UNIVERSITAT POLITÈCNICA
DE CATALUNYA

A novel path planning proposal based on the combination of deterministic sampling and harmonic functions

Jan Rosell, Carlos Vázquez, Alexander Pérez, Pedro Íñiguez

IOC-DT-P-2007-06
Març 2007

Institut d'Organització i Control
de Sistemes Industrials

A novel path planning proposal based on the combination of deterministic sampling and harmonic functions

Jan Rosell, Carlos Vázquez, Alexander Pérez and Pedro Iñiguez

Abstract

The sampling-based approach is currently the most successful and yet more promising approach to path planning problems. Sampling-based methods are demonstrated to be probabilistic complete, being their performance reliant on the generation of samples. To obtain a good set of samples, this paper proposes a new sampling paradigm based on a deterministic sampling sequence guided by an harmonic potential function computed on a hierarchical cell decomposition of $\mathcal{C}$-space. In the proposed method, known as *Kautham* sampler, samples are not isolated configurations but parts of a whole. As samples are generated they are dynamically grouped into cells that capture the $\mathcal{C}$-space structure. This allows the use of harmonic functions to share information and guide further sampling towards more promising regions of $\mathcal{C}$-space. Finally, using the samples obtained, a roadmap is easily built taking advantage of the known neighborhood relationships.

Index Terms

Path planning, sampling-based methods, deterministic sampling, harmonic functions.

I. Introduction

The sampling-based approach to path planning consists in the generation of collision-free samples of configuration space ($\mathcal{C}$-space) and in their interconnection with free paths, forming either roadmaps (PRM [1]) or trees (RRT [2]). PRM planners are conceived as multi-query planners, while RRT planners are developed to rapidly solve a single-query problem. Some middle ground methods are also proposed, like the predictive roadmap [3] based on a statistical model of the $\mathcal{C}$-space which is incrementally refined using the information provided by each single query.

The sampling-based approach is giving very good results in robot path-planning problems with many degrees of freedom. Its success is mainly due to its sampling-based nature, i.e. this approach does not require the explicit

Manuscript submitted to IEEE-TR December 29, 2006.

This work was partially supported by the CICYT projects DPI2004-03104 and DPI2005-00112

Jan Rosell, Carlos Vázquez and Alexander Pérez are with the Institute of Industrial and Control Engineering at the Technical University of Catalonia, Barcelona, Spain (email: jan.rosell@upc.edu)

Pedro Iñiguez is with the Dept. of Electronics, Electrics and Automatic Engineering at the Rovira i Virgili University, Tarragona, Spain

characterization of the obstacles of $\mathcal{C}$-space and its efficiency relies on the sample set. Therefore, the generation of samples is one of the crucial factors in the performance of sampling-based planers. Taking into account this fact, this paper proposes a sampling-based path planner that uses a new sampling paradigm.

The paper is structured as follows. Section II analyzes some performance considerations about sampling-based planners and outlines the new sampling paradigm. Section III makes an overview of the approach, that is based on: a) the generation of samples with a deterministic sampling sequence (Section IV); b) the grouping of samples into cells of a hierarchical cell decomposition of $\mathcal{C}$-space (Section V); and c) the computation of harmonic potential functions to guide the sampling process (Section VI). Section VII summarizes and illustrates the proposal and Section VIII discusses the contributions.

II. Problem analysis

A. Performance considerations

The performance of sampling-based methods depends on the number of samples required, being the computational cost related to their generation and interconnection.

a) Sample generation: Sampling-based methods based on probabilistic sampling are demonstrated to be probabilistic complete, e.g. for the basic PRM method the number of samples necessary to achieve a probability of failure below a given threshold has been determined [4]. For difficult path-planning problems, like those involving narrow passages, this number might be quite large and, therefore, importance sampling methods have been introduced (e.g. [5]–[8]). Those strategies increase the density of sampling in some areas of $\mathcal{C}$-space, thus facilitating the finding of a solution using a reasonable amount of samples. Nevertheless, the computational cost of selecting samples in the critical regions of $\mathcal{C}$-space is usually high. Also, the collision-check test is one of the costly operations of the sample generation process. To cope with that, lazy-evaluation approaches have been introduced (for single-query problems) to delay collision-checks until it is absolutely necessary [9].

b) Sample interconnection: The cost of the construction of roadmaps or trees is due to both the computation of neighborhood relationships between samples, and the need to use a local planner to connect neighbor samples with a free path. The cost of computing neighborhood relationships can be reduced using deterministic sampling methods [10] that, besides providing good incremental and uniform coverage of $\mathcal{C}$-space, they have a lattice structure that is useful for neighborhood computations. The cost of using the local planner can be avoided as much as possible, for single-query problems, if lazy-evaluation approaches are used. A comparative study of local planning techniques and of sampling methods can be found in [11] for PRM planners.

c) Sampling profit: Sampling-based methods usually discard collision configurations and consider samples as collision-free isolated configurations, being the information about the $\mathcal{C}$-space only captured by the interconnections of the samples through the roadmap or tree. The information contributed by the collision configurations is not fully profited. Some exception is the model-based approach [12] that uses a first sample set of both free and collision configurations to build a statistical model of the $\mathcal{C}$-space that is then used to bias the sampling for the roadmap

construction, or those based on probabilistic cell decompositions that combine probabilistic sampling with cell decomposition techniques [13], [14].

B. Objective

Taking into account the previous considerations, this paper has as a main objective the proposal of a new sampling paradigm that provides a set of key samples that allow to construct a local roadmap able to solve a given single-query problem. It is desired that the sampling process:

- generate samples that produce an incremental and uniform coverage of the $\mathcal{C}$-space
- have a lazy phylosophy in order not to collision-check all the samples generated
- generate a structured set of samples that allow an easy computation of neighborhood relationships
- group the samples generated in order to capture the structure of the $\mathcal{C}$-space

C. A new sampling paradigm

A sampling method, called *Kautham* sampling or k-sampling for short, is proposed to achieve the objectives stated. The k-sampler follows a new sampling paradigm where the samples are not isolated configurations but parts of a whole. The sampling process dynamically groups samples into cells that capture the $\mathcal{C}$-space structure. This allows the use of harmonic functions to share information and guide further sampling towards more promising regions of $\mathcal{C}$-space.

