International Journal of Advanced Smart Convergence Vol.12 No.4 361-369 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.4.361

IJASC 23-4-44

A study on the Performance of Hybrid Normal Mapping Techniques for Real-time Rendering

ZhengRan Liu, KiHong Kim, YuanZi Sang
Master, Department of Visual Contents, Dongseo University, China.
Professor, Department of Visual Contents, Dongseo University, Korea.
Doctor, Department of Visual Contents, Dongseo University, China.

dongtian991005@gmail.com, khkim@g.dongseo.ac.kr, syzcc0306@gmail.com

Abstract

Achieving realistic visual quality while maintaining optimal real-time rendering performance is a major challenge in evolving computer graphics and interactive 3D applications. Normal mapping, as a core technology in 3D, has matured through continuous optimization and iteration. Hybrid normal mapping as a new mapping model has also made significant progress and has been applied in the 3D asset production pipeline. This study comprehensively explores the hybrid normal techniques, analyzing Linear Blending, Overlay Blending, Whiteout Blending, UDN Blending, and Reoriented Normal Mapping, and focuses on how the various hybrid normal techniques can be used to achieve rendering performance and visual fidelity. performance and visual fidelity. Under the consideration of computational efficiency, visual coherence, and adaptability in different 3D production scenes, we design comparative experiments to explore the optimal solutions of the hybrid normal techniques by analyzing and researching the code, the performance of different hybrid normal mapping in the engine, and analyzing and comparing the data. The purpose of the research and summary of the hybrid normal technology is to find out the most suitable choice for the mainstream workflow based on the objective reality. Provide an understanding of the hybrid normal mapping technique, so that practitioners can choose how to apply different hybrid normal techniques to the corresponding projects. The purpose of our research and summary of mixed normal technology is to find the most suitable choice for mainstream workflows based on objective reality. We summarized the hybrid normal mapping technology and experimentally obtained the advantages and disadvantages of different technologies, so that practitioners can choose to apply different hybrid normal mapping technologies to corresponding projects in a reasonable manner.

Keywords: *Real-Time Rendering, Normal Mapping, Blended Normals, Visual Fidelity*

1. Introduction

In computer graphics, normal mapping is a common application of textures. In general, the higher the

Manuscript Received: November. 4, 2023 / Revised: November. 9, 2023 / Accepted: November. 18, 2023
Corresponding Author: syzcc0306@gmail.com
Tel: +82-10-8379-3060
Doctor, Department of Visual Contents, Dongseo University, China.

Copyright© 2023 by The Institute of Internet, Broadcasting and Communication. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0)

number of model faces, the more details can be shown and the better the result. However, with more faces and more vertices, the amount of computation is larger. To solve the problem normal mapping appears. By recording the directions of the normals on the surface of an object on a normal map, and using this normal information to calculate the lighting on each pixel when coloring, it is possible to increase the bump detail on the surface of an object without increasing the number of vertices. The principle of using normal maps to simulate the bumpiness of an object is that the normals are used to calculate diffuse reflections and highlights in the Bling-Phong lighting model.

2. Theoretical Examination

In the process of 3D asset development we often come across situations where we need to blend normals, such as: mossy mountains and rocks; clothing fabrics with detail normal; broken walls, rust-stained metals, etc. There may be some people ask, why do we need to mix the normals? We can prepare the normal map in advance or use materials to mix them in Substance Designer (SD) or PS. This not only saves the number of maps, reduces the number of samples, and reduces the calculation of mixing. But on the contrary, the real-time mixing of normals is precisely to save performance and better results, for example, if we want to show a very detailed sense of wool interspersed in the performance of the fabric. For example, if we want to show a very detailed interlacing of wool threads in a fabric, if we only use one normal to show it, then we may need a 4K or even 8K normal map to show the details with high accuracy. But these tiny details are often four-way continuous, so the current mainstream game practice is to use a basic normal + very small size detail normal with a relatively high tilling value for mixing. Not to mention blending between different terrains in a large world to get natural transitions and normal effects. So using blended normals is the optimal solution in Real-time rendering.

