Curvature-Based Rejection Sampling for Arbitrary Parameterized Surfaces

Andreas Orthey

June 2023

Uniform sampling of spaces is fundamental to many algorithms like Monte Carlo methods Kroese and Rubinstein [2012] or sampling-based planning Orthey et al. [2024]. However, when the spaces are non-euclidean, it is often difficult to generate uniform samples. Even for simple non-euclidean spaces like the sphere, it is not trivial to sample uniformly Yershova and LaValle [2004]. If spaces become more complicated, it is sometimes almost impossible to find a sampling sequence which is really uniform.

Fortunately, there exists a method to generate uniform sampling sequences for parameterized surfaces, which is called curvature-based rejection sampling Williamson [1987].

1 Curvature-Based Rejection Sampling

The curvature-based rejection sampling method Williamson [1987] is able to generate uniform sampling sequences over arbitrary parameterized surfaces. This method works for parameterized surfaces which are defined by mappings $f : (u, v) \to (x, y, z)$. The norm of the gradient of this mapping gives a measure of the curvature of this space. Note how the curvature is important for sampling:

- If a point of the space has a high curvature (small norm of the gradient), there are sharper bends at this point, and consequently smaller surface area elements. Sampling should be less in those areas.
- If a point of the space has a low curvature (high norm of the gradient), the area is relatively flat, and the surface area element larger. Sampling should be higher at those areas.

This curvature-based rejection sampling method Williamson [1987] exploits this fact by utilizing the norm of the gradient as a proxy for curvature and rejecting points more often in high-curvature areas.

The full algorithm is shown in Alg. 1. It takes as input the coordinate mapping $f : (u, v) \to (x, y, z)$ and the maximum norm gradient on the surface

$n_{\max}$. The output is a sample drawn from the parameterized surface which is uniformly distributed. This works in the following way. First, we loop while we found a sample (Line 1). We sample points (u, v) (Line 2,3), and compute the norm gradient on the surface (Line 4). This gradient is normed over the surface by dividing through the maximum gradient norm (Line 5). We then sample a random number in $[0, 1]$ (Line 6) and check if the normed gradient is above this number. If yes, we accept the sample and return the surface point. If not, we reject the sample and continue.

Algorithm 1 Curvature-Based Rejection Sampling

Input: $f : (u, v) \rightarrow (x, y, z), n_{\max}$
Output: Uniform Sample (x, y, z)
1: **while** True **do**
2: $\quad u \leftarrow \text{UniformSampling}(U)$
3: $\quad v \leftarrow \text{UniformSampling}(V)$
4: $\quad n \leftarrow \text{GradientNorm}(f, u, v)$
5: $\quad n_{\text{normed}} \leftarrow n/n_{\max}$
6: $\quad r \leftarrow \text{UniformSampling}(0, 1)$
7: $\quad$ **if** $n_{\text{normed}} \geq r$ **then**
8: $\quad\quad$ **return** $f(u, v)$
9: $\quad$ **end if**
10: **end while**

2 Practical Considerations

If this method is to be implemented for a new space, there are two important considerations. First, the gradient has to be computed, which is often difficult to do in practice. It is recommended to use a symbolic representation of the coordinate mapping $f : (u, v) \rightarrow (x, y, z)$ and then use tools like SymPy Meurer et al. [2017] to compute a symbolic formula of the derivation of this mapping. This has been used, for example, to compute the gradient on the Klein bottle Orthey et al. [2021]. Second, the maximum norm gradient $n_{\max}$ has to be known for this method to work. This is sometimes available if the spaces are well known like spheres or ellipsoids. However, for general spaces, this value has to be estimated. One method is to sample a large number of points and compute the maximum gradient norm over them. If this should be done online, one can use a burn-in period for the sampler, where the maximum gradient norm is directly computed and constantly updated from the samples drawn. This would give a wrong distribution in the beginning, but would converge more and more to a uniform distribution when the correct maximum norm gradient is known.

References

Dirk P Kroese and Reuven Y Rubinstein. Monte carlo methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1):48–58, 2012.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.

Andreas Orthey, Florian T. Pokorny, and Marc Toussaint. Approximate topological optimization using multi-mode estimation for robot motion planning. RSS Workshop on Geometry and Topology in Robotics, 2021.

Andreas Orthey, Constantinos Chamzas, and Lydia E. Kavraki. Sampling-based motion planning: A comparative review. *Annual Review of Control, Robotics, and Autonomous Systems*, 7(1):285–310, 2024.

JF Williamson. Random selection of points distributed on curved surfaces. *Physics in Medicine & Biology*, 32(10):1311, 1987.

Anna Yershova and Steven M LaValle. Deterministic sampling methods for spheres and so (3). In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3974–3980. IEEE, 2004.