Motion Planning for Haptic Guidance

Jan Rosell, Carlos Vázquez, Alexander Pérez and Pedro Iñiguez

Abstract

Haptic devices allow a user to feel either reaction forces from virtual interactions, or reaction forces reflected from a remote site during a bilateral teleoperation task. Also, guiding forces can be exerted to train the user in the performance of a virtual task, or to assist him to safely teleoperate a robot. The generation of guiding forces rely on the existence of a motion plan that provides the direction to be followed to reach the goal from any free configuration of the configuration space ($\mathcal{C}$-space). This paper proposes a method to obtain such a plan that interleaves a sampling-based exploration of $\mathcal{C}$-space with an efficient computation of harmonic functions. A deterministic sampling sequence (with a bias based on harmonic function values) is used to obtain a hierarchical cell decomposition model of $\mathcal{C}$-space. An harmonic function is iteratively computed over the partially known model using a novel approach. The harmonic function is the navigation function used as motion plan. The approach has been implemented in a planner (called *Kautham* planner) that, given an initial and a goal configuration, it provides: a) a channel of cells connecting the cell that contains the initial configuration with the cell that contains the goal configuration; b) two harmonic functions over the whole $\mathcal{C}$-space: one that guides motions towards the channel and the other that guides motions within the channel towards the goal; c) a path computed over a roadmap built with the free samples of the channel. The harmonic functions and the solution path are then used to generate the guiding forces for the haptic device. The planning approach is illustrated with examples on 2D and 3D workspaces.

Index Terms

Haptic guidance, feedback motion planning, sampling-based methods, deterministic sampling, harmonic functions.

I. Introduction

During the last decade robotics has extended its potential applications by the increasing use of haptic devices, either in medicine, industry, entertainment or education (see [41] for a taxonomy and first introduction to the field of haptics). Haptic devices are being used both as master devices for the teleoperation of robotic manipulators, or in the execution of virtual tasks. Haptic devices provide the user with force feedback, allowing him to feel the reaction forces that arise when the teleoperated device interacts with the environment (e.g. [7], [9]), or to feel those artificial forces that should be sensed when the virtual manipulated object interacts with the virtual environment (e.g. [34], [39]).

Besides these reaction forces, the haptic device can also exert some forces to train the user in the performance of a virtual task, or to assist him to safely teleoperate a robot. Some simple guiding forces may constrain the user motions along a line or curve or over a given working plane or surface, e.g. for a peg-in-hole task a line can be defined along the axis of the hole and the user may feel an increasing force as he moves the peg away from that line. Although these simple guides can already be a good help, some tasks may require more demanding guiding forces to aid the user all along the task execution. This is, for instance, the case of assembly tasks where the part being manipulated must perform compliant motions to successfully accomplish the task [10], of manipulation tasks that must be done on a micro or nano-scale [1], [42]; or in training complex manipulation skills [8].

Feedback motion planning strategies can cope with those guiding requirements. These strategies have arisen in the context of robot path planning when the existence of a nominal solution path is not enough to guarantee the successful performance of a task, even if feedback control laws are used to follow it [27]. Feedback motion planning strategies assume that any unexpected configuration can be achieved (i.e. implicitly consider uncertainty) and therefore provide a feedback plan with the proper action to be applied from any configuration to allow the robot reaching the goal.

Feedback plans can be defined using the gradient descent of potential functions [22], called navigation functions when they have a single minimum at the goal configuration [25], [45]. A kind of navigation function can be obtained from harmonic functions [40]. An harmonic function ϕ on a domain $\Omega \subset \mathbb{R}^n$ is a function that satisfies Laplace's equation:

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

The analytical computation of harmonic functions is a difficult issue in $\mathcal{C}$-spaces with non-trivial obstacles. Alternatively, the solution of the Laplace's equation is usually numerically found using finite difference methods, i.e. by discretizing 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 (extensions to non-regular grids have also been proposed [21], [36]).

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

The complete knowledge of the required discrete and approximate model of $\mathcal{C}$-space is either a difficult issue if it relies on the analytical characterization of the obstacles in $\mathcal{C}$-space, or it may be computationally very expensive if collision detection is to be used and a high resolution is required. An alternative is to rely on sampling-based methods, e.g. [33]. The idea is to iteratively sample, evaluate and classify configurations, and update the cell partition if necessary. This exploration of $\mathcal{C}$-space requires a good uniform coverage, which can be provided by deterministic sampling sequences [31].

The sampling-based characterization of $\mathcal{C}$-space (although as one-dimen-sional paths forming roadmaps [19] or trees [23] capturing the connectivity of the free $\mathcal{C}$-space either globally or relevant to a query), is currently one of the main approaches to robot path planning. The generation of samples and their interconnection is one of the crucial factors in the performance of sampling-based planers. For the haptic guidance purposes, the existence of a solution path, besides the feedback plan, is considered relevant.

This paper proposes a motion planner for free-flying robots that generates both a solution path and a motion plan. Given a single query problem, this plan provides the direction to be followed to reach the goal from any free configuration of the configuration space ($\mathcal{C}$-space) and is used to generate haptic guiding forces. The key issues of the proposed method are: a) a sampling-based exploration of $\mathcal{C}$-space based on deterministic sampling; b) a novel method to compute harmonic functions taking into account the partially known nature of the $\mathcal{C}$-space model; and c) the efficient interleaving of the $\mathcal{C}$-space exploration with the harmonic function computation.

The paper is structured as follows. First, Section II discusses some issues related with sampling and with the computation of harmonic functions. Section III then presents the objectives, outlines the contributions of the proposal and sketches an overview of the whole approach. $\mathcal{C}$-space modelling is tackled in Sections IV and V that present, respectively, a deterministic sampling procedure and the generation of a hierarchical cell decomposition. The computation of harmonic potential functions is tackled in Section VI. The paper ends with Section VII that summarizes and illustrates the planner, Section VIII that shows the application to haptic guidance, and finally Section IX that discusses the contributions.