3. Experimental cases

We designed an experiment to mix two basic normal maps, namely normal map a and normal map b, through different normal blending techniques, and the result is normal map c. Then, we can intuitively judge the advantages and disadvantages of different normal blending technologies by comparing the normal map c. The following is an experimental analysis of several mainstream mixing normal methods:

3.1 Linear Blending

Linear blending is a fundamental concept in computer graphics and image processing where two images or layers are combined by linearly interpolating their pixel values. It's a basic form of blending that uses a weighted average of the pixel values from each layer to produce a final composite image. Linear blending is a very simple and crude method, and probably the one we use most of the time, where the normals are derived from direct post-stacking sampling and then interpolated linearly. The output of this method averages the normals of the two inputs, resulting in flattening and blurring of details. This is because the normal map stores not the color, but the current pixel normal information, the same way to deal with color to deal with the normal will inevitably result in data being averaged. The mixing effect is shown in [Figure 1].

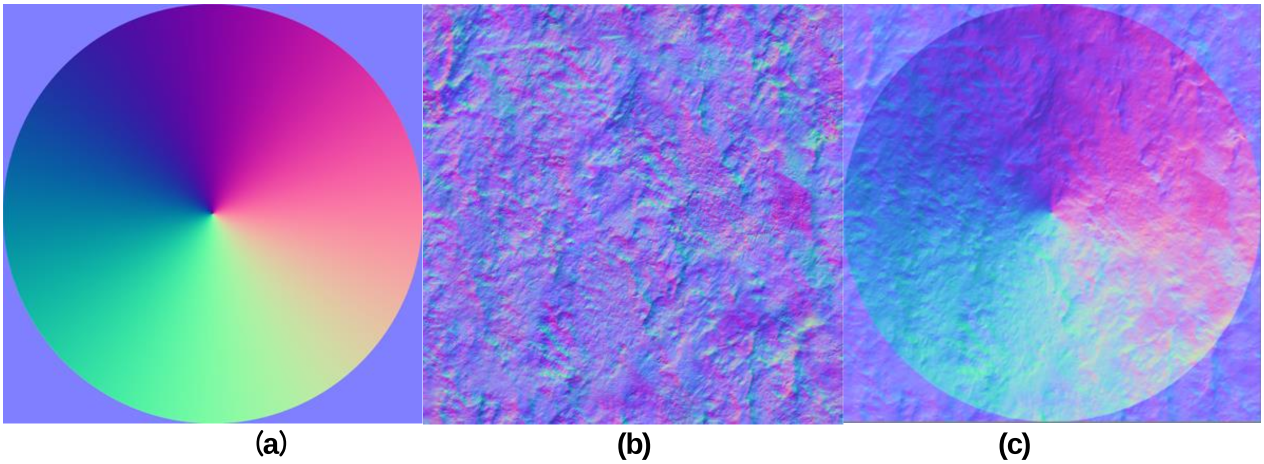


Figure 1. (a) is the basic normal map (b) is the basic normal map (b)

(c) is the result of (a) and (b) through Linear Blending

3.2 Overlay Blending

Overlay Blending is a technique used in graphics, imaging, and photography to combine two images or layers together. It works by enhancing the contrast and color saturation of a base layer while preserving its highlights and shadows to create an effect. Although Overlay has overall improvements compared to Linear Blending, but the output is still not correct, it is the same way as Linear Blending is to process the channels independently, which is basically the same as mixing them in PS. This algorithm seems reasonable, but in fact, it still shows incorrect, we still do unified processing of the mapping channel, and not according to the characteristics of the vector to the channel for separate processing. The reason why some people use Overlay Blending, may be because this blending method compared to other PS blending display better results. The mixing effect is shown in [Figure 2].

