www.cad-journal.net
Computer-Aided Design AND APPLICATIONS

Risk-Aware Path-Planning for Object Transport with Pose Transitions in Point-Cloud-Based Environments

Linxuan He[1] and Hiroaki Date[2]

[1]Hokkaido University, karinken@163.com
[2]Hokkaido University, hdate@ist.hokudai.ac.jp

Corresponding author: Linxuan He, karinken@163.com

Abstract. In narrow and obstacle-dense indoor environments, transporting large equipment often requires pose transitions of a transport unit composed of the object and the worker. Therefore, translation-only and fixed-pose planning are insufficient because they may fail to find a collision-free path. Even when a collision-free transport path is found, the transport unit may pass too close to the surrounding obstacles. In such cases, the path has insufficient clearance for safe transport. This paper addresses fixed-height planar transport with yaw changes in measured point-cloud environments. First, the measured point clouds of the environment and the transport unit are converted into a unified voxel representation. Second, for each discretized yaw, a pose-dependent configuration-space (C-space) obstacle voxel set is precomputed using an inverse offset. Third, a pose-dependent risk field is defined by combining a binary collision term for collision checking with a clearance-based penalty to evaluate the clearance from surrounding obstacles. Finally, an extended A* search is performed in a discrete search space for the planar position and yaw, and the intermediate rotation of each yaw change is explicitly validated. A yaw change is accepted only when the entire rotation remains collision-free. Experiments in two real scanned environments and one synthetic environment with a narrow passage showed that the proposed method reduces the high-risk rate and improves the minimum clearance, while maintaining a path length comparable to that of translation-only planning. In addition, in the most constrained real scanned environment, the proposed method found a collision-free transport path, whereas the fixed-yaw methods failed. These results show that the explicit handling of yaw change and clearance is effective for transport planning in measured point-cloud environments.

Keywords: Point clouds, risk-aware path-planning, transportation, pose transitions
DOI: https://doi.org/10.14733/cadaps.2027.203-217

1 INTRODUCTION

In indoor industrial environments, the transport of large equipment and precision machinery is often planned by combining blueprint-based checks with onsite judgments based on experience. In narrow and obstacle-dense spaces, whether a collision-free transport path exists may depend on yaw during motion. Therefore, path-planning cannot be reduced to translation-only or fixed-yaw motions. Even if the transport unit is collision-free at both the initial and target yaw angles, collisions may still occur during the intermediate rotation. Therefore, the yaw change must be verified throughout the complete yaw transition, not only at the initial and target yaw angles. Collision-free motion alone is insufficient for safe transportation. A collision-free path may still leave insufficient clearance. This requirement is consistent with the machinery safety distance requirements [3,5]. Therefore, transport planning in narrow indoor industrial environments should consider not only collision avoidance, but also clearance from surrounding obstacles. In such environments, maintaining sufficient clearance may require yaw adjustments during transport, and a feasible transport path may not be found if yaw changes are not considered.

Measured point clouds are increasingly being used in engineering applications. Advances in terrestrial laser scanning, mobile or handheld laser scanning, RGB-D-based three-dimensional reconstruction [18,25], and photogrammetry [26] have made it feasible to obtain high-precision geometric information of real environments. Such measured point clouds can be represented and used in path planning [6,11]. However, these studies did not directly address the present transport problem. They did not jointly consider yaw change, clearance from surrounding obstacles, or the occupied shape of the transport unit under different yaw configurations. Therefore, it is difficult to apply them directly to transport path-planning for large equipment in narrow indoor industrial environments. In addition, environmental point clouds may be large; therefore, the planning method must also maintain a practical computation time.

Based on these observations, the path-planning method required in this study should satisfy three requirements. First, it should directly use the measured point clouds of both the environment and the transport unit. Second, it should explicitly handle yaw changes and clearance from surrounding obstacles, rather than relying only on binary collision checking. Third, the computation time should be practical for large voxelized environments.

To satisfy these requirements, this study proposes a risk-aware path-planning framework for object transport in measured point-cloud environments. The details of the framework are presented in Section 3.

The main contributions of this study are as follows.

- A transport-path-planning framework is proposed that directly uses the measured point clouds of both the environment and the transport unit.
- A pose-dependent feasibility model is introduced such that each candidate yaw transition is validated not only at discrete poses but also throughout the intermediate rotation.
- A clearance-based evaluation is incorporated into path-planning so that the planner can distinguish between collision-free paths and paths that maintain sufficient clearance from surrounding obstacles.
- Quantitative experiments in two real scanned environments and one synthetic narrow-passage environment show that the explicit consideration of yaw change and clearance improves both the feasibility and safety margin.

The remainder of this paper is organized as follows: Section 2 reviews the related work on measured point-cloud representations for planning, pose-dependent collision checking, and clearance-aware path-planning. Section 3 describes the proposed method. Section 4 presents experimental results. Finally, Section 5 concludes the paper.

2 RELATED WORK

This section reviews related work from three perspectives corresponding to the requirements stated in Section 1. Section 2.1 reviews how the measured point clouds are obtained and represented for planning. Section 2.2 reviews how the collision-free feasibility is modeled when yaw is considered.

Section 2.3 reviews how clearance and risk-related cost terms are incorporated into path-planning. From these three perspectives, the purpose of this review is to identify the aspects of the present transport problem addressed by the existing studies and the aspects that remain unresolved.

2.1 Measured Point-Cloud Representation for Path-Planning

Transport path-planning in an actual indoor industrial environment requires a representation based on the measured geometry. Malhan and Gupta [16] studied planning algorithms for acquiring high-fidelity point clouds for accurate and fast three-dimensional reconstruction. Newcombe et al. [18] proposed KinectFusion, which reconstructs dense surfaces from RGB-D data in real time. Curless and Levoy [4] integrated multiple range images into a volumetric representation, and Whelan et al. [25] extended volumetric fusion to large-scale RGB-D SLAM. These studies demonstrated how real environments can be measured, reconstructed, and represented for subsequent computations.

