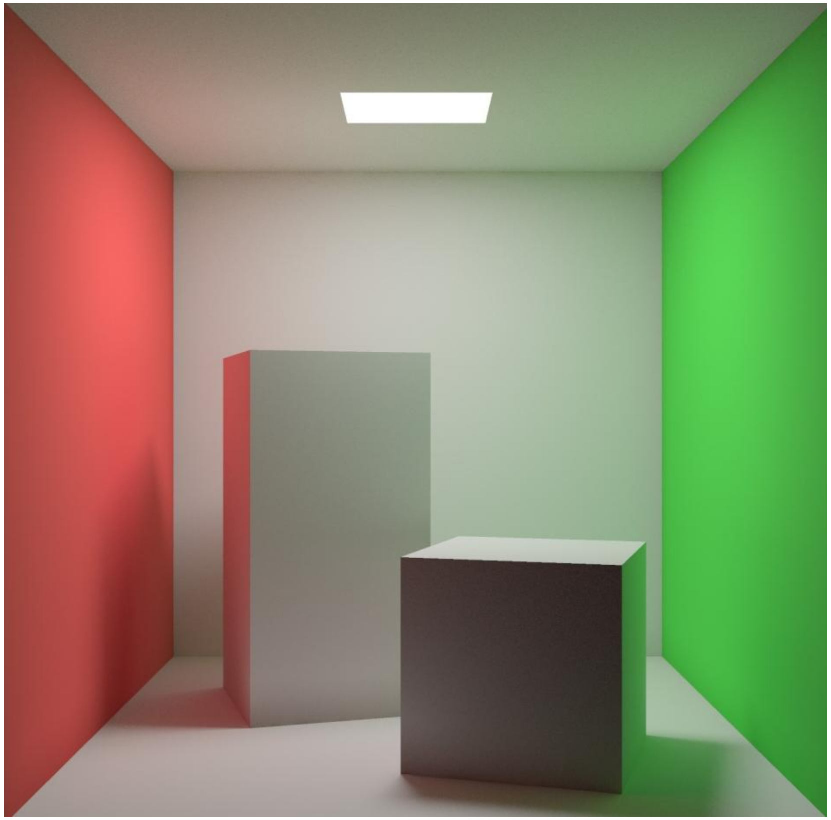
Encoding Normal Vectors using Optimized Spherical Coordinates

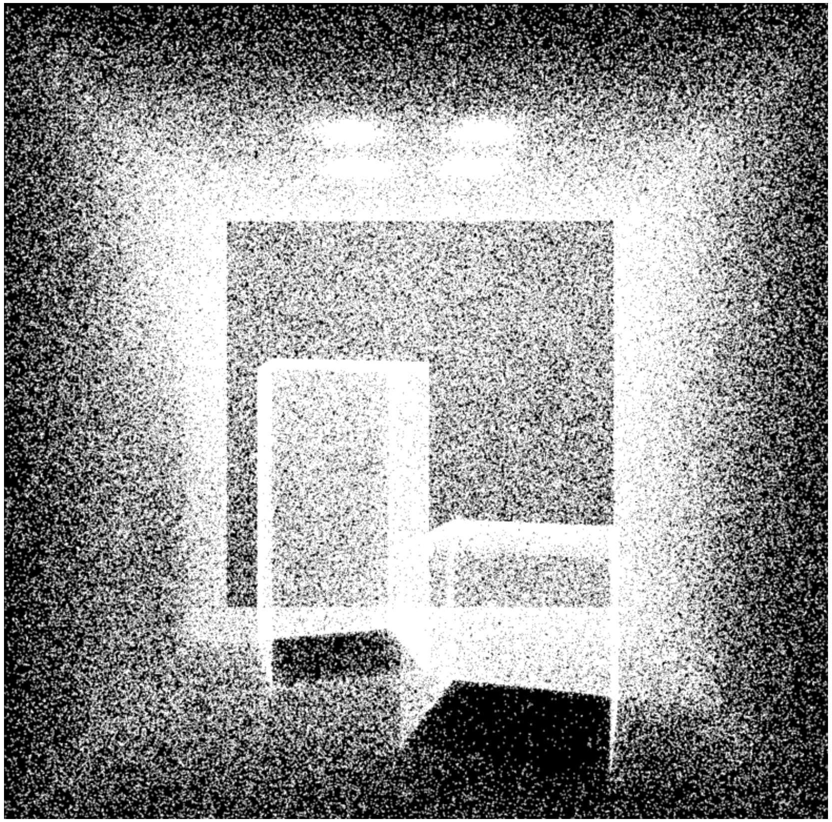
J. Smith, G. Petrova, S. Schaefer

Texas A&M University

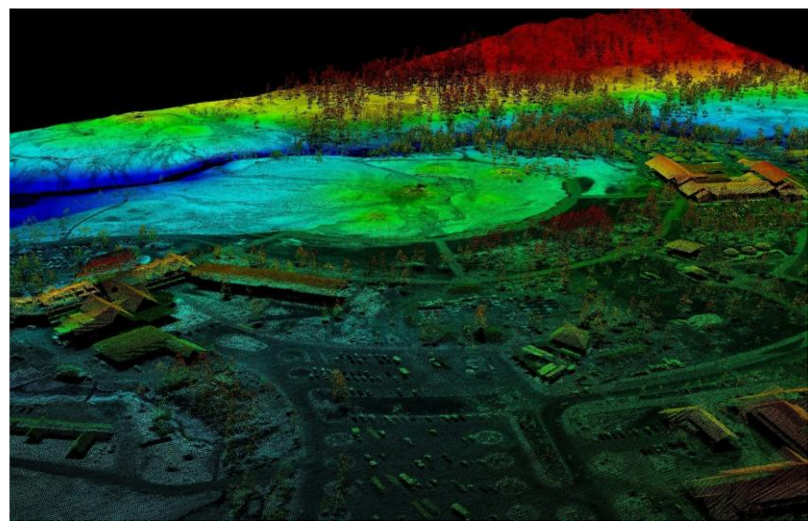
Motivation

Motivation





Motivation

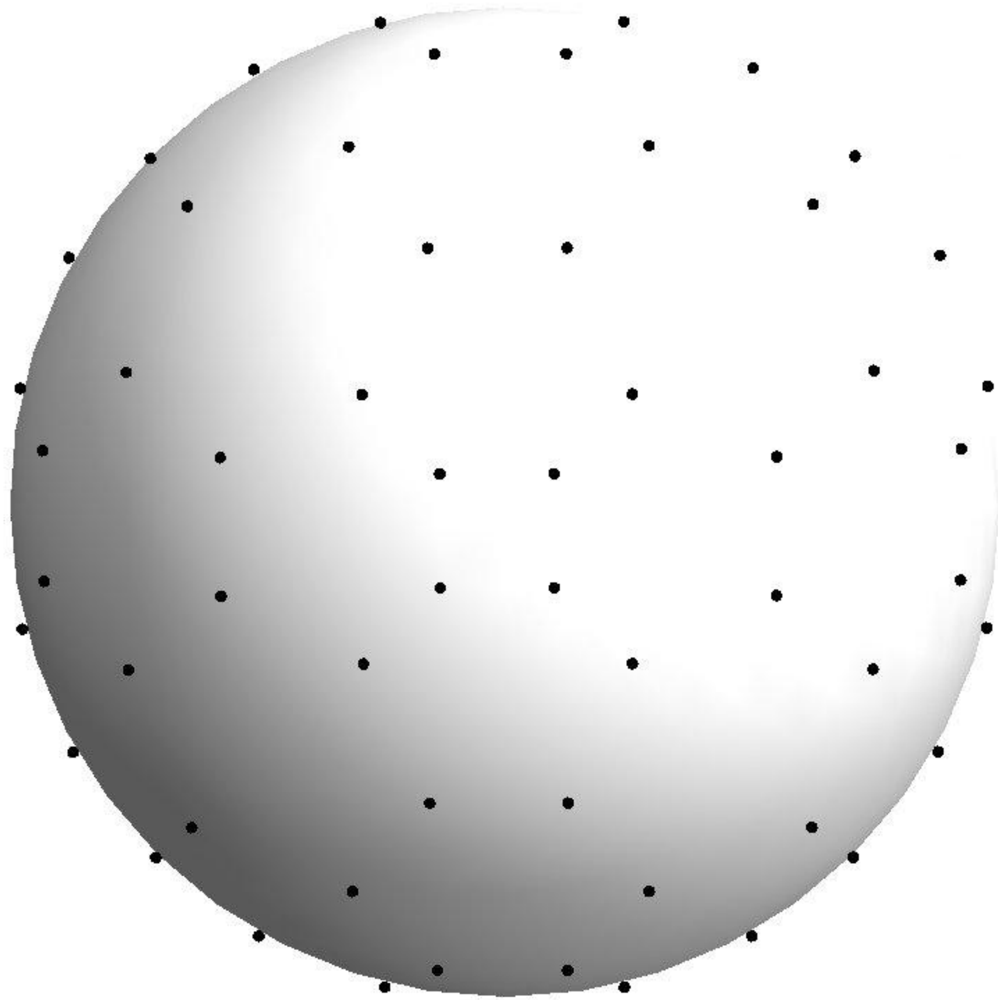


Motivation

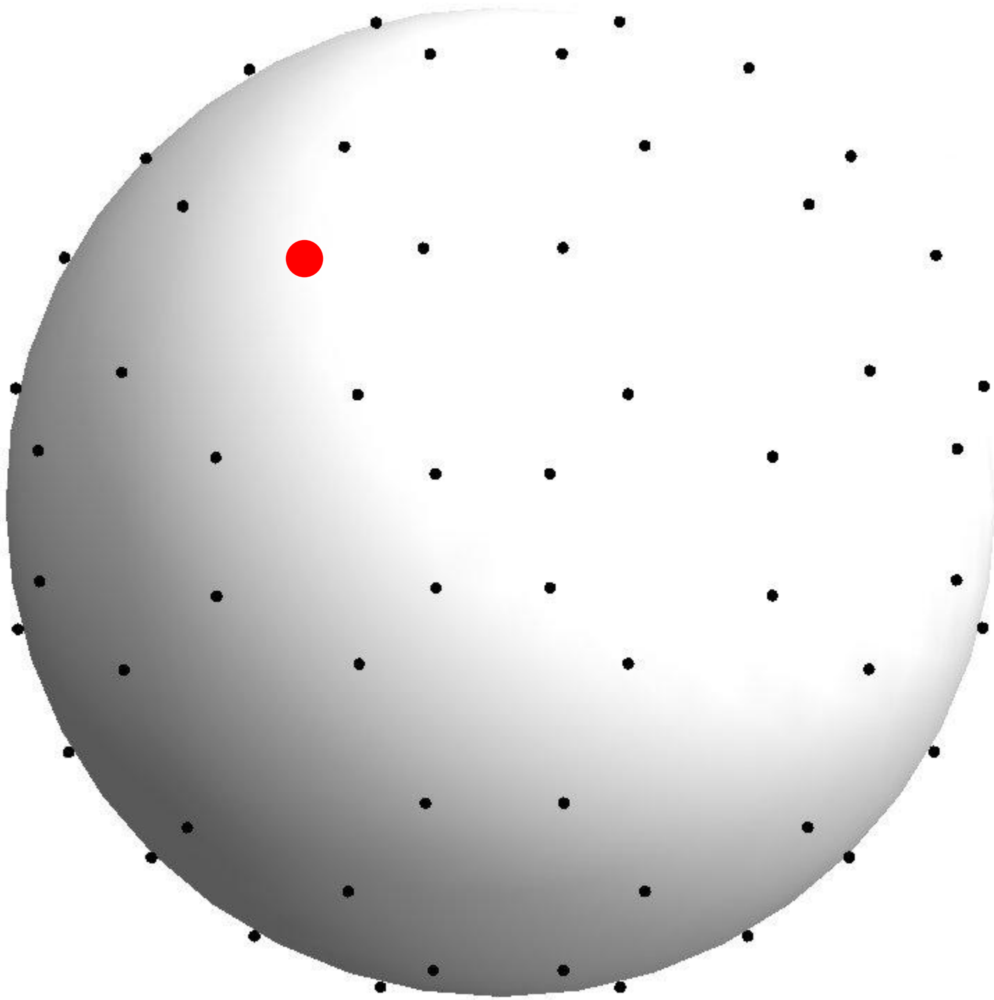
Normal Vectors

- On floating-point normal vectors [Meyer et al. 2010]
  – 96 bit vectors redundant
  – Only 51 bits are sufficient to represent floating point accuracy
- Is floating point necessary?
  – Bound error
  – Robust
  – Efficient encode / Decode

