
with honors in Aerospace Engineering

Reviewed and approved* by the following:

Junyi Geng
Assistant Professor of Aerospace Engineering
Thesis Supervisor

Sven Schmitz
Professor of Aerospace Engineering
Honors Adviser

*Signatures are on file in the Schreyer Honors College.

Abstract

Three-dimensional reconstruction from UAV imagery is a powerful tool for inspection, mapping, and documentation, but current approaches have a significant inefficiency built into them. Because standard trajectory planners treat every part of an environment the same way, the UAV ends up collecting dense imagery everywhere — over flat walls and open ground just as much as over the corners, edges, and structural features that actually drive reconstruction quality. The result is a massive dataset, most of which contributes little to the final model, and that dataset has to be processed offline before anything useful can be extracted from it.

The core argument of this thesis is that a smarter trajectory can change this. If the UAV spends more of its flight time around geometrically complex regions — the parts of an environment where surface detail, occlusions, and structural variation make accurate reconstruction genuinely difficult — and moves more quickly through simple areas, it collects a smaller total dataset that is disproportionately made up of the imagery that actually matters. Less data means less post-processing, and the data that does get collected is the data the reconstruction algorithm needs most.

To achieve this, two complementary path planners are developed, both built on the A* search framework. The first is a β-weighted formulation that blends a complexity reward directly into the planning cost, with a tunable parameter controlling how strongly the UAV is drawn toward complex regions. This approach works and is easy to use, but it cannot guarantee that the path it

finds is the best possible one for any given flight budget.

The main contribution of this thesis addresses that limitation. The problem is recast as a constrained maximization: find the path that collects the most geometric complexity while staying within a hard limit on total flight distance. This formulation is equivalent in structure to the orienteering problem from combinatorial optimization. Solving it exactly requires an effective admissible heuristic, and designing that heuristic is the central technical challenge. The result is the *ellipse heuristic*: at any point during planning, a cell can only be visited on a feasible continuation of the current path if the total distance to travel to that point and continue to the goal is less than the remaining flight budget. This condition defines an ellipse with foci at the current position and the goal. Restricting the heuristic's optimistic complexity estimate to cells inside this ellipse produces a bound that is provably tighter than any geometry-blind alternative. Formal proofs of admissibility, consistency, and dominance over the global baseline are provided.

The planner is tested on synthetic environments designed to isolate different aspects of the algorithm's behavior, and then integrated with real sensor data from a recorded UAV inspection flight. The high fidelity simulation uses voxel occupancy and complexity maps extracted from a ROS bag, allowing a direct comparison between the trajectory the algorithm would have planned and the one the UAV actually flew. The algorithm also handles dynamic environments through a replanning extension that correctly tracks remaining flight budget across replanning events. In every case, the planned paths satisfy the single-visit constraint exactly and route meaningfully through the high-complexity regions that the original inspection trajectory ignored.

Table of Contents

List of Figures

List of Tables

Nomenclature

$f(n)$	=	A* total evaluation function
$g(n, \ell)$	=	Accumulated (negated) cost-to-come
$h(n, \ell)$	=	Heuristic estimate of cost-to-go
$h^*(n, \ell)$	=	True optimal cost-to-go
$c(n, m)$	=	Edge transition cost
$d(n, m)$	=	Euclidean distance between n and m
$C(n)$	=	Environmental complexity at node n
β	=	Complexity weighting parameter
n_0, n_g	=	Start and goal nodes
D^*	=	Straight-line start-to-goal distance
δ	=	Detour budget
ℓ	=	Accumulated step count
R	=	Remaining budget: $(D^* + \delta) - \ell$
$\mathcal{E}(n, \ell)$	=	Reachable ellipse at state (n, ℓ)
a, b, c	=	Ellipse semi-major axis, semi-minor axis, focal half-distance
$S_{\mathcal{E}}(R)$	=	Top-R complexity prefix sum within $\mathcal{E}$

Acknowledgements

My sincerest thanks to Dr. Geng and Xiangfu Li who have been instrumental in my understanding of the research process. All of this work is a product of what you have taught me and how you have helped me grow as a researcher. I look forward to continuing this work with you!

Additional thanks to Dr. Schmitz for being my Honors Advisor for the past two years. I have appreciated your guidance immensely.

Thank you to all of my friends and family who listened to me complain about simulations crashing and pretended that anything I said made sense.

To the 2015 laptop I used for all of my simulations, thank you for not giving out on me (yet). You didn't work very well, but we got there in the end. I appreciate that you usually only crashed after I got the test results I needed.

To my roommates, Madeline and Patti, thank you for putting up with my three different laptops being scattered throughout the apartment. I promise they were all necessary for the work found in this thesis.

Finally, I would like to acknowledge the Panera on South Allen for being the location where all of this work was done. I could not have completed this thesis without the long, Diet Pepsi fueled work sessions at the Panera. So, to both Panera and Diet Pepsi I am eternally grateful.

Chapter 1

Introduction

1.1 Motivation

Unmanned aerial vehicles have become indispensable platforms for the reconstruction of 3D environments in infrastructure inspection, cultural heritage documentation, and autonomous navigation [1]. Three-dimenstional reconstruction is the process of building a geometric model of a physical enviroment from sensor data. In the context of UAV-based reconstruction, a drone flies through or around a scene collecting overlapping images, and photogrammetry software processes those images to produce a point cloud or mesh that represents the structure's geometry. Standard approaches prescribe uniform coverage trajectories that treat geometrically simple and complex regions identically, producing large redundant datasets which cannot be processed online [2]. Rather

than consuming additional resources in post-processing efforts, there is a push towards online processing. In order to achieve this online processing, it is necessary to develop a planner that allocates sensing effort *selectively*, dwelling in geometrically rich regions, and transiting simple ones efficiently.

Next-Best-View (NBV) planning [3, 4] provides an information-driven alternative by iteratively selecting the viewpoint maximizing the information-theoretic utility. These methods demonstrate real efficiency gains, but treat navigation as discrete viewpoint selection: the path *between* viewpoints is geometrically planned with no awareness of intermediate complexity.

A more principled approach is to embed informational value directly into the path search itself. A* is a graph search algorithm that finds optimal paths by combining the actual cost to reach a node with a heuristic estimate of the remaining cost to the goal [5], and its optimality guarantees make it a natural foundation for exact planning on grid-structured environments. Embedding the informational value directly into the A* path planning algorithm ensures that every step is guided toward geometrically informative regions [6], rather than treating complexity as a post-hoc selection criterion between pre-planned waypoints.

Adapting A* to this setting, however, introduces a central challenge: heuristic design. In classical shortest-path planning, Euclidean distance serves naturally as the heuristic because the true objective — path length — is geometrically grounded and trivially bound. When the objective changes to maximizing accumulated complexity, no such natural bound exists, and a new heuristic is needed that optimistically estimates the remaining collectible complexity within the budget while remaining admissible and consistent. Designing such a heuristic that is tight enough to prune the search space effectively is the core technical contribution of this thesis.

1.2 Problem Statement

Let $G = (V, E)$ be a graph of the navigable environment where each vertex $v \in V$ carries a nonnegative complexity score $C(v) \geq 0$. Given start n_0, goal n_g, and detour budget $\delta \geq 0$, the primary objective is to find a path π solving:

$$\max_{\pi} \sum_{v \in \pi} C(v) \quad \text{subject to} \quad L(\pi) \leq D^* + \delta, \quad v \text{ visited at most once}, \tag{1.1}$$

where $D^* = d(n_0, n_g)$ is the straight-line distance and $L(\pi)$ is the total path length. This is structurally equivalent to the orienteering problem [7], which is NP-hard in general [8]. The approach developed here exploits grid topology and the geometric budget constraint to make exact search tractable via an effective, consistent heuristic.

1.3 Related Work

1.3.1 UAV Reconstruction and Coverage Planning

UAV photogrammetry reconstructs 3D models from overlapping images along a planned path. Quality depends critically on the spatial distribution of image capture [1]. Naive lawnmower patterns guarantee coverage of flat surfaces but fail to resolve complex features such as overhangs and recesses [2]. Model-based methods compute viewpoints maximizing coverage and stereo overlap [6]. A key limitation across these pipelines is reliance on offline processing; online reconstruction demands trajectory planning methods producing smaller, higher-quality datasets [9].

1.3.2 Next-Best-View and Informative Path Planning

NBV planners iteratively select the viewpoint maximizing an information-theoretic gain [3]. Delmerico et al. [4] compare volume-based and surface-based information gains, demonstrating that utility function choice significantly affects efficiency. The broader *informative path planning* (IPP) framework treats the problem as optimizing total information gathered along a route [10]. Lim and Shan [11] extend IPP to adaptive settings with online model updates. Zhou et al. [12] propose topology-guided UAV reconstruction but require a pre-computed topological map and provide no optimality guarantee. The present work provides exact optimality for a fully observed environment map via A*-based search, distinguishing it from sampling-based IPP approaches.

