Sampling the GGX Distribution of Visible Normals

Eric Heitz

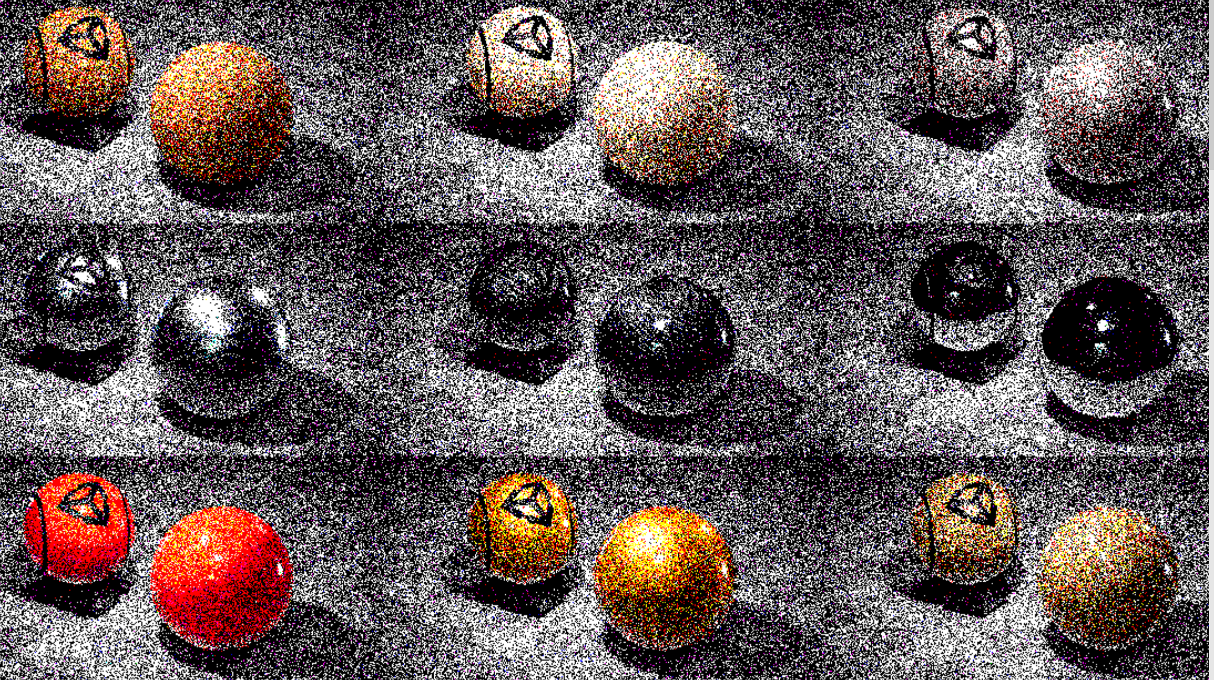
unity

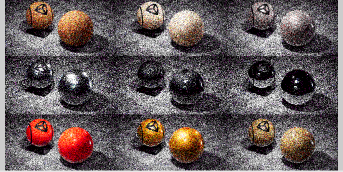
Sampling the GGX Distribution of Visible Normals
Eric Heitz
unity

This talk is focused on importance sampling BRDFs.

One of the most fundamental operations for ray tracing is to choose the next direction of a ray after it hits a surface. To to that, we need to importance sample the BRDF of the surface.

https://blogs.unity3d.com/2019/02/08/we-have-you-covered-with-the-measured-materials-library/

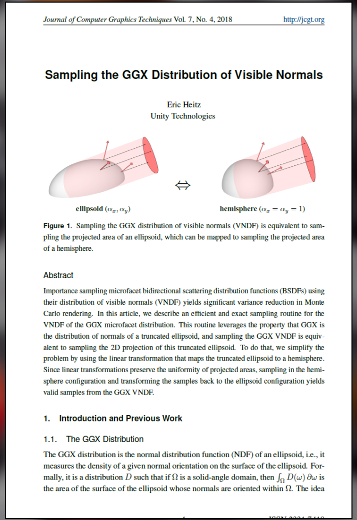




Importance sampling has to be done carefully. If we do it wrong, we will get so much noise that the image will never converge and it will be challenging to denoise.

If we do it right, we will still get noise – after all, it is Monte Carlo rendering – but the amount will be more manageable and we will be able to remove it with denoising or TAA to obtain a decent image.

An optimized sampling routine for today's BRDFs



Journal of Computer Graphics Techniques Vol. 7, No. 4, 2018 http://jcgt.org

Sampling the GGX Distribution of Visible Normals

Eric Heitz
Unity Technologies

ellipsoid (α_x, α_y) ⇔ hemisphere $(\alpha_x = \alpha_y = 1)$

Figure 1. Sampling the GGX distribution of visible normals (VNDF) is equivalent to sampling the projected area of an ellipsoid, which can be mapped to sampling the projected area of a hemisphere.

Abstract

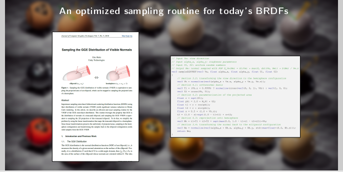
Importance sampling microfacet bidirectional scattering distribution functions (BSDFs) using their distribution of visible normals (VNDF) yields significant variance reduction in Monte Carlo rendering. In this article, we describe an efficient and exact sampling routine for the VNDF of the GGX microfacet distribution. This routine leverages the property that GGX is the distribution of normals of a truncated ellipsoid, and sampling the GGX VNDF is equivalent to sampling the 2D projection of this truncated ellipsoid. To do that, we simplify the problem by using the linear transformation that maps the truncated ellipsoid to a hemisphere. Since linear transformations preserve the uniformity of projected areas, sampling in the hemisphere configuration and transforming the samples back to the ellipsoid configuration yields valid samples from the GGX VNDF.

1. Introduction and Previous Work

1.1. The GGX Distribution

The GGX distribution is the normal distribution function (NDF) of an ellipsoid, i.e., it measures the density of a given normal orientation on the surface of the ellipsoid. Formally, it is a distribution D such that if Ω is a solid-angle domain, then $\int_\Omega D(\omega)\,\partial\omega$ is the area of the surface of the ellipsoid whose normals are oriented within Ω. The idea

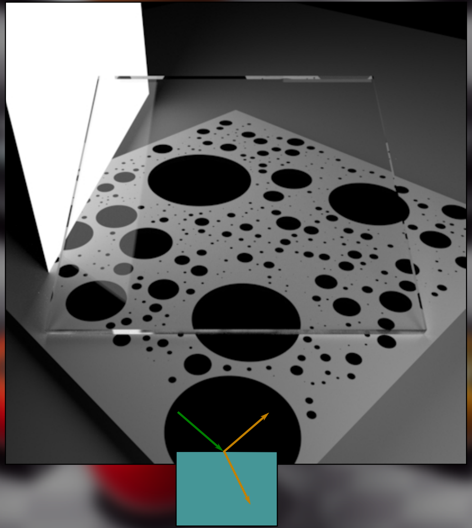
```
// Input Ve: view direction
// Input alpha_x, alpha_y: roughness parameters
// Input U1, U2: uniform random numbers
// Output Ne: normal sampled with PDF D_Ve(Ne) = G1(Ve) * max(0, dot(Ve, Ne)) * D(Ne) / Ve.z
vec3 sampleGGXVNDF(vec3 Ve, float alpha_x, float alpha_y, float U1, float U2)
{
    // Section 3.2: transforming the view direction to the hemisphere configuration
    vec3 Vh = normalize(vec3(alpha_x * Ve.x, alpha_y * Ve.y, Ve.z));
    // Section 4.1: orthonormal basis
    vec3 T1 = (Vh.z < 0.9999) ? normalize(cross(vec3(0, 0, 1), Vh)) : vec3(1, 0, 0);
    vec3 T2 = cross(Vh, T1);
    // Section 4.2: parameterization of the projected area
    float r = sqrt(U1);
    float phi = 2.0 * M_PI * U2;
    float t1 = r * cos(phi);
    float t2 = r * sin(phi);
    float s = 0.5 * (1.0 + Vh.z);
    t2 = (1.0 - s)*sqrt(1.0 - t1*t1) + s*t2;
    // Section 4.3: reprojection onto hemisphere
    vec3 Nh = t1*T1 + t2*T2 + sqrt(max(0.0, 1.0 - t1*t1 - t2*t2))*Vh;
    // Section 3.4: transforming the normal back to the ellipsoid configuration
    vec3 Ne = normalize(vec3(alpha_x * Nh.x, alpha_y * Nh.y, std::max<float>(0.0, Nh.z)));
    return Ne;
}
```



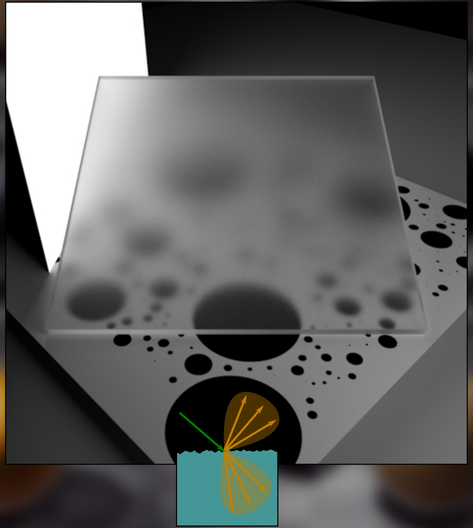
The motivation behind this paper is to share an efficient sampling routine that can be applied to the vast majority of BRDFs that are currently used in production.

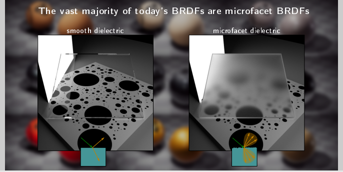
The vast majority of today's BRDFs are microfacet BRDFs

smooth dielectric



microfacet dielectric





This is because most of today's BRDFs are based on microfacet theory.

Microfacet BRDF models are based on the assumption that the glossiness of surfaces is due to their microscopic imperfections. This is modelled by the Cook-Torrance equation, which is the base component of the shading models that you will find in a modern renderer.

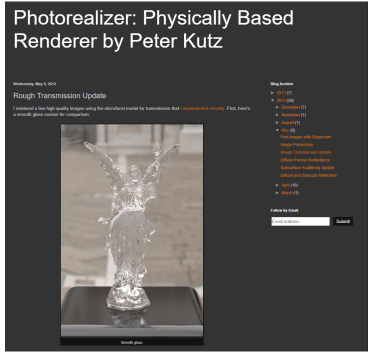
It is even more specific than this. Today's BRDFs are not just based on the generic formulation of the Cook-Torrance equation. They are almost all based on one specific instance of this equation that is commonly referred to as ''Smith GGX''.

Smith GGX

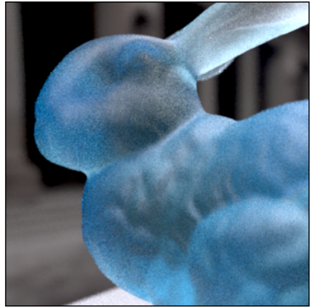
SIGGRAPH Physically Based Shading course 2012 – 2017

That's all the companies that have confessed over the years in the SIGGRAPH Physically Based Shading course that they built their shading system with Smith GGX as a base component.

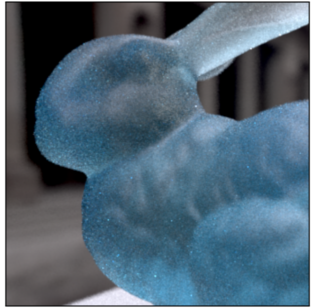
2012 observation: problems with Smith GGX in offline



Beckmann
(previous popular model)

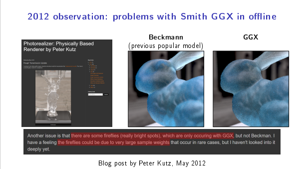


GGX



Another issue is that there are some fireflies (really bright spots), which are only occuring with GGX, but not Beckman. I have a feeling the fireflies could be due to very large sample weights that occur in rare cases, but I haven't looked into it deeply yet.

Blog post by Peter Kutz, May 2012



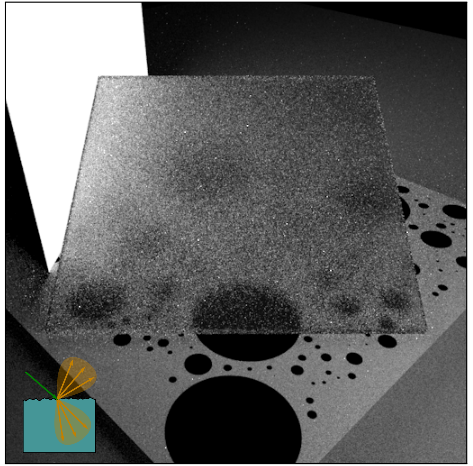
However, Smith GGX was not always so easy to use, especially in offline rendering.

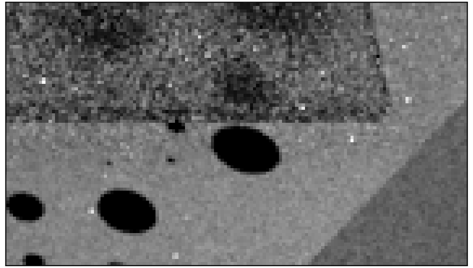
In 2012, I found this blog post by Peter Kutz that commented on using Smith GGX. His observation was that he would get so much noise that the images would not converge and he suspected that it was due to the importance sampling.

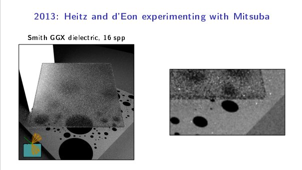
https://photorealizer.blogspot.com/2012/05/rough-transmission-update.html

2013: Heitz and d'Eon experimenting with Mitsuba

Smith GGX dielectric, 16 spp



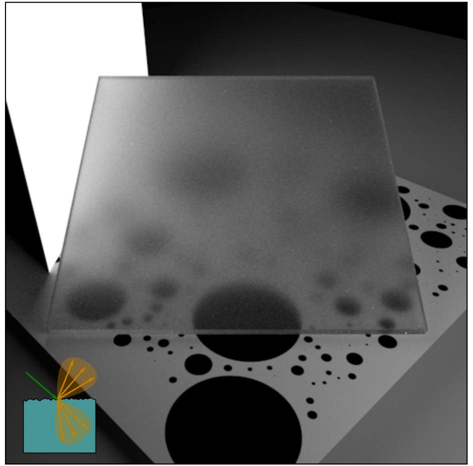


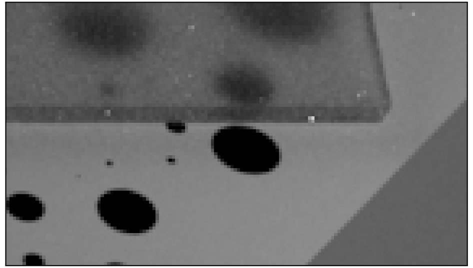


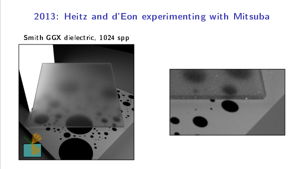
In 2013, Eugene d'Eon and I made the same experiment. We set up a simple scene in Mitsuba (simple objects, simple lighting) and started rendering. We obtained a lot of noise, but in this case we were only using 16 spp so maybe this was OK.