The k-sampler is structured into three constituent parts with the following features:

a) Deterministic sampling sequence: The use of a deterministic sampling sequence allows an uniform and incremental coverage of $\mathcal{C}$-space, i.e. it has a spatial an temporal **continuity** feature. Moreover the set of samples results with a lattice structure that facilitates the computation of neighbors.

b) Hierarchical cell decomposition: The use of a hierarchical cell decomposition of $\mathcal{C}$-space allows the grouping of samples into non-uniform cells, capturing the structure of the $\mathcal{C}$-space. Cells are not classified as free or collision cells (i.e. white and black cells) as usually done in cell decomposition methods, not even in a fuzzy manner as done in [14]. Instead, cells are all considered equal and characterized by a transparency parameter computed as a function of the number of free and collision samples they contain, i.e. the cell decomposition has a **non-duality** or **unity** feature. The transparency parameter is used as a control parameter for both controlling the necessity of performing collision-checks (i.e. as a lazy-evaluation control), and controlling the partitioning procedure of the cell decomposition.

c) Harmonic functions: The use of two harmonic functions (H_1 and H_2), computed at each iteration of the sampling process, allows to globally capture the current knowledge of $\mathcal{C}$-space. H_1 is used to find a solution channel from the initial cell to the goal cell on the current cell decomposition of $\mathcal{C}$-space. H_2 is used to propagate the information of the channel in order to bias the sampling towards the regions around it. The harmonic functions are not only computed over the free cells (fixing the obstacle cells at a high value), as it is usually done, since as commented above this cell classification is not considered here. Instead, the harmonic functions are computed over

the whole set of cells (using the transparency as a weighting parameter), i.e. the harmonic function computation has an **interconnection** feature between the whole set of cells of the $\mathcal{C}$-space.

The k-sampling process identifies the regions where the solution of a single-query problem may probably lie. The free samples pertaining to those regions are called k-samples and are the main output of the k-sampler. The k-samples constitute the key samples needed to construct a local roadmap able to solve a given single-query problem.

III. Approach Overview

The k-sampling is the iterative process that generates the set of k-samples whose interconnection must permit to find a solution path between an initial and a goal configurations of $\mathcal{C}$-space. Let c_{ini} and c_{goal} be, respectively, those configurations. Then, at each iteration the k-sampler:

1) Obtains a set of samples from the deterministic sampling sequence (Section IV).
2) For each sample s_i (Section V):
    a) Classifies it into the corresponding cell, C_i, of the $\mathcal{C}$-space partition (Section V-A).
    b) Computes the transparency of C_i (Section V-B).
    c) Performs or not a collision check a the configuration associated to s_i depending on the transparency of C_i and on its H_2 value (Section V-C).
    d) Partions or not cell C_i depending on the transparency of C_i and on its H_2 value (Section V-D).
3) Computes the harmonic functions (Section VI):
    a) Computes an harmonic function, H_1, with goal cell the cell containing c_{goal}.
    b) Searches a channel of cells connecting the cell containing c_{ini} with the cell containing c_{goal}, following the negated gradient of H_1.
    c) Resamples and partitions channel cells whenever necessary.
    d) Computes an harmonic function, H_2, using as goal cells the cells of the solution channel computed with H_1.
4) Returns the free samples of the channel cells.

The returned samples are the k-samples. They are connected as a roadmap, as well as c_{ini} and c_{goal}. Finally, a solution path between them is searched in the roadmap.

IV. Deterministic sampling sequence

The deterministic sampling sequence proposed is based on: a) a hierarchical decomposition of a unit cube of parameters (Section IV-A); b) a low-dispersion ordering of the descendant cells of any given cell of the hierarchical decomposition (Section IV-B); c) a recursive application of that ordering to sample the parameter space (Section IV-C); and d) the mapping from samples of parameter space to configurations of $\mathcal{C}$-space (Section IV-D).

A. Hierarchical cell decomposition

A 2^d-tree decomposition of a d-dimensional unit cube of parameters is considered in a similar way as done in [14]. The initial cell with sides with unitary size is the tree root. The levels in the tree are called partition levels. A cell of a given partition level m is called an m-cell. Partition levels are enumerated such that the tree root is the partition level 0 and the maximum resolution[1] corresponds to partition level M, also called sampling level. A maximum allowable partition level P is defined, with $P \leq M$, which determines the depth of the 2^d-tree. The M-cells are also called samples, and up to $2^{(M-P)d}$ are found in each P-cell. Cell coding is done as follows. Consider first the coding of M-cells. Let:

- The index matrix V^M be the binary $d \times M$ matrix whose rows are the binary representation of the indices $v_j^M \ \ \forall j \in 1 \dots d$ of an M-cell on the regular grid of partition level M:

$$V^M = \begin{pmatrix} v_1^M \\ \vdots \\ v_j^M \\ \vdots \\ v_d^M \end{pmatrix} = \begin{pmatrix} a_{M1} & \dots & a_{i1} & \dots & a_{11} \\ \vdots & & \vdots & & \vdots \\ a_{Mj} & \dots & a_{ij} & \dots & a_{1j} \\ \vdots & & \vdots & & \vdots \\ a_{Md} & \dots & a_{id} & \dots & a_{1d} \end{pmatrix} \tag{1}$$

  being a_{Mj} and a_{1j} the most and the least significant bits, respectively, of the binary representation of v_j^M.

- The weight matrix W^M be a $d \times M$ matrix

$$W^M = \begin{pmatrix} w_{11} & \dots & w_{1j} & \dots & w_{1M} \\ \vdots & & \vdots & & \vdots \\ w_{i1} & \dots & w_{ij} & \dots & w_{iM} \\ \vdots & & \vdots & & \vdots \\ w_{d1} & \dots & w_{dj} & \dots & w_{dM} \end{pmatrix} \tag{2}$$

  with $w_{ij} = 2^{(M-j)d+i-1}$ for $i \in 1 \dots d \ \ j \in 1 \dots M$.

Then, the sample code C^M and its index matrix V^M are related as follows:

$$C^M = V^M \cdot W^M \tag{3}$$

$$V^M = C^M \& W^M \tag{4}$$

where the operation $A \cdot B$ represents the scalar product of matrices A and B, and the operation $a \& B$ between a scalar a and a matrix B computes the bit-AND operation between a and all the components b_{ij} of B.

[1]The maximum resolution needed is a fixed value determined by the clearance of the path planning problem to be solved.

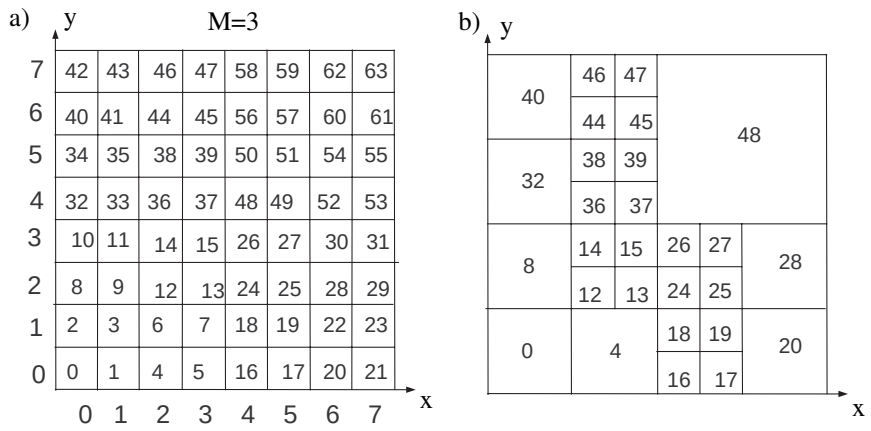

Fig. 1. a) Coding of M-cells; b) Coding of a hierarchical cell decomposition.