For path-planning, measured point clouds are often converted into discrete spatial representations. Elfes [6] introduced occupancy grids for mobile robot perception and navigation. Felzenszwalb and Huttenlocher [7] proposed a distance transform for sampled functions that enabled the efficient computation of the distance from each grid cell to the nearest occupied cell. Hornung et al. [11] proposed the OctoMap, an octree-based three-dimensional mapping framework for large environments. These studies are useful because they convert measured data into representations that can be directly used in planning.

These studies provide a basis for representing real environments using measured data and discrete spatial representations, such as occupancy grids and voxel-based models. In this study, we adopted a voxel-based representation to enable efficient geometric computation during path planning. In contrast to previous studies that focused primarily on representing the environment, we represented the transport unit as an occupancy voxel model under varying yaw configurations. This enables a unified and consistent evaluation of both the environment and the transport unit geometry during planning.

2.2 Pose-dependent Feasibility Modeling and Collision Detection

In the current transport problem, the planner must determine whether the transport unit is collision-free at a given position and yaw. Niwa and Masuda [19] studied collision detection in engineering plants based on large-scale point clouds. Pan et al. [21] proposed Flexible Collision Library (FCL), which is a general-purpose library for collision and proximity queries. These studies demonstrate that collision checking can be performed for complex shapes in large geometric environments. However, path-planning methods based on on-the-fly collision checking incur high computational costs. When multiple candidate paths must be explored, or fine adjustments of paths are required, the following configuration-space-based approach is more suitable:

From the viewpoint of motion planning, Latombe [12] described configuration-space (C-space) obstacles as a standard framework for pose-dependent feasibility. Li and McMains [13] studied the voxelized Minkowski sum computation on a GPU, and Varadhan and Manocha [24] studied the Minkowski-sum approximation of polyhedral models. These studies show how pose-dependent collision regions can be constructed from the object and environment shapes.

These studies provide the basis for pose-dependent collision checking using configuration-space representations. However, these methods typically only evaluate the feasibility of discrete poses. In the current transport problem, feasibility must be ensured not only in individual yaw configurations, but also during transitions between them. A candidate yaw change may be collision-free at both the initial and target yaw configurations, while the intermediate rotation still intersects the environment. Therefore, this study explicitly verifies the collision-free conditions throughout the continuous yaw transition, enabling a reliable evaluation of the pose-dependent feasibility during path planning.

2.3 Path-Planning Based on Clearance and Risk

For the present transport problem, collision checking alone is insufficient because the path must also maintain clearance from surrounding obstacles. Felzenszwalb and Huttenlocher [7] proposed a distance transform that can be used to compute the distance between each grid cell and the nearest obstacle. Fox et al. [9] used obstacle proximity in the Dynamic Window Approach for local obstacle avoidance. A key advantage of this study is that it incorporates obstacle proximity into path-planning rather than considering only collision avoidance.

For search-based path-planning, Hart et al. [10] established A* as the standard shortest-path search method. The advantage of A* is that it provides a standard framework for path search. Likhachev et al. [14] proposed ARA*, which can quickly determine an initial path and improve it as the search continues. The advantage of this method is that it shows how the search efficiency can be improved without considering the first solution as a final step. Nordström et al. [20] studied risk-aware A*-based planning in a time-dependent distributed multi-agent setting. The advantage of this work is that it showed that risk-related cost terms can be incorporated into the search process. Liu et al. [15] reviewed path-planning methods for mobile robots and summarized widely used planning frameworks. The advantage of this review is that it places several planning methods in a common context.

These studies provided a basis for incorporating clearance- and risk-related cost terms into path-planning. However, they did not directly address transport path-planning, in which yaw change and clearance must be considered together. Therefore, this study combines clearance-based evaluation with yaw-dependent feasibility in a path-planning framework.

3 METHOD

3.1 Method Framework and Problem Formulation

Figure 1 illustrates the workflow of the proposed method. This method takes measured 3D point clouds of both the environment and the transport unit as input. After preprocessing and voxelization, pose-dependent configuration-space (C-space) obstacles are generated for the discretized yaw poses using an inverse offset. Based on these C-space obstacles, a risk field is constructed to represent both hard collision constraints and clearance-related safety margins. Finally, an extended A* search is performed in a discrete position–yaw state space, together with the explicit validation of intermediate rotations during pose transitions. Before describing each component of the method in detail, we first summarize the planning problem addressed in this study. The proposed workflow was designed such that most geometry-dependent computations were performed during preprocessing. Specifically, ground removal, voxelization, transport-unit voxel generation for discretized yaws, and pose-dependent C-space obstacle generation were performed before the search. Consequently, the online search stage relies mainly on voxel occupancy queries, risk-field queries, and cost updates. This design shifts most computational overhead to preprocessing, and enables an efficient path search in large-scale point-cloud environments.

The proposed formulation is based on the following assumptions and scope. This study focuses on fixed-height planar transport on approximately flat indoor industrial floors. This scope is consistent with industrial floor design, construction, and acceptance practices in which floor flatness, levelness, grade, elevation, and surface regularity are commonly specified or measured. For example, the American Concrete Institute guide ACI 302.1R-15 addresses concrete floor and slab construction for industrial, commercial, and institutional buildings, and outlines floor flatness and floor levelness requirements and measurements [1]. The ASTM International Standard E1155/E1155M-23 provides a standard test method for determining floor flatness and floor levelness numbers [2]. In Japan, the Public Building Standard Specifications for Architectural Works issued by the Ministry of Land, Infrastructure, Transport, and Tourism define the standard values for the flatness of finished concrete surfaces [17]. Therefore, yaw was considered as the dominant pose variable during transport, whereas pitch and roll were not included as active planning variables in the present formulation. In this approximately flat-floor setting, adding active pitch and roll is unlikely to substantially change the planar transport path; however, it would increase the dimensionality of the search space and the number of pose-dependent C-space obstacle sets.