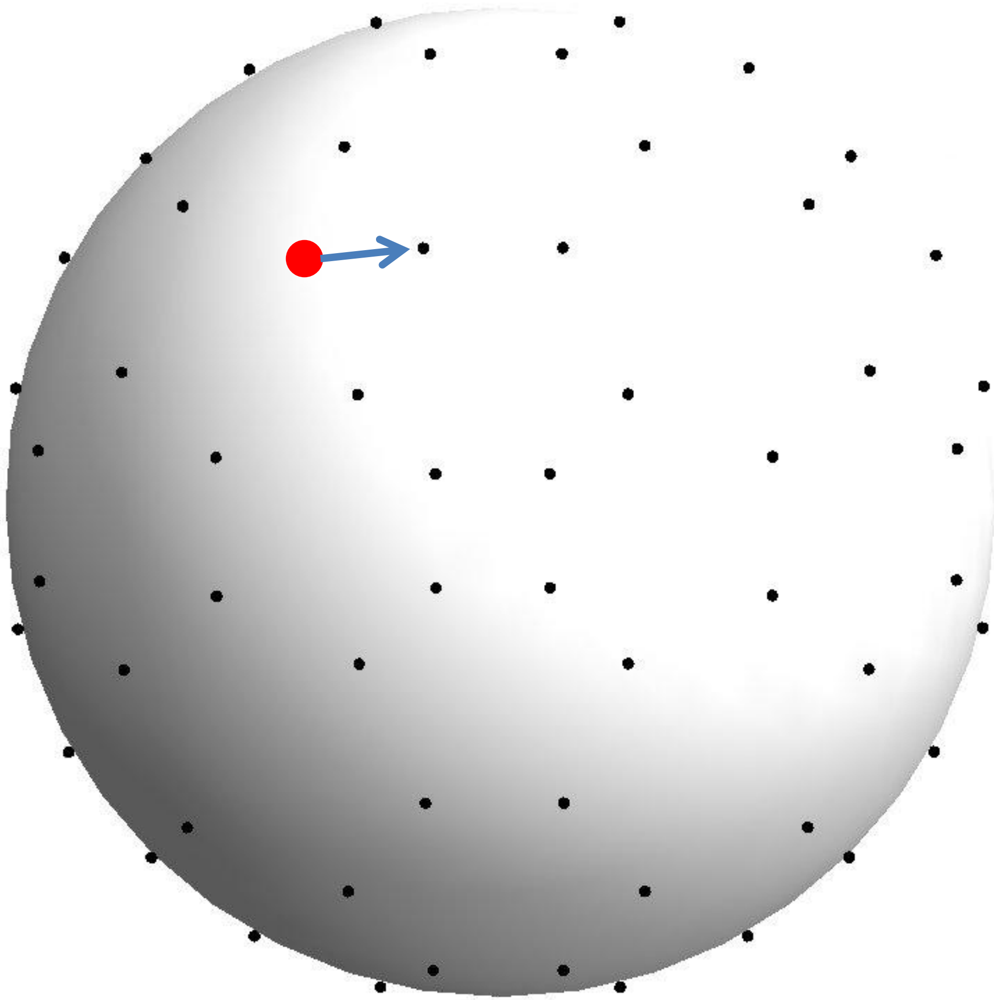
Point Distribution



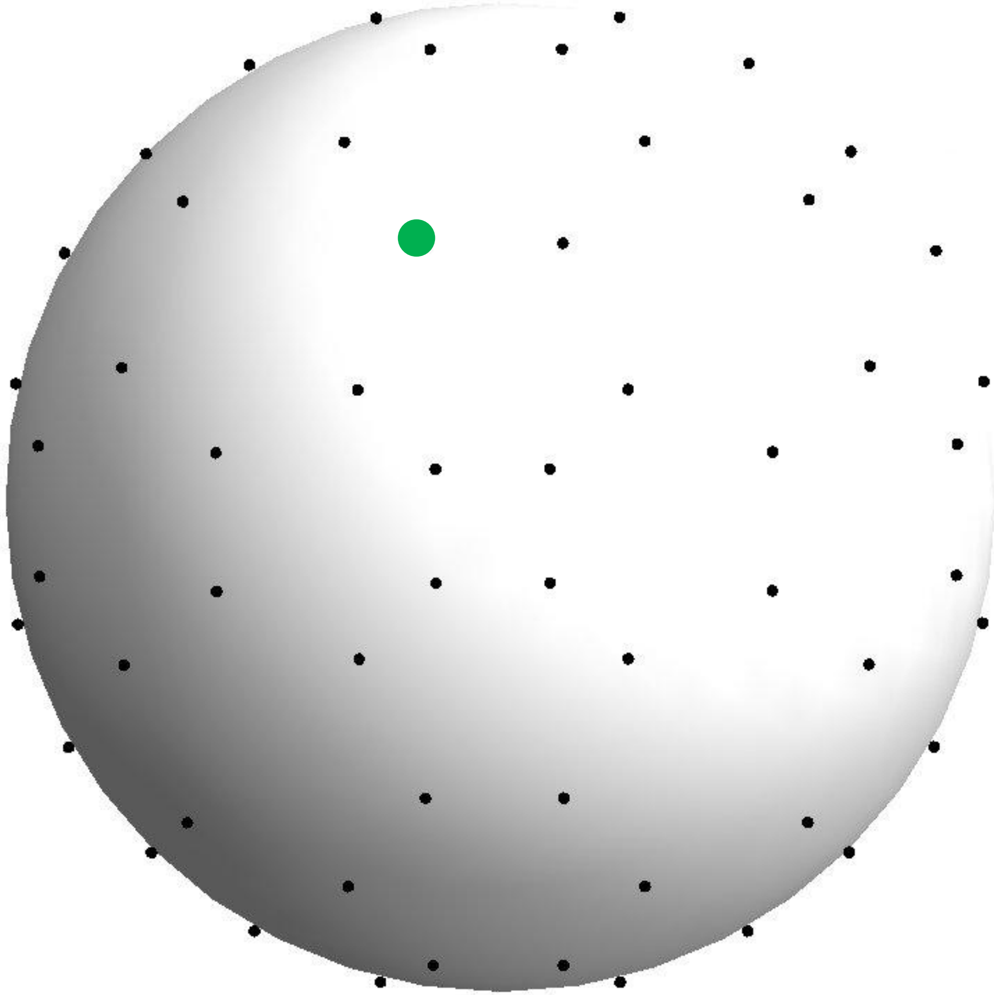
Point Distribution



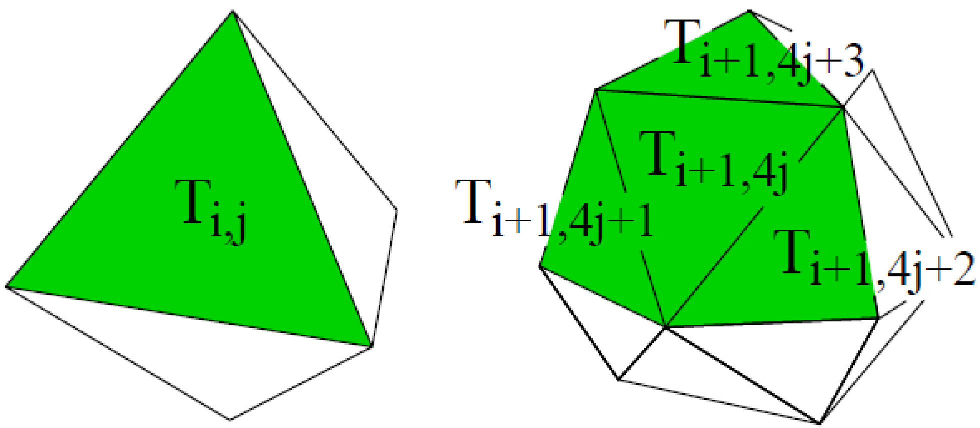
Point Distribution



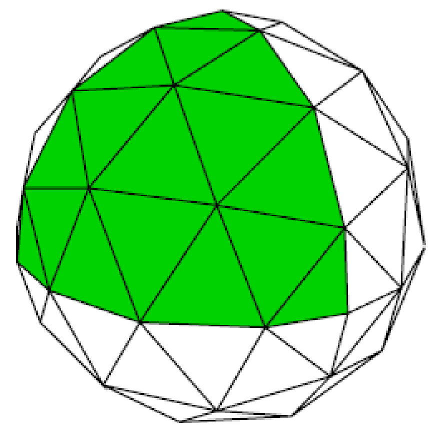
Point Distribution



Related Work

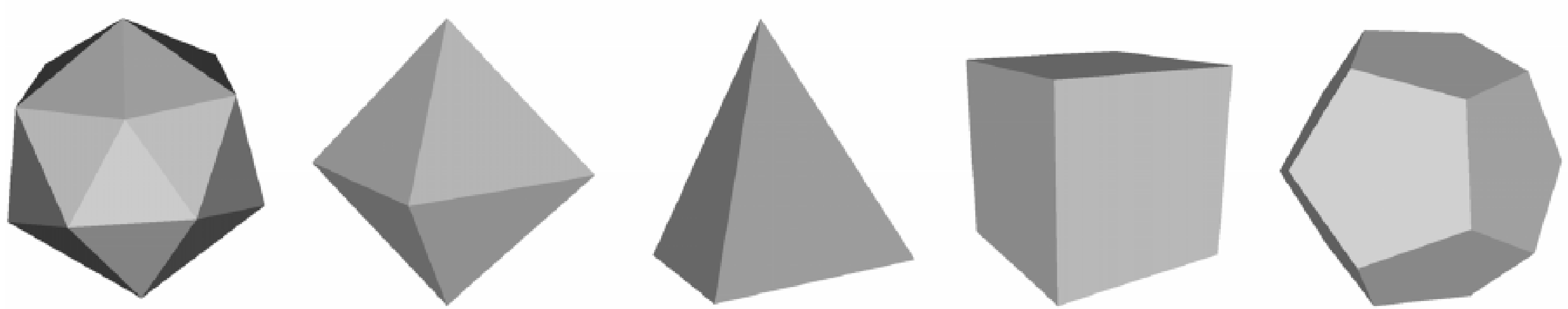


[Botsch et al. 2002]

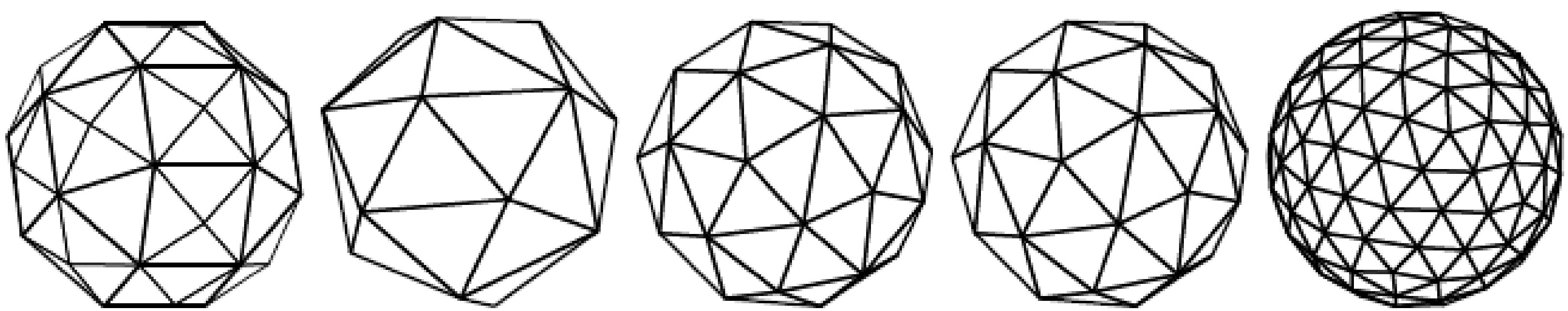


[Taubin et al. 1998]

Related Work

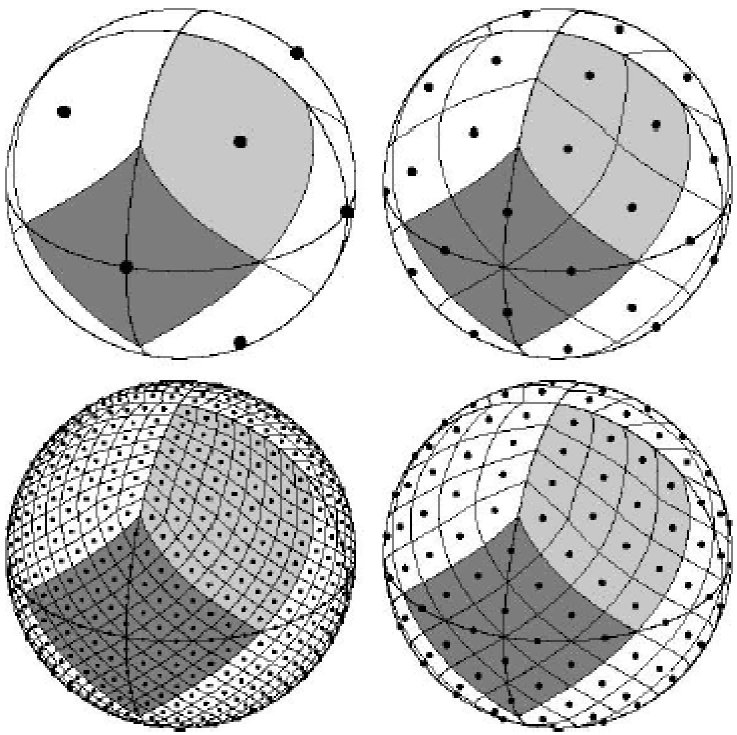


[Oliveira and Buxton 2006]

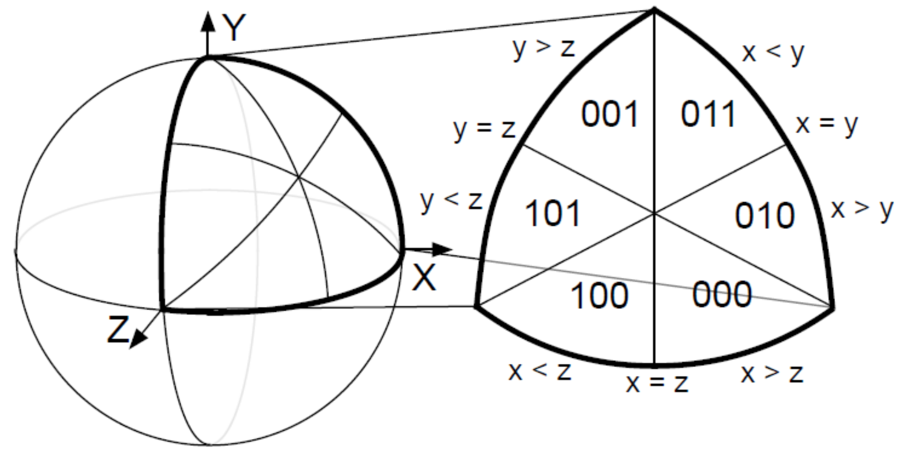


[Griffith et al. 2007]

