UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Use of path planning techniques based on harmonic functions for the haptic guidance of teleoperated assembly tasks

Carlos Vázquez, Jan Rosell.

IOC- Divisió de Robòtica

IOC-DT-P-2007-2
Febrer 2007

Institut d'Organització i Control
de Sistemes Industrials

Use of path planning techniques based on harmonic functions for the haptic guidance of teleoperated assembly tasks

Carlos Vázquez and Jan Rosell

Abstract

Haptic devices allow the user to feel the reaction forces and torques that arise when the virtual object attached to the user-manipulated probe touches the other objects in the virtual environment. Additionally, the user may feel some guiding constraints and forces that aid him in the completion of a virtual task. Also, haptic devices are used as master devices in teleoperation tasks and may include force reflection from the real forces sensed by a force sensor located at the robot wrist. Both operation modes can be combined to set up an assisted teleoperation system able to execute assembly tasks. This paper proposes the use of path planning techniques based on harmonic functions to generate a guiding force that aids the user during the teleoperation.

Keywords: Haptic devices, teleoperation, assembly tasks, path planning, 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 [1] 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. [2][3]), or to feel those artificial forces that should be sensed when the virtual manipulated object interacts with the virtual environment (e.g. [4][5]).

Besides these reaction forces, either from the real or virtual world, the haptic device can also exert some forces to aid the user in the execution of the teleoperated task. 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 robot tool with the peg away from that line. Although these simple guides can already be a good help, some teleoperated tasks may require more demanding guiding forces to aid the user all along the task execution. This is, for instance, the case with assembly tasks where the part being manipulated by the robot must either perform compliant motions to successfully accomplish the task [6], or move on a micro or nano-scale [7].

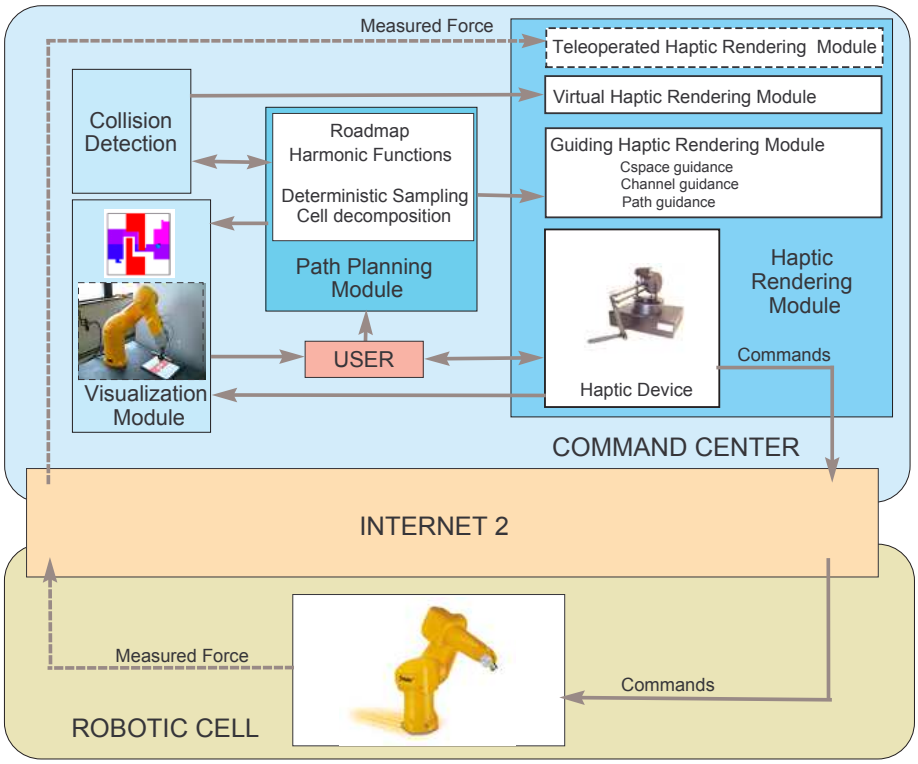

Fig. 1. System setup.