II. Problem analysis

This section discusses some problems related with some key issues to be tackled: sampling (for the $\mathcal{C}$-space modelling) and the computation of harmonic functions. This discussion helps the establishment of the objectives presented in Section III.

A. Sampling issues

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 [18]. 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. [3], [15], [29], [44]). 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. Even some approaches make this sampling bias adaptive or dynamic [17], [20], [24]. 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.

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 [28] 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. These approaches delay collision-checks until it is absolutely necessary [2]. 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 [4] 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 [33], [36], [46].

B. Computing Harmonic functions

The numerical solution of an harmonic function depends on the boundary conditions: the Dirichlet boundary condition sets the obstacles' border at a fixed high value; the Newmann boundary condition forces the gradient vectors to be tangent to them. Whichever the case, the solution relies on the knowledge of the free or the obstacle nature of the cells. If the model of

$\mathcal{C}$-space is obtained from sampling, then this knowledge can only be partial. Taking into account this fact the following should be considered:

a) Relaxation methods and convergence: The quickest convergence is achieved with the successive over-relaxation method (S.O.R) that computes the harmonic function value of each cell (called HF-value) as a weighted average between the previous iterate and the computed Gauss-Seidel iterate (that which uses the most recent available HF-values of the neighbor cells) [40]. Relaxation methods iterate until the error is below a given threshold. Nevertheless, a small and fixed number of iterations can be considered each time taking into account that the $\mathcal{C}$-space is only partially known and more iterations will be successively performed as further knowledge is acquired [35].

b)Boundary Conditions: Since the free or the obstacle nature of the cells is not known with certainty, a new method must be envisioned to consider the boundary conditions.

III. Objectives and proposed solution

A. Objectives

This paper has as a main objective the proposal of a motion planner for a free-flying robot that, given a single query problem, generates both a motion plan based on harmonic functions and a solution path. These are then used for the generation of forces for the haptic guidance, either to train the user in the performance of a virtual task, or to assist him during a teleoperatation.

This objective can be subdivided in the following subobjectives:

- *The obtaining of a $\mathcal{C}$-space model in a simple and quick incremental way.* This is to be achieved by following a model-based sampling approach using deterministic sampling to give: a) a good uniform coverage of the space; b) a simple sample interconnection; c) the chance to use the current model information to perform a lazy evaluation. A cooperative interleaving between $\mathcal{C}$-space exploration and the computation of the harmonic function is also required to allow importance sampling.
- *The efficient computation of harmonic functions over a partially known model of $\mathcal{C}$-space.* This item requires the adaptation of classical relaxation methods to hierarchical cell decompositions, the consideration of a small and fixed number of iterations and the management of fuzzy boundary conditions.
- *The generation of an haptic-display force field.* A force-field must be generated form the harmonic functions to aid the user in the performance of the task by smoothly guiding him towards a solution channel and, within the channel, towards the goal.

B. Main results

A sampling method, called *Kautham* sampling or k-sampling for short, is proposed to achieve the objective 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.

Deterministic sampling has proven slightly better results than random sampling in roadmap planners as demonstrated in [28] and even acknowledged in [16], although asserting that this is certain for few degrees of freedom.

Our approach contributes with a general, simple and yet efficient deterministic sequence, that is tailored to the modeling of $\mathcal{C}$-space using a hierarchical cell decomposition.

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 [5], [26], not even in a fuzzy manner as done in [36]. 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.

Our approach contributes with a new cell decomposition procedure based on an incremental sampling exploration of $\mathcal{C}$-space, and tailored to be interleaved with the computation of harmonic functions.

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 [27], [40], 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.

Our approach contributes with a new and efficient method to compute harmonic functions over partially known $\mathcal{C}$-spaces, which is tailored to be used as a guide for further exploration.

The k-sampling process identifies the regions where the solution of a single-query problem lies, obtaining a channel of cells of a hierarchical cell decomposition model of $\mathcal{C}$-space. The free samples pertaining to the channel are called k-samples and are used to construct a local roadmap able to find a solution path. Our approach outputs both the solution path and a feedback motion plan composed of the two harmonic functions H_1 and H_2. Both the path and the feedback motion plan are the basis for the generation of haptic guiding forces. When more queries are required in the same environment, the $\mathcal{C}$-space model can be reused.

C. Approach Overview

Given an initial and a goal configuration (respectively c_{ini} and c_{goal}), the k-sampling procedure iteratively executes the following steps:

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) Partitions 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.

The procedure returns the free samples of the channel cells (k-samples) and the harmonic functions H_1 and H_2 over the $\mathcal{C}$-space decomposition model. The k-samples are connected as a roadmap, as well as c_{ini} and c_{goal}, and a solution path between them is easily searched. The harmonic functions conform a feedback motion planning strategy.

IV. Deterministic sampling sequence

For motion planing purposes, deterministic sampling sequences are desired to have the following features [30]: a) uniformity; b) incremental quality; and c) lattice structure. The Halton sequence [14] is a low-discrepancy sequence that satisfies the first two. In order to consider also the third, a digital construction method based on a multiresolution grid was proposed in [32]. This section summarizes a deterministic sampling sequence ($s_d(k)$), we introduced in [37], that follows this latter line. The approach is an efficient and general method, i.e. for any dimension, that is based on: a) a hierarchical decomposition of a unit cube of parameters; b) a low-dispersion ordering of the descendant cells of any given cell of the hierarchical decomposition; c) a recursive application of that ordering to sample the parameter space; and d) the mapping from samples of parameter space to configurations of $\mathcal{C}$-space.

A. Hierarchical cell decomposition

A 2^d-tree decomposition of a d-dimensional unit cube of parameters is considered. 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 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.

