

C-space decomposition using deterministic sampling and distances

Jan Rosell, Carlos Vázquez and Alexander Pérez

Abstract—Hierarchical cell decompositions of Configuration Space can be of great value for enhancing sampling-based path planners, as well as for other robotic tasks with requirements beyond the planning of free paths. This paper proposes an efficient method to obtain a hierarchical cell decomposition of C-space that is based on: a) the use of a deterministic sampling sequence that allows an uniform and incremental exploration of the space, and b) the use of distance measurements to handle as much information as possible from each sample in order to make the procedure more efficient. The proposed cell decomposition procedure is applied to different path planning methods.

I. INTRODUCTION AND OVERVIEW

Motion planning for robotic applications is usually performed in the robot's Configuration Space (C-space), where the robot is mapped to a point and the obstacles in the workspace are enlarged accordingly (C-obstacles). Due to the difficulty of the explicit characterization of C-obstacles, sampling-based methods have ousted other classical methods like those based on cell decompositions.

Nevertheless, cell decomposition methods can still be of interest to enhance sampling-based methods (e.g. [1], [2]), or to deal with tasks with more requirements than those of path planning (e.g. for the generation of a guiding force over the whole workspace to guide the operator in a teleoperated robotic task [3]). Obviously those methods are restrained to low-dimensional C-spaces.

With all these considerations in mind, this paper proposes an efficient method to obtain a hierarchical cell decomposition of C-space. The method is based on the iterative execution of three steps:

- 1) *Sampling*: The exploration of C-space is done using a deterministic sampling sequence (that has a good incremental and uniform coverage feature) and a local randomization procedure.
- 2) *Classification and evaluation*: Samples are classified into cells; then a lazy evaluation philosophy is followed, i.e. a collision check (with distance information) is performed at the sampled configurations only if the cells that contain them have heterogeneous samples.
- 3) *Partition*: A cell is partitioned if, after having collision-checked its samples, it does not yet satisfy a given homogeneity threshold.

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

The authors are with the Institute of Industrial and Control Engineering - Technical University of Catalonia, Barcelona, Spain
jan.rosell@upc.edu

The paper is structured as follows. Section II introduces an efficient coding of the hierarchical cell decomposition of C-space. Sections III, IV and V develop, respectively, the three steps of the procedure. The whole general procedure for obtaining the hierarchical cell decomposition model of C-space is summarized in Section VI, and its use illustrated through several examples in Section VII. Finally, Section VIII concludes the work.

II. HIERARCHICAL CELL DECOMPOSITION MODEL

A hierarchical cell decomposition model of a d -dimensional C-space can be coded as a 2^d -tree in a d -dimensional parameter space, provided a correspondence mapping is defined between spaces. This Section proposes a simple and yet efficient coding of the cells in the 2^d -tree. Let:

- the tree root be a cell with sides of unitary size,
- a cell of a given partition level m be called an m -cell,
- partition levels be enumerated such that the tree root is partition level 0,
- the maximum resolution level (also called sampling level) be partition level M ,
- the depth of the 2^d -tree be determined by a maximum allowable partition level P , with $P \leq M$.

Then, M -cells are labeled with a code computed from its indices in the regular grid of level M . Let the index matrix V^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 an M -cell on the regular grid of partition level M , and W^M be a $d \times M$ matrix of weights, 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 is:

$$C^M = V^M \cdot W^M \quad (1)$$

where the operation $A \cdot B$ represents the scalar product of matrices A and B . Fig. 1a shows for $d = 2$ the regular grid of level $M = 3$ with the cell codes. When $m < M$, m -cells are labeled with the codes of the first M -cell they contain (i.e. with the codes of the descendant M -cell with lowest cell code), as illustrated in Fig. 1b.

