Multi-resolution H-Cost Motion Planning: A New Framework for Hierarchical Motion Planning for Autonomous Mobile Vehicles

Raghvendra V. Cowlagi and Panagiotis Tsiotras

***Abstract*— This paper summarizes some recent developments on a new motion planning framework for autonomous vehicles. The main novelties of the current work include: a provably complete multi-resolution path planning scheme using wavelet-based workspace cell decompositions; a general technique for incorporating vehicle dynamic constraints in the geometric path planner; and a local trajectory generation scheme based on model predictive control.**

I. Introduction

The traditional “sense-plan-execute” hierarchical architecture for autonomous mobile vehicles is being reevaluated by the robotics community (cf. [1], [2]) to ensure compatibility between the different levels in the hierarchy. Currently, the navigation and sensing systems seem independent of the motion planning system [3]. Within the motion planning system itself, the high-level discrete path planners often ignore (or capture inadequately) the vehicle dynamical constraints [1].

References [4]–[6] address multi-resolution path planning for vehicles with limited on-board computational resources, whereas Ref. [7], addresses the issue of incorporating vehicle kinematic and dynamic constraints in the high-level motion planning phase. In the current paper, we integrate these results and develop a new motion planning framework that is “multi-resolution” both in the sense of representing the environment with high accuracy only locally, *and* in the sense of considering the vehicle dynamical constraints for path planning only locally. The various inter-related aspects of the proposed framework are introduced as follows.

Wavelet-based multi-resolution path planning: Multi-resolution path planning involves representing the vehicle’s environment with different levels of accuracy to construct an overall representation that allows for efficient online path planning. Examples of such path planning schemes appear, for instance, in Refs. [8] (quadtree-based cell decompositions); Ref. [9] (multi-resolution estimates of object locations); and Ref. [10] (hierarchy of spheres for collision avoidance). References [4]–[6], [11] discuss path planning using multi-resolution cell decompositions, where high resolution cells (i.e. cells of small sizes) are used in the vehicle’s immediate vicinity and larger cells approximating the environment are used in regions farther away. This approach relates to two alternative perspectives on path planning: (a) multi-resolution cell decompositions significantly reduce the number of vertices in the cell decomposition graphs, thereby compressing *too much information* about the environment to enable efficient online computation, and (b) the associated path planning scheme requires accurate environment information only locally, thereby addressing *uncertainty or partial knowledge* about the environment in regions farther away from the vehicle’s location.

The discrete wavelet transform is a powerful and computationally efficient tool widely used in multi-resolution signal processing [12]. Several recent works have investigated the applications of wavelet transforms to vision-based navigation and vision-based SLAM: see, for instance, Ref. [13] (appearance-based vision-only SLAM); Refs. [14] and [15] (wavelet analysis for local feature extraction); and Ref. [16] (stereo image processing). With the plethora of available sensors, and in light of the fact that multiple sensors are typically used for autonomous navigation [3], the wavelet transform may soon become the common standard of the representation and analysis of signals [17]. In this context, several recent works address wavelet-based data representation: see, for instance, Ref. [18] (occupancy grids); Ref. [19] (standardized representation of road roughness characteristics); Ref. [20] (image registration using wavelets); and Ref. [21] (wavelet-based approximations of a terrain map).

History-based costs in motion planning: Motion planning for autonomous vehicles often involves a hierarchy consisting of a high-level *geometric path planner* that uses a discrete representation of the vehicle’s workspace (such as cell decompositions) and deals with the satisfaction of the task specifications (such as obstacle avoidance), and a low-level *trajectory planner* that deals with the vehicle’s kinematic and dynamic constraints. In order to ensure “consistency” between the two planners, i.e., to ensure that the geometric path remains feasible when vehicle dynamical constraints are considered at the trajectory planning level, we have introduced in [7] a motion planning framework based on assigning costs to *multiple* edge transitions – the so-called H-costs – in the graphs associated with cell decompositions. These costs are assigned by solving the so-called *tile motion planning* problem.

The main contribution of this paper is a new motion planning framework that combines the efficiency of multi-resolution path planning of [6] with the guarantee of consistency provided by the H-cost motion planner of [7].

The rest of the paper is organized as follows. In Section II, we summarize the wavelet transform-based multi-resolution path planning scheme. In Section III, we discuss the notion of H-cost path planning. In Section IV, we discuss a tile

R. V. Cowlagi is a post-doctoral fellow at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA. rcowlagi@gatech.edu

P. Tsiotras is with the Faculty of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA. tsiotras@gatech.edu

motion planning scheme based on model predictive control. In Section V, we discuss the integration of the preceding results to develop a multi-resolution, H-cost motion planning framework, and we provide simulation results of implementation of this framework. Finally, we conclude the paper with comments about its generality and possibilities for future extensions.

