2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Multiresolution Path Planning with Wavelets: A Local Replanning Approach

Raghvendra V. Cowlagi and Panagiotis Tsiotras

Abstract— **A path planning algorithm based on multiresolution cell decomposition of the environment using wavelets is proposed. The environment is assumed to be given by an occupancy grid at fine resolution. The algorithm constructs a cell decomposition at several levels of resolution (cell sizes) and constructs an optimal path to the destination from the current location of the agent. At each step the algorithm iteratively refines a coarse approximation to the path through local replanning. The replanning process uses previous information to refine the original cell channel in the immediate area of the path. This is done efficiently using the wavelet coefficients. Numerical tests show a speed-up of an order of magnitude over the baseline algorithm with minimal impact on the overall optimality of the resulting path. A comparative study with the well-known D* algorithm is also provided.**

I. INTRODUCTION

The problem of planning a path for an autonomous mobile robot in a given workspace, while avoiding obstacles, has been studied for several years (see [1], [2], and more recently, [3]). Solution methods fall into three broad categories: cell decomposition methods, roadmap methods, and artificial potential field methods. The first two approaches transform the path planning problem into a graph search problem. In particular, cell decomposition methods partition the free space into convex, non-overlapping regions, called cells, and then employ techniques, such as the Dijkstra algorithm, to search the connectivity graph for a sequence of adjacent cells from the initial point to the goal [1, Ch. 5 and 6].

Although several sophisticated approaches for path planning have been reported in the literature, approaches based on cell decompositions are most common and are widely used in applications because of their simplicity. Often, it is advantageous to decompose the free space into as few cells as possible, in order to make the search of the corresponding graph faster. Working with multiresolution cell decompositions is beneficial when one is primarily interested in online implementation. A multiresolution scheme can keep the size of the resulting graph search tractable so that its search can be achieved using the limited on-board computational resources, while keeping the required accuracy.

Multiresolution schemes have been proposed recently, for instance, by Behnke [4] and Tsiotras and Bakolas [5]. Hwang et al [6] describe a multiresolution technique using triangles, instead of rectangles, as cells. Other implementations of multiresolution techniques include Prazenica et al [7], who present a model-predictive (receding horizon) control formulation of the path planning problem using multiresolution estimates of object locations; Kim and Lee [8] present a multiresolution potential field approach to path planning; Verwer [9] describes the use of a hierarchy of imaginary spheres encapsulating the robot for collision avoidance.

Cowlagi and Tsiotras [10] proposed a multiresolution cell decomposition algorithm based on wavelet transforms using the Haar wavelet, where the adjacency matrix is easily computed from the indices of non-zero wavelet coefficients. Tsiotras and Bakolas [5] describe the details of a path planning scheme based on such a cell decomposition. A drawback of that scheme is that it involves global replanning at each step, and that it discards prior information about the approximate global path obtained during previous iterations. This paper extends the results of [5] by proposing an algorithm that remedies this drawback by planning globally once and then progressively refining the path locally. To implement such a local replanning scheme, the proposed algorithm uses the localization property of the wavelet transform: the wavelet transform coefficients of a large image[1] can each be uniquely identified with a particular, smaller region in that image, and more importantly, the converse is also true. That is, given any region of the image, one may isolate indices of coefficients that serve as the wavelet transform coefficients of that region, and the intensity map in that region may be fully reconstructed using only those coefficients. Path planning algorithms based on local replanning reported in literature include Stentz's D* algorithm [11], the D*-Lite algorithm [12], which compute a global path once and then perform local changes if the observed environment is different from its map that was used to compute the first global path. Hwang and Ahuja [13] present a potential field based approach to local planning.

Among the several path planning algorithms reported in the literature, we consider it appropriate to elucidate a comparison of our algorithm to the well-known D* algorithm developed by Stentz [11] and [14], since they both *appear* similar as "moving-window" algorithms. We highlight a subtle but fundamental difference between the two algorithms, and we also provide a comparison of performance in terms of execution time.

