Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Sampling $SE(3)$ with a Deterministic Sequence for 3D Rigid-Body Path Planning*

Jan Rosell
Institute of Industrial and Control Engineering (IOC)
Technical University of Catalonia (UPC)
Barcelona, SPAIN, Email: jan.rosell@upc.edu

***Abstract*— Sampling-based path planners are giving very good results for problems with high degrees of freedom, being the obtention of samples a crucial factor in their performance. Sampling generation is a difficult issue that entails the challenge of obtaining an incremental and uniform coverage of the configuration space. This objective is even harder when the planning of motions of 3D rigid bodies is tackled, since the corresponding configuration space is $SE(3)$, which requires the careful handling of rotations. This paper proposes a deterministic sampling sequence that generates configurations of $SE(3)$ for the planning of 3D rigid-body motions using sampling-based methods. The proposed sequence is able to generate samples in an incremental low-dispersion manner, giving a good uniform coverage and a lattice structure.**

***Index Terms*— Path planning, sampling-based methods, deterministic sampling.**

I. Introduction

Sampling-based motion planners, like Probabilistic Roadmap Methods (PRM [1]) or those based on the Rapidly-exploring Random Trees (RRT [2]), are giving very good results in robot path planning problems with many degrees of freedom. Its success is mainly due to the fact that they are sampling-based, i.e. the explicit characterization of $\mathcal{C}$-obstacles is not required and only sample configurations of $\mathcal{C}$-space are checked for collision. Therefore, sampling efficiency is a crucial point for the good performance of those planners. Two research lines have been followed to improve sampling efficiency. On the one hand, there are the methods that use random sampling with a sample distribution tailored using task-specific knowledge in order to bias the samples towards critical regions (e.g. [3], [4], [5] [6] [7]). On the other hand there are the methods based on deterministic sampling sequences, that have interesting properties for path planning like a lattice structure (that allows to easily determine the neighborhood relations) and a good uniform coverage of the space. Deterministic sampling sequences applied to PRM-like planners are demonstrated by LaValle et al. [8] to achieve the best asymptotic convergence rate and experimental results showed that they outperformed random sampling in nearly all motion planning problems. These authors also presented an exhaustive and interesting discussion on the different aspects that intervene in sampling-based methods [9].

The use of sampling-based methods for the planning of paths for a 3D rigid-body that can both translate and rotate requires the generation of samples over the Special Euclidean Group in three dimensions $SE(3)$. This group is the cartesian product of the Euclidean space $\Re^3$ with the Special Orthogonal Group $SO(3)$ of 3×3 orthonormal matrices that represent, respectively, arbitrary translations and rotations in 3 dimensions [10]. Sampling methods are usually developed for sampling configurations over a unit d-dimensional cube and, therefore, if rotations are not carefully handled, their parameterization may induce these sampling methods to give a non-uniform coverage of the space. This problem has been tackled by Kuffner [11] that discusses random sampling and interpolation algorithms for $SO(3)$, and by Yershova and LaValle [12] that propose a deterministic sampling sequence for $SO(3)$ based on a deterministic sampling sequence over the surface of a cube.

In this paper a deterministic sampling sequence for the obtention of samples on $SE(3)$ is proposed. It is based on the incremental low-dispersion sampling of a multigrid representation of a six-dimensional unit cube of parameters, and the proper mapping to position and orientation coordinates. The orientation mapping is based on a hierarchical triangular decomposition of the surface of a tetrahedron inscribed in the unit sphere. The proposed sampling sequence fulfils the requirements to provide a uniform coverage of the space (with an increasing quality proportional to the number of samples). The sequence is simple and efficient, and provides a unified sampling of positions and orientations of a 3D rigid body.

The paper is structured as follows. The first part is composed by Section II, that presents a multigrid space decomposition with a code convention to label and locate the cells, and by Sections III, IV and V that use it in the decomposition of $\Re^3$, $SO(3)$ and $SE(3)$, respectively. The second part is composed by Section VI, that introduces the proposed deterministic sampling sequence, and Sections VII, VIII and IX that make use of it to sample the spaces $\Re^3$, $SO(3)$ and $SE(3)$, respectively. Finally Section X concludes the work.

