Multiresolution Path Planning Via Sector Decompositions Compatible to On-Board Sensor Data

Efstathios Bakolas[*] and Panagiotis Tsiotras[†]

In this paper we present a hybrid local-global path planning scheme for the problem of operating a moving agent inside an unknown environment in a collision-free manner. The path planning algorithm is based on information gathered on-line by the available on-board sensor devices. The solution minimizes the total length of the path with respect to a metric that includes actual path length along with a risk-induced metric. We use a multi-resolution cell decomposition of the environment in order to solve the path-planning problem using the wavelet transform in conjunction with a conformal mapping to polar coordinates. By performing the cell decomposition in polar coordinates we can naturally incorporate sector-like cells that are adapted to the data representation collected by the on-board sensor devices. Simulations are presented to test the efficiency of the algorithm using a non trivial scenario.

Keywords: Mobile agents, wavelet decomposition, conformal mapping, sensor devices, shortest path, collision avoidance, path planning.

I. Introduction

Path planning inside a partially known environment is a topic that has been under intense investigation for many years. A typical application involves, for instance, an unmanned aerial vehicle (UAV) that operates in a hostile environment, with the mission to identify an enemy target, while avoiding areas of high risk (which may not be known a priori). A nonconservative approach for solving this problem is to use a replanning algorithm that processes the information about potential threats gathered solely by on-board sensors. In the absence of global knowledge of the environment, the vehicle will design a route and navigate towards its destination (and away from the areas of high risk) using only information gathered by the on-board sensor devices.

Different sensor devices (cameras, radars, laser scanners, satellite imagery, etc) provide information about the environment at different ranges and resolution levels. Typically, on-board sensors have higher resolution but their range is limited to the vicinity of the vehicle. Off-board sensors, on the other hand, provide non-local (even global) information about the environment, but at coarser resolutions. In this work we assume that this is always the case.

A natural question that arises in the context of path-planning is how to blend and process the data collected by different sensors (both on-board and off-board) at different ranges and resolutions so as to achieve *computationally efficient* replanning in order for the agent to avoid obstacles or popup threats.

Several multi-resolution or hierarchical algorithms have been proposed in the literature for path planning,[1–7] most of them using quadtrees. A recent path planning algorithm, based on a more flexible multiresolution scheme using the wavelet transform has been developed in Ref. [8]. Wavelets are powerful tools for

[*] E. Bakolas is a Ph.D. candidate at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA, Email: gth714d@mail.gatech.edu

[†] P. Tsiotras is a Professor at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA, Email: tsiotras@gatech.edu, Associate Fellow AIAA.

approximating the environment (both locally and globally) when the data is provided at different resolutions, and possibly at different locations. This is due to the ability of the wavelet transform to blend the available nonhomogeneous data coming from the different sensors, and provide decompositions and reconstructions of the environment using data at several, district levels of fidelity (multiresolution scheme). The wavelet approximation scheme of Ref. [8] provides the designer with the flexibility to focus the limited on-board computational resources of the vehicle on the part of the path (spatial and temporal) that needs it most.

The key idea behind the multiresolution path-planning scheme in Ref. [8] is to use the properties of wavelets to locate accurately the obstacle boundaries *only* when these obstacles are in the vicinity of the agent. Obstacle boundaries far away from the vehicle are inconsequential for the short-term planning of the route to be followed. These obstacle boundaries will be accurately approximated only when (or if) the vehicle gets close to these obstacles. On the other hand, quadtree decomposition schemes cannot distinguish between obstacles based on their distance from the agent – at least not without any extensive modifications.[9,10] Therefore, these path-panning schemes spend precious computational resources to represent in high detail parts of the environment that are not essential to the *immediate* reaction of the vehicle. In addition, the adjacency matrix of the resulting cell decomposition can be computed directly from the wavelet coefficients thus speeding up the process even further.[11] This algorithm has been implemented in a microcontroller and tested in a HIL environment for the autonomous navigation and guidance of a small UAV.[12–14]

Information about the environment for path-planning problems is obtained using either on-board or off-board sensors. Some of this information is provided off-line and some is gathered on-line. Furthermore, most typical sensor devices provide sector-like representations of the environment (see Fig. 1). This type of information is not in the most efficient form for the majority of planning algorithms, which employ rectangle or square cell approximations, typically using quadtrees.[6,8,15] Such approximations are not compatible to the sector-based representations obtained by most sensor devices.

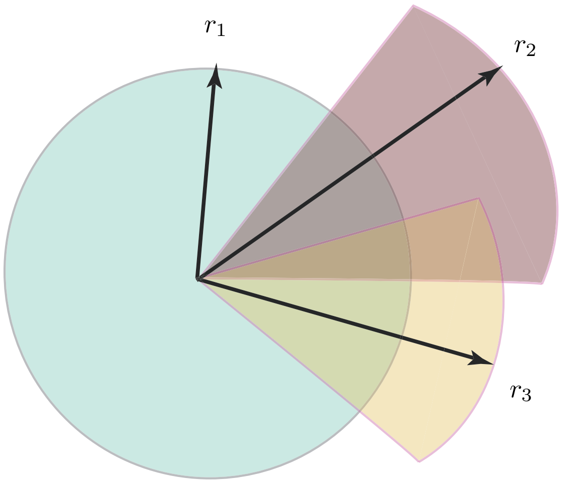


Figure 1. Sensors have different ranges, fields of view and resolution. Ideally, the algorithm that processes this data should conform to this topology.

In this paper we extend the results of Ref. [8], by employing a conformal mapping to devise a hybrid local/global path planning algorithm using sector cell decompositions instead of decompositions that employ only rectangular or square cells. Sector cells are compatible to the on-board sensors, and thus process the data more efficiently, in a manner that does not contradict its original sector-based form. We provide approximations with special localized attributes by combining efficiently data from sensors of different resolutions and ranges. Furthermore, in our previous work[8] the whole environment was assumed to be known a priori and the wavelet approximation scheme allowed us to plan the path using only a small fraction of the available information. The result is a reduced number of computations, which can be handled by the available computational resources on-board the vehicle. Contrary to Ref. [8], the proposed methodology in this paper employs on-line data of the agent's immediate environment as it is obtained by the on-board sensors. This approach is the most natural way to deal with the navigation problem of an autonomous vehicle operating in an unexplored environment, for which no prior knowledge is available.