1.3.3 Heuristic Search and the Orienteering Problem

The consistent A* formulation is grounded in heuristic search theory [5]. A heuristic h_2 dominates h_1 if $h_2(n) \geq h_1(n)$ for all n while both remain admissible; a dominating heuristic prunes at least as many nodes. The orienteering problem — maximize collected rewards along a bounded-length tour [7] — is the combinatorial analogue of Equation 1.1. Its NP-hardness [8] motivates the need for effective heuristic pruning to make exact search tractable on grid-scale instances.

1.4 Contributions

1. A β**-weighted complexity-aware A*** baseline incorporating complexity reward into the cost function with a corridor constraint, establishing the architectural foundation and exposing the limitations that motivate the main contribution.

2. An **consistent complexity-maximizing A* formulation** providing a provably optimal solution to Equation 1.1 via negated costs, augmented state (n, ℓ), and a bitmask single-visit constraint.

3. The **ellipse heuristic**: at planning state (n, ℓ) with remaining budget $R = (D^* + \delta) - \ell$, the heuristic restricts its optimistic complexity estimate to the reachable ellipse $\mathcal{E}(n, \ell) = \{v : d(n, v) + d(v, n_g) \leq R\}$. Formal proofs establish admissibility, consistency, and strict dominance over the global prefix-sum baseline.

4. **Experimental validation** across synthetic multi-hotspot environments and integration with real sensor data from a UAV inspection flight, including dynamic replanning under moving obstacles.

Chapter 2

Methods

2.1 Environmental Complexity Field

The planner operates on a discrete graph $G = (V, E)$ where each vertex $v \in V$ represents a grid cell and carries a non-negative complexity score $C(v) \geq 0$ encoding the geometric information density of that cell. In the synthetic experiments, this field is constructed as a sum of Gaussian profiles approximating the spatial structure expected from a real onboard estimation pipeline. In the simulated experiments, it is derived directly from a voxel map resulting from simulated sensor data. The planner's architecture accepts any non-negative $C(v)$ without modification.

The complexity score $C(v)$ assigned to each voxel quantifies the local geometric richness observable from that position. In this work, scores are derived from the density of distinct surface

normal directions within each voxel's neighbourhood: voxels near planar surfaces receive low scores, while voxels near edges, corners, or structurally varied geometry receive high scores. This reflects the intuition that geometrically complex regions yield more informative sensor observations and therefore deserve greater coverage effort. Any scoring pipeline producing $C(v) \geq 0$ over the same grid is a valid drop-in input to the planner.

2.2 Motivating Baseline: β-Weighted A*

Classical A* finds a least-cost path from start n_0 to goal n_g by maintaining a priority queue ordered by $f(n) = g(n) + h(n)$, where $g(n)$ is the cost accumulated from n_0 to n, and $h(n)$ is a consistent heuristic estimating the remaining cost to n_g [13, 5]. Admissibility — the requirement that $h(n)$ never overestimates the true remaining cost — guarantees that the first path found to the goal is optimal. The efficiency of A* depends directly on the tightness of h: a tighter heuristic prunes more of the search space without sacrificing optimality. All planners developed in this work are built on this foundation.

2.2.1 Formulation

Throughout this section, n and m denote grid cells (vertices of G), n_g denotes the goal cell, $d(n, m)$ denotes Euclidean distance between cells, $g_L(n)$ denotes the accumulated path length from n_0 to n, $D^* = d(n_0, n_g)$ is the straight-line start-to-goal distance, and Δ is the maximum permitted detour above D^*.

The β-weighted formulation embeds complexity reward directly into the A* edge cost:

$$c(n, m) = d(n, m) - \beta\, C(m), \qquad \beta \geq 0 \tag{2.1}$$

At $\beta = 0$ this recovers classical A* exactly [13]. As β increases, high-complexity nodes become increasingly rewarded. The Euclidean heuristic $h(n) = d(n, n_g)$ is retained, and a corridor constraint $g_L(n) + d(n, n_g) \leq D^* + \Delta$ bounds total path length regardless of β. The formulation does not provide an optimality guarantee for the complexity objective: the blended cost function $c(n, m) = d(n, m) - \beta\, C(m)$ is not consistent as a minimization objective for complexity, and the specific path found depends on the implicit tradeoff encoded in β rather than on direct optimization of complexity [5].

2.2.2 Limitations Motivating the Consistent Formulation

Three observations from the β-weighted experiments motivate the main contribution. First, β has no direct correspondence to any quantifiable resource constraint, making it difficult to specify a value that achieves a desired complexity-length tradeoff. Second, and most fundamentally, the blended cost $c(n, m) = d(n, m) - \beta\, C(m)$ rewards only the complexity of the immediate next cell: it has no mechanism for estimating or accounting for the complexity collectible along the remainder of the path. This myopic one-hop reward means the agent cannot reason about whether detouring now will lead to richer regions later. Third, without a formal optimality guarantee it is impossible to know whether the returned path is the best achievable for any given length budget. These gaps motivate the consistent formulation, whose central design challenge is constructing a heuristic that explicitly predicts future collectible complexity rather than relying on immediate

reward.

2.3 Consistent Complexity-Maximizing A*

2.3.1 Problem Reformulation

The goal of the consistent formulation is to design a planner that reasons explicitly about future collectible complexity — not just the immediate next step — while respecting a hard path-length constraint. This requires an A* heuristic that estimates how much complexity remains reachable from the current state given the remaining budget, serving as a principled prediction of future reward. Formally, the objective is to solve Equation 1.1 from the introduction: maximize $\sum_{v \in \pi} C(v)$ subject to $L(\pi) \leq D^* + \delta$ and single-visit. The detour budget δ directly encodes the available energy margin above the minimum-energy mission, and the formulation is structurally equivalent to the orienteering problem [7].

2.3.2 Reduction to A* Minimization

The complexity-maximizing objective is not a reformulation of the β-weighted approach but a direct encoding of Equation 1.1 into A*. Since A* minimizes, the sign of all complexity terms is negated so that maximizing complexity becomes equivalent to minimizing its negation:

$$g(n, \ell) = -\sum_{v \in \pi_{n_0 \to n}} C(v), \qquad c(n, n') = -C(n') \tag{2.2}$$

The state is augmented from a single node n to a pair (n, ℓ), where ℓ is the number of steps taken from n_0 to reach n along the current path. This augmentation is necessary because the

remaining budget $R = (D^* + \delta) - \ell$ — which determines which future cells are still reachable — depends on ℓ: two paths that arrive at the same node via different numbers of steps have consumed different fractions of the budget and therefore face different future opportunities [5]. Without this augmentation, A* could incorrectly treat these two situations as equivalent and discard the better one. Before generating any successor, the feasibility check $\ell + d(n, n_g) \leq D^* + \delta$ is applied, pruning states from which the goal is unreachable within budget.

2.3.3 Single-Visit Enforcement

The single-visit constraint is enforced via a per-path bitmask: a variable-precision integer where bit $(x \cdot G + y)$ records whether cell (x, y) has been visited on the current path. Membership testing and update are both $O(1)$. Crucially, the bitmask is path-specific: each heap entry carries an independent copy, preventing cross-contamination between search branches. Path reconstruction therefore uses linked predecessor entry objects rather than a keyed parent table, since two entries at the same (n, ℓ) may have different visited histories.

2.4 The Ellipse Heuristic

2.4.1 Geometric Foundation

The key insight is that not all cells can contribute to any feasible continuation from state (n, ℓ). A cell v can be visited while still reaching n_g within remaining budget R if and only if:

$$d(n, v) + d(v, n_g) \leq R \tag{2.3}$$

This is the equation of an ellipse with foci at n and n_g. The geometric parameters are:

$$a = \frac{R}{2}, \qquad c = \frac{d(n, n_g)}{2}, \qquad b = \sqrt{a^2 - c^2} = \sqrt{\left(\frac{R}{2}\right)^2 - \left(\frac{d(n, n_g)}{2}\right)^2} \tag{2.4}$$

The semi-major axis a is controlled by the remaining budget R; the semi-minor axis b encodes the lateral reach — how far the agent can deviate from the direct n-to-n_g line. As ℓ increases and R shrinks, both a and b shrink and the ellipse contracts automatically toward the direct path. At the degenerate limit $R = d(n, n_g)$, $b = 0$ and the ellipse collapses to a line segment with no lateral freedom. Figures 2.1 and 2.2 demonstrate the physical meaning of the ellipse constructed within this heuristic formulation.