III. DETERMINISTIC SAMPLING SEQUENCE

This section summarizes a deterministic procedure, introduced in [4] and based on the cell decomposition model presented in Section II, that obtains a sequence of M -cells that incrementally and uniformly cover the space. The procedure is based on the digital construction method, first proposed

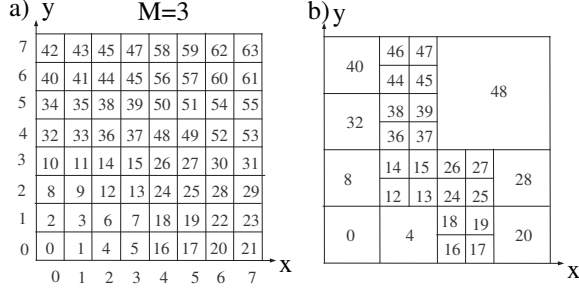


Fig. 1. Coding of: a) M -cells; b) a hierarchical cell decomposition.

in [5]. The Section ends with a procedure to associate a configuration of \mathcal{C} -space to each of the generated M -cells.

A. Sequence of M -cells

Let:

- $k \geq 0$ be the index of the sequence,
- T_d be a binary matrix that defines an ordering of the descendant cells of any given cell, e.g. for $d = 2$ and $d = 3$:

$$T_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad T_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad (2)$$

Reference [4] provides a general method to find T_d for any d .

- V_k^M be the index matrix corresponding to code k .
- W'^M be a $d \times M$ matrix of weights, with $w'_{ij} = 2^{(j-1)d+i-1}$ for $i \in 1 \dots d$ $j \in 1 \dots M$. (Note that matrix W'^M coincides with W^M if the order of its columns is reversed).

Then, the sequence of M -cell codes is:

$$s_d(k) = (T_d V_k^M) \cdot W'^M \quad (3)$$

where the operation AB represents the standard binary matrix multiplication between matrices A and B .

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

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

The first 20 samples generated by $s_2(k)$ are shown in Table I. Following these sequences over Fig. 1a gives a good understanding of how they incrementally and uniformly cover the sampling space.

B. Configurations

A configuration c of \mathcal{C} -space is associated to each M -cell generated using the deterministic sampling sequence introduced. c is called a sample and is obtained by the correspondence mapping function (between \mathcal{C} -space and the parameter space) evaluated at either:

| 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 .

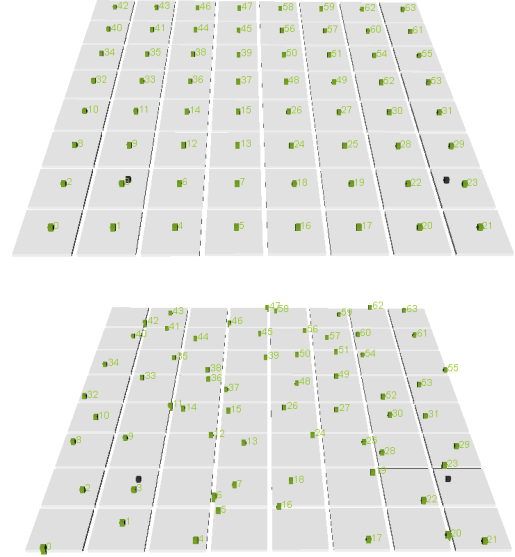


Fig. 2. Configuration generation using the center of M -cells (top) and a random point inside M -cells (bottom).

- the center of the M -cell, or
- a random point inside the M -cell, or
- a random point inside the P -cell that contains the M -cell.

Fig. 2 shows the samples¹ obtained using the first two options. Fig. 3 illustrates the last one showing that, in this latter case, there can be up to $2^{(M-P)d}$ samples in each P -cell.

The use of a deterministic sampling sequence of M -cells allow an incremental and uniform coverage of the whole \mathcal{C} -space useful for the broad exploration that is necessary for obtaining the cell decomposition. The proposed method to associate configurations to those sampled M -cells, enhances the exploration with the following features:

- The (local) randomization allows any configuration to be chosen, widening the chance of capturing the \mathcal{C} -space.
- The selection of $P \neq M$ allow the possibility of constraining the maximum level of partition to a reasonable value in order to not have too many cells, but with the possibility of having them with more than a single configuration.

¹For rigid bodies with two d.o.f. of translation the correspondence mapping is the identity (or a simple scale) and therefore the figures represent both the parameter space and the \mathcal{C} -space.