M-cells are labeled with a code computed from its indices in the regular grid of level M, as illustrated in Figure 1a for $d = 2$. 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), and also allows an efficient and simple implementation of the deterministic sampling sequence proposed (Section IV-B).

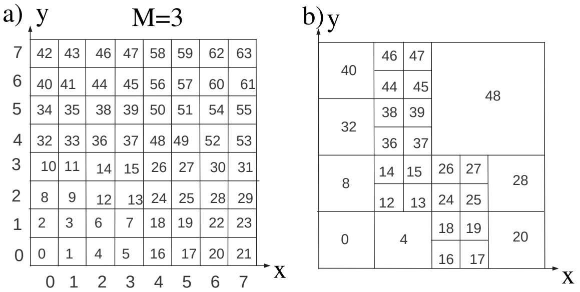

Fig. 1. a) Coding of M-cells; b) Coding of a hierarchical cell decomposition where for example cells 12 and 14 are 3-cells, cells 0 and 4 are 2-cells and cell 48 is a 1-cell.

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

TABLE I

FIRST 20 SAMPLES OF SEQUENCE s_2.

B. 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.

Let $k \geq 0$ be the index of the sequence and:

- T_d be a $d \times d$ binary matrix that defines a low-dispersion ordering of the descendant cells of any given cell, e.g. for $d = 2$ and $d = 3$:

$$T_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \qquad T_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{2}$$

  Reference [37] provides a general method to find T_d for any d.
- V_k^M be a binary $d \times M$ matrix whose rows are the binary representation of the indices $v_j^M \ \ \forall j \in 1 \dots d$ of the M-cell with code k on the regular grid of partition level M.
- W'^M be a $d \times M$ matrix of weights, with:

$$w'_{ij} = 2^{(j-1)d+i-1} \text{ for } i \in 1 \dots d \ \ j \in 1 \dots M \tag{3}$$

Then:

$$s_d(k) = (T_d V_k^M) \cdot W'^M \tag{4}$$

As an example, with $M = 3$ and the expression of T_2 proposed in Eq. (2), the sample corresponding to $k = 6$ is:

$$\begin{aligned} s_2(6) &= \left[\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} 1 & 4 & 16 \\ 2 & 8 & 32 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 & 16 \\ 2 & 8 & 32 \end{pmatrix} = 44 \end{aligned} \tag{5}$$

The first 20 samples generated by $s_2(k)$ 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.

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

Finally, the M-cells of parameter space generated by the sampling sequence are mapped to configurations of $\mathcal{C}$-space as follows.

First, the P-cell to which a sampled M-cell pertains is easily determined using the cell coding detailed in Section IV-A. Let $(w_1^P, \dots, w_d^P)$ be its indices. Then, a point within the P-cell is randomly chosen:

$$x_j = \text{rand}\{[w_j^P s_P, (w_j^P + 1)s_P)\} \qquad \forall j \in 1 \dots d \tag{6}$$

For robot manipulators of d d.o.f. the $\mathcal{C}$-space is the unit cube $[0,1]^d \subset \mathbb{R}^d$ (if the proper scaling is performed) and the parameter space is d-dimensional. Then, the selected point in Eq. (6) is the configuration associated to the sampled M-cell.

For 3D rigid-bodies that can both translate and rotate (free flying robots) the $\mathcal{C}$-space is $SE(3)$. A configuration can be defined by a position (x, y, z) (scaled to the unit cube $[0,1]^3 \subset \mathbb{R}^3$), and an orientation represented by a rotation direction (r_x, r_y, r_z) and a rotation angle θ. The parameter space considered in the present approach will be 3-dimensional, i.e. only used to generate positions. From Eq. (6) with $j \in 1 \dots 3$:

$$\begin{aligned} x &= x_1 && (7)\\ y &= x_2 && (8)\\ z &= x_3 && (9)\end{aligned}$$

Orientations will be randomly chosen within all their ranges using cylindrical coordinates:

$$x_j = \text{rand}\{[0,1)\} \qquad \forall j \in 4 \dots 6 \tag{10}$$

$$\begin{aligned} \alpha &= 2\pi x_4 && (11)\\ r_z &= 1 - 2x_5 && (12)\\ r_y &= \sin\alpha\sqrt{1 - r_z^2} && (13)\\ r_x &= \cos\alpha\sqrt{1 - r_z^2} && (14)\\ \theta &= 2\arccos(x_6) && (15)\end{aligned}$$

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:

$$L_c[j] \le s_i < L_c[j+1] \tag{16}$$

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:

$$color_i = \begin{cases} +2 & \text{if } c_i \text{ is a free configuration} \\ -2 & \text{if } c_i \text{ is an obstacle configuration} \end{cases} \tag{17}$$

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:

$$color_i = \begin{cases} +1 & \text{if } s_i \text{ belongs to a cell with more} \\ & \text{free than obstacle samples} \\ -1 & \text{otherwise} \end{cases} \tag{18}$$

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

$$T_j = \frac{\sum_{i=1}^{i=K_j} color_i}{2K_j} \tag{19}$$

The transparency satisfies $-1 \le T_j \le 1$. 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. (19) and they have $|color_i| = 1$.

Transparency is used as an easy and simple way to locally model the degree of collision of the $\mathcal{C}$-space. An enhanced alternative has been explored to include distance information [38].

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.

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

$$-\beta_{H_2}\Delta_{collision} < T_j < \beta_{H_2}\Delta_{collision} \tag{20}$$

being $\Delta_{collision}$ a fixed threshold with value ranging between 0 and 1; and β_{H_2} a weight ranging between 0 and 1 and dependant on the harmonic function[1] H_2:

$$\beta_{H_2} = 0.5 - 0.5H_2 \tag{21}$$

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 (20) has more difficulties to be satisfied resulting in less collision checks.