As an example, the conversion operations of cell code 22 with indices (6,1) on the grid of partition level $M = 3$ (Figure 1a) are:

$$C^3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 16 & 4 & 1 \\ 32 & 8 & 2 \end{pmatrix} = 22 \tag{5}$$

$$V^3 = 22\& \begin{pmatrix} 16 & 4 & 1 \\ 32 & 8 & 2 \end{pmatrix} =$$

$$= 010110\& \begin{pmatrix} 010000 & 000100 & 000001 \\ 100000 & 001000 & 000010 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

The cell code of any m-cell, with $m < M$, is made coincident with the code of the first M-cell it contains (i.e. the descendant M-cell with lowest cell code), as illustrated in Figure 1b. This cell coding facilitates the classification of any given sample to the corresponding cell in the hierarchical decomposition (Section V-A).

B. Ordering of descendant cells

Dispersion is a metrics-based measure of the uniformity of a sample set [10]. If $X = [0,1]^d \subset \Re^d$ is the space where samples are to be generated, ρ is any metrics on X and P is a set of samples taken from X, then the dispersion is defined as:

$$\delta(P,\rho) = \sup_{q \in X} \min_{p \in P} \rho(q,p) \tag{7}$$

For a given P, a good dispersion is obtained if the mutual distance of the samples is maximized [15], being the mutual distance defined as:

$$\rho_m(P) = \min_{x,y \in P} \rho(x,y) \tag{8}$$

Therefore, for a sequence of samples it is desired that, as samples are generated, the decrease of the mutual distance be as slow as possible.

This section proposes an ordering, L_d, of the 2^d descendant cells of any parent cell in a d-dimensional space that has a good performance in terms of the decrease ratio of the mutual distance. The ordering L_d is obtained using a digital construction method [15]: the ordering is found by multiplying a $d \times d$ binary matrix, T_d, by the binary representation of the indices of the sequence:

$$L_d(i) = T_d \;\; i = T_d \begin{pmatrix} n_1 \\ \vdots \\ n_d \end{pmatrix} \tag{9}$$

The performance of L_d (in terms of mutual distance) depends on the choice of T_d. A comparative study of different expressions of T_d is found in [16], being the best alternative based on a prime decomposition. This approach first defines T_d for each prime dimension:

$$T_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad T_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{10}$$

$$T_d = \text{Trunc}_d(T_{(d+1)}) \;\; \forall d \text{ prime s.t. } d \geq 5 \tag{11}$$

Then, for any non-prime d, a recursive construction is done based on the prime decomposition of d, e.g.:

$$T_6 = \begin{pmatrix} T_3 & 0 \\ T_3 & T_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{12}$$

$$T_9 = \begin{pmatrix} T_3 & T_3 & 0 \\ 0 & T_3 & 0 \\ T_3 & 0 & T_3 \end{pmatrix} \tag{13}$$

C. The sampling sequence

The sampling sequence, $s_d(k)$, is a sequence of sample codes that specifies the ordering in which the d-dimensional parameter space (also called sampling space) is explored. The sequence $s_d(k)$ is based on the recursive use of L_d.

Let $k \geq 0$ be the index of the sequence and T_d be the matrix that determines the cell ordering of the descendant cells as introduced in the previous section. Then:

$$s_d(k) = (T_d V_k^M) \cdot W'^M \tag{14}$$

where V_k^M is the index matrix corresponding to k, the product $T_d V_k^M$ is the standard binary matrix multiplication between matrices T_d and V_k^M, and W'^M is a $d \times M$ matrix of weights, with:

$$w'_{ij} = 2^{(j-1)d+i-1} \text{ for } i \in 1 \ldots d \;\; j \in 1 \ldots M \tag{15}$$

k	0	1	2	3	4	5	6	7	8	9
$s_2[k]$	0	48	32	16	12	60	44	28	8	56
k	10	11	12	13	14	15	16	17	18	19
$s_2[k]$	40	24	4	52	36	20	3	51	35	19
k	0	1	2	3	4	5	6	7	8	9
$r_2^{48}[k]$	48	60	56	52	51	63	59	55	50	62

TABLE I

FIRST 20 SAMPLES OF SEQUENCE s_2 AND FIRST 10 SAMPLES OF r_2^{48}.

(Note that matrix W'^M coincides with W^M if the order of its columns is reversed).

As an example, with $M = 3$ and the expression of T_2 proposed in Eq. (10), the sample corresponding to $k = 6$ is:

$$\begin{aligned} s_2(6) &= \left[\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} 1 & 4 & 16 \\ 2 & 8 & 32 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 & 16 \\ 2 & 8 & 32 \end{pmatrix} = 44 \end{aligned} \tag{16}$$

If only the samples of a given cell are necessary, they can be obtained with the following (re)sampling sequence. Let m_K be the partition level of that cell and K be its code. Then:

$$r_d^K(j) = K + (T_d V_j^{(M-m_K)}) \cdot W'^{(M-m_K)} \text{ with } j \geq 0 \tag{17}$$

As an example, the sample generated by $r_2(k)$ over the 1-cell 48 (i.e. the top right corner of Figure1b) for $k = 6$ is:

$$\begin{aligned} r_2^{48}(6) &= 48 + \left[\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} 1 & 4 \\ 2 & 8 \end{pmatrix} \\ &= 48 + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 \\ 2 & 8 \end{pmatrix} = 48 + 11 = 59 \end{aligned} \tag{18}$$

As a final example, the first 20 samples generated by $s_2(k)$ and the first 10 generated by $r_2^{48}(k)$ are shown in Table I. Following these sequences over Figure 1a gives a good understanding of how they incrementally and uniformly cover the sampling space.

D. Mapping to configurations of $\mathcal{C}$-space

This Section describes how the M-cells of parameter space generated by the sampling sequence are mapped to configurations of $\mathcal{C}$-space.

For robot manipulators of d d.o.f. the $\mathcal{C}$-space is the unit cube $[0, 1]^d \subset \Re^d$ (if the proper scaling is performed) and the parameter space is d-dimensional. Consider first the correspondence that associates to a given m-cell C_K^m

of parameter space, with indices $(v_1^m, \ldots, v_d^m)$, all the configurations of a set, called m-mapping set $\mathcal{S}_K^m$, whose coordinates are:

$$x_j \in [v_j^m s_m, (v_j^m + 1)s_m) \quad \forall j \in 1 \ldots d \tag{19}$$

being s_m the size of the sides of the m-cell.

Then, to each M-cell generated by the sampling sequence a single configuration of $\mathcal{C}$-space is associated as follows. First the P-cell to which the sampled M-cell pertains is easily determined using the cell coding detailed in Section IV-A. Let $(w_1^P, \ldots, w_d^P)$ be its indices. Then, the configuration associated to the M-cell is randomly chosen within the corresponding P-mapping set:

$$x_j = \text{rand}\{[w_j^P s_P, (w_j^P + 1)s_P)\} \quad \forall j \in 1 \ldots d \tag{20}$$

For 3D rigid-bodies that can both translate and rotate (free flying robots) the $\mathcal{C}$-space is $SE(3)$ but the parameter space used in the present approach is 3-dimensional, i.e. it is only used to generate positions since orientations will be randomly chosen within all their ranges. Let positions be scaled to the unit cube $[0, 1]^3 \subset \Re^3$, and orientations be represented by a rotation direction (r_x, r_y, r_z) and a rotation angle θ.

