Sampling with Halton Points on n-Sphere

Wai-Shing Luk

# Abstract

## Abstract

* Sampling on -sphere () 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 a few lines of Python code in our implementation!
  + Allow incremental generation.
* Numerical results show that the proposed method outperforms the randomly generated sequences and other proposed methods.

# Motivation and Applications

## Problem Formulation

Desirable properties of samples over

* 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.
* Yet it is still unknown how to generate for .
* Potential applications (for ):
  + Robotic Motion Planning ( and SO(3)) [1]
  + Spherical coding in MIMO wireless communication [2]:
    - Cookbook for Unitary matrices
    - A code word = a point in
  + Multivariate empirical mode decomposition [3]
  + Filter bank design [4]

## Halton Sequence on

* Halton sequence on has been well studied [5] by using cylindrical coordinates.
* Yet it is still little known for where .
* Note: The generalization of cylindrical coordinates does NOT work in higher dimensions.

# Review of Low Discrepancy Sequence

## Basic: Van der Corput sequence

* Generate a low discrepancy sequence over
* Denote as a Van der Corput sequence of points, where is the base of a prime number.

## Python code

def vdc\_basic(n, base=2):  
 vdc, denom = 0.0, 1.0  
 while n:  
 denom \*= base  
 n, remainder = divmod(n, base)  
 vdc += remainder / denom  
 return vdc  
  
def vdc(n, base=2):  
 '''  
 n - number of vectors  
 base - seeds  
 '''  
 for i in range(n):  
 yield vdc\_basic(i, base)

## Halton sequence on

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

## Halton sequence on

* Generally we can generate Halton sequence in a unit hypercube :
* A wide range of applications on Quasi-Monte Carlo Methods (QMC).

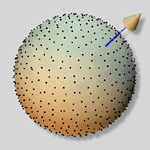
## Unit Circle

Can be generated by mapping the Van der Corput sequence to

## Unit Sphere

Has been applied for computer graphic applications [6]

* Use cylindrical mapping.
* =   
  =
* % map to
* % map to



image

## Sphere and SO(3)

* Deterministic point sets
  + Optimal grid point sets for , SO(3)[1,7]
* No Halton sequences so far to the best of our knowledge.
* Note that cylindrical mapping method cannot be extended to higher dimensions.

## SO(3) or Hopf Coordinates

* Hopf coordinates (cf. [1])
* is a principal circle bundle over the



image

## Hopf Coordinates for SO(3) or

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

* % map to
* % map to for SO(3), or
* % map to for
* % map to

## Python Code

def sphere3\_hopf(k, b):  
 vd = zip(vdc(k, b[0]), vdc(k, b[1]), vdc(k, b[2]))  
 for vd0, vd1, vd2 in vd:  
 phi = 2\*math.pi\*vd0 # map to [0, 2\*math.pi]  
 psy = 4\*math.pi\*vd1 # map to [0, 4\*math.pi]  
 z = 2\*vd2 - 1 # map to [-1., 1.]  
 theta = math.acos(z)  
 cos\_eta = math.cos(theta/2)  
 sin\_eta = math.sin(theta/2)  
 s = [cos\_eta \* math.cos(psy/2),  
 cos\_eta \* math.sin(psy/2),  
 sin\_eta \* math.cos(phi + psy/2),  
 sin\_eta \* math.sin(phi + psy/2)]  
 yield s

# Our approach

## 3-sphere

* Polar coordinates:
* Spherical surface element:

## n-sphere

* Polar coordinates:
* Spherical surface element:

## How to Generate the Point Set

* where
* Let = , where .  
  + Note 1: can be defined recursively as:
  + Note 2: is a monotonic increasing function in
* Map uniformly to :
* Let
* Define recursively as:

# Numerical Experiments

## 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:
  + where

## Random sequences

* To generate random points on , spherical symmetry of the multidimensional Gaussian density function can be exploited.
* Then the normalized vector () is uniformly distributed over the hypersphere . (Fishman, G. F. (1996))

## Convex Hull with points

![image](data:application/eps;base64,)

image

Left: our, right: random

## Result for

Compared with Hopf coordinate method.

image

image

## Result for (II)

Compared with cylindrical mapping method.

image

image

## Result for

Compared with cylindrical mapping method

image

image

# Conclusions

## 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
* Python code is available at [here](http://github.com/luk036/n-sphere/)

## References

[1] YERSHOVA A, JAIN S, LAVALLE S M, 等. Generating uniform incremental grids on SO (3) using the Hopf fibration[J]. The International journal of robotics research, SAGE Publications, 2010, 29(7): 801–812.

[2] UTKOVSKI Z, LINDNER J. On the construction of non-coherent space time codes from high-dimensional spherical codes[C]//Spread Spectrum Techniques and Applications, 2006 IEEE Ninth International Symposium on. IEEE, 2006: 327–331.

[3] REHMAN N, MANDIC D P. Multivariate empirical mode decomposition[J]. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, The Royal Society, 2010, 466(2117): 1291–1302.

[4] MANDIC D, OTHERS. Filter bank property of multivariate empirical mode decomposition[J]. Signal Processing, IEEE Transactions on, IEEE, 2011, 59(5): 2421–2426.

[5] CUI J, FREEDEN W. Equidistribution on the sphere[J]. SIAM Journal on Scientific Computing, SIAM, 1997, 18(2): 595–609.

[6] WONG T-T, LUK W-S, HENG P-A. Sampling with Hammersley and Halton points[J]. Journal of graphics tools, Taylor & Francis, 1997, 2(2): 9–24.

[7] MITCHELL J C. Sampling rotation groups by successive orthogonal images[J]. SIAM Journal on Scientific Computing, SIAM, 2008, 30(1): 525–547.