If condition (20) 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 (obviously whenever m is not the maximum partition level, i.e. $m < P$ as defined in Section IV-A). 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:

- 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:

$$-\beta_{H_2}\Delta_{partition} < T_j < \beta_{H_2}\Delta_{partition} \tag{22}$$

being the weight β_{H_2} defined in Eq. (21) and $\Delta_{partition}$ a threshold ranging between 0 and 1, and taking two possible values: a lower value when the cell does not contain evaluated samples of different color, and a higher value otherwise.

If condition (22) 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. Some illustrative examples can be found in [38].

VI. Harmonic Functions

This section first introduces the computation of the harmonic function proposed (Section VI-A), and then discusses its use to search the solution channel and further explore the $\mathcal{C}$-space (Section VI-B).

A. Harmonic function values

Harmonic functions are computed over a hierarchical cell decomposition using the successive over-relaxation method (S.O.R) that computes for each cell a weighted average between the previous iterate and the computed Gauss-Seidel iterate [40]. The proposed approach takes into account that the cell decomposition used is non-regular and that the knowledge of the collision nature of the cells is only partially known. Let:

- N_j be the number of neighbors of an m-cell C_j^m in the hierarchical cell partition.
- $\omega_{i,j}$ be the size of the border between cell C_i^n and cell C_j^m measured in M-cells:

$$\omega_{i,j} = 2^{(d-1)(M-\max(m,n))} \tag{23}$$

- U_H and U_L be, respectively, the high and low value of the harmonic function. They are fixed to $U_H = 0$ and $U_L = -1$.

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

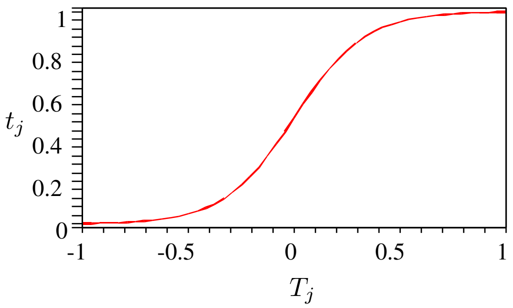

Fig. 2. Parameter t_j as a function of T_j, using $Q = 3$ (Eq. 27). This parameter weights the influence of the neighborhood in the computation of the harmonic function values of a cell.

- U_j be the harmonic function value of cell C_j^m.
- G_j be the Gauss-Seidel average of the neighbors of cell C_j^m:

$$G_j = \frac{\sum_{i=1}^{N_j}(T_i+1)\omega_{i,j}U_i}{\sum_{i=1}^{N_j}(T_i+1)\omega_{i,j}} \tag{24}$$

  This average:

  - Is computed in the Gauss-Seidel manner, i.e. using always the most recent available HF-values of the neighbor cells.
  - Weights the influence of the neighbors a as function of the seize of the border.
  - Considers the boundary conditions as Newmann boundary conditions but modified by the transparency, i.e. the lesser transparent a cell is the smaller its weight (the limit being the cells with transparency equal to -1 that have a null weight, as considered in the Newmann boundary conditions).

- S_j^k be the S.O.R average of cell C_j^m computed at iteration k:

$$S_j^k = (1-w)G_j + wS_j^{(k-1)} \tag{25}$$

  For an optimum convergence the following value of w is proposed [13]:

$$w = \frac{2}{1+\sqrt{1-\cos(\pi/h)}} \tag{26}$$

  where h is the number of partitions of each axis. For a partition level $P = 5$ the value of h is 32 and the optimum w is 1.82.

If the knowledge of the $\mathcal{C}$-space were complete, i.e. only containing cells with transparency -1 or 1, the value of the S.O.R. average, S_j, would be considered the value of the harmonic function. Nevertheless, since in our approach the $\mathcal{C}$-space is only partially known, the following is proposed. Let t_j be a weight between 0 and 1 dependant on the transparency (Figure 2):

$$t_j = \left[\frac{\tanh QT_j}{\tanh Q} + 1\right]/2 \tag{27}$$

Then, the harmonic function value of an m-cell C_j^m is computed as:

$$U_j = S_j t_j + (1-t_j)U_H \tag{28}$$

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.

To be exact, expression (28) only represents an harmonic functions for the extreme values of t_j (0 and 1). This fact may lead to local minima during the exploration process while the transparency of cells is changing and taking any of its possible values. This is not a serious problem since further exploration within the loop of the main algorithm usually reverses the situation.

Eq. (28) is used for all cells except for the goal one, whose value is fixed at the low value U_L. Eq. (28) is applied only a small and fixed number of times, i.e. the relaxation method does not iterate until convergence. The reason has to be understood in the scope of the interleaving of the $\mathcal{C}$-space exploration and the relaxation of the harmonic function, i.e. more iterations will be successively performed as further knowledge is acquired.

parameter	value	used in
Q	10	Eq. (27)
$\Delta_{collsion}$	0.6	Eq. (20)
$\Delta_{partition}$	0.6 / 0.9	Eq. (22)
$\Delta_{channel}$	0.6	Eq. (30)
$\Delta_{acceptance}$	0.6	Eq. (29)
K	10	Algorithm
n_{H1}	10	Algorithm
n_{H2}	1	Algorithm

TABLE II
PARAMETERS USED FOR THE EXPERIMENTS.

B. Channel searching and refinement

The search of a channel is done following the negated gradient of the harmonic function H_1. Starting 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 proposed as follows:

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

$$T_j \geq \Delta_{acceptance} \tag{29}$$

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; otherwise the collision-check test is applied at the first non-evaluated sample. Afterwards, condition (29) 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.:

$$T_{channel} \geq \Delta_{channel} \tag{30}$$

then either a good channel has been found or some thin obstacles have been unnoticed. To avoid this latter problem, when Eq. (30) 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. (21)) that influences the way the $\mathcal{C}$-space is explored (sampled and partitioned).