II. Multigrid Space Decomposition

Regular spatial structures are usually used to obtain an approximate cell decomposition of the space, since they have good properties for planning purposes such as implicit

*This work was partially supported by the CICYT projects DPI2002-03540 and DPI2004-03104

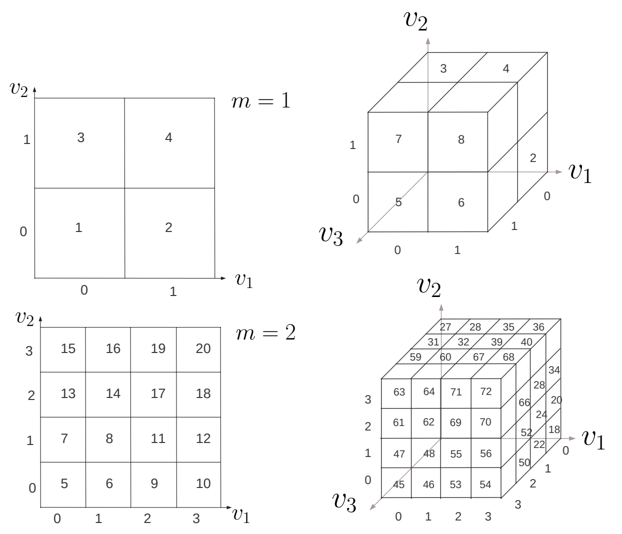


Fig. 1. Cell codes for different levels in the hierarchy in 2D and 3D $\mathcal{C}$-spaces.

neighborhood information needed in roadmap based methods, or the possibility to be used as computational grids for potential-field based methods. If different resolution levels are required, multiresolution grids can be defined such that all the cells of the grid of level m are contained in the grid of level $m+1$ (with level enumeration increasing with the resolution). Local resolution requirements can be managed by using hierarchical 2^d-tree structures (i.e. quadtrees and octrees for spaces of dimension $d=2$ and $d=3$, respectively).

Consider the space defined by the d-dimensional unit cube $[0,1]^d \subset \Re^d$. The following multiresolution grid, $\mathcal{G}^d$, is defined with an associated code convention that univocally labels and locates the cells of each level in the grid hierarchy. The levels in the grid hierarchy are called partition levels and are enumerated such that the minimum resolution grid, $\mathcal{G}_0^d$, has partition level 0 (with a unique cell covering the entire space), and the maximum resolution grid, $\mathcal{G}_M^d$, has partition level M. A cell of $\mathcal{G}_m^d$ is called an m-cell, and is denoted as b^m. The m-cells have sides of size $s_m = 1/2^m$.

The proposed cell labelling univocally relates cell code with the indices of the cell in the corresponding grid. With this code convention, any subset of cells can be managed as a list of codes, in a similar way as the *linear quadtrees* proposed by Gargantini [13] for $d=2$. Cell codes are non-negative integers that, for a given partition level m, range from C_{ini}^m to C_{end}^m, with:

$$C_{ini}^m = \frac{2^{dm}-1}{2^d-1} \quad (1)$$

$$C_{end}^m = 2^d C_{ini} \quad (2)$$

Since $C_{ini}^m = C_{end}^{(m-1)} + 1$, the proposed code convention uses all non-negative integers. Fig. 1 shows the codes used for the cells of different partition levels for 2D and 3D spaces.

For an m-cell, b_k^m, let:

- $(v_1^m, \ldots, v_d^m)$ be its indices on $\mathcal{G}_m^d$, with each component expressed in base 2 as $v_j^m = (a_j^m \ldots a_j^1)_2$.
- The conversion table $T(b_k^m)$ be the binary $m \times d$ matrix whose columns are the binary representation of $v_j^m \ \ \forall j \in 1 \ldots d$:

Code	v_d^m	$\ldots$	v_j^m	$\ldots$	v_1^m
r_m	a_d^m	$\ldots$	a_j^m	$\ldots$	a_1^m
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
r_i	a_d^i	$\ldots$	a_j^i	$\ldots$	a_1^i
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
r_1	a_d^1	$\ldots$	a_j^1	$\ldots$	a_1^1

Then, the code k of the cell is computed using the following expression:

$$k = C_{ini}^m + \sum_{i=1}^{m} r_i 2^{d(i-1)} \quad (3)$$

where the coefficients r_i are retrieved from the rows of $T(b_k^m)$, i.e. $r_i = (a_d^i \ldots a_1^i)_2 \ \ \forall i \in 1 \ldots m$.