Related Work

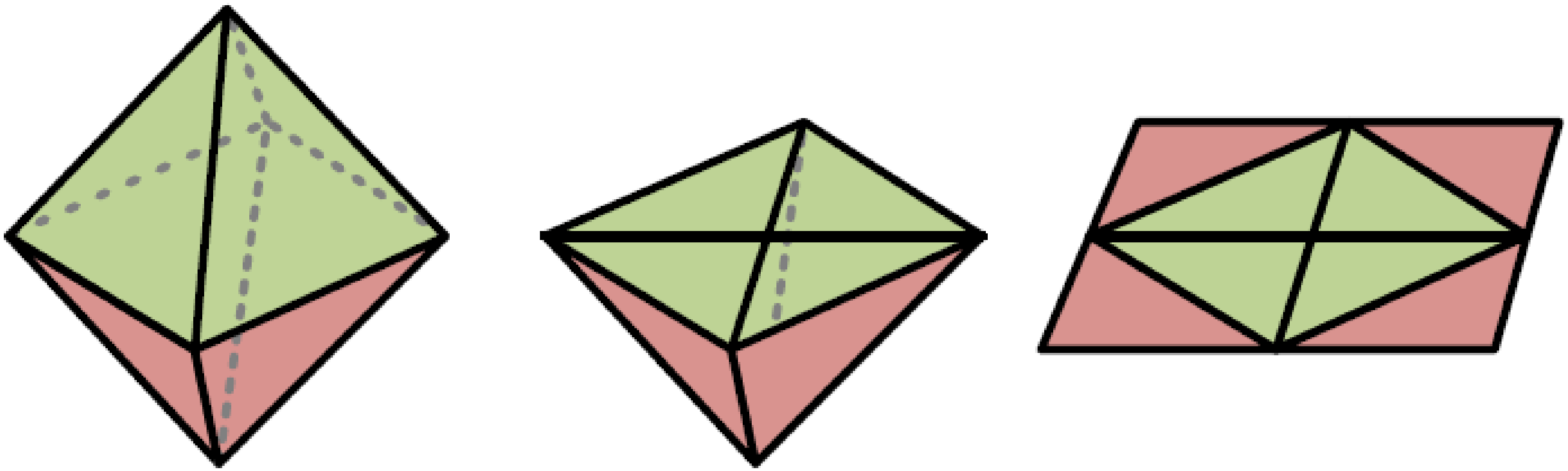


[Górski et al. 2004]



[Deering 1995]

Related Work

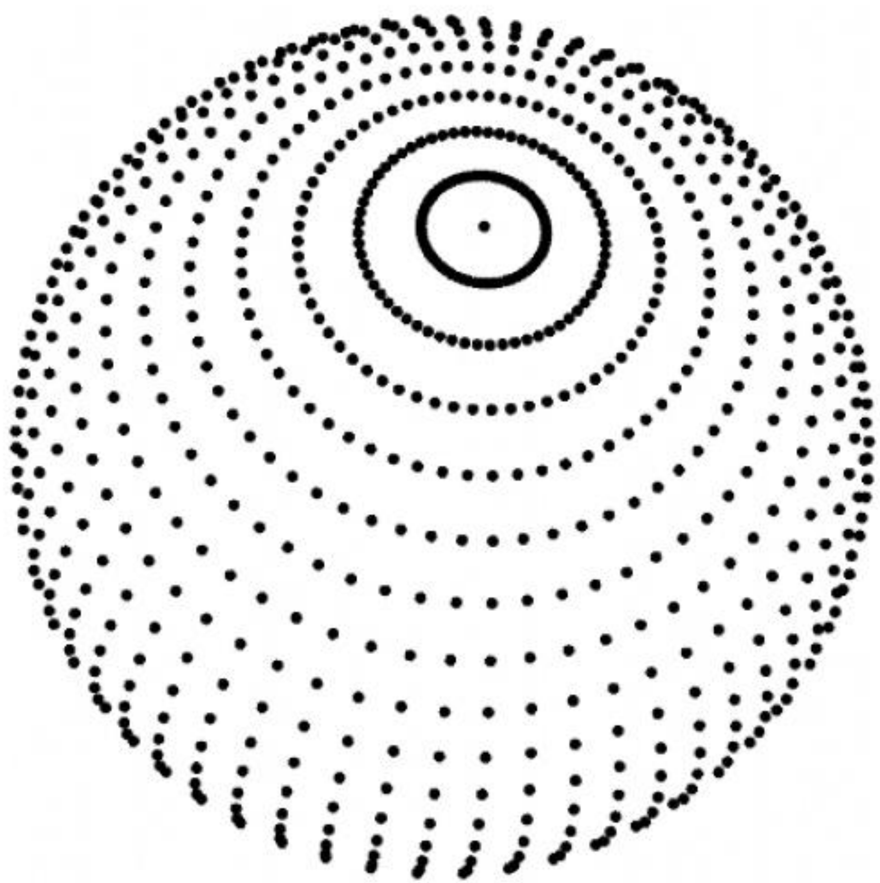


[Meyer et al. 2010]

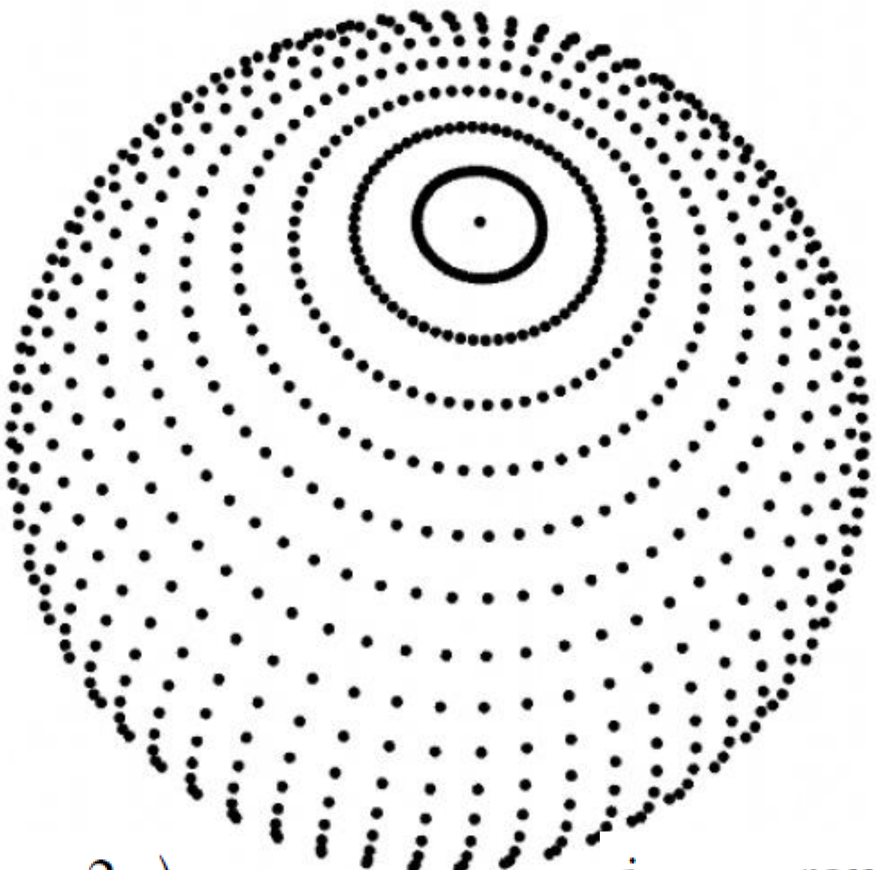
Contributions

- User Specified Maximal Bounded Error
- Variable Bit Encoding
- Constant Time Encode and Decode
  - Independent of accuracy
- Differential Encoding Method
  - Usable with other methods

Spherical Coordinates



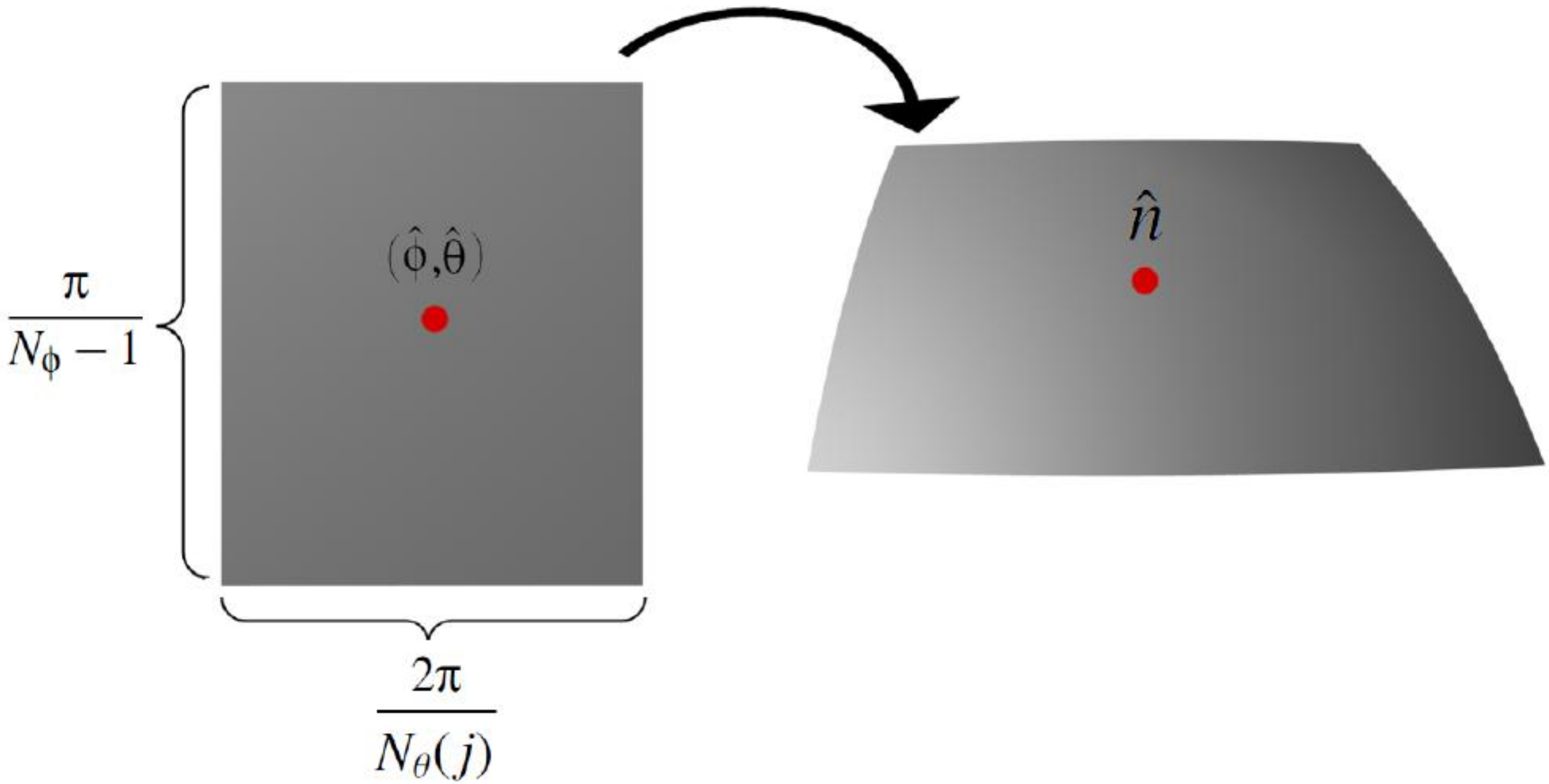
Spherical Coordinates



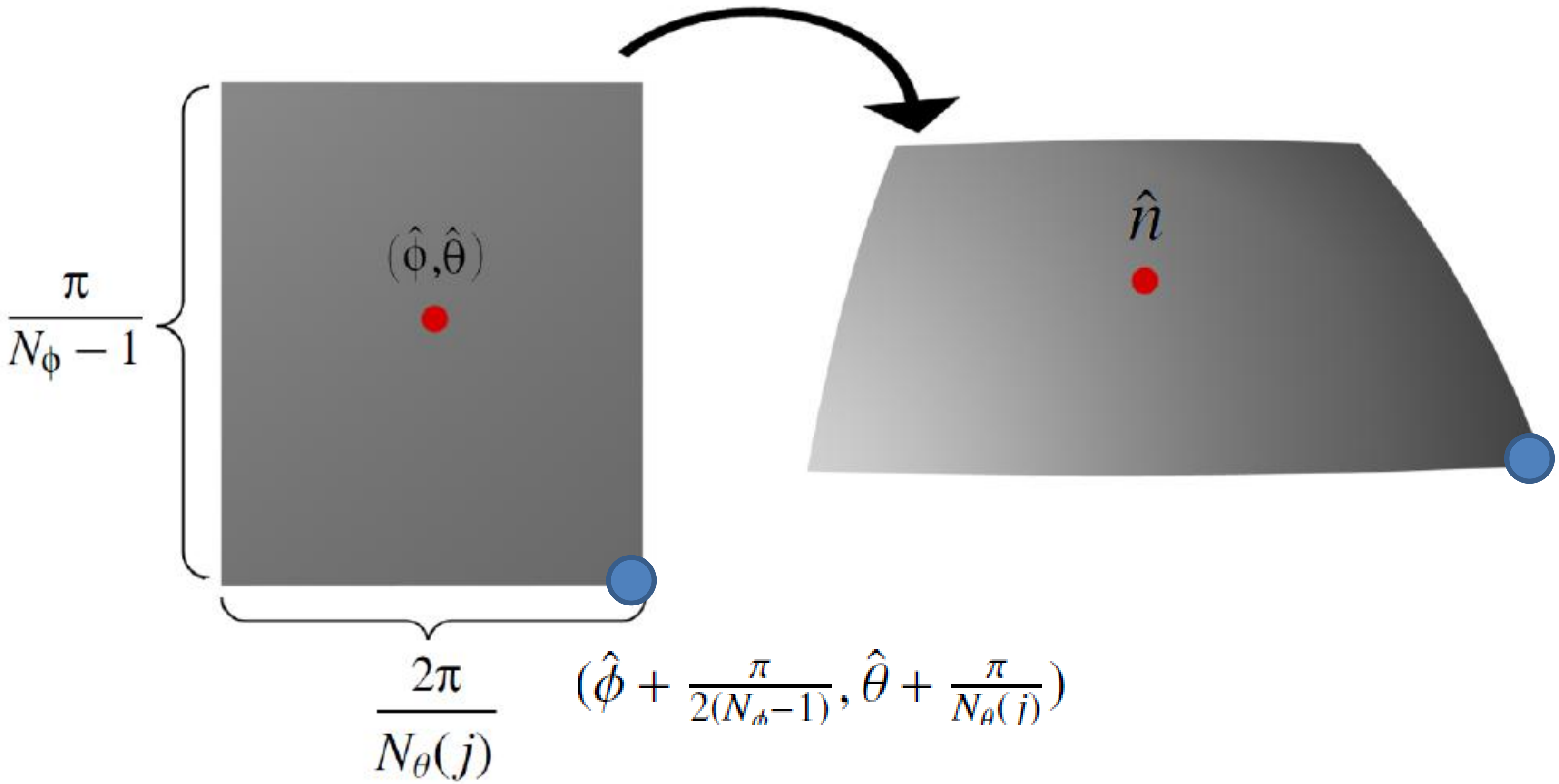
$$(\hat{\phi}, \hat{\theta}) = \left(j \frac{\pi}{N_\phi - 1}, k \frac{2\pi}{N_\theta} \right)$$

$$\begin{aligned} j &= \text{round}\left(\frac{\phi(N_\phi - 1)}{\pi}\right) \\ k &= \text{round}\left(\frac{\theta N_\theta}{2\pi}\right) \bmod N_\theta \end{aligned}$$