Problem Formulation: The planning problem addressed in this study is defined as follows.

- **Input:** Measured 3D point cloud of the environment P_E; measured 3D point cloud of the transport unit P_O; discretized yaw set Θ; and fixed transport height layer k_{fix} for planar transport; start configuration $\mathrm{x}_s = (\mathbf{v}_s, \theta_s)$; goal configuration $\mathrm{x}_g = (\mathbf{v}_g, \theta_g)$.
- **State:** $\mathrm{x} = (\mathbf{v}, \theta)$, where $\mathbf{v}$ denotes the voxel position on the fixed-height layer and $\theta \in \Theta$ denotes the discretized yaw.
- **Actions:** 8-neighborhood translations in the xy plane, each optionally accompanied by a local yaw transition at the candidate successor state.
- **Constraints:** Each accepted state must be collision-free, and each accepted yaw transition must remain collision-free throughout the intermediate rotation.
- **Output:** A collision-free sequence of states connecting the start and goal configurations while minimizing the accumulated path cost under the above constraints.

The following subsections detail the main components of the proposed framework, including preprocessing, pose-dependent C-space obstacle generation, risk field construction, and extended A* search.

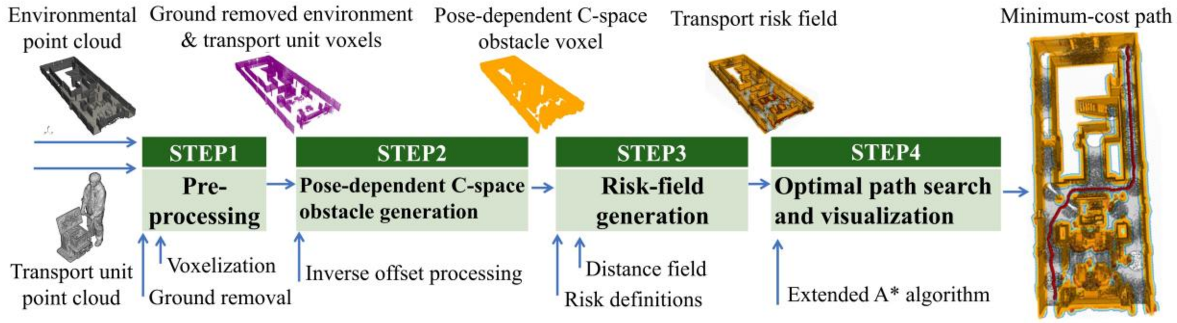

Figure 1: Workflow of the proposed method.

3.2 Voxel Generation and Unified Representation Based on Measured Point Cloud

First, we preprocessed and voxelized the environmental point cloud P_E and converted it into a regular voxel grid with a voxel size of s. A voxel is labeled occupied if it contains at least one point after preprocessing, and the set of all occupied voxels is denoted by V_E. Because the target scenario was planar transport on flat ground, the ground points were removed before voxelization. Specifically, the ground plane was detected using RANSAC-based plane fitting [8,22], and the points classified as belonging to that plane were removed. After ground removal, the remaining points were voxelized to obtain the environment occupancy voxel set V_E for subsequent path-planning.

The point cloud of the transport unit P_O represents the transported object and the worker as a whole. To maintain a consistent discrete representation with the environment, this study also voxelizes the transport unit. Because the path-planning process must account for the geometric occupancy of the transport unit at different yaw angles, a corresponding set of occupancy voxels V_O^{θ} is generated for each discrete yaw $\theta \in \Theta$.

For planar transport, the translational motion is restricted to the fixed-height layer k_{fix} introduced in Section 3.1. In this study, k_{fix} was set to the voxel layer at the midpoint of the transport-unit height after voxelization.

The unified voxel representation serves two purposes. First, it allows the environment and the transport unit to be represented within the same discrete geometric framework, thereby facilitating the subsequent pose-dependent configuration-space construction and risk assessment. Second, it allows collision detection, risk field queries, and search expansions to be performed on a voxel grid, providing a foundation for efficient planning in large-scale point-cloud environments.

3.3 Generation of Pose-dependent Configuration Space Obstacle Voxels Based on Inverse Offset

To efficiently determine whether the transport unit will collide with the environment at a given position and yaw during the path search, we precompute a configuration-space obstacle voxel set for each discretized yaw in Θ. During the search phase, collision checking can be reduced to voxel occupancy queries instead of repeatedly performing geometric interference calculations for every candidate state.

For a given discrete yaw θ, we compute the corresponding pose-dependent C-space obstacle set $V_E^{(\theta,Offset)}$ using an inverse offset. Here V_E represents the environmental occupancy voxel set, V_O^{θ} represents the occupancy voxel set of the transport unit at yaw θ, $(V_O^{\theta})^{-1}$ represents the inverse reflection set of V_O^{θ}, and $\oplus$ denotes the Minkowski sum. The resulting set is defined by Eq. (3.1).

$$V_E^{(\theta,Offset)} = V_E \oplus (V_O^{\theta})^{-1} \tag{3.1}$$

Through Eq. (3.1), the original environment occupancy voxel set V_E is transformed into the configuration-space obstacle voxel set $V_E^{(\theta,Offset)}$, corresponding to the yaw θ. When the transport unit is placed at a position voxel $\mathbf{v}$ with a pose θ, if $\mathbf{v}$ falls within this set, the current configuration collides with the environment; otherwise, the configuration is collision-free. Therefore, in any state of $\mathrm{x} = (\mathbf{v},\theta)$, collision detection can be simplified to check whether the position voxel $\mathbf{v}$ belongs to $V_E^{(\theta,Offset)}$. This occupancy-query formulation provides a unified geometric foundation for subsequent risk-field construction and path search.