Then, to each M-cell generated by the sampling sequence the position of the associated configuration is set in a similar way as before, i.e.:

$$x_j = \text{rand}\{[w_j^P s_P, (w_j^P + 1)s_P)\} \quad \forall j \in 1 \ldots 3 \tag{21}$$

The orientation of the configuration is set as follows:

$$x_j = \text{rand}\{[0, 1)\} \quad \forall j \in 4 \ldots 6 \tag{22}$$

and using cylindrical coordinates:

$$\alpha = 2\pi x_4 \tag{23}$$

$$r_z = 1 - 2x_5 \tag{24}$$

$$r_y = \sin\alpha\sqrt{1 - r_z^2} \tag{25}$$

$$r_x = \cos\alpha\sqrt{1 - r_z^2} \tag{26}$$

$$\theta = 2\arccos(x_6) \tag{27}$$

V. Sampling space partition

When a given sample s_i is generated by the deterministic sampling sequence, it is first classified into one of the cells of the cell partition. Let C_j be such cell. Then, both the necessity of performing a collision-check at the configuration c_i associated to s_i, and the necessity of partitioning cell C_j depend on the transparency and on the value of the harmonic function H_2. Section V-A discusses sample classification issues, Section V-B formally defines the transparency parameter and Section V-C and V-D introduce, respectively, the collision-check condition and the partition condition.

A. Sample classification

Let L_c be an ordered list of cell codes, such that $L_c[j] < L_c[j+1]$. Then, a given sample with code s_i is classified into a cell with code $L_c[j]$ if:

$$L_c[j] \leq s_i < L_c[j+1] \tag{28}$$

This condition can be evaluated using a simple and quick 1-dimensional range searching algorithm over the list of cell codes.

B. Transparency

Let $color_i$ associated to a given sample s_i be the parameter that stores the information related to the free or obstacle nature of its corresponding configuration c_i. If a collision-check is performed at c_i then:

$$color_i = \begin{cases} +2 & \text{if } c_i \text{ is a free configuration} \\ -2 & \text{if } c_i \text{ is an obstacle configuration} \end{cases} \tag{29}$$

If no collision-check is performed at c_i the parameter $color_i$ is set as follows depending on the free or collision nature of the cell where s_i is classified:

$$color_i = \begin{cases} +1 & \text{if } s_i \text{ belongs to a cell with more} \\ & \text{free than obstacle samples} \\ -1 & \text{otherwise} \end{cases} \tag{30}$$

Let K_j be the number of samples pertaining to a given cell C_j. Then, its transparency T_j is defined as:

$$T_j = \frac{\sum_{i=1}^{i=K_j} color_i}{2K_j} \tag{31}$$

The transparency satisfies $-1 \leq T_j \leq 1$. It is close to zero if there are roughly the same number of free and obstacle samples, and close to one of the extremes if they are mainly either free or obstacle samples. Note that samples not collision-checked make the absolute value of the transparency to decrease since there is a 2 factor in the denominator of Eq. (31) and they have $|color_i| = 1$.

C. Collision-check condition

Following a lazy evaluation philosophy, not all the generated samples have their associated configurations collision-checked, i.e. when the cell where a sample is classified contains basically samples of the same color (i.e. either free or obstacle samples), then there is no point in performing an extra collision-check. The collision-check condition is set with the following guidelines:

- The transparency parameter captures the homogeneity of a cell, i.e. when the transparency is within a given interval I around zero the cell is not homogeneous and the collision-check test must be performed.
- The limits of I do not have to be uniform over the whole space, i.e. in regions far away from the potential solution it is not desired to perform many collision-checks and therefore I is set small since the smaller its size the lesser collision-checks are performed.

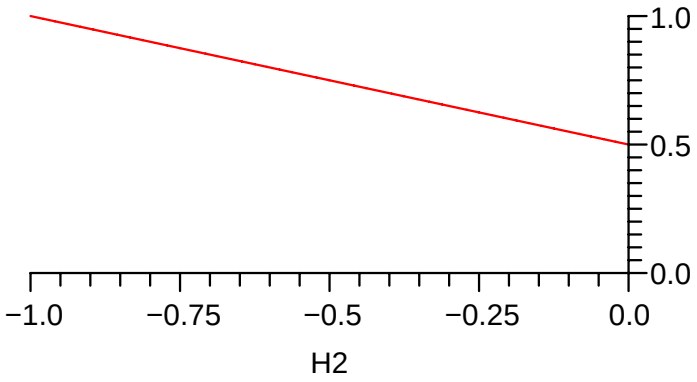

Fig. 2. Weight β_{H_2} as a function of H_2 for $\beta = 0.5$.

- The limits of I do not have to be symmetric, e.g. if the negative part of I is set smaller then the procedure is more demanding for the obstacle cells in terms of homogeneity.

Then, the proposed collision-check condition is the following:

$$\beta_{H_2}\Delta^{-}_{collision} < T_j < \beta_{H_2}\Delta^{+}_{collision} \tag{32}$$

being $\Delta^{-}_{collision}$ and $\Delta^{+}_{collision}$ fixed thresholds with values ranging between -1 and 0 and between 0 and 1, respectively; and β_{H_2} a weight ranging between 0 and 1 and dependant on the harmonic function[2] H_2 (Figure 2):

$$\beta_{H_2} = (\beta - 1)H_2 + \beta \quad 0 \leq \beta \leq 1 \tag{33}$$

Cells located far away from promising regions (i.e. where the solution channel seems not to be) have an H_2 value near zero and consequently β_{H_2} is low. Therefore, I is smaller and condition (32) has more difficulties to be satisfied resulting in less collision checks.

If condition (32) is satisfied and the cell already contained samples not collision-checked, then the collision-check is iteratively performed to the configurations of the previous not-checked samples until the condition does not hold any more, or until the last one is checked.

D. Partition condition

After performing the collision-check test, a partition condition is verified at the m-cell that contains the generated sample (if it is not of the maximum partition level, i.e. $m < M$). The cell may need to be partitioned if it is not homogeneous enough. This is evaluated following the same guidelines as for the collision-check condition, plus the following one:

[2]The harmonic function values range between -1 and 0 as detailed in Section VI-A

- The limits of I may vary as a function of weather the cell contains or not evaluated samples of different color, i.e. when the cell do have samples of different color then I is set bigger and the cell becomes more susceptible to be partitioned.

With those guidelines, the proposed partition condition is the following:

$$\beta_{H_2}\Delta^{-}_{partition} < T_j < \beta_{H_2}\Delta^{+}_{partition} \tag{34}$$

being the weight β_{H_2} defined in Eq. (33) and $\Delta^{-}_{partition}$ and $\Delta^{+}_{partition}$ two thresholds ranging between -1 and 0 and between 0 and 1, respectively, each one taking two possible values: a lower (absolute) value when the cell does not contain evaluated samples of different color, and a higher (absolute) value otherwise.

If condition (34) does not hold, then the cell is not partitioned. Otherwise, the cell is partitioned into its 2^d descendant cells, and the transparency recomputed for each descendant cell.