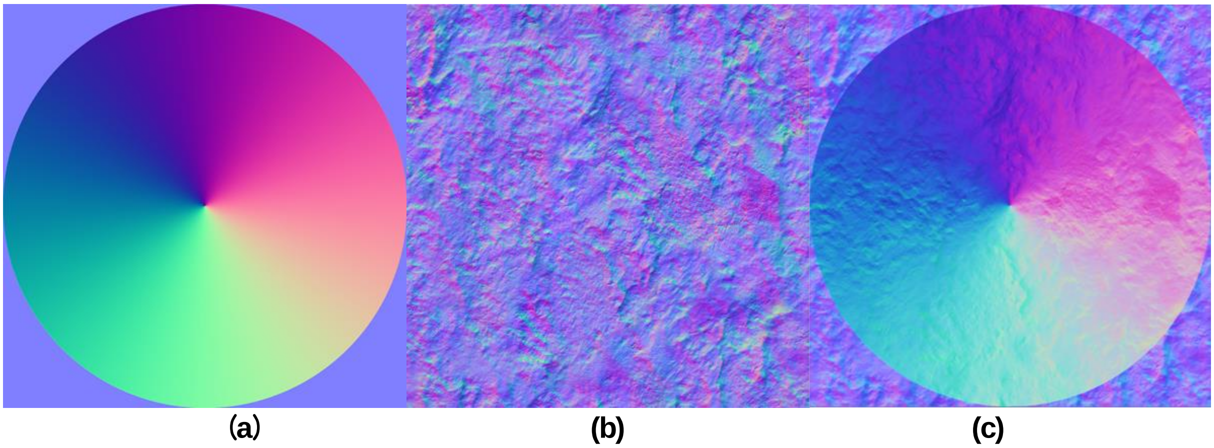
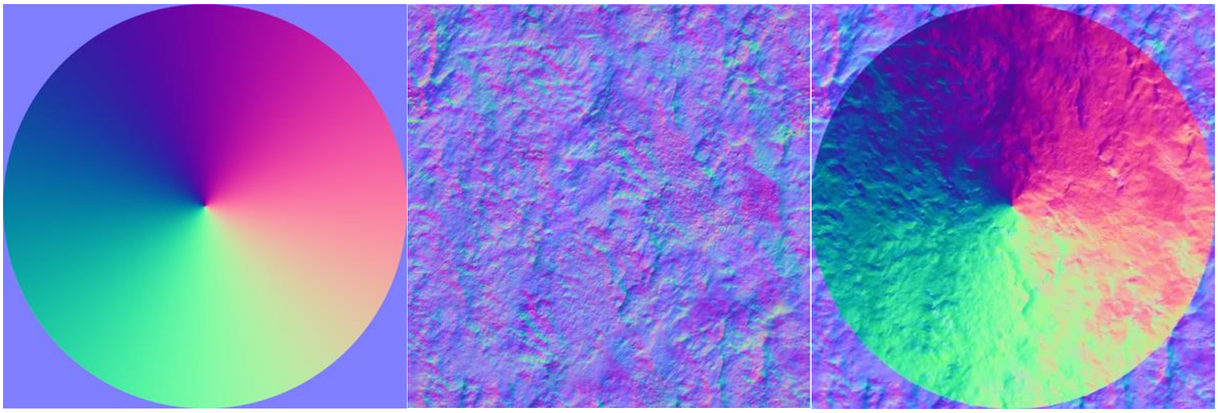


Figure 2. (a) is the basic normal map (b) is the basic normal map

(c) is the result of (a) and (b) through Overlay Blending

3.3 Whiteout Blending

The whiteout blending method was first proposed at SIGGRAPH 2007. This method is similar to partial derivative blending, except that the Z component is not multiplied on the xy channel. This method aims to solve the "flattening problem" in cone surface detail mapping. The purpose is to preserve detail and prevent excessive flattening of surfaces when using a blending technique like Whiteout Blending. Judging from the mixing results, this method solves the problem of flattening of the cone surface detail map well, but there is still the problem of surface stretching. The mixing effect is shown in [Figure 3].



(a) (b) (c)

Figure 3. (a) is the basic normal map (b) is the basic normal map

(c) is the result of (a) and (b) through Whiteout Blending

3.4 UDN Blending

UDN Blending, also known as Unit Dual Number blending, is a technique used in computer graphics and image processing. A dual number consists of a real part and a dual part, which can be thought of as an infinitesimal displacement. In the context of UDN Blending, these double numbers are used to represent not only color information, but also its gradient. This allows for a more accurate representation of gradients and derivatives during blending operations, making it particularly useful in tasks involving smooth transitions, such as image processing, anti-aliasing, or gradient-based effects. This mixing method was proposed on the Unreal Engine Developer Forum. This mixing method only cancels the addition of the z component based on linear mixing. Compared with Whiteout Blending, the blending effect of this blending method at the boundary will be worse, and the flatten effect is more obvious. However, because this blending method saves shader instructions, it can better handle gradients and smooth transitions between pixels or elements. , thereby reducing artifacts and improving the overall quality of the blending process. And it will be used more frequently on low-end machines. The mixing effect is shown in [Figure 4].