Note that from (3), it can be seen that the values $r_i \ \ \forall i \in 1 \ldots m$ are the digits of the representation in base 2^d of $(k - C_{ini}^m)$. Taking into account this fact, given the code k of a cell b_k^m, the corresponding indices on the grid $\mathcal{G}_m$ can be retrieved from the columns of the conversion table built with the binary representation of the digits of $(k - C_{ini}^m)_{2^d}$ as rows.

The level of a cell with code k can be obtained as follows:

$$m = (\text{int}) \left\{ \frac{1}{d} \log_2[(2^d - 1)k + 1] \right\} \quad (4)$$

III. Mapping to $\Re^3$

The position of a 3D rigid body is defined by the (x,y,z) coordinates of the origin of its reference frame with respect to a reference frame fixed at the workspace. Taking into account the maximum displacements in each direction, those coordinates can be scaled to the unit cube $[0,1]^3 \subset \Re^3$.

If the unit cube is represented by the multigrid $\mathcal{G}^3$ then, considering each level as a Sukharev grid [8], each cell is mapped to the position coordinates of its center; i.e. the indices (v_1^m, v_2^m, v_3^m) of an m-cell on the grid $\mathcal{G}_m^3$ are used to compute the position coordinates (x_1^m, x_2^m, x_3^m) as follows:

$$x_j^m = v_j^m s_m + \frac{s_m}{2} \qquad \forall j \in 1 \ldots 3 \quad (5)$$

For instance, the cells with codes 2 and 72 have, respectively, the indices $(1,0,0)$ on grid $\mathcal{G}_1^3$ and $(3,3,3)$ on grid $\mathcal{G}_2^3$. Since $s_1 = 0.5$ and $s_2 = 0.25$, then the resulting coordinates are $(0.75, 0.25, 0.25)$ and $(0.875, 0.875, 0.875)$.

IV. Mapping to SO(3)

There are several conventions to parameterize rotations in $SO(3)$, like rotation matrices, Euler angles and quaternions. Their characteristics and tradeoffs are commented by Kuffner [11] and Shoemake [14], both concluding that quaternions are the best alternative to represent rotations, since: a) they provide efficient and effective algorithms to obtain random orientation samples, and b) they allow the best way to interpolate orientations.

Quaternions parameterize rotations by defining angle-axis pairs, i.e. a rotation is defined by an angle $\theta \in [0 \ldots \pi]$ to be rotated around a given axis defined by an unitary vector $w = (w_x, w_y, w_z)$, called rotation vector.

The following mapping between the parameters $v_1 \in [0, 1]$, $v_2 \in [0, 1]$ and $v_3 \in [0, 1]$ of the multigrid $\mathcal{G}^3$ and the rotation vector and angle is proposed. The rotation angle is obtained from v_3 as $\theta = v_3\pi$. For the rotation vector, v_1 and v_2 are used to parameterize a hierarchical triangular decomposition of the surface of a tetrahedron. Then, an iterative procedure is used to map a given triangle to a point on the surface of the unit sphere that represents the corresponding (unitary) rotation vector.

A. Rotation vector mapping

The procedure that relates (v_1, v_2) with the rotation vector $w = (w_x, w_y, w_z)$ is based on the following hints:

1) Considering only the variables v_1 and v_2, the indices of an m-cell over the grid $\mathcal{G}^2_m$ are (v_1^m, v_2^m). Their binary representation is $((a_1^m \ldots a_1^1)_2, (a_2^m \ldots a_2^1)_2)$.
2) Each rotation vector w can be defined as the unitary vector in the direction of the linear combination of three unitary vectors W_0, W_1 and W_2, called its generatrix vectors, i.e.:
$$w = \sum_{k=0}^{2} W_k / |\sum_{k=0}^{2} W_k| \tag{6}$$
3) The generatrix vectors W_0, W_1 and W_2 are computed with an iterative procedure based on a hierarchical triangular decomposition of a the surface of a tetrahedron with vertices on the unit sphere. The faces of the tetrahedron are numbered from 0 to 3. The triangles in the hierarchy are located using $((a_1^m \ldots a_1^1)_2, (a_2^m \ldots a_2^1)_2)$.
4) The initial generatrix vectors coincide with the three vertices of the tetrahedron face that is selected by the value $(a_2^m a_1^m)_2$. They are called W_0^m, W_1^m and W_2^m.
5) At each step $i = m \ldots 2$ of the iterative procedure, a triangle of a lower hierarchical level is selected depending on the values of a_1^{i-1} and a_2^{i-1}. The vertices of the selected triangle are used to update the generatrix vectors values to W_0^{i-1}, W_1^{i-1} and W_2^{i-1}.
6) The last generatrix vectors values, obtained at step $i = 2$, are W_0^1, W_1^1 and W_2^1. They are used to determine w as expressed in (6).