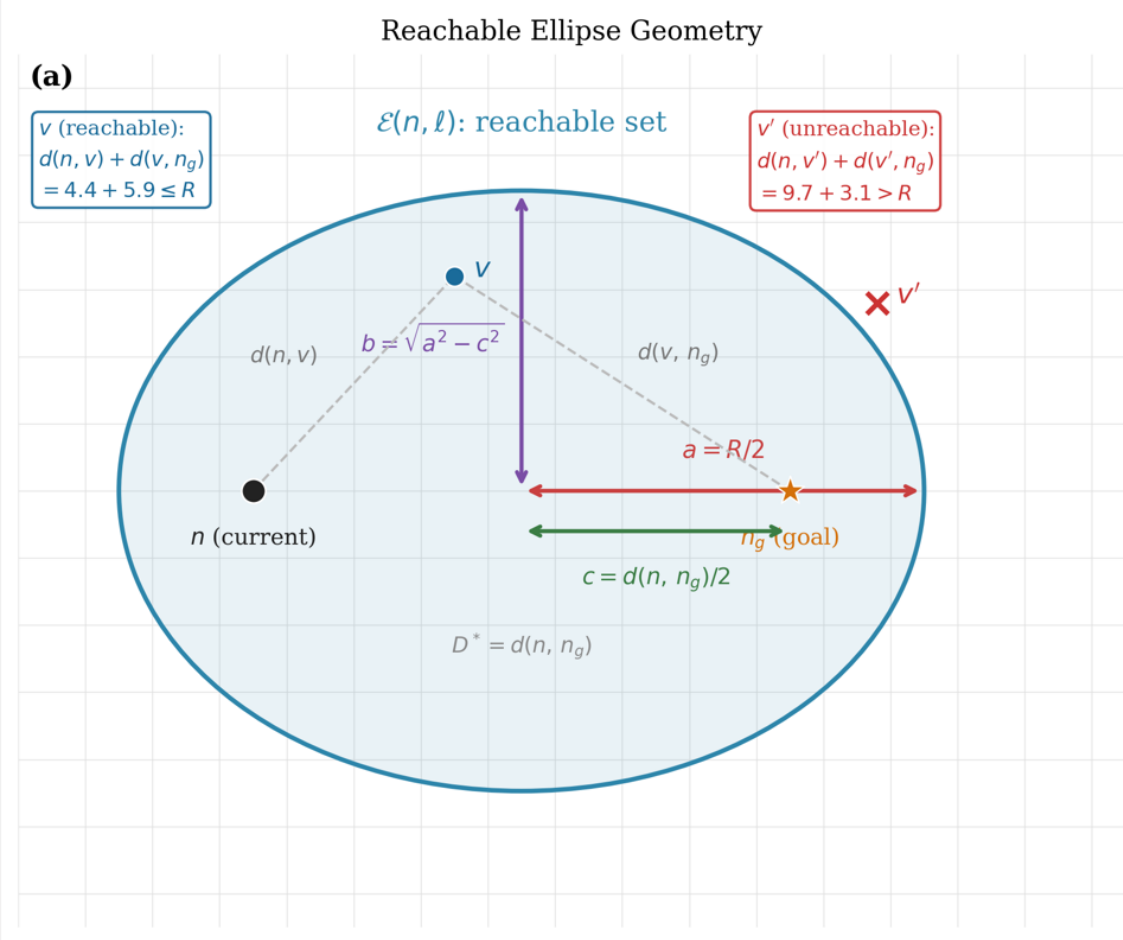

Figure 2.1: **(a)** The reachable ellipse $\mathcal{E}(n, \ell)$ with foci at the current node n and goal n_g. The three annotated axes show the semi-major axis $a = R/2$ (red, rightward from centre), the semi-minor axis $b = \sqrt{a^2 - c^2}$ (purple, upward from centre), and the focal half-distance $c = d(n, n_g)/2$ (green). Dashed lines indicate the two-leg distances used in the reachability check. Cell v lies inside the ellipse and satisfies $d(n, v) + d(v, n_g) \leq R$; cell v' lies outside it and violates the inequality, so it cannot be visited on any feasible completion of the current path.

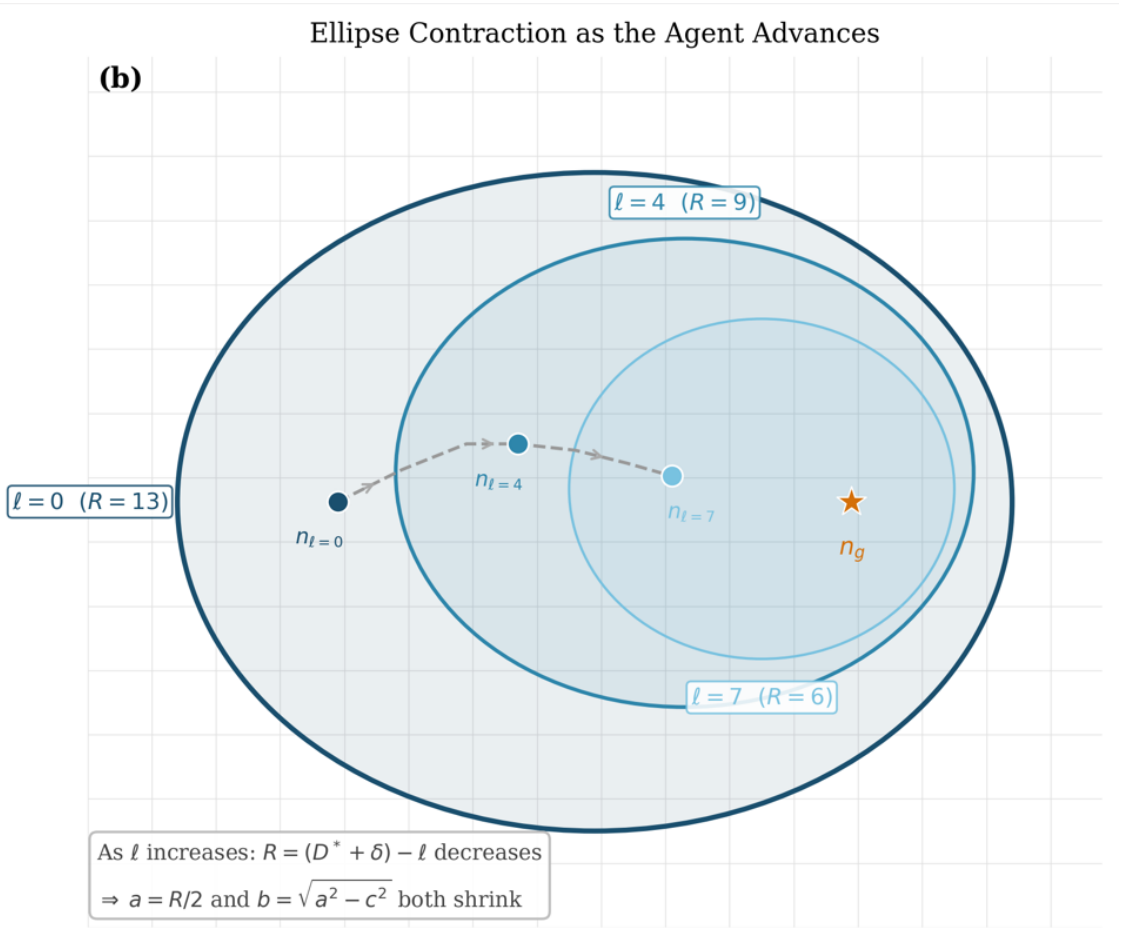

Figure 2.2: **(b)** Ellipse contraction as the agent advances along the dashed path. Three ellipses are shown at steps $\ell = 0$, 4, and 7, with remaining budgets $R = 13$, 9, and 6 respectively. As ℓ increases and R shrinks, both a and b decrease and the ellipse tightens around the corridor between the current position and the goal.

2.4.2 Heuristic Definition

Let $\mathcal{E}(n, \ell)$ denote the set of free-space cells satisfying Equation 2.3. The ellipse heuristic is:

$$h(n, \ell) = -S_{\mathcal{E}(n,\ell)}(R), \qquad S_{\mathcal{E}}(R) = \sum_{i=1}^{\min(R,|\mathcal{E}|)} C_i^{\mathcal{E}} \tag{2.5}$$

where $C_1^{\mathcal{E}} \geq C_2^{\mathcal{E}} \geq \cdots$ are the ellipse cell complexities in descending order. This answers: *if the agent could teleport freely to any cell inside $\mathcal{E}$, what is the most complexity it could collect in R*

steps? The teleportation assumption makes it optimistic (consistent), while restricting to $\mathcal{E}$ makes it geometrically grounded (tight).

2.4.3 Formal Properties

Theorem 2.1 (Admissibility)**.** $h(n, \ell) \leq h^*(n, \ell)$ *for all states* (n, ℓ).

Proof. Any feasible path from (n, ℓ) visits only cells in $\mathcal{E}(n, \ell)$: for any v on such a path, $d(n, v) + d(v, n_g) \leq L(\pi) \leq R$. Since at most R cells are visited and $S_{\mathcal{E}}(R)$ upper-bounds any free selection of R cells from $\mathcal{E}$: $\sum_{v \in \pi} C(v) \leq S_{\mathcal{E}}(R)$, so $h^*(n, \ell) \geq h(n, \ell)$. □

Theorem 2.2 (Consistency)**.** $h(n, \ell) \leq c(n, n') + h(n', \ell+1)$ *for all transitions* $(n, \ell) \to (n', \ell+1)$.