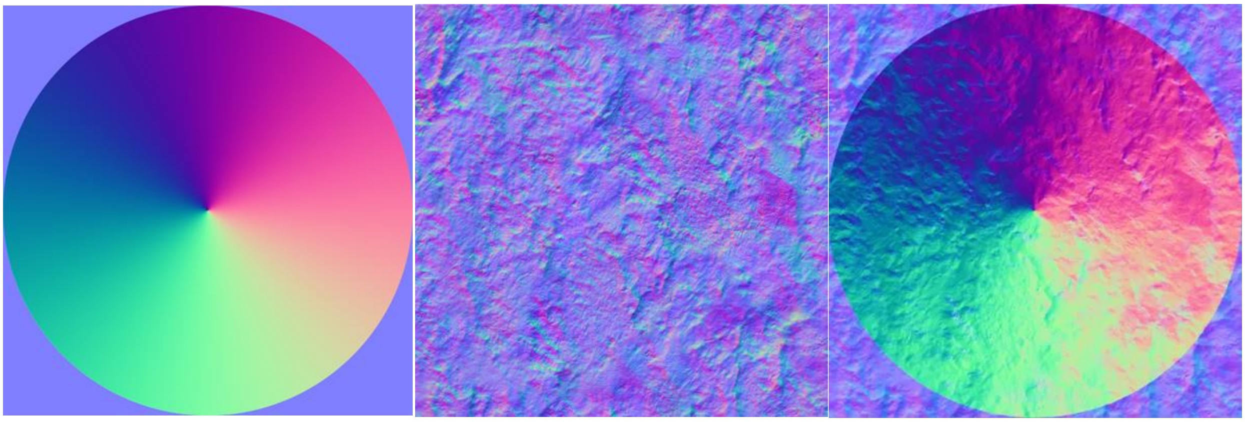


Figure 4. (a) is the basic normal map (b) is the basic normal map (b)

(c) is the result of (a) and (b) through UDN Blending

3.5 Reoriented Normal Mapping

Reoriented Normal Mapping (ORM) is a technique used in computer graphics to enhance the visual detail of 3D surfaces. It improves the fidelity of normal maps by recalculating or "reorienting" the normals stored in a normal map to better align with the surface geometry of a model. It involves converting the normals stored in the normal map to align with the current surface tangent space. This recalculation ensures that lighting and shading effects rendered using normal maps are more accurate and consistent with deformed surfaces. By employing reorientation normal maps, you can significantly improve the visual quality of your 3D models, especially when the surface is deformed or transformed non-rigidly. This technology helps maintain realistic lighting and shadow effects, thereby increasing the visual fidelity of real-time rendering and computer-generated images. The mixing effect is shown in [Figure 5].