The proposed path planning scheme is computationally efficient, suitable for *on-line* implementation and combines short-term tactics (reaction to unforeseen threats) with long-term strategy (planning towards the ultimate goal) based on the available data.

II. From Rectangular to Sector Cell Decompositions

The goal of the proposed path planning scheme is to find a sequence of points in the world $\mathcal{W}$ (this is the agent's operating space and includes the obstacle space $\mathcal{O}$) which should be visited by the agent in order to reach the final destination x_f. The initial state x_0 is assumed to be known, but the final state x_f may or may not be accurately known a priori. All the points of the sequence should lie in a polygonally connected space $\mathcal{F} = \mathcal{W}\backslash\mathcal{O}$, the free configuration space. The space $\mathcal{F}$ contains all admissible states of $\mathcal{W}$. The requirement that $\mathcal{F}$ be polygonally connected is needed so that the polygonal line that connects the points of the sequence lies completely inside $\mathcal{F}$.

It is assumed that the available information of the surrounding area of the agent is given by the superposition of circular or conical sectors obtained by different sensor devices as depicted in Fig. 1. This information about $\mathcal{W}$ needs to be processed by the path planner to compute a collision free path. In order to do so, the path-planning algorithm typically computes a cell decomposition of the environment. As can be easily observed by Fig. 2, if rectangular cells are used (as with any quadtree-based approach) the algorithm may waste computational resources by subdividing the cells in order to resolve the sector boundaries.

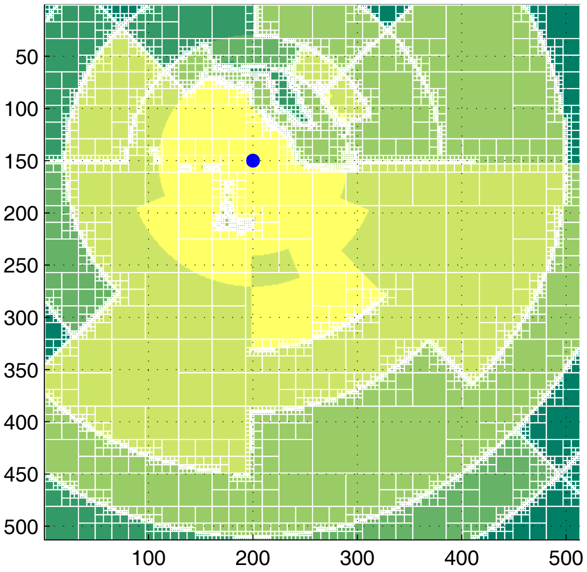


Figure 2. A cell decomposition based on the available sector approximation of the environment obtained by the on-board sensor devices of the agent (denoted with the blue dot). In order to resolve the geometry of the arc-boundary of each sector the standard quadtree algorithm generates a large number of cells close to the boundaries of these arcs.

A simple way to overcome this difficulty is to employ a conformal mapping to map the sector cells to rectangular cells in a new coordinate system. The latter approach is proposed in this paper. The motivation for this idea is simple. Let us assume that $\mathcal{R}$ is a sector domain in the (x, y) plane specified by the radii $r_{\min}$ and $r_{\max}$ and the angles $\theta_{\min}$ and $\theta_{\max}$. Mapping this domain to the (r, θ) plane using the polar transformation one obtains a rectangular domain $\mathcal{R}'$ defined by the same radii and angles. If the angle varies from $\theta_{\min} = 0$ up to $\theta_{\max} = 2\pi$ the whole annulus cut defined by the radii $r_{\min}$ and $r_{\max}$ is mapped to a rectangular cell in the (r, θ) plane as shown in Fig. 3.

More precisely, recall that the polar coordinates r, θ are related to x and y via the equations

$$x = r\cos\theta, \qquad y = r\sin\theta, \tag{1}$$

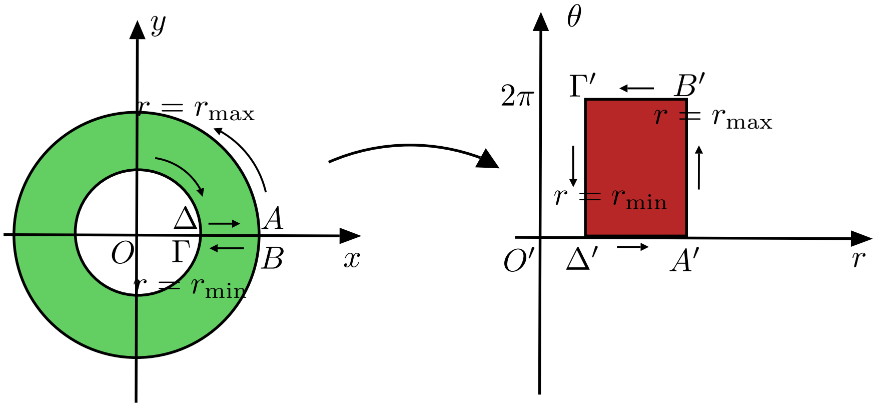


Figure 3. An annulus cut in the (x, y) plane is mapped to a rectangle in the (r, θ) plane via a polar (conformal) mapping.

where the Jacobian of the transformation is given by

$$J = \frac{\partial(x, y)}{\partial(r, \theta)} = r,$$

and thus $J > 0$ provided that $r > 0$. Thus, the inverse transformation $(x, y) \mapsto (r, \theta)$ exists for $r > 0$.