The following subsections detail the initialization and update of the generatrix vectors and show a formal recursive expression for the computation of w.

B. Initial generatrix vectors

The tetrahedron used as a base to define the first generatrix vectors is arbitrarily chosen with the following vertices defined over the unit sphere:

$$\begin{aligned} P_0 &= \{\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3\} \\ P_1 &= \{\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3\} \\ P_2 &= \{-\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3\} \\ P_3 &= \{-\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3\} \end{aligned} \tag{7}$$

This vertices are grouped into a 4×3 matrix, V, whose rows contain the vertices of each face:

$$V = \begin{bmatrix} P_0 & P_2 & P_1 \\ P_0 & P_1 & P_3 \\ P_0 & P_3 & P_2 \\ P_3 & P_2 & P_1 \end{bmatrix} \tag{8}$$

The indices of V are considered starting from zero, e.g. $V[3][0] = P_3$ and $V[0] = \{P_0, P_2, P_1\}$.

The initial generatrix vectors coincide with the three vertices of the face of the tetrahedron that is selected by the value $(a_2^m a_1^m)_2 = 2a_2^m + a_1^m \in 0, \ldots, 3$:

$$\begin{aligned} W_0^m &= V[(a_2^m a_1^m)_2][0] \\ W_1^m &= V[(a_2^m a_1^m)_2][1] \\ W_2^m &= V[(a_2^m a_1^m)_2][2] \end{aligned} \tag{9}$$

C. Generatrix vectors update

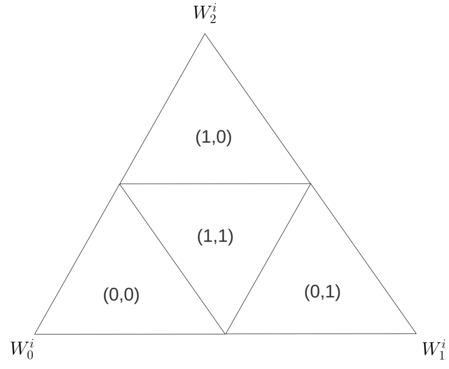


Fig. 2. Partition of a triangle into subtriangles

Consider an equilateral triangle with vertices at W_0^i, W_1^i and W_2^i such that W_0^i is selected as the reference origin and the reference axes are defined by $(W_1^i - W_0^i)$ and $(W_2^i - W_0^i)$. This triangle can be partitioned into four equal triangles as shown in Fig. 2. The indices of these subtriangles are (a_1^{i-1}, a_2^{i-1}) and determine the relative position with respect to the parent triangle. Then, defining δ_1^{i-1} and δ_2^{i-1} as:

$$\begin{aligned} \delta_1^{i-1} &= (W_1^i - W_0^i)/2 \\ \delta_2^{i-1} &= (W_2^i - W_0^i)/2 \end{aligned} \tag{10}$$

and the function sign(a_1, a_2) as:

$$\text{sign}(a_1, a_2) = \begin{cases} -1 \text{ if } a_1 = a_2 = 1 \\ +1 \text{ otherwise} \end{cases} \tag{11}$$

the vertices of the subtriangles are found as:

$$\begin{aligned} t_0^{i-1} &= W_0^i + \delta_1^{i-1} a_1^{i-1} + \delta_2^{i-1} a_2^{i-1} \\ t_1^{i-1} &= t_0^{i-1} + \delta_1^{i-1} \text{sign}(a_1^{i-1}, a_2^{i-1}) \\ t_2^{i-1} &= t_0^{i-1} + \delta_2^{i-1} \text{sign}(a_1^{i-1}, a_2^{i-1}) \end{aligned} \quad (12)$$

This decomposition leads to a hierarchical subdivision of triangles similar to the hierarchical subdivision of squares done by the multigrid $\mathcal{G}^2$, the relation between cell labelling and indices being the same.