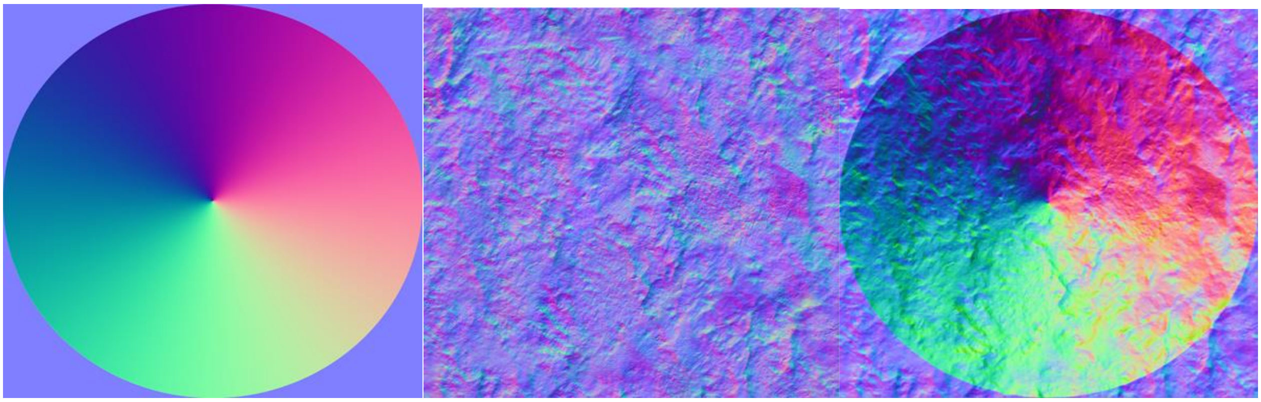


Figure 5. (a) is the basic normal map (b) is the basic normal map

(c) is the result of (a) and (b) through Reoriented Normal Mapping

Reoriented Normal Mapping is currently the most effective, but also the mainstream engines are using the idea. You can obviously find that this method of mixing out of the normal, whether base normal or detail normal information are well preserved, the mixing results are very successful. Here is a brief description of its principle. As shown in [Figure 6], the white dotted line on the left is our model plane, S is the normal direction of this model plane, and U is the normal direction of the detail normal we want to give. The t line in the picture on the right is the base normal direction, which is the direction of our main normal. The principle of this algorithm is to redirect the detail normal to obtain the normal r by calculating the angle from the model surface normal S to T.

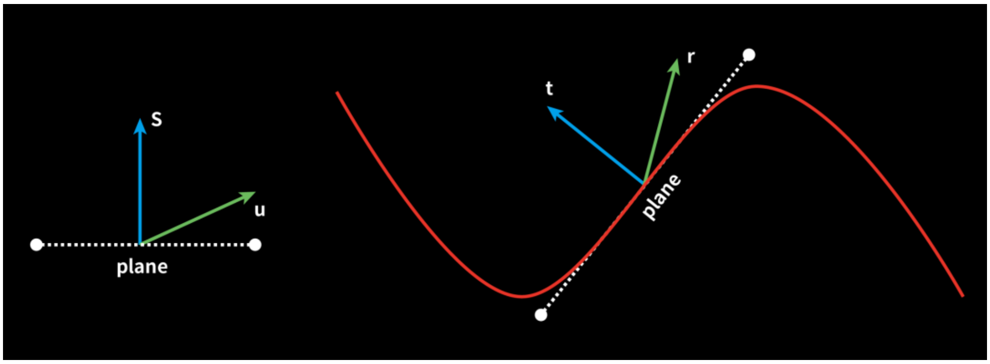
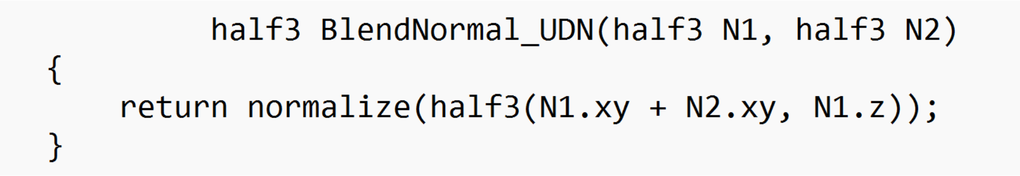


Figure 6. Reoriented Normal Mapping curve function

4. Data Comparison

From the above experiments, we have come up with two blending methods that are the most effective and adaptable in the engine, UDN Blending and Reoriented Normal Mapping, which will be tested in Unity in the following section. The shader used is the Unity implementation of the Normal Mapping Blending Surface Shader. It takes two normal maps (BumpMap and BumpDetailMap) and blends them according to the value of Bump Blending. The resulting normals are then used in lighting calculations to determine the final color of the pixels.

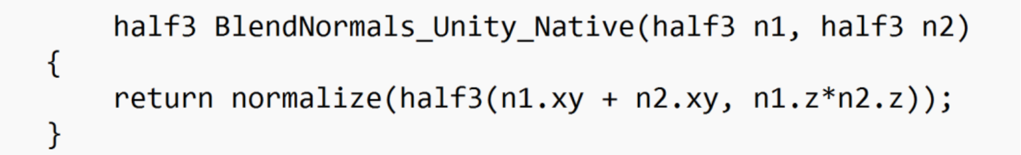
Blend Normal_UDN takes two input normals, N1 and N2, and blends them using a specific technique. Blending is done by adding the xy components of both normals and keeping the z component of the first normal. The resulting vector is then normalized to ensure that it represents a unit vector. A code demonstration is shown in [Figure 7].



```
            half3 BlendNormal_UDN(half3 N1, half3 N2)
{
    return normalize(half3(N1.xy + N2.xy, N1.z));
}
```