The rest of the paper is organized as follows: Section

R. V. Cowlagi is a graduate student at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, Email: rcowlagi@gatech.edu

P. Tsiotras is with the Faculty of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, Email: tsiotras@gatech.edu

[1] In this paper, and in order to be consistent with the terminology used in the literature of wavelet image processing, we use the term "image" to denote any 2-D matrix of data representing the environment. This matrix can be constructed using real image data or, most often, it assigns the probability that a specific location in the environment is occupied by an obstacle.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

II briefly reviews the mathematics of wavelet transform, and its application to multiresolution cell decompositions as applied to path-planning problems. The reader is referred to [15], [16], and [17] for more detailed expositions on the mathematical theory of the wavelet transform, and to [5] and [10] for wavelet-based cell decomposition. Section III describes the proposed path planning algorithm. Section IV provides sample results, along with a comparison with the algorithm presented in [5]. Section V provides a comparison of this algorithm with the well-known D* algorithm.

II. CELL DECOMPOSITION USING WAVELETS

A. The Wavelet Transform

The discrete wavelet transform provides a framework for multiresolution analysis (MRA) of a function, that is, the construction of a hierarchy of functional approximations by projecting the function onto a sequence of nested linear spaces. Such a sequence of nested linear spaces is generated by translated and scaled versions of two functions $\phi : \mathbb{R} \to \mathbb{R}$ and $\psi : \mathbb{R} \to \mathbb{R}$ of unit energy, called the *scaling function* and *wavelet* respectively, satisfying the *orthogonality equations*

$$\langle \phi(t), \phi(t-n) \rangle = \delta(n), \tag{1a}$$
$$\langle \psi(t), \psi(t-n) \rangle = \delta(n), \tag{1b}$$
$$\langle \psi(t), \phi(t-n) \rangle = 0, \tag{1c}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, and such that there exist sequences $h(n)$ and $g(n)$ of scalars satisfying the following relations, known as the *dilation equations*

$$\phi(t) = \sum_{n=-\infty}^{\infty} h(n)\phi(2t-n), \tag{2}$$
$$\psi(t) = \sum_{n=-\infty}^{\infty} g(n)\phi(2t-n). \tag{3}$$

The scaling and wavelet functions for the two-dimensional wavelet transform can be defined by taking the tensor product of the one-dimensional scaling and wavelet functions[2]

$$\Phi_{j,k,\ell}(x,y) = \phi_{j,k}(x)\phi_{j,\ell}(y), \tag{4a}$$
$$\Psi^1_{j,k,\ell}(x,y) = \phi_{j,k}(x)\psi_{j,\ell}(y), \tag{4b}$$
$$\Psi^2_{j,k,\ell}(x,y) = \psi_{j,k}(x)\phi_{j,\ell}(y), \tag{4c}$$
$$\Psi^3_{j,k,\ell}(x,y) = \psi_{j,k}(x)\psi_{j,\ell}(y), \tag{4d}$$

where $\{\phi_{j,k}\} \stackrel{\text{def}}{=} \{\sqrt{2}^j \phi(2^j t - k) : k \in \mathbb{Z}\}$ and $\{\psi_{j,k}\} \stackrel{\text{def}}{=} \{\sqrt{2}^j \psi(2^j t - k) : k \in \mathbb{Z}\}$. Defining V_j as the linear space spanned by $\{\Phi_{j,k,\ell}(x,y) : k,\ell \in \mathbb{Z}\}$, it can be shown (see, for instance, [15, Ch. 3]) that $\{V_j\}_{j\in\mathbb{Z}}$ is a sequence of nested subspaces dense in $\mathscr{L}^2(\mathbb{R}^2)$. The discrete wavelet transform of a function $f : \mathbb{R}^2 \to \mathbb{R}$, $f \in \mathscr{L}^2(\mathbb{R}^2)$ is given by

$$c_{j_0,k,\ell} = \langle \Phi_{j,k,\ell}(x,y), f(x,y) \rangle, \quad d^i_{j,k,\ell} = \langle \Psi^i_{j,k,\ell}(x,y), f(x,y) \rangle,$$

and the reconstruction equation is

$$f(x,y) = \sum_{k,\ell=-\infty}^{\infty} c_{j_0,k,\ell}\Phi_{j,k,\ell}(x,y) + \sum_{i=1}^{3}\sum_{j=j_0}^{\infty}\sum_{k,\ell=-\infty}^{\infty} d^i_{j,k,\ell}\Psi^i_{j,k,\ell}(x,y), \tag{5}$$

where the scalars $c_{j_0,k,\ell}$ and $d^i_{j,k,\ell}$ are the *approximation* and *detail* coefficients respectively. The first term in (5) is the approximation of $f(x,y)$ at *resolution* j_0, while the second term is the difference between approximations at two successive levels of resolution.