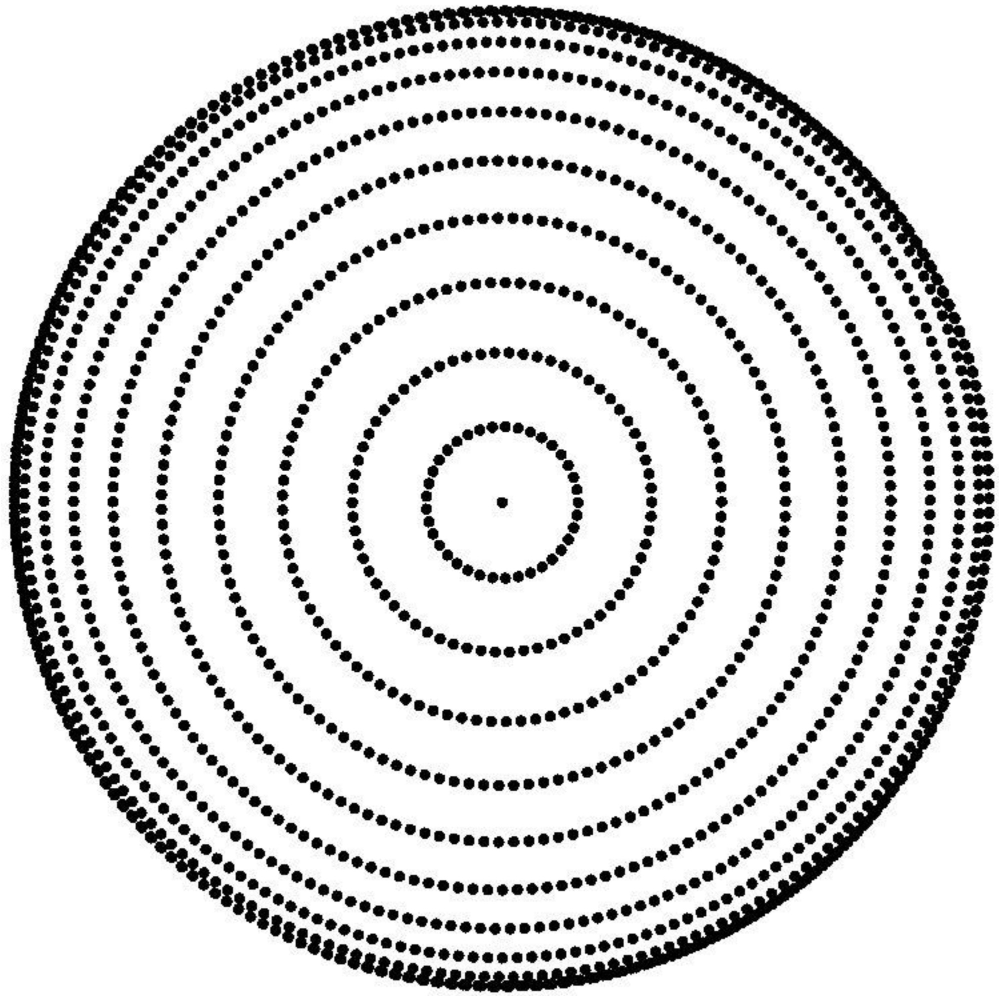
Rectangular Domain



Solve for Minimum $N_\theta(j)$

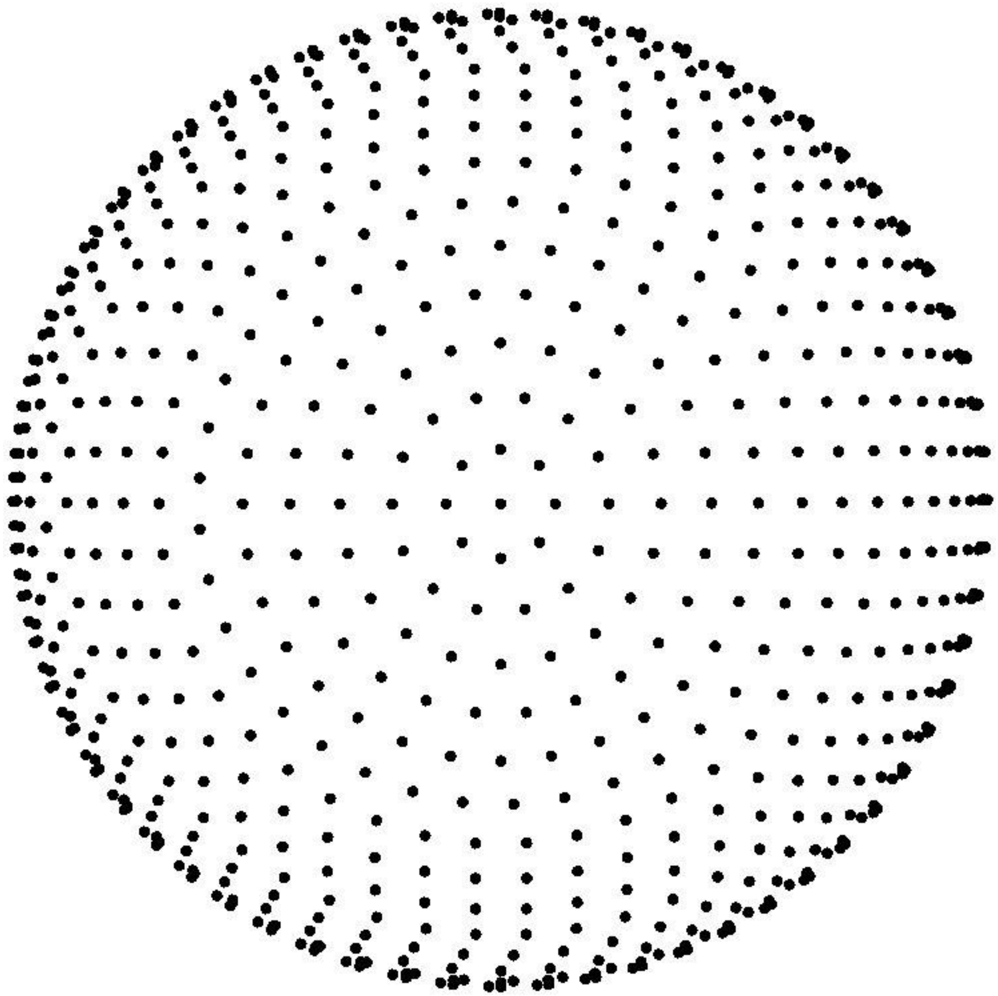


Choose N_ϕ



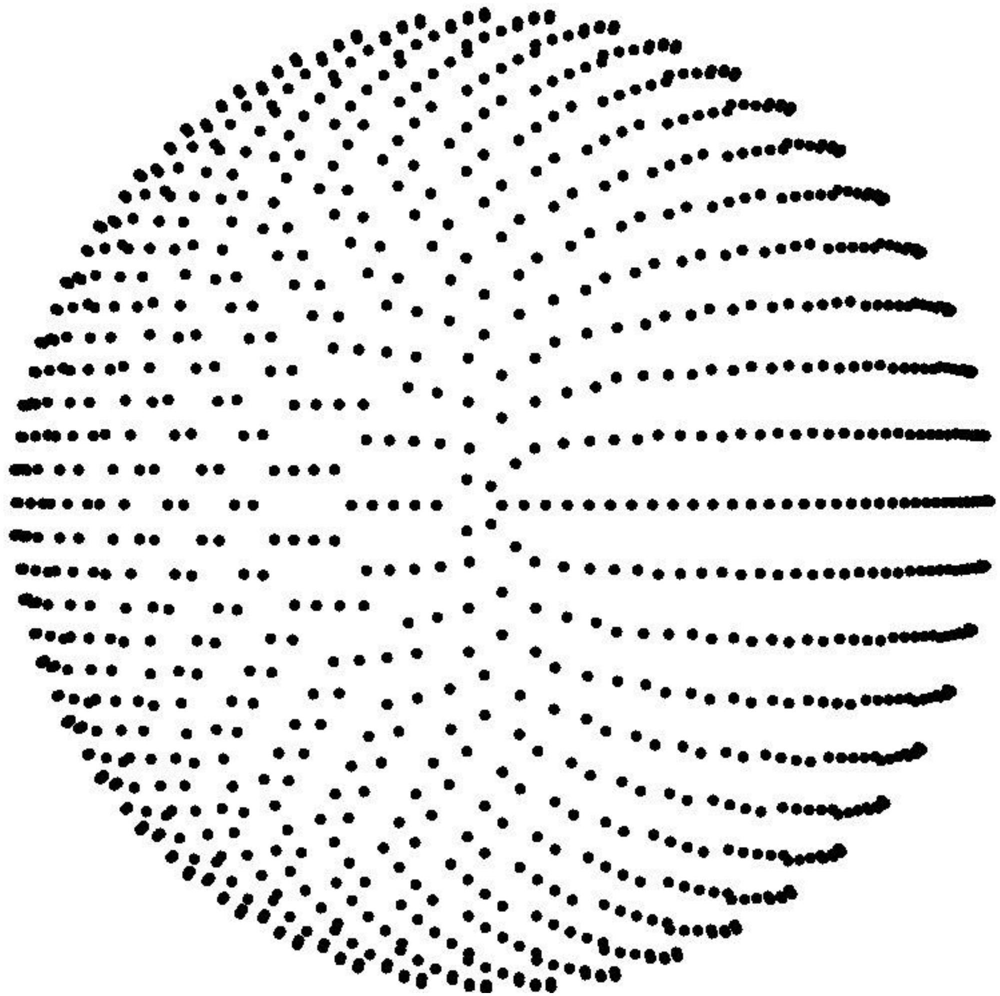
$\varepsilon = 4^o$ $N_\phi = 23$

Choose N_ϕ



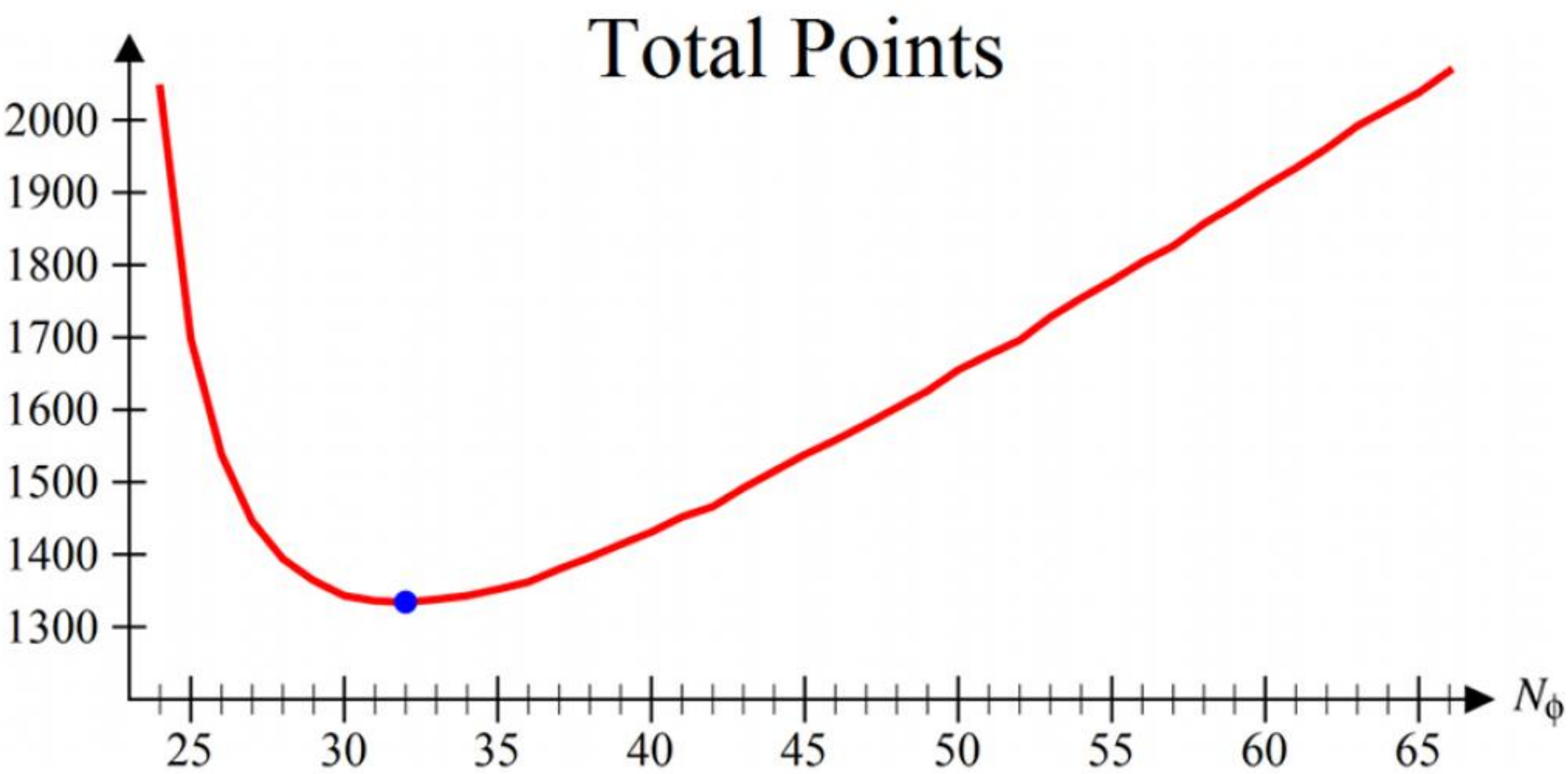
$\varepsilon = 4^o$ $N_\phi = 34$

Choose N_ϕ

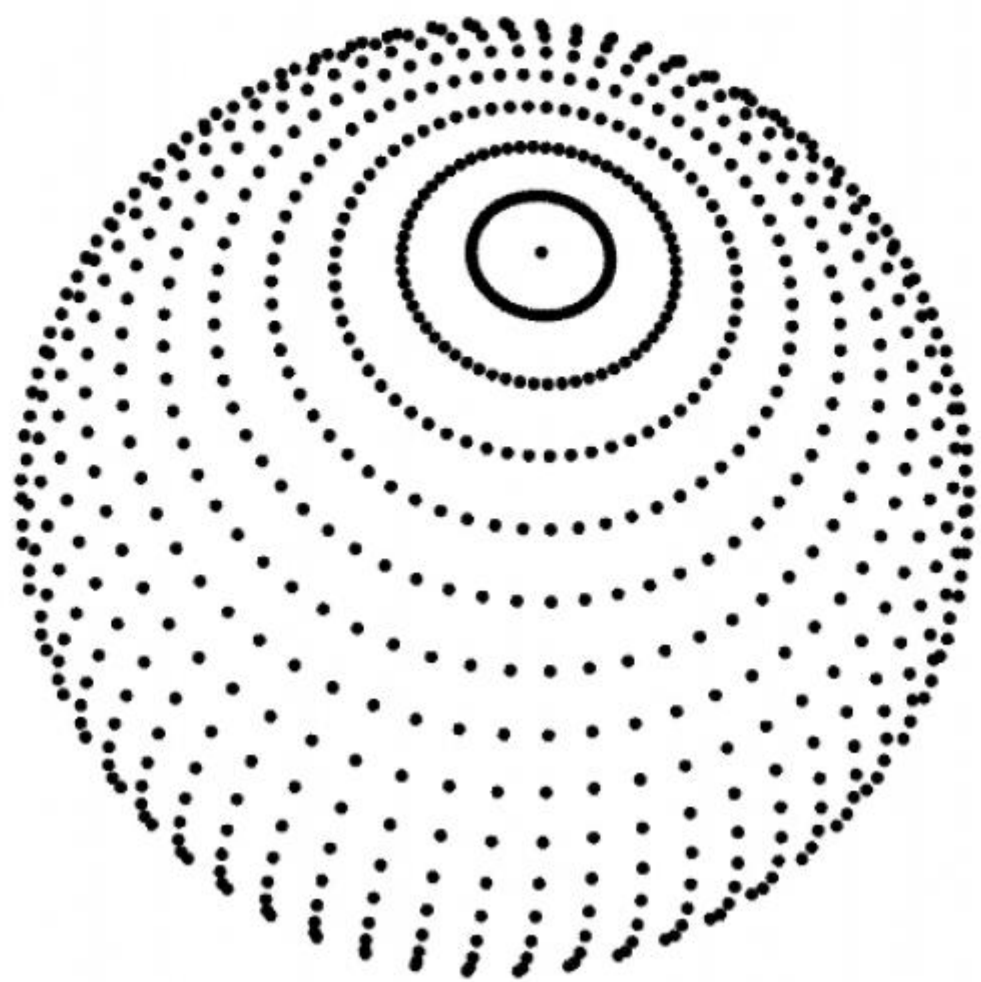


$\varepsilon = 4^o$ $N_\phi = 81$