3.4 Construction of the Risk Field

The risk field at pose θ consists of two components: the binary collision term $R_c^{\theta}(\mathbf{v})$ is set to 1 if the voxel $\mathbf{v}$ lies inside $V_E^{(\theta,Offset)}$ (i.e., a colliding configuration), and 0 otherwise. This reduces collision checking to a simple occupancy query, thereby avoiding expensive geometric interference tests. The clearance-based risk $R_d^{\theta}(\mathbf{v})$ quantifies clearance as a continuous safety margin. Let $d^{\theta}(\mathbf{v})$ denote the Euclidean distance from the voxel $\mathbf{v}$ to the nearest occupied voxel in the pose-dependent C-space obstacle set $V_E^{(\theta,Offset)}$. The risk field $R_d^{\theta}(\mathbf{v})$ is defined as follows:

$$R_d^{\theta}(\mathbf{v}) = max(0, 1 - d^{\theta}(\mathbf{v}) / d_{\max}) \tag{3.2}$$

This definition yields $R_d^{\theta}(\mathbf{v}) = 1$ at zero clearance (on the obstacle surface), which decreases monotonically as the clearance increases and becomes zero once the distance exceeds $d_{\max}$. We set $d_{\max} = 1$ m, rounded from the recommended human working-reach value of 1.07 m [23]

for practical application. Overall, the risk field provides a unified input that combines hard collision constraints $R_c^{\theta}(\mathbf{v})$ with a soft clearance-based risk $R_d^{\theta}(\mathbf{v})$, which is later queried in the state space $(\mathbf{v},\theta)$ during a risk-aware path search.

3.5 Extended A* Search with Pose Transition Validation

We perform an extended A* search in the discrete state space introduced in Section 3.1. Each expansion considers two types of actions: planar translations to 8-neighborhood voxels and pose transitions between neighboring discretized poses. For each candidate successor state x' generated from the current state, the evaluation function is defined by Eq. (3.3).

$$F(x') = H(x') + G(x') \tag{3.3}$$

Here, $H(x')$ denotes a weighted heuristic term defined on the 8-neighborhood voxel grid and excludes pose-transition costs, $G(x)$ is the accumulated cost from the start to the current state x, and $G(x')$ is the accumulated cost from the start to the candidate successor state x'. The accumulated cost is updated using Eqs. (3.4) and (3.5):

$$G(x) = 0 \ (x = x_0) \tag{3.4}$$

$$G(x') = G(x) + \Delta g(x, x') \tag{3.5}$$

where $\Delta g(x, x')$ is the single-step cost. Because collision states are excluded by the binary collision term $R_c^{\theta}(\mathbf{v})$ and the pose-transition feasibility constraints described below, we use only clearance-based risk in the cost, defined as

$$\Delta g(x, x') = w_l s \sqrt{(\Delta i)^2 + (\Delta j)^2} + w_d R_d^{\theta'}(\mathrm{v}') \tag{3.6}$$

where w_l and w_d are weighting factors. Here, if $\mathrm{v} = (i, j, k_{fix})$ and $\mathrm{v}' = (i', j', k_{fix})$, then $\Delta i = i' - i, \Delta j = j' - j$, where s is the voxel size. To keep the branching factor bounded while maintaining safety, we adapt the action set based on the clearance-based risk evaluated at the candidate next state x'. If the clearance-based risk at the candidate next voxel $\mathbf{v}'$, evaluated under the discretized yaw aligned with the translational direction from $\mathbf{v}$ to $\mathbf{v}'$, is zero, the successor is accepted and the next yaw is set to that direction-aligned yaw without additional local pose search. If the clearance-based risk at the candidate next voxel $\mathbf{v}'$, evaluated under the discretized yaw aligned with the translational direction from $\mathbf{v}$ to $\mathbf{v}'$, is greater than zero, local pose transitions are additionally allowed so that the transport unit can adjust its yaw when passing through narrow sections. When pose transitions are enabled, we select the next yaw θ' by minimizing the clearance-based risk at the next position $\mathbf{v}'$ as follows.

$$\theta' = \arg\min_{\vartheta \in A_c(x, \mathbf{v}')} R_d^{\vartheta}(\mathbf{v}') \tag{3.7}$$

Here, ϑ is a feasible candidate yaw. The set $A_c(x, \mathbf{v}')$ restricts pose transitions to candidates that remain collision-free during the entire rotation and that preserve next-step feasibility in the search. Therefore, $A_c(x, \mathbf{v}')$ is determined by collision and rotation-feasibility constraints. Here, the intended translational direction means the planar translation direction from the current voxel $\mathbf{v}$ to the candidate next voxel $\mathbf{v}'$, before any local yaw adjustment is applied. Among feasible candidates, directional consistency with the intended translational direction is used as a secondary tie-breaking criterion. Specifically,

$$A_c(x, \mathbf{v}') = \{\vartheta \in \{\theta, \theta \pm 10^{\circ}\} \mid R_c^{(\phi_m)}(\mathbf{v}) = 0, R_c^{(\phi_m)}(\mathbf{v}') = 0, \forall \phi_m \in \Phi(\theta \to \vartheta, M)\} \tag{3.8}$$

For a candidate pose transition from $\mathrm{x} = (\mathbf{v}, \theta)$ to $x' = (\mathbf{v}', \vartheta)$ the yaw change is discretized into M intermediate yaws $\phi_1, \dots, \phi_M$ as in Eq. (3.8). For each ϕ_m, occupancy is checked at both the current voxel $\mathbf{v}$ and the next candidate voxel $\mathbf{v}'$ by occupancy-based collision tests on the voxel grid. The transition is accepted only if both $R_c^{(\phi_m)}(\mathbf{v}) = 0$ and $R_c^{(\phi_m)}(\mathbf{v}') = 0$ hold true for all the $\phi_m \in \Phi(\theta \to \vartheta, M)$. To reduce repeated checks during A* expansions, the feasibility result for $(\mathbf{v}, \mathbf{v}', \theta, \vartheta)$ is stored and reused whenever the same transition is evaluated again.