The simplest example of scaling function and wavelet is the Haar family, defined as

$$\phi(t) \stackrel{\text{def}}{=} \begin{cases} 1 & 0 \le t < 1 \\ 0 & \text{otherwise} \end{cases}, \quad \psi(t) \stackrel{\text{def}}{=} \begin{cases} 1 & 0 \le t < 1/2 \\ -1 & 1/2 \le t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

B. Application to Cell Decompositions

An *image* of the environment is a compact, square region $\mathscr{R} \subset \mathbb{R}^2$ along with an associated intensity map $\mathscr{F} : \mathscr{R} \to \mathbb{R}$. In the context of path planning, the image could represent an elevation map of the terrain on which the robot is to move, or a risk measure that represents the probability that the corresponding location is occupied by an obstacle [5].

Let the coarse resolution level j_0 be given, and let $c_{j_0,k,\ell}$ and $d^i_{j,k,\ell}$ be the two-dimensional discrete wavelet transform coefficients of the intensity map of a given image $\mathscr{F}$. Let $\mathscr{A}_{j_0} \stackrel{\text{def}}{=} \{(j_p,k_p,\ell_p)\}$ be a set of triplets of integers such that $j_p \ge j_0$, $p = 0,1,2,\ldots$. An *approximation* of $\mathscr{F}$, say $\hat{\mathscr{F}}$, is any image obtained by the reconstruction of $a_{j_0,k,\ell}$ and $\hat{d}^i_{j,k,\ell}$, where

$$\hat{d}^i_{j,k,\ell} = \begin{cases} d^i_{j,k,\ell} & i = 1,2,3;\ (j,k,\ell) \in \mathscr{A}_{j_0} \\ 0 & \text{otherwise.} \end{cases}$$

In the rest of this paper, we denote an approximate image by its associated set of non-zero detail coefficients $\mathscr{A}_{j_0}$, in a minor abuse of notation.

A cell decomposition of the environment is achieved through an appropriate selection of $\mathscr{A}_{j_0}$, along with the use of a compactly supported scaling function and wavelet. The Haar wavelet and the Daubechies, symlet, and coiflet families of wavelets [20] are all examples of compactly supported wavelets.

Consider the 2-D Haar scaling function and wavelets in (4a)-(4d). The approximation of the environment at resolution j_0, is piecewise constant over the cell decomposition $\mathscr{P}_{j_0} = \{\mathscr{C}_{j_0,k,\ell} : \bigcup_{k,\ell\in\mathbb{Z}} \mathscr{C}_{j_0,k,\ell} = \mathscr{R}\}$. The intensity of the approximation over the cell $\mathscr{C}_{j,k,\ell}$ is equal to $2^{j_0}c_{j_0,k,\ell}$. It is possible to construct a *multiresolution approximation*, $\mathscr{P}^{\mu}_{j\ge j_0}$ by first constructing an approximation at resolution j_0, and then successively expressing Φ_{j,k_j,ℓ_j}, for some pairs (k_j,ℓ_j), as a linear combination of $\{\Phi_{j+1,k_{j+1},\ell_{j+1}}(x,y) : k_{j+1},\ell_{j+1} \in \mathbb{Z}\}$, similar to (3) for the 1-D case. A multiresolution approximation is a piecewise constant function over a cell

[2]The tensor product induces a directional bias in the transform. Isotropic, nonseparable two-dimensional transforms are also possible, but the corresponding functions are more difficult to construct [18], [19].

decomposition in which cells are of different dimensions at different locations, that is, $\mathscr{P}^{\mu}_{j\geq j_0} = \{\mathscr{C}_{j,k,\ell} : \bigcup_{j,k,\ell}\mathscr{C}_{j,k,\ell} = \mathscr{R},\ j \geq j_0,\ k \in K(j),\ \ell \in L(j)\}$, where $K(j)$ and $L(j)$ are some resolution-dependent index subsets of $\mathbb{Z}$.