With this assistance aim, this paper proposes the generation of a guiding force using path planning techniques based on harmonic functions. This harmonic functions are computed over an approximate cell decomposition of Configuration Space ($\mathcal{C}$-space). A simple method is then introduced to generate the guiding force from the harmonic function values of the cells of the $\mathcal{C}$-space decomposition.

II. Approach Overview

Fig. 1 shows the assisted teleoperation system setup, with the two main modules that currently constitute it:

Path Planning Module. This module receives as input the model of the assembly task to be teleoperated and its starting and goal configurations. It outputs a path that connect those configurations, a channel of cells that connect the cells that contain them, and two harmonic functions.

Haptic Rendering Module. This module sends to the haptic device the combination of the following forces: a) the real force sensed by the force sensor attached to the wrist of the robot; b) a guiding force computed from the output of the path planning module; and c) a virtual reaction force computed from the interpenetration between the manipulated object and the obstacles in the virtual environment.

The user reacts at that force moving the haptic device accordingly. The motions done are sent to the teleoperated robot. Although this is the intended complete system, currently the teleoperation is not yet bilateral, i.e. the real

forces measured by the sensor attached at the robot wrist are not fed back (indicated with dashed lines in Fig. 1).

The teleoperation scheme set up at the IOC's Robotic Lab is described in [8]. It makes use of a communication channel passivation done using wave variables with an impedance adaptation and TCP/UDP and IPv6 protocols.

The paper describes the main ideas of the path planning module in Section III and presents the haptic rendering module in Section IV. Finally, Section V illustrates the approach in a simple 2D example and Section VI concludes the work.

III. Path planning

The planning of collision-free paths for a manipulated part through the other parts or obstacles in an assembly is a difficult problem that is usually tackled using a configuration space approach. In $\mathcal{C}$-space the manipulated part is mapped to a point and the obstacles are enlarged accordingly ($\mathcal{C}$-obstacles). The $\mathcal{C}$-space explicitly captures the motion freedom of the manipulated part, which facilitates path planning.

Of the several path planning techniques that exist, those based on potential field methods are the best alternatives, although not for very high dimensional $\mathcal{C}$-spaces since usually they require an approximate decomposition of $\mathcal{C}$-space. Among them, those based on harmonic functions are interesting because they give rise to practical, resolution-complete planners without local minima [9]. To compute the $\mathcal{C}$-space decomposition, hierarchical structures can be considered in order to reduce the number of cells; also, the combination with sampling techniques may overcome the requirement of the difficult characterization of the $\mathcal{C}$-obstacles [10]. The solution found by these planners is a channel of cells connecting the cell that contains the initial configuration with the cell that contains the goal configuration.

Another family of successful path planners are those based on the sampling of $\mathcal{C}$-space and the construction of a roadmap [11]. The roadmap is obtained by connecting the free sampled configurations with free paths using a simple local planner. The initial and the goal configurations are then connected to the roadmap and a free path connecting them can be found using graph searching techniques.

The path planning module considered in this paper, described in detail in [12], combines a sampling based approach with an harmonic function based approach. The combination relies on the alternating execution of two steps:

a) The exploration of the $\mathcal{C}$-space and the construction of the hierarchical cell decomposition that consists in the iterative sampling of configurations, their classification into cells and in the partition of those cells that are not homogeneous enough (Section III-A).

b) the computation of an harmonic function to find a channel connecting the cell containing the initial configuration with the cell containing the goal one (Section III-B).

A. Cell decomposition

The $\mathcal{C}$-space is considered as a d-dimensional unit cube; it is partitioned using a 2^d-tree decomposition. The initial cell with sides with unitary size is the tree root. The levels in the tree are called partition levels and are

enumerated such that the tree root is the partition level 0 and the maximum resolution[1] corresponds to partition level M. Cells of a given partition level m are called m-cells.