II. Multi-resolution Path Planning using the Discrete Wavelet Transform

Multi-resolution analysis using the wavelet transform [12] involves the construction of a hierarchy of approximations of a scalar function by projecting it onto a sequence of nested linear spaces. These linear spaces are the spans of translated and scaled versions of two functions called, respectively, the *scaling function* and the *mother wavelet*. The *2-D discrete wavelet transform* of a function $F \in \mathcal{L}^2(\mathbb{R}^2)$ refers to two collections of scalars, called, respectively, the *approximation* coefficients and the *detail* coefficients, defined, respectively, as the inner products of F with translated and scaled versions of the scaling function and the mother wavelet.

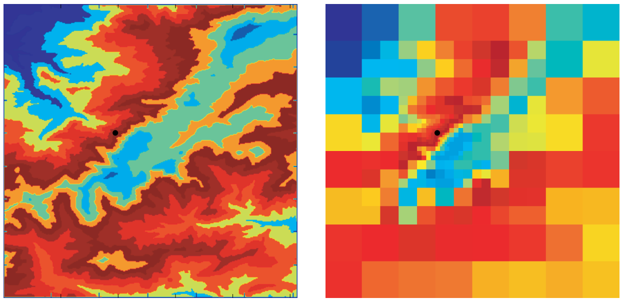
We define an *image* as a pair (R, F), where $R \subset \mathbb{R}^2$ is a compact, square region, and $\mathbb{L}^2(R) \ni F : R \to \mathbb{R}$ is an intensity map. We assume that $R = \left[0, 2^D\right] \times \left[0, 2^D\right]$, with $D \in \mathbb{Z}$, and that the image intensity map F is known at a finite resolution $m_\mathrm{f} > -D$, i.e., the function F is piecewise constant over each of the square regions $S_{m_\mathrm{f},k,\ell} := [2^{-m_\mathrm{f}}k, 2^{-m_\mathrm{f}}(k+1)] \times [2^{-m_\mathrm{f}}\ell, 2^{-m_\mathrm{f}}(\ell+1)]$, for $k, \ell = 0, 1, \ldots, 2^{D-m_\mathrm{f}}-1$. Without loss of generality, we may assume $m_\mathrm{f} = 0$. For path planning, the intensity map F may represent, for instance, the terrain elevation [21], a risk measure [4], or a probabilistic occupancy grid [18].

We assume that the smallest cell size is $2^{-m_\mathrm{f}} = 1$. We define a cell decomposition Ω consisting of uniformly spaced cells, each of size 1. Whereas we intend to find a path in the graph associated with Ω, the number of cells in Ω is 2^{2D}, which makes the graph search inefficient when D is large. To alleviate this difficulty, we construct multi-resolution cell decompositions that have significantly fewer cells.

Let $a_{m_0,k,\ell}$ and $d^i_{m,k,\ell}$ be, respectively, the approximation and detail coefficients of the 2-D discrete wavelet transform of the function F, where $m_0 \in \mathbb{Z}$ is pre-specified. Let $\mathcal{A}$ be a set of triplets of integers, and let $\hat{d}^i_{m,k,\ell}$ be defined as

$$\hat{d}^i_{m,k,\ell} \quad := \quad \begin{cases} d^i_{m,k,\ell} & i = 1,2,3, \text{ and } (m,k,\ell) \in \mathcal{A}, \\ 0 & \text{otherwise.} \end{cases}$$

The image $(R, \hat{F})$, where $\hat{F}$ is obtained by the reconstruction of $a_{m_0,k,\ell}$ and $\hat{d}^i_{m,k,\ell}$, is called the *approximation* associated with $\mathcal{A}$. The set $\mathcal{A}$ contains the indices of detail coefficients that are "significant" for the approximation $(R, \hat{F})$.



(a) Original image

(b) Approximation

Fig. 1. Example of an image and its approximation using in (1).

The cell decomposition Ω^{mr} associated with $(R, \hat{F})$ is a partition of R into square cells of different sizes such that $\hat{F}$ is constant over each of the cells. Of interest in this work is the approximation that retains detail coefficients only in the immediate vicinity of the vehicle's location and gradually discards them in regions farther away, described as follows. Let $(x_0, y_0) \in R$ be the location of the vehicle, and let $\varrho : \mathbb{Z} \to \mathbb{N}$ be a "window" function that specifies, for each level of resolution, the vicinity of the vehicle's location in which the detail coefficients at that level are considered significant. Then the set $\mathcal{A}$ of significant detail coefficients is given by

$$\begin{aligned} \mathcal{A} \quad = \quad & \big\{ d^i_{m,k,\ell} : m_0 \leqslant m < 0, \quad i = 1,2,3, \\ & \lfloor 2^m x_0 \rfloor - \varrho(m) \leqslant k \leqslant \lfloor 2^m x_0 \rfloor + \varrho(m), \\ & \lfloor 2^m y_0 \rfloor - \varrho(m) \leqslant \ell \leqslant \lfloor 2^m y_0 \rfloor + \varrho(m) \big\}, \end{aligned} \qquad (1)$$

where $m_0 \in \mathbb{Z}$ is pre-specified. An example of an approximation using (1) is shown in Fig. 1, with $m_0 = -10$, $(x_0, y_0) = (390, 449)$, and $\varrho(m) = 4$ for each $m_0 \leqslant m < 0$.