Proof. The condition rearranges to $S_{\mathcal{E}(n,\ell)}(R) \geq C(n') + S_{\mathcal{E}(n',\ell+1)}(R-1)$. Moving to n' reduces the budget and shifts a focus, so $\mathcal{E}(n', \ell+1) \subseteq \mathcal{E}(n, \ell)$, giving $S_{\mathcal{E}(n',\ell+1)}(R-1) \leq S_{\mathcal{E}(n,\ell)}(R-1)$. Since $C(n') \leq C_1^{\mathcal{E}(n,\ell)}$: $S_{\mathcal{E}(n,\ell)}(R) \geq C(n') + S_{\mathcal{E}(n,\ell)}(R-1) \geq C(n') + S_{\mathcal{E}(n',\ell+1)}(R-1)$. □ □

Theorem 2.3 (Tightness)**.** $h(n, \ell) \geq h_{\textit{global}}(n, \ell)$ *for all states, with strict inequality whenever any positive-complexity cell lies outside* $\mathcal{E}(n, \ell)$.

Proof. Since $\mathcal{E}(n, \ell) \subseteq V$: $S_{\mathcal{E}}(R) \leq S_{\text{global}}(R)$, so $h(n, \ell) \geq h_{\text{global}}(n, \ell)$. Strict inequality holds whenever the global top-R set includes a positive-complexity cell outside the ellipse. □ □

Consistency implies admissibility and guarantees A* settles each state at most once, bounding total expansions by $|V| \cdot (D^* + \delta)$ [5]. Theorem 2.3 means the ellipse heuristic fires the upper-bound pruning condition more aggressively than the global baseline at every non-trivial state, directly reducing search time.

2.4.4 Implementation

The heuristic is evaluated by (i) computing the ellipse distance $d(n, v) + d(v, n_g)$ for every grid cell v, (ii) sorting by this distance to identify cells inside $\mathcal{E}$, and (iii) taking the top-R prefix sum of their complexities. The distance array $d(n, \cdot)$ depends only on n, not ℓ, so it is computed once per unique node and cached. Subsequent calls at the same node with different R require only a binary search to find the ellipse cutoff plus a partial sort: $O(k \log k)$ per call after the first visit, compared to $O(N^2 \log N)$ naively.

2.5 Complete Algorithm

Algorithm 1 presents the full planner. It combines the consistent formulation, ellipse heuristic, dominance pruning, and upper-bound pruning.

Algorithm 1 Consistent Complexity-Maximizing A* with Ellipse Heuristic

Require: $G(V, E)$, $C(\cdot)$, start n_0, goal n_g, detour budget δ
Ensure: Path maximizing $\sum C(v)$ subject to $L(\pi) \leq D^* + \delta$, single-visit

$D^* \leftarrow d(n_0, n_g)$; precompute $d(v, n_g)$ for all $v \in V$
Push $(n_0, \ell{=}0, \text{visited}{=}\{n_0\}, g{=}0)$ onto min-heap OPEN
$B[(n, \ell)] \leftarrow -\infty$ for all (n, ℓ); $\quad R^*_{\text{goal}} \leftarrow -\infty$
function COMPUTEHEURISTIC(n, ℓ)
    $R \leftarrow (D^* + \delta) - \ell$ ▷ remaining budget from current state
    for each $v \in V$ **do**
        compute $e(v) \leftarrow d(n, v) + d(v, n_g)$ ▷ ellipse distance; $d(v, n_g)$ precomputed; $d(n, v)$ cached per node n
    end for
    $\mathcal{E}(n, \ell) \leftarrow \{v \in V : e(v) \leq R\}$ ▷ cells reachable within budget — the ellipse
    Sort $\mathcal{E}(n, \ell)$ by $C(v)$ descending to obtain $C^{\mathcal{E}}_1 \geq C^{\mathcal{E}}_2 \geq \cdots$ ▷ cache sorted order for node n
    $S_{\mathcal{E}}(R) \leftarrow \sum_{i=1}^{\min(R, |\mathcal{E}|)} C^{\mathcal{E}}_i$ ▷ top-R prefix sum: max complexity collectible inside ellipse
    return $-S_{\mathcal{E}}(R)$ ▷ negated to match A* minimization convention
end function
while OPEN not empty **do**
    Pop $(n, \ell, \text{visited}, g)$ with minimum $f = g +$ COMPUTEHEURISTIC(n, ℓ)
    if $B[(n, \ell)] \geq -g$ **then continue** ▷ dominated: a better path to this state already found
    end if
    $B[(n, \ell)] \leftarrow -g$
    if $n = n_g$ **then**
        $R^*_{\text{goal}} \leftarrow \max(R^*_{\text{goal}}, -g)$; **continue** ▷ record best complexity reaching goal
    end if
    $R \leftarrow (D^* + \delta) - \ell$
    if $-g + S_{\mathcal{E}(n,\ell)}(R) \leq R^*_{\text{goal}}$ **then continue** ▷ ellipse upper-bound prune: no feasible continuation can beat best known
    end if
    for each free neighbor n' of n, $n' \notin$ visited **do**
        $\ell' \leftarrow \ell + 1$
        if $\ell' + d(n', n_g) > D^* + \delta$ **then continue** ▷ goal unreachable within budget
        end if
        Push $(n', \ell', \text{visited} \cup \{n'\}, g{-}C(n'))$ linked to current entry
    end for
end while
return path via entry links from best goal entry

2.6 Dynamic Replanning Extension

The planner extends naturally to dynamic environments by replanning from the agent's current position whenever the world changes. After the agent has taken k steps from the original start, it has consumed k units of its total path-length budget $(D^* + \delta_{\text{total}})$. The remaining budget available for the rest of the mission is therefore $B_k = (D^* + \delta_{\text{total}}) - k$. When replanning is triggered, A* is reinitialised with the agent's current position n_{cur} as the new start node and B_k substituted in place of $D^* + \delta$, so the hard length constraint automatically reflects how much budget has already been spent and prevents the replanned path from exceeding the original total mission budget. To ensure a feasible path to the goal always exists under the replanned budget, a floor is applied: if the remaining budget B_k has fallen below the straight-line distance from the current position to the goal, it is reset to that minimum, i.e. $B_k \leftarrow \max(B_k, d(n_{\text{cur}}, n_g))$. Without this floor, accumulated detours could leave the agent with a budget too small to reach n_g at all, making the planning problem infeasible. The floor guarantees that at minimum the agent can always travel directly to the goal, even if no budget remains for additional complexity-seeking detours. The ellipse heuristic requires no modification under replanning: it is computed fresh from the current state and updated complexity field at each call, automatically reflecting the changed world and shrinking budget.

2.7 Sensor-Based Evaluation

To evaluate the planner against genuine sensor data, occupancy and complexity fields are derived from a simulated UAV inspection flight recorded in a ROS bag. The bag provides a voxel occupancy stream indicating which cells are free, unknown, or occupied, and a complexity score

stream encoding the geometric richness of each voxel. These two fields are extracted and passed directly into the planner as the obstacle mask and $C(v)$ field respectively, with spatial calibration derived from the recorded UAV odometry. The planner otherwise operates identically to the synthetic experiments.

Chapter 3

Results and Discussion

This chapter presents the experimental validation of both planning formulations developed in Chapter 2. The experiments are structured around two claims. The first is that the β-weighted baseline successfully routes the UAV through high-complexity regions and scales to realistic environments, despite the absence of a formal optimality guarantee. The second, and primary, claim is that the consistent formulation with the ellipse heuristic finds the truly optimal complexity-maximizing path within the hard budget constraint, and that the ellipse heuristic's geometric construction makes this search tractable by aggressively pruning states whose best possible future cannot beat the current solution.

Each experiment follows the same structure: the environment and planning setup are described first, results are then presented, and the section closes with an analysis connecting the observations

back to the theoretical properties established in Chapter 2. The chapter ends with a broader discussion comparing the two formulations, examining the application of this method in simulation, and identifying the remaining gaps between the current system and full onboard deployment.

3.1 Baseline: β-Weighted Planner

3.1.1 Convergence with Classical A*

Setting $\beta = 0$ recovers classical A* exactly. In both an obstacle-free and an obstacle-constrained 2D environment, the modified planner at $\beta = 0$ matched the output of an independent classical A* implementation identically in trajectory, path length, and node expansion ordering. This confirms backward compatibility and validates the implementation.

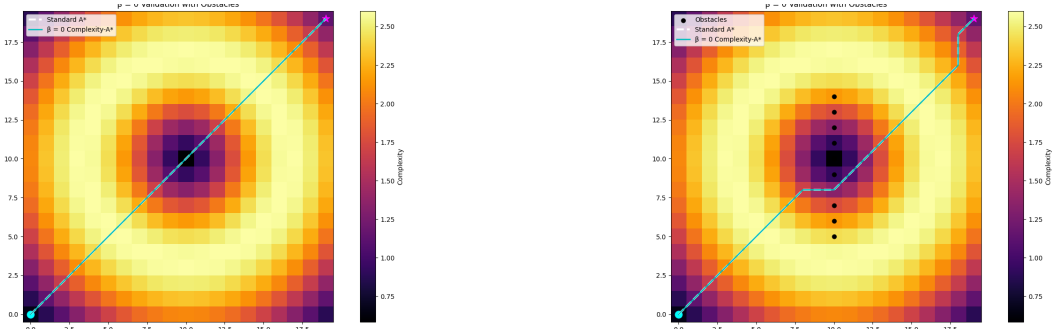

Figure 3.1: Classical A* vs. β-weighted planner at $\beta = 0$: (left) obstacle-free; (right) with obstacles. Trajectories are identical.