The closed curve defined by the union of the circles $C_1 = \{(x, y) \in \mathbb{R}^2,\ x^2 + y^2 = r_{\min}^2\}$, $C_2 = \{(x, y) \in \mathbb{R}^2,\ x^2 + y^2 = r_{\max}^2\}$ and the line segment $L = \{(x, 0) \in \mathbb{R}^2,\ r_{\min} \leq x \leq r_{\max}\}$ (traversed twice), maps to a rectangle $r = r_{\min}, r = r_{\max}, \theta = 0, \theta = 2\pi$ in the (r, θ) plane. Alternatively, the area inside a rectangle defined by $r = r_{\min}, r = r_{\max}, \theta = \theta_{\min}, \theta = \theta_{\max}$ where $0 \leq \theta_{\min} \leq\ \theta_{\max} \leq 2\pi$ is mapped via the inverse transformation to the intersection of two sectors defined by $r_{\min}, \theta_{\min}$ and $r_{\max}, \theta_{\max}$ respectively in the (x, y) plane.

Thus, given a sector cell decomposition in the (x, y) plane, a rectangle cell decomposition can be obtained in the (r, θ) plane via the polar coordinate transformation (1). Conversely, if we have a multiresolution approximation of the rectangular domain in the (r, θ) plane, then by applying the inverse mapping, the rectangular area can be approximated by a collection of cells forming a sector domain in the (x, y) plane, as seen in Fig. 4.

III. World Space in the New Coordinate System

Let f be a function $f : \mathfrak{Domain}(f) = \mathcal{W} \mapsto \mathbb{R}$ and let $\mathfrak{Image}_f(\mathcal{W}) = \{f(x) : x \in \mathcal{W}\} \subseteq \mathbb{R}$. In order to map the set $\mathcal{W} \times \mathfrak{Image}_f(\mathcal{W})$ to the polar coordinate system we proceed as follows.

First, we discretize $\mathcal{W}$ using a uniform grid of dimension $2^N \times 2^N$. For each point (x_i, y_i), $(1 \leq i \leq 2^N)$ we compute the corresponding (r_i, θ_i) point in polar form using the following equations:

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}, \qquad \theta_i = \operatorname{atan2}(y_i - y_0, x_i - x_0), \tag{2}$$

where $\mathsf{x}_0 = (x_0, y_0)$ is the agent's current position and (r_i, θ_i) is the distance and angle position with respect to the agent. Let $\phi_{\mathsf{x}_0} : \mathbb{R}^2 \mapsto \mathbb{R} \times S^1$ be the transformation in (2). We can then associate to each pair (r_i, θ_i) the function value $f(x_i, y_i)$, that is, $f'_{\mathsf{x}_0}(r_i, \theta_i) = f \circ \phi_{\mathsf{x}_0}^{-1}(r_i, \theta_i) = f(x_i, y_i)$.

The next step is to obtain a multiresolution (rectangular) cell approximation of the function f' in $\mathcal{W}'$. As we shall see in the next section, the wavelet transform provides us with such a multiresolution cell based approximation in any rectangular domain. Based on this cell decomposition we can proceed with the design of our path planning algorithm working entirely in the $\mathcal{W}'$ domain by applying the wavelet-based path planning algorithm of Ref. [8]. The details of this approach are given in the next section. Before proceeding, we familiarize the reader with the basic concepts of the 2-D wavelet transform and elucidate the way we use it in order to obtain multiresolution, piecewise constant, approximations of a function over its domain.

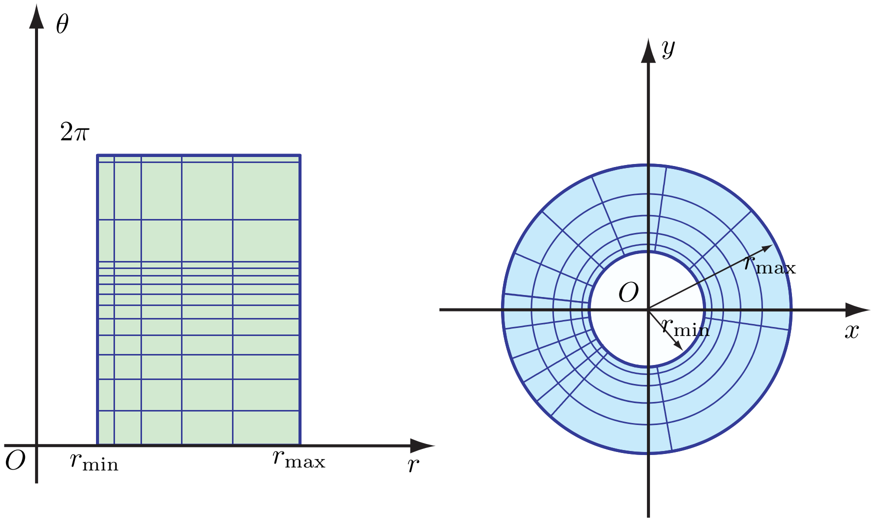


Figure 4. A multiresolution approximation of the rectangular domain in (r, θ) system defined by the radii $r_{\min}$ and $r_{\max}$ under the inverse conformal mapping gives a multiresolution sector approximation of an annulus cut defined by the same radii.

