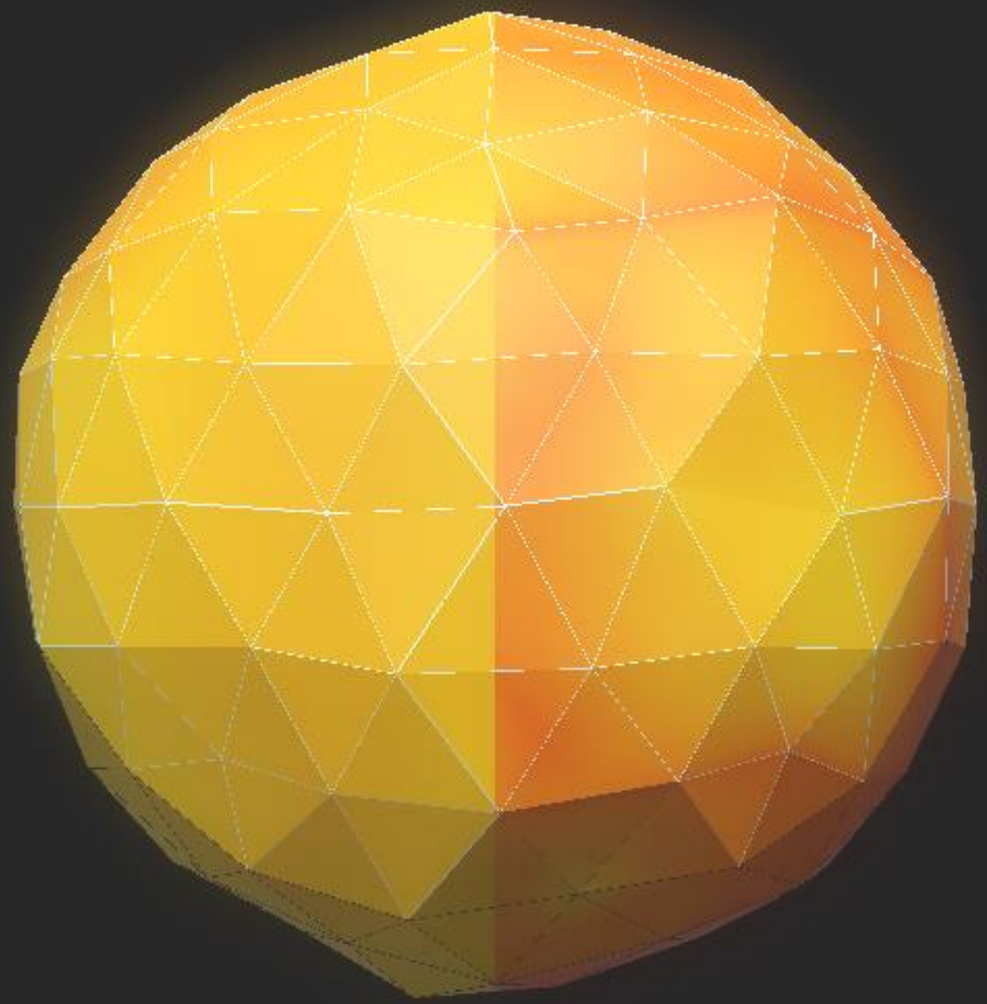


[Per VERTEX VS Per PIXEL SHADING]

When a per vertex shaded mesh, has advantages over a per pixel shaded mesh in a WebGL application?