Reference [10] details a procedure for computing the adjacency matrix and intensity map of such a multiresolution cell decomposition.

III. A LOCAL REPLANNING ALGORITHM

In Ref. [5] the authors describe a path planning algorithm based on multiresolution cell decomposition arising from the 2-D wavelet transform. At the i^{th} iteration, a cell decomposition $\mathscr{P}^{\mu,i}_{j\geq j_0}$ is computed such that the cells are of high resolution (i.e., small dimension) in the vicinity of the current location of the agent, while of low resolution in regions farther away. Dijsktra's algorithm is used to find the optimal sequence of cells leading to the goal. The agent steps forward and the next iteration is performed using the agent's new location as a starting point.

This algorithm performs a new cell decomposition at each iteration, computing a path over the entire environment. Thus, it discards all information about the path other than the location of the immediately next cell. This section describes a refinement of this algorithm based on local replanning ideas.

A. Description of the Algorithm

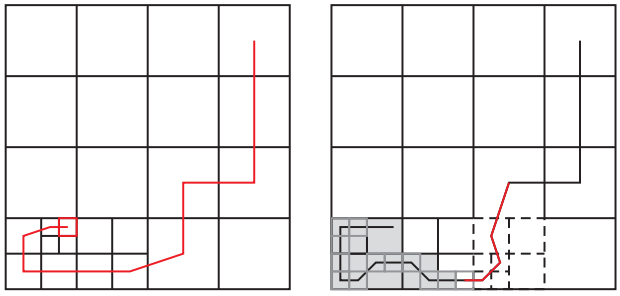
The central idea used for the implementation of the algorithm is that given a cell $\mathscr{C}_{J,K,L}$, the reconstruction equation for the restriction $\mathscr{F}|_{\mathscr{C}_{J,K,L}}$ is

$$\mathscr{F}|_{\mathscr{C}_{J,K,L}}(x,y) = c_{j_0,\hat{k},\hat{\ell}}\Phi_{j_0,\hat{k},\hat{\ell}}(x,y) + \sum_{i=1}^{3}\sum_{j=j_0}^{\infty}\sum_{k=\lfloor 2^{j-J}K\rfloor}^{\lfloor 2^{j-J}(K+1)\rfloor}\sum_{\ell=\lfloor 2^{j-J}L\rfloor}^{\lfloor 2^{j-J}(L+1)\rfloor} d^{i}_{j,k,\ell}\Psi^{i}_{j,k,\ell}(x,y), \quad (7)$$

where $\hat{k} = \lfloor 2^{j_0-J}K\rfloor$ and $\hat{\ell} = \lfloor 2^{j_0-J}L\rfloor$. Thus, for any cell $\mathscr{C}_{j,k,\ell} \subset \mathscr{R}$, the wavelet transform coefficients of $\mathscr{F}|_{\mathscr{C}_{J,K,L}}$ is a subset of those of $\mathscr{F}$. This is true for the Haar wavelet since the compact supports of wavelets do not overlap at a given level of resolution and an integral number of supports of higher resolution wavelets are contained in the support of a low resolution wavelet.

A multiresolution approximation of an image representing the environment map is created as described in Section II, such that a high resolution is maintained in the vicinity of the vehicle, while lower resolutions are used in the regions farther away. Dijkstra's algorithm is used to determine a sequence of cells leading to the goal. Note that these cells would be of different sizes, and, owing to the manner of approximation, would be larger (i.e., of low resolution) towards the goal. Figure 1 a) illustrates this step schematically, where the red cell (which is of highest resolution) denotes the initial position. Suppose the sequence of cells determined in this iteration, highlighted by the red path, is $\{\mathscr{C}_n\} \stackrel{\text{def}}{=} \{\mathscr{C}_{j_n,k_n,\ell_n}\}$, $n \in [1,N]$, $\mathbb{N} \ni N \geq 2$.