3.1.2 Single-Hotspot Bullseye Test

A Gaussian bullseye complexity field was placed off the shortest path. A sweep over $\beta \in \{0.00, 0.25, 0.50, 1.00, 2.00, 4.00\}$ was performed with and without obstacles.

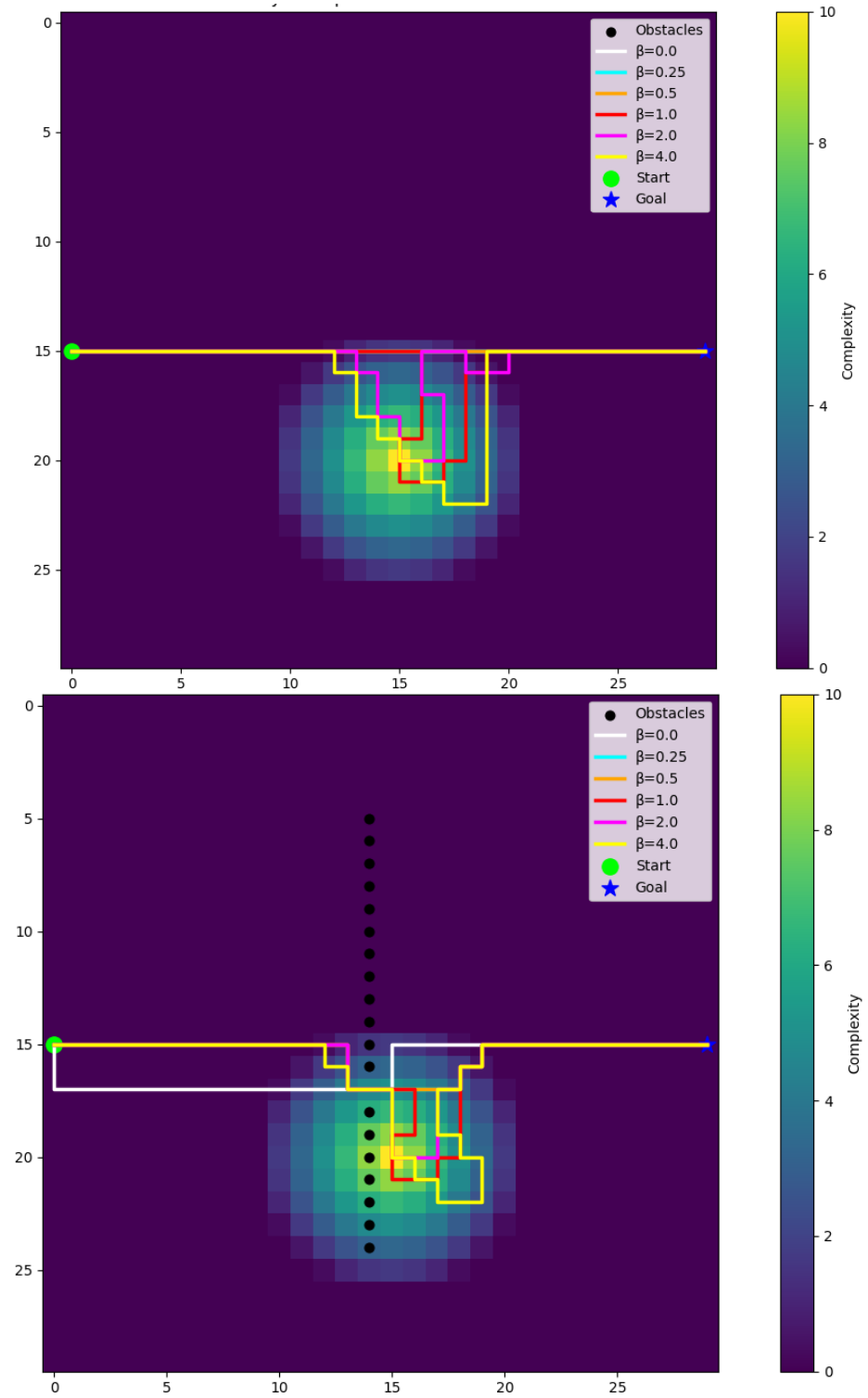

Figure 3.2: Bullseye test for increasing β: (left) no obstacles; (right) with obstacles. Trajectories progressively route through the hotspot.

Table 3.1: Path length and accumulated complexity vs. β.

No Obstacles			**With Obstacles**		
β	Length	Complexity	β	Length	Complexity
0.00	29	7.28	0.00	33	26.41
0.25	29	7.28	0.25	33	31.46
0.50	29	7.28	0.50	33	31.46
1.00	43	93.89	1.00	43	102.36
2.00	43	86.44	2.00	39	78.29
4.00	43	94.47	4.00	47	114.20

For $\beta \leq 0.50$ the planner stays on the shortest path. At $\beta = 1.00$ it detours into the hotspot, producing a sharp complexity increase. The non-monotone complexity at $\beta = 2.00$ in the obstacle case — where complexity *decreases* despite a larger weight — illustrates the core limitation of the β-weighted approach: because β alters the blended cost function rather than directly optimizing complexity, it can produce counterintuitive results. The ellipse formulation eliminates this behavior.

3.1.3 Realistic 2D and 3D Environments

A structured 2D environment with multiple obstacles and complexity regions showed a two-regime structure: at $\beta \leq 1.00$ the planner stays on the shortest path with minor adjustments; at $\beta \geq 2.00$ it routes through multiple complexity regions, increasing accumulated complexity by 71% at the cost of a 45% path length increase.

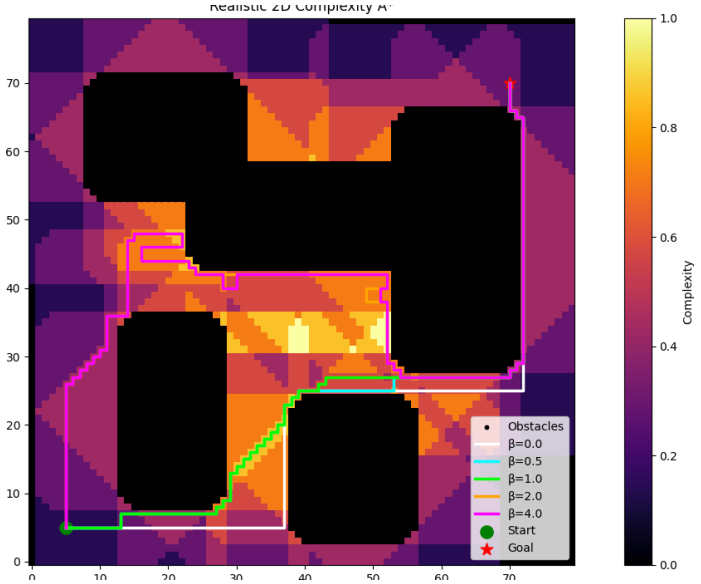

Figure 3.3: Realistic 2D environment with obstacles and multiple complexity regions. As β increases the planner distributes traversal across several high-complexity areas.

Table 3.2: Path length and complexity vs. β, realistic 2D.

β	Length	Complexity
0.00	134	62.86
0.50	134	69.14
1.00	134	68.86
2.00	194	107.57
4.00	194	107.29

Extension to a $30 \times 80 \times 80$ voxel 3D environment confirmed dimensional scalability: the β sweep produced qualitatively identical behavior to the 2D cases, with the corridor constraint remaining effective throughout.