The 2^d-tree decomposition is obtained by the iterative sampling of configurations and partitioning of cells. Sampling is performed using a deterministic sequence that uniformly and gradually covers the $\mathcal{C}$-space [13].

Partitioning is done as follows. The proposed approach does not classify cells between free and obstacle cells, but with a continuous parameter, called transparency, computed as a function of the number of free or obstacle samples that the cell contains, i.e. it captures the homogeneity of a cell:

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

with K_j being the number of samples of the cell and

$$color_i = \begin{cases} +1 & \text{if } c_i \text{ is a free configuration} \\ -1 & \text{if } c_i \text{ is an obstacle configuration} \end{cases} \tag{2}$$

When the transparency is within a given interval around zero the cell is considered to be not homogeneous enough and then it is partitioned into its 2^d descendant cells, and the transparency for each descendant cell is then recomputed.

B. Harmonic functions

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

$$\nabla^2 \phi = \sum_{i=1}^{n} \frac{\partial^2 \phi}{\partial x_i^2} = 0 \tag{3}$$

The solution of the Laplace's equation is usually found numerically using finite difference methods, i.e. by constraining ϕ on a regular grid and using relaxation methods that iteratively update the value of a cell as the mean of its neighbor cells. An extension to non-regular grids is also possible, as done in [10] using a 2^d-tree decomposition.

In this paper the same approach is followed, although the harmonic function is not only computed over the free cells (fixing the obstacle cells at a high value), as it is usually done, since this cell classification is not considered here. Instead, the harmonic function values (called HF-values) are computed over the whole set of cells using the transparency as a weighting parameter, except for the goal cell that is fixed at a low value $U_L = -1$. The computation is done as follows:

First, for any cell C_j^m a weighted mean, U_j, of the HF-values of the cells neighboring C_j^m is computed. The sizes of the border between cells are used as weights, such that the bigger a neighbor cell is the greater influence it has on U_j:

$$U_j = \frac{\sum_{i=1}^{N_j} \omega_{i,j} u_{i,j}}{\sum_{i=1}^{N_j} \omega_{i,j}} \tag{4}$$

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

with N_j being the number of neighbors, $u_{i,j}$ the harmonic function value of a neighbor cell C_i^n, and $\omega_{i,j}$ the size of the border between C_i^n and C_j^m measured in M-cells:

$$\omega_{i,j} = 2^{(d-1)(M-\max(m,n))} \tag{5}$$

Then, the HF-value of the cell is computed in such a way that the more transparent the cell is the more it is influenced by its neighbors and the less it is fixed at a high potential value, i.e. if U_H is the high value of the harmonic function (fixed at $U_H = 0$), the HF-value is computed as:

$$H_j = \frac{1}{2}[U_j(1 + T_j) + (1 - T_j)U_H] \tag{6}$$

with $T_j \in [-1, 1]$ being the transparency of C_j^m.

Using this procedure, two harmonic functions are computed over the $\mathcal{C}$-space:

HF1: This harmonic function is computed by considering as goal cell the cell that contains the goal configuration, i.e. the HF-value of this cell is set at the low value U_L. The HF-values of all the other cells are computed by relaxation, i.e. by iteratively (re)computing them until a channel (called solution channel) from the cell containing the initial configuration to the goal cell is found.

The solution channel is computed following the negated gradient of the harmonic function, i.e. starting at the initial cell each next cell in the channel is selected as the neighboring one with lowest HF-value.

HF2: This harmonic function is computed by considering as goal cells all the cells of the channel obtained using HF1. i.e. the HF-value of these cells are set at the low value U_L. The HF-values of all the other cells are also computed by relaxation.

Following the negated gradient of HF2 from any cell leads to the nearest cell of the solution channel.