IV. A Multiresolution Decomposition Scheme

IV.A. The 2D wavelet transform

The idea behind the theory of the wavelet transform is to represent a function $f \in \mathcal{L}^2(\mathbb{R})$ as a summation of elementary basis functions $\phi_{J,k}$ and $\psi_{j,k}$ as follows

$$f(x) = \sum_{k\in\mathbb{Z}} a_{J,k}\phi_{J,k}(x) + \sum_{j\geq J}\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(x), \tag{3}$$

where $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$ and $\psi_{j,k} = 2^{j/2}\psi(2^j x - k)$. In the ideal case, both $\phi(x)$ (scaling function) and $\psi(x)$ (mother wavelet) have compact support or they decay very fast outside a small interval so they can capture localized features of f. The first summation in (3) gives a low resolution, or coarse, approximation of f. The second term in (3) gives the difference (details) between the original function and its low resolution approximation. For example, when analyzing a signal at the coarsest level (low resolution) only the general, most salient, features of the signal will be revealed. The index j denotes the resolution level. For each increasing index j, a higher, or finer resolution term is added, which adds more and more details. The expansion (3) thus reveals the properties of f at different levels of resolution.[16–18]

This idea can be readily extended to the two-dimensional case by introducing the following families of functions

$$\Phi_{j,k,\ell}(x,y) = \phi_{j,k}(x)\phi_{j,\ell}(y) \tag{4}$$

$$\Psi^1_{j,k,\ell}(x,y) = \phi_{j,k}(x)\psi_{j,\ell}(y) \tag{5}$$

$$\Psi^2_{j,k,\ell}(x,y) = \psi_{j,k}(x)\phi_{j,\ell}(y) \tag{6}$$

$$\Psi^3_{j,k,\ell}(x,y) = \psi_{j,k}(x)\psi_{j,\ell}(y) \tag{7}$$

Given a function $f \in \mathcal{L}^2(\mathbb{R}^2)$ we can then write

$$f(x,y) = \sum_{k,\ell\in\mathbb{Z}} a_{J,k,\ell}\Phi_{J,k,\ell}(x,y) + \sum_{i=1}^{3}\sum_{j\geq J}\sum_{k,\ell\in\mathbb{Z}} d^i_{j,k,\ell}\Psi^i_{j,k,\ell}(x,y), \tag{8}$$

where, for the case of orthonormal wavelets the approximation coefficients are given by[a]

$$a_{j,k,\ell} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\phi_{j,k,\ell}(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{9}$$

and the detail coefficients by

$$d^i_{j,k,\ell} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\psi^i_{j,k,\ell}(x,y)\,\mathrm{d}x\,\mathrm{d}y. \tag{10}$$

The key property of wavelets used in this paper is the fact that the expansion (8) induces the following decomposition of $\mathcal{L}^2(\mathbb{R}^2)$

$$\mathcal{L}^2(\mathbb{R}^2) = \mathcal{V}_J \oplus \mathcal{W}^{detail}_J \oplus \mathcal{W}^{detail}_{J+1} \oplus \cdots \tag{11}$$

where $\mathcal{V}_J = \overline{\mathrm{span}}\{\phi_{J,k,\ell}\}_{k,\ell\in\mathbb{Z}}$ and similarly $\mathcal{W}^{detail}_j = \overline{\mathrm{span}}\{\psi^1_{j,k,\ell}, \psi^2_{j,k,\ell}, \psi^3_{j,k,\ell}\}_{k,\ell\in\mathbb{Z}}$ for $j \geq J$.

By using the Haar family of wavelets, each scaling function $\phi_{j,k}(x)$ and wavelet function $\psi_{j,k}(x)$ in the Haar system is supported on the dyadic interval $I_{j,k} \overset{\triangle}{=} [k/2^j, (k+1)/2^j]$ of length $1/2^j$ and does not vanish in this interval.[16,19] Subsequently, and via the tensor product in (4), we may associate the functions $\Phi_{j,k,\ell}$ and $\Psi^i_{j,k,\ell}$ $(i = 1,2,3)$ in the 2D case with the square cell $c^j_{k,\ell} \overset{\triangle}{=} I_{j,k} \times I_{j,\ell}$.

IV.B. Wavelet decomposition of the risk measure

Without loss of generality, in the sequel we take $\mathcal{W} = [0,1]\times[0,1]$. Using the conformal mapping of Section II, $\mathcal{W}$ is mapped to $\mathcal{D}'$ in the polar coordinate system. The new domain, however, is not rectangular. Assuming without loss of generality (renormalize $\mathcal{W}$, if necessary) that $\mathcal{D}' \subset [0,1]\times[0,1]$, we let $[0,1]\times[0,1]\backslash\mathcal{D}'$ lie completely inside the obstacle configuration space $\mathcal{O}'$ in the (r,θ) domain. Thus, by adding this artificial obstacle corresponding to the boundary of $\mathcal{D}'$ we can assume that the world $\mathcal{W}'$ in the polar coordinate system to be again $\mathcal{W}' = [0,1]\times[0,1]$.

We describe $\mathcal{W}'$ using a discrete (fine) grid of $2^N \times 2^N$ dyadic points. The finest level of resolution $J_{\max}$ is therefore bounded by N. It follows from the previous discussion that the Haar wavelet decomposition of a function f' defined over $\mathcal{W}'$ at resolution level $J \geq J_{\min}$, given by,

$$f'(r,\theta) = \sum_{k,\ell=0}^{2^{J_{\min}}-1} a_{J_{\min},k,\ell}\,\Phi_{J_{\min},k,\ell}(r,\theta) + \sum_{i=1}^{3}\sum_{j=J_{\min}}^{J-1}\sum_{k,\ell=0}^{2^j-1} d^i_{j,k,\ell}\Psi^i_{j,k,\ell}(r,\theta) \tag{12}$$

induces a cell decomposition of $\mathcal{W}'$ of square cells of size $1/2^J \times 1/2^J$ in the (r,θ) coordinate system.

Assume now that we are given a function $\mathsf{rm} : \mathcal{W} \mapsto [0,1]$ that represents the "risk measure" at the location $\mathsf{x} = (x,y)$. For instance, one may choose

$$\mathsf{rm}(\mathsf{x}) = \begin{cases} (d_{\max} - \min_{\mathsf{y}\in\mathcal{O}} \|\mathsf{x}-\mathsf{y}\|_2)/d_{\max}, & \text{if } \mathsf{x}\in\mathcal{F}, \\ 1, & \text{if } \mathsf{x}\in\mathcal{O}, \end{cases} \tag{13}$$

where $d_{\max} \overset{\triangle}{=} \max_{\mathsf{x}\in\mathcal{F}} \min_{\mathsf{y}\in\mathcal{O}} \|\mathsf{x}-\mathsf{y}\|_2$. Alternatively, one may think of rm as the probability that $(x,y)\in\mathcal{O}$. Given the current location of the vehicle x_0 the function rm induces a similar function over $\mathcal{W}'$, that is, $\mathsf{rm}'_{\mathsf{x}_0} : \mathcal{W}' \mapsto [0,1]$ via $\mathsf{rm}'_{\mathsf{x}_0}(r,\theta) = \mathsf{rm} \circ \phi^{-1}_{\mathsf{x}_0}(r,\theta)$.