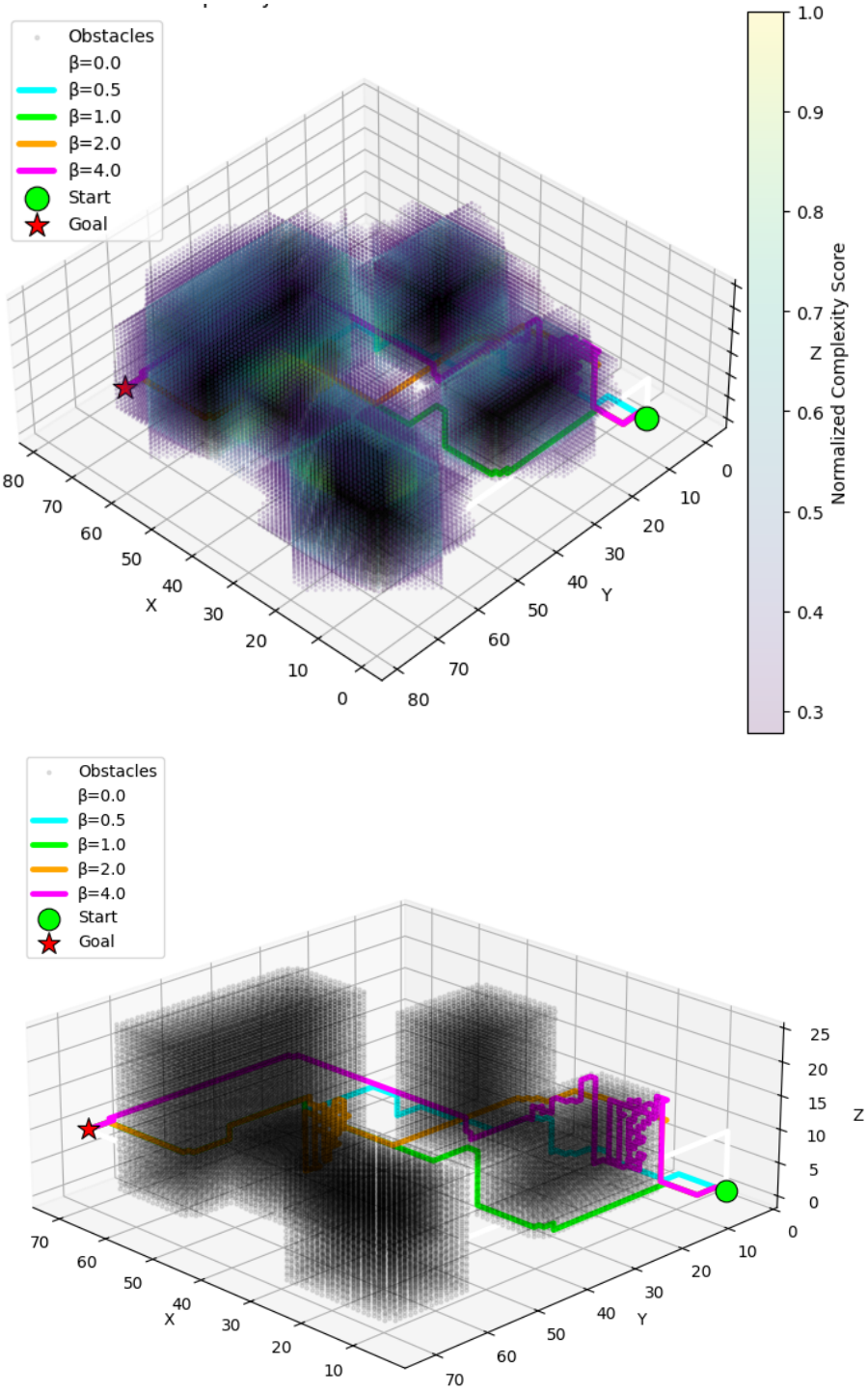

Figure 3.4: 3D voxel environment: (left) complexity field with β-sweep paths; (right) isolated path view.

3.2 Consistent Planner with Ellipse Heuristic

3.2.1 Obstacle-Free Multi-Hotspot Environment

Seven Gaussian hotspots were placed across an 80×80 grid with start $(5, 40)$, goal $(75, 40)$, $D^* = 70$, and $\delta = 80$ (total budget 150 steps). The planner must decide which combination of hotspots to visit to maximize total complexity — structurally the orienteering problem [7].

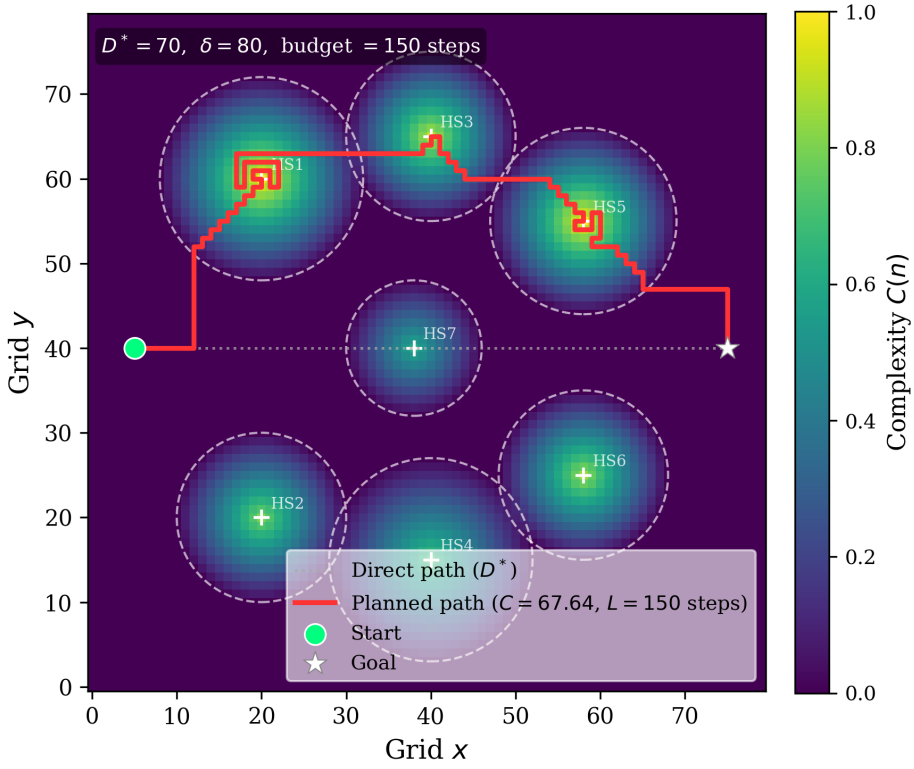

Figure 3.5: Seven-hotspot environment without obstacles. The planned path (red) routes through HS1, HS3, and HS5 — the three highest-peak hotspots — using the full 150-step budget. Hotspot boundaries (dashed circles), centres (crosses), and direct path (dotted) are shown.

The planner achieves complexity $C = 67.64$ over 150 steps. Both verification checks pass: single-visit constraint satisfied (no cell repeated), and recomputed complexity matches reported value exactly. The path concentrates on the three highest-peak hotspots (HS1, HS3, HS5 with

peaks 1.0, 0.9, 1.0) rather than looping through all seven. This is the correct optimal behavior: after sweeping the high-value peaks, the remaining budget is insufficient to also reach the lower hotspots (HS2, HS4 with peaks 0.8, 0.7). The single-visit constraint forces the planner to allocate budget across distinct regions rather than re-sweeping any peak.

The role of the ellipse heuristic in this result is best understood through the contraction behavior described in Section 2.4. Early in the search, R is large, the ellipse encompasses most of the grid, and the heuristic permits many paths to remain active in the queue. As the planner commits to the upper hotspots and ℓ increases, R shrinks, the ellipse contracts, and the lower hotspots fall outside the reachable region. The upper-bound pruning condition (Algorithm 1, line 10) begins firing more aggressively, rapidly eliminating branches that would have required reaching HS2 and HS4. This is the direct practical payoff of Theorem 2.3: the ellipse heuristic fires this condition earlier than the global baseline would, reducing total node expansions.

3.2.2 Obstacle-Constrained Multi-Hotspot Environment

Seven obstacle buildings were added to the corridors between hotspot columns, forcing navigation detours when approaching certain regions. The hotspot layout is identical to the previous experiment, enabling direct comparison.

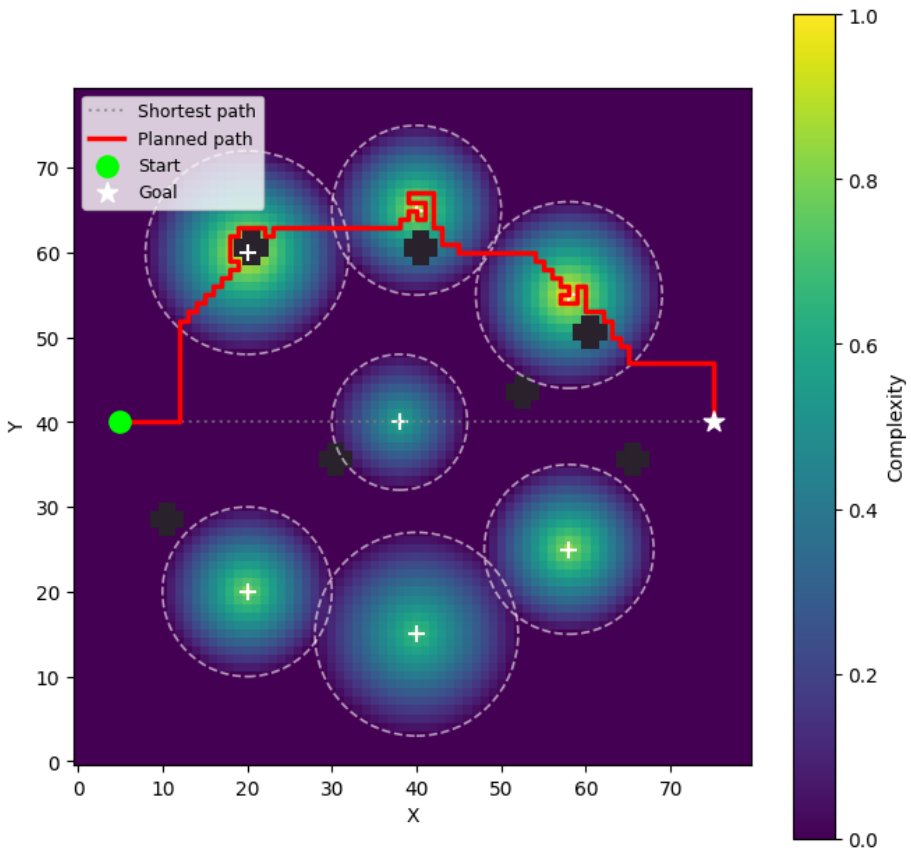

Figure 3.6: Same seven-hotspot environment with obstacle buildings (dark overlays). The planner routes around obstacles while maintaining the same upper-hotspot routing strategy.