VI. Harmonic Functions

An harmonic function ϕ on a domain $\Omega \subset \Re^n$ is a function that satisfies Laplace's equation:

$$\nabla^2\phi = \sum_{i=1}^{n} \frac{\partial^2\phi}{\partial x_i^2} = 0 \tag{35}$$

Harmonic functions are useful for motion planers based on potential-field methods since they do not have local minima [17]. The solution of the Laplace's equation is usually found numerically using finite difference methods, i.e. by sampling ϕ and its derivatives on a regular grid and using relaxation methods that iteratively update the value of a cell by the mean of its neighbor cells.

Solutions over non-regular grids are also possible [14]. In this case, the value of the harmonic function at each cell (called the HF-value) is computed as a weighted mean of the HF-values of its neighbors, being the weights dependant on the size of the border between cells.

The computation of the harmonic function is introduced in Section VI-A, and Section VI-B discusses its use to search the solution channel and further explore the $\mathcal{C}$-space.

A. Harmonic function values

An harmonic function is computed over a hierarchical cell decomposition using a relaxation method that iteratively computes the harmonic function value of each cell. Let:

- N_j be the number of neighbors of an m-cell C_j^m in the hierarchical cell partition.
- U_H and U_L be, respectively, the high and low value of the harmonic function. They are fixed to $U_H = 0$ and $U_L = -1$.
- h_i be the harmonic function value of cell C_i^n.
- $\omega_{i,j}$ be the size of the border between cell C_i^n and cell C_j^m measured in M-cells:

$$\omega_{i,j} = 2^{(d-1)(M-\max(m,n))} \tag{36}$$

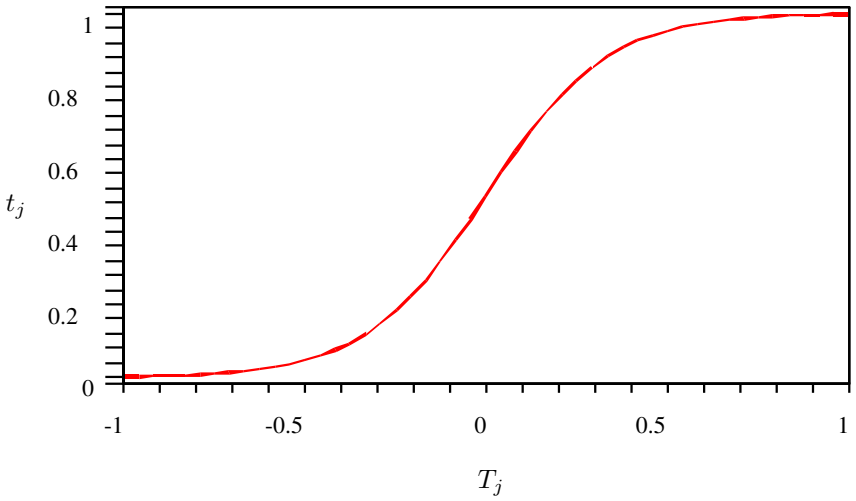

Fig. 3. Parameter t_j as a function of T_j. This parameter weights the influence of the neighborhood in the computation of the harmonic function values of a cell.

- U_j be the neighbor average computed as:

$$U_j = \frac{\sum_{i=1}^{N_j}(T_i+1)\omega_{i,j}h_i}{\sum_{i=1}^{N_j}(T_i+1)\omega_{i,j}} \tag{37}$$

  i.e. obstacles neighbors (with transparency near -1) have a nearly null weight; they ignored as usually done when considering Newmann initial conditions.

- t_j be a weight between 0 and 1 dependant on the transparency:

$$t_j = \left[\frac{\tanh GT_j}{\tanh G} + 1\right]/2 \tag{38}$$

  This weight is illustrated in Figure 3 for $G = 3$.

The harmonic function value of the goal cell is fixed at the low value U_L. The harmonic function value of any other m-cell C_j^m is computed as:

$$h_j = U_j t_j + (1 - t_j)U_H \tag{39}$$

i.e. the more transparent a cell is the more it is influenced by its neighbors and the less it is fixed at a high potential value. The non-linear expression of t_j differentiates the behavior of the harmonic function between cells of different level of transparency, highlighting the extreme values.

B. Channel searching and refinement

The search of a channel is done following the negated gradient of the harmonic function H_1. Staring at the initial cell, the next cell is iteratively chosen among the neighbors such that it has the lowest H_1 value, until the goal cell with H_1 value fixed at U_L is reached.

The obtained channel is composed of cells with different transparency values. A further exploration of those cells is done as follows:

a) A minimum value of transparency is required for each channel cell as expressed in the following test:

$$T_j \geq \Delta_{acceptance} \tag{40}$$

When a channel cell does not satisfy this condition then (if the cell already had all its samples collision-checked) a new sample of the cell is generated using Eq. (17); otherwise the collision-check test is applied at the first non-evaluated sample. Afterwards, condition (40) is checked again and if it is not yet satisfied then the cell is partitioned.

b) The transparency of the channel, $T_{channel}$, is defined as the lowest value of transparency of the cells it contains. When the channel found has its transparency above a given threshold, i.e.:

$$T_{channel} \geq \Delta_{channel} \tag{41}$$

then either a good channel has been found or some thin obstacles have been unnoticed. To avoid this latter problem, when Eq. (41) is satisfied, a further sampling is applied like that done in step (a). Afterwards, the partition test is evaluated and if necessary the cell is partitioned.

Finally, once the channel is found, its cells are used as goal cells to compute the harmonic function H_2, whose values module the weight β_{H_2} (Eq. (33)) that influences the way in how the $\mathcal{C}$-space is explored (sampled and partitioned).

VII. The Kautham planner

The proposed approach is summarized as an algorithm in Section VII-A, and some implementation issues and the values of the parameters used are presented in Sections VII-B and VII-C, respectively. Finally, the performance of the proposed approach is evaluated with a test bed with different 2-dof $\mathcal{C}$-spaces (Section VII-D) and 6-dof $\mathcal{C}$-spaces (Section VII-E).

A. Planning algorithm

The planning from an intitial configuration c_{ini} to a goal configuration c_{goal} is performed as shown in Fig.4.