In Fig. 3 (left) the four faces of the tetrahedron, labelled from f_0 to f_3, are drawn coplanar. Each one is partitioned into four subtriangles. The coordinates of each subtriangle are $(a_1^2 a_1^1, a_2^2 a_2^1)$. The most significant bits a_1^2 and a_2^2 are used to locate the face as detailed in the previous subsection, e.g. triangle 11 is of face f_1 with vertices $\{P_0, P_1, P_3\}$ since its indices are $(a_1^2 a_1^1, a_2^2 a_2^1) = (10, 01)$ and therefore $V[(a_2^2 a_1^2)_2] = V[(01)_2] = \{P_0, P_1, P_3\}$. The least significant bits, a_1^1 and a_2^1, locate the subtriangles within each face. Fig. 3 (right) shows a further subdivision of f_0 with the corresponding indices of the new subtriangles defined by three bits.

Generatrix vectors are found by iteratively updating their values. The procedure is done in $m-1$ steps. At each step i with $i = m \dots 2$ a triangle of a lower hierarchical level is selected, depending on the values of bits a_1^{i-1} and a_2^{i-1}. Its vertices are determined by (12). Then, the generatrix vectors W_0^{i-1}, W_1^{i-1} and W_2^{i-1} are the unitary vectors in the directions of t_0^{i-1}, t_1^{i-1} and t_2^{i-1}:

$$\begin{aligned} W_0^{i-1} &= t_0^{i-1}/|t_0^{i-1}| \\ W_1^{i-1} &= t_1^{i-1}/|t_1^{i-1}| \\ W_2^{i-1} &= t_2^{i-1}/|t_2^{i-1}| \end{aligned} \quad (13)$$

As an example Fig. 4a shows face f_0 that is decomposed in Fig. 4b into triangles 5 to 8. Fig. 4c illustrates how the new generatrix vectors are obtained from the vertices of triangle 8.

D. Recursive procedure

Let $w^i(W_0^i, W_1^i, W_2^i)$ be the computation of w at step i from the unitary vectors W_0^i, W_1^i and W_2^i. Then, the following recursive expression summarizes the procedure to compute the coordinates of w:

$$w^i(W_0^i, W_1^i, W_2^i) = \begin{cases} w^{i-1}(W_0^{i-1}, W_1^{i-1}, W_2^{i-1}) & \text{if } i = m \dots 2 \\ \sum_{k=0}^{2} W_k^1 / |\sum_{k=0}^{2} W_k^1| & \text{otherwise} \end{cases} \quad (14)$$

The initial values W_0^m, W_1^m and W_2^m are set using (9), and are updated using (13).

Fig. 5 shows the four rotation vectors illustrated as points over the unit sphere obtained from the four triangles of face f_0 of the tetrahedron.

V. Mapping to $SE(3)$

The multigrid $\mathcal{G}^6$ is used to represent configurations of $SE(3)$. The three first variables (v_1, v_2, v_3) are used to parameterize rotations as detailed in Section IV, i.e. v_1 and v_2 are used to fix the direction of the rotation axis and v_3 to fix the rotation angle. The three last variables (v_3, v_4, v_5) are used to parameterize the (x, y, z) translations as detailed in Section III.

VI. Deterministic Sampling sequence

This section proposes a deterministic sequence to sample the cells of a multigrid $\mathcal{G}^d$ in such a way that the cell centers uniformly and incrementally cover the space.

On a d-dimensional space each cell of a grid $\mathcal{G}_m^d$ can be partitioned into 2^d cells on the grid of the next partition level, $\mathcal{G}_{m+1}^d$. The deterministic sampling sequence proposed, called $s_d(k)$, is based on the recursive application of a low-dispersion ordering, called L_d, of the 2^d descendant cells of a given parent cell.

Let each descendant cell be described by a binary word with d bits, one for each axis, that locates it with respect to its parent cell (e.g. cell 2 in the 3D space of Fig. 1 is represented by 001). Then, a low-dispersion ordering of these descendant cells can be found as the sequence of 2^d binary words such that each element of the sequence maximizes the distance (measured as the number of bits that differ) to the previous elements of the sequence [15].