VII. THE KAUTHAM PLANNER

The proposed approach is summarized as an algorithm in Figure 3. Some implementation issues and the values of the parameters used are discussed in Section VII-A and the performance of the proposed approach is evaluated in Section VII-B with a test bed with different 2-dof and 6-dof $\mathcal{C}$-spaces.

A. Implementation issues

The k-sampler is structured around two lists:

- 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 [12].

The parameters used are shown in Table II. They are valid for a wide range of $\mathcal{C}$-spaces, like those 2-dof $\mathcal{C}$-spaces shown in Figure 4 which involve narrow passages, regions crowded with small $\mathcal{C}$-obstacles and spaces with thin $\mathcal{C}$-obstacles.

```
**Kautham**($c_{\text{ini}}$, $c_{\text{goal}}$)
  Find the $M$-cell $s_{\text{goal}}$ that contains $c_{\text{goal}}$
  Channel Loop ($N$ times):
    Sample Loop ($K$ times):
        Get sample from sequence - Eq. (4)
        Find cell that contains it - Eq. (16)
        Compute the transparency - Eq. (19)
        Check collision if condition (20) is satisfied
        Partition cell if condition (22) is satisfied
    End Sample Loop
    Relaxation Loop for $H_1$ ($n_{H1}$ times):
        For each cell compute $H_1$ - Eq. (28)
    End Relaxation Loop
    Search Channel from $s_{\text{ini}}$ to $s_{\text{goal}}$ following $(-\nabla H_1)$
    Resample and partition channel cells not satisfying (29)
    If (30) is satisfied then:
        Resample each channel cell
        Partition cell if condition (22) is satisfied
    Relaxation Loop for $H_2$ ($n_{H2}$ times):
        For each cell compute $H_2$ - Eq. (28)
    End Relaxation Loop
  End Channel Loop
  Construct a roadmap $R$ with the free samples of the channel
  Add $c_{\text{ini}}$ and $c_{\text{goal}}$ to the roadmap
  Search $R$ for a solution path $p$ between $c_{\text{ini}}$ and $c_{\text{goal}}$
  Return $H_1$, $H_2$ and $p$
END
```

Fig. 3. The Kautham algorithm.

B. Examples

Figure 5 shows the graphical output of the $Kautham$ sampler for the example of Figure 4a using $M = 6$ and $P = 6$. 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 that are easily connected as a roadmap. Without considering the roadmap, the solution was found in 1.8s on a standard PC.

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

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

Figure 6 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 4,000 samples, being 3,739 of them collision-checked. The total number of cells is 505, being the solution channel composed of 22 cells with 412 free samples and having a final transparency of $T_{channel} = -0.56$. The solution was found in 8.48s, using $M = 5$ and $P = 3$.

Figure 7 shows a 6-dof example with a cluttered environment. The path is found using the same parameters and 2,000 samples, being 1,795 of them collision-checked. The total number of cells is 512, being the solution channel composed of 22 cells with 143 free samples and having a final transparency of $T_{channel} = 0.0$. The solution was found in 4.02s, using $M = 5$ and $P = 3$.

Figure 8 shows a 6-dof example with complex objects. The path is found using 1,000 samples, being 974 of them collision-checked. The total number of cells is 505, being the solution channel composed of 13 cells with 96 free samples and having a final transparency of $T_{channel} = -0.33$. The solution was found in 5.13s, using $M = 5$ and $P = 3$.

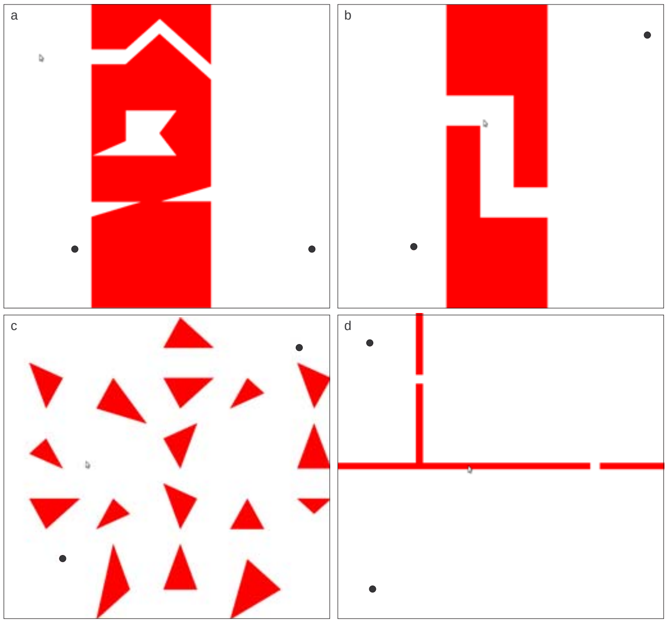

Fig. 4. 2D $\mathcal{C}$-spaces with different features used as a test bed: a) narrow passage, hole and dead ends (taken from [6]); b) bend corridor; c) crowded region with small $\mathcal{C}$-obstacles; d) thin $\mathcal{C}$-obstacles.