Minimize Encoding Points

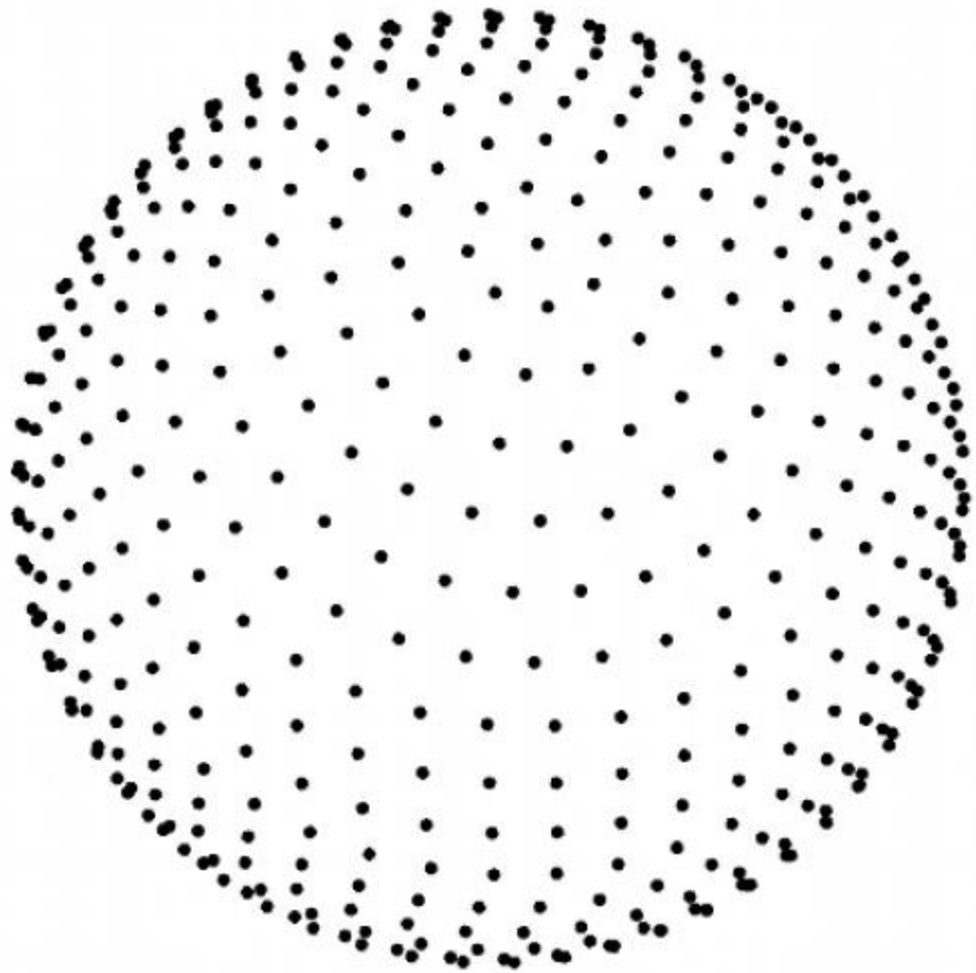


Uniform Encoding Points



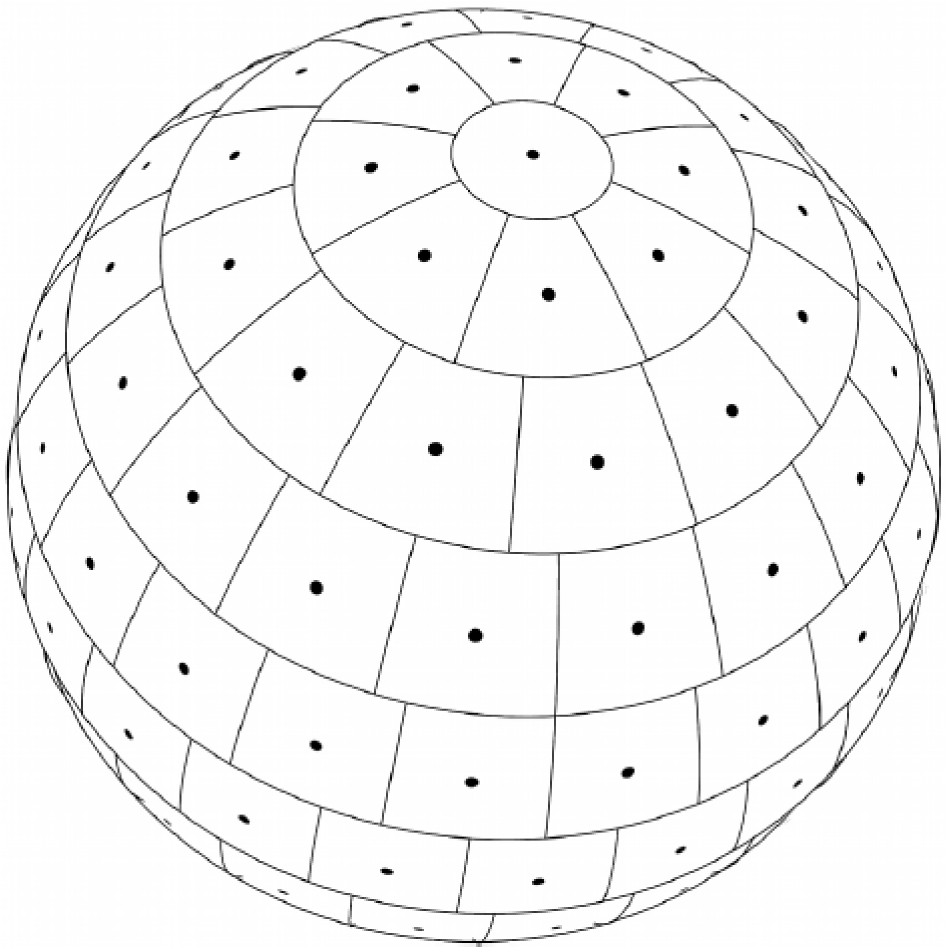
# of symbols = 2112

Optimized Encoding Points

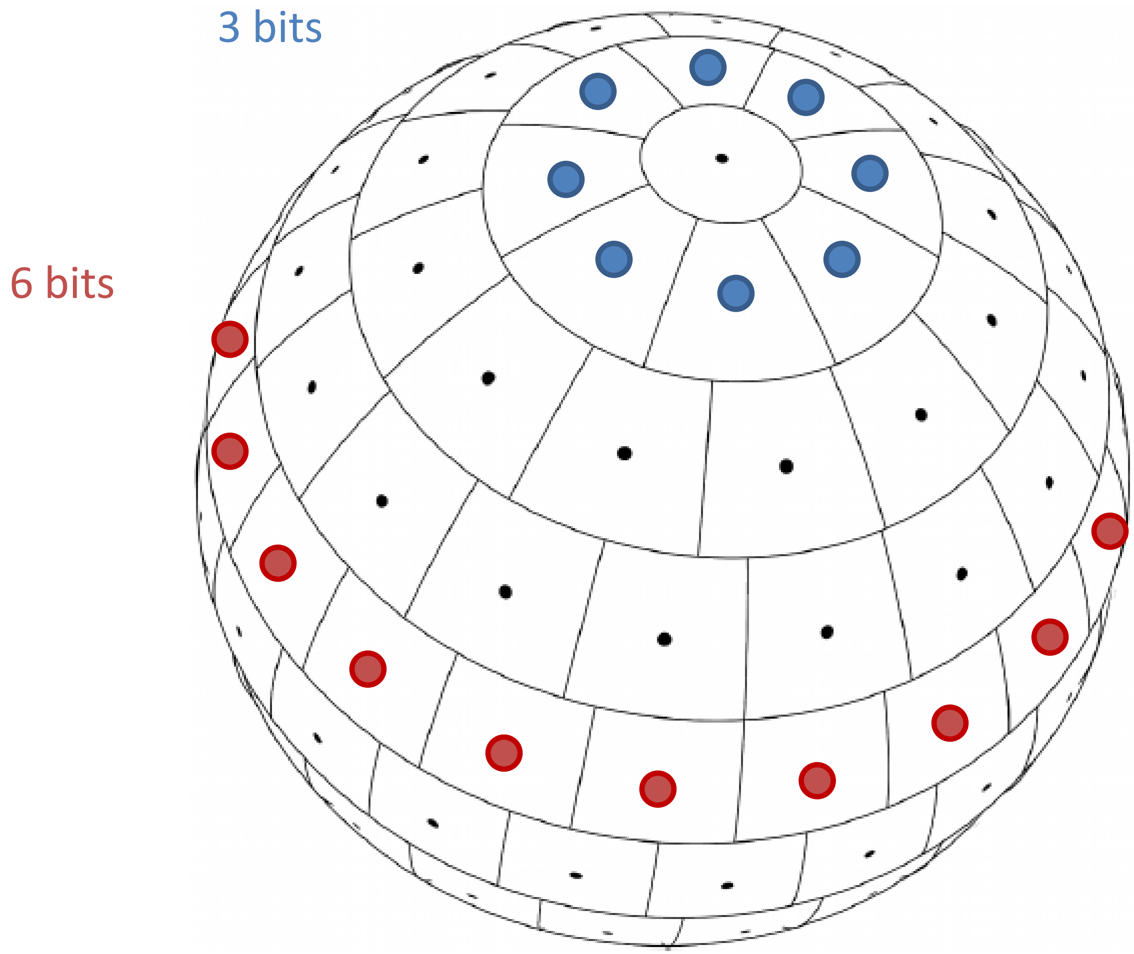


# of symbols = 1334

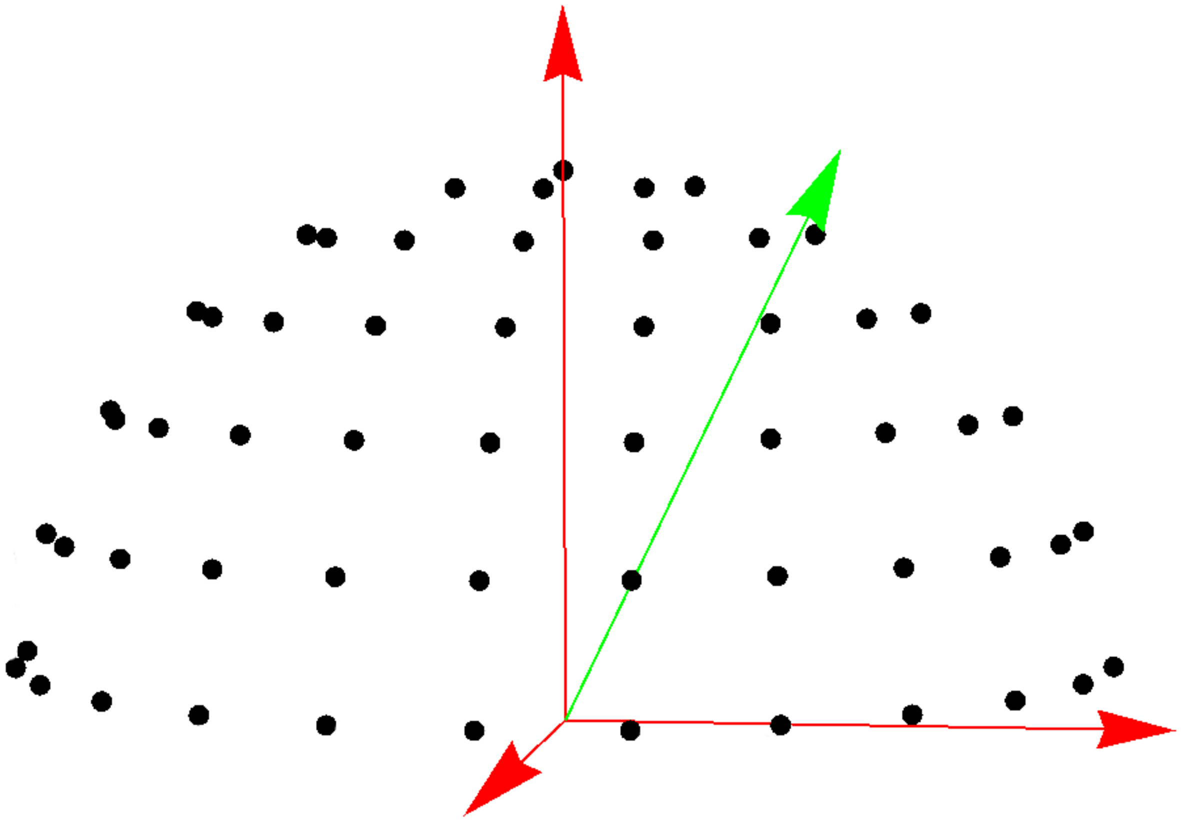
Regions on the Sphere



Variable Bit Encoding

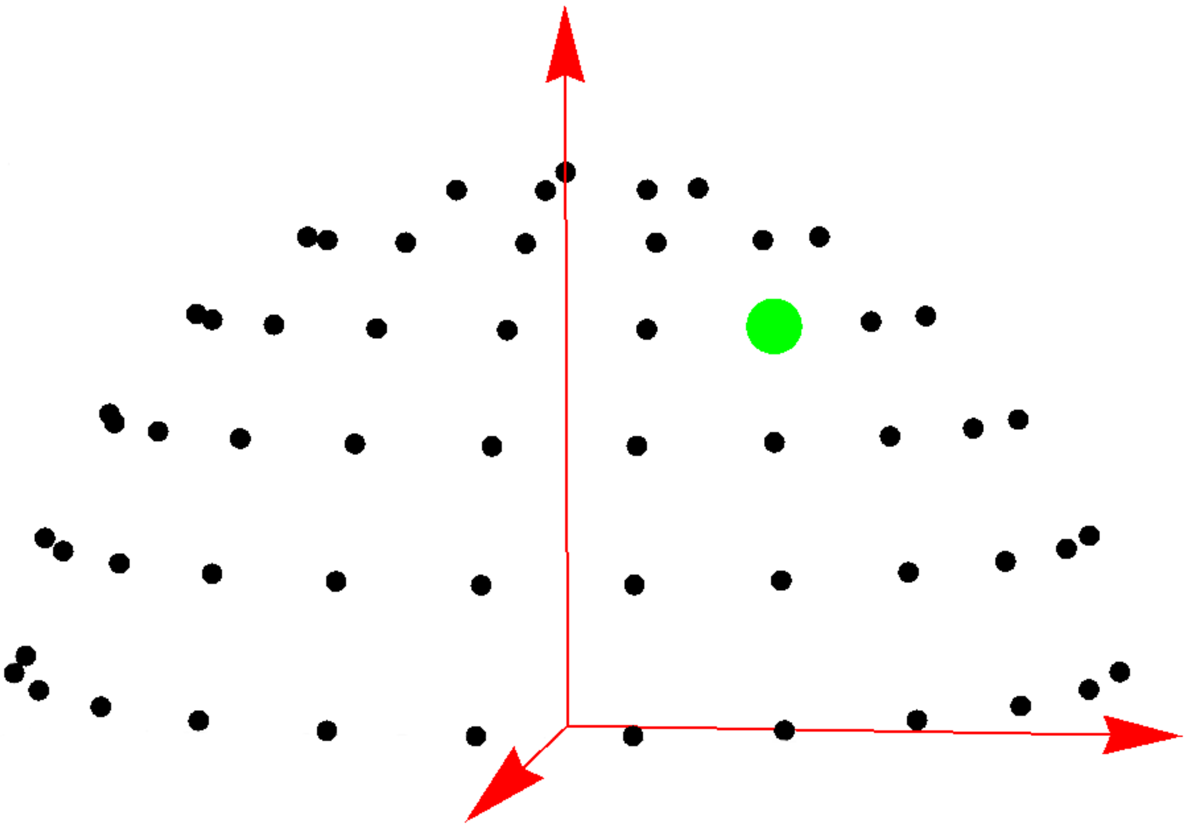


Moving Frame