In [23], we provide a procedure to determine the locations, the intensities, and the adjacency relations of the cells in Ω^{mr}. At each iteration, the path planning algorithm searches the graph associated with Ω^{mr}, then advances the vehicle's location by one cell in the path thus found, and then constructs another multi-resolution cell decomposition via (1) using the vehicle's new location. The results of experimental evaluation of this scheme on a UAV platform are available in Ref. [5]. In [6], we discuss modifications to this scheme and rigorously proved the completeness of the modified scheme. In [6], we also discuss fast recomputation of the new cell decomposition corresponding to advances in the vehicle location.

III. Motion Planning based on H-Cost Paths

Single-edge transition costs in workspace cell decomposition graphs cannot capture adequately the vehicle's kinematic and dynamic constraints [7], [24]. In light of this observation, we formalize in [7] the idea of defining transition costs on *multiple* successive edges. We denote by $\mathcal{G} = (V, E)$ the graph associated with the cell decomposition, where each obstacle-free cell corresponds to a unique vertex in V and each pair of adjacent geometrically cells correspond to a unique edge in E. Next, for every integer $H \geqslant 0$, we define

$$\begin{aligned} V_H \quad := \quad & \{(j_0, \ldots, j_H) : \{j_{k-1}, j_k\} \in E, \ k = 1, \ldots, H, \\ & j_k \neq j_m, \ \text{ for } k, m \in \{0, \ldots, H\}, \ \text{ with } k \neq m\}. \end{aligned}$$

An element of V_{H+1}, which is a sequence of $H+1$ successive edges in $\mathcal{G}$, is called a *H-history*. We define the *H-cost shortest path problem* as a shortest path problem on $\mathcal{G}$ with transition costs defined on H-histories. This problem

```
Tile Motion Planning Algorithm
procedure TILEPLAN
  Determine if there exist $t_\mathrm{f} \in \mathbb{R}$ and admissible control input $u$ such that $\xi(\cdot\,;\xi_0,u)$ satisfies
      $\mathsf{x}(\xi(t;\xi_0,u)) \in \bigcup_{k=1}^{H} \mathsf{cell}([J]_k),\ \ t \in (0,t_\mathrm{f})\,,$   (2)
      $\mathsf{x}(\xi(t_\mathrm{f};\xi_0,u)) \in \mathsf{cell}([J]_H) \cap \mathsf{cell}([J]_{H+1})$   (3)
  if $\exists t_\mathrm{f}$ and $\exists u$ then
      Find $t_1$ such that
          $\mathsf{x}(\xi(t_1;\xi_0,u)) \in \mathsf{cell}([J]_1) \cap \mathsf{cell}([J]_2)$   (4)
      Return $t_1$, $u_{[0,t_1]}$, $\xi_1 := \xi(t_1;\xi_0,u)$, and $\Lambda := \int_0^{t_1} \ell(\xi(t;\xi_0,u),u,t)\,\mathrm{d}t$
  else
      Return $\Lambda = \infty$
```

Fig. 2. General form of the tile motion planning algorithm.

is equivalent to a standard shortest path problem on the *lifted graph* $\mathcal{G}_H := (V_H, E_H)$, where the edge set E_H is defined as follows. Let $I, J \in V$; then J is adjacent to I, i.e., $(I,J) \in E_H$, if $[I]_k = [J]_{k-1}$, for every $k = 2,\ldots,H+1$, and $[I]_1 \neq [J]_{H+1}$, where $[I]_k$ denotes the k^{th} element of the $(H+1)$-tuple represented by I.

The solution of the H-cost shortest path problem is computationally expensive because the number of vertices in the lifted graph $\mathcal{G}_H$ grows exponentially with H. In [7], we present an algorithm that efficiently computes the shortest H-cost path. Crucially, this algorithm accepts a parameter that allows a trade-off between optimality and execution time. In [7], we also present a motion planning framework based on H-cost shortest paths; in this framework, the transition costs on H-histories are assigned by solving a low-level trajectory generation problem described next.

A. Specifications for Tile Motion Planning

We consider a vehicle model described as follows. Let $(x,y,\theta) \in \mathcal{C} := \mathbb{R}^2 \times \mathbb{S}^1$ denote the position coordinates of the vehicle in a pre-specified Cartesian axis system, and let ψ denote any additional state variables required to describe the state of the vehicle. We assume that $\psi \in \Psi$, where Ψ is a n-dimensional smooth manifold. The state of the vehicle is thus $\xi := (x,y,\theta,\psi) \in \mathcal{D} := \mathcal{C} \times \Psi$. Let $U \in \mathbb{R}^m$ denote the set of admissible control values. The evolution of the vehicle state ξ over a given time interval $[0,t_\mathrm{f}]$ is described by the differential equation $\dot{\xi}(t) = f(\xi(t),u(t))$ for all $t \in [0,t_\mathrm{f}]$, where u is a piecewise continuous function on $[0,t_f]$ taking values in U (henceforth referred to as an admissible control), and f is sufficiently smooth to guarantee global existence and uniqueness of solutions. We denote by $\xi(\cdot\,;\xi_0,u)$ the state trajectory that is the unique solution to the preceding differential equation with initial condition $\xi(0) = \xi_0$. Finally, we denote by $\mathsf{x}(\xi)$ the projection of a state ξ on $\mathbb{R}^2$.