2013: Heitz and d'Eon experimenting with Mitsuba

Smith GGX dielectric, 1024 spp



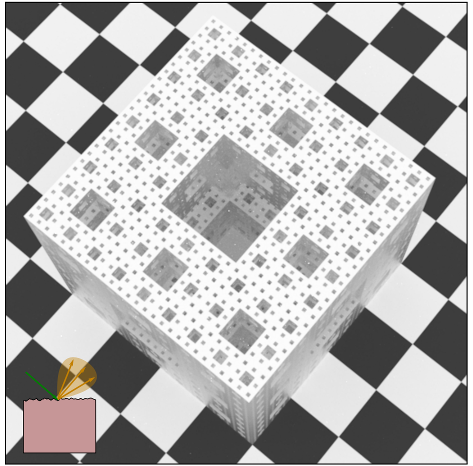


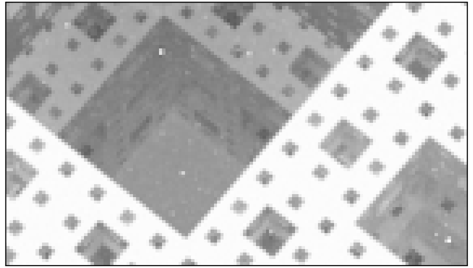


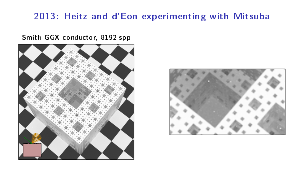
We tried to make the image converge by using 1024 spp but we could still observe some remaining fireflies.

2013: Heitz and d'Eon experimenting with Mitsuba

Smith GGX conductor, 8192 spp





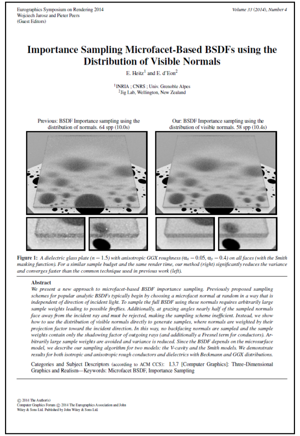


This is another pretty dramatic example: even at 8k spp the image does not converge.

One might argue here that we could use a denoiser or a firefly remover. But let's face it: if even at 8k spp we cannot bring such a simple scene to convergence without cheating then it must mean that something is wrong.

EGSR 2014: introduction of VNDF sampling

[Heitz&d'Eon2014]



Eurographics Symposium on Rendering 2014
Wojciech Jarosz and Pieter Peers
(Guest Editors)

Volume 33 (2014), Number 4

Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals

E. Heitz[1] and E. d'Eon[2]

[1]INRIA ; CNRS ; Univ. Grenoble Alpes
[2]Jig Lab, Wellington, New Zealand

Previous: BSDF Importance sampling using the distribution of normals. 64 spp (10.0s)

Our: BSDF Importance sampling using the distribution of visible normals. 58 spp (10.4s)

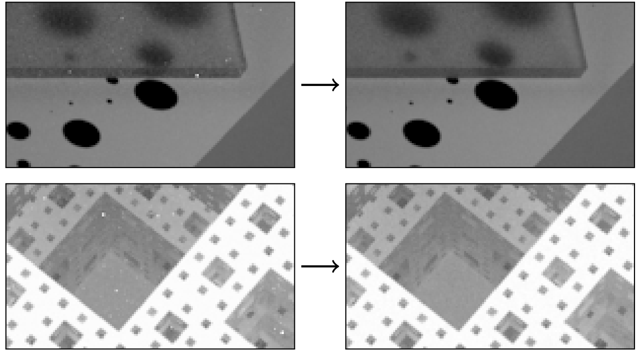
Figure 1: *A dielectric glass plate ($n = 1.5$) with anisotropic GGX roughness ($\alpha_x = 0.05$, $\alpha_y = 0.4$) on all faces (with the Smith masking function). For a similar sample budget and the same render time, our method (right) significantly reduces the variance and converges faster than the common technique used in previous work (left).*

Abstract

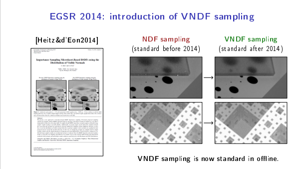
We present a new approach to microfacet-based BSDF importance sampling. Previously proposed sampling schemes for popular analytic BSDFs typically begin by choosing a microfacet normal at random in a way that is independent of direction of incident light. To sample the full BSDF using these normals requires arbitrarily large sample weights leading to possible fireflies. Additionally, at grazing angles nearly half of the sampled normals face away from the incident ray and must be rejected, making the sampling scheme inefficient. Instead, we show how to use the distribution of visible normals directly to generate samples, where normals are weighted by their projection factor toward the incident direction. In this way, no backfacing normals are sampled and the sample weights contain only the shadowing factor of outgoing rays (and additionally a Fresnel term for conductors). Arbitrarily large sample weights are avoided and variance is reduced. Since the BSDF depends on the microsurface model, we describe our sampling algorithm for two models: the V-cavity and the Smith models. We demonstrate results for both isotropic and anisotropic rough conductors and dielectrics with Beckmann and GGX distributions.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Keywords: Microfacet BSDF, Importance Sampling

NDF sampling (standard before 2014) → **VNDF sampling** (standard after 2014)



VNDF sampling is now standard in offline.



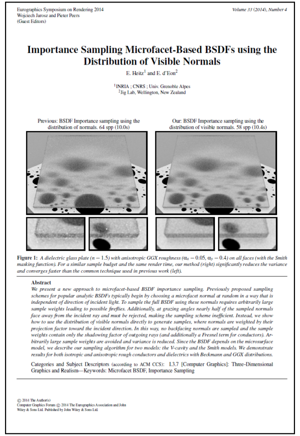
We investigated this problem and arrived at the same conclusion as Peter Kutz' blog post: it is because of the way importance sampling was done.

The standard sampling technique for microfacet BRDFs was NDF sampling. It has been used over more than two decades and is cited in dozens of rendering papers. However, it is far from perfect and its shortcoming (such as persistent fireflies) are amplified in the case of the Smith GGX model.

We published another approach for importance sampling microfacet BRDFs that is called VNDF sampling. By construction, this cannot produce fireflies and is generally less noisy than NDF sampling. It was pretty quickly adopted by the offline rendering industry and today it is fair to say that it has become a standard technique for this problem.

JCGT 2018: VNDF sampling specialized for Smith GGX

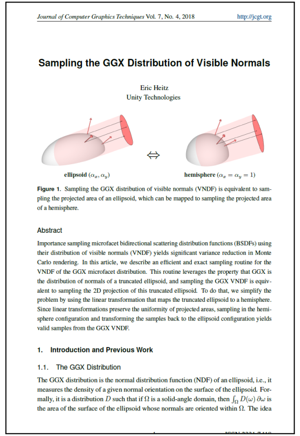
[Heitz&d'Eon2014]

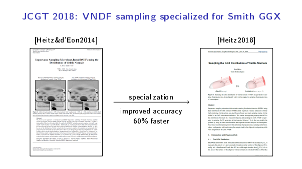


specialization

improved accuracy
60% faster

[Heitz2018]





However, VNDF sampling as published in our EGSR 2014 paper was not specifically meant for Smith GGX. It was a generic solution that could be applied to Smith GGX.

The point of this JCGT 2018 paper is to specialize the concept of VNDF sampling to the specific case of Smith GGX, with the motivation being that since it's the model everyone currently uses in the rendering industry, it just makes sense to provide an optimized implementation.

JCGT 2018: VNDF sampling specialized for Smith GGX

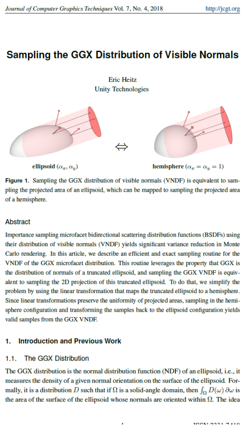
The concepts were already in the air.

Goal: compile these concepts into an accurate and optimized sampling routine.

JCGT spirit: copy-pastable ''gem'' usable by 99% of today's renderers.

```
// Input Ve: view direction
// Input alpha_x, alpha_y: roughness parameters
// Input U1, U2: uniform random numbers
// Output Ne: normal sampled with PDF D_Ve(Ne) = G1(Ve) * max(0, dot(Ve, Ne)) * D(Ne) / Ve.z
vec3 sampleGGXVNDF(vec3 Ve, float alpha_x, float alpha_y, float U1, float U2)
{
    // Section 3.2: transforming the view direction to the hemisphere configuration
    vec3 Vh = normalize(vec3(alpha_x * Ve.x, alpha_y * Ve.y, Ve.z));
    // Section 4.1: orthonormal basis
    vec3 T1 = (Vh.z < 0.9999) ? normalize(cross(vec3(0, 0, 1), Vh)) : vec3(1, 0, 0);
    vec3 T2 = cross(Vh, T1);
    // Section 4.2: parameterization of the projected area
    float r = sqrt(U1);
    float phi = 2.0 * M_PI * U2;
    float t1 = r * cos(phi);
    float t2 = r * sin(phi);
    float s = 0.5 * (1.0 + Vh.z);
    t2 = (1.0 - s)*sqrt(1.0 - t1*t1) + s*t2;
    // Section 4.3: reprojection onto hemisphere
    vec3 Nh = t1*T1 + t2*T2 + sqrt(max(0.0, 1.0 - t1*t1 - t2*t2))*Vh;
    // Section 3.4: transforming the normal back to the ellipsoid configuration
    vec3 Ne = normalize(vec3(alpha_x * Nh.x, alpha_y * Nh.y, std::max<float>(0.0, Nh.z)));
    return Ne;
}
```

[Heitz2018]


Journal of Computer Graphics Techniques Vol. 7, No. 4, 2018 http://jcgt.org

Sampling the GGX Distribution of Visible Normals

Eric Heitz
Unity Technologies

ellipsoid (α_x, α_y) $\Leftrightarrow$ hemisphere $(\alpha_x = \alpha_y = 1)$

Figure 1. Sampling the GGX distribution of visible normals (VNDF) is equivalent to sampling the projected area of an ellipsoid, which can be mapped to sampling the projected area of a hemisphere.

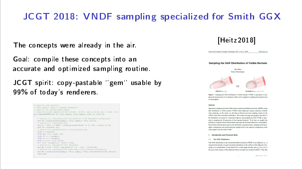
Abstract

Importance sampling microfacet bidirectional scattering distribution functions (BSDFs) using their distribution of visible normals (VNDF) yields significant variance reduction in Monte Carlo rendering. In this article, we describe an efficient and exact sampling routine for the VNDF of the GGX microfacet distribution. This routine leverages the property that GGX is the distribution of normals of a truncated ellipsoid, and sampling the GGX VNDF is equivalent to sampling the 2D projection of this truncated ellipsoid. To do that, we simplify the problem by using the linear transformation that maps the truncated ellipsoid to a hemisphere. Since linear transformations preserve the uniformity of projected areas, sampling in the hemisphere configuration and transforming the samples back to the ellipsoid configuration yields valid samples from the GGX VNDF.

1. Introduction and Previous Work

1.1. The GGX Distribution

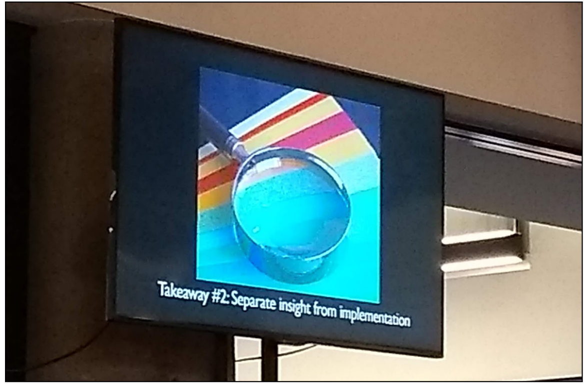
The GGX distribution is the normal distribution function (NDF) of an ellipsoid, i.e., it measures the density of a given normal orientation on the surface of the ellipsoid. Formally, it is a distribution D such that if Ω is a solid-angle domain, then $\int_\Omega D(\omega)\,\partial\omega$ is the area of the surface of the ellipsoid whose normals are oriented within Ω. The idea



Note that to obtain this specialized routine, I did not invent brand new concepts. Most of the ideas were already available in existing papers. The contribution is rather to gather all of these intuitions and use them to obtain this sampling routine and share it such that everyone can just copy-paste it into their own renderer.

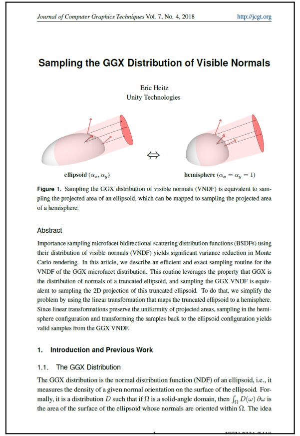
This is also what you would expect from a JCGT paper: a focus on usability and practicability.

JCGT 2018: VNDF sampling specialized for Smith GGX



This talk:
explaining these concepts without any equation.

[Heitz2018]



Journal of Computer Graphics Techniques Vol. 7, No. 4, 2018 http://jcgt.org

Sampling the GGX Distribution of Visible Normals

Eric Heitz
Unity Technologies

ellipsoid (α_x, α_y) $\Leftrightarrow$ **hemisphere** $(\alpha_x = \alpha_y = 1)$

Figure 1. Sampling the GGX distribution of visible normals (VNDF) is equivalent to sampling the projected area of an ellipsoid, which can be mapped to sampling the projected area of a hemisphere.