All points within range ρ from the current location of the agent, when expressed in the (r,θ) system, are given by

$$\mathcal{N}(\rho) \overset{\triangle}{=} \{(r,\theta)\in\mathcal{W}' : |r| \leq \rho,\ \theta\in[\theta_{\min},\theta_{\max}]\}. \tag{14}$$

The region that corresponds to $\mathcal{N}(\rho)$ in the (r,θ) coordinate system is a strip of width equal to ρ and height $\theta_{\max} - \theta_{\min}$. For simplicity, in the sequel we will assume that $\theta_{\min} = -\pi$ and $\theta_{\max} = \pi$.

[a] In the more general case of biorthogonal wavelets projections on the space spanned by the dual wavelets and dual scaling functions should be used in (9) and (10).

Suppose now that we are given the desired levels of resolution of $\mathcal{W}'$ as $J_{\min} \leq j \leq J_{\max}$ where $J_{\min}, J_{\max} \in \{1, \ldots, N\}$, with corresponding ranges ρ_j from the agent's current location. By this we mean that we wish all points $(r, \theta) \in \mathcal{N}(\rho_{J_{\max}})$ to be described by resolution $J_{\max}$, all points $(r, \theta) \in \mathcal{N}(\rho_{j-1}) \backslash \mathcal{N}(\rho_j)$ to be described by resolution j, where $J_{\min} < j \leq J_{\max}$, and all points $(r, \theta) \notin \mathcal{N}(\rho_{J_{\min}+1})$ to be described by resolution $J_{\min}$. Since we require finer resolution closer to the agent we assume, of course, that $\rho_{j-1} > \rho_j$.

The choice of $J_{\max}$ is dictated by the requirement that at this level all cells should be resolved into either free or occupied cells. The choice of $J_{\min}$ as well as the values of ρ_j are typically dictated by the sensor specifications and/or the on-board computational resources.

We obtain the distinct resolution levels at the given required distances from the current location of the agent by applying the Haar wavelet transform to $\mathsf{rm}'_{\mathsf{x}_0}$[b]. To this end, let $\mathcal{I}(j) \stackrel{\triangle}{=} \{0, 1, \ldots, 2^j - 1\}$ and let

$$\mathcal{K}(j) \stackrel{\triangle}{=} \{k \in \mathcal{I}(j) : I_{j,k} \cap (0, \rho_j] \neq \varnothing\},$$
$$\mathcal{L}(j) \stackrel{\triangle}{=} \{\ell \in \mathcal{I}(j) : I_{j,\ell} \cap [\theta_{\min}, \theta_{\max}] \neq \varnothing\}.$$

The wavelet decomposition of $\mathsf{rm}'_{\mathsf{x}_0}$, given by

$$\mathsf{rm}'_{\mathsf{x}_0}(r, \theta) = \sum_{k,\ell \in \mathcal{I}(J_{\min})} a_{J_{\min},k,\ell}\, \Phi_{J_{\min},k,\ell}(r, \theta) + \sum_{i=1}^{3} \sum_{j=J_{\min}}^{J_{\max}-1} \sum_{\substack{k \in \mathcal{K}(j) \\ \ell \in \mathcal{L}(j)}} d^i_{j,k,\ell} \Psi^i_{j,k,\ell}(r, \theta) \tag{15}$$

induces, via a slight abuse of notation, the following cell decomposition on $\mathcal{W}'$

$$\mathcal{C}_d = \Delta C_d^{J_{\min}} \oplus \cdots \oplus \Delta C_d^{J_{\max}} \tag{16}$$

where, ΔC_d^j is a union of square cells $c^j_{k,\ell}$ in the (r, θ) domain of dimension $1/2^j \times 1/2^j$. Furthermore, by applying the inverse mapping based on the analysis of Section II we obtain a sector decomposition $\mathcal{S}_d$ of $\mathcal{W}$, where $\mathcal{S}_d$ is defined as

$$\mathcal{S}_d = \Delta S_d^{J_{\min}} \oplus \cdots \oplus \Delta S_d^{J_{\max}} \tag{17}$$

where, $\Delta S_d^J = \phi_{\mathsf{x}_0}^{-1} \circ \Delta C_d^J$, with $J_{\min} \leq J \leq J_{\max}$.

V. Sector-based Multiresolution Path Planning

The proposed multiresolution path planning algorithm takes place completely in the (r, θ) coordinate system. The agent in this system remains at the origin at all times. Let $\mathsf{x}' = (r, \theta)$ denote the coordinate vector of a point in $\mathcal{W}'$ in this coordinate system. Assuming that $\mathsf{x}' = (0, 0)$ at $t = t_0$, we construct using the approach of Section IV, a cell decomposition $\mathcal{C}_d(t_0)$ of $\mathcal{W}'$. Denote by $\mathcal{G}(t_0)$ be the corresponding connectivity graph. The adjacency relationships of $\mathcal{G}(t_0)$ can be quickly computed *directly* from the non-zero indices of the wavelet coefficients in (15). This property is one of the main advantages of this approach, completely bypassing usual quadtree decompositions.[11,12]

On the graph $\mathcal{G}(t_0)$ we impose transition penalties from each node to the next (e.g., Euclidean distance) as well as node penalties for visiting a specific node (e.g., the risk measure). The former penalty depends on the departing node in the edge, while the latter is independent of the departing node. Additional penalties are possible depending on the problem at hand. These node and transition penalties can be weighted appropriately to reflect the specific agent's mission objectives as it navigates inside the environment. The particular choice of these penalties and weights obviously affects the final path. Additional details can be found in Ref. [8].