We define a *tile* as the sequence of cells associated with a H-history (I,J), where $(I,J) \in E_H$. A *tile motion planner* (TILEPLAN) is any algorithm that determines if a given tile may be feasibly traversed. A precise and general description of TILEPLAN is given in Fig. 2.

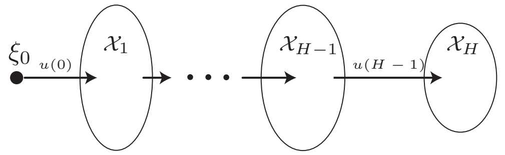


Fig. 3. Schematic illustration of the idea of effective target sets.

In Ref. [23], we provide numerical simulation data that demonstrates the superiority of the H-cost motion planning approach over randomized sampling-based approaches (such as those based on RRTs [25] and its variants).

IV. TILE MOTION PLANNING VIA MPC

The implementation of TILEPLAN is difficult mainly because (2) imposes a non-convex constraint on the state trajectory. To alleviate this difficulty, we use the idea of *effective target sets* introduced in [26], which enable the use of MPC-based techniques for implementing TILEPLAN by transforming the constraint in (2) to a convex constraint defined over a single cell.

Consider the tile associated with the H-history $(I,J) \in E_H$. We define a sequence $\{\mathcal{X}_k\}_{k=1}^{H+1}$ of subsets of the vehicle state space called the *effective target sets* as follows. Let

$$\mathcal{X}_H := (\mathsf{cell}([J]_H) \cap \mathsf{cell}([J]_{H+1})) \times [-\pi,\pi] \times \Psi.$$

For each $k = 1,\ldots,H-1$, we define the effective target set $\mathcal{X}_k$ as the set of all states $\xi_k \in \mathcal{D}$ such that $\mathsf{x}(\xi_k) \in \mathsf{cell}([J]_k) \cap \mathsf{cell}([J]_{k+1})$ and such that there exists $t_{k+1} \in \mathbb{R}_+$ and an admissible control u_{k+1} such that

$$\mathsf{x}(\xi(t;\xi_k,u_{k+1})) \in \mathsf{cell}([J]_{k+1}),\ \ t \in (0,t_{k+1})\,, \tag{5}$$
$$\xi(t_{k+1};\xi_k,u_{k+1}) \in \mathcal{X}_{k+1}. \tag{6}$$

The preceding definition of effective target sets allows a simplification of the tile motion planning problem as follows. Suppose there exist a time t_1 and an admissible control u_1 such that the resultant state trajectory $\xi(\cdot\,;\xi_0,u_1)$ satisfies

$$\mathsf{x}(\xi(t;\xi_0,u_1)) \in \mathsf{cell}(j_1),\ \ t \in (0,t_1)\,, \tag{7}$$
$$\xi_1 := \xi(t_1;\xi_0,u_1) \in \mathcal{X}_1. \tag{8}$$

Note that the conditions (7)-(8) imply the satisfaction of (2)-(3) for $H = 1$. Continuing recursively the preceding argument (Fig. 3), it follows that, for each $H \geqslant 2$, there exist $t_{k+1} \in \mathbb{R}_+$ and admissible inputs u_{k+1}, for $k = 1,\ldots,H-1$, such that the admissible input u defined by

$$u(t) \ \ := \ \ \begin{cases} u_1(t), & t \in [0,T_1)\,, \\ \vdots & \vdots \\ u_H(t), & t \in [T_{H-1},T_H]\,, \end{cases} \tag{9}$$

where $T_k = \sum_{m=1}^{k} t_m$ solves the tile planning problem.

Thus, if the effective target sets $\mathcal{X}_k$, the corresponding times of traversal t_{k+1} and the control inputs u_{k+1} in (9) are known for each $k = 1,\ldots,H-1$, then the tile motion planning problem is equivalent to the problem of finding u_1 and t_1 as described above. Crucially, (7) constrains the position components of the state trajectory to lie within a convex set. Furthermore, we may replace $\mathcal{X}_1$ in (8) by an interior convex approximating set $\tilde{\mathcal{X}}_1 \subset \mathcal{X}_1$ thus transforming the tile motion planning problem into the problem of finding u_1 and t_1 subject to convex constraints.