In each of the further iterations, the cell $\mathscr{C}_n$ for the smallest n satisfying $j_n < \infty$ is decomposed into, say, M_i smaller cells by including higher level wavelet coefficients in its reconstruction. The adjacency matrix for the set of cells $\{\mathscr{C}_n, \mathscr{C}_{n_q}, \mathscr{C}_{n+1}\}$, $q \in [1,M_i]$ can be obtained using the same algorithm that was used for determining the adjacency matrix for the entire cell-decomposed image [10]. For local replanning, Dijkstra's algorithm is used again to find the optimal sequence of, say, N_i cells from $\mathscr{C}_{n-1}$ to $\mathscr{C}_{n+1}$, which we denote by $\{\mathscr{C}_{n_p}\}, p \in [1,N_i]$. The original sequence of cells from the initial point to the goal is thus updated with $\{\mathscr{C}_{n_p}\}, p \in [1,N_i]$ replacing $\mathscr{C}_n$. The cells $\{\mathscr{C}_{n_p}\}, p \in [1,N_i]$ are renumbered $\{\mathscr{C}_p\}, p \in [n,n+N_i]$, and $\mathscr{C}_{n+1}$ is renumbered $\mathscr{C}_{n+N_i+1}$. Thus, at the end of the i^{th} iteration, the number of cells in the sequence from initial point to goal is $N+\sum_{k=1}^{i}N_k$. The next iteration is performed on the updated path. Figure 1 b) illustrates this step schematically. The cells with dotted edges denote $\{\mathscr{C}_{n_q}\}$, $q \in [1,M_i]$. The optimal subpath through the set of cells $\{\mathscr{C}_n, \mathscr{C}_{n_q}, \mathscr{C}_{n+1}\}$, $q \in [1,M_i]$ is shown in red.



(a) (b)

Fig. 1. Schematic illustration of local replanning algorithm

The selection of the cost function was done as follows. Let $\hat{g} : \mathscr{R} \times \mathscr{R} \to \mathbb{R}$ denote the true transition cost between two points and $g : \mathscr{P}^{\mu}_{j\geq j_0} \times \mathscr{P}^{\mu}_{j\geq j_0} \to \mathbb{R}$ denote the transition cost function that must be used in the proposed algorithm. We require the cost function to penalize paths in high terrain and in regions far away from the goal. Therefore, we chose the following function:

$$\hat{g}(x_1,x_2) = k_1(\mathscr{F}(x_1)+\mathscr{F}(x_2)) + k_2\|x_{\text{G}}-x_2\|_2, \quad (8)$$

where x_{G} denotes the goal, and k_1 and k_2 are constants. This structure lends itself to the formulation of g as follows:

$$g(\mathscr{C}_1,\mathscr{C}_2) = k_1(2^{-j_1}\hat{\mathscr{F}}(\mathscr{C}_1)+2^{-j_2}\hat{\mathscr{F}}(\mathscr{C}_2)) + 2^{-j_2}k_2 r(\mathscr{C}_{\text{G}},\mathscr{C}_2), \quad (9)$$

where $\mathscr{C}_{\text{G}}$ indicates the cell containing x_{G}, and $r(\cdot,\cdot)$ denotes the Euclidean distance between the centers of the cells. The multipliers 2^{-j_i} are necessary in order to account for the size of the cells.

B. Comments

The primary benefits of the proposed algorithm are its speed and simplicity. The algorithm is faster than the technique proposed in [5] since it concentrates on local planning at each iteration, thus drastically cutting down the number of nodes on which a Dijkstra graph search is performed. Section IV demonstrates simulation results that corroborate this claim. The proposed algorithm saves computational

time because it iteratively *refines* a coarsely known path, as opposed to the algorithm in [5], which computes a *new* path at each iteration.

The simplicity of the proposed algorithm arises from the fact that it recursively uses the algorithms for multiresolution decomposition and for computing the adjacency matrix of the resulting cell decomposition. It exploits the localization property of the Haar wavelet, as explained earlier, to do so.