C. Roadmap

The samples contained inside the channel cells are connected using a simple local planner. Since the transparency of channels cells is high, these samples are usually connected successfully. The resulting roadmap is searched using the A* algorithm to obtain a solution path connecting the initial and the goal configurations.

IV. The Haptic Guidance

It is assumed that the user allways starts the assembly task around a given starting configuration, and that the task completion is determined by a given goal configuration. Those configurations and the task model are the input to the path planning module, being the outputs: a) the two harmonic functions HF1 and HF2; b) the channel of cells that connects the cell containing the initial configuration with the cell that contains the goal configuration; and c) a solution path through the channel that connects the starting and goal configurations and that is computed using the configurations of the channel cells.

This path planning output is used by the haptic module to compute the haptic guidance, that is performed at three levels:

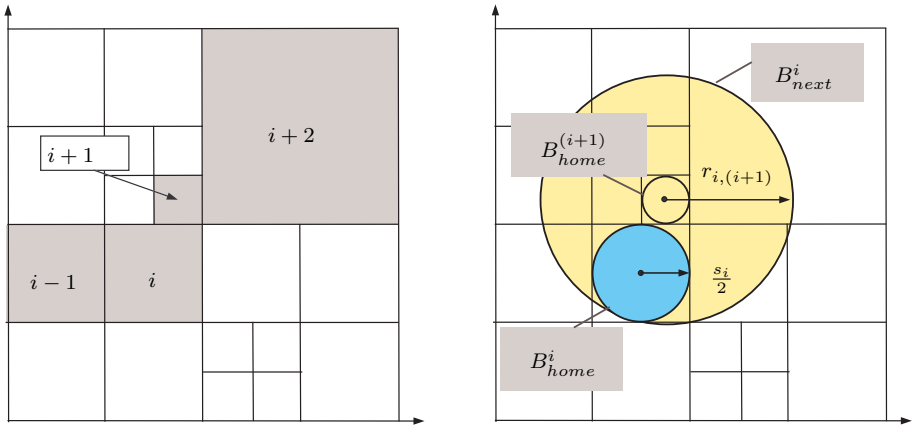

Fig. 2. Solution channel (left); balls B^i_{home} and B^i_{next} (right).

$\mathcal{C}$-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 HF2.

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 HF1.

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

The two first guidances are done using the cell decomposition and the harmonic functions (HF2 for the $\mathcal{C}$-space guidance and HF1 for the channel guidance). The procedure is detailed in Section IV-A. The last one is done using the solution path and is explained in Section IV-B. Both type of guiding procedures are implemented using simple haptic programming primitives:

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.

A. Cell guidance

The cell guidance iteratively uses the point attraction primitive to generate attractive force towards the centers of the cells following the negated gradient of the harmonic functions.

The procedure is the following. 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[2] (Figure 2).

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 $B^{(i+1)}_{next}$.

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.

B. Path guidance

The solution path output by the path planning module is a set of linear segments of $\mathcal{C}$-space. The path guidance uses a simple attractive force towards that set of segments using the segment attraction primitive.

V. VALIDATION

The software has been implemented in C++ using the cross-platform tools Qt (as application framework), Coin3D (as graphics toolkit) and OpenHaptics [14] as haptics library.

As an example a 2D bend corridor has been considered. The task has been defined directly over $\mathcal{C}$-space, i.e. the manipulated object is a point that has to move from right to left through the corridor. This greatly simplifies the collision detection tests and the computation of the virtual reaction forces when the manipulated point tries to interpenetrate the obstacles defining the corridor. Thus, the example focuses on the guiding part of the haptic rendering module.

Figure 3 (top) shows the harmonic functions HF1 and HF2. The solution channel is highlighted in green over HF1. As it can be observed on HF2, the channel cells have a low HF2-value. Figure 3 (bottom) shows the samples over the solution channel and the solution path found using them.

Figure 4 shows the user manipulating the Phantom haptic device (top) and the Stäubli TX-90 robot executing the task (bottom). The haptic guidances are shown between the two photographs (the channel guidance on the left, the path guidance on the middle and the three together on the right).

