Sampling with Halton Points on n-Sphere

Wai-Shing Luk[1]

[1]School of Microelectronics
Fudan University

April 6, 2014

Agenda

Agenda

Abstract

- Sampling on n-sphere (S^n) has a wide range of applications, such as:
  - Spherical coding in MIMO wireless communication
  - Multivariate empirical mode decomposition
  - Filter bank design
- We propose a simple yet effective method which:
  - Utilizes low-discrepancy sequence
  - Contains only 10 lines of MATLAB code in our implementation!
  - Allow incremental generation.
- Numerical results show that the proposed method outperforms the randomly generated sequences and other proposed methods.

Agenda

Problem Formulation

Desirable properties of samples over S^n

- Uniform
- Deterministic
- Incremental
  - The uniformity measures are optimized with every new point.
  - Reason: in some applications, it is unknown how many points are needed to solve the problem in advance

Motivation

- The topic has been well studied for sphere in 3D, i.e. $n = 2$
- Yet it is still unknown how to generate for $n > 2$.
- Potential applications (for $n > 2$):
  - Robotic Motion Planning (S^3 and SO(3)) [YJLM10]
  - Spherical coding in MIMO wireless communication [UL06]:
    - Cookbook for Unitary matrices
    - A code word = a point in S^n
  - Multivariate empirical mode decomposition [RM10]
  - Filter bank design [M$^+$11]

Halton Sequence on S^n

- Halton sequence on S^2 has been well studied [CF97] by using cylindrical coordinates.
- Yet it is still little known for S^n where $n > 2$.
- Note: The generalization of cylindrical coordinates does NOT work in higher dimensions.

Agenda

Basic: Van der Corput sequence

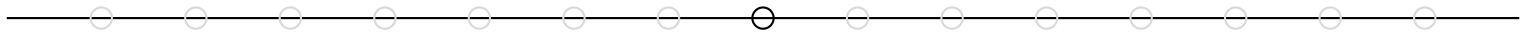
- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

Example

Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathtt{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

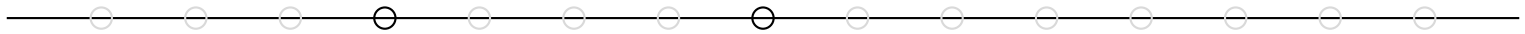
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at `http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence`

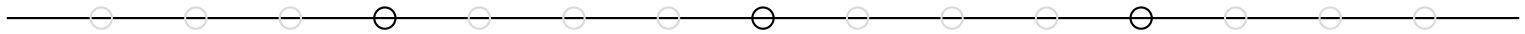
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

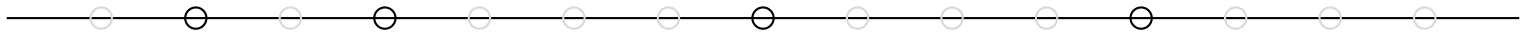
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at `http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence`

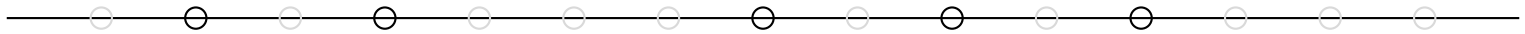
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at `http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence`

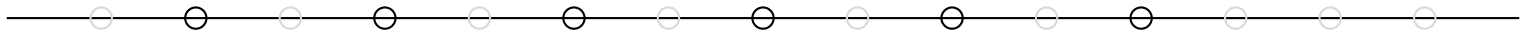
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathtt{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

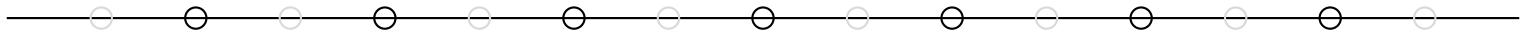
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\text{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

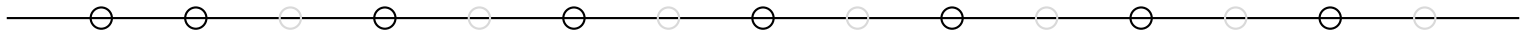
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathtt{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at `http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence`

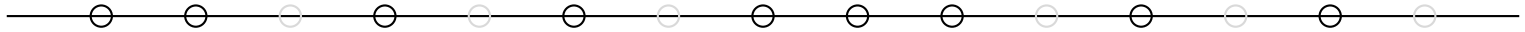
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at `http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence`

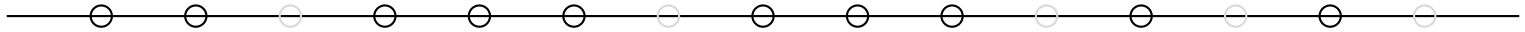
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathtt{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