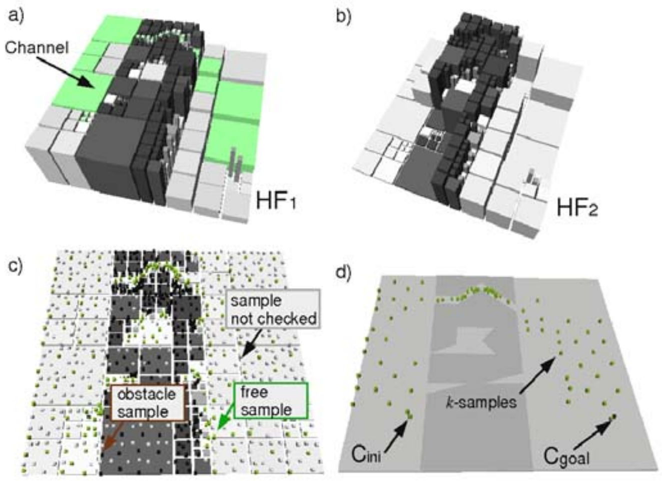

Fig. 5. a) Harmonic functions H_1 with the solution channel shown in green; b) Harmonic functions H_2 (the gray tone of the cells is proportional to their transparency); c) The samples generated, those collision checked are either green (free) or red (obstacles); d) The resulting k-samples.

VIII. Application to Haptic Guidance

The availability of a haptic programming toolkit with the following simple primitives is assumed:

Point attraction primitive: generates a force directed towards a given configuration when the haptic interface point is within a given distance from it.

Segment attraction primitive: generates a force directed towards a given segment defined by its two extreme configurations, provided that the haptic interface point is within a given distance from the segment.

Then, using the harmonic functions and the solution path output by the *Kautham* planner, the haptic guidance is performed at three levels [43]:

$\mathcal{C}$-space guidance: It is felt on all the cells of $\mathcal{C}$-space that do not pertain to the solution channel. This guiding force pushes the user towards the solution channel. It is computed using H_2.

Channel guidance: It is felt on the cells of the channel. This force guides the user within the channel towards the goal cell. It is computed using H_1.

Path guidance: It is felt near the path and pushes the user towards it.

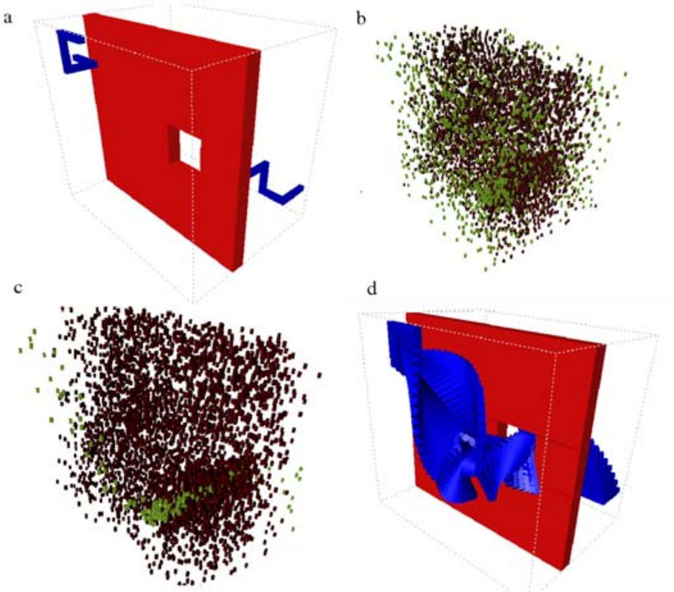

Fig. 6. A 6-dof bend-corridor example: a) Initial and goal configurations; b) Free and obstacle samples; c) k-samples and obstacle samples; d) Solution path.

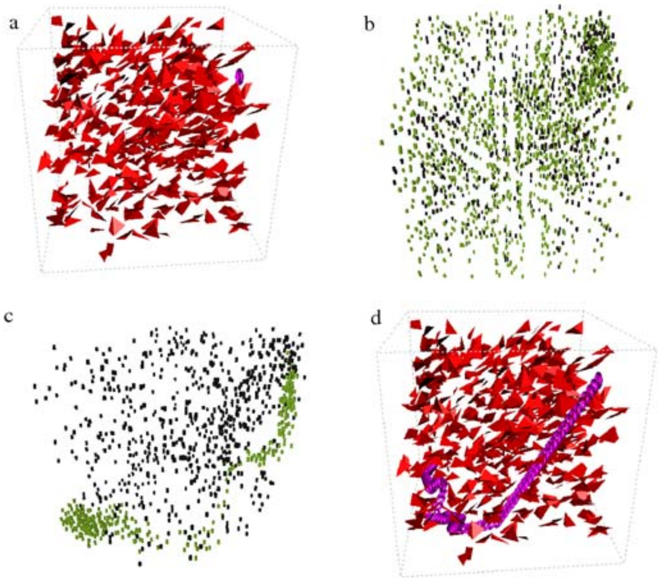

Fig. 7. A 6-dof example with a cluttered environment: a) Initial and goal configurations; b) Free and obstacle samples; c) k-samples and obstacle samples; d) Solution path.

Path guidance uses the segment attraction primitive to generate forces towards the path, that is formed by a set of linear segments of $\mathcal{C}$-space. Both the $\mathcal{C}$-space guidance and the channel guidance iteratively use the point attraction primitive to generate attractive force towards the centers of the next cell following the negated gradient of the harmonic functions.

This iterative procedure is as follows. For each cell i with side s_i the following two balls are defined:

- B^i_{home}: A ball of radius $\frac{s_i}{2}$ centered at the cell center.
- B^i_{next}: A ball centered at the center of the neighbor cell j with lowest HF-value and radius $r_{i,(i+1)} = d_{i,(i+1)} + \frac{s_i}{2}$, with $d_{i,(i+1)}$ being the distance between centers (Figure 9).

Then, starting within the ball B^i_{home} of a given cell i, an attractive force is set that guides the haptic device towards the center of B^i_{next}. This force is felt inside B^i_{next} until the haptic interface point enters the ball $B^{(i+1)}_{home}$ (note that B^i_{next} and $B^{(i+1)}_{home}$ are concentric balls). At this point the force changes to become an attractive force centered at $B^{(i+1)}_{next}$. Note that the radius of B^j_{next} is defined in such a way that the ball B^j_{next} contains B^j_{home} and therefore a continuity in the guiding is guaranteed.