For a three dimensional space $L_3 = \{000, 111, 010, 101, 001, 110, 011, 100\}$, and for a six dimensional space $L_6 =$ 000000, 111111, 010111, 101000, 001111, 110000, 011000, 100111, 000101, 111010, 010010, 101101, 001010, 110101, 011101, 100010, 000011, 111100, 010100, 101011, 001100, 110011, 011011, 100100, 000110, 111001, 010001, 101110, 001001, 110110, 011110, 100001, 000001, 111110, 010110, 101001, 001110, 110001, 011001, 100110, 000100, 111011, 010011, 101100, 001011, 110100, 011100, 100011, 000010, 111101, 010101, 101010, 001101, 110010, 011010, 100101, 000111, 111000, 010000, 101111, 001000, 110111, 011111, 100000$\}$

These orderings are not unique. Alternative orderings satisfying the same feature are also possible.

Let k be the index of the sequence, r_i be the digits of $(k - C_{ini}^m)$ in base 2^d as expressed in (3), and m be the hierarchical level associated to k as expressed in (4). Then:

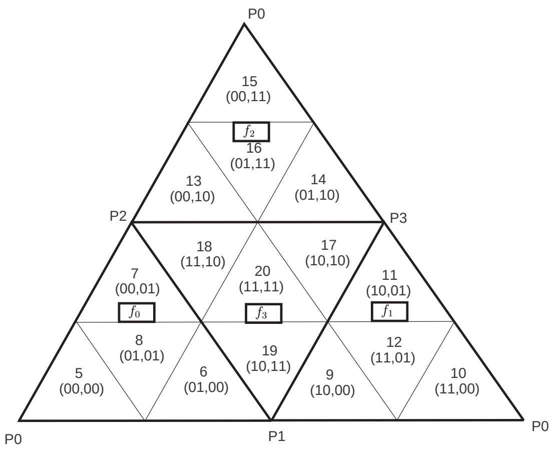
$$s_d(k) = C_{ini}^m + \sum_{i=1}^{m} L_d(r_i) 2^{d(m-i-1)} \qquad k \geq 0 \quad (15)$$

As an example, the first cell samples generated over a 2D space are:

$$\begin{aligned} s_2(k) = \{&0, 1, 4, 3, 2, 5, 17, 13, 9, 8, 20, 16, 12, \\ &7, 19, 15, 11, 6, 18, 14, 10, 21, 69, 53, \dots\} \end{aligned} \quad (16)$$

Following this sequence on Fig. 1 gives a good intuition of how $s_d(k)$ works.

If samples from lower partition levels are not to be generated, the sequence $s_d(k)$ can be started from a greater index k. For instance, if samples from partition levels 0 to P are not desired, the sequence is started from $k = C_{ini}^{P+1}$. As an example, the cell samples generated over a 3D space



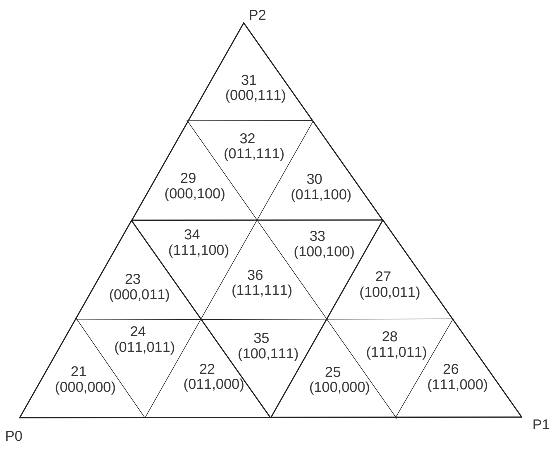


Fig. 3. Labelling of the hierarchical triangular decomposition of a tetrahedron (left), and labelling of the subtriangles of face f_0 (right).