4 EXPERIMENTS

4.1 Experimental Setup

The proposed method was evaluated in three environments. Figure 2 shows the datasets used for evaluation: Figure 2 (a) shows the transport unit, and Figure 2 (b), (c), and (d) show Environments 1, 2, and 3. The transport unit, consisting of the transported object and the worker, was digitized using smartphone-based LiDAR scanning with the Scaniverse app and then manually denoised; its overall dimensions are 1.83 m × 1.21 m × 0.63 m. Environment 1 is a real indoor scene from a university building at Hokkaido University, measured using terrestrial laser scanning with a Leica BLK360 G2. The environment has dimensions of 26.6 m × 10.2 m × 1.9 m. Environment 2 is a synthetic narrow-passage environment modeled in Unreal Engine, with dimensions of 50.0 m × 50.0 m × 2.1 m. Environment 3 is a real indoor scene from the heat-source plant at Hokkaido University, measured using mobile handheld laser scanning with a FARO Orbis. The environment has dimensions of 35.1 m × 17.1 m × 2.0 m.

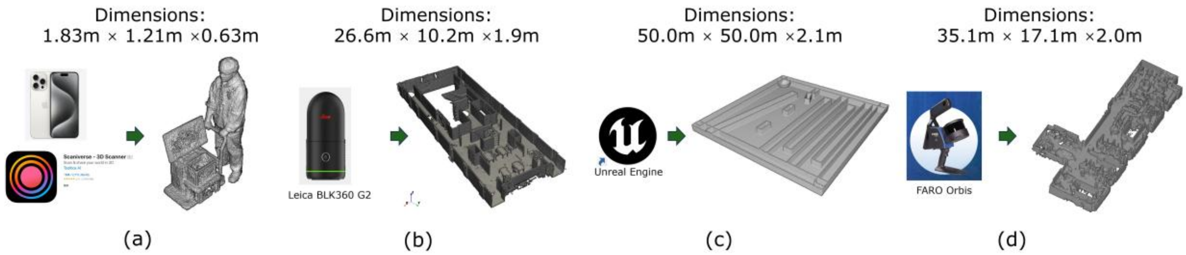

Figure 2: Point-cloud datasets used for evaluation: (a) Transport unit, (b) Environment 1, (c) Environment 2, and (d) Environment 3.

Item		Env.1	Env.2	Env.3
Number of Points in Environment Point Cloud P_E		1,771,669	358,785	3,132,441
Number of Environment Voxels V_E		133,049	110,044	361,113
Total Number of Voxels in all C-space Obstacle Sets $V_E^{(\theta,Offset)}$		199,770,444	964,309,388	322,426,168
Computation Time	Voxelization Time [s]	6.8	19.5	11.9
	Inverse Offset Processing Time [s]	22.5	346.7	33.1
	Risk-field Construction Time [s]	7.8	47.3	11.3
	Path Computation Time [s]	12.8	192.8	2.1
	Total Computation Time [s]	49.8	606.4	58.4

Table 1: Point-cloud inputs, voxel statistics, and computation times for the three environments.

Table 1 summarizes the point-cloud inputs, voxel statistics, and computation times for the three environments. Specifically, it reports the point count of each environment point cloud, number of environment voxels, total number of pose-dependent C-space obstacle voxels, and computation times for voxelization, inverse offset processing, risk-field construction, path search, and the overall pipeline. The voxel size was set to 0.05 m because all evaluated scenes contained narrow passages. All experiments were conducted on a 64-bit Windows workstation equipped with an Intel Core i9-14900K 3.20 GHz CPU, NVIDIA GeForce RTX 4090 GPU, and 64 GB RAM. The implementation was developed using Python 3.11.

4.2 Metrics and Experimental Conditions

We defined the high-risk rate as the percentage of sampled path states whose clearance-based risk values exceeded the threshold of 0.75. This threshold corresponds to a clearance of approximately 25 cm in the real world, indicating a limited safety margin during transport. To isolate the effects of the clearance-based risk term and pose transitions, we compare three cases, as summarized in Table 2. Case 1 (C1) uses neither a clearance-based risk term nor pose transitions. Case 2 (C2) uses a clearance-based risk term but does not allow pose transitions. Case 3 (C3) uses both the clearance-based risk term and pose transitions, and corresponds to the proposed method. Figure 3 and Figure 4 compare the resulting transport paths in Environments 1 and 2 under conditions C1–C3 along with their corresponding pose transitions. Figure 5 shows only the C3 results for Environment 3 because no feasible path was found under C1 or C2.

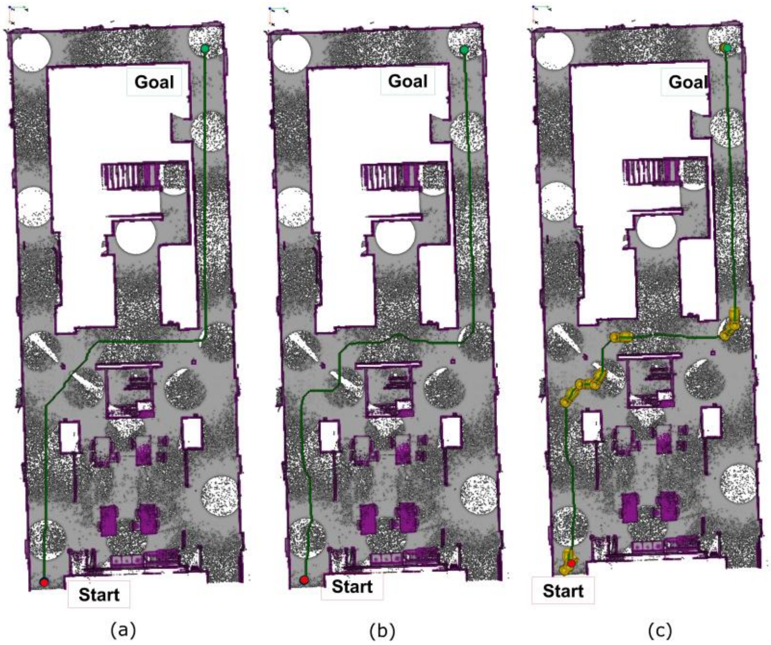

Figure 3: Path and pose-transition results in Environment 1: (a) C1, (b) C2, and (c) C3.