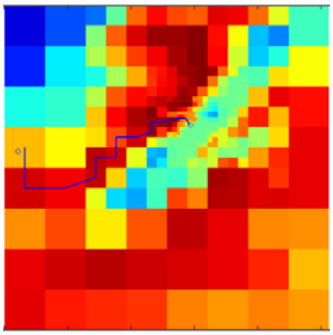
The primary drawback of this algorithm is that the cost of the resultant path dependents on the sequence of cells computed initially. By confining future local searches to the sequence of cells computed in the first iteration, it "shuts itself out" to possibly better paths that may lie outside that sequence. For example, small obstacles may be obscured due to the averaging operation of the Haar wavelet, and as a result, the initial sequence of cells may contain small but insurmountable obstacles. Section IV, however, illustrates an example where the algorithm successfully computed an obstacle-free path through a cluttered environment, indicating that this may occur only when obstacles are extremely small.

Although the described algorithm applies to a static environment, it can be easily extended to deal with an unexpected environment by performing the initial, global planning step each time the agent encounters an unexpected map of the environment. A new sequence of cells leading up to the goal can be computed, and the local replanning would continue using the new sequence of cells. However, in the extreme circumstance of encountering a different environment at *each* step, the above extension to the algorithm would make its functioning identical to the algorithm in [5].

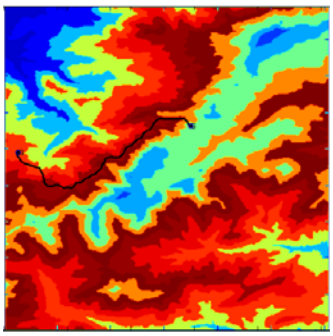
IV. SIMULATION AND RESULTS

Figures 2 and 3 show the progression of the proposed algorithm. The initial point and goal are indicated by a square and diamond respectively. The image shown in the figures corresponds to terrain height, where the red shades indicate low terrain (favorable), and blue shades indicate high terrain (unfavorable). Figure 2 a) shows the initial cell decomposition and the corresponding global path, while Fig. 2 b) shows the final path traversed. It can be seen that the path remains in the darker shades of red throughout. Figure 3 shows an intermediate stage with progressive decomposition of cells along the path. This cell decomposition should be compared with that in Fig. 2 a). The uniformly maroon areas indicate that they were ignored altogether, once they were identified as lying outside the initial sequence of cells.

Table I shows sample comparison results between the proposed algorithm and that in [5]. It is seen that the local replanning algorithm is an order of a magnitude faster but produces paths with slightly higher costs. Also note that the savings in execution time increases with the size of the image being processed. The image sizes are in pixels, and the cost difference is the extra cost of the proposed algorithm's resultant path as a percentage of the cost of path resulting from the algorithm in [5]. The suboptimality of the path with local replanning is typically at the order of 10%. Case 2 in



(a)



(b)

Fig. 2. Initial global planning and the final path traversed

TABLE I
SAMPLE COMPARISON RESULTS

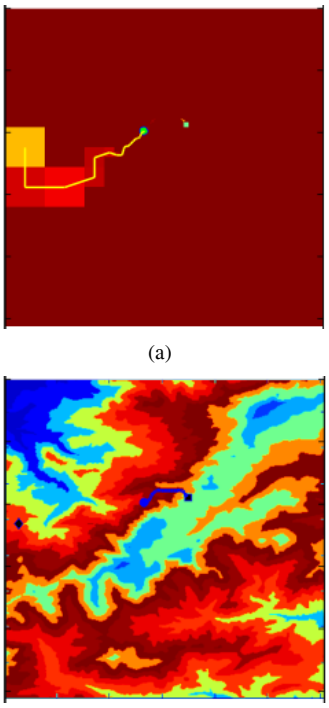
	Image size (pixels)	Execution time ratio	Cost difference (%)
1.	64×64	8.46	7.17
2.	64×64	10.7	34.5
3.	64×64	9.04	16.8
3.	128×128	18.5	16.4
4.	128×128	15.4	7.95
5.	128×128	10.6	10.5
6.	256×256	51.5	2.87
7.	256×256	51.2	5.52
8.	256×256	34.1	18.8

Table I is an anomalous result where the true optimal path lied mostly outside the initial sequence of cells.

Figure 4 shows an example of the proposed algorithm applied to a cluttered environment. Despite the seemingly "short sighted" nature of local replanning, it is evident that the algorithm succeeds in finding reasonable collision-free paths.