Extensions to 6 d.o.f. tasks are currently being implemented. The path planning module is already applicable as is (and therefore the guiding haptic rendering too). The virtual part of the haptic rendering module is being modified to compute the virtual reaction forces that appear when the manipulated part interacts with the obstacles in the virtual environment. The procedure followed is based on the use of configuration space [15].

VI. CONCLUSIONS

The combined use of a haptic device both as a teleoperation master device and as a way to interact with virtual worlds allows the set up of an assisted teleoperation system. This paper has introduced the use of a path planning

[2]Note that by construction the ball B^i_{next} always contains B^i_{home}, which guarantees the continuity in the guiding procedure.

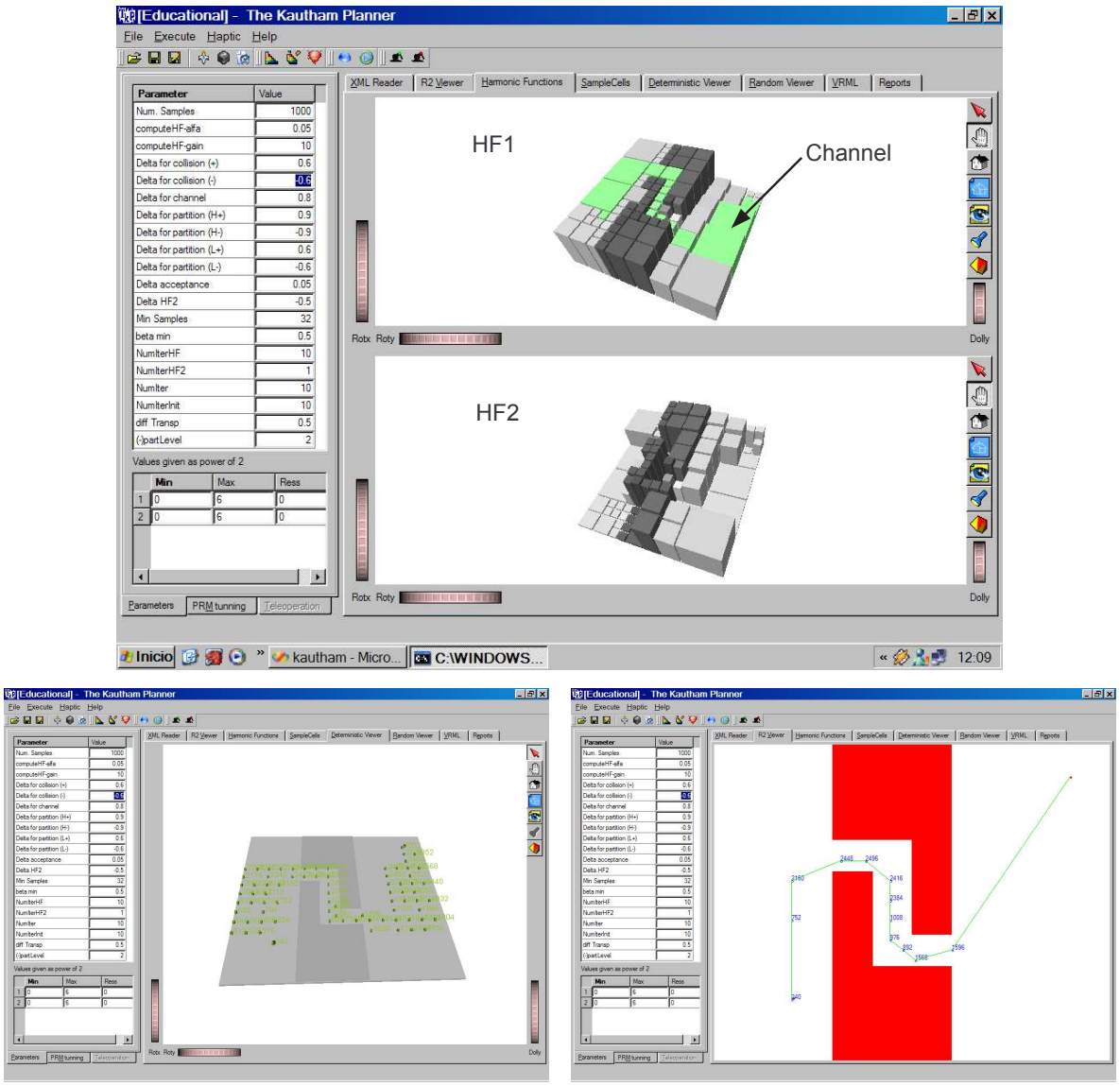

Fig. 3. Top: Harmonic functions HF1 and HF2; Bottom: samples on the solution channel and the path obtained.

technique based on harmonic functions to generate guiding forces that aid a user in the teleoperation of assembly tasks. Two harmonic functions are computed over a hierarchical cell decomposition of $\mathcal{C}$-space: a) one to find a