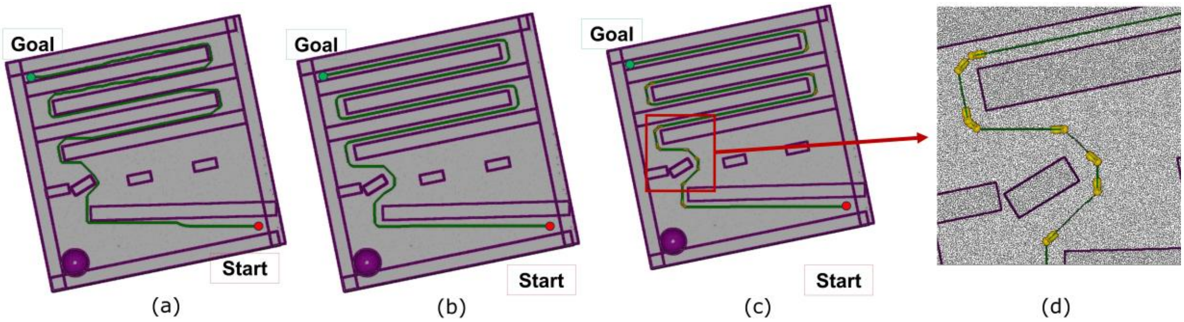

Figure 4: Path and pose-transition results in Environment 2: (a) C1, (b) C2, and (c) C3; (d) Magnified view of the boxed region in (c), showing pose transitions in the narrow passage.

Figure 5: C3 result in Environment 3.

		Env.1			Env.2			Env.3		
		C1	C2	C3	C1	C2	C3	C1	C2	C3
Case Setting	Clearance-based Risk Term	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes
	Pose Transitions	No	No	Yes	No	No	Yes	No	No	Yes
Performance Metrics	Path Length [m]	34.22	37.61	36.0	228.0	241.7	238.1	-	-	24.6
	Minimum Clearance [m]	0.00	0.18	0.35	0.00	0.49	0.49	-	-	0.49
	Average Clearance [m]	0.21	0.54	0.58	0.30	0.74	0.73	-	-	0.50
	High-risk Rate [%]	46.2	1.8	0.0	64.3	0.0	0.0	-	-	18.4
	Total Cost	172.2	142.1	126.7	1245.2	689.3	662.8	-	-	129.8

Table 2: Quantitative comparison of the three cases in Environments 1–3.

4.3 Results

Table 2 shows the consistent trends across Environments 1 and 2. Compared with C1, introducing the clearance-based risk term in C2 substantially reduced the high-risk rate from 46.2% to 1.8% in Environment 1 and from 64.3% to 0.0% in Environment 2.

The minimum clearance increases from 0.00 m to 0.18 m in Environment 1 and from 0.00 m to 0.49 m in Environment 2, whereas the average clearance increases from 0.21 m to 0.54 m and from 0.30 m to 0.74 m, respectively. However, it also increases the path length by 9.9% and 6.0% because the fixed-pose planner must detour to maintain safety margins. Allowing pose transitions in C3 preserves these safety benefits while partially recovering path efficiency. In both Environments 1 and 2, the path length is reduced compared with C2 (from 37.61 m to 36.0 m and from 241.7 m to 238.1 m, respectively), while maintaining low or zero high-risk rates (0.0% in both environments) and comparable clearance (minimum clearance of 0.35 m and 0.49 m, respectively). These results indicate that local pose adjustment enables the transport unit to pass through narrow regions with fewer detours.

The strongest effect was observed in Environment 3. Under the fixed-pose settings (C1 and C2), no feasible path was found, whereas C3 succeeded in generating a solution. This result indicates that in cluttered and spatially constrained real-world environments, pose transitions are not only beneficial for improving path quality, but may also be necessary for path feasibility.

Figure 6 further visualizes the spatial distribution of the clearance-based risk field across the three environments. The heatmaps show that high-risk regions are concentrated in narrow passages and obstacle-dense areas, where the available clearance is limited. Warmer colors indicate smaller clearance and higher risk, whereas cooler colors indicate larger clearance and lower risk. Blank regions indicate free-space cells with zero clearance-based risk within the plotted planning area. These visualizations complement the quantitative results in Table 2, including minimum clearance, average clearance, and high-risk rate.

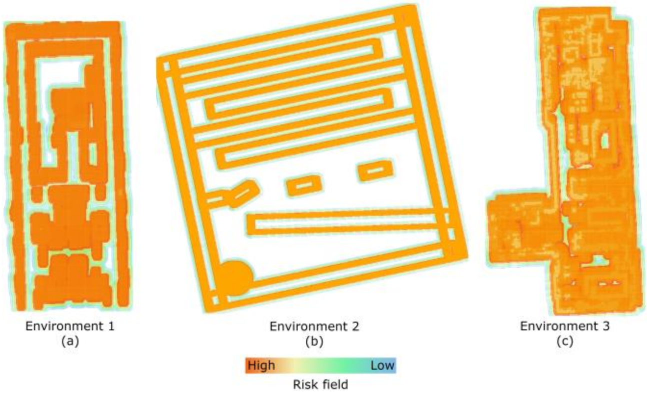

Figure 6: Risk field heatmap visualizations in the three environments: (a) Environment 1, (b) Environment 3, and (c) Environment 2.