V. COMPARISON WITH D* ALGORITHM

The D* algorithm, developed by Stentz [11], is a computationally efficient alternative to replanning with the A* algorithm in a dynamic environment or when the environment is only known locally. As with the proposed multiresolution strategy suggested in this work, D* employs a moving window approach, where data is processed as it becomes available. However, there is a significant difference between D* and the wavelet-based multi-resolution strategy developed in this work. First, D* is not a path-planning strategy per se, but rather an efficient recursive implementation of A* to search over a given graph when new data becomes



(a)

(b)

Fig. 3. Intermediate step in path planning

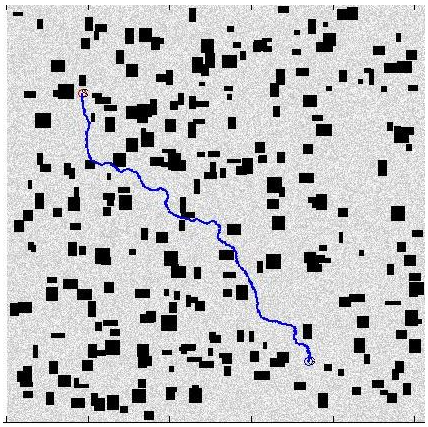


Fig. 4. Local replanning successfully applied to a cluttered environment

available. On the contrary, the proposed multi-resolution strategy *computes* a suitable graph at each iteration step. The algorithm to search over this graph to find the optimal path is a secondary objective; any of the existing algorithms will do (such as Dijkstra or A*). Nonetheless, both algorithms provide a solution to the path-planning problem in a (partially) unknown environment and in that sense a suitable comparison between the two algorithms is warranted.

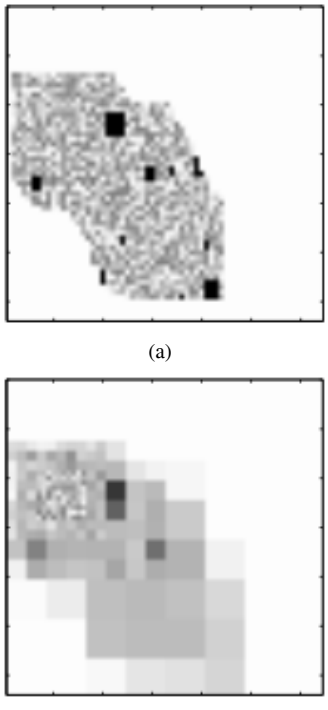
Because of the difference in their scope, a head-to-head comparison between the two algorithms is not easy. To this end, we have chosen a setting where the environment is not changing, but the data is made available to the agent only incrementally, within a given window around the vehicle's current location. The rest of the environment is assumed to be unobstructed. A sample of such a map available to the vehicle is shown in Fig. 5. Note that the wavelet-based algorithm is also given the same information, that is, at each iteration it will compute the wavelet transform of the data shown in Fig. 5, and then use the results of the transform to compute a multiresolution cell decomposition.

Table II shows sample results comparing the execution times between the two algorithms. The information window w_I indicates the area in which information is made available to both algorithms. The resolution window was kept fixed, since it is a characteristic of the wavelet algorithm, not the path planning problem itself.

TABLE II
SAMPLE RESULTS

	w_I (pixels)	Execution time ratio
1.	4	0.564
2.	4	0.538
3.	10	1.22
4.	10	1.38
5.	25	2.84
6.	25	3.21

The wavelet algorithm is slower when the information window is small. In this situation, the available information about the environment is too little for the wavelet algorithm's approximation scheme to provide any advantage in terms of reduced number of computations. The algorithm is slower since it replans at each step by calling A*, while the D* algorithm does not (see [11] for details). However, the wavelet algorithm executes significantly faster than D* as w_I gets larger. The reason for this is that D* needs to correct more backpointers in this case. On the other hand,



(a)

(b)

Fig. 5. Sample of image that the algorithms process at each step. The wavelet approximation of the image is also shown.

the number of nodes processed at each step in the wavelet algorithm does not depend on the information window (it depends on the resolution window.)