B. Implementation issues

The k-sampler is structured around two lists:

Kautham(c_{ini}, c_{goal})

Find the M-cells $(s_{\text{ini}}, s_{\text{goal}})$ corresponding to $(c_{\text{ini}}, c_{\text{goal}})$

Channel Loop (N times):

Sample Loop (K times):

Get sample from sequence - Eq. (14)

Find cell that contains it - Eq. (28)

Compute the transparency - Eq. (31)

Check collision if condition (32) is satisfied

Partition cell if condition (34) is satisfied

End Sample Loop

Relaxation Loop for H_1 (n_{H1} times):

For each cell compute H_1 - Eq. (39)

End Relaxation Loop

Search Channel from s_{ini} to s_{goal} following $(-\nabla H_1)$

Resample and partition channel cells not satisfying (40)

If (41) is satisfied then:

Resample each channel cell

Partition cell if condition (34) is satisfied

Relaxation Loop for H_2 (n_{H2} times):

For each cell compute H_2 - Eq. (39)

End Relaxation Loop

End Channel Loop

Construct a roadmap with the free samples of the channel

Add c_{ini} and c_{goal} to the roadmap

Search the roadmap for a solution between c_{ini} and c_{goal}

END

Fig. 4. The Kautham algorithm.

parameter	value	used in
β	0.5	Eq. (33)
G	10	Eq. (38)
$\Delta^-_{collsion}$	-0.6	Eq. (32)
$\Delta^+_{collsion}$	0.6	Eq. (32)
$\Delta^-_{partition}$	-0.6 / -0.9	Eq. (34)
$\Delta^+_{partition}$	0.6 / 0.9	Eq. (34)
$\Delta_{channel}$	0.6	Eq. (41)
$\Delta_{acceptance}$	0.6	Eq. (40)
K	10	Algorithm
n_{H1}	10	Algorithm
n_{H2}	1	Algorithm

TABLE II

PARAMETERS USED FOR THE EXPERIMENTS.

- A list of the samples generated by the deterministic sampling sequence. Each sample contains the following information: code number, color and the coordinates of the configuration in $\mathcal{C}$-space.
- A list of cells. Each cell contains the following information: code number, level, transparency, harmonic function values H_1 and H_2, number of samples, number of collision-checked samples, list of neighbor cells and type of cell (initial/goal/channel/normal).

Memory efficiency is obtained by maintaining these lists with the minimum required information. Computing efficiency is obtained by the compact representation of the hierarchical cell decomposition used and the delay of collision-checks as much as possible.

The user interface has been programmed in C++ using the cross-platform tools Qt (as application framework) and Coin3D (as graphics toolkit). Collision detection is performed using the PQP library [18].

C. Parameters

Table II shows the parameters used. They are valid for a wide range of $\mathcal{C}$-spaces, like those 2-dof $\mathcal{C}$-spaces shown in Figure 5 which involve narrow passages, regions crowded with small $\mathcal{C}$-obstacles and spaces with thin $\mathcal{C}$-obstacles (the $\mathcal{C}$-space of Figure 5a is taken from [19]).

D. 2-dof examples

Figure 6 shows the graphical output of the $Kautham$ sampler for the example of Figure 5a using a maximum partition level of $M = 6$. A total number of 812 samples have been generated by the deterministic sampling sequence and 462 have been collision-checked. The total number of cells is 286, being the solution channel composed of 40 cells and having a transparency of $T_{channel} = 0.62$. The 97 samples contained in those cells are the k-samples.

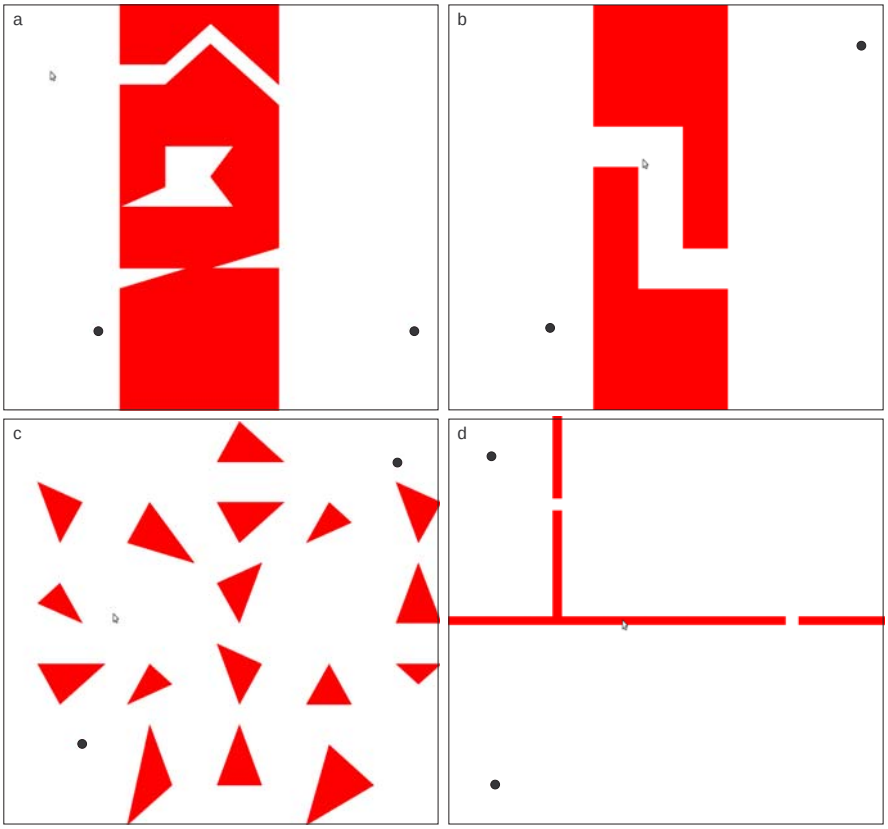

Fig. 5. 2D $\mathcal{C}$-spaces with different features used as a test bed: a) narrow passage, hole and dead ends (taken from [19]); b) bend corridor; c) crowded region with small $\mathcal{C}$-obstacles; d) thin $\mathcal{C}$-obstacles.

Different trials of a basic PRM using the random sampling approach with the same number of collision-checked samples gave no satisfactory results.

E. 6-dof examples

Three 6-dof examples illustrate the proposed approach; they have been obtained from [11], although with minor differences.

Figure 7 shows a known 6-dof example where the $\mathcal{C}$-space has two large open regions with a narrow bend corridor between them. The path is found using the same parameters and 10,000 samples, being 6,944 of them collision-checked. The total number of cells is 2,290, being the solution channel composed of 28 cells with 634 free samples and having a final transparency of $T_{channel} = -0.78$. The solution was found using $M = 5$ and $P = 4$. A supplementary AVI file which contains the execution of the solution path will be available at http://ieeexplore.ieee.org.

Figure **??** shows a 6-dof example with a cluttered environment. The path is found using the same parameters and 3,992 samples, being 2,539 of them collision-checked. The total number of cells is 1,688, being the solution

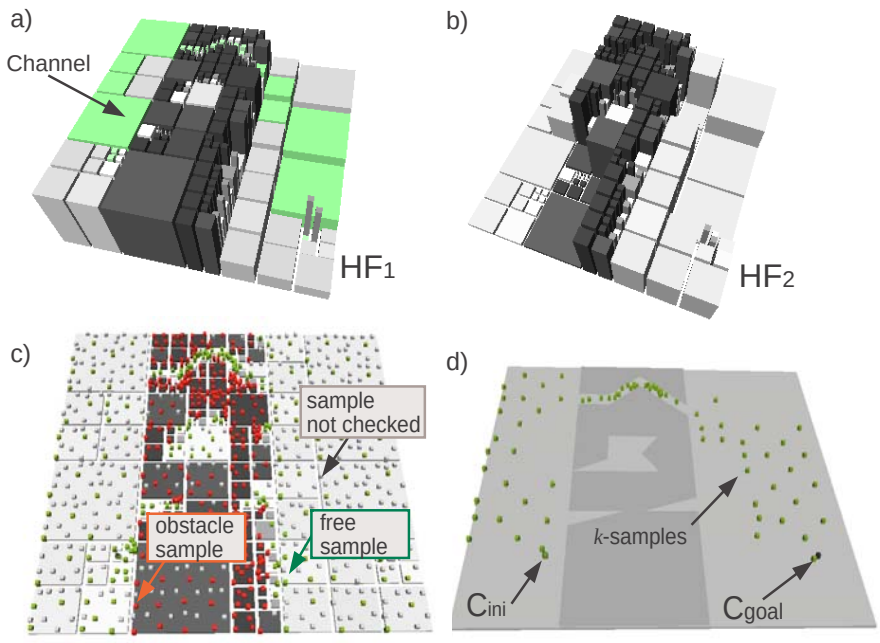

Fig. 6. a) Harmonic functions H_1 with the solution channel shown in green; b) Harmonic functions H_2 (the gray tone of the cells is proportional to their transparency); c) The samples generated, those collision checked are either green (free) or red (obstacles); d) The resulting k-samples.