Figure 7. UDN Blending code demonstration

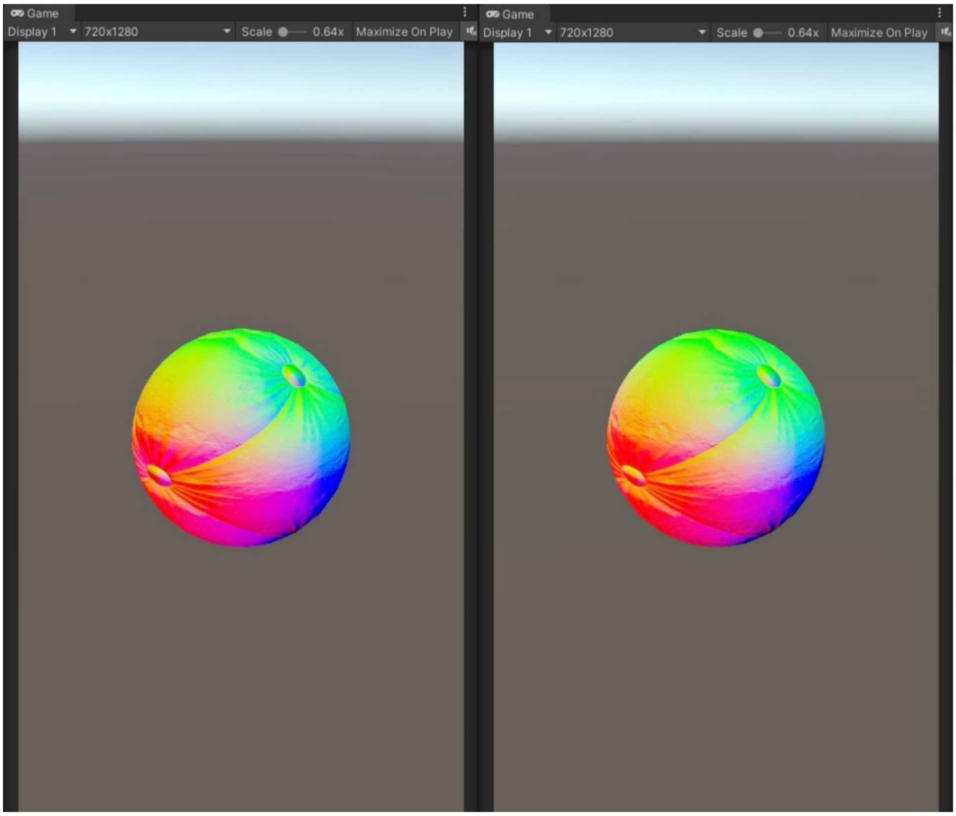
Reoriented Normal Mapping This hybrid method combines information from the two normals in a way that affects the xy component in a different way than the z component. The multiplication of the z-component indicates that the blending method takes into account the contributions of both normals in the z-direction and that the xy blending is more directly additive. The code demonstration is shown in [Figure 8].



```
    half3 BlendNormals_Unity_Native(half3 n1, half3 n2)
{
    return normalize(half3(n1.xy + n2.xy, n1.z*n2.z));
}
```

Figure 8. Reoriented Normal Mapping code demonstration

For normal texture blending, we usually need to satisfy the following three points: (1) Logical: the operation process can be realized using simple mathematical geometry. (2) Special case: when one of the textures is planar, the result is shown as the other texture. (3) Avoiding Flattening: the intensity of both normal textures is preserved. the method of Reoriented Normal Mapping will match the detail texture to the base texture by redirecting the detail mapping. This process is like transforming the tangent space normal when performing lighting calculations on a geometric object. The experimental results are shown in [Figure 9].



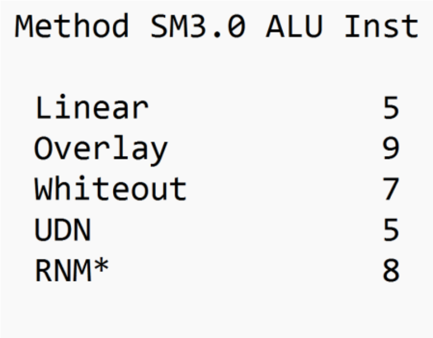
(a) **(b)**

Figure 9. Effect comparison chart

(a) is UDN Blending, (b) is Reoriented Normal

Analyzing the consumption of Arithmetic Logic Unit (ALU) instances under Shader Model 3.0 (SM3.0) for different normal fusion methods involves examining how these methods utilize computational resources, particularly ALU operations, within the context of graphics processing. Normal fusion methods like

Reoriented Normal Mapping (ORM), Whiteout Blending, or UDN Blending would have different computational requirements and ALU usage patterns due to their varying algorithms and calculations involved in manipulating normal maps. Develop shader implementations for each normal fusion method within the constraints and capabilities of Shader Model 3.0.Measure ALU Usage, Profile and measure the ALU usage for each method. This involves utilizing profiling tools provided by the graphics API or GPU vendors that can track ALU instructions executed per frame or per unit of time. Compare ALU Consumption: Compare the ALU usage among the different fusion methods. We'll want to look at metrics like the number of ALU instructions executed, time taken per frame, or any other relevant metrics provided by the profiling tools. The comparison results are shown in [Figure 10].