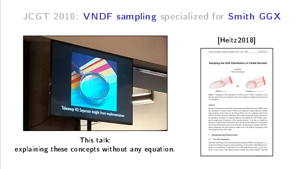
Abstract

Importance sampling microfacet bidirectional scattering distribution functions (BSDFs) using their distribution of visible normals (VNDF) yields significant variance reduction in Monte Carlo rendering. In this article, we describe an efficient and exact sampling routine for the VNDF of the GGX microfacet distribution. This routine leverages the property that GGX is the distribution of normals of a truncated ellipsoid, and sampling the GGX VNDF is equivalent to sampling the 2D projection of this truncated ellipsoid. To do that, we simplify the problem by using the linear transformation that maps the truncated ellipsoid to a hemisphere. Since linear transformations preserve the uniformity of projected areas, sampling in the hemisphere configuration and transforming the samples back to the ellipsoid configuration yields valid samples from the GGX VNDF.

1. **Introduction and Previous Work**

1.1. The GGX Distribution

The GGX distribution is the normal distribution function (NDF) of an ellipsoid, i.e., it measures the density of a given normal orientation on the surface of the ellipsoid. Formally, it is a distribution D such that if Ω is a solid-angle domain, then $\int_\Omega D(\omega)\,\partial\omega$ is the area of the surface of the ellipsoid whose normals are oriented within Ω. The idea



In his keynote from this morning (I3D, 22th March 2019) Steve McAuley recommended gaining insights when reading papers instead of rushing to an implementation.

This is what I would like to share in this presentation: the intuitions behind the concepts of VNDF sampling and Smith GGX. I will try do to that by showing only simple drawings and without using any equations.

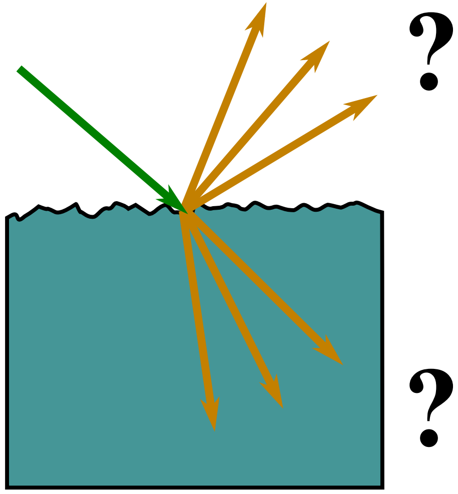
The problem with VNDF Sampling

(previous standard technique for sampling microfacet BRDFs)

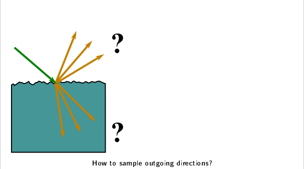
The problem with VNDF Sampling
(previous standard technique for sampling microfacet BRDFs)

Let's start by talking about NDF sampling, the previous standard technique for importance sampling microfacet BRDFs before we introduced VNDF sampling in 2014.

The problem with VNDF Sampling



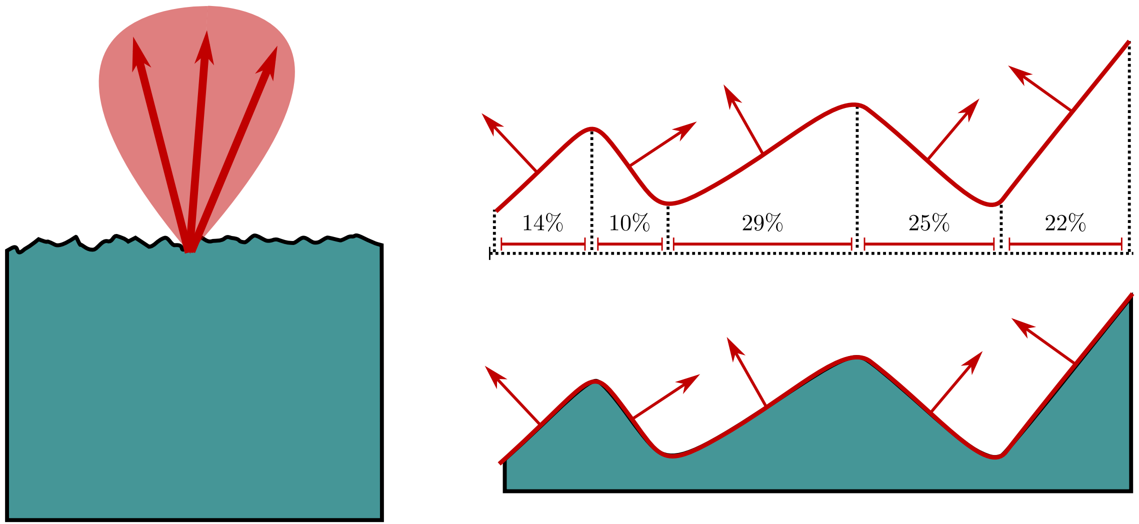
How to sample outgoing directions?



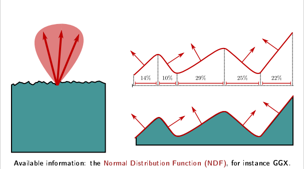
The problem is the following: we have a ray that intersects a surface. Due to the microscopic imperfection of the surface's interface, the ray is going to be deviated from a pure mirror reflection and this is what is going to produce a glossy reflection.

The question is: how do we choose one of the directions randomly?

The problem with VNDF Sampling

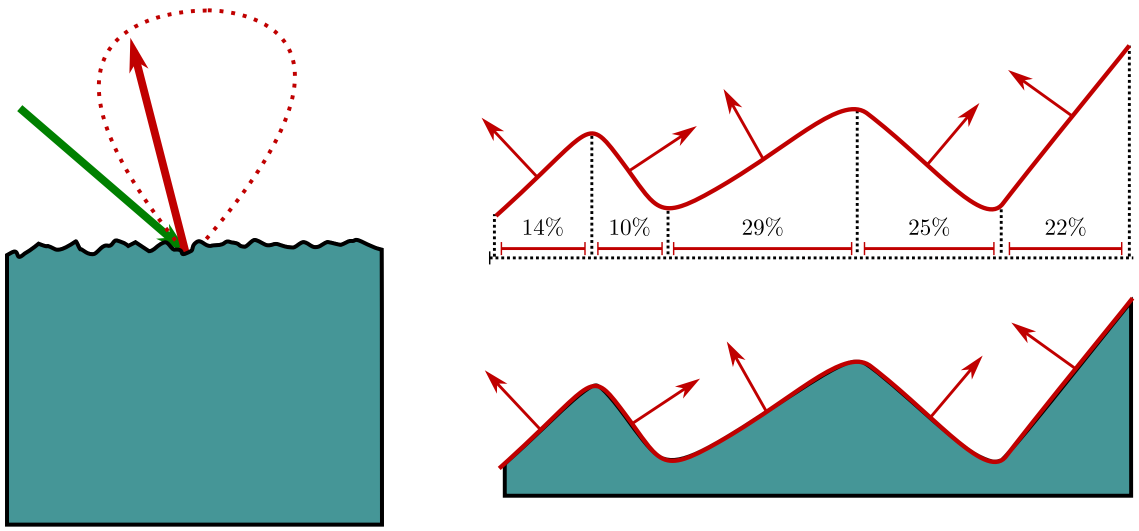


Available information: the Normal Distribution Function (NDF), for instance GGX.

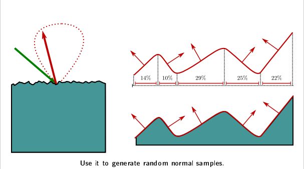


The information that we have at our disposal is the Normal Distribution Function (NDF). It is the statistical representation of the surface's imperfections via the surface normal that they produce.

The problem with VNDF Sampling

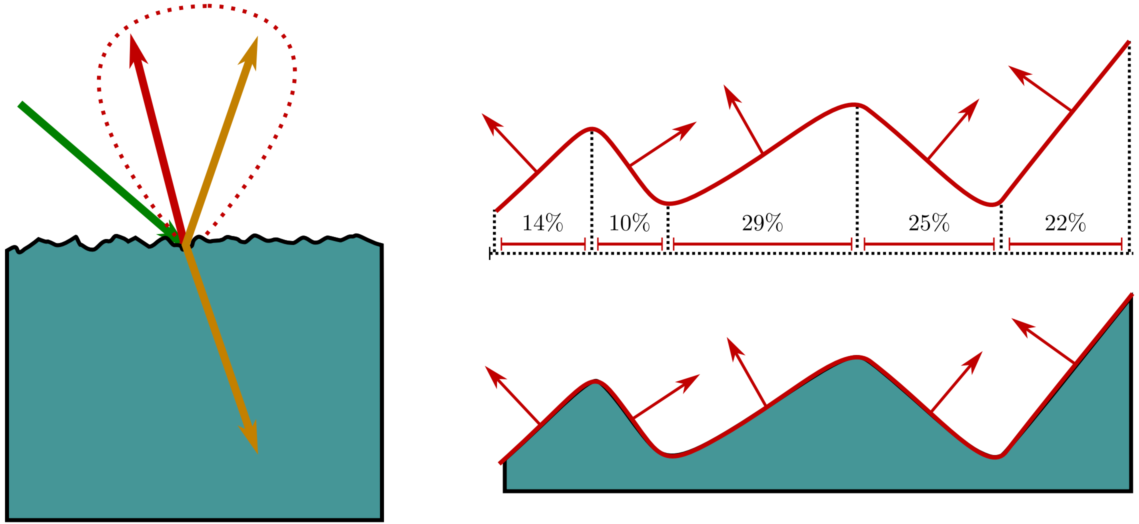


Use it to generate random normal samples.

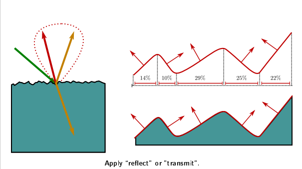


NDF sampling means generating one normal from the NDF randomly...

The problem with VNDF Sampling

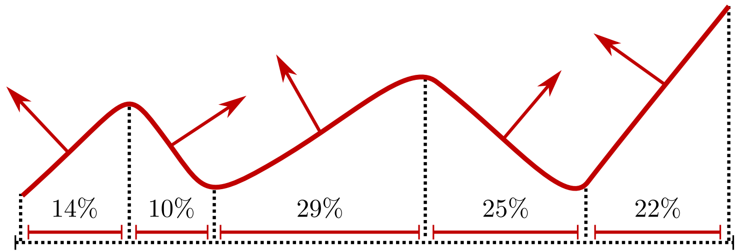


Apply “reflect” or ”transmit”.

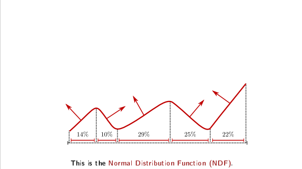


…and using it to reflect (or transmit) the incident ray. This is how people used to obtain random outgoing directions for the incident ray.

The problem with VNDF Sampling

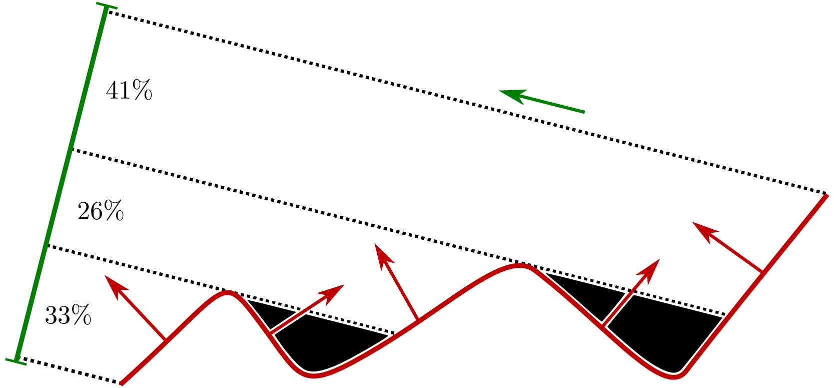


This is the Normal Distribution Function (NDF).

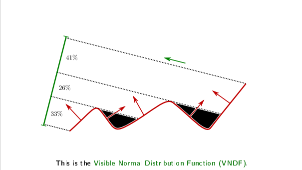


So this is the NDF. Basically, NDF sampling means generating a normal according to the NDF and making this normal interact with the incident ray.

The problem with VNDF Sampling

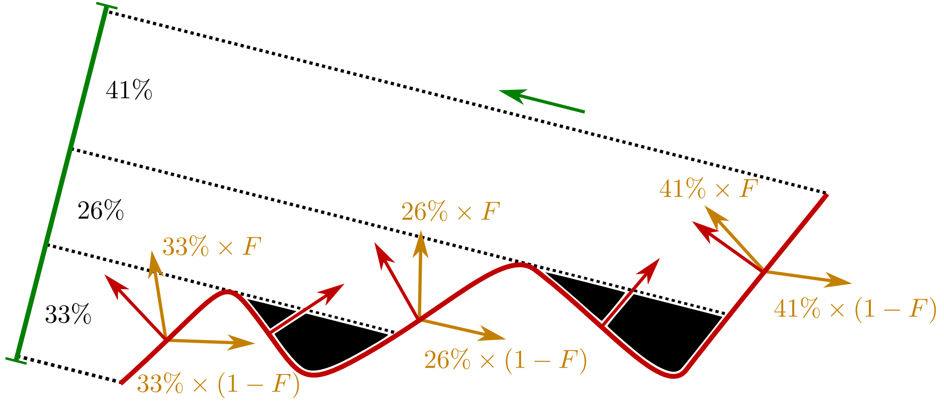


This is the Visible Normal Distribution Function (VNDF).

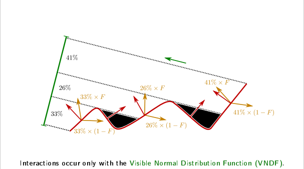


However, physically, the incident ray can only interact with normals that are visible to it. Normals that are shadowed or backfacing will never be intersected by the incident ray. These normals are modelled by the Visible Normal Distribution Function (VNDF).

The problem with VNDF Sampling

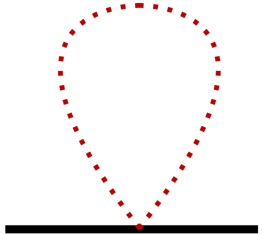


Interactions occur only with the Visible Normal Distribution Function (VNDF).

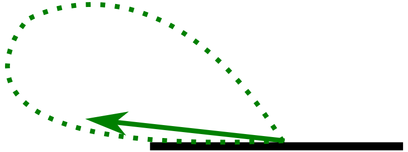


Only the normals represented by the VNDF can reflect or transmit the incident ray.