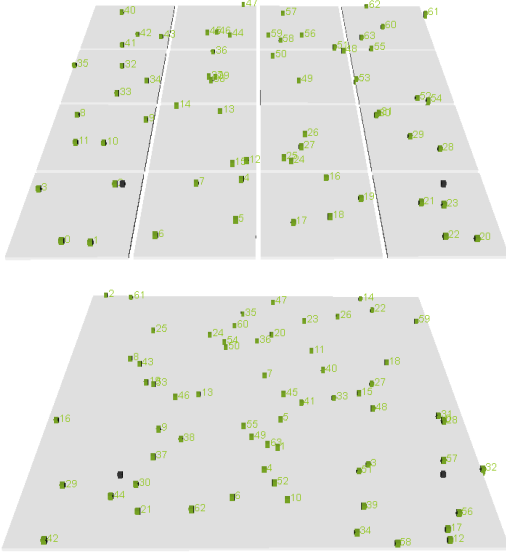


Fig. 3. Configuration generation using a random point inside P -cells when $P \neq M$. Top: $M = 3$ and $P = 2$; Bottom: $M = 3$ and $P = 0$.

IV. SAMPLE CLASSIFICATION AND COLORING

When a sample is generated it must be classified into one of the cells of the hierarchical partition. This is easily done as follows. Let L_c be an ordered list of cell codes, such that $L_c[j] < L_c[j + 1]$. Then, a given sample with code² s_i is classified into a cell with code $L_c[j]$ if:

$$L_c[j] \leq s_i < L_c[j + 1] \quad (5)$$

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

A parameter, called $color_i$, is associated to each sample c_i to store the information about its free or collision nature. Following a lazy evaluation approach, not all the samples will be evaluated. Those not evaluated will have the parameter set to zero. Those evaluated will have the parameter set as follows:

$$color_i = \begin{cases} k + (1 - k) \frac{r}{D} & \text{if } c_i \text{ is free and } r < D \\ 1 & \text{if } c_i \text{ is free and } r \geq D \\ -1 & \text{if } c_i \text{ is not free} \end{cases} \quad (6)$$

being r the distance³ measurement, k a fixed offset and D a distance threshold.

The decision to perform the collision-check test or not depends on the free or collision nature of the samples already contained in the cell where the sample has been classified. This is captured through a parameter, called transparency, defined as follows:

$$T_j = \frac{\sum_{i=1}^{K_j} color_i}{K_j} \quad (7)$$

²The code of a sample is that of the corresponding M -cell.

³The distance considered is the Euclidean translational distance from the robot to the nearest obstacle in the workspace.

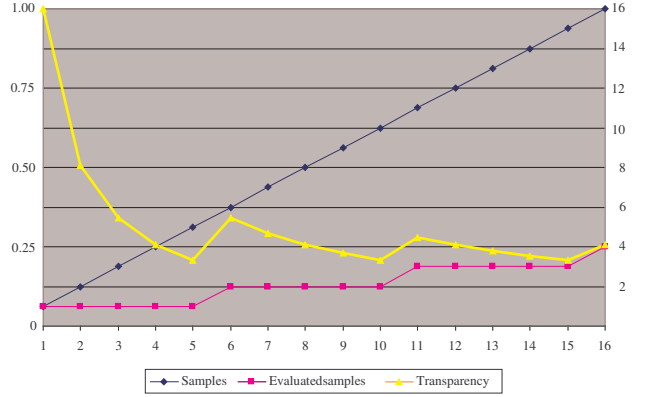


Fig. 4. Evolution of transparency of a cell of free \mathcal{C} -space when the collision threshold is $\Delta_{collision}^+ = 0.2$.

where K_j is the number of samples pertaining to the cell. The transparency satisfies $-1 \leq T_j \leq 1$. It is close to zero if there are either roughly the same number of (evaluated) free and collision samples, or a great proportion of non-evaluated samples. It is close to one of the extremes if they are mainly either free or collision (evaluated) samples. Note that samples not collision-checked make the absolute value of the transparency to decrease. Fig. 4 shows the evolution of the transparency of a cell of the free \mathcal{C} -space: the transparency decreases periodically while non-evaluated samples are added to the cell until a collision-check is performed, making its value to increase.