Method	SM3.0 ALU Inst
Linear	5
Overlay	9
Whiteout	7
UDN	5
RNM*	8

Figure 10. Results of ALU depletion by different normal fusion methods

Conclusion

From Figure 10 and the comparative results shown above, we can see that Linear and Overlay can be basically eliminated. Even on a lower-end machine, UDN has a better display than Linear. In the above example, it is hard to tell the difference between the visual effects of White Out and UDN, of course, this has a lot to do with our project environment. However, given that the consumption of White Out and UDN is not much different, I personally recommend using UDN to get better visual effects. UDN, compared to Reoriented Normal, is suitable for fast production of the required blended texture mapping, and requires less computer performance for fast computation rate, while Reoriented Normal, as the most mainstream blending mode, has the most comprehensive performance and is suitable for blending textures. Reoriented Normal, as the most mainstream blending mode, has the most comprehensive performance and is suitable for all kinds of projects. In specific projects, we can quickly preview the blending effect through UDN, and if it meets the requirements of the project, then we can use Reoriented Normal to create the final required normal maps. In this exploration, Linear Blending, Overlay Blending, and Reoriented Normal Mapping, representing mainstream hybrid normal techniques, have been subjected to a meticulous analysis, focusing on the intricate balance between rendering performance and visual fidelity. Through comparative experiments, it can be concluded that the Reoriented Normal Mapping method has the best degree of detail retention. While Linear Blending and Overlay Blending exhibit relatively weaker performance in terms of handling detail normals, they redeem themselves by not averaging the base normals, showcasing computational efficiency. This highlights the pivotal trade-off between detail fidelity and computational cost, enabling developers to make informed choices tailored to their project requirements, prioritizing either high-quality visuals or real-time performance efficiency. These findings underscore the multifaceted nature of hybrid normal mapping techniques, acknowledging the versatile solutions they offer in the ongoing quest for the optimal balance between performance and visual fidelity in modern 3D rendering.

References

[1] Pharr, Matt, Wenzel Jakob, and Greg Humphreys. Physically based rendering: From theory to implementation. MIT Press, 2023.

[2] S.Y. ZI, K.K. Hong and Y. Pan. "From Technology to Content: Research on the Development of VR Flow Experience" The International Journal of Advanced Smart Convergence 11, no.3, 2022, pp: 93-101.
DOI: http://dx.doi.org/10.7236/IJASC.2022.11.3.93

[3] Shirley, Peter, and R. Keith Morley. Realistic ray tracing. AK Peters, Ltd., 2008.

[4] Florentí Solano P. A real time hybrid pipeline in Vulkan. 2021.
DOI: https://repositori.upf.edu/handle/10230/48669

[5] W.Qgang, et al. “An active contour model based on fused texture features for image segmentation. ” Neurocomputing 151 ,pp. 1133-1141, 2015.

[6] Shum, H. Y., Chan, S. C., & Kang, S. B. Image-based rendering. Springer Science & Business Media, 2008.

[7] Jensen, H. W. Realistic image synthesis using photon mapping ,Vol. 364. Natick: Ak Peters, 2001.

[8] Yu, X., Dai, P., Li, W., Ma, L., Liu, Z., & Qi, X.. “Texture Generation on 3D Meshes with Point-UV Diffusion,” in Proc. the IEEE/CVF International Conference on Computer Vision ,pp. 4206-4216, 2023.

[9] Y. Wei, et al. “Enforcing geometric constraints of virtual normal for depth prediction. ” in Proc. the IEEE/CVF International Conference on Computer Vision, 2019.

[10] Haines, Eric, and Tomas Akenine-Möller, eds. Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs, 2019.

[11] Suffern K. Ray Tracing from the Ground up. CRC Press, 2016.