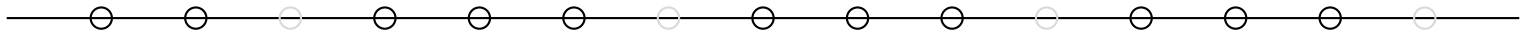
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathtt{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

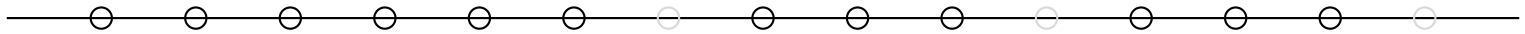
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

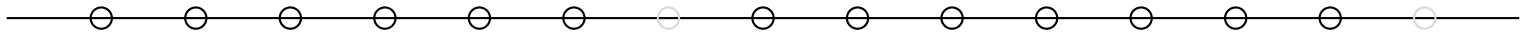
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

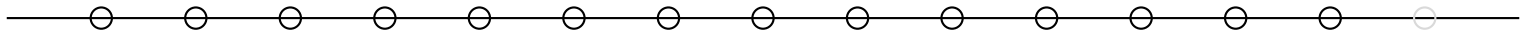
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathtt{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence

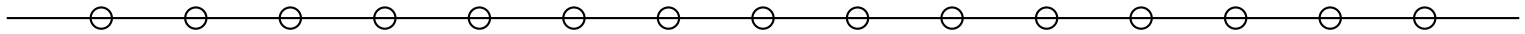
Example



Basic: Van der Corput sequence

- Generate a low discrepancy sequence over $[0, 1]$
- Denote $\mathrm{vd}(k, b)$ as a Van der Corput sequence of k points, where b is the base of a prime number.
- MATLAB source code is available at `http://www.mathworks.com/matlabcentral/fileexchange/15354-generate-a-van-der-corput-sequence`