The tile motion planning problem can now be expressed in the following, standard, discrete-time MPC form:

$$\begin{aligned}
&\min_{H_{\mathrm{P}}\in\mathbb{N},(u(0),\ldots,u(H_{\mathrm{P}}))} \left\{ \tilde{\Lambda}_{\mathrm{f}}(\xi(H_{\mathrm{P}})) + \sum_{k=0}^{H_{\mathrm{P}}-1} \tilde{\ell}(\xi(k),u(k)) \right\},\\
&\text{subject to } \xi(H_{\mathrm{P}}) \in \tilde{\mathcal{X}}_1, \qquad \xi(k) \in \mathsf{cell}([J]_1), \qquad (10)\\
&\text{and } u(k) \in U, \text{ for each } k \in \{0,\ldots,H_{\mathrm{P}}-1\},
\end{aligned}$$

where H_{P} denotes the prediction horizon, $\tilde{\ell} : \mathcal{D} \times U \to \mathbb{R}_+$ denotes a pre-specified incremental cost function, and $\tilde{\Lambda}_{\mathrm{f}} : \mathcal{D} \to \mathbb{R}_+$ denotes a pre-specified terminal cost function. Note that the state constraints in (10) are convex.

For general nonlinear dynamical systems, the construction of the effective target sets is difficult. However, in light of the fact that the vehicle state includes the configuration (x, y, θ), we may compute *geometrically* the intersections $\mathcal{C}_k := \mathcal{X}_k \cap \mathcal{C}$ of the effective target sets with the configuration space $\mathcal{C}$. In particular, for the vehicle dynamical model introduced in Section III, it can be shown that the curvature of feasible geometric paths is upper-bounded by

$$\kappa^{\max} \quad := \quad \min_{t\in[0,t_{\mathrm{f}}]} \max_{u(t)\in U} \left| \dot{\theta}(\xi(t),u(t))/v(t) \right|. \qquad (11)$$

The sets $\mathcal{C}_k$ for a given tile may then be computed by analyzing curvature bounded traversal of this tile. Accordingly, we provide in [23] a procedure to determine the existence of curvature bounded paths traversing sequences of rectangles.

V. Description of the Overall Motion Planner

The overall motion planner searches for H-cost short paths, enabled by the MPC-based TilePlan algorithm, on the graphs associated with the multi-resolution cell decompositions described in Section II. However, it is unnecessary to consider H-costs on the entire multi-resolution cell decomposition graph because (a) large cells in Ω^{mr} correspond to coarse information about the environment in the regions associated with those cells, and hence trajectories passing through large cells will need to be refined and/or replanned in future iterations, (b) curvature-constrained paths are guaranteed to exist in rectangular channels wider than a certain threshold width [27], and (c) searches for H-cost short paths are computationally expensive because the numbers of vertices in the lifted graph are large.

Keeping with the multi-resolution idea of using high-accuracy information only locally, the proposed motion planner searches for H-cost short paths on a "partially" lifted graph, such that the vehicle dynamical constraints are considered (via H-costs) only locally. More precisely, for each $J = (j_0, \ldots, j_H) \in V_H$, we define the *projection* $\mathcal{P}_L(J)$ of J onto V_L, by $\mathcal{P}_L(J) := (j_0, \ldots, j_L) \in V_L$, where $L \in \{1, \ldots, H-1\}$. For each $L \in \{1, \ldots, H\}$, we define the set $U_L \subseteq V_L$ by

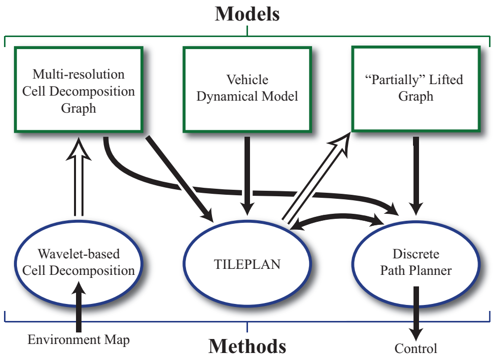


Fig. 4. Schematic illustration of the overall motion planning framework. The hollow arrows indicate modification of a model by a method. The bold arrows indicate dependencies between the various methods and models.

$$\begin{aligned}
U_L \quad := \quad &\big\{(j_0,\ldots,j_L) \in V_L : \mathsf{size}(\mathsf{cell}(j_k)) < \bar{d},\\
&\text{for } k = 0,\ldots,L-1, \text{ and } \mathsf{size}(\mathsf{cell}(j_L)) \leqslant \bar{d}\big\}, \qquad (12)
\end{aligned}$$

where $\bar{d}$ is pre-specified, and $\mathsf{size}(\mathsf{cell}(j_k))$ denotes the size of the cell corresponding to the vertex j_k in the multi-resolution cell decomposition graph. This definition alludes to the previously stated notion of including in the "partially" lifted graph only the cells small enough for the curvature constraints to be significant. The "partially" lifted graph $\tilde{\mathcal{G}}_H = (\tilde{V}_H, \tilde{E}_H)$ is then defined by

$$\begin{aligned}
\tilde{V}_H \quad &:= \quad \cup_{L=1}^{H} U_L \backslash \mathcal{P}_L(U_H), \qquad (13)\\
\tilde{E}_H \quad &:= \quad \cup_{L=1}^{H} \left\{(I,J) : I \in U_L, [I]_1^L = J \in U_{L-1}\right\}. (14)
\end{aligned}$$