The transparency of a cell captures its homogeneity. When the transparency is within a given interval around zero the cell is not homogeneous and the collision-check test is required to gather more information, otherwise it is not necessary, i.e. the following collision-check condition is proposed:

$$\Delta_{collision}^- < T_j < \Delta_{collision}^+ \quad (8)$$

being $\Delta_{collision}^-$ and $\Delta_{collision}^+$ fixed thresholds with values ranging between -1 and 0 and between 0 and 1, respectively.

If condition (8) holds and the cell contains samples not collision-checked, then the collision-check is iteratively performed to the configurations of the not-checked samples until the condition does not hold any more, or until the last one is checked.

Fig. 5 illustrates how the number of evaluated samples increases for higher absolute values of the collision thresholds.

V. CELL PARTITION

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. $m < M$, and after having performed the collision-check test. The cell may need to be partitioned if it still is not homogeneous enough. This is evaluated with the following proposed partition condition:

$$\Delta_{partition}^- < T_j < \Delta_{partition}^+ \quad (9)$$

being $\Delta_{partition}^-$ and $\Delta_{partition}^+$ two thresholds ranging between -1 and 0 and between 0 and 1, respectively.

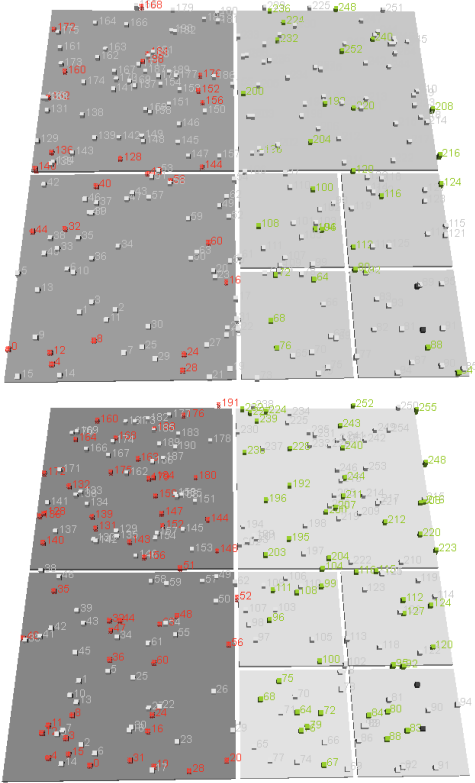


Fig. 5. Top: Collision thresholds $|\Delta_{collision}^-| = |\Delta_{collision}^+| = 0.2$; Bottom: Collision thresholds $|\Delta_{collision}^-| = |\Delta_{collision}^+| = 0.4$. The number of evaluated samples increases when the absolute values of the collision thresholds are higher. Free samples are drawn in green, obstacle samples in red and non-evaluated samples in gray. Cells are drawn in gray tones proportional to the transparency.

If condition (9) holds then the cell is partitioned into its 2^d descendant cells, and the transparency recomputed for each descendant cell.

Fig. 6 illustrates the effect of this condition. In Fig. 6 (bottom) the distance threshold D is set higher than in Fig. 6 (top). For higher values of D the color of the free samples gets lower, as shown in (6), and therefore the transparency of the cell where they pertain too. As a consequence, in this case condition (9) holds more easily and more partitions are performed.

This partition condition can be enhanced if thresholds $\Delta_{partition}^-$ and $\Delta_{partition}^+$ can, each one, take two possible values: a lower (absolute) value when the cell does not contain evaluated samples of different color, and a higher (absolute) value otherwise. This can force not to have cells with both evaluated free and collision samples, as shown in Fig 7 (bottom).

Cells containing the initial and the goal configuration are labeled as initial and goal cells and are recursively partitioned until level P is reached, independently of the partition condition. This can be seen in Fig. 6 where the initial and the goal configurations, drawn as black squares, are located in P-cells (in this case $P = 4$ and $M = 5$).