Example

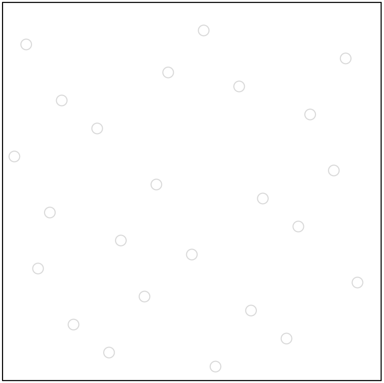


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

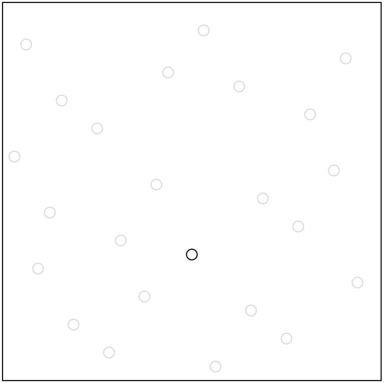


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

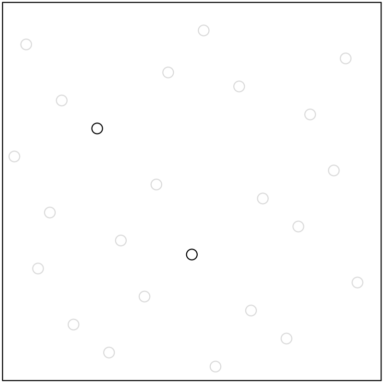


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

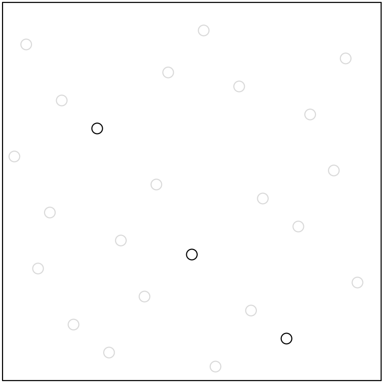


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

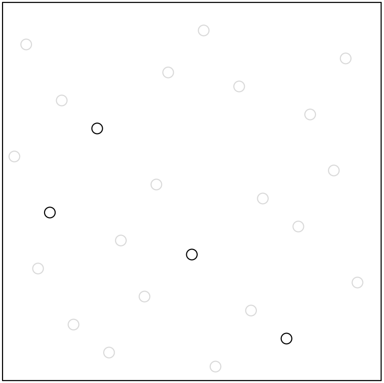


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

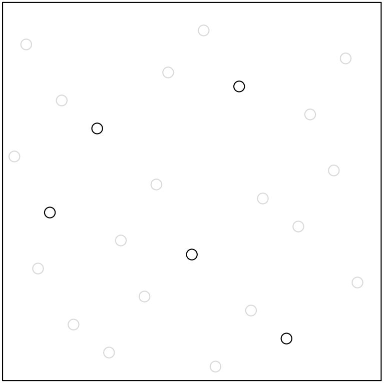


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

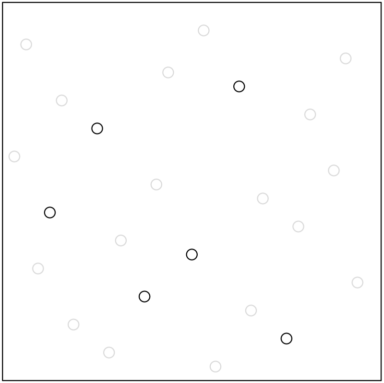


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

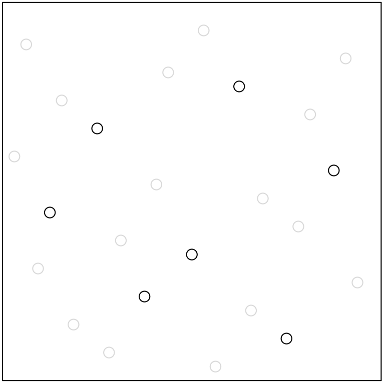


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

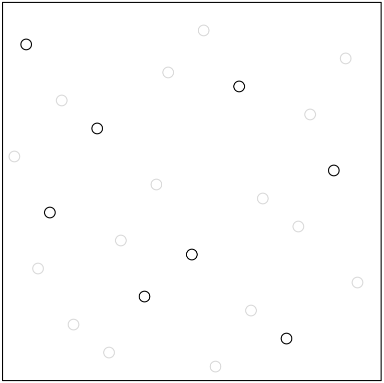


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

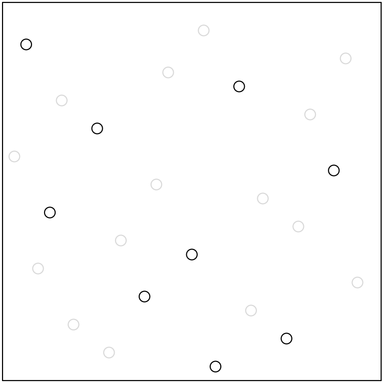


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

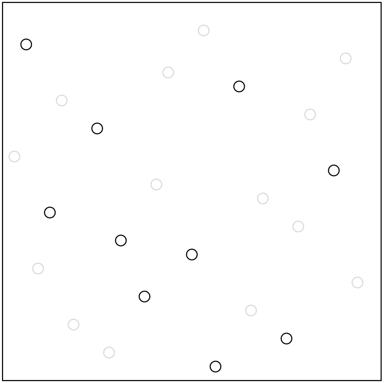


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\text{vd}(k, 2), \text{vd}(k, 3)]$

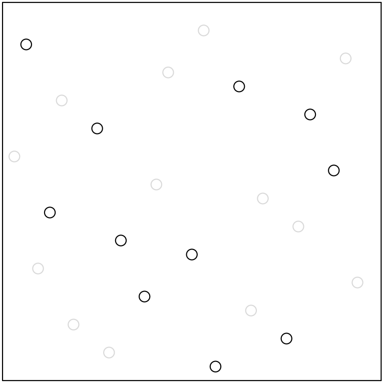


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

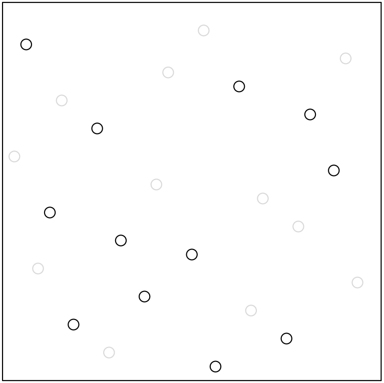


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

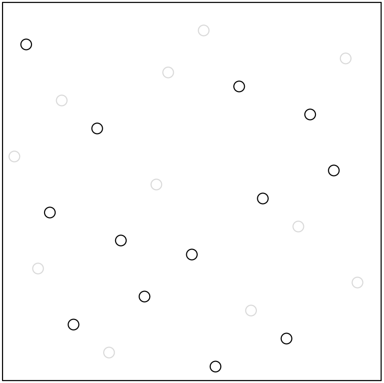


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\text{vd}(k, 2), \text{vd}(k, 3)]$