solution channel (sequence of cells) that connects the cell containing the initial configuration to the cell containing the goal configuration; b) the other has all the cells of the solution channel as goal cells and is aimed at guiding the motions towards the channel. Also, a path is computed within the channel connecting the initial and the goal configurations.

From these harmonic functions and from the solution path, haptic guidance is performed at three levels: a) as a force felt on all the cells of $\mathcal{C}$-space that pushes the user towards the solution channel; b) as a force felt on the

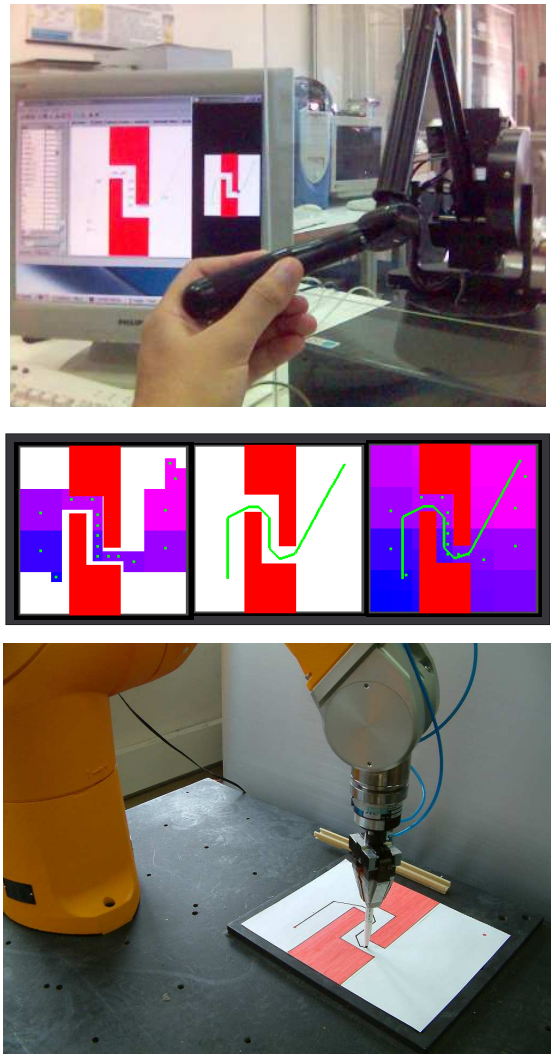

Fig. 4. Channel and path guidance for the teleoperation using the Phantom haptic device teleoperating a Stäubli TX-90 robot.

cells of the channel guiding the user within the channel towards the goal cell; c) a force felt near the path guiding the user towards it.

The approach has been validated with a 2D bend-corridor problem, using a Phantom haptic device. The motions of the haptic device exerted by the user (and directed by the guiding forces) are sent to a Stäubli TX-90 robot using a teleoperation scheme over the internet using IPv6 protocols.