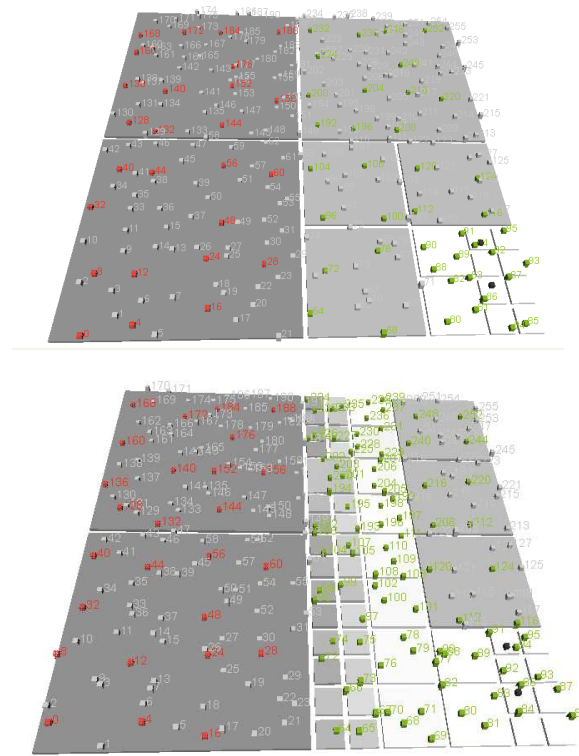


Fig. 6. Partition condition with a given partition threshold applied using different distance thresholds. The effect of distance measurements is stressed with a biggest threshold D (bottom) that results in a lower cell transparency and a higher number of partitions (since the partition condition is more easily hold).

VI. HIERARCHICAL CELL DECOMPOSITION PROCEDURE

The complete partition algorithm is shown in Fig. 8. Its use in a \mathcal{C} -space with a narrow corridor is shown in Fig. 9 with:

- 100 evaluated samples (top) and 250 (bottom)
- $M = 5$ and $P = 4$
- $|\Delta_{collision}^-| = |\Delta_{collision}^+| = 0.2$
- $|\Delta_{partition}^-| = |\Delta_{partition}^+| = \{0.2, 0.9\}$

VII. EXAMPLES

The \mathcal{C} -space partition procedure introduced in this paper can be applied to different classical path planning algorithms, giving them particular and interesting features:

Path planning based on harmonic functions: Harmonic functions are used to compute a potential field without local minima and, therefore, allows to find a channel of cells connecting the initial and the goal ones (Fig. 10).

These harmonic functions are usually computed over the free cells of a regular or hierarchical cell decomposition, being obstacle cells fixed at a high value. The value of the harmonic function at each cell is iteratively computed as the average of the values at its neighbor cells.

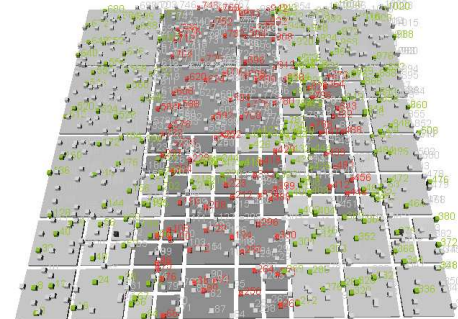
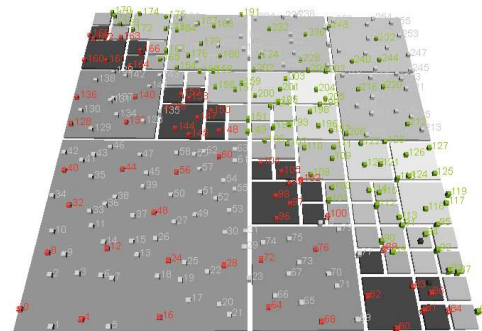
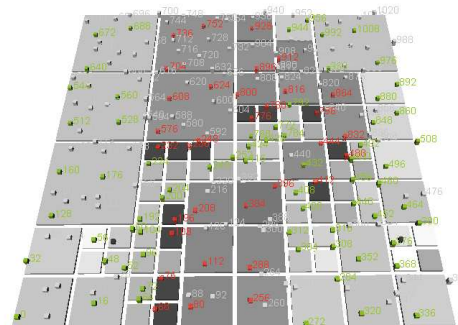
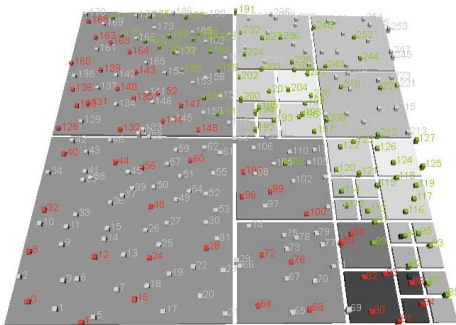