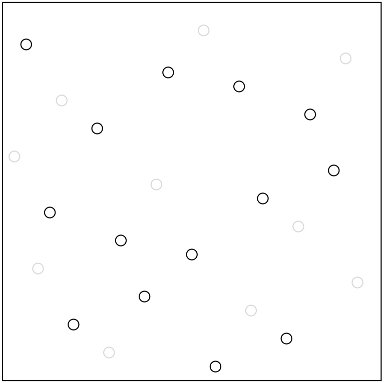


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

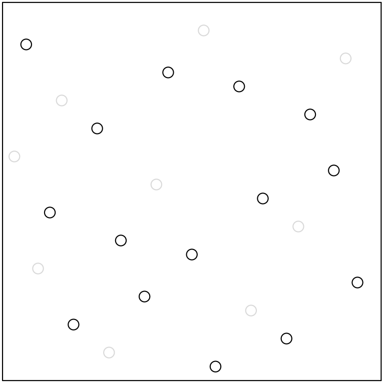


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

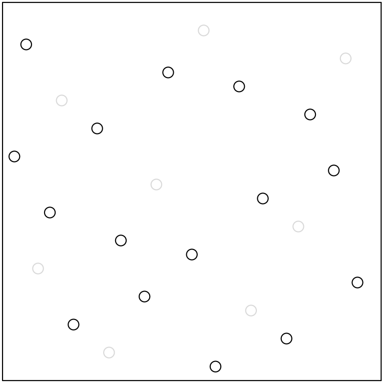


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

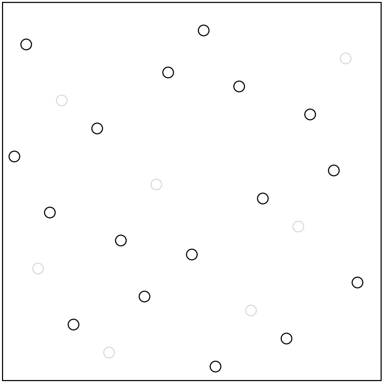


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

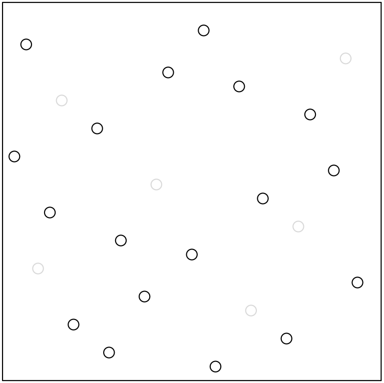


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

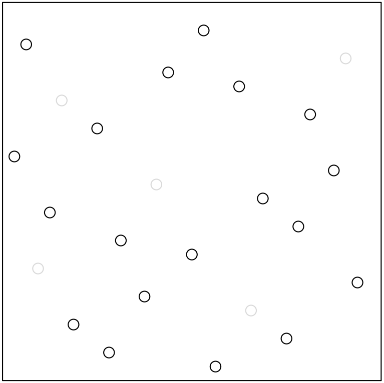


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

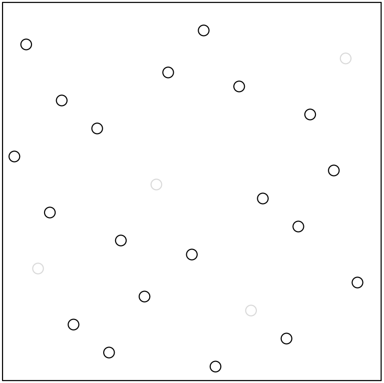


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

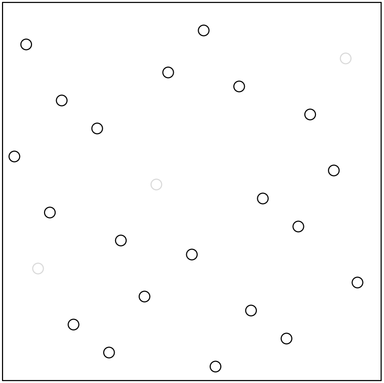


Unit Square $[0,1] \times [0,1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k,2), \mathrm{vd}(k,3)]$

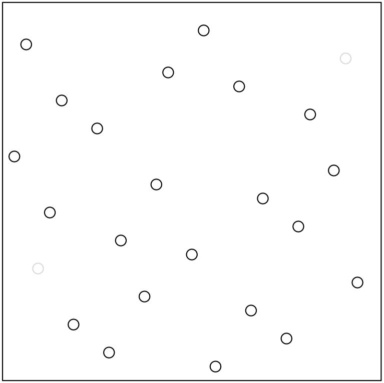


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$

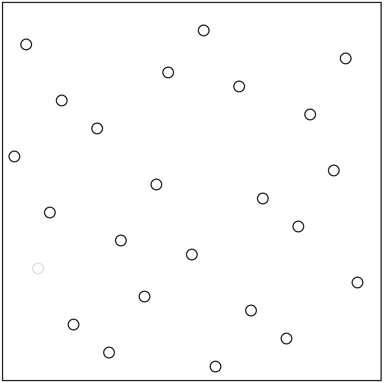


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\text{vd}(k, 2), \text{vd}(k, 3)]$

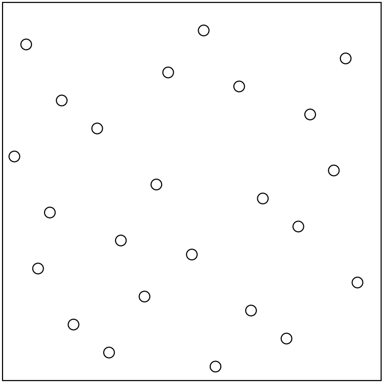


Unit Square $[0, 1] \times [0, 1]$

Halton sequence: using 2 Van der Corput sequences with different bases.