Vertex Vs Pixel Shading Comparison

Presentation by

Martin Niehoff (326516)

info@martin-niehoff.de

As specialization topic in the Game Creation and Producing Course at the Saxion University of Applied Science

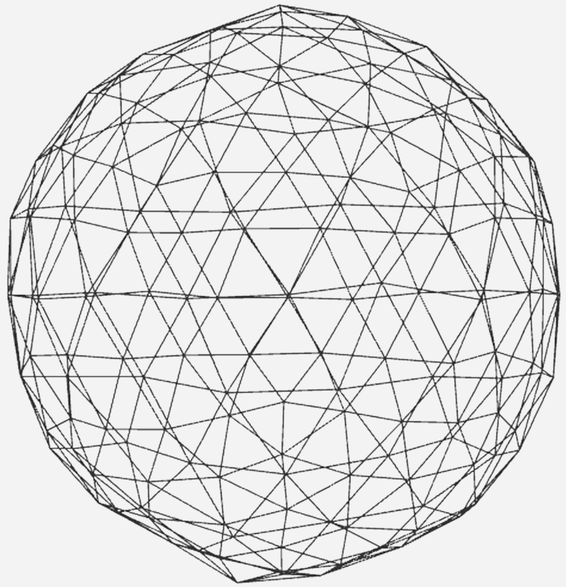
Supervising Teachers

Lukas Malec & Taco van Loon

Sonntag, 8. August 2021

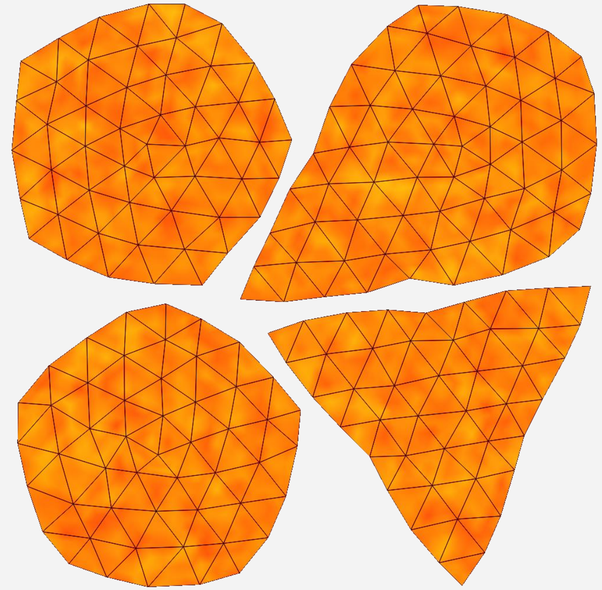
Per Vertex vs. Per Pixel Shading

Comparison for the use in WebGL



Per Vertex
Color

VS



Per Pixel
COLOR

???	**Features**	???
???	**Performance**	???
???	**Compatibility**	???
???	**File Sizes**	???

Introduction

the reason behind this research

I choose this topic for my research, because I wanted to know, if vertex shading would allow me to further optimize my assets for the use in WebGL.

I worked on a few web based projects during my internship, which showed me the different requirements, that interactive applications have, which are build for the web. So I was curious, how I could further optimize my assets for that use and after I read a article about vertex shading the topic for my research was quite clear.

The paper is examining the differences between per vertex and per pixel based shading, to find out when a per vertex shaded mesh, has advantages over a per pixel shaded mesh in a WebGL application, regarding their file sizes and VRAM usage.

Research Questions

Theoretical Background

When a per vertex shaded mesh, has advantages over a per pixel shaded mesh in a WebGL application?

When are per vertex shaded meshes resulting in smaller asset sizes?

When have per vertex shaded meshes a lower VRAM consummation?

Research Questions

Theoretical Background

- What means per vertex shading?
- What means per pixel shading?
- What are the requirements of WebGL?
- Are there different requirements between the two shading methods in WebGL?

Research Questions

Theoretical Background

- What is the average size of a website?
- What is the average bandwidth of users?
- What is the difference between mobile and static internet bandwidth?
- What is the maximum average time, that a website should take to load?

Scope

The boundaries of the research

Asset Sizes

What are the file sizes of the required assets?