4.4 Sensitivity Analysis

We conducted a sensitivity analysis to investigate the effects of parameter settings on planning performance. Figure 7 qualitatively compares the resulting transport paths and pose transitions in Environment 1 under the four parameter settings, while Table 3 summarizes the corresponding quantitative results.

Figure 7: Qualitative comparison under the four parameter settings in Environment 1.

Category	Item	Setting A	Setting B	Setting C	Setting D
Parameters	Voxel Size [m]	0.05	0.05	0.05	0.10
	Risk Weight w_d	6	6	12	6
	Path-length Weight w_l	1	2	1	1
Performance	Total Computation Time [s]	49.8	48.9	49.0	13.5
	Path Length [m]	36.0	36.1	36.7	37.1
	Minimum Clearance [m]	0.35	0.35	0.18	0.10
	Average Clearance [m]	0.58	0.57	0.59	0.50
	High-risk Rate [%]	0.0	0.0	1.7	3.7

Table 3: Sensitivity analysis under the four parameter settings in Environment 1.

In the tested range, the effects of w_d and w_l were limited and non-monotonic. Increasing w_d from 6 to 12 did not improve the minimum clearance; instead, the minimum clearance decreased from 0.35 m to 0.18 m and the high-risk rate increased from 0.0% to 1.7%. This suggests that in Environment 1, the minimum clearance was dominated by a small number of bottleneck regions, where the achievable clearance was mainly constrained by the local geometry and feasible discretized yaw set. Under such conditions, a larger w_d primarily affects the preference among already feasible paths rather than increasing the bottleneck clearance itself. Changing w_l from 1 to 2 resulted in only minor changes in the total computation time, path length, clearance, and high-risk rate. In contrast, the effect of voxel size was more pronounced. Increasing the voxel size from 0.05 m to 0.10 m substantially reduced the total computation time, from approximately 49 s to 13 s, but also reduced the minimum clearance from 0.35 m to 0.10 m and increased the high-risk rate from 0.0% to 3.7%. Therefore, in our experiments, a finer resolution (0.05 m) was necessary to preserve safe passages in confined spaces.

To further evaluate scalability, we conducted an additional controlled experiment in Environment 1. Unlike Table 1, which reports the full planning pipeline for the three evaluation environments, this experiment focused on the effects of the voxel resolution and yaw discretization on the explicit C-space storage and construction costs. The voxel size was set to 0.05 m, 0.10 m, and 0.20 m, and the number of yaw layers was set to 9, 18, 36, and 72, corresponding to yaw intervals of 40°, 20°, 10°, and 5°, respectively.

Table 4 reports the stored C-space array size in MB. The results show that explicit C-space storage increases approximately with the number of yaw layers for a fixed voxel size. For example, at a voxel size of 0.05 m, increasing the yaw layers from 9 to 72 increases the C-space storage from 618.57 MB to 4937.52 MB. For a fixed yaw discretization, finer voxel sizes substantially increase both C-space storage and construction time. This trend is expected because a separate pose-dependent C-space obstacle set is generated for each yaw layer.

For larger facilities, explicit precomputation can become memory intensive. Further memory reduction may be achieved through octree-based representations or by selectively constructing C-space obstacle sets for only those regions required during planning. Such alternatives can reduce storage requirements in large sparse environments or avoid the generation of C-space obstacles in unexplored regions, although they may introduce additional query-time overhead. Therefore, these strategies are expected to improve scalability rather than directly improve path quality, and their integration remains a topic for future work.

Voxel Size [m]	Yaw Layers	Total C-space Voxels	C-space Storage [MB]	C-space Construction Time [s]
0.05	9	51,547,312	618.57	15.08
	18	103,147,909	1237.77	19.69
	36	199,770,444	2397.24	22.53
	72	411,460,185	4937.52	46.26

0.1	9	7,730,783	92.77	3.19
	18	15,351,283	184.22	4.17
	36	30,655,531	367.87	6.07
	72	61,394,517	736.73	9.74
0.2	9	1,221,252	14.66	0.81
	18	2,446,355	29.36	1.06
	36	4,901,804	58.82	1.45
	72	9,801,605	117.62	2.34

Table 4: Scaling of explicit C-space storage and construction time with voxel resolution and yaw discretization.

5 CONCLUSIONS

This paper proposes a risk-aware path-planning method for object transport that combines distance-field-based risk evaluation with an extended A* search in point-cloud environments. This method targets a transport unit composed of a transported object and a worker. Based on a voxelized representation of the environment, the method jointly considers obstacle-clearance risk, pose transitions during transport, and path cost, thereby achieving joint planning of the transport path and pose sequence.

In conclusion, the proposed method enables path-planning for object transport in complex and narrow environments, while jointly considering feasibility and safety margins. The results showed that integrating pose-dependent C-space obstacles, a clearance-based risk field, and extended A* search within a unified framework was effective in real-world scanned environments. In highly constrained environments, pose transitions are not only useful for improving path quality but may also be necessary for obtaining a feasible path. The current framework depends on the quality of the input point clouds, because noisy or incomplete measurements may affect the voxel occupancy, pose-dependent C-space obstacles, and clearance-based risk fields. Future work will extend the framework to include height variation and non-planar floors, richer position–pose changes beyond planar yaw transitions, dynamic transport scenes in which both surrounding obstacles and worker arrangements change over time, and systematic evaluations under noisy or incomplete point-cloud conditions.

Linxuan He, https://orcid.org/0009-0009-3599-6913
Hiroaki Date, https://orcid.org/0000-0002-6189-2044

6 ACKNOWLEDGEMENTS

This work was supported by JST SPRING, Grant Number JPMJSP2119.