Example

$[x, y] = [\mathrm{vd}(k, 2), \mathrm{vd}(k, 3)]$



Unit Hypercube $[0,1]^n$

- Generally we can generate Halton sequence in a unit hypercube $[0,1]^n$:

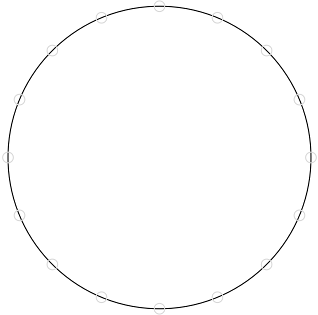
$$[x_1, x_2, \ldots, x_n] = [\mathrm{vd}(k, b_1), \mathrm{vd}(k, b_2), \ldots, \mathrm{vd}(k, b_n)]$$

- A wide range of applications on Quasi-Monte Carlo Methods (QMC).

Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

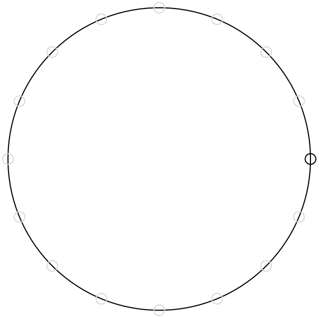
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

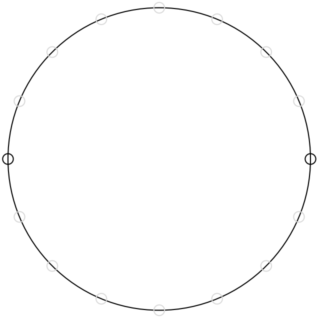
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

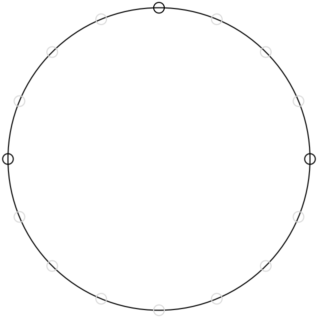
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

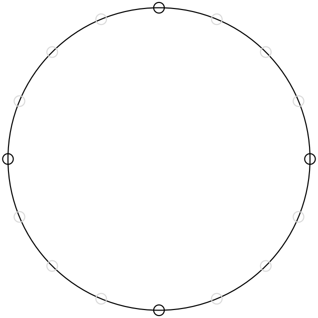
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

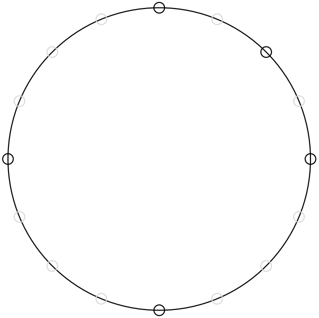
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

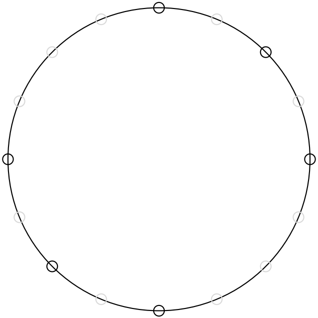
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

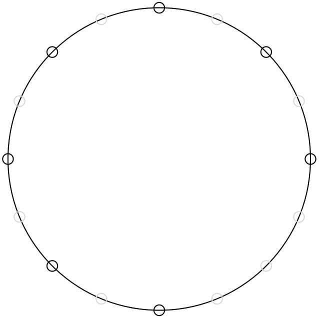
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

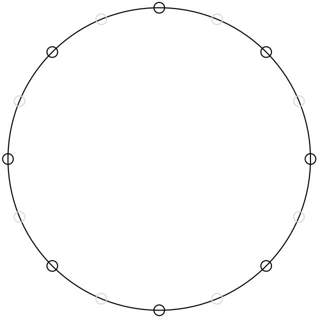
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