channel composed of 27 cells with 365 free samples and having a final transparency of $T_{channel} = 0.15$. The solution was found using $M = 5$ and $P = 5$.

Figure 9 shows a 6-dof example with complex objects. The path is found using 4,007 samples, being 1,123 of them collision-checked. The total number of cells is 229, being the solution channel composed of 14 cells with 170 free samples and having a final transparency of $T_{channel} = -0.22$. The solution was found using $M = 5$ and $P = 4$.

VIII. Discussion

The success of sampling-based path planners relies on their ability to use a good set of samples whose interconnection captures the $\mathcal{C}$-space connectivity relevant to the query to be solved.

In this paper a new sampling paradigm, called *Kautham* sampling or simply k-sampling, has been proposed to obtain such a set of samples. k-sampling uses both free and obstacle samples and organizes them as a hierarchical cell decomposition of $\mathcal{C}$-space. This model of the $\mathcal{C}$-space is then used to support the computation of harmonic functions that allow to direct further exploration.

The main features of the proposal are the following:

a) A deterministic sampling sequence is used to allow the exploration of $\mathcal{C}$-space in an uniform and incremental way, while facilitating the organization of samples into cells and the computation of neighborhood relationships. This is the reason why, in comparison to the approaches based on the probabilistic generation of samples, the

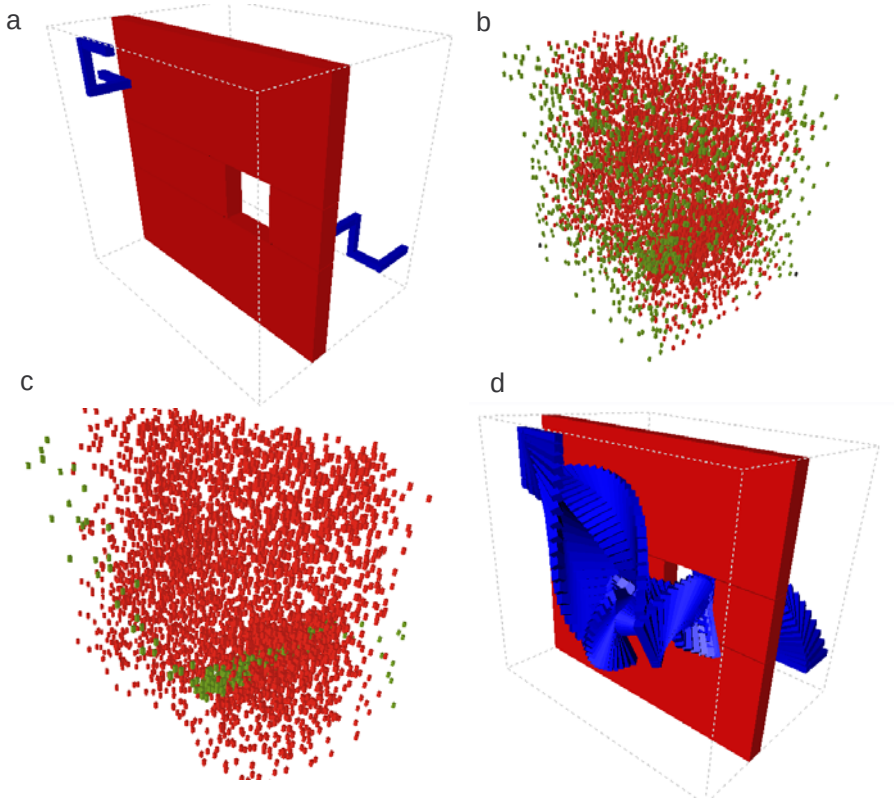

Fig. 7. A 6-dof bend-corridor example: a) Initial and goal configurations; b) Free and obstacle samples; c) k-samples and obstacle samples; d) Solution path.

proposed method results in a better management and profit of the information of the samples. As a drawback, like other methods based on deterministic sampling, the performance can seriously decrease for certain type of artificial $\mathcal{C}$-spaces with thin axis-aligned $\mathcal{C}$-obstacles [20].

b) A lazy-evaluation approach is followed to reduce collision-checks, since not all the samples are collision-checked but only those that lie in more uncertain regions. Uncertainty is measured by a parameter of the cells, called transparency, that considers the number of free and obstacle samples that the cells contain.

Unlike other lazy-evaluation methods, the proposed approach has a broader nature since it is not attached to the query phase but relies on the progressive knowledge of $\mathcal{C}$-space that is incrementally obtained as new samples are generated.

c) Sampling is biased towards more promising regions, i.e. the degree of certainty required for not collision-checking is not fixed for all the cells but is dependant on the region of interest. During the iterative sampling process this region is recomputed as the channel of cells (connecting the cell containing the initial configuration with the cell containing the goal configuration) obtained by following the negated gradient of an harmonic function computed over the hierarchical cell decomposition.

Although it may be argued that the computation of the harmonic functions has a high computational cost, it