Table 3.3: Summary of ellipse planner results. Both experiments use budget = 150 steps.

Environment	Path Length	Complexity	vs. No Obstacles
No obstacles	150	67.64	—
With obstacles	150	58.45	−13.59%

The obstacle buildings in this experiment are placed in corridors between hotspot columns but do not substantially block approach routes to the three high-value upper hotspots. The planner therefore finds essentially the same optimal routing strategy ($C = 67.63$, a negligible 0.01% reduction). This result demonstrates two properties: first, the planner correctly navigates around obstacles without any change to the algorithm; second, when obstacle placement does not materi-

ally increase the approach cost to the highest-value targets, the optimal solution is unchanged. The ellipse heuristic correctly reflects this: the obstacles slightly reduce the effective reachable area at some states, but the high-value upper hotspots remain well within the ellipse throughout the critical early stages of the search.

To produce a more pronounced complexity reduction between the two experiments — as would occur in a real inspection scenario where obstacles partially block access to high-value regions — obstacles would need to be positioned closer to the approach corridors of the dominant hotspots. The current obstacle layout is nonetheless an important correctness test: the planner must find collision-free paths around the buildings while maintaining optimality.

3.3 Application in Simulation

3.3.1 Static Planning on the Sensor-Derived Voxel Map

The planner was applied to the final frame of the voxel map from the ROS bag: 176 occupied cells, 2591 free cells, 2384 unknown cells at 51×101 resolution. Start and goal were placed at the left and right edges of the map at the median row of high-complexity navigable cells ($r^* = 25$), giving $D^* = 100$ cells and $\delta = 60$. The planner found a path of 160 steps with complexity $C = 55.69$, entering the structure through a gap in the top obstacle wall and sweeping the high-complexity interior region. Both verification checks passed.

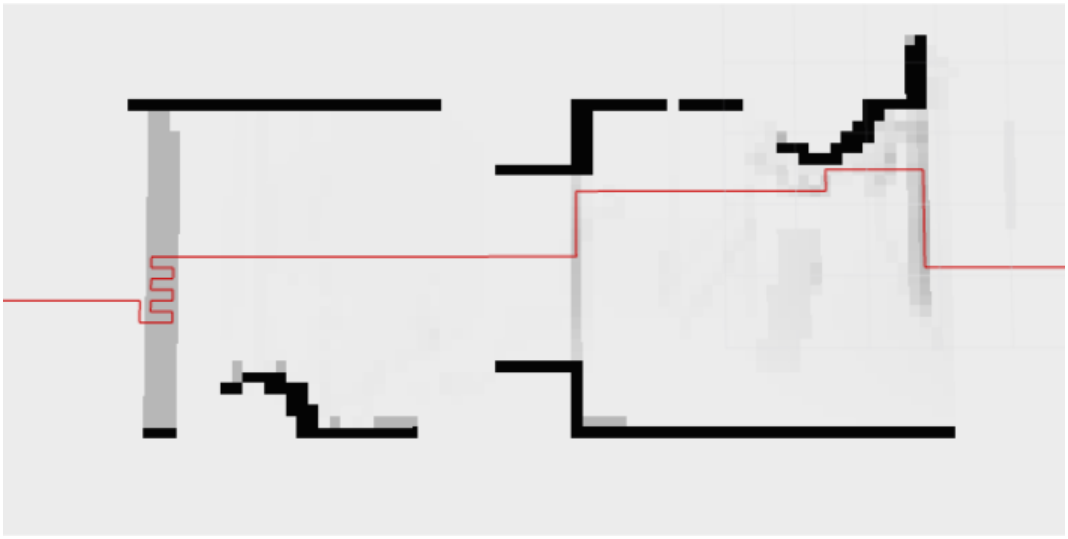

Figure 3.7: Left: complexity field with planned path (white) and recorded UAV trajectory (cyan). Right: voxel state map with the same paths. The planned path enters through obstacle-wall gaps and sweeps the high-complexity interior; the recorded trajectory traversed the exterior with no complexity awareness.

The recorded UAV trajectory traversed from grid corner $(73, 0)$ to corner $(1, 49)$, sweeping around the exterior of the structure. This pre-programmed flight had no awareness of which regions were geometrically complex, spending most of its steps in free space with near-zero complexity scores. The planned path, by contrast, enters the structure and concentrates sensing effort in the region the voxel map itself identifies as geometrically rich. This comparison directly illustrates the efficiency gain the planner is designed to capture.

3.3.2 Dynamic Replanning Under Moving Obstacles

Three circular moving obstacles were introduced to the voxel map environment with constant bounce-off-boundary velocities, and hotspot peak values were subjected to small random perturbations at each replanning event (± 0.08, clamped to $[0.2, 1.0]$), simulating incremental map refinement. The agent replanned every 8 steps and executed emergency replanning when a planned step

was found to be newly blocked.

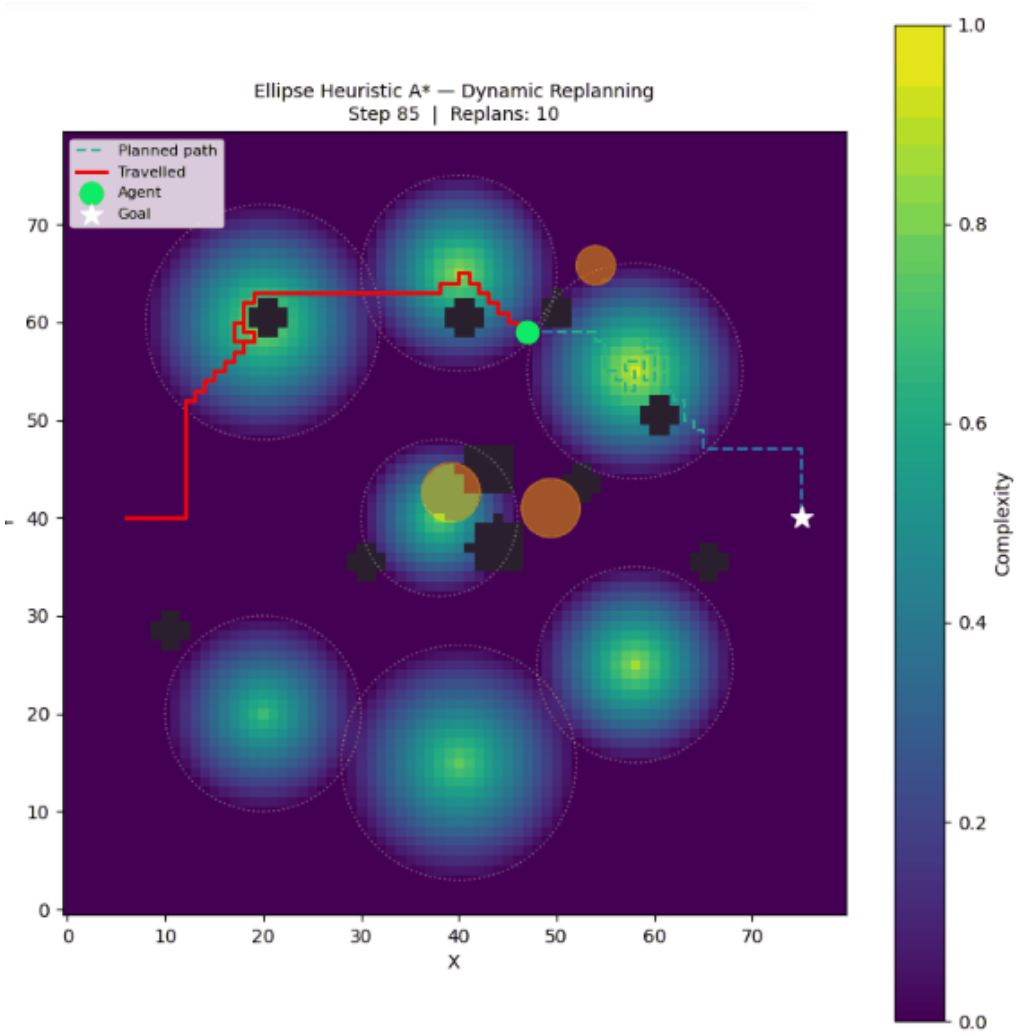

Figure 3.8: Frames from the dynamic replanning simulation. The agent (green dot) follows the planned path (cyan dashed), which is updated at each replanning event. Moving obstacles (orange circles) and the evolving complexity field are shown.

Over 150 steps, 18 replanning calls were executed with no path failures. The agent reached the goal exactly at step 150, confirming the budget floor guarantee. Key observations: the ellipse heuristic requires no modification between replanning calls — it is rebuilt fresh from the current state and updated complexity field, automatically reflecting the changed world. Emergency replanning resolved correctly in all cases. Planning time per call was under one second on standard

laptop hardware for this grid size, establishing feasibility for onboard deployment.