This procedure is repeated iteratively until a final cell is reached (for the $\mathcal{C}$-space guidance the final cells are the channel cells; for the channel guidance the final cell is the goal cell). Final cells have empty B_{next} balls.

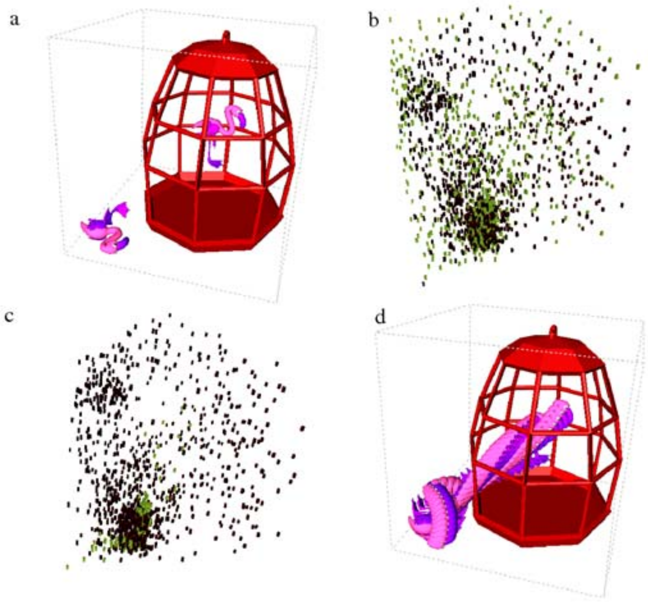

Fig. 8. A 6-dof example with complex objects: a) Initial and goal configurations; b) Free and obstacle samples; c) k-samples and obstacle samples; d) Solution path.

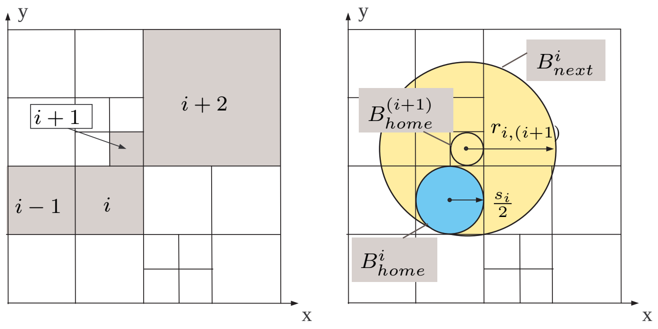

Fig. 9. Solution channel (left); balls B^i_{home} and B^i_{next} (right).

Figure 10 illustrates a 2D teleoperation task. The use of guiding forces has reduced up to a 30% the execution time and made the task less demanding for the operator.

IX. Summary

This paper has proposed a motion planner for free-flying robots that, given a single query problem, generates both a motion plan based on harmonic functions and a solution path. These are used for the generation of forces for the haptic guidance, either to train the user in the performance of a virtual task, or to assist him during a teleoperatation.

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.
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.
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.
d) A novel approach to compute an harmonic function over a hierarchical cell decomposition model of a partially known $\mathcal{C}$-space has been proposed, using the transparency as a way to modulate the propagation of the HF-values.

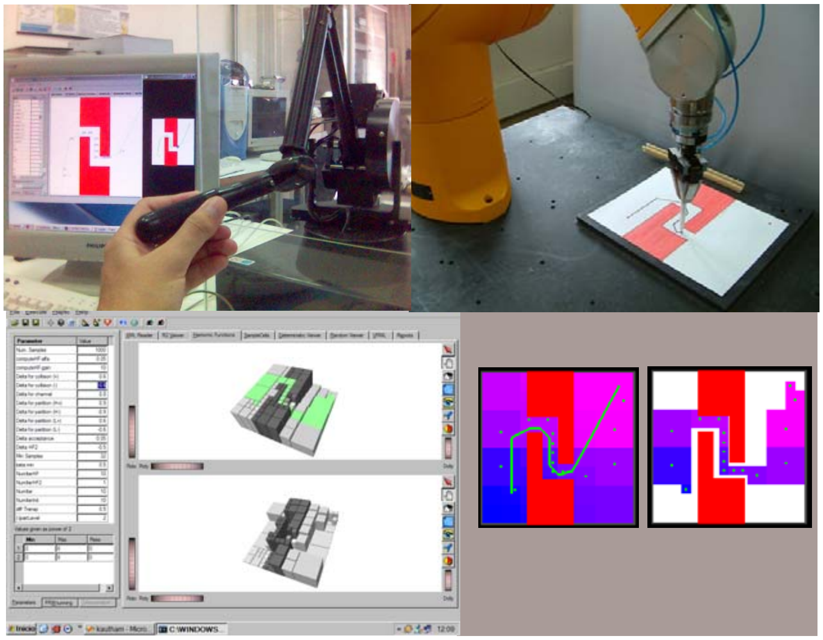

Fig. 10. Teleoperation task (top) using haptic guiding forces generated from H_1, H_2 and the solution path (bottom).

e) The planner outputs two harmonic functions over the whole $\mathcal{C}$-space to be used as a feedback motion plan: H_2 guides motions towards the channel and H_1 guides motions within the channel towards the goal. Also a path computed over a roadmap is easily built with the free samples of the channel, 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. Both the harmonic functions and the solution path are used to generate haptic guiding forces.

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. Also, the generation of a continuous force field within the cells will be studied to improve the guiding.

References

[1] C Basdogan, A. Kiraz, I. Bukusoglu, A. Varol, and S. Doganay. Haptic guidance for improved task performance in steering microparticles with optical tweezers. *Optics Express*, 15(18):11616–11621, 2007.

[2] R. Bohlin and L. Kavraki. Path planning using lazy PRM. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 521–528, 2000.

[3] 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*, pages 1018–1023, 1999.