Using Dijkstra's algorithm (or any other similar algorithm) we then find a *tentative* path $\mathcal{P}(t_0)$ in $\mathcal{G}(t_0)$ of free and mixed nodes which connects the initial node to the final node in $\mathcal{G}(t_0)$. The path $\mathcal{P}(t_0)$ is therefore an ordered sequence of nodes

$$\mathcal{P}(t_0) = (v^1_1, v^1_2, \cdots, v^1_{\ell_1 - 1}, v^1_{\ell_1} = v^1_f). \tag{18}$$

[b] For embedded implementation it is advisable to use the integer Haar wavelet transform instead, especially in conjunction with a lifting scheme. This can significantly increase the execution speed and reduce the memory requirements; see also Ref. [12, 14].

where $v_i^1 = \mathsf{node}_{\mathcal{G}(t_0)}(\mathsf{x}_i')$ and $\mathsf{x}_i' = \mathsf{cell}_{\mathcal{G}(t_0)}(v_i^1)$ is a representative, arbitrary point of the cell that corresponds to node v_i^1. For simplicity we will assume that x_i' is the center of the cell.

The first cell in the sequence (18) is the cell that contains the origin and the final cell in the sequence is the cell that contains the final destination (r_f, θ_f) provided that such a sequence exists. Since we impose a high resolution decomposition inside $\mathcal{N}(\rho_{J_{\max}})$ it is natural to assume that the first two cells in $\mathcal{C}_d(t_0)$ corresponding to v_1^1 and v_2^1 are free for a feasible path. The agent subsequently moves from $\mathsf{cell}_{\mathcal{G}(t_0)}(v_1^1)$ to $\mathsf{cell}_{\mathcal{G}(t_0)}(v_2^1)$. This means that only the points x_1' and x_2' will actually be visited by the agent. In order to find the subsequent points of the actual path to be followed we apply a continuous replanning scheme. This is accomplished by constructing a cell decomposition $\mathcal{C}_d(t_k)$ at each next time step $t_k > t_0,\ (k = 1, 2, \ldots)$, and by inserting in the list of the visiting points only the point $\mathsf{x}_k' = \mathsf{cell}_{\mathcal{G}(t_k)}(v_2^k)$ which corresponds to the second node of the tentative path $\mathcal{P}(t_k)$. We repeat the process until the goal is inside the second cell of $\mathcal{P}(t_k)$.

Since the approach does not eliminate the possibility that the same point may be revisited during the subsequent replanning, thus resulting in an endless loop, we overcome this issue by keeping a list of all points already visited and by removing the corresponding nodes form the graph $\mathcal{G}(t_k)$ at each time step t_k. A pseudo-code implementation of the above algorithm is given in Fig. 5. For more details of this multiresolution path planning algorithm, the reader may refer to Ref. [8].

```
BEGIN PATH PLANNING ALGORITHM
{
     $\mathsf{x}_0' = 0$;
     $i = 0$;
     $\mathsf{x}_i \longleftarrow \mathsf{x}_0$;
     $\{\mathsf{L}_{\mathsf{Visited}}\}$ = $\{\varnothing\}$;
     (while $\|\mathsf{x}_f - \mathsf{x}_i\| > \epsilon$)
     {
       compute $\mathsf{rm}(\mathsf{x}, i)$ for all $\mathsf{x} \in \mathcal{W}$;
       compute $\mathsf{rm}(\mathsf{x}', i)$ for all $\mathsf{x}' \in \mathcal{W}'$ via the mapping $\phi_{\mathsf{x}_0}$;
       construct $\mathcal{C}_d(i)$ on $\mathcal{W}'$ topology ;
       construct $\mathcal{G}(i) = (E(i), V(i))$;
       (if $\mathsf{L}_{\mathsf{Visited}}$ is nonempty)
         for $v \in V(i)$
           $\mathsf{x}'_v(i) = \mathsf{cell}_{\mathcal{G}(i)}(v)$;
           if $\mathsf{x}_v(i) \in \mathsf{L}_{\mathsf{Visited}}$
             extract $v$ from $V(i)$;
             for all $u$ adjacent to $v$
               remove $(v, u)$ from $E(i)$;
           end if;
       end if;
       $v_1^i \longleftarrow \mathsf{node}_{\mathcal{G}(i)}(\mathsf{x}'_0)$;
       $v_f^i \longleftarrow \mathsf{node}_{\mathcal{G}(i)}(\mathsf{x}'_f)$;
       $\mathcal{P}(i) \longleftarrow$ Dijkstra$(v_1^i, v_f^i, V(i), E(i))$;
       if $\mathcal{P}(i) = \{\varnothing\}$
         report FAILURE;
         break;
       $\mathsf{x}'_i(1) = \mathsf{cell}_{\mathcal{G}(i)}(v_1^i)$;
       $\mathsf{x}'_i(2) = \mathsf{cell}_{\mathcal{G}(i)}(v_2^i)$;
       $\{\mathsf{L}_{\mathsf{Visited}}\} \longleftarrow \{\mathsf{L}_{\mathsf{Visited}}\} \bigoplus \{\mathsf{x}'_i(1), \mathsf{x}_i(2)\}$;
       $\mathsf{x}'_{i+1} \longleftarrow \mathsf{x}'_i(2)$;
       $i \longleftarrow (i + 1)$;
       compute $\mathsf{x}_{i+1}$ via inverse mapping $\phi_{\mathsf{x}_0}^{-1}$;
       $\mathsf{x}_0 \longleftarrow \mathsf{x}_{i+1}$;
     }
}
END PATH PLANNING ALGORITHM
```

Figure 5. Pseudo-code implementation of proposed multiresolution path planning scheme.