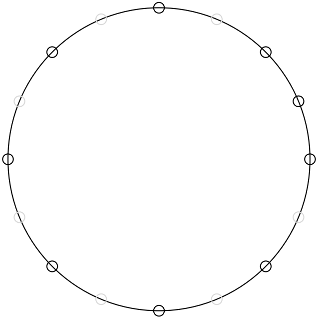
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

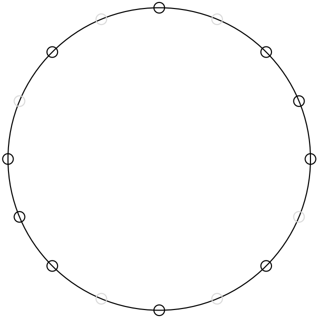
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

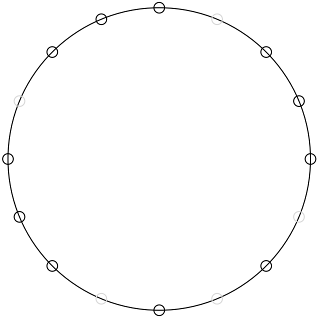
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

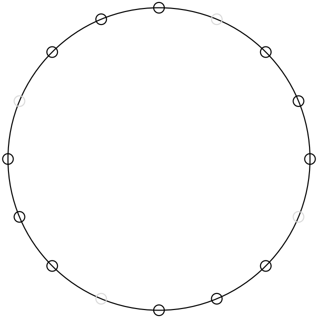
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

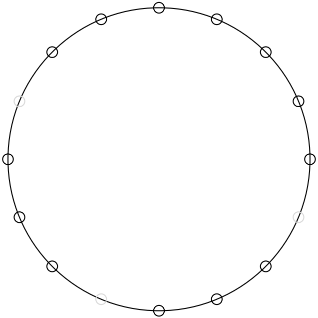
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

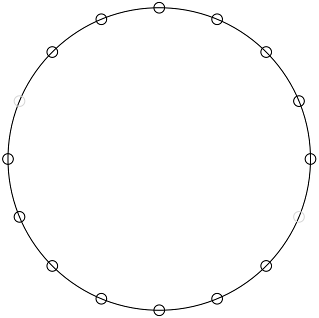
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

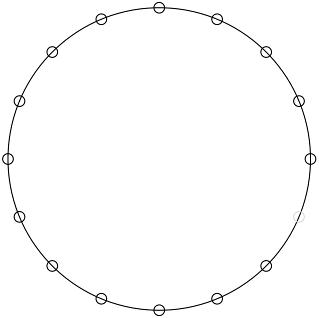
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

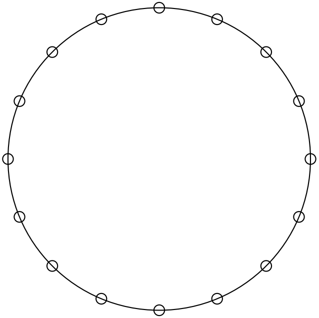
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Circle S^1

Can be generated by mapping the Van der Corput sequence to $[0, 2\pi]$

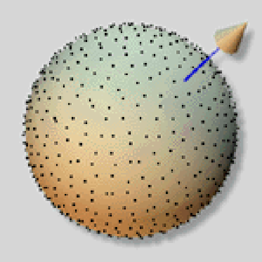
- $\theta = 2\pi \cdot \mathrm{vd}(k, b)$
- $[x, y] = [\cos\theta, \sin\theta]$



Unit Sphere S^2

Has been applied for computer graphic applications [WLH97]

- $[z, x, y]$
  $= [\cos\theta, \sin\theta\cos\varphi, \sin\theta\sin\varphi]$
  $= [z, \sqrt{1-z^2}\cos\varphi, \sqrt{1-z^2}\sin\varphi]$
- $\varphi = 2\pi \cdot \mathrm{vd}(k, b_1)$ % map to $[0, 2\pi]$
- $z = 2 \cdot \mathrm{vd}(k, b_2) - 1$ % map to $[-1, 1]$



Sphere S^3 and SO(3)

- Deterministic point sets
  - Optimal grid point sets for S^3, SO(3) [Lubotzky, Phillips, Sarnak 86] [Mitchell 07]
- No Halton sequences so far to the best of our knowledge

Agenda

SO(3) or S^3 Hopf Coordinates

- Hopf coordinates (cf. [YJLM10])
  - $x_1 = \cos(\theta/2)\cos(\psi/2)$
  - $x_2 = \cos(\theta/2)\sin(\psi/2)$
  - $x_3 = \sin(\theta/2)\cos(\varphi + \psi/2)$
  - $x_4 = \sin(\theta/2)\sin(\varphi + \psi/2)$
- S^3 is a principal circle bundle over the S^2

Hopf Coordinates for SO(3) or S^3

Similar to the Halton sequence generation on S^2, we perform the mapping:

- $\varphi = 2\pi \cdot \mathrm{vd}(k, b_1)$ % map to $[0, 2\pi]$
- $\psi = 2\pi \cdot \mathrm{vd}(k, b_2)$ % map to $[0, 2\pi]$ for SO(3), or
- $\psi = 4\pi \cdot \mathrm{vd}(k, b_2)$ % map to $[0, 4\pi]$ for S^3
- $z = 2 \cdot \mathrm{vd}(k, b_3) - 1$ % map to $[-1, 1]$
- $\theta = \cos^{-1} z$

10 Lines of MATLAB Code

```
function[s] = sphere3_hopf(k,b)
% sphere3_hopf   Halton sequence
varphi = 2*pi*vdcorput(k,b(1));    % map to [0, 2*pi]
psi = 4*pi*vdcorput(k,b(2));       % map to [0, 4*pi]
z = 2*vdcorput(k,b(3)) - 1;        % map to [-1, 1]
theta = acos(z);
cos_eta = cos(theta/2);
sin_eta = sin(theta/2);
s = [cos_eta .* cos(psi/2), ...
     cos_eta .* sin(psi/2), ...
     sin_eta .* cos(varphi + psi/2), ...
     sin_eta .* sin(varphi + psi/2)];
```

3-sphere

- Polar coordinates:
  - $x_0 = \cos\theta_3$
  - $x_1 = \sin\theta_3 \cos\theta_2$
  - $x_2 = \sin\theta_3 \sin\theta_2 \cos\theta_1$
  - $x_3 = \sin\theta_3 \sin\theta_2 \sin\theta_1$

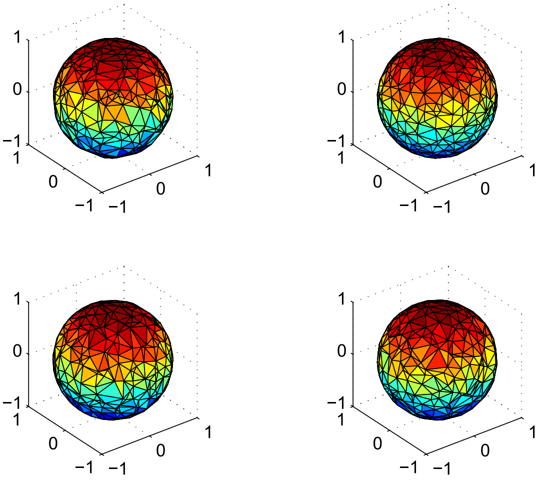
n-sphere

- Polar coordinates:
  - $x_0 = \cos\theta_n$
  - $x_1 = \sin\theta_n \cos\theta_{n-1}$
  - $x_2 = \sin\theta_n \sin\theta_{n-1} \cos\theta_{n-2}$
  - $x_3 = \sin\theta_n \sin\theta_{n-1} \sin\theta_{n-2} \cos\theta_{n-3}$
  - $\cdots$
  - $x_{n-1} = \sin\theta_n \sin\theta_{n-1} \sin\theta_{n-2} \cdots \cos\theta_1$
  - $x_n = \sin\theta_n \sin\theta_{n-1} \sin\theta_{n-2} \cdots \sin\theta_1$

How to Generate the Point Set

- $p_0 = [\cos\theta_1, \sin\theta_1]$ where $\theta_1 = 2\pi \cdot \mathrm{vd}(k, b_1)$
- Let $f_j(\theta) = \int \sin^j \theta d\theta$, where $\theta \in (0, \pi)$.
  Note: $f_j(\theta)$ is a monotonic increasing function in $(0, \pi)$
- Map $\mathrm{vd}(k, b_j)$ uniformly to $f_j(\theta)$:
  $t_j = f_j(0) + (f_j(\pi) - f_j(0))\mathrm{vd}(k, b_j)$
- Let $\theta_j = f_j^{-1}(t_j)$
- Define p_n recursively as:
  $p_n = [\cos\theta_n, \sin\theta_n \cdot p_{n-1}]$

S^3 projected on four different spheres



Agenda

Testing the Correctness

- Compare the dispersion with the random point-set
  - Construct the convex hull for each point-set
  - Dispersion roughly measured by the difference of the maximum distance and the minimum distance between every two neighbour points:

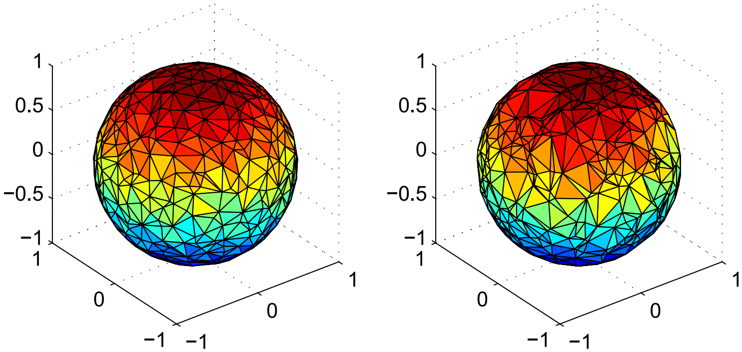
$$\max_{a \in \mathcal{N}(b)} \{D(a, b)\} - \min_{a \in \mathcal{N}(b)} \{D(a, b)\}$$

where $D(a, b) = \sqrt{1 - a^{\mathrm{T}} b}$

Random sequences

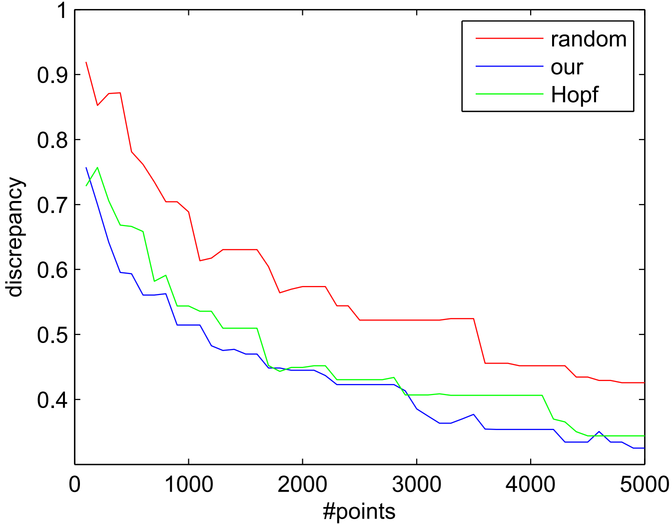
- To generate random points on S^n, spherical symmetry of the multidimensional Gaussian density function can be exploited.
- Then the normalized vector $(x_i/||x_i||)$ is uniformly distributed over the hypersphere S^n. [Fishman, G. F. (1996)]

Convex Hull with ≈400 points

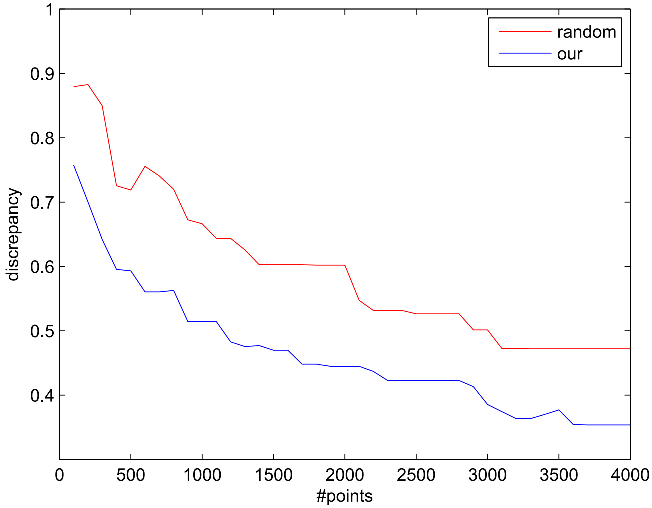


Left: our, right: random

Result for S^3



Result for S^4



Agenda

Conclusions

- Proposed method generates low-discrepancy point-set in nearly linear time
- The result outperforms the corresponding random point-set, especially when the number of points is small
- The MATLAB source code is available in public (or upon request)

References I

[CF97] Jianjun Cui and Willi Freeden, *Equidistribution on the sphere*, SIAM Journal on Scientific Computing **18** (1997), no. 2, 595–609.

[M⁺11] DP Mandic et al., *Filter bank property of multivariate empirical mode decomposition*, Signal Processing, IEEE Transactions on **59** (2011), no. 5, 2421–2426.

[RM10] Naveed Rehman and Danilo P Mandic, *Multivariate empirical mode decomposition*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science **466** (2010), no. 2117, 1291–1302.

[UL06] Zoran Utkovski and Juergen Lindner, *On the construction of non-coherent space time codes from high-dimensional spherical codes*, Spread Spectrum Techniques and Applications, 2006 IEEE Ninth International Symposium on, IEEE, 2006, pp. 327–331.

[WLH97] Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng, *Sampling with hammersley and halton points*, Journal of graphics tools **2** (1997), no. 2, 9–24.

[YJLM10] Anna Yershova, Swati Jain, Steven M LaValle, and Julie C Mitchell, *Generating uniform incremental grids on so (3) using the hopf fibration*, The International journal of robotics research **29** (2010), no. 7, 801–812.