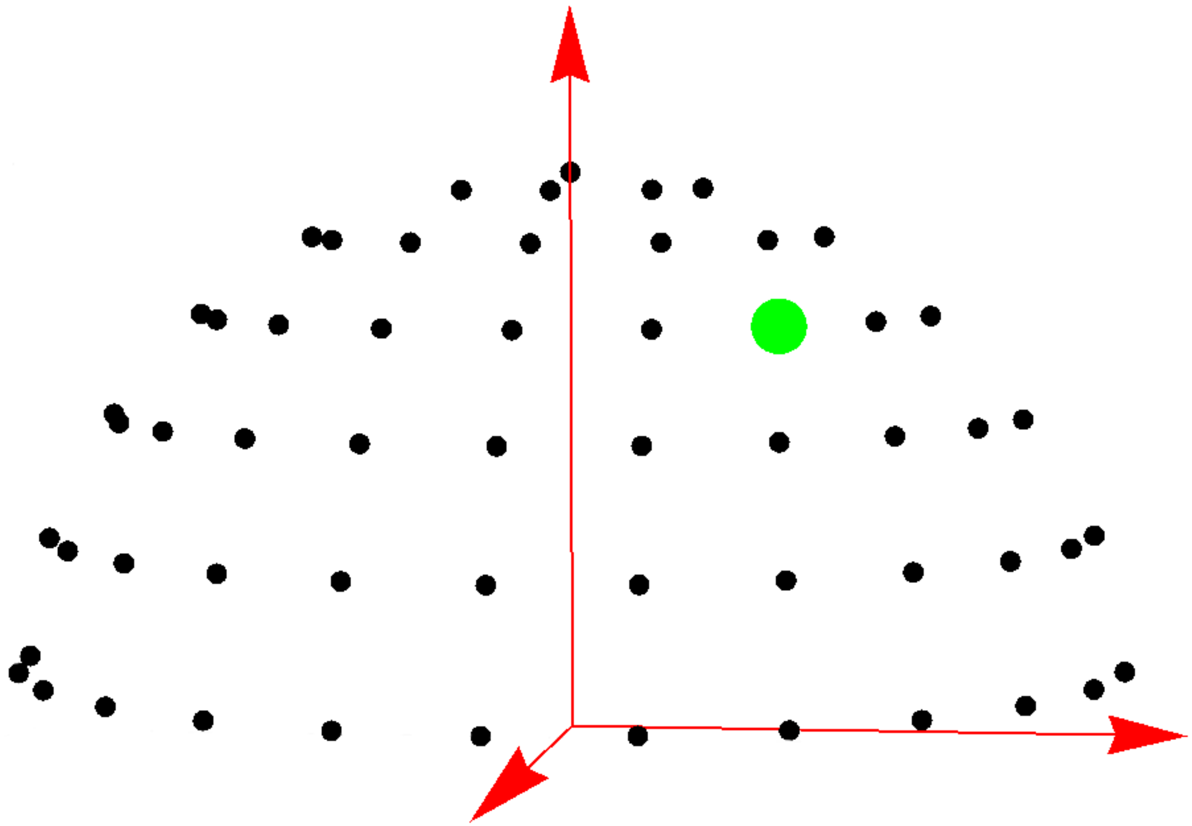


$$(F_x^i, F_y^i, F_z^i)$$

Moving Frame

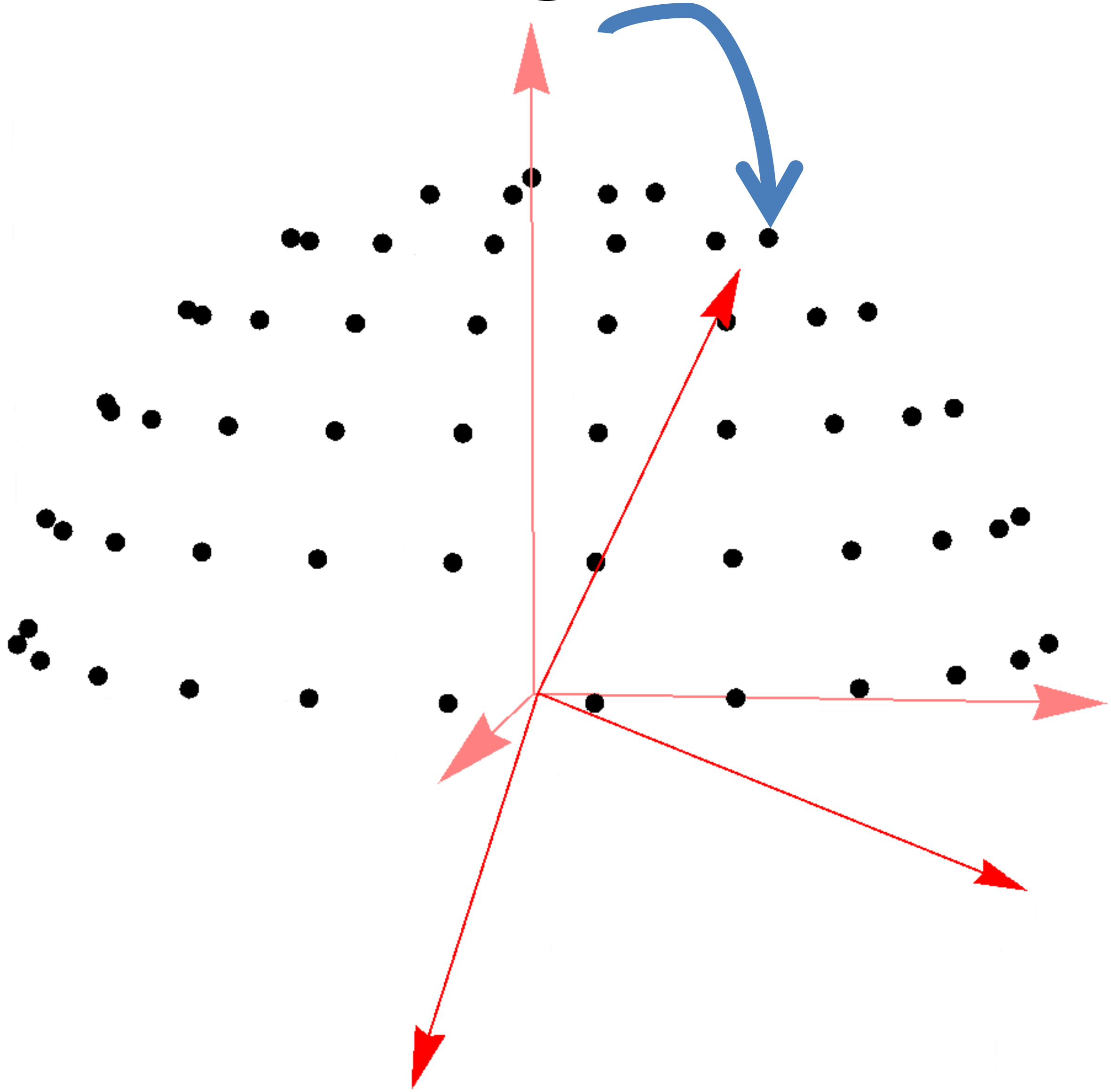


Moving Frame

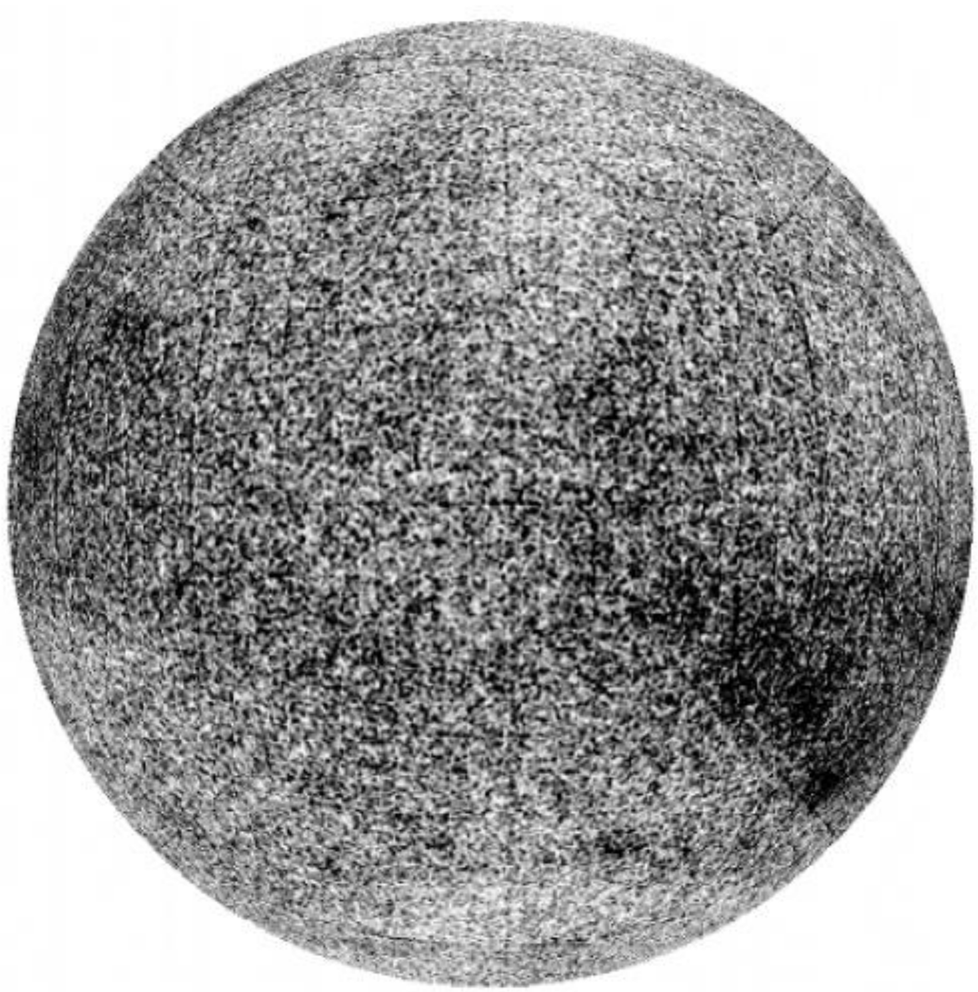


$$
\begin{aligned}
F_z^{i+1} &= \hat{n}^i \\
F_x^{i+1} &= ((F_z^i \cdot \hat{n}^i)\hat{n}^i - F_z^i)\|(F_z^i \cdot \hat{n}^i)\hat{n}^i - F_z^i\|^{-1} \\
F_y^{i+1} &= F_z^{i+1} \times F_x^{i+1}
\end{aligned}
$$