TABLE III

CASE WITH EQUAL INFORMATION AND RESOLUTION WINDOWS.

	w_{I} (pixels)	Execution time ratio
1.	10	0.708
2.	10	0.964
3.	10	0.711
4.	10	1.037
5.	25	0.882
6.	25	0.693
7.	25	0.818
8.	25	0.916

To confirm these observations we also compared the results from the two algorithms when the information and fine resolution windows are the same for both case. In this case, the main difference is how the new information that becomes available by the two algorithms as it enters the fine resolution horizon is handled. D* needs to update only the edges that approximately correspond to the boundary of the window. It always works with the same graph, whose dimensionality remains the same at each iteration. The size of the graph is set during the initialization stage once and for all. The multiresolution approach constructs a new graph at each time step, whose dimensionality is roughly determined by the size of the fine resolution window. The results are shown in Table III for different fine resolution window lengths. It is seen that in terms of the execution time, the two algorithms are comparable. Nonetheless, in terms of memory requirements the D* has a major disadvantage. This may hinder its implementation using small size embedded microcontrollers; see also [21]. Furthermore, as indicated by the results in the previous section, a local replanning implementation can speed up the multiresolution of the wavelet-based algorithm about an order of magnitude.

VI. CONCLUSIONS

In this paper we have proposed a computationally efficient path planning algorithm that uses multiresolution cell decomposition of the environment based on the Haar wavelet transform. The algorithm's efficiency stems from its recursive use of the fast wavelet transform and from the ease in computing the adjacency matrix directly from the wavelet coefficients. Simulations demonstrate the speed and effectiveness of the proposed algorithm. A comparative study with the well-known D* algorithm demonstrates that the proposed algorithm provides efficient processing of the data about the environment the vehicle operates in.

Acknowledgment: This work has been supported in part by NSF award CMS-0510259 and ARO award W911NF-05-1-0331.

REFERENCES

[1] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[2] Y. K. Hwang and N. Ahuja, "Gross motion planning - a survey," *ACM Computing Surveys*, vol. 24, no. 3, pp. 219–291, September 1992.

[3] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

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

[5] P. Tsiotras and E. Bakolas, "A hierarchical on-line path planning scheme using wavelets," in *European Control Conference*, Kos, Greece, 2007.

[6] J. Y. Hwang, J. S. Kim, S. S. Lim, and K. H. Park, "A fast path planning by path graph optimization," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 33, no. 1, pp. 121–127, January 2003.

[7] 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," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[8] C.-T. Kim and J.-J. Lee, "Mobile robot navigation using multi-resolution electrostatic potential field," in *32nd Annual Conference of IEEE Industrial Electronics Society, 2005, IECON 2005*, 2005.

[9] B. J. H. Verwer, "A multiresolution workspace, multiresolution configuration space approach to solve the path planning problem," in *Proc. 1990 IEEE Intl. Conf. Robotics & Automation*, 1990, pp. 2107–12.

[10] R. V. Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decomposition for path planning using the wavelet transform," in *46th IEEE Conf. Decision and Control, New Orleans*, 2007, pp. 1392–1397.

[11] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Intl. Conf. Robotics and Automation (ICRA)*, 1994, pp. 3310 – 3317.

[12] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robotics & Automation*, vol. 21, no. 3, pp. 354 – 363, June 2005.

[13] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robotics & Automation*, vol. 8, no. 1, pp. 23 – 32, Feb. 1992.

[14] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," Carnegie Mellon Robotics Institute, Tech. Rep. CMU-RI-TR-93-20, 1993.

[15] R. M. Rao and A. S. Bopardikar, *Wavelet Transforms - Introduction to Theory and Applications*. Addison-Wesley, 1998.

[16] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–93, 1989.

[17] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms - A Primer*. Prentice Hall, 1998.

[18] A. Cohen and I. Daubechies, "Nonseparable bidimensional wavelet bases," *Rev. Mat. Iberoamericana*, vol. 9, pp. 51–137, 1993.

[19] E. Belogay and Y. Wang, "Arbitrarily smooth orthogonal nonseparable wavelets in $\mathbb{R}^2$," *SIAM Journal on Mathematical Analysis*, vol. 30, no. 3, pp. 678–697, 1999.

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

[21] D. Jung and P. Tsiotras, "Multiresolution on-line path planning for small unmanned aerial vehicles," in *American Control Conference*, Seatlle, WA, 2008.