As an example clarifying the definitions of $\tilde{V}_H$ and $\tilde{E}_H$, consider the graph associated with the cell decomposition shown in Fig. 5, and let $\bar{d} = 2$ units (the various sizes of the cells shown in Fig. 5 are $1, 2$, and 4 units). Let $H = 2$. According to the above definitions,

$$\begin{aligned}
U_1 \quad &= \quad \{(3,2),(3,5),\ldots,(8,7),\ldots,(9,6),\ldots,(10,8)\},\\
U_2 \quad &= \quad \{(3,9,8),(3,9,6),\ldots,(8,9,3),(8,9,2),\ldots,\\
&\qquad (9,8,10),(9,8,7),\ldots,(10,3,5),\ldots\}
\end{aligned}$$

Note that elements such as $(1,2),(1,5),(2,3),(4,7)$, etc. that are in V_1 do not appear in U_1. Similarly, elements such as $(3,2,1),(3,5,10)$, etc. that are in V_2 do not appear in U_2. The projection $\mathcal{P}_1$ of U_2 onto V_1 is $\mathcal{P}_1(U_2) = \{(3,9),(3,10),(8,9),(8,10),\ldots,(10,3),(10,8)\}$. By (13),

$$\begin{aligned}
\tilde{V}_2 \quad = \quad &\{(3,9,8),\ldots,(10,3,5),\ldots(3,2),(3,5),\\
&(8,6),(8,7),(9,2),(9,6),(10,5),(10,7)\},
\end{aligned}$$

and by (14), the edge set $\tilde{E}_2$ is

$$\begin{aligned}
\tilde{E}_2 \quad = \quad \Big\{ \; &\overbrace{((3,9,8),(9,8,6))\,,((3,9,8),(9,8,7))\,,\ldots,}^{\text{edges common with } E_2}\\
&\underbrace{((3,9,6),(9,6))\,,((3,10,7),(10,7))\,,\ldots}_{\text{edges } (I,J) \text{ of the type } I\in U_2, J\in U_1} \;\Big\}
\end{aligned}$$

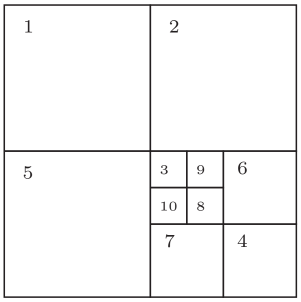


Fig. 5. A sample multi-resolution cell decomposition.

The overall motion planner, schematically illustrated in Fig. 4, then operates as follows. At each iteration, a multi-resolution cell decomposition is first constructed using the wavelet-based scheme outlined in Section II and discussed in detail in [6], [23]. The cells in this decomposition may be categorized according to their sizes into two classes, namely, cells with sizes at most $\bar{d}$, and cells with sizes greater than $\bar{d}$. We define *boundary cells* in the multi-resolution cell decomposition as the cells of sizes at most $\bar{d}$ that have at least one neighboring cell in each of the two previously defined classes (see Fig. 7(a)). A multiple-source, single-goal implementation of the A* algorithm is used to determine the costs of optimal paths in the multi-resolution cell decomposition graph from the vertices associated with each of the boundary cells to the goal vertex. These costs are then used as terminal penalty costs in the execution of the H-cost motion planning algorithm (outlined in Section III and discussed in detail in [7], [23]) on the "partially" lifted graph previously discussed. This H-cost motion planner returns a sequence of cells from the current location to one of the boundary cells, along with an admissible vehicle control input that enables the traversal of this sequence of cells. The vehicle state is advanced by traversing one cell, and the process is repeated until the vehicle reaches the goal.

A. Simulation Results and Further Discussion

We consider a point-mass aircraft navigational model described by

$$\begin{aligned}
\dot{x}(t) &= v(t)\cos\gamma(t)\cos\psi(t),\\
\dot{y}(t) &= v(t)\cos\gamma(t)\sin\psi(t),\\
\dot{z}(t) &= v(t)\sin\gamma(t),\\
\dot{\psi}(t) &= -\frac{q(t)C_{\rm L}(t)}{mv(t)\cos\gamma(t)},\\
\dot{v}(t) &= \left(T(t)-q(v(t))C_{\rm D,0}-KC_{\rm L}^2(t)\right)/m,\\
\dot{\gamma}(t) &= \frac{1}{mv(t)}\big(q(v(t))C_{\rm L}(t)\cos\phi(t)-mg\cos\gamma(t)\big),
\end{aligned}$$

where x, y, and z denote the inertial position coordinates, v denotes the speed, ψ denotes the aircraft heading, γ denotes the flight path angle, $q(v) := \frac{1}{2}\rho v^2 S$ denotes the dynamic pressure, m denotes the mass of the aircraft, and $C_{\rm D,0}$ and K are pre-specified constants. The control inputs are the thrust $T \in [T_{\min}, T_{\max}]$, the lift coefficient $C_{\rm L} \in [C_{\rm L,min}, C_{\rm L,max}]$, and the bank angle $\phi \in [\phi_{\min}, \phi_{\max}]$, where the bounds for the admissible control inputs are pre-specified.