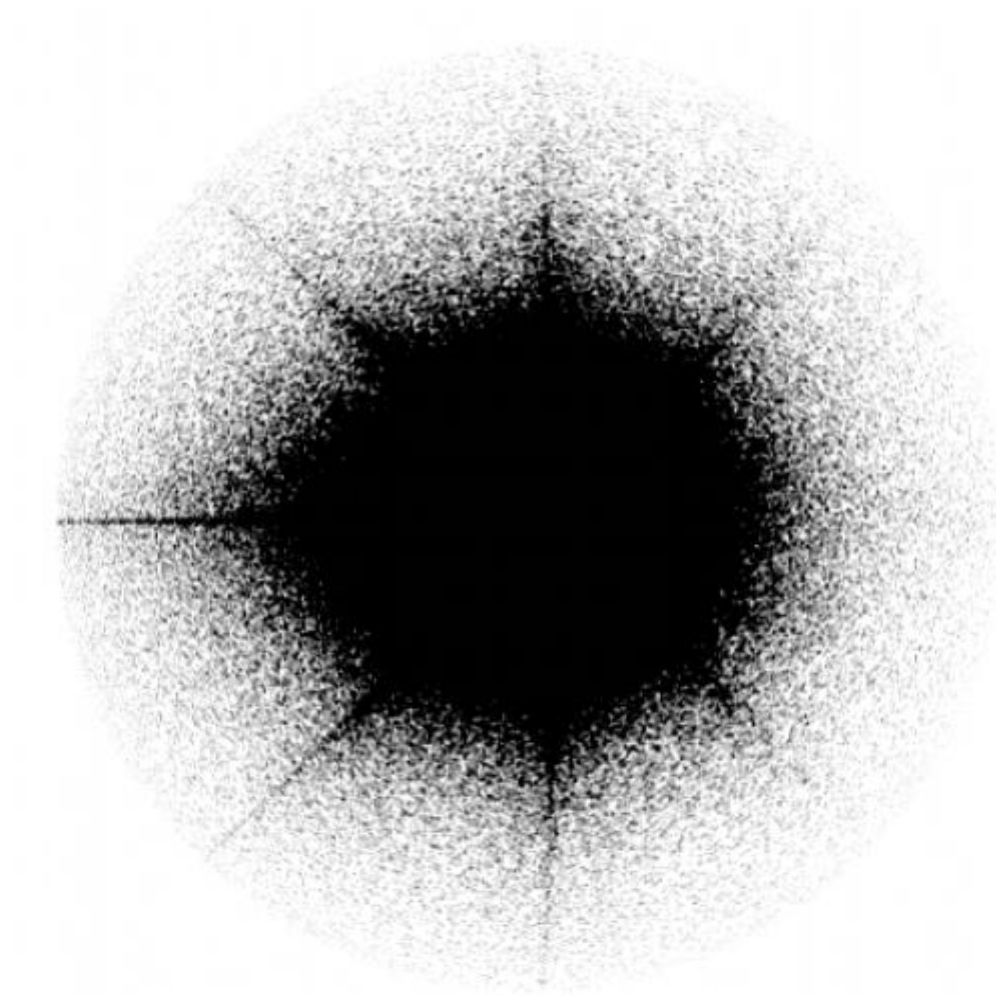
Moving Frame



Moving Frame

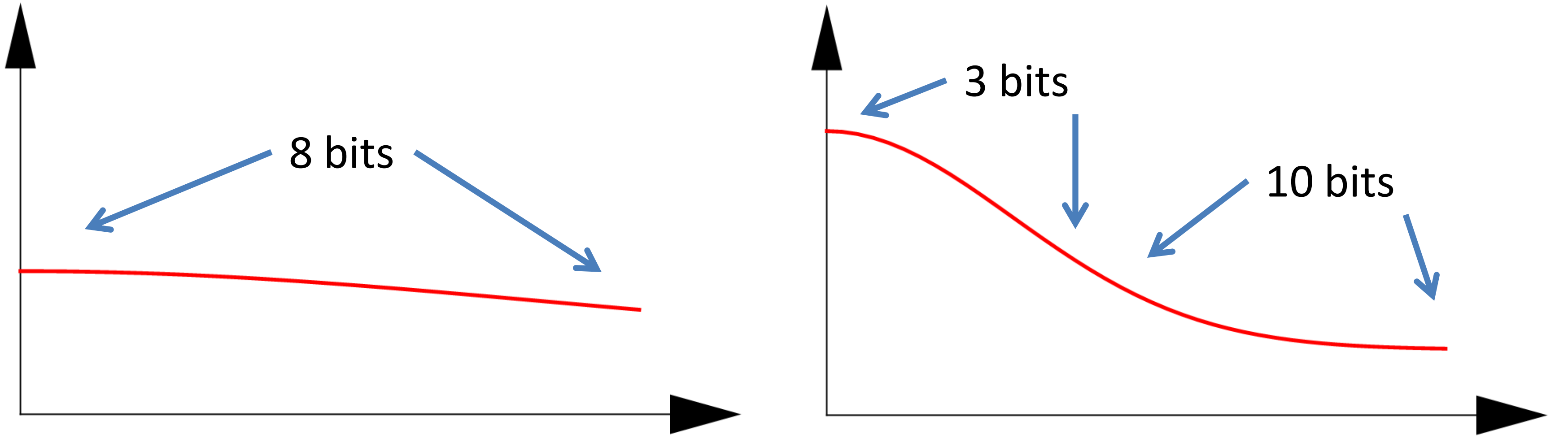


Moving Frame

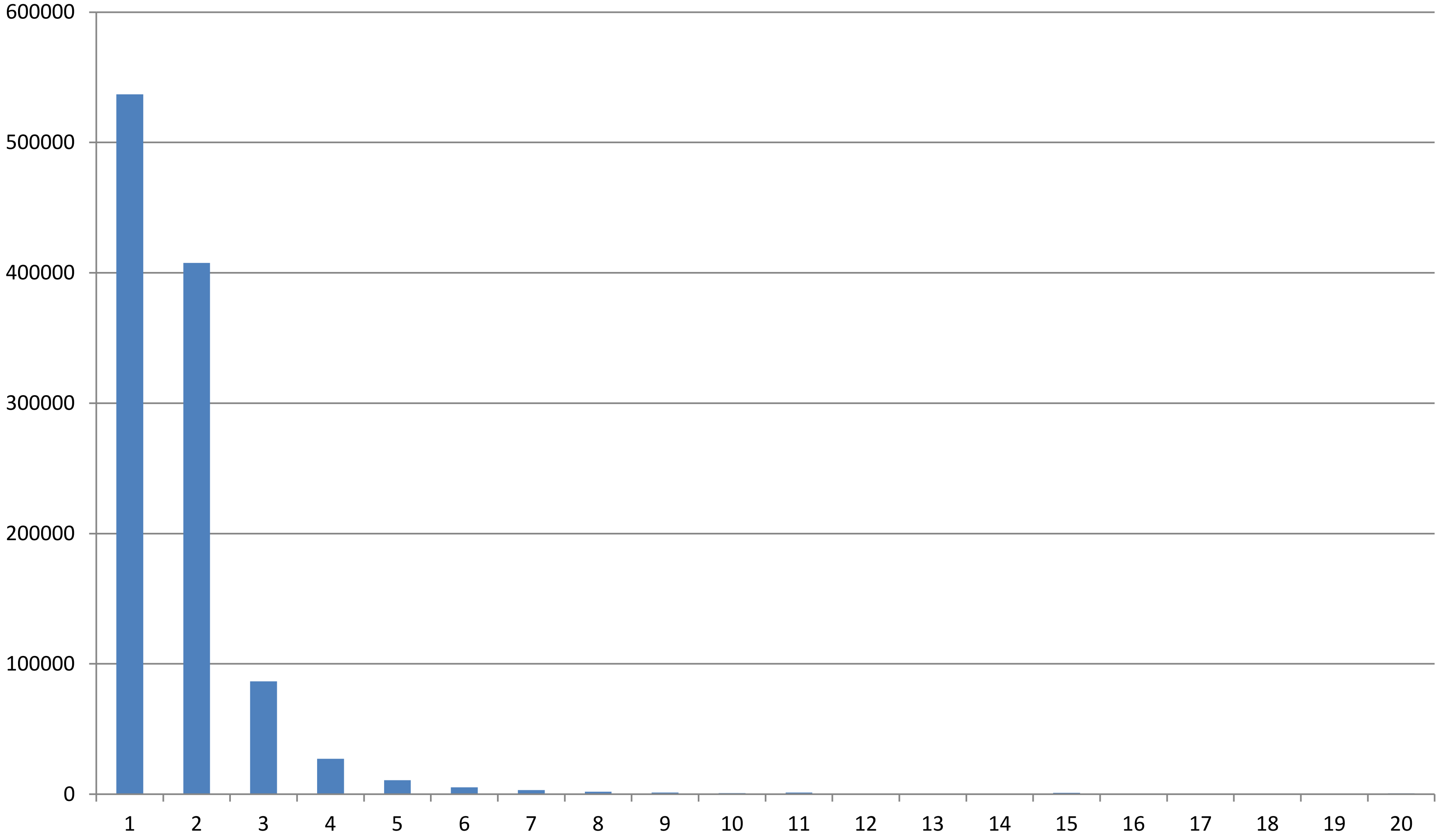


Arithmetic Encoder

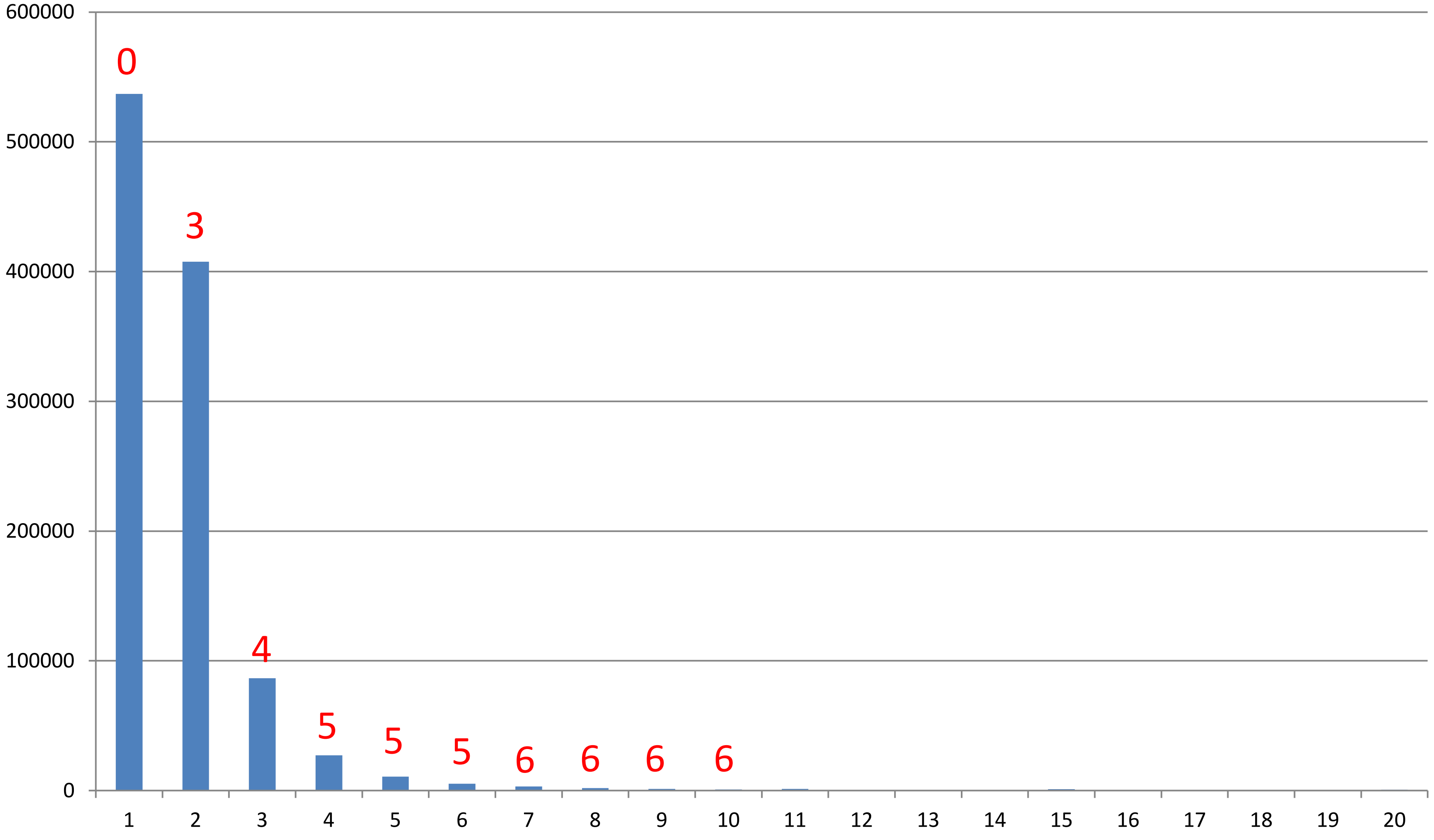
- Adaptive Arithmetic Coding [F. Wheeler 1996]
  - Source code at http://www.cipr.rpi.edu/~wheeler/ac



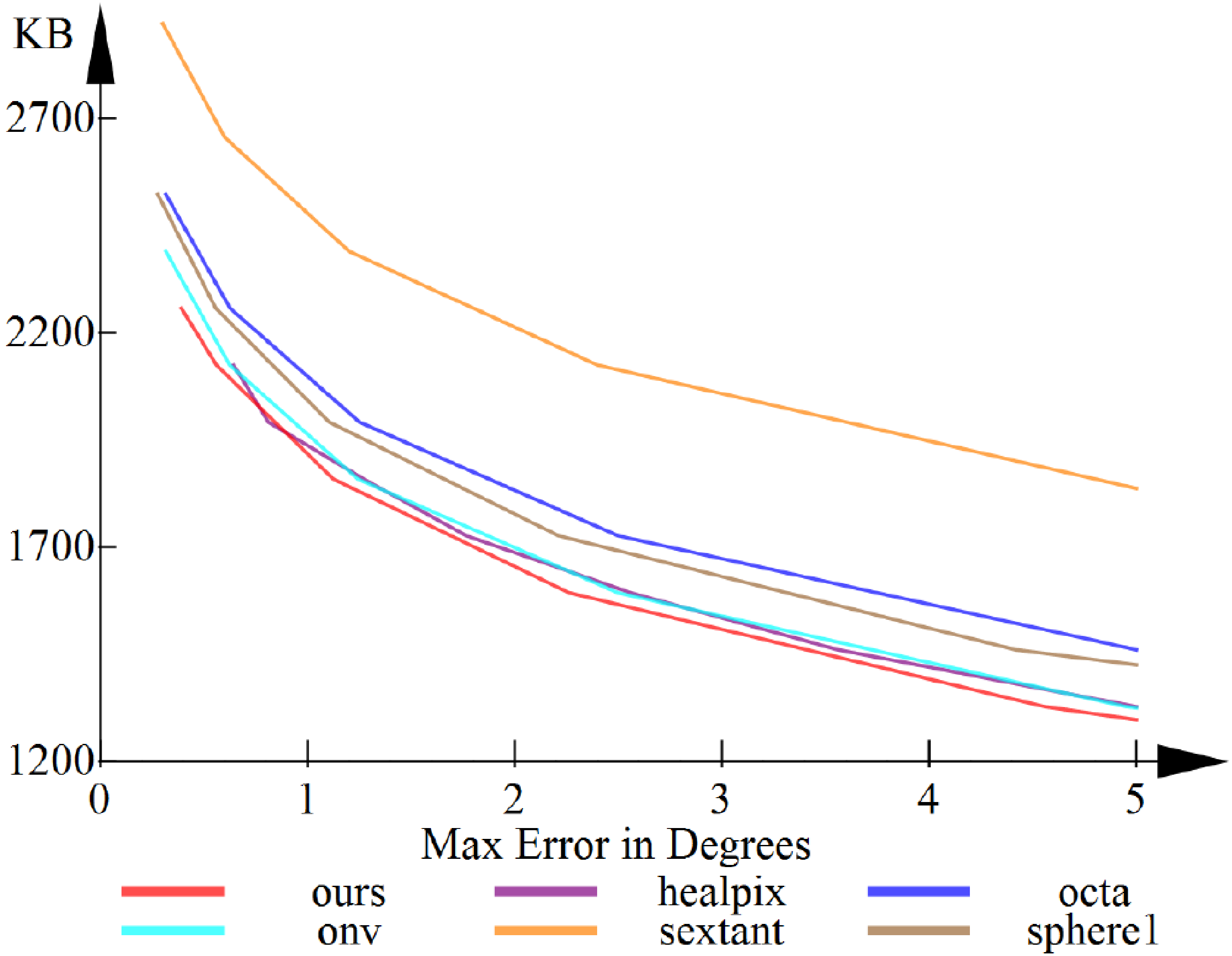
Arithmetic Encoder - Phi



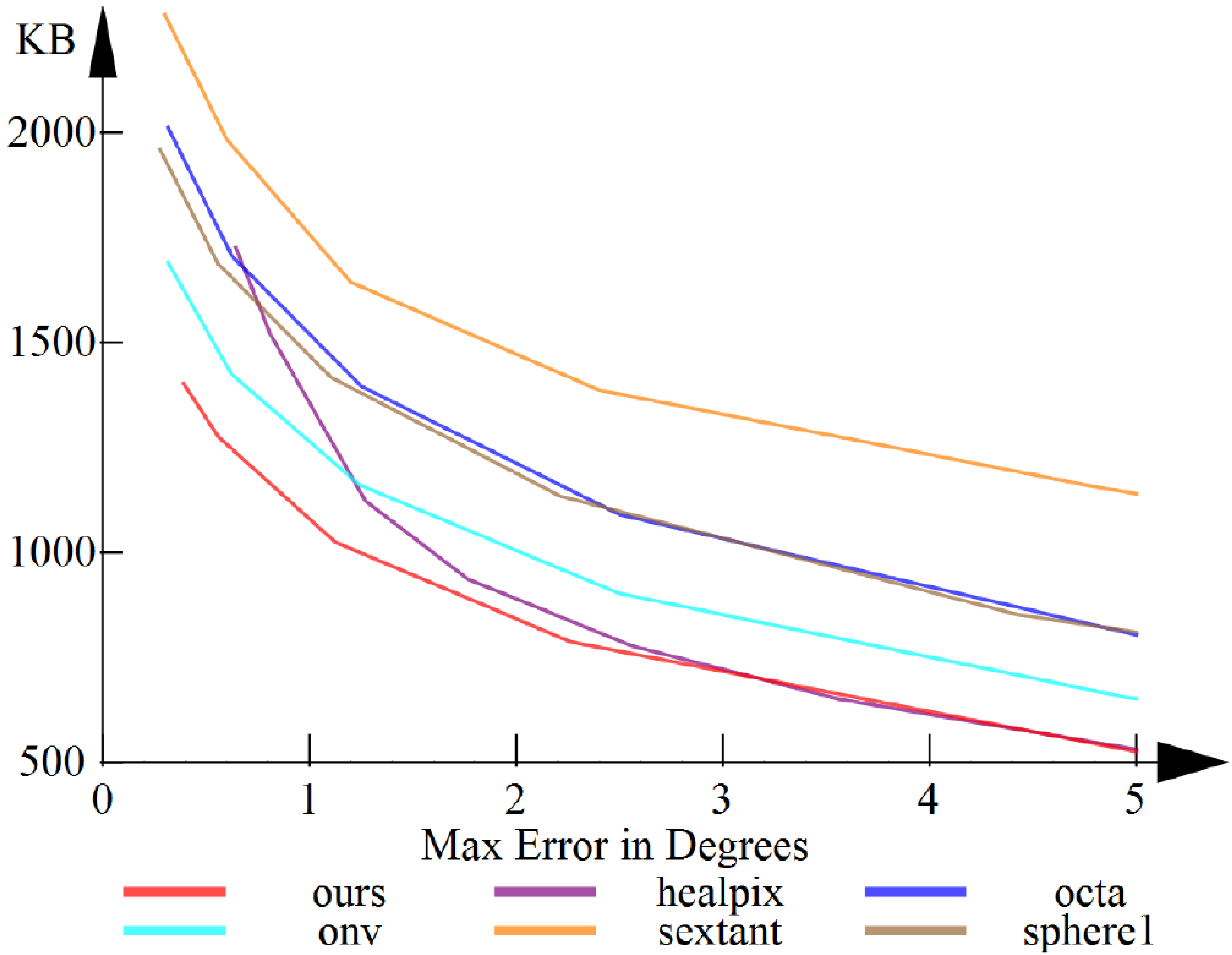