REFERENCES

[1] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Computer and Graphics*, vol. 21, no. 4, pp. 393–404, 1997.

[2] I. Elhajj, N. Xi, W. K. Fung, Y. H. Liu, W. Li, T. Kaga, and T. Fukuda, "Haptic information in internet-based teleoperation," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 295 – 304, 2001.

[3] 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*, vol. 3, 2004, pp. 2513 – 2518.

[4] 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*, 1997, pp. 345–352.

[5] M. Otaduy and M. C. Lin, "A modular haptic rendering algorithm for stable and transparent 6-dof manipulation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 751 – 762, 2006.

[6] 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.

[7] 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*, 2006, pp. 485 – 489.

[8] E. Nuño and L. Basañez, "Force feedback to assist teleoperation systems via high speed networks," in *Proc. of the IFR/DGR 37th Int. Symp. on Robotics*, 2006.

[9] 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*, 1990, pp. 2102–2106.

[10] 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*, 2005, pp. 1815–1820.

[11] 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*, vol. 12, no. 4, pp. 566–580, August 1996.

[12] J. Rosell, C. Vázquez, A. Pérez, and P. Iñiguez, "The Kautham planner: combining deterministic sampling and harmonic functions," IOC-UPC, Tech. Rep., 2006.

[13] J. Rosell, M. Roa, A. Pérez, and F. García, "A general deterministic sequence for sampling d-dimensional configuration spaces," IOC-UPC, Tech. Rep., 2006.

[14] B. Itkowitz, J. Handley, and W. Zhu, "The OpenHaptics toolkit: a library for adding 3D touch navigation and haptics to graphics applications," in *Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005, pp. 590 – 591.

[15] J. Rosell and I. Vázquez, "Haptic rendering of compliant motions using contact tracking in Cspace," in *Proc. of the the IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 4223–4228.

APPENDIX A: CELL CODING

The cells of the hierarchical cell partition are coded as follows. Let:

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

$$V^M = \begin{pmatrix} v_1^M \\ \vdots \\ v_j^M \\ \vdots \\ v_d^M \end{pmatrix} = \begin{pmatrix} a_{M1} & \ldots & a_{i1} & \ldots & a_{11} \\ \vdots & & \vdots & & \vdots \\ a_{Mj} & \ldots & a_{ij} & \ldots & a_{1j} \\ \vdots & & \vdots & & \vdots \\ a_{Md} & \ldots & a_{id} & \ldots & a_{1d} \end{pmatrix} \tag{7}$$

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

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

$$w_{ij} = 2^{(M-j)d+i-1} \text{ for } i \in 1 \ldots d \ \ j \in 1 \ldots M \tag{8}$$

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

$$C^M = V^M \cdot W^M \tag{9}$$

$$V^M = C^M \& W^M \tag{10}$$

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. As an example, the conversion operations of cell code 22 with indices (6,1) on the grid of partition level $M = 3$ (Figure 5a) are:

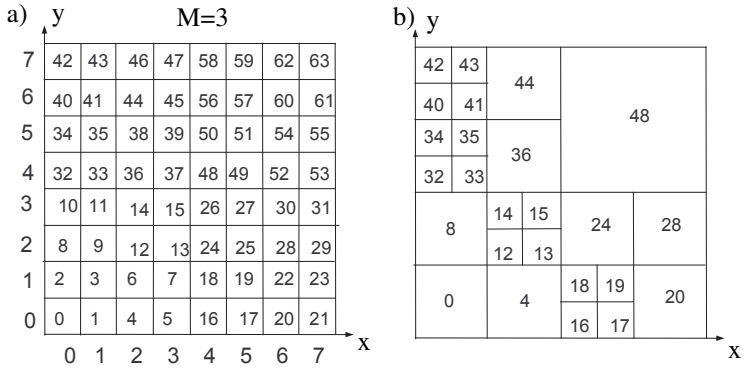

Fig. 5. a) Coding of M-cells; b) Coding of a hierarchical cell decomposition.

$$C^3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 16 & 4 & 1 \\ 32 & 8 & 2 \end{pmatrix} = 22 \tag{11}$$

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

$$= 010110\& \begin{pmatrix} 010000 & 000100 & 000001 \\ 100000 & 001000 & 000010 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

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 5b.