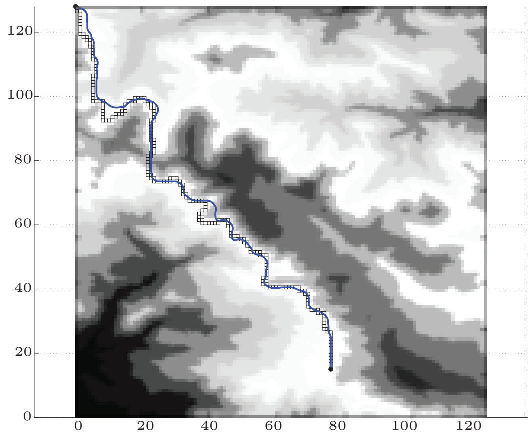


Fig. 6. Result of motion planning simulation using the aircraft navigational model. The channel of cells in black is the result of executing A* algorithm (without vehicle dynamical constraints). The initial position is at the top left corner of the map. The units for the x and y axes are kilometers.

We consider the motion of the aircraft in the horizontal plane, i.e., $\gamma(t) = 0$ and $\dot{\gamma}(t) = 0$, and to this end we set

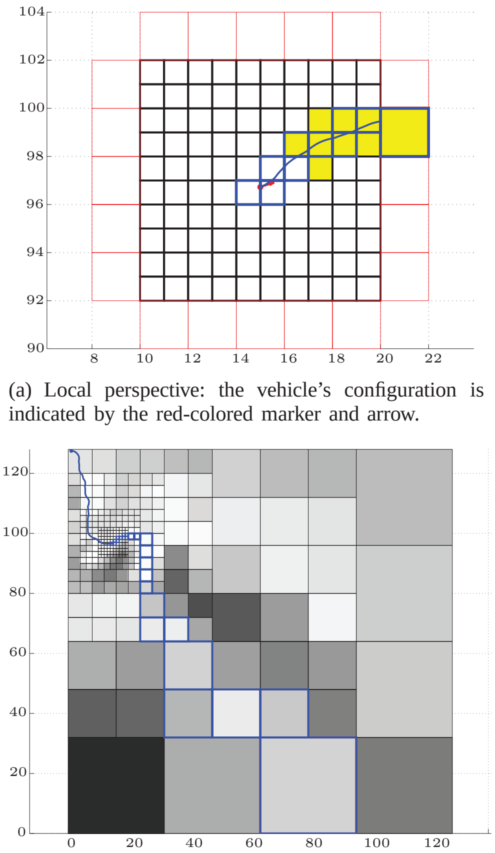
$$C_{\rm L}(t) = mg/\big(q(v(t))\cos\phi(t)\big).$$

Figure 6 shows the result of simulating the proposed motion planner for the aircraft navigational model with the following parameters: $C_{\rm D,0} = 0.02$, $K = 0.04$, $S = 30$ m^2, $mg = 50$ kN, and $v_{\rm cr} = 85$ m/s. The aircraft speed was assumed to be constant, and the (asymmetric) bounds on the bank angle control input were $\phi_{\min} = -45°$ and $\phi_{\max} = 20°$. The objective is to minimize a cost defined on the environment (indicated by regions of different intensities in Fig. 6; the darker regions correspond to higher costs). The details of the implementation of TILEPLAN for this vehicle model are provided in Ref. [23].

Figure 7 illustrates an intermediate iteration in this simulation example. Figure 7(a) shows the cells of size at most $\bar{d}$, with the boundary cells indicated in red. The sequence of cells outlined in blue and the blue-colored curve within this sequence are the results of the H-cost motion planner (the yellow-colored cells indicate the vertices explored during the H-cost search). Figure 7(b) shows the overall multi-resolution cell decomposition at the same iteration; the blue-colored cells indicate the optimal path to the goal from the boundary cell chosen by the H-cost motion planner. The blue-colored curve in Fig. 7(b) indicates the geometric path traversed by the vehicle in previous iterations. In this simulation example, we chose $\bar{d} = 2$ km.

VI. CONCLUSIONS AND FUTURE WORK

We presented a new motion planning framework that combines the efficiency of a wavelet-based multi-resolution cell decomposition scheme with the guarantee of compatibility with vehicle dynamical constraints offered by the H-cost motion planning scheme. The proposed framework makes clear distinctions between the models used for path- and



(a) Local perspective: the vehicle's configuration is indicated by the red-colored marker and arrow.

(b) Global perspective.