REFERENCES

[1] ACI Committee 302: ACI 302.1R-15, Guide to Concrete Floor and Slab Construction, American Concrete Institute (ACI), 2015. Available at: https://www.concrete.org/publications/internationalconcreteabstractsportal.aspx?id=51687974&m=details
[2] ASTM E1155/E1155M-23, Standard Test Method for Determining FF Floor Flatness and FL Floor Levelness Numbers. ASTM International, 2023. https://doi.org/10.1520/E1155_E1155M-23
[3] BS EN ISO 13857:2019, Safety of machinery — Safety distances to prevent hazard zones being reached by upper and lower limbs. British Standards Institution (BSI), 2019. https://doi.org/10.3403/30358017

[4] Curless, B.; Levoy, M.: A Volumetric Method for Building Complex Models from Range Images, Proc. SIGGRAPH, 1996, 303–312. https://doi.org/10.1145/237170.237269
[5] DIN EN ISO 13854:2020-01, Safety of machinery — Minimum gaps to avoid crushing of parts of the human body (ISO 13854:2017); German version EN ISO 13854:2019. DIN Media, 2020. https://doi.org/10.31030/3113240
[6] Elfes, A.: Using Occupancy Grids for Mobile Robot Perception and Navigation, Computer, 22(6), 1989, 46–57. https://doi.org/10.1109/2.30720
[7] Felzenszwalb, P. F.; Huttenlocher, D. P.: Distance Transforms of Sampled Functions, Theory of Computing, 8(1), 2012, 415–428. https://doi.org/10.4086/toc.2012.v008a019
[8] Fischler, M. A.; Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, 24(6), 1981, 381–395. https://doi.org/10.1145/358669.358692
[9] Fox, D.; Burgard, W.; Thrun, S.: The Dynamic Window Approach to Collision Avoidance, IEEE Robotics & Automation Magazine, 4(1), 1997, 23–33. https://doi.org/10.1109/100.580977
[10] Hart, P. E.; Nilsson, N. J.; Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics, 4(2), 1968, 100–107. https://doi.org/10.1109/TSSC.1968.300136
[11] Hornung, A.; Wurm, K. M.; Bennewitz, M.; Stachniss, C.; Burgard, W.: OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, 34(3), 2013, 189–206. https://doi.org/10.1007/s10514-012-9321-0.
[12] Latombe, J.-C.: Robot Motion Planning, Springer, 1991. https://doi.org/10.1007/978-1-4615-4022-9
[13] Li, W.; McMains, S.: Voxelized Minkowski sum computation on the GPU with robust culling, Computer-Aided Design, 43(10), 2011, 1270–1283. https://doi.org/10.1016/j.cad.2011.06.022
[14] Likhachev, M.; Gordon, G. J.; Thrun, S.: ARA*: Anytime A* with Provable Bounds on Sub-Optimality, Advances in Neural Information Processing Systems (NIPS), 2003. Available at: https://papers.nips.cc/paper/2382-ara-anytime-a-with-provable-bounds-on-sub-optimality
[15] Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P.: Path planning techniques for mobile robots: Review and prospect, Expert Systems with Applications, 227, 2023, 120254. https://doi.org/10.1016/j.eswa.2023.120254
[16] Malhan, R. K.; Gupta, S. K.: Planning algorithms for acquiring high-fidelity point clouds using a robot for accurate and fast 3D reconstruction, Robotics and Computer-Integrated Manufacturing, 78, 2022, 102372. https://doi.org/10.1016/j.rcim.2022.102372.
[17] Ministry of Land, Infrastructure, Transport and Tourism: Public Building Standard Specifications, Architectural Works, 2025 Edition. Ministry of Land, Infrastructure, Transport and Tourism (MLIT), Japan, 2025. Available at: https://www.mlit.go.jp/gobuild/kenchiku_hyoushi.html
[18] Newcombe, R. A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A. J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A.: KinectFusion: Real-time Dense Surface Mapping and Tracking, Proc. IEEE ISMAR, 2011, 127–136. https://doi.org/10.1109/ISMAR.2011.6092378
[19] Niwa, T.; Masuda, H.: Interactive collision detection for engineering plants based on large-scale point-clouds, Computer-Aided Design & Applications, 13(4), 2016, 511–518. https://doi.org/10.1080/16864360.2015.1131546
[20] Nordström, S.; Bai, Y.; Lindqvist, B.; Nikolakopoulos, G.: A Time-dependent Risk-aware distributed Multi-Agent Path Finder based on A*, arXiv preprint, 2025, arXiv:2504.19593. https://doi.org/10.48550/arXiv.2504.19593
[21] Pan, J.; Chitta, S.; Manocha, D.: FCL: A General Purpose Library for Collision and Proximity Queries, Proc. IEEE ICRA, 2012. https://doi.org/10.1109/ICRA.2012.6225337
[22] Schnabel, R.; Wahl, R.; Klein, R.: Efficient RANSAC for Point-Cloud Shape Detection, Computer Graphics Forum, 26(2), 2007, 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x
[23] U.S. Department of Energy: DOE Handbook: Human Factors/Ergonomics Handbook for the Design for Ease of Maintenance (DOE-HDBK-1140-2001), U.S. Department of Energy,

Washington, DC, 2001. Available at: https://www.standards.doe.gov/standards-documents/1100/1140-bhdbk-2001-pt1

[24] Varadhan, G.; Manocha, D.: Accurate Minkowski Sum Approximation of Polyhedral Models, Graphical Models, 68(4), 2006, 343–355. https://doi.org/10.1016/j.gmod.2005.11.003

[25] Whelan, T.; et al.: Real-time Large-scale Dense RGB-D SLAM with Volumetric Fusion, The International Journal of Robotics Research, 34(4–5), 2015, 598–626. https://doi.org/10.1177/0278364914551008

[26] Xu, Z.; Niu, Y.; Jiang, J.; Qin, R.; Cui, X.: Pose-graph optimization for efficient tie-point matching and 3D scene reconstruction from oblique UAV images, ISPRS Journal of Photogrammetry and Remote Sensing, 225, 2025, 461–491. https://doi.org/10.1016/j.isprsjprs.2025.04.013