Arithmetic Encoder - Phi



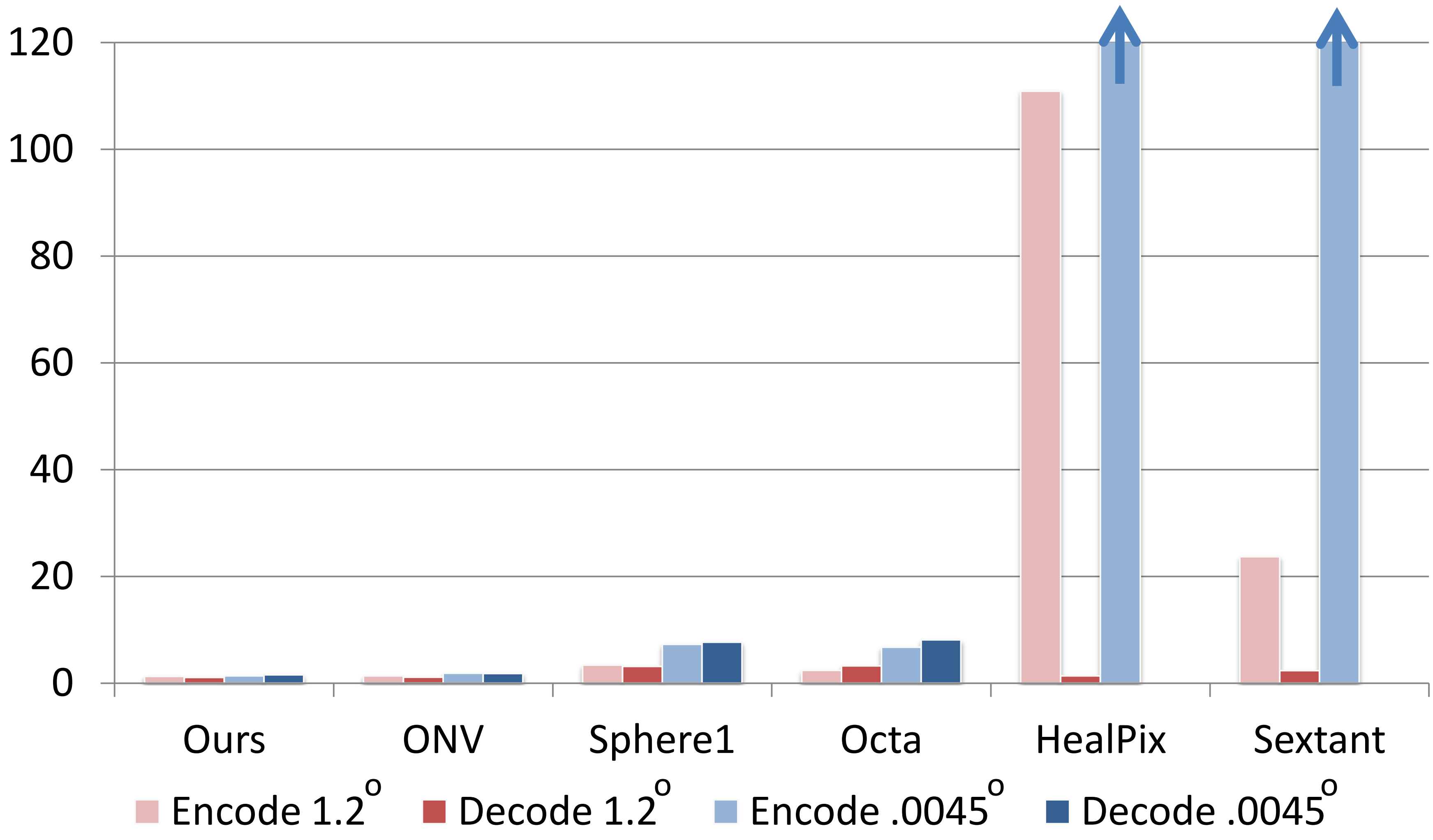
No Moving Frame



Moving Frame



Timings


120
100
80
60
40
20
0
Ours
ONV
Sphere1
Octa
HealPix
Sextant
Encode 1.2°
Decode 1.2°
Encode .0045°
Decode .0045°

Conclusions

- Encoding method that produces the smallest file sizes for a given maximum error
- Constant time encoding and decoding
- Differential encoding frame to improve encoding techniques
- Variable Bit Encoding