VRAM Usage

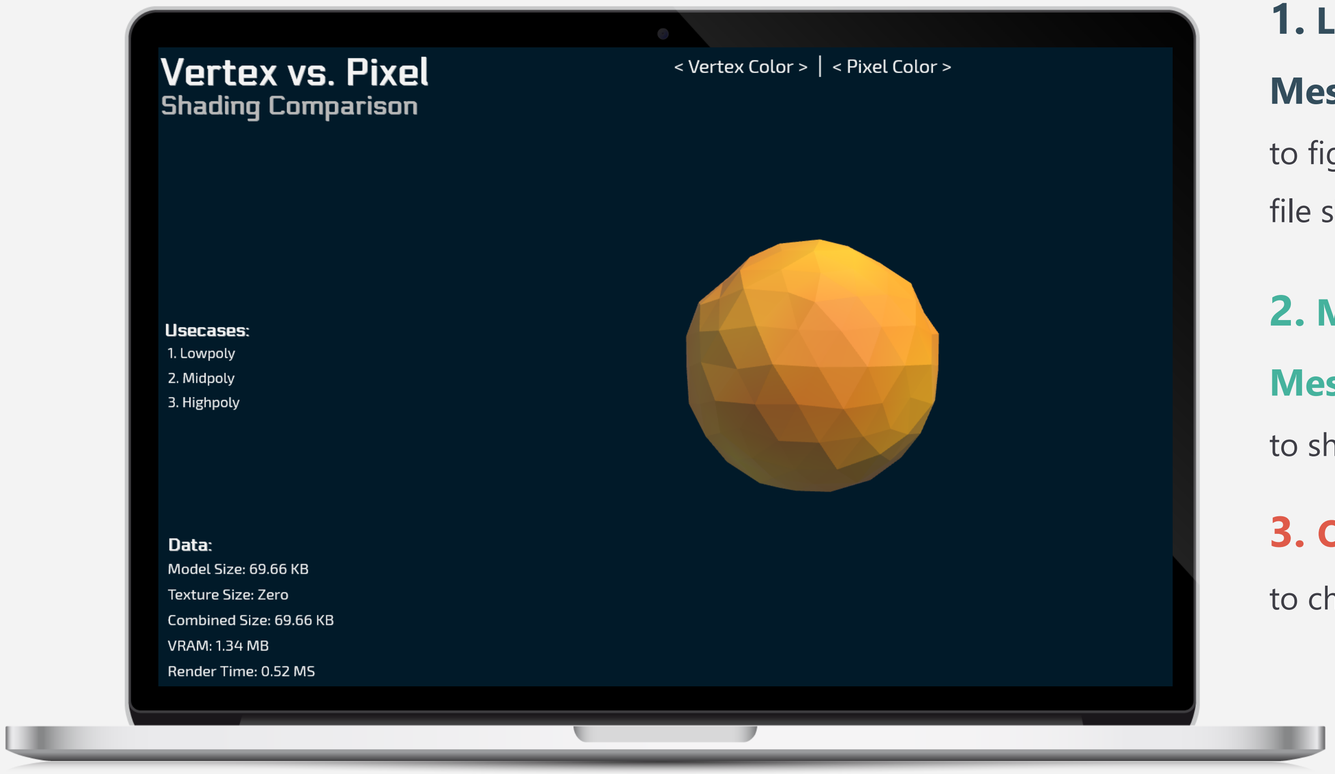
How much VRAM are both techniques consuming?

Usability

How user friendly are both techniques?

Prototype

Praxis oriented prototype



1. Low Poly Vertex Shaded Mesh vs. Low Poly Pixel Shaded Mesh

to figure out the influence, that the color and UVs information have on the file size of the assets

2. Mid Poly Vertex Shaded Mesh vs. Low Poly Pixel Shaded Mesh

to show the influence of the textures detail on the usability of vertex color.

