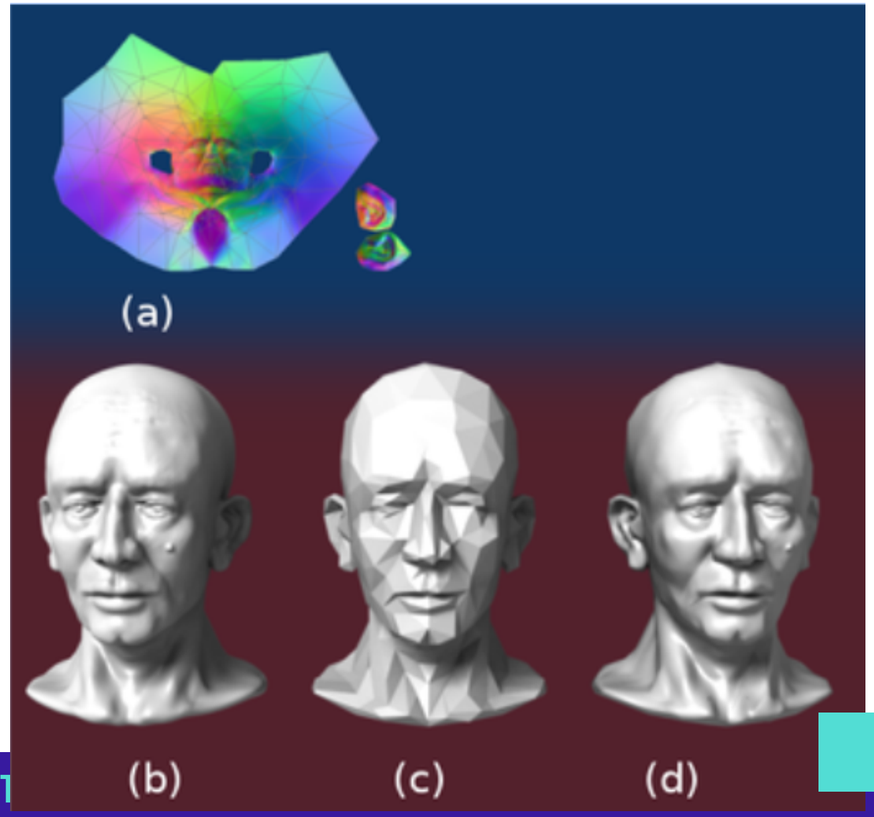
COMPUTACIÓN GRÁFICA

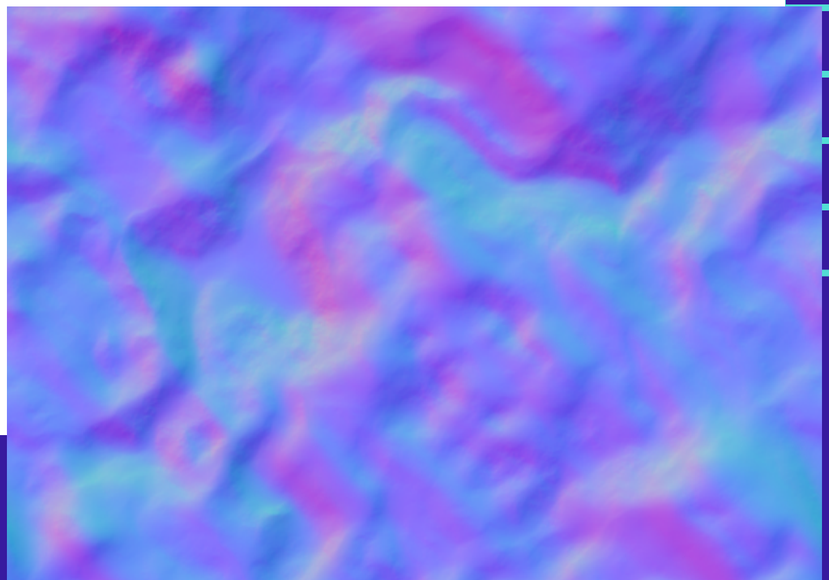
NORMAL MAPS