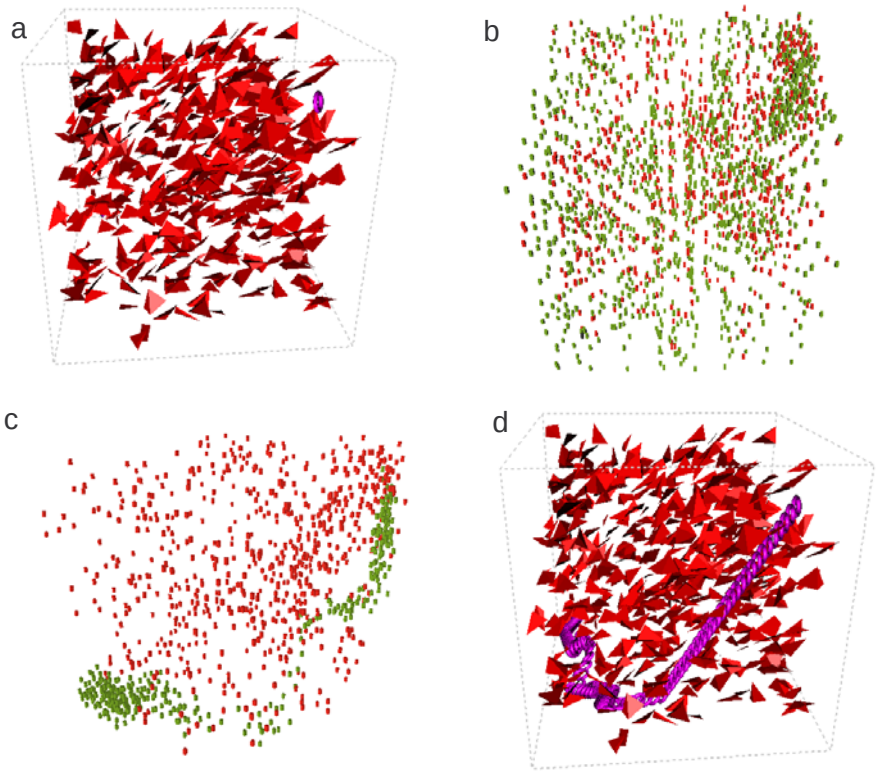

Fig. 8. A 6-dof example with a cluttered environment: a) Initial and goal configurations; b) Free and obstacle samples; c) k-samples and obstacle samples; d) Solution path.

is certain that to solve difficult path planning problems, like those with narrow passages, importance sampling is necessary and, as previous approaches show, the computational cost is always high. The proposed approach uses the harmonic functions as a way to bias the sampling, but once computed, further usage of the harmonic function can be envisioned like its use to generate a guiding force field for haptic-based teleoperation tasks [21].

d) If enough samples have been generated, the iterative sampling process ends with a channel with enough free samples connecting the initial and the goal configurations. Those free samples are easily connected as a roadmap since the neighborhood is implicitly known and the probability to find free paths between them is very high because they belong to cells with a high transparency.

Unlike probabilistic roadmap methods, neighborhood computations are not expensive. The proposed method is conceived as a single query method, thus obtaining a partial roadmap that solves the given query, although like model-based methods, successive single queries may result in a complete roadmap.

Future developments of the *Kautham* planner in order to improve its performance are directed towards considering kd-trees decompositions instead of 2^d-trees, and the possibility to use distance checks.

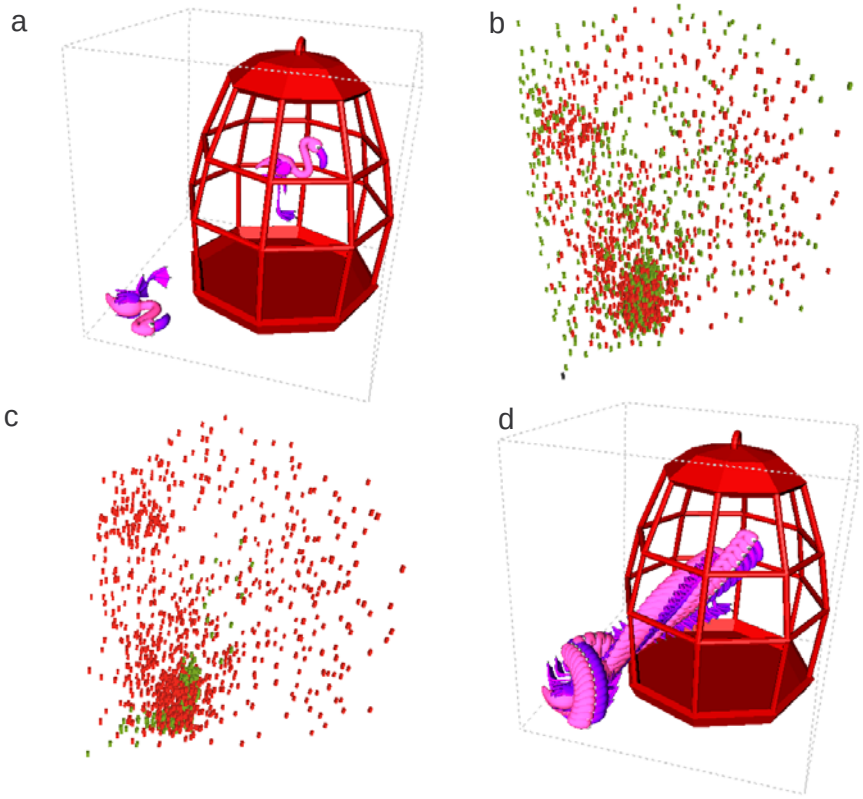

Fig. 9. A 6-dof example with complex objects: a) Initial and goal configurations; b) Free and obstacle samples; c) k-samples and obstacle samples; d) Solution path.

REFERENCES

[1] L. E. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2138–2145, 1994.

[2] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 995–1001, 2000.

[3] B. Burns and O. Brock, "Sampling-based motion planning using predictive models," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3131–3136, 2005.

[4] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. on Robotics and Automation*, vol. 14, pp. 166 –171, Feb. 1998.

[5] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The gaussian sampling strategy for probabilistic roadmap planners," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1018–1023, 1999.

[6] P. Leven and S. Hutchinson, "Using manipulability to bias sampling during the construction of probabilistic roadmaps," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 6, pp. 1020–1026, 2003.

[7] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 4420–4426, 2003.

[8] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1024–1031, 1999.

[9] R. Bohlin and L. Kavraki, "Path planning using lazy PRM," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 521–528, 2000.

[10] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. J. of Robotics Res.*, vol. 23, no. 7-8, pp. 673–692, 2004.

[11] R. Geraerts and M. H. Overmars, "Sampling and node adding in probabilistic roadmap planners," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 165–173, 2006.

[12] B. Burns and O. Brock, "Toward optimal configuration space sampling," in *Proceedings of Robotics: Science and Systems*, (Cambridge, USA), June 2005.

[13] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 467–472, 2004.

[14] J. Rosell and P. Iñiguez, "Path planning using harmonic functions and probabilistic cell decomposition," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1815–1820, 2005.

[15] S. R. Lindemann, A. Yershova, and S. M. LaValle, "Incremental grid sampling strategies in robotics," in *Proc. of the Sixth Int. Workshop on the Algorithmic Foundations of Robotics*, 2004.

[16] J. Rosell, M. Roa, A. Pérez, and F. García, "A general deterministic sequence for sampling d-dimensional configuration spaces." Tech. Report IOC-UPC,2006, submitted to ICRA'07.

[17] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," in *Proc. of the IEEE Int. Conf. on Robotics & Automation*, pp. 2102–2106, 1990.

[18] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," *Computer Graphics*, vol. 30, no. Annual Conference Series, pp. 171–180, 1996.

[19] L. K. Dale and N. M. Amato, "Probabilistic roadmaps - putting it all together," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1940–1947, 2002.

[20] S. R. Lindemann and S. LaValle, *Proc. Int. Symp. on Robotics Research*, ch. Current issues in sampling-based motion planning. Springer-Verlag, 2004.

[21] C. Vázquez and J. Rosell, "Haptic guidance based on harmonic functions for the execution of teleoperated assembly tasks." Tech. Report IOC-UPC,2006, submitted to ICRA'07.