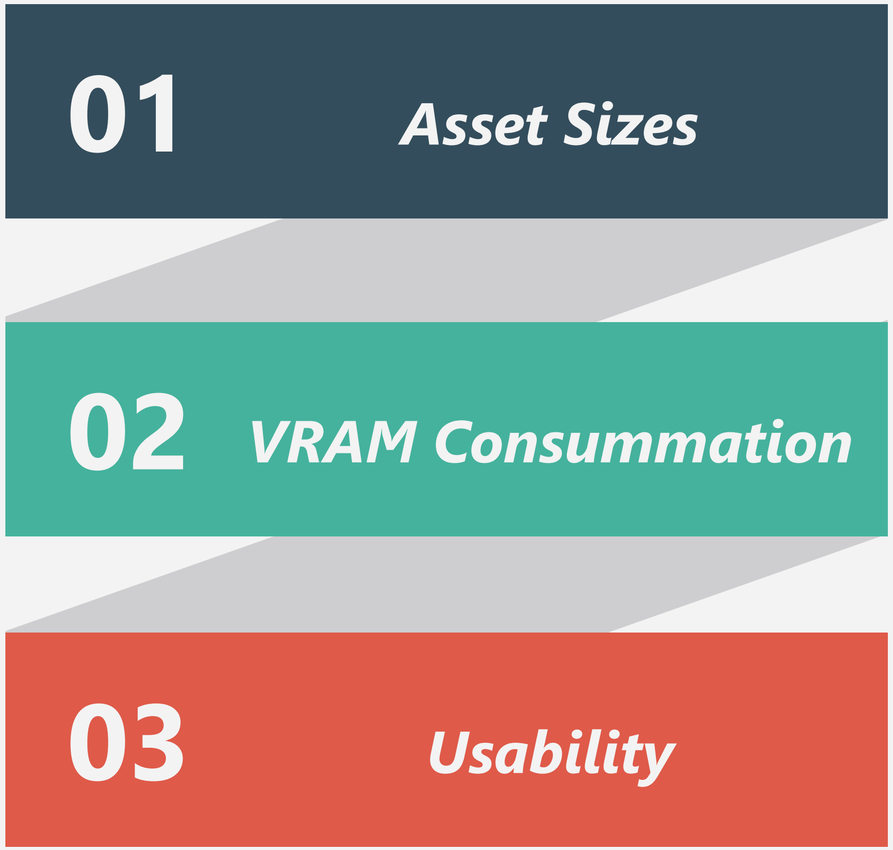
3. Optimized Vertex Mesh vs. Low Poly Pixel Shaded Mesh

to check the maximal possible data reduction with both techniques.

Results

The Outcome

The prototype allowed a comparison of both shading techniques under different circumstances, to compare their impact on the asset sizes as well as their VRAM usage. Furthermore, the creation of the required assets allowed a inside of the asset creation process behind both techniques.



Asset Size

Research Results

Vertex shaded meshes are only requiring 38Bytes of data per vertex

vs

While Pixel shaded mesh are claiming two additional bytes per vertex

This makes vertex color always the more lightweight shading technique, regarding the required file sizes, as long as the mesh does not require additional vertices, which only store color information.

However, if more vertices are required, it's depended of the amount of additional vertices. As each pixel of a texture takes up way less bytes, than a single vertex does.

Its therefore only cheaper to add additional vertices for supplementary color information to a mesh, if their additional data is not exceeding the data required for an additional texture.

VRAM Consummation

Research Results

The prototype showed, that the VRAM usage of vertex shaded meshes is taking up only a fraction of their pixel shaded opponents, even if their asset sizes are bigger.

While the entire scene of the first test case with the vertex shaded mesh only required **1.19MB** of VRAM, the same scene with the pixel shaded mesh already required **23.6MB** of VRAM.

> This makes vertex shaded meshes particular interesting for applications with the requirement of low VRAM usage.

Usability

Research Results

The asset creation for the prototype showed, that vertex shaded meshes can be created quicker, as the additional step of unwrapping the model falls away.

Furthermore, most modeling applications support vertex color painting out of the box, so that no additional software is required, for the creation of vertex shaded meshes.