VI. Simulation Results

We next present the results from the numerical simulation of the proposed algorithm using a non-trivial scenario. The environment is assumed to be square of dimension 512×512 units (pixels). Hence $N = 9$ is the finest resolution. For simplicity, only two levels of resolution have been chosen to represent the environment. Inside a circle of radius of 100 unit cells, we employ a high resolution approximation; outside this area we employ a low resolution approximation of $\mathcal{W}'$.

The original environment $\mathcal{W}$ (in the (x, y) coordinate system) , shown in Fig. 6, is an actual topographic (elevation) map from the US.[20] The initial and final positions of the agent are also shown in this figure. The objective is for the agent to follow a path from A to B while avoiding the areas of dark color. Areas with bright colors in Fig. 6 correspond to areas of low risk and darker colors correspond to areas of high risk that should be avoided. Solving the path-planning problem on-line at this resolution can be computationally prohibitive.

In order to apply the proposed multiresolution scheme we choose six distinct risk measure levels $M_1, \ldots, M_6$ between 0 and 1. The two levels with the highest values ($M_5 = 0.9$ and $M_6 = 1$) denote the obstacle space; the rest four levels $M_1, \ldots, M_4$ denote feasible states. Level M_1 represents definitely unoccupied cells (that is, with probability 100%), and thus characterizes the most desirable locations.

The results from the proposed multiresolution path-planning algorithm using a fine resolution level $J_{\max} = 5$, and a low resolution at level $J_{\min} = 3$ are shown in Fig. 7. Specifically, Fig. 7 shows the evolution of the path at different time steps as the agent moves to the final destination. At each instant of time, the solid line shows the path followed by the agent up to that instant in time and the dashed line shows the predicted path the agent will follow given the current information. Notice that the predicted path may be different than the ultimate path actually followed by the agent. This is because constant re-planning provides additional information to the agent as it approaches areas that initially may have been considered safe.

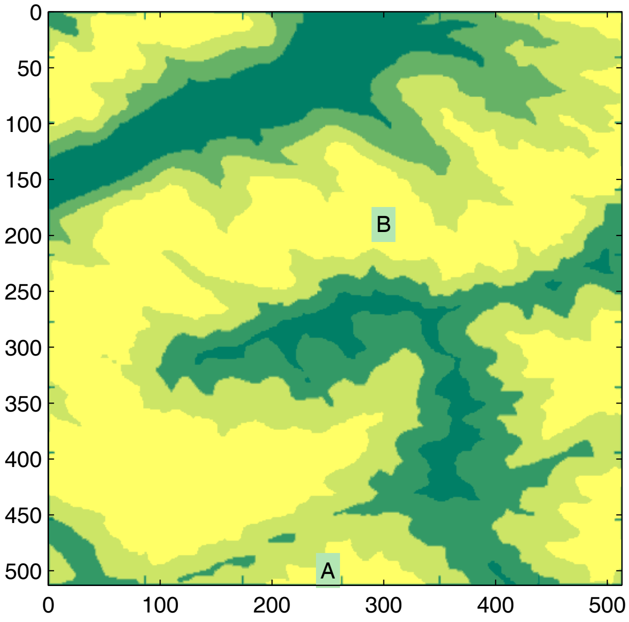


Figure 6. The original risk measure function. Dark green denotes areas of high risk, whereas yellow denotes areas of low risk. The letters A and B indicate the initial and final configurations, respectively.

Conclusions

In this paper we have proposed an *on-line* hierarchical path planning scheme for navigating an autonomous agent inside an unknown environment based on information obtained by its available on-board sensor devices. At each step, the algorithm computes a sector-based multiresolution decomposition of the environment, which is adapted to the range and resolution of the data obtained by the on-board sensors using the wavelet transform in conjunction with a conformal mapping to new (polar) coordinates. This multiresolution approach thus allows the agent to blend information arising from different sensors at different ranges and resolutions. By keeping only the information from the processed data that is necessary to avoid the obstacles, the agent invests the available computational resources where and when is needed most. That is, typically in the vicinity of the current position where a more accurate representation of the environment is needed. The algorithm is especially suitable for embedded autonomous navigation of vehicles with limited on-board computational resources. Extensions of the proposed methodology, aiming at further improving the efficiency of the algorithm, will be addressed in the future.

Acknowledgement: This work has been supported in part by NSF (award no. CMS-0510259) and ARO (award no. W911NF-05-1-0331).

References

[1] Brooks, R. and Lozano-Perez, T., "A Subdivision Algorithm on Configuration Space for Findpath with Rotation," *IEEE Trans. Syst., Man, Cybern.*, Vol. 15, 1985, pp. 224–233.

[2] Zhu, D. and Latombe, J., "New Heuristic Algorithms for Efficient Hierarchical Path Planning," *IEEE Transactions on Robotics and Automation*, Vol. 7, 1991, pp. 9–20.

[3] Noborio, H., Naniwa, T., and Arimoto, S., "A Quadtree-Based Path Planning Algorithm for a Mobile Robot," *Journal of Robotic Systems*, Vol. 7, No. 44, 1990, pp. 555–574.

[4] Godbole, D., Samad, T., and Gopal, V., "Active Multi-Model Control for Dynamic Maneuver Optimization of Unmanned Air Vehicle," *IEEE Proceedings of the IEEE International Conference of Robotics and Automation*, San Francisco, CA, 2000.

[5] Pai, D. and Reissell, L., "Multiresolution Rough Terrain Motion Planning," *IEEE Transactions on Robotics and Automation*, Vol. 14, 1998, pp. 19–33.

[6] Kambhampati, S. and Davis, L., "Multiresolution Path Planning for Mobile Robots Planning for Mobile Robots," *IEEE Journal of Robotics and Automation*, 1986, pp. 135–145.

[7] Behnke, S., "Local Multiresolution Path Planning," *Lecture Notes in Computer Science*, 2004, pp. 332–343.

[8] Tsiotras, P. and Bakolas, E., "A Hierarchical On-Line Path-Planning Scheme Using Wavelets," *European Control Conference*, Kos, Greece, July 2–5 2007, pp. 2806–2812.