[4] Brendan Burns and Oliver Brock. Toward optimal configuration space sampling. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[5] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion*, chapter Cell decompositions, pages 161 – 196. The MIT Press, 2005.

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

[7] I. Elhajj, N. Xi, W. K. Fung, Y. H. Liu, W.J. Li, T. Kaga, and T. Fukuda. Haptic information in internet-based teleoperation. *IEEE/ASME Transactions on Mechatronics*, 6(3):295 – 304, 2001.

[8] D. Feygin, M. Keehner, and F. Tendick. Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill. In *HAPTICS'02: Proc. of the 10th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 40–47, 2002.

[9] I. Font, S. Weiland, M. Franken, M. Steinbuch, and L. Rovers. Haptic feedback designs in teleoperation systems for minimal invasive surgery. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 3, pages 2513 – 2518, 2004.

[10] D. Galeano and S. Payandeh. Artificial and natural force constraints in haptic-aided path planning. In *IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications*, 2005.

[11] R. Geraerts and M. H. Overmars. Sampling and node adding in probabilistic roadmap planners. *Robotics and Autonomous Systems*, 54(2):165–173, 2006.

[12] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Computer Graphics*, 30(Annual Conference Series):171–180, 1996.

[13] L. A. Hageman and D. M. Young. *Applied Iterative Methods*. Academic Press, 1981.

[14] J. Halton. On the effciency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90, 1960.

[15] 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*, pages 4420–4426, 2003.

[16] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Int. Journal of Robotics Research*, 25(7):627 – 643, 2006.

[17] D. Hsu, G. Sanchez-Ante, and Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3874 – 3880, 2005.

[18] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Trans. on Robotics and Automation*, 14(1):166 –171, Feb. 1998.

[19] 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*, volume 3, pages 2138–2145, 1994.

[20] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. K. Overmars. Probabilistic roadmaps for path planning in high - dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, August 1996.

[21] M. Kazemi, M. Mehrandezh, and K. Gupta. An incremental harmonic function-based probabilistic roadmap approach to robot path planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2136 – 2141, 2005.

[22] O. Khatib. Real-time obstacle aboidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90 – 98, 1986.

[23] 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*, pages 995–1001, 2000.

[24] H. Kurniawati and D. Hsu. Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning. In S. Akella and et.al., editors, *Algorithmic Foundations of Robotics VII*. Springer–Verlag, 2006.

[25] J.-C. Latombe. *Robot Motion Planning*, chapter Potential Field Methods, pages 296 – 355. Kluwer Academic Publishers, 1991.

[26] J.-C. Latombe. *Robot Motion Planning*, chapter Approximate cell decomposition, pages 248 – 294. Kluwer Academic Publishers, 1991.

[27] S. M. LaValle. *Planning Algorithms*, chapter Feedback Motion Planning, pages 367 – 430. Cambridge University Press, 2006.

[28] S. M. LaValle, M. S. Branicky, and S. R. Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *Int. Journal of Robotics Research*, 23(7-8):673–692, 2004.

[29] P. Leven and S. Hutchinson. Using manipulability to bias sampling during the construction of probabilistic roadmaps. *IEEE Trans. on Robotics and Automation*, 19(6):1020–1026, 2003.

[30] S. R. Lindemann and S. M. LaValle. Incremental low-discrepancy lattice methods for motion planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2920–2927, 2003.

[31] S. R. Lindemann and S. M. LaValle. *Proc. 8th Int. Symp. on Robotics Research*, chapter Current issues in sampling-based motion planning. Springer-Verlag, 2004.

[32] 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*, pages 297 – 312, 2004.

[33] F. Lingelbach. Path planning using probabilistic cell decomposition. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 467–472, 2004.

[34] M.A. Otaduy and M. C. Lin. A modular haptic rendering algorithm for stable and transparent 6-dof manipulation. *IEEE Transactions on Robotics*, 22(4):751 – 762, 2006.

[35] E. Prestes, P. M. Engel, M. Trevisan, and M. A. P. Idiart. Exploration method using harmonic functions. *Robotics and Autonomous Systems*, 40:25–42, 2002.

[36] 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*, pages 1815–1820, 2005.

[37] J. Rosell, M. Roa, A. Pérez, and F. García. A general deterministic sequence for sampling d-dimensional configuration spaces. *J. of Intelligent and Robotic Systems*, 50(4):361–374, 2007.

[38] J. Rosell, C. Vázquez, and A. Pérez. Cspace decomposition using deterministic sampling and distances. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 15 – 20, 2007.

[39] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Proc. of the 24th annual Conf. on Computer graphics and Interactive Techniques*, pages 345–352, 1997.

[40] K. Souccar, J. Coelho, C. Connolly, and R. Grupen. *Practical Motion Planning in Robotics*, chapter Harmonic Functions for Path Planning and Control, pages 277–301. John Wiley & Sons Ltd, 1998.

[41] M. A. Srinivasan and C. Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computer and Graphics*, 21(4):393–404, 1997.

[42] A. Varol, I. Gunev, and C. Basdogan. A virtual reality toolkit for path planning and manipulation at nano-scale. In *14th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 485 – 489, 2006.

[43] C. Vázquez and J. Rosell. Haptic guidance based on harmonic functions for the execution of teleoperated assembly tasks. In *Preprints of the 2007 IFAC Workshop on Intelligent Assembly and Disassembly*, pages 88–93, 2007.

[44] 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*, pages 1024–1031, 1999.

[45] L. Yang and S. M. LaValle. The sampling-based neighborhood graph: An approach to computing and executing feedback motion strategies. *IEEE Trans. on Robotics and Automation*, 20(3):419– 432, 2004.

[46] L. Zhang, Y. J. Kim, and D. Manocha. A hybrid approach for complete motion planning. In *Accepted to IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.