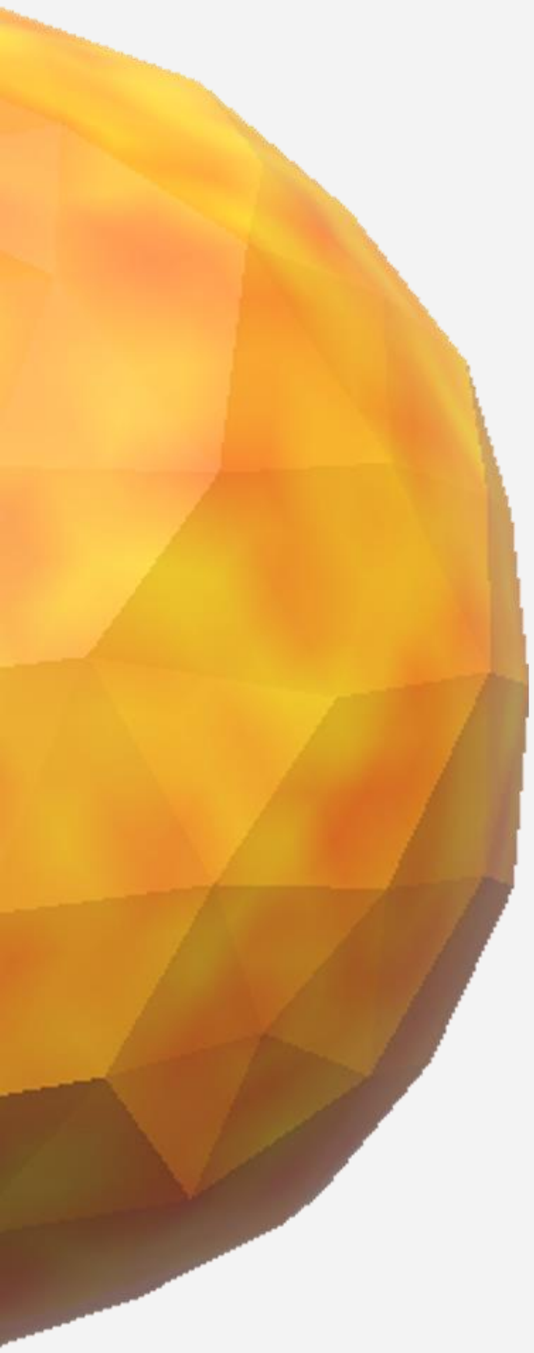
> Another interesting technique is the conversion of textures into geometry, a technique that was amongst other things used in Homeworld.

Conclusion

Of the research

Vertex shading offers a tempting alternative to reduce asset sizes and VRAM consummation, to some extend and therefore allows to optimize web based applications to download faster and run smoother on a greater amount of target devices, which can positively influence the user experience.

But the question of its usability remains a personal decision. While the creation of meshes with vertex color is generally quicker, due the lacking requirement of unwrapping the mesh, as well as the reduced number of production steps, it is quite different method of texturing, which might not fit to everybody and every project.



Recommendations

Our company missions

The use of per vertex colored meshes is a valid solution to reduce the asset size and VRAM consummation of WebGL applications, as long as some limitations are taken into account.

1. If the art style allows to color meshes via vertex color, without the addition of too many additional vertices, use vertex color. As this will reduce the asset sizes and VRAM consummation of the application. However, if this is not the case, a per pixel shaded mesh is cheaper regarding the asset sizes.

2. If vertex shaded meshes are requiring too many additional vertices, evaluate the advantage of the lower VRAM consummation for the target group. If that's a limiting factor for the application, consider the use of vertex color, even if that increases the asset sizes.

3. If low VRAM usage and tiny asset sizes are key requirement for the target group, consider using a low poly art style, as used in the prototype, because this offers an ideal use case for vertex color.

Thank you

Questions?