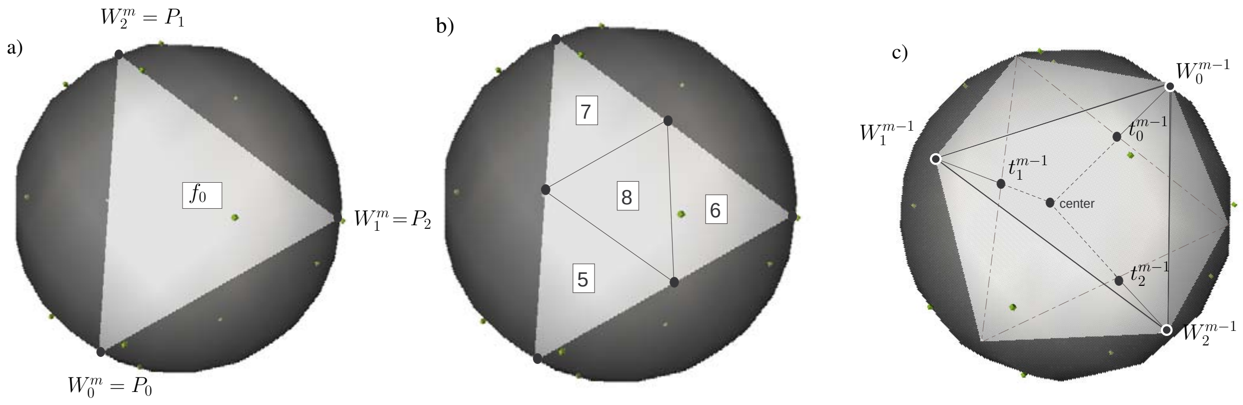


Fig. 4. Illustration of the procedure to update the generatrix vectors.

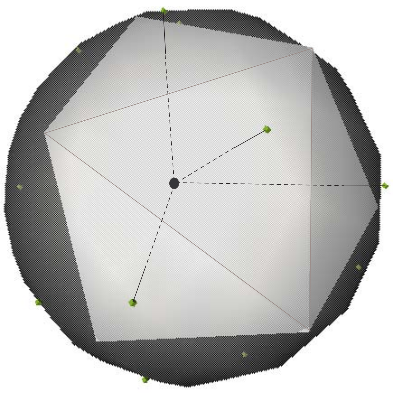


Fig. 5. Samples over the unit sphere computed from the four triangles of face f_0 of the tetrahedron.

staring from level 2 are generated by $s_3(k)$ with $k \geq 9$:

$$s_3(k) = \{9, 65, 25, 49, 41, 33, 57, 17, 16, 72, 32, 56, 48, 40, 64, 24, 11, 67, \dots\} \quad (17)$$

VII. Sampling $\Re^3$

Sampling of $\Re^3$ is performed using the sampling sequence $s_3(k)$ and the mapping to position coordinates expressed in Section III (Fig. 6 (left) shows the first eight samples of the sequence).

For a given path planning problem, the resolution required determines the maximum partition level (M) to be used, i.e. it is supposed that paths can be found with a resolution of $1/2^M$. Then, samples are generated over the finest grid, $\mathcal{G}_M^d$, staring the sequence $s_3(k)$ at $k = C_{ini}^M$.

As an example assume that the resolution required is satisfied with $M = 3$. In this case Fig. 6 (middle) shows the first 80 samples obtained by $s_3(k)$ starting at $k = C_{ini}^3 = 73$. Fig. 6 (right) shows the whole set of 512 samples of $\mathcal{G}^3$. It can be seen that the procedure incrementally and uniformly covers $\Re^3$ in a low-dispersion manner.

VIII. Sampling $SO(3)$

Sampling of $SO(3)$ is performed using the sampling sequence $s_3(k)$ and the mapping to orientation coordinates expressed in Section IV.

As done for the sampling of $\Re^3$, once the maximum partition level (M) is determined by the resolution requirements, samples are generated over the finest grid, $\mathcal{G}_M^d$, staring the sequence $s_3(k)$ at $k = C_{ini}^M$.

As an example assume that the resolution required is satisfied with $M = 5$. In this case there is a maximum of 1024 rotation axis and 32 angles to be rotated around them. Fig. 7 (top) shows the first 300 rotation axis directions drawn as points over the unit sphere, and Fig. 7 (bottom) shows the whole set of 1024 possible rotation axis direc-

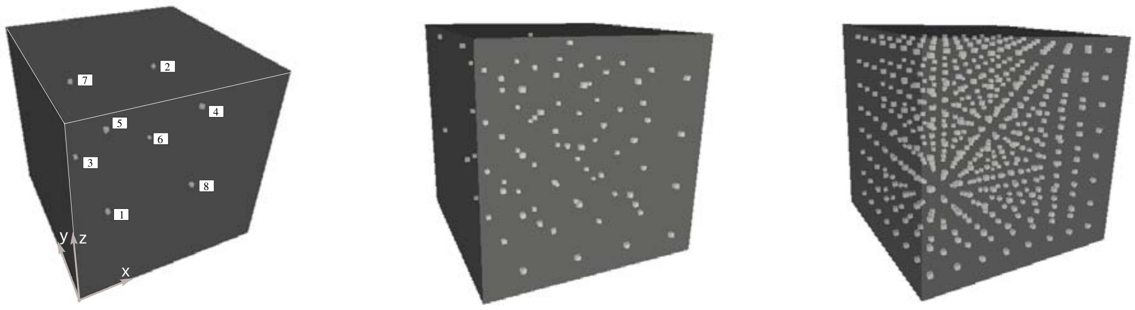


Fig. 6. Sequence of 8, 80 and 512 samples distributed on the unit cube.