APPENDIX B: NEIGHBORHOOD RELATIONSHIP

The set of neighbors of a cell in a hierarchical cell decomposition is composed of cells of different partition level and its number is obviously not fixed. This set of cells is computed as follow.

Let the m-neighbors of an m-cell be the set of m-cells that neighbor on it. Those m-neighbors are called virtual neighbors since they may not belong to the hierarchical cell partition (Figure 6).

Consider a cell C_a^m and its virtual m-neighbor C_b^m in a given direction and sense. Then, for each direction and sense, a recursive procedure to compute the set of neighbors is proposed:

case 1 - If C_b^m exists in the cell partition then return C_b^m as the unique neighbor of C_a^m in that direction and sense.

case 2 - Otherwise, if C_b^m belongs to a bigger cell C_c^n (i.e. with $n < m$) then return C_c^n as the unique neighbor of C_a^m in that direction and sense.

case 3 - Otherwise Partition(C_b^m): consider each descendant cell C_d^{m+1} of C_b^m and compute its virtual $(m+1)$-neighbor C_e^{m+1} in the same direction and opposite sense. If C_e^{m+1} lies inside C_a^m then:

3.1. If C_d^{m+1} exists in the cell partition then set C_d^{m+1} as a neighbor of C_a^m.

3.2. Otherwise Partition(C_d^{m+1}).

Those computations are coded very efficiently due to the recursive nature of the procedure and due to the use of the compact representation of the hierarchical cell decomposition introduced in Appendix A.

As an example it can be seen in Figure 5b that the 2-cell 24 neighbors on:

- The 2-cell 28 since this is a virtual 2-neighbor that actually exists in the cell partition (case 1).
- The 1-cell 48 since the virtual 2-neighbor 48 lies inside the existing 1-cell 48 (case 2).
- The 3-cell 15 since the virtual 2-neighbor 12 has cell 15 as a descendant 3-cell, and cell 15 has its virtual 3-neighbor 26 inside the 2-cell 24 (case 3).
- The 3-cells 13, 18 and 19 for an analogous reason as the 3-cell 15.

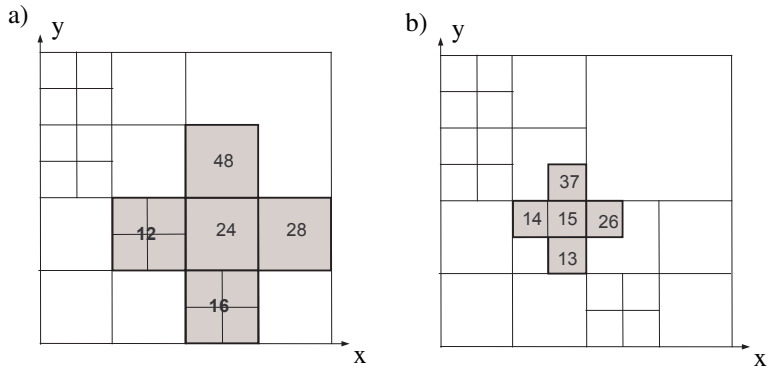

Fig. 6. Examples of virtual neighbors on the hierarchical cell decomposition of Figure 5b: a) The virtual neighbors of the 2-cell 24 are the 2-cells 12, 28, 16 and 48. b) The virtual neighbors of the 3-cell 15 are the 3-cells 14, 26, 13 and 37. Note that the 2-cells 12, 16 and 48 and the 3-cells 26 and 37 do not exist in this hierarchical cell decomposition.