Fig. 7. Top: simple partition condition; Bottom: enhanced partition condition. In the bottom case, the high value of the threshold is set not to have cells with both free and collision samples.

Fig. 9. Partition algorithm applied at a given \mathcal{C} -space using 100 evaluated samples (top) and 250 (bottom).

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

Partition cells until c_{ini} and c_{goal} are located inside a P -cell.

Do:

Get sample from sequence - Equation (3)

Find cell that contains it - Equation (5)

Compute the transparency - Equation (7)

Check collision if condition (8) is satisfied

Partition cell if condition (9) is satisfied

Until K samples have been evaluated

END

Fig. 8. Hierarchical cell partition algorithm.

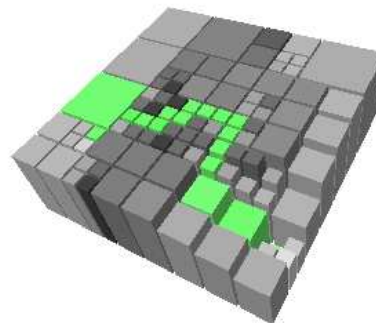


Fig. 10. Harmonic function computed over a hierarchical cell partition of the 2 d.o.f. \mathcal{C} -space of Fig. 9. The solution channel is shown in green.

By using the cell partition introduced here, the computation can be done differently. Now the cells are not classified as being free or obstacle, i.e. in a binary way, but in a continuous manner using the transparency. Then, the harmonic function values can be computed over all the cells of the partition using the transparency as a permeability parameter, i.e. the more transparent a cell is the more it is influenced by its neighbors.

This change makes it possible the use such an interesting planning method in partially explored \mathcal{C} -spaces.

Sampling-based Methods: The samples generated for the partition procedure can be used to create a roadmap to search a path between the initial and the goal configurations. Then, the neighborhood relationship between samples can easily

be determined by the neighborhood of cells, i.e. the local planner will try to connect samples of one cell to samples that lies inside its Manhattan neighbor cells. Also, the weights of the arcs of the roadmap can be set as a function of both its length and its distance to the obstacles, since this information is known for the samples. As an example Fig. 11 shows a roadmap computed using samples generated for obtaining the hierarchical cell decomposition.

Fig. 12 shows a 6 d.o.f. example. Fig. 12 (left) shows the position coordinates of the free and the collision samples (green and red, respectively; non-evaluated samples are not shown) illustrating that, like in Fig. 6, the use of a correct distance threshold allows to have more evaluated samples near the obstacles' surface. Free samples have been connected by a roadmap and used to find the solution path that is shown in Fig. 12 (right). Collision detection has been performed using the PQP library [6].

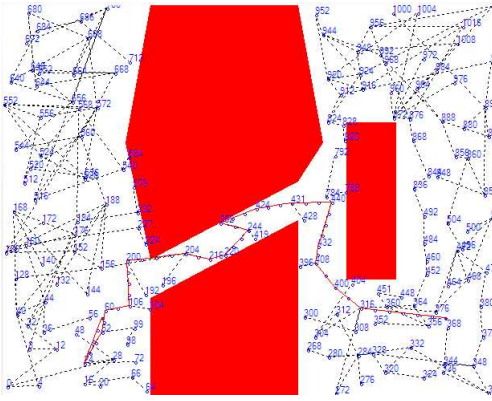


Fig. 11. Roadmap computed using samples generated for obtaining the hierarchical cell decomposition. The solution path is shown in red.