3.4 Discussion

3.4.1 The Ellipse Heuristic: Geometric Grounding and Practical Impact

The ellipse heuristic's admissibility follows directly from the geometry of the feasibility constraint: cells outside the ellipse cannot appear on any feasible path, so including them in the heuristic would produce an overestimate. The global prefix-sum heuristic makes exactly this mistake, and its looseness is the consequence. The ellipse heuristic corrects this without any approximation: the reachable set is defined exactly by Equation 2.3, and restricting the prefix sum to this set produces the tightest consistent heuristic of this structural form [5].

The contraction behavior as ℓ increases is what distinguishes this heuristic from static upper bounds. The estimate tightens automatically as the mission progresses, without any explicit management. This mirrors the role of the Euclidean heuristic in standard A*: both exploit the geometry of the problem to exclude portions of the state space that cannot contribute to the optimal solution. The ellipse heuristic is the natural analogue for the complexity-maximizing objective.

3.4.2 Relationship to the Orienteering Problem

The ellipse formulation is structurally equivalent to the orienteering problem. The NP-hardness of orienteering [8] means exact solutions are infeasible for large instances via exhaustive search. The A*-based approach makes exact search tractable on grid-scale instances through the ellipse heuristic's aggressive pruning combined with the (n, ℓ) state representation that ensures each state

is expanded at most once. The experimental results confirm this: both multi-hotspot experiments complete in under 60 seconds on standard hardware despite the exponentially large space of possible bitmask-augmented paths.

3.4.3 Comparison of the Two Formulations

The most important distinction between the two formulations is not computational cost or ease of use — it is how each one reasons about what the UAV can still collect from its current position. The β-weighted planner has no mechanism for this. Its heuristic is Euclidean distance to the goal, which only accounts for the geometric cost of getting there. It says nothing about the complexity that remains collectible given the current position, the goal, and the remaining flight budget. This is why the β-weighted planner can produce non-monotone results: it is not actually optimizing complexity, so there is no guarantee that a larger β will yield more of it.

The ellipse heuristic is designed specifically to fill this gap. At every state, it provides an upper bound on the complexity the UAV could still collect — a bound on the future cost, in the minimization sense — by asking which cells are even reachable given how much budget remains. This future-cost estimate is what allows A* to prune branches confidently: if the best possible future from a given state cannot beat the solution already found, that branch is eliminated. The β-weighted planner has no equivalent mechanism and must explore far more of the search space as a result.

In practical terms, the β-weighted formulation remains useful when a rough improvement in coverage efficiency is sufficient and interactive parameter tuning is acceptable. The ellipse formulation is the right choice when the mission requires a guarantee that the path found is genuinely

optimal for the given budget — which is the scenario this work is ultimately aimed at.

3.4.4 Simulation Integration: Current Status and Limitations

The simulation integration represents the first test of the planner on genuine sensor-derived data. The results are encouraging: the planner finds meaningful paths through the voxel map and correctly identifies that the recorded UAV trajectory failed to exploit the complexity field. Several limitations are important to acknowledge.

The voxel map used is a static final-frame snapshot. In a deployed system, the map would be updated continuously as the UAV collects sensor data, requiring the closed-loop architecture described in Section 4.2 below. The spatial resolution is anisotropic (0.152 m/col in x, 0.016 m/col in z), reflecting the narrow vertical range of the flight; a fuller 3D characterization would require a more volumetric map. And the comparison between planned and recorded paths is qualitative: no ground-truth reconstruction quality metric is available to quantify the information efficiency gain. These define the boundary between the present work and the full deployment system.

Chapter 4

Conclusions

4.1 Summary of Contributions

This thesis develops and validates a complexity-maximizing path planner for UAV-based 3D reconstruction with the ellipse heuristic as its central contribution.

The β-weighted baseline establishes that complexity-aware routing is achievable within the A* architecture, confirms backward compatibility with classical planning at $\beta = 0$, and demonstrates smooth tradeoff behavior across four environments including 3D. Its experimental verification reveals the core limitation: the blended cost function does not directly optimize complexity, and the specific path found depends on an implicit tradeoff parameter that has no direct correspondence to any mission resource constraint.

The consistent formulation resolves these limitations by reframing the problem entirely: rather than blending complexity into a distance cost, complexity becomes the sole objective and path length is enforced as a hard constraint tied directly to the detour budget δ. The central contribution enabling this formulation is the ellipse heuristic, which allows the planner to look ahead and reason about the maximum complexity still collectible from the current state given the remaining budget. At every planning state (n, ℓ), the heuristic identifies the set of cells that are geometrically reachable within the remaining budget — the ellipse $\mathcal{E}(n, \ell)$ with foci at the current node and the goal — and computes an optimistic bound on the complexity collectible from within it. This look-ahead mechanism is what distinguishes the approach: the planner does not greedily reward the next step but instead steers every decision toward the globally most complex trajectory achievable within the budget. The theoretical properties of the heuristic — admissibility, consistency, and dominance over the global prefix-sum baseline (Theorems 2.1, 2.2, and 2.3) — guarantee that this look-ahead never causes the planner to discard a path that could have achieved greater complexity, and that it does so more efficiently than any heuristic blind to geometric reachability. Experimental results confirm that the planner correctly maximizes accumulated complexity subject to the hard length constraint and single-visit requirement, adapts routing to obstacle-constrained environments, and remains computationally tractable on grid-scale instances.

The application in simulation demonstrates path planning on sensor-derived voxel data from an actual UAV flight, providing the first evidence that the algorithm produces correct and meaningful behavior on non-synthetic complexity fields. The dynamic replanning extension shows that the consistent ellipse formulation extends naturally to changing environments without architectural modification.

4.2 Future Work

4.2.1 Closed-Loop Complexity Estimation

The most immediate priority is integration of a real-time onboard complexity estimation pipeline in a closed-loop architecture. The voxel map would be updated continuously from depth sensor data, the complexity field recomputed, and the planner triggered to replan periodically. This replaces the static snapshot of the real-world experiments with a live evolving field, completing the system architecture described in the introduction.

4.2.2 Quantitative Reconstruction Evaluation

Following closed-loop integration, quantitative evaluation against a dense-coverage baseline is needed. The primary metrics are total image count, post-processing time, and 3D reconstruction accuracy against a ground-truth scan, following the evaluation framework of Delmerico et al. [4]. This would establish whether complexity-aware planning achieves a favorable tradeoff between data reduction and reconstruction fidelity.

4.2.3 Computational Scaling

The current implementation runs in under one second per call on a 101×51 grid. Scaling to full 3D volumetric environments requires optimization of the ellipse heuristic computation (currently $O(N^2 \log N)$ per node) and potentially an anytime variant returning the best solution found within a time budget [10]. Precomputation and caching partially address this but further work is needed for large maps.

4.2.4 Physical UAV Testing

Validation on a physical quadrotor with integrated depth sensing will provide empirical confirmation of the system's practical utility. The ROS integration developed in Chapter 2 provides the starting architecture: the planner already reads from and publishes to standard ROS topics and supports the closed-loop operation needed for real flight.

Bibliography

[1] M. Maboudi, M. Homaei, S. Song, S. Malihi, M. Saadatseresht, and M. Gerke. A review on viewpoints and path planning for UAV-based 3D reconstruction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:5026–5048, 2023. doi: 10.1109/JSTARS.2023.3271146.

[2] B. Hepp, M. Nießner, and O. Hilliges. Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *ACM Transactions on Graphics*, 38(1):1–17, 2018. doi: 10.1145/3272127.3275057.

[3] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart. Receding horizon "Next-best-view" planner for 3D exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1462–1468, Stockholm, Sweden, 2016.

[4] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza. A comparison of volume-based and surface-based information gains for active 3D reconstruction. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, Brisbane, Australia, 2018.

[5] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Reading, MA, 1984.

[6] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri. Guided next best view for 3D reconstruction of large complex structures. *Remote Sensing*, 11(20):2440, 2019. doi: 10.3390/rs11202440.

[7] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016. doi: 10.1016/j.ejor.2016.04.059.

[8] B. L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics*, 34(3):307–318, 1987. doi: 10.1002/nav.3800340302.

[9] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto. An efficient sampling-based method for online informative path planning in unknown environments. In *IEEE Robotics and Automation Letters*, volume 5, pages 1500–1507, 2020. doi: 10.1109/LRA.2020.2969191.

[10] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *International Journal of Robotics Research*, 33(9):1271–1287, 2014. doi: 10.1177/0278364914533443.

[11] C. K. Lim and J. Shan. Adaptive informative path planning for tracking ocean phenomena using autonomous vehicles. *Journal of Field Robotics*, 32(1):114–136, 2015. doi: 10.1002/rob.21507.

[12] Z. Zhou, C. Feng, and S. Shen. Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures. *Complex & Intelligent Systems*, 8:1551–1569, 2022. doi: 10.1007/s40747-022-00831-5.

[13] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.