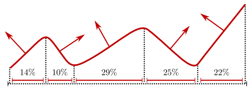
The problem with VNDF Sampling

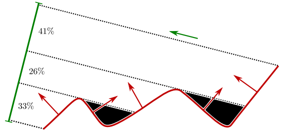


NDF
(generates the samples)

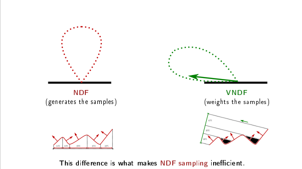


VNDF
(weights the samples)



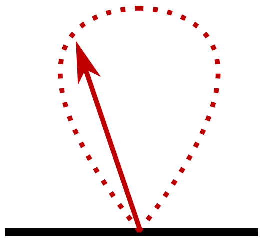


This difference is what makes NDF sampling inefficient.

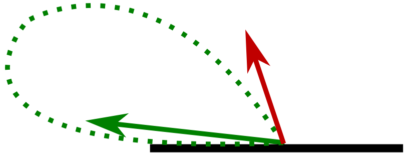


With NDF sampling, the normal samples are generated by the NDF but they are weighted by the VNDF. It is the difference between the NDF and the VNDF that makes this sampling technique inefficient.

The problem with VNDF Sampling



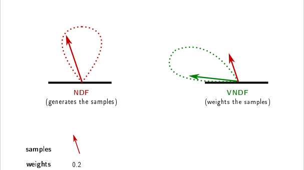
NDF
(generates the samples)



VNDF
(weights the samples)

samples 

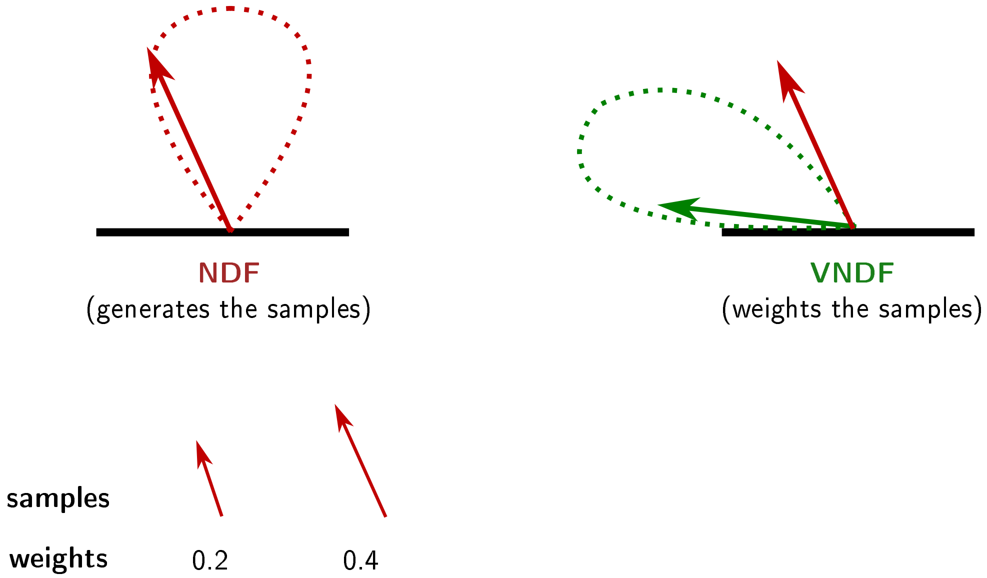
weights 0.2

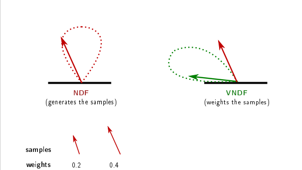


Let's do a sampling simulation to observe the problem.

Imagine that the NDF has generated this sample and proposes it to the VNDF for weighting. The NDF is pretty happy with it, but the VNDF does not like it so much because it not located within the main part of its lobe. Because of this, the VNDF is going to assign it a small weight: 0.2.

The problem with VNDF Sampling

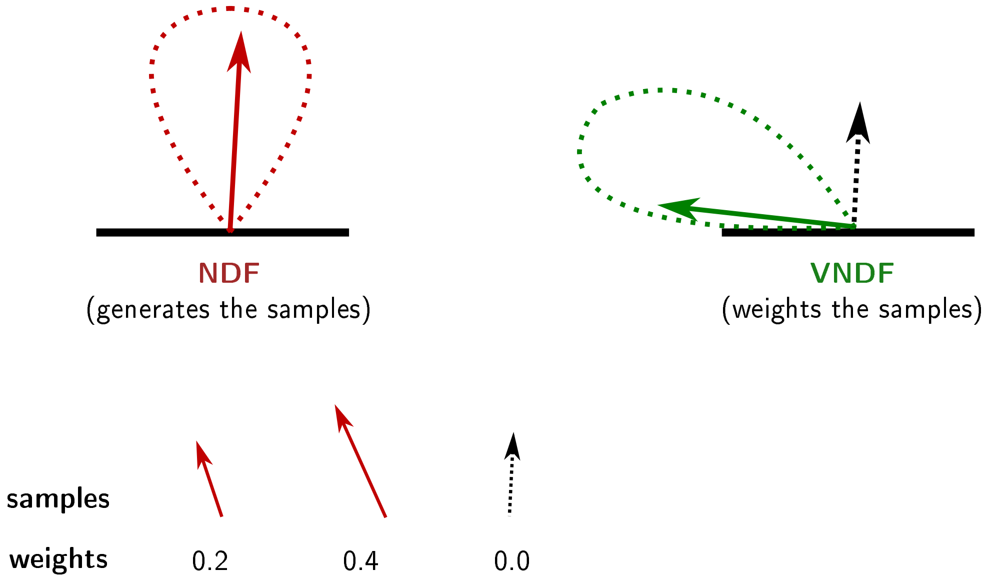


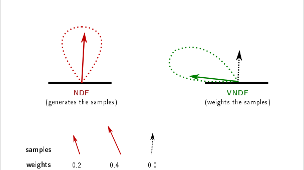


Let's try another sample.

This one is a little bit more oriented towards the incident ray, so the VNDF will give it a higher weight, but it's still not really well oriented for the VNDF: 0.4.

The problem with VNDF Sampling

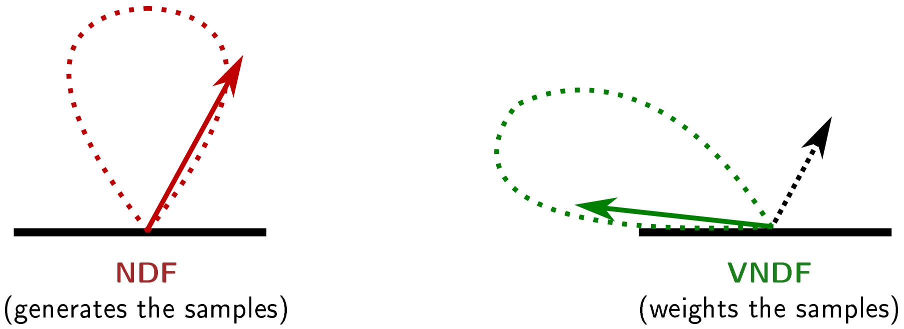


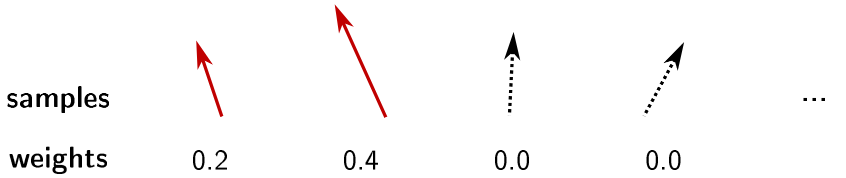


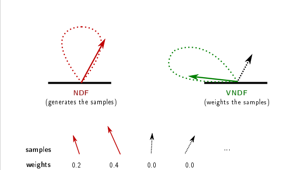
Let's try another sample.

This one is oriented more than 90 degrees from the incident ray direction. This is typically a normal that the incident ray could never intersect, so the VNDF is going to reject it completely: 0.

The problem with VNDF Sampling



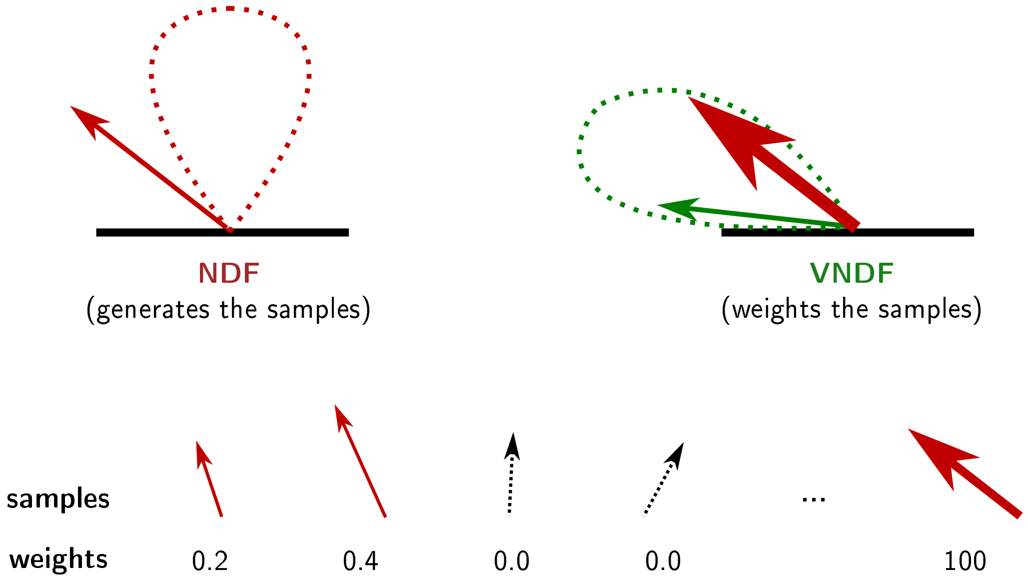


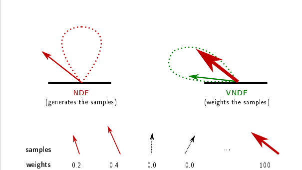


This is another backfacing sample: 0 as well.

And this goes on and on: in this configuration, most of the samples generated by the NDF will be assigned a low weight or even plainly rejected by the VNDF.

The problem with VNDF Sampling



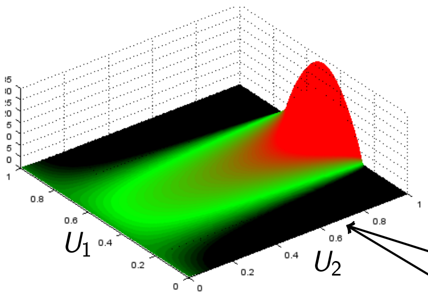


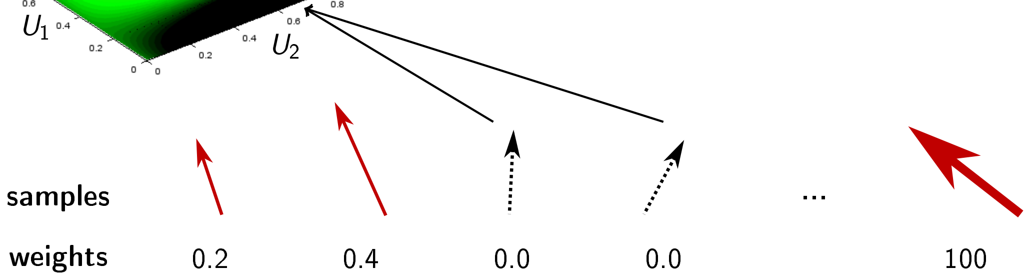
However, sometimes the NDF will finally manage to generate a sample that the VNDF likes. Since this is very rare and since the VNDF has been waiting for such a sample for a long time, it is going to assign it a crazy high weight to compensate for all the very low weights given before.

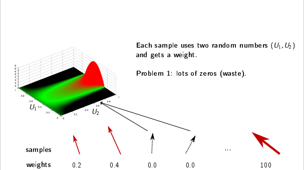
The problem with VNDF Sampling

Each sample uses two random numbers (U_1, U_2) and gets a weight.

Problem 1: lots of zeros (waste).



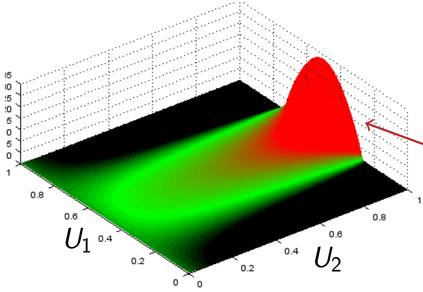




Another view on what happens here is to look at the sampling space. To generate a sample, we typically use two random numbers (U_1, U_2). This 3D plot represents the weights assigned to the samples as a function of the random numbers used to generate them.

The first problem we can observe is that a large portion of the sampling space is filled with zeros (shown in black). This represents all of the samples that are backfacing and thus will get a weight of 0. These samples are a waste of computational power since they don't contribute to the result.

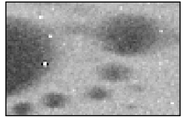
The problem with VNDF Sampling

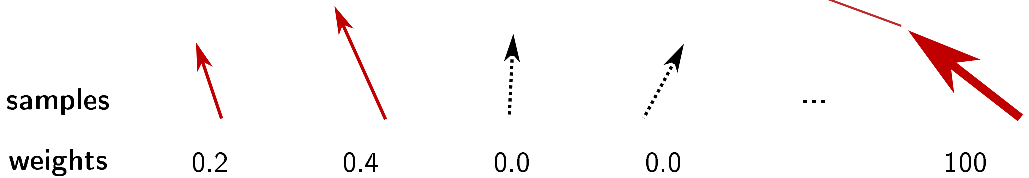


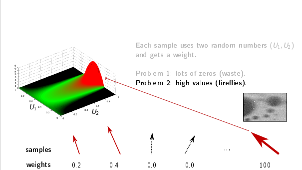
Each sample uses two random numbers (U_1, U_2) and gets a weight.

Problem 1: lots of zeros (waste).

Problem 2: high values (fireflies).



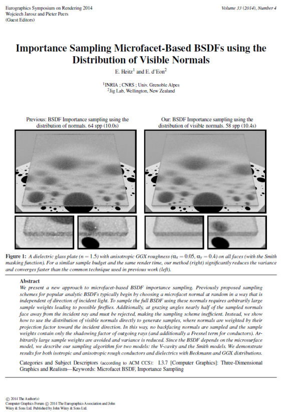




The second more important problem is this red peak. It represents the samples that have very high weights and that are directly responsible for the fireflies that are persistent even at 8k spp.