Fig. 7. Illustration of an intermediate iteration of the overall motion planner. The x and y axes indicate inertial position coordinates, in kilometers.

motion planning and the algorithms interacting with these models, and its generality is due to the fact that we specify only these interactions (e.g., the specification of TILEPLAN), and not the details of the methods themselves. Whereas we provide specific algorithms to demonstrate the efficacy of the overall framework, in the future these algorithms can easily be replaced to suit specific applications.

References

[1] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion," *IEEE Robotics and Automation Magazine*, pp. 61 – 70, March 2007.

[2] T. Wongpiromsarn, "Formal methods for design and verification of embedded control systems: Application to an autonomous vehicle," Ph.D. dissertation, California Institute of Technology, 2010.

[3] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008. [Online]. Available: http://dx.doi.org/10.1002/rob.20255

[4] P. Tsiotras and E. Bakolas, "A hierarchical on-line path planning scheme using wavelets," in *Proceedings of the European Control Conference*, Kos, Greece, July 2–5 2007, pp. 2806–2812.

[5] D. Jung, "Hierarchical path planning and control of a small fixed-wing UAV: Theory and experimental validation," Ph.D. dissertation, Georgia Institute of Technology, 2007.

[6] R. V. Cowlagi and P. Tsiotras, "Multi-resolution path planning: Theoretical analysis, efficient implementation, and extensions for dynamic environments," in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, GA, 15 – 17 December 2010.

[7] ——, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *IEEE Transactions on Robotics*, 2011, to appear.

[8] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 3, pp. 135–45, September 1986.

[9] R. J. Prazenica, A. J. Kurdila, R. C. Sharpley, and J. Evers, "Multiresolution and adaptive path planning for maneuver of micro-air-vehicles in urban environments," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, 2005, pp. 1–12.

[10] B. J. H. Verwer, "A multiresolution workspace, multiresolution configuration space approach to solve the path planning problem," in *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 2107–12.

[11] S. Behnke, "Local multiresolution path planning," *Lecture Notes in Artificial Intelligence*, vol. 3020, pp. 332–43, 2004.

[12] I. Daubechies, *Ten Lectures on Wavelets*. CBMS-NSF Lecture Notes, 61, SIAM, 1994.

[13] W. L. D. Lui and R. Jarvis, "A pure vision-based approach to topological SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18 – 22 2010, pp. 3784 – 3791.

[14] J. Ilkyun, J. Seewong, and K. Youngouk, "Mobile robot navigation using difference of wavelet SIFT," in *Proceedings of the 2009 Second International Conference on Machine Vision*, 2009, pp. 286 – 292.

[15] J. T. Cho and B. H. Nam, "A study on the fuzzy control navigation and the obstacle avoidance of mobile robot using camera," in *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Conference*, vol. 4, 2000, pp. 2993 – 2997.

[16] M. Shim, J. Kurtz, and A. Laine, "Multi-resolution stereo algorithm via wavelet representations for autonomous navigation," in *Proceedings of the SPIE*, vol. 3723, Orlando, FL, April 1999, pp. 319 – 328.

[17] B. Cipra, "Parlez-vous wavelets?" in *What's Happening in the Mathematical Sciences*. American Mathematical Society, 1994, vol. 2.

[18] M. Yguel, O. Aycard, and C. Laugier, "Wavelet occupancy grids: A method for compact map building," in *Field and Service Robotics, STAR 25*, P. Corke and S. Sukkarieh, Eds. Springer-Verlag, 2006, pp. 219 – 230.

[19] L. Wei and T. F. Fwa, "Characterizing road roughness by wavelet transform," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1869, pp. 152 – 158, 2004.

[20] C. Paulson, S. Ezekiel, and D. Wu, "Wavelet-based image registration," in *Evolutionary and Bio-Inspired Computation: Theory and Applications IV, Proceedings of the SPIE*, T. H. O'Donnel, M. Blowers, and K. Priddy, Eds., vol. 7704, 2010.

[21] D. K. Pai and L.-M. Reissell, "Multiresolution rough terrain motion planning," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 19–33, February 1998.

[22] B. Sinopoli, M. Micheli, G. Donato, and T. J. Koo, "Vision based navigation for an unmanned aerial vehicle," in *Proceedings of the 2001 IEEE Conference on Robotics and Automation*, 2001, pp. 1757–64.

[23] R. V. Cowlagi, "Hierarchical motion planning for autonomous aerial and terrestrial vehicles," Ph.D. dissertation, Georgia Institute of Technology, 2011.

[24] E. Rippel, A. Bar-Gill, and N. Shimkin, "Fast graph-search algorithms for general aviation flight trajectory generation," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 801–811, July-August 2005.

[25] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[26] D. P. Bertsekas and I. B. Rhodes, "On the minimax reachability of target sets and target tubes," *Automatica*, vol. 7, pp. 233–247, 1971.

[27] S. Bereg and D. Kirkpatrick, "Curvature-bounded traversals of narrow corridors," in *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, Pisa, Italy, 2005, pp. 278–287.