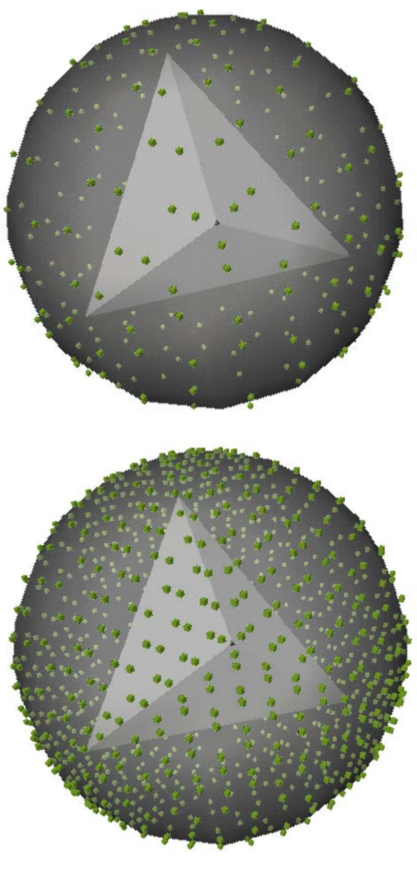


Fig. 7. Sequences of 300 (top) and 1024 (bottom) samples distributed over the unit sphere indicating different rotation axis directions.

tions. It can be seen that the procedure incrementally and uniformly covers $SO(3)$ in a low-dispersion manner.

IX. Sampling $SE(3)$

Sampling of $SE(3)$ is performed using the sampling sequence $s_6(k)$. The indices (v_1, v_2, v_3) of the cell codes obtained by $s_6(k)$ are used to compute the orientation coordinates (using the expressions in Section IV). The indices (v_4, v_5, v_6) are used to compute the position coordinates (using the expressions in Section III).

X. Conclusions

The obtention of samples for sampling-based motion planners have the requirements to: $a)$ provide a uniform coverage of the space, $b)$ improve the uniform coverage as the number of samples increases, $c)$ have a lattice structure that reduces the cost of computing neighborhood relationships. If the planning of 3D rigid-body motions is required, the uniform coverage of the rotation space must be carefully handled. This paper proposed a deterministic sampling sequence for the obtention of samples on $SE(3)$ that fulfils those requirements. The sequence is simple and yet efficient, satisfactorily generating deterministic samples over $SE(3)$ in an incremental low-dispersion manner.

References

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

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

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

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

[5] G. Sánchez and J.-C. Latombe, "On delaying collision checking in PRM planning: application to multi-robot coordination," *The Int. J. Robotics Research*, vol. 21, pp. 5–26, Jan. 2002.

[6] D. Aarno, D. Kragic, and H. I. Christiansen, "Artificial potential biased probabilistic roadmap method," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 461–466, 2004.

[7] M. Kazemi and M. Mehrandezh, "Robotic navigation using harmonic function-based probabilistic roadmaps," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 4763–4770, 2004.

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

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

[10] J.-C. Latombe, *Robot Motion Planning*, ch. Configuration Space of a Rigid Object, pp. 58–104. Kluwer Academic Publishers, 1991.

[11] J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3993–3998, 2004.

[12] A. Yershova and S. LaValle, "Deterministic sampling methods for spheres and SO(3)," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3974–3980, 2004.

[13] I. Gargantini, "An effective way to represent quadtrees," *Communications of the ACM*, vol. 25, no. 12, pp. 905–910, 1982.

[14] K. Shoemake, "Animating rotation with quaternion curves," *Proc. of the 12th ACM Conf. on Computer Graphics and Interactive Techniques*, vol. 19, no. 3, pp. 245–254, 1985.

[15] J. Rosell and M. Heisse, "An efficient deterministic sequence for sampling-based motion planners." unpublished. Submitted to the IEEE Int. Symp. on Assembly and Task Planning - July 2005.