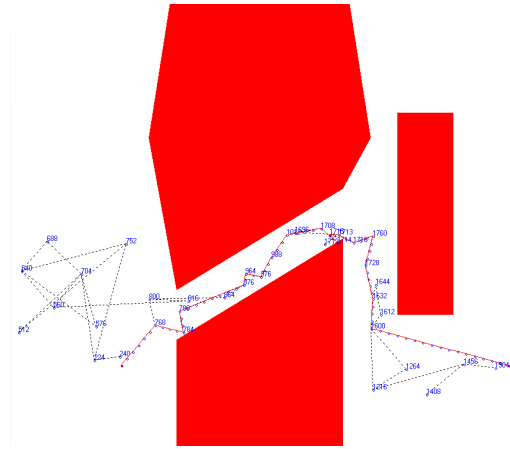


Fig. 13. Kautham planner: local roadmap computed over the channel.

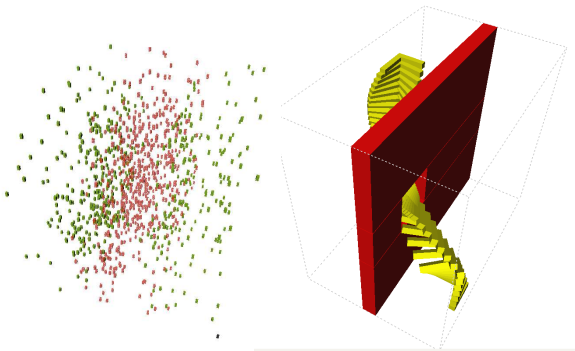


Fig. 12. Path planning problem in 3D: Free and collision samples (left); solution path (right).

A comparative study with using other sampling methods is reported in [4].

Combined Methods: The combination of both the use of harmonic functions and sampling methods can take full potential of the cell partition procedure introduced here. This is what has been proposed in a novel planner, known as Kautham planner, proposed in [7].

The interleaving of both methods allow a better bias of samples towards interesting regions of \mathcal{C} -space, like narrow passages, and therefore a better characterization of the hierarchical cell decomposition using less evaluated samples. The combination also allows to work with a local roadmap composed by the samples contained in the cells of the solution channel, as shown in Fig. 13.

VIII. CONCLUSIONS

This paper has presented an efficient and simple method to obtain a hierarchical cell decomposition model of \mathcal{C} -space. Its main contributions are the followings:

- The combined use of a deterministic sampling of cells and a (local) randomization in the selection of the corresponding configurations, gives the required equilibrium for the efficient exploration of \mathcal{C} -space (i.e. for obtaining the hierarchical cell decomposition using the less number of collision-checked samples as possible).

- The information of the distance between free samples and the obstacles in \mathcal{C} -space is used to characterize the hierarchical cell partition in a more efficient way. Besides this, the distance information can then be used to weight the edges of a roadmap build with the samples used to obtain the partition.
- The hierarchical cell decomposition model generated does not distinguish between free and obstacle cells, but instead considers all cells as equal and characterized by a transparency parameter computed as a function of the number of free and collision samples they contain (also considering distance information). 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. Besides this, transparency can be envisioned as a parameter for revisiting classical planning methods like those based on harmonic functions.

REFERENCES

- [1] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 467–472.
- [2] 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.
- [3] 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*, 2007, pp. 88–93.
- [4] J. Rosell, M. Roa, A. Pérez, and F. García, "A general deterministic sequence for sampling d-dimensional configuration spaces," *Accepted to Journal of Intelligent and Robotic Systems*, 2007.
- [5] 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.
- [6] 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. [Online]. Available: citeseer.ist.psu.edu/gottschalk96obbtree.html
- [7] J. Rosell, C. Vázquez, A. Pérez, and P. Iñiguez, "A novel path planning proposal based on the combination of deterministic sampling and harmonic functions," IOC-DT-P-2007-06, Tech. Rep., 2007.