[9] Latombe, J., *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.

[10] Lavalle, S., *Planning Algorithms*, Cambridge University Press, New York, NY, 2006.

[11] Cowlagi, R. and Tsiotras, P., "Beyond Quadtrees: Cell Decomposition for Path Planning using the Wavelet Transform," *46th IEEE Conference on Decision and Control*, Orleans, LA, Dec. 12–14 2007, pp. 1392–1397.

[12] Jung, D. and Tsiotras, P., "Multiresolution On-Line Path Planning for Small Unmanned Aerial Vehicles," *American Control Conference*, Seattle, WA, June 11-13 2008, pp. 2744–2749.

[13] Jung, D. and Tsiotras, P., "Modelling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle," *AIAA Infotech at Aerospace*, Rohnert Park, CA, May 7–10 2007, AIAA Paper 07-2768.

[14] Jung, D., *Hierarchical Path Planning and Control of a Small Fixed-Wing UAV: Theory and Experimental Validation*, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, Dec. 2007.

[15] Lozano-Perez, T. and Wesley, M., "Automatic Planning for Planning Collision-free Paths among Polyhedral Obstacles," *IEEE Trans. Syst., Man, Cybern.*, Vol. 11, 1981, pp. 681–698.

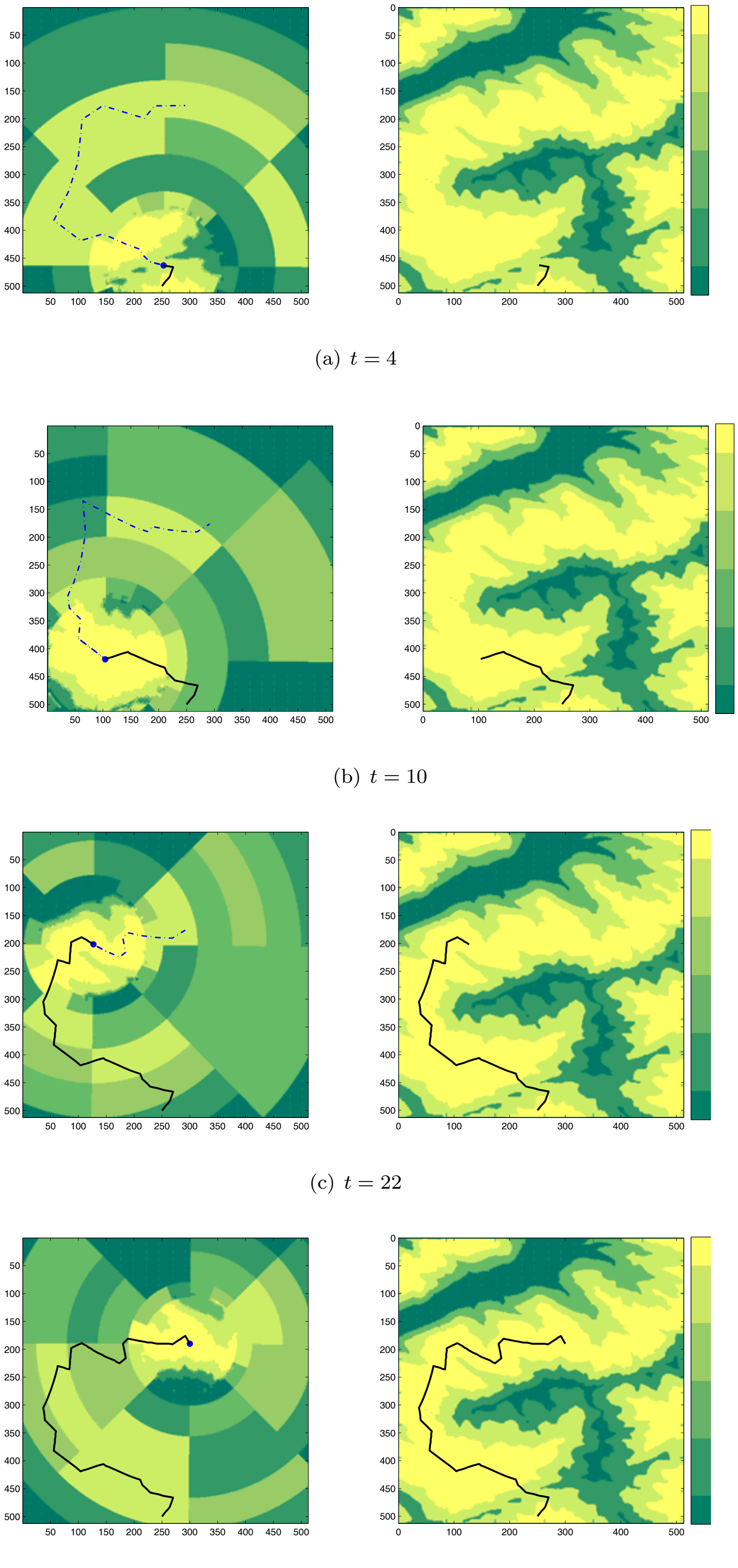
[16] Burrus, C., Gopinath, R., and Guo, H., *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, New Jersey, NJ, 1998.

[17] Mallat, S. G., "A Theory of Multi-Resolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, 1989, pp. 674–693.

[18] Cohen, A., *Numerical Analysis of Wavelet Methods*, Vol. 32, Elsever Science, Amsterdam, 2003.

[19] Walnut, D., *An Introduction to Wavelet Analysis*, Vol. 32, Birkhauser, Boston, 2003.

[20] on-line, http://edc.usgs.gov/geodata/.



(a) $t = 4$

(b) $t = 10$

(c) $t = 22$

(d) $t = t_f$

Figure 7. Path evolution with time. The figures show the actual path (solid line) along with the most recent tentative path (dotted line) of the agent at each time step. The agent reaches the final destination at $t = t_f$.