VNDF Sampling

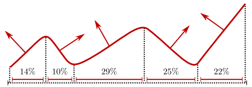
[Heitz&d'Eon2014]

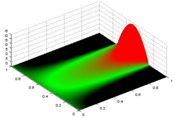

Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals

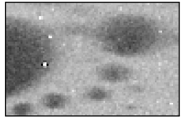
E. Heitz and E. d'Eon

''Sampling the NDF produces too much noise. We should rather sample the VNDF.''

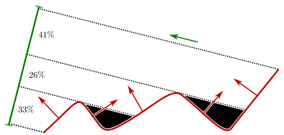
NDF sampling

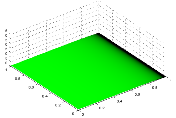


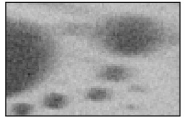


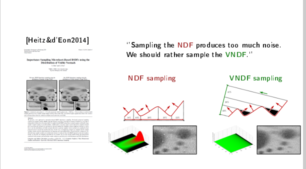


VNDF sampling



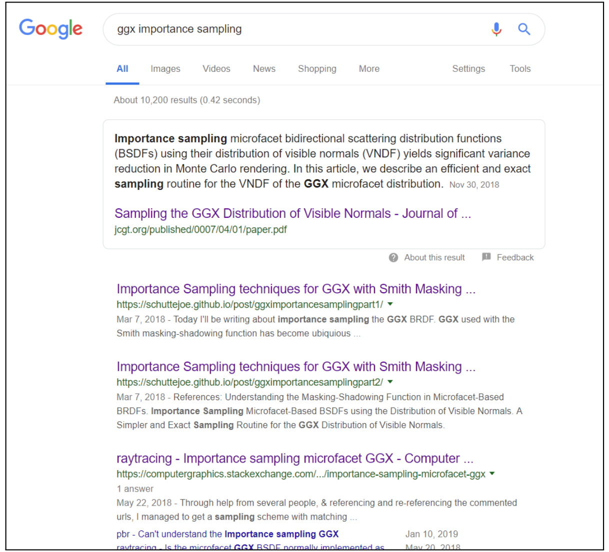


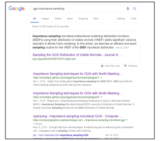




The message of the VNDF sampling paper was thus that NDF sampling is not efficient because it does not correctly account for the visibility of the samples. A better sampling strategy can be obtained by sampling directly from the VNDF. In this case, the sampling space becomes very smooth, has less zero values and does not have the red peak. As a result, the images do not exhibit these fireflies and converge faster.

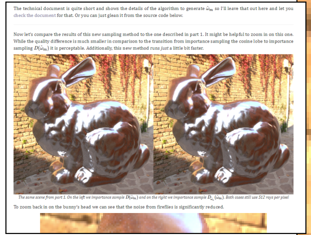
VNDF Sampling





Since 2014, people have been using VNDF sampling and reported their experience with it. You can find multiple blog posts about it by searching on Google.

VNDF Sampling

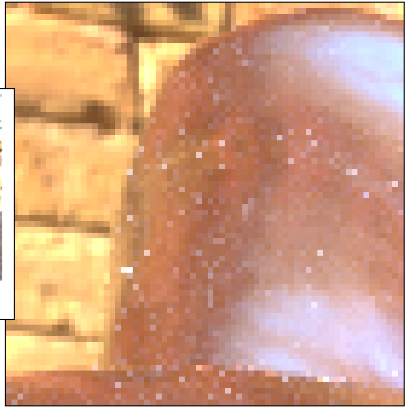

The technical document is quite short and shows the details of the algorithm to generate $\hat{\omega}_m$ so I'll leave that out here and let you check the document for that. Or you can just glean it from the source code below.

Now let's compare the results of this new sampling method to the one described in part 1. It might be helpful to zoom in on this one. While the quality difference is much smaller in comparison to the transition from importance sampling the cosine lobe to importance sampling $D(\hat{\omega}_m)$ it is perceptable. Additionally, this new method runs *just* a little bit faster.

The same scene from part 1. On the left we importance sample $D(\hat{\omega}_m)$ and on the right we importance sample $D_{\hat{\omega}_o}(\hat{\omega}_m)$. Both cases still use 512 rays per pixel

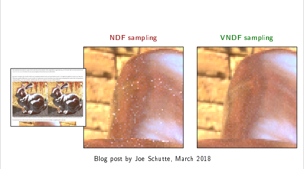
To zoom back in on the bunny's head we can see that the noise from fireflies is significantly reduced.

NDF sampling



VNDF sampling

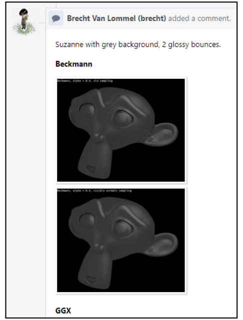
Blog post by Joe Schutte, March 2018



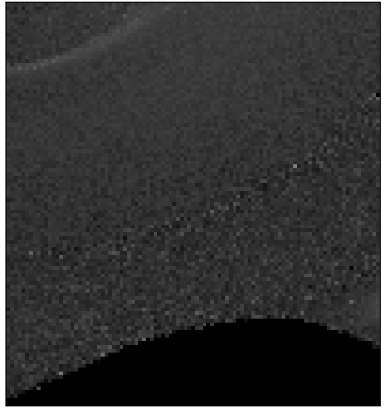
This is an example that shows how VNDF sampling prevents fireflies.

https://schuttejoe.github.io/post/ggximportancesamplingpart2/

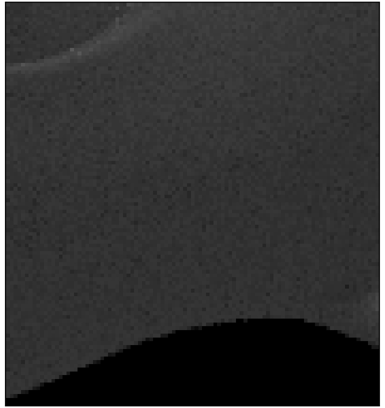
VNDF Sampling



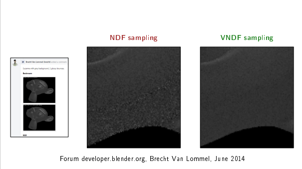
NDF sampling



VNDF sampling



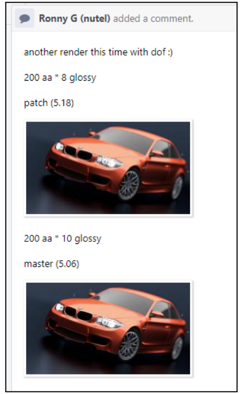
Forum developer.blender.org, Brecht Van Lommel, June 2014



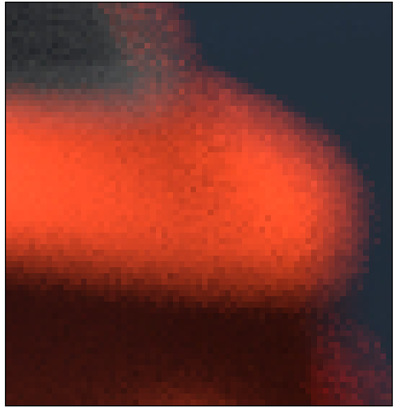
This is another example that shows that the resulting noise is lower.

https://developer.blender.org/D572

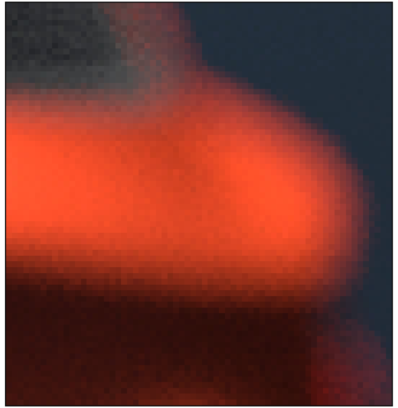
VNDF Sampling



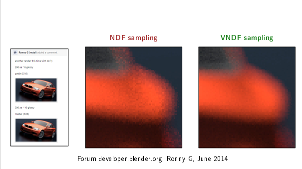
NDF sampling



VNDF sampling



Forum developer.blender.org, Ronny G, June 2014

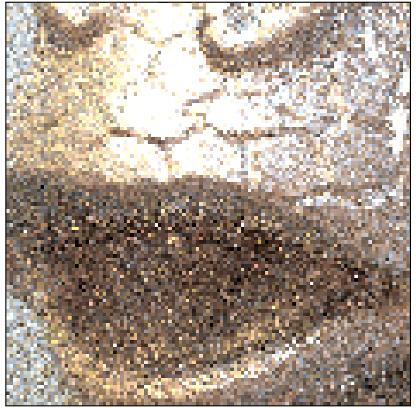


Another example.

https://developer.blender.org/D572

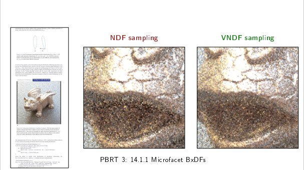
VNDF Sampling

NDF sampling



VNDF sampling

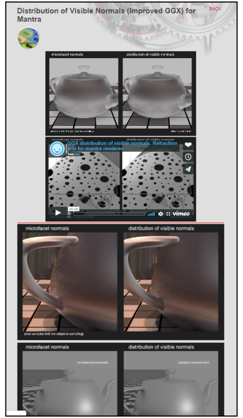
PBRT 3: 14.1.1 Microfacet BxDFs



VNDF sampling is referenced in PBRT 3 and even got a comparison image.

https://www.pbr-book.org/3ed-2018/Light_Transport_I_Surface_Reflection/Sampling_Reflection_Functions.html

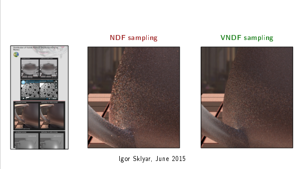
VNDF Sampling



NDF sampling

VNDF sampling

Igor Sklyar, June 2015

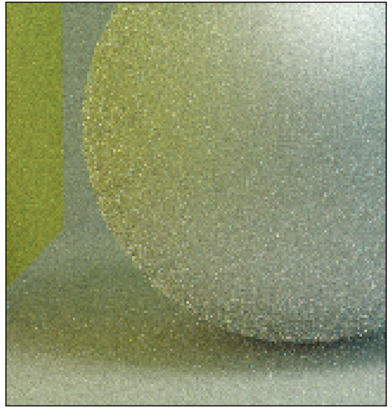


On this page you can even videos comparing NDF and VNDF sampling with time-varying roughness.

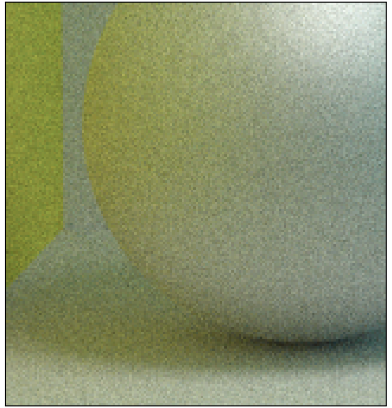
http://www.igorsklyar.com/main/development_description/24

VNDF Sampling

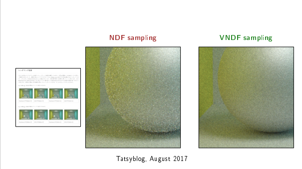
NDF sampling



VNDF sampling



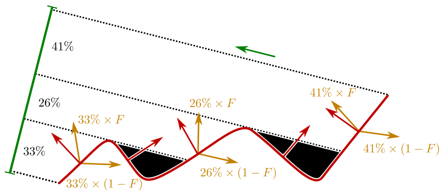
Tatsyblog, August 2017



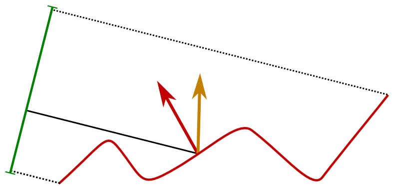
One more example.

http://tatsyblog.sakura.ne.jp/wordpress/applications/graphics/1742/

VNDF Sampling



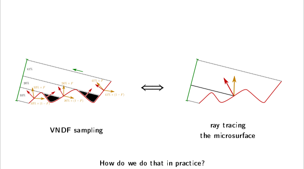
$\Longleftrightarrow$



VNDF sampling

ray tracing
the microsurface

How do we do that in practice?

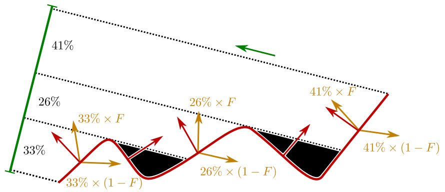


Conceptually, sampling the VNDF (the incident ray intersects a point of the surface that is visible) and applying a light transport operator (the ray is reflected or refracted) is equivalent to tracing rays on the microsurface, which is precisely what is modelled by the equation of a microfacet BRDF: tracing rays on a microsurface. This is why VNDF sampling is a good importance sampling technique for microfacet BRDFs.

However, this microsurface is not an actual surface. It is not represented as a mesh or a texture stored in memory that could be ray traced. It is just an abstraction represented by a handful of statistical parameters.

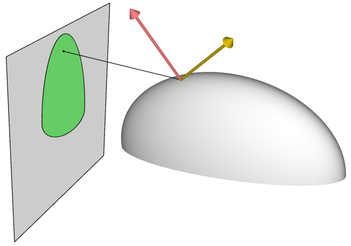
How do we ray trace an abstraction? It depends on the microsurface model that we use.

VNDF Sampling



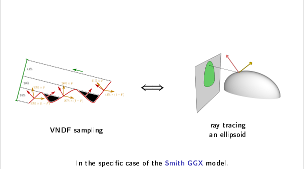
VNDF sampling

$\Longleftrightarrow$



ray tracing
an ellipsoid

In the specific case of the Smith GGX model.



We will see that in the specific case of the Smith GGX model, ray tracing the microsurface is equivalent to ray tracing an ellipsoid, which is something that can be implemented.

What is a Smith GGX microsurface?

What is a Smith GGX microsurface?

To understand why this is true, we will have to understand what modelling a Smith GGX microsurface means.

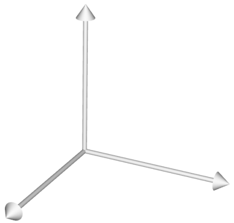
What is a Smith **GGX** microsurface?

What is a Smith **GGX** microsurface?

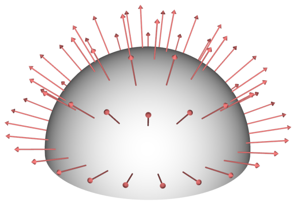
And we will start with the keyword GGX.

GGX comes from the [Walter et al. 2007] paper and means "Ground Glass Unknown" since they used it to fit the BSDF of ground glass. However, they did not provide the intuition behind the equation of the GGX normal distribution function.

What is a Smith **GGX** microsurface?

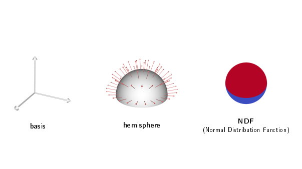


basis



hemisphere

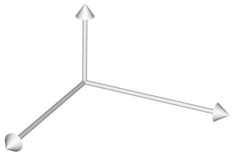
NDF
(Normal Distribution Function)



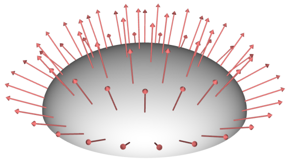
Let's consider a hemisphere (a sphere cut in two parts and we keep the upper part).

This hemisphere has normals that are all equally represented. So, the continuous distribution of these normals (the NDF) is a constant distribution.

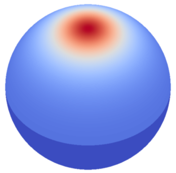
What is a Smith **GGX** microsurface?



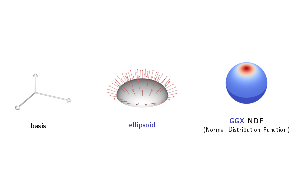
basis



ellipsoid



GGX NDF
(Normal Distribution Function)

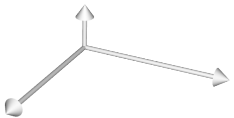


Now, imagine that we apply a linear transformation to this hemisphere, represented by the arrows in the basis figure.

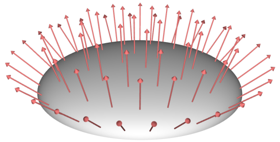
Undergoing the transformation, the hemisphere will become a truncated ellipsoid and the surface normals will follow the transformation.

The NDF of the transformed normals is exactly the GGX distribution.

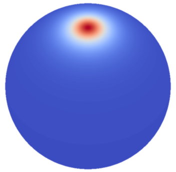
What is a Smith **GGX** microsurface?



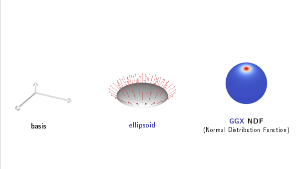
basis



ellipsoid



GGX NDF
(Normal Distribution Function)

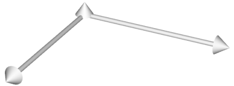


Now, imagine that we apply a linear transformation to this hemisphere, represented by the arrows in the basis figure.

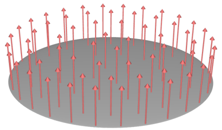
Undergoing the transformation, the hemisphere will become a truncated ellipsoid and the surface normals will follow the transformation.

The NDF of the transformed normals is exactly the GGX distribution.

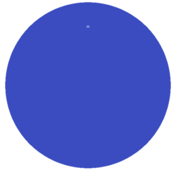
What is a Smith **GGX** microsurface?



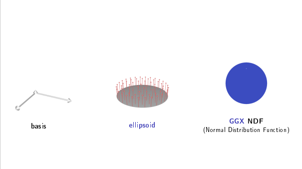
basis



ellipsoid



GGX NDF
(Normal Distribution Function)

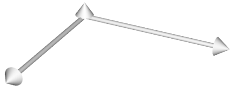


Now, imagine that we apply a linear transformation to this hemisphere, represented by the arrows in the basis figure.

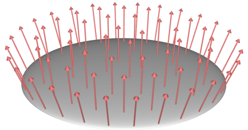
Undergoing the transformation, the hemisphere will become a truncated ellipsoid and the surface normals will follow the transformation.

The NDF of the transformed normals is exactly the GGX distribution.

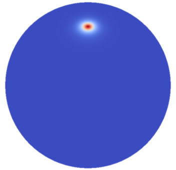
What is a Smith **GGX** microsurface?



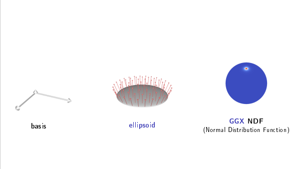
basis



ellipsoid



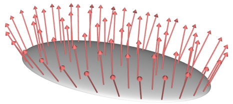
GGX NDF
(Normal Distribution Function)



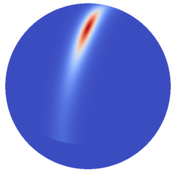
If we also compress one of the other axes, we obtain the anisotropic GGX distribution.

What is a Smith **GGX** microsurface?

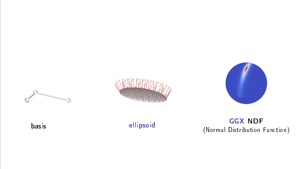
basis



ellipsoid



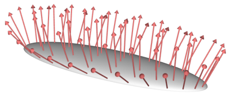
GGX NDF
(Normal Distribution Function)



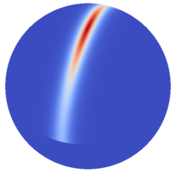
If we also compress one of the other axes, we obtain the anisotropic GGX distribution.

What is a Smith **GGX** microsurface?

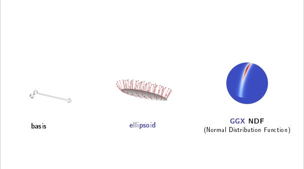
basis



ellipsoid



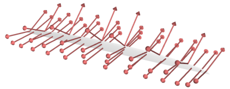
GGX NDF
(Normal Distribution Function)



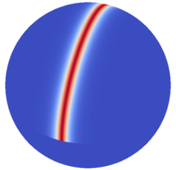
If we also compress one of the other axes, we obtain the anisotropic GGX distribution.

What is a Smith **GGX** microsurface?

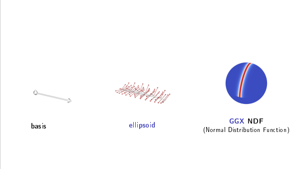
basis



ellipsoid

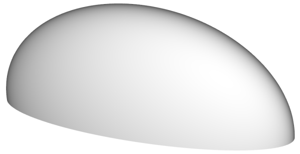


GGX NDF
(Normal Distribution Function)



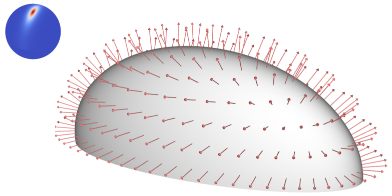
If we also compress one of the other axes, we obtain the anisotropic GGX distribution.

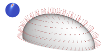
What is a Smith **GGX** microsurface?



Consider this ellipsoid.

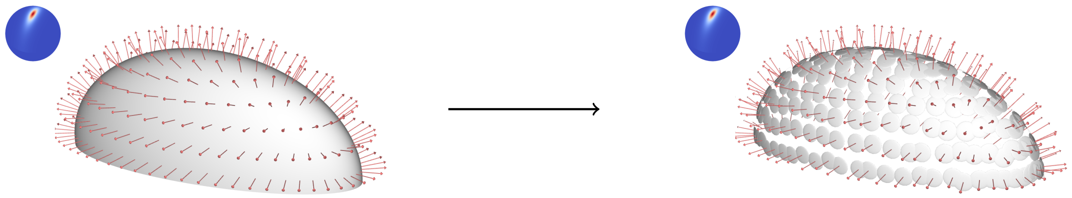
What is a Smith GGX microsurface?

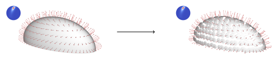




It has normals whose statistical distribution is a GGX distribution.

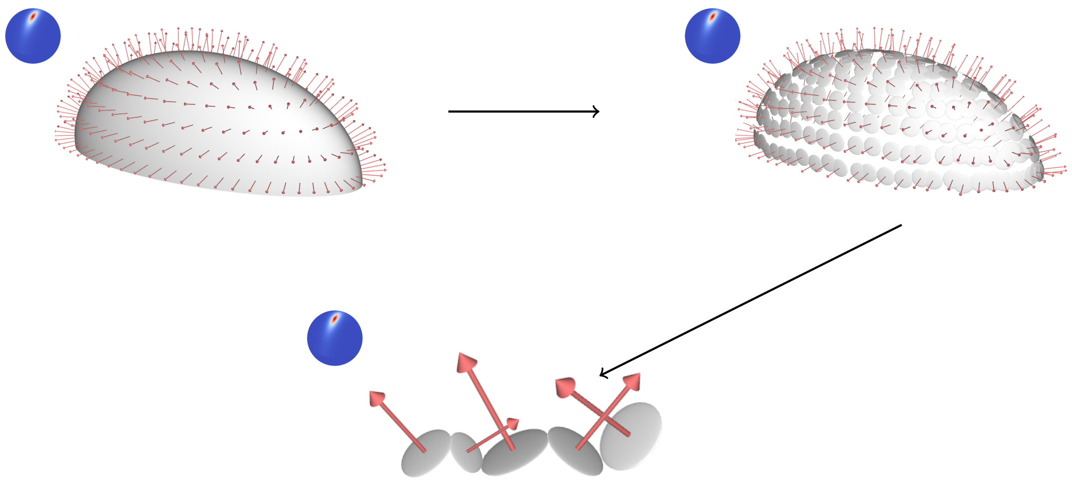
What is a Smith **GGX** microsurface?

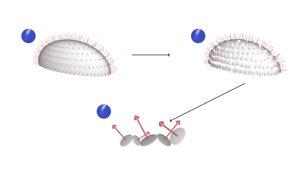




Now imagine that we cut this ellipsoid into infinitely small pieces.

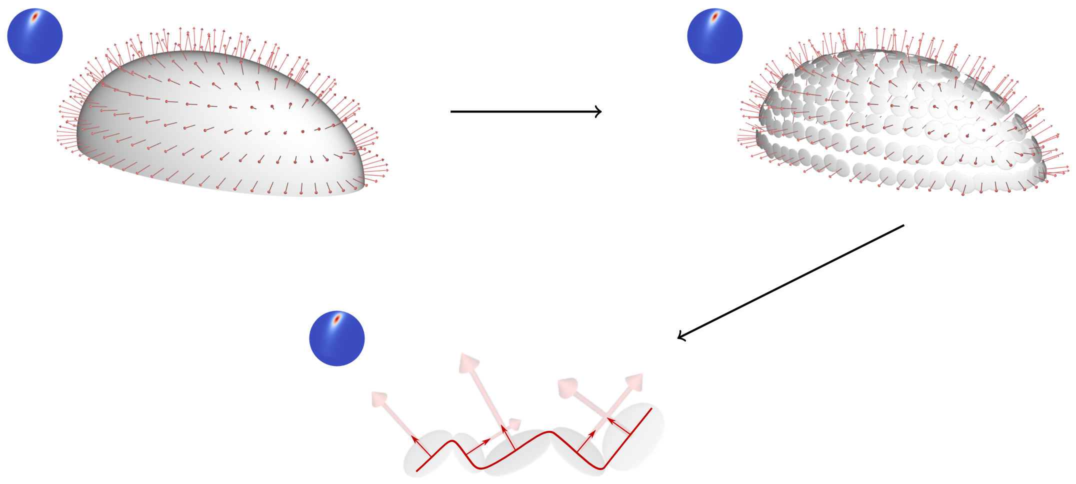
What is a Smith **GGX** microsurface?

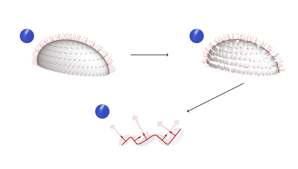




And we assemble them to produce a heightfield.

What is a Smith **GGX** microsurface?

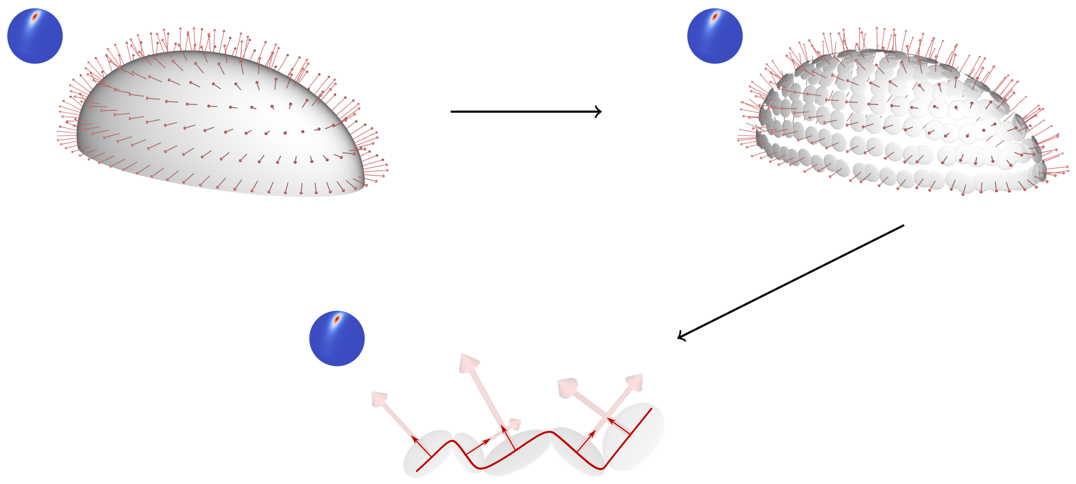




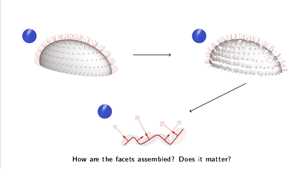
This is the continuous microsurface we have been considering.

Note that in this figure the facets are very roughly connected but imagine that they are infinitely small, much smaller than the microsurface's variations.

What is a Smith **GGX** microsurface?

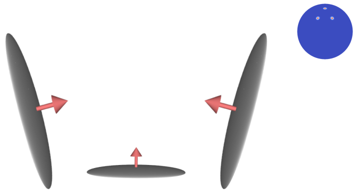


How are the facets assembled? Does it matter?

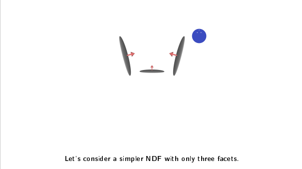


To properly define this microsurface, the last question we need to answer is how the microfacets obtained from the ellipsoid are connected together. And does it actually matter?

What is a Smith **GGX** microsurface?

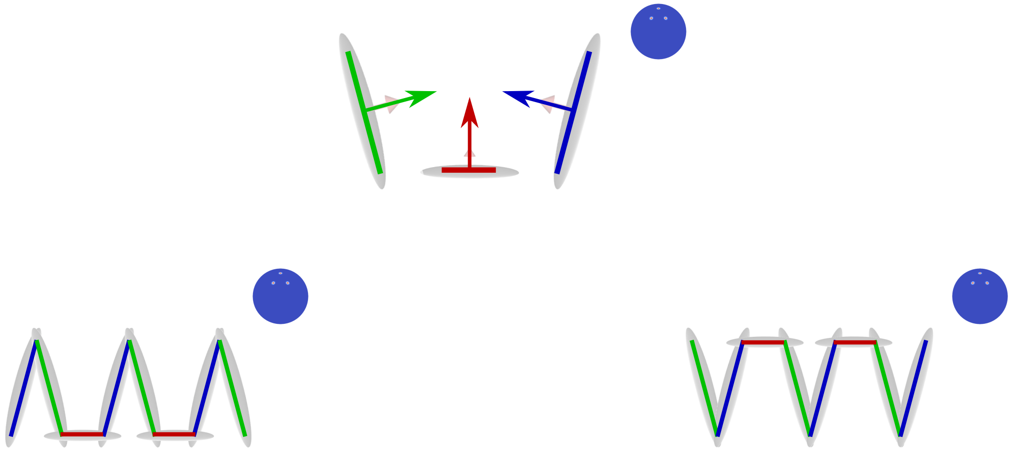


Let's consider a simpler NDF with only three facets.

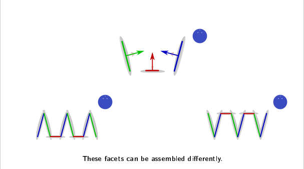


To convince us that how the facets are assembled matters, let's consider a simpler experiment with a discrete NDF of three facets.

What is a Smith **GGX** microsurface?

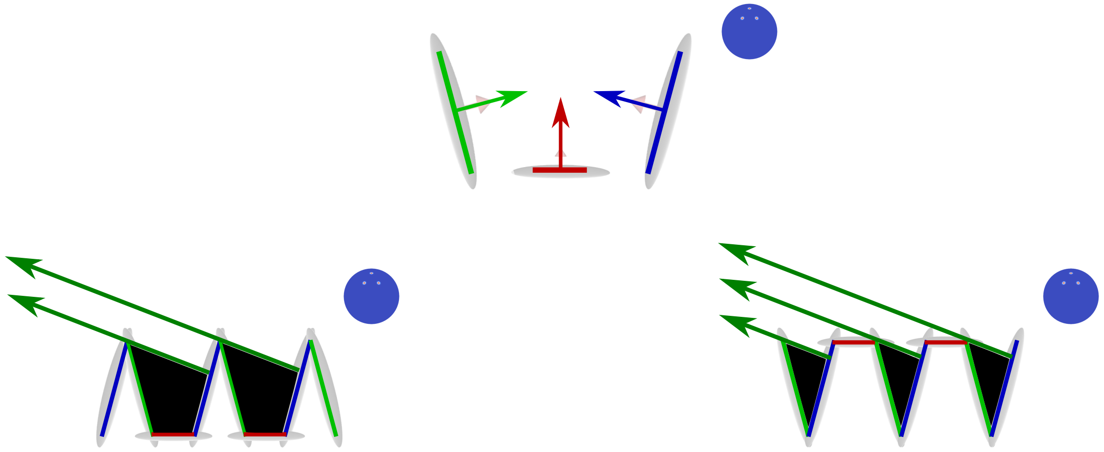


These facets can be assembled differently.

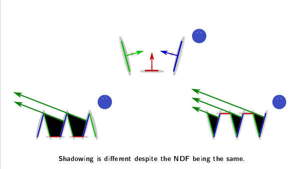


Even with only three facets, we can assemble them different ways. In this example, we obtain two different microsurfaces. They have the same NDF (the facets are the same) but they are different.

What is a Smith GGX microsurface?



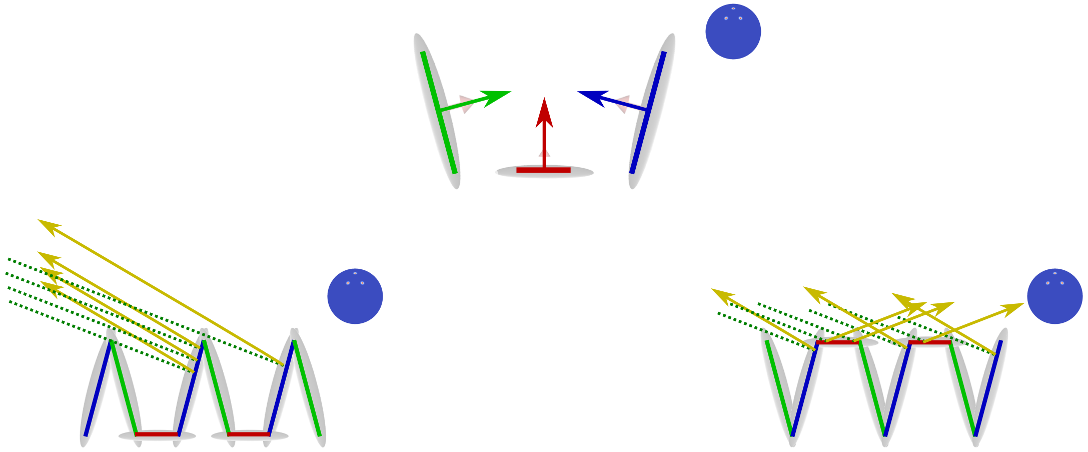
Shadowing is different despite the NDF being the same.



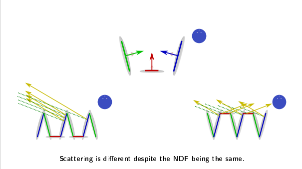
Because they are different, the shadowing is different.

For instance, the red facet is totally shadowed on the right while it is totally visible on the left.

What is a Smith GGX microsurface?

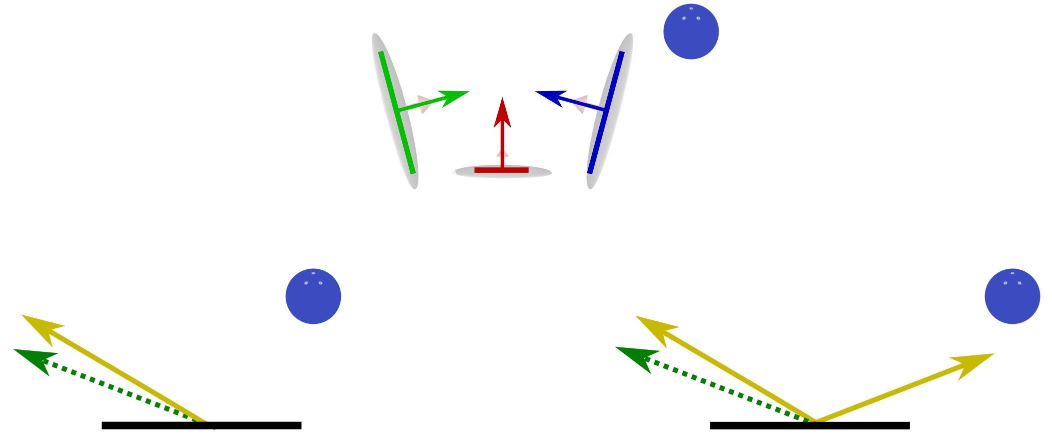


Scattering is different despite the NDF being the same.

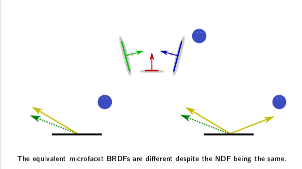


Because the surfaces are different, the scattering is different.

What is a Smith **GGX** microsurface?



The equivalent microfacet BRDFs are different despite the NDF being the same.



Hence, if we computed the microfacet BRDFs resulting from these surface models, we would obtain two different BRDFs despite their NDFs being the same.

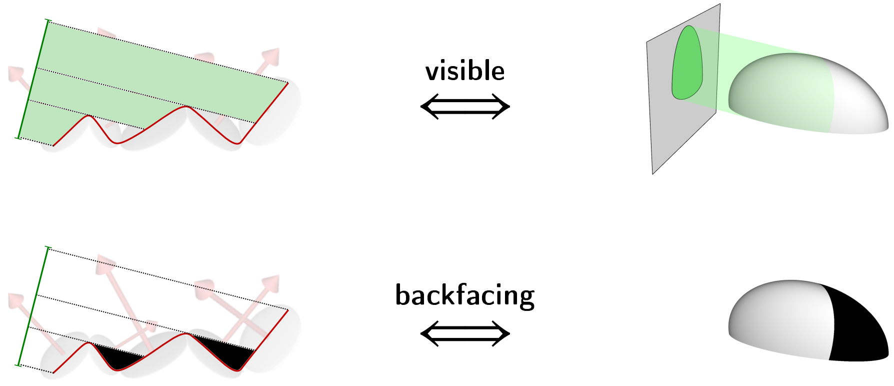
This shows that the NDF alone is not enough to determine the scattering behavior of a microsurface. For instance, saying ''GGX'' alone does not really make sense because different BRDFs can be based on a GGX NDF. We also need to choose a microsurface profile that models how the facets are assembled together.

What is a **Smith** GGX microsurface?

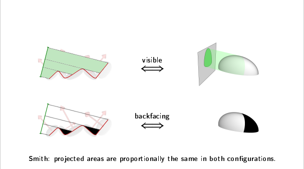
What is a **Smith** GGX microsurface?

In our case, this relates to the second keyword of the model: Smith.

What is a **Smith** GGX microsurface?



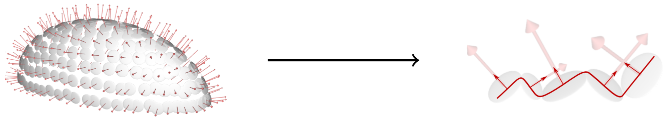
Smith: projected areas are proportionally the same in both configurations.



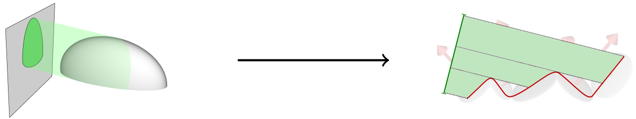
The fundamental assumption of the Smith model is that the facets are assembled in such a way that their projected area on the microsurface is independent of their shadowing probability. This means that all the facets that are not backfacing contribute proportionally the same as they do on the ellipsoid.

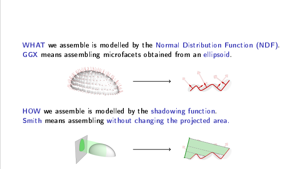
What is a Smith GGX microsurface?

WHAT we assemble is modelled by the **Normal Distribution Function (NDF)**.
GGX means assembling microfacets obtained from an **ellipsoid**.



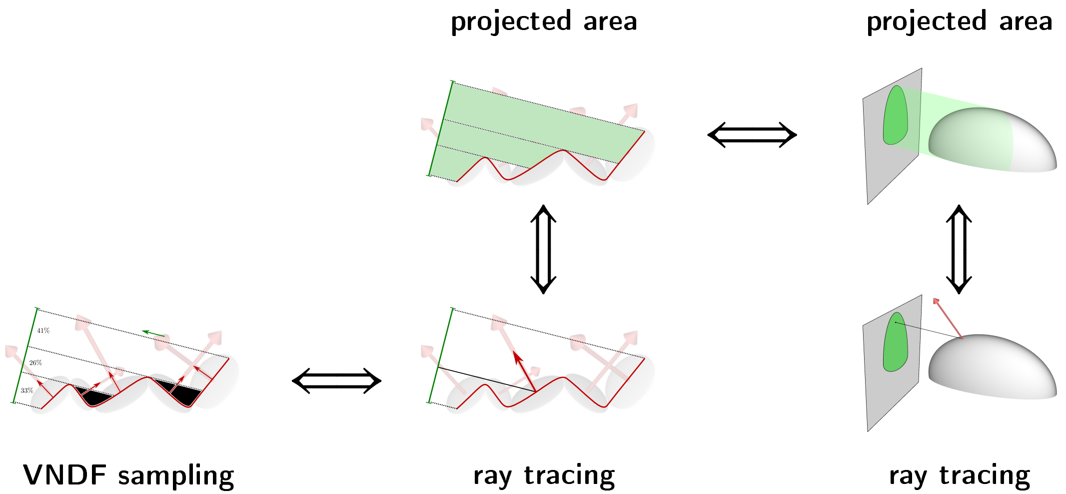
HOW we assemble is modelled by the **shadowing function**.
Smith means assembling **without changing the projected area.**

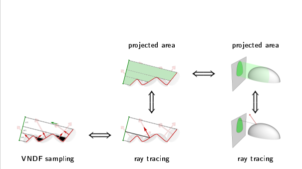




In summary, modelling a microsurface means answering these two questions: "WHAT is the microsurface assembled from?" (NDF) and "HOW is it assembled?" (shadowing function).

Using the properties of a Smith GGX microsurface



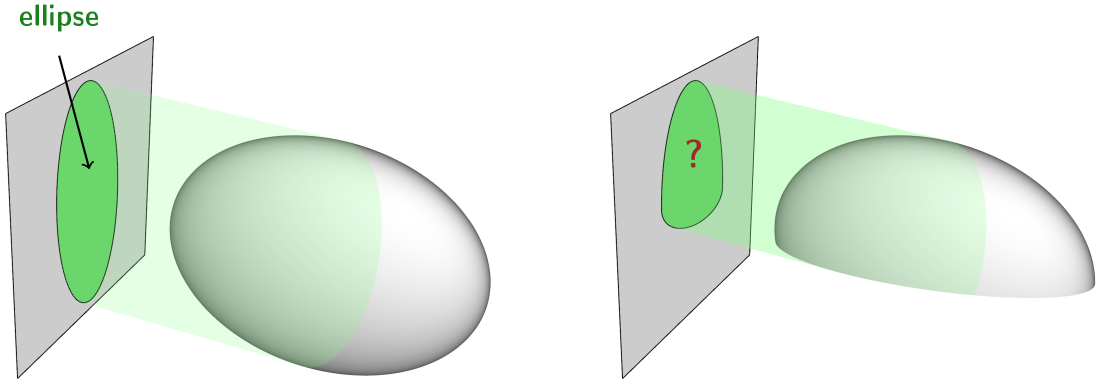


Now that we understand the Smith GGX microsurface model, it becomes obvious that VNDF sampling with Smith GGX is equivalent to ray tracing an ellipsoid.

Implementing VNDF Sampling for Smith GGX

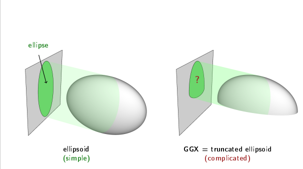
And we now know what we need to implement: a ray-tracing routine for an ellipsoid.

Implementing VNDF Sampling for Smith GGX



ellipsoid
(simple)

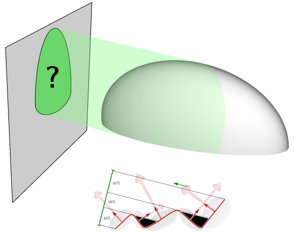
GGX = truncated ellipsoid
(complicated)

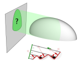


There is one last minor difficulty: while the projected area of a full ellipsoid is an ellipse (and thus trivial to sample), the shape of a truncated ellipsoid is more difficult to describe and then to sample.

Fortunately, as we shall see, it is not that complicated either.

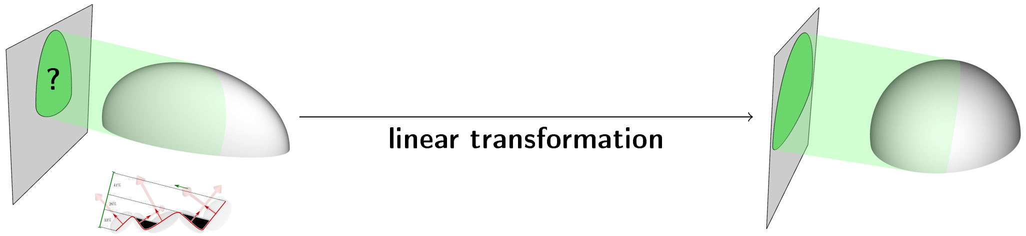
Implementing VNDF Sampling for Smith GGX

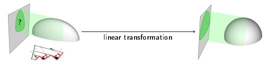




Our problem is thus to sample the projected area of a truncated ellipsoid.

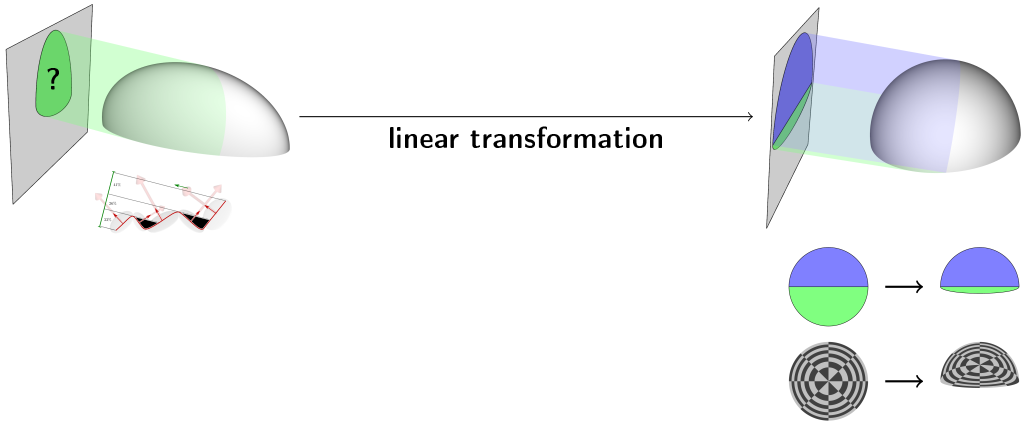
Implementing VNDF Sampling for Smith GGX

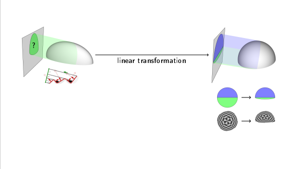




As we have seen, a truncated ellipsoid is a hemisphere that has undergone a linear transformation. We can thus reverse transformation and move to a configuration where we sample the projected area of a hemisphere instead.

Implementing VNDF Sampling for Smith GGX

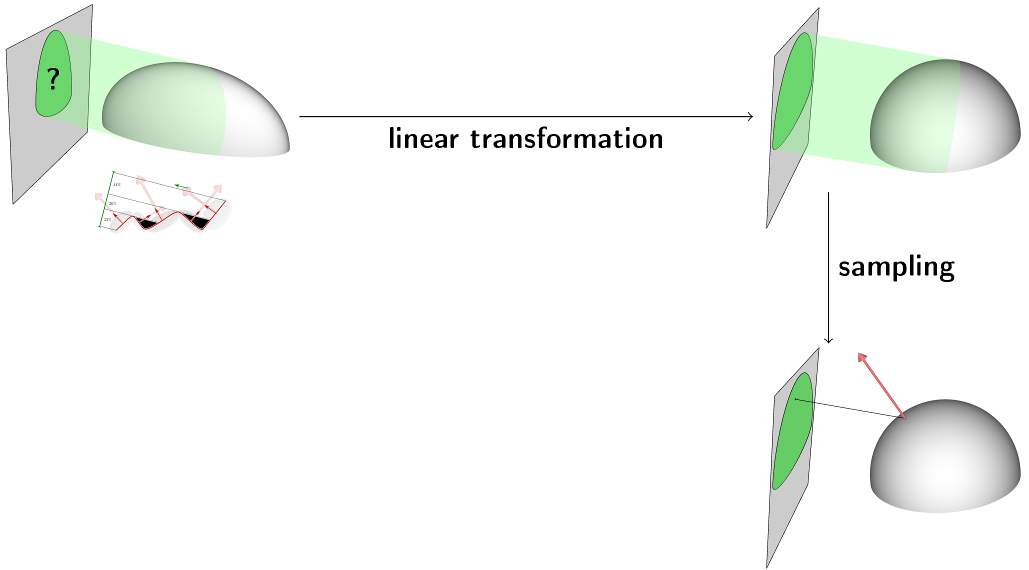


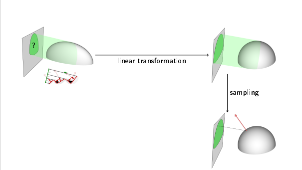


Fortunately, doing this is simple because the projected area of a hemisphere is always the (signed) sum of two half disks, the bottom one being compressed towards the top.

A sampling scheme for this shape can thus be obtained by sampling a point from a disk and compressing its vertical coordinate. Simple!

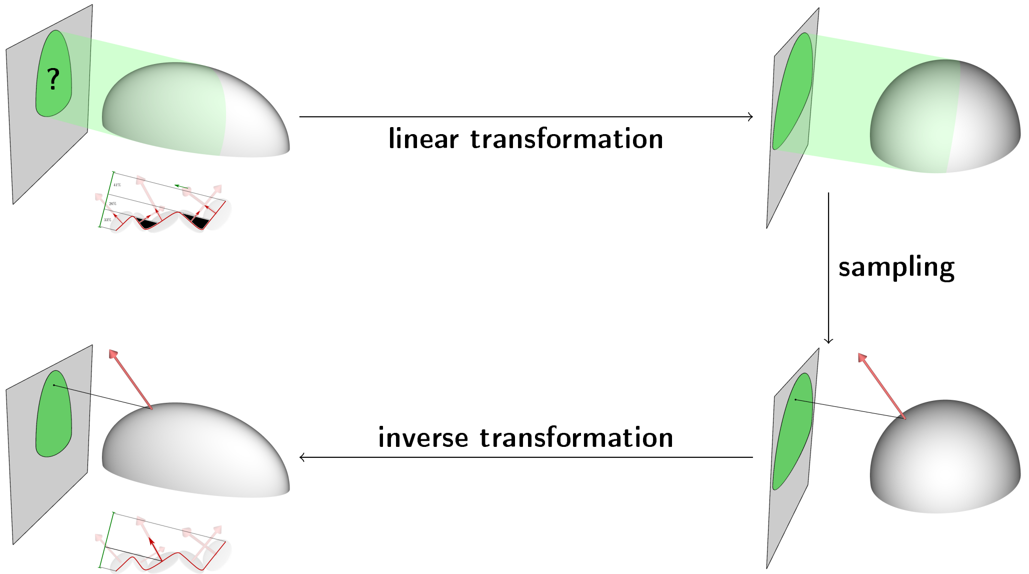
Implementing VNDF Sampling for Smith GGX

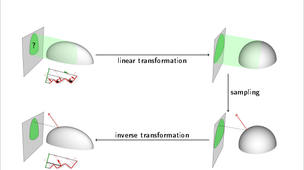




This is what we do to obtain a sample in the hemispherical configuration.

Implementing VNDF Sampling for Smith GGX

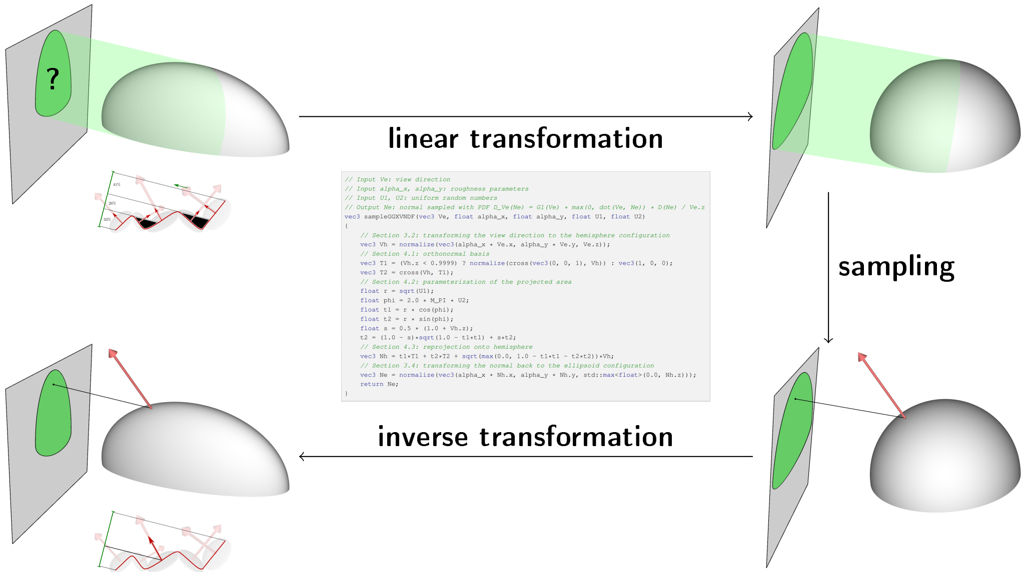




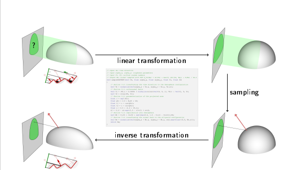
Finally, by restoring the linear transformation of the ellipsoid, we obtain a sample in the ellipsoidal configuration.

Obtaining this sample is effectively equivalent to applying VNDF sampling to the associated Smith GGX microsurface.

Implementing VNDF Sampling for Smith GGX

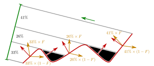
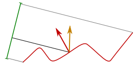
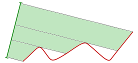
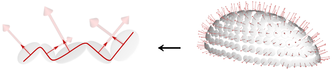
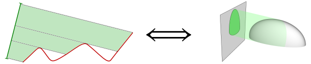
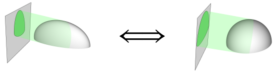
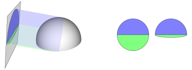


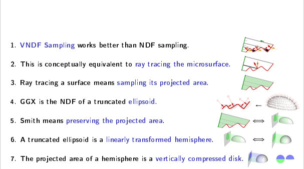

```
// Input Ve: view direction
// Input alpha_x, alpha_y: roughness parameters
// Input U1, U2: uniform random numbers
// Output Ne: normal sampled with PDF D_Ve(Ne) = G1(Ve) * max(0, dot(Ve, Ne)) * D(Ne) / Ve.z
vec3 sampleGGXVNDF(vec3 Ve, float alpha_x, float alpha_y, float U1, float U2)
{
    // Section 3.2: transforming the view direction to the hemisphere configuration
    vec3 Vh = normalize(vec3(alpha_x * Ve.x, alpha_y * Ve.y, Ve.z));
    // Section 4.1: orthonormal basis
    vec3 T1 = (Vh.z < 0.9999) ? normalize(cross(vec3(0, 0, 1), Vh)) : vec3(1, 0, 0);
    vec3 T2 = cross(Vh, T1);
    // Section 4.2: parameterization of the projected area
    float r = sqrt(U1);
    float phi = 2.0 * M_PI * U2;
    float t1 = r * cos(phi);
    float t2 = r * sin(phi);
    float s = 0.5 * (1.0 + Vh.z);
    t2 = (1.0 - s)*sqrt(1.0 - t1*t1) + s*t2;
    // Section 4.3: reprojection onto hemisphere
    vec3 Nh = t1*T1 + t2*T2 + sqrt(max(0.0, 1.0 - t1*t1 - t2*t2))*Vh;
    // Section 3.4: transforming the normal back to the ellipsoid configuration
    vec3 Ne = normalize(vec3(alpha_x * Nh.x, alpha_y * Nh.y, std::max<float>(0.0, Nh.z)));
    return Ne;
}
```

This is exactly what this routine does.

Implementing VNDF Sampling for Smith GGX

1. VNDF Sampling works better than NDF sampling. 
2. This is conceptually equivalent to ray tracing the microsurface. 
3. Ray tracing a surface means sampling its projected area. 
4. GGX is the NDF of a truncated ellipsoid. 
5. Smith means preserving the projected area. 
6. A truncated ellipsoid is a linearly transformed hemisphere. 
7. The projected area of a hemisphere is a vertically compressed disk. 


1. VNDF Sampling works better than NDF sampling.
2. This is conceptually equivalent to ray tracing the microsurface.
3. Ray tracing a surface means sampling its projected area.
4. GGX is the NDF of a truncated ellipsoid.
5. Smith means preserving the projected area.
6. A truncated ellipsoid is a linearly transformed hemisphere.
7. The projected area of a hemisphere is a vertically compressed disk.

That's all the intuitions that we used to get there.

One remarkable observation is that, thanks to these geometric intuitions, we could implement a VNDF sampling routine for Smith GGX without even having to look at the equation of Smith GGX.

Isn't that beautiful?

If you are interested in more details about Smith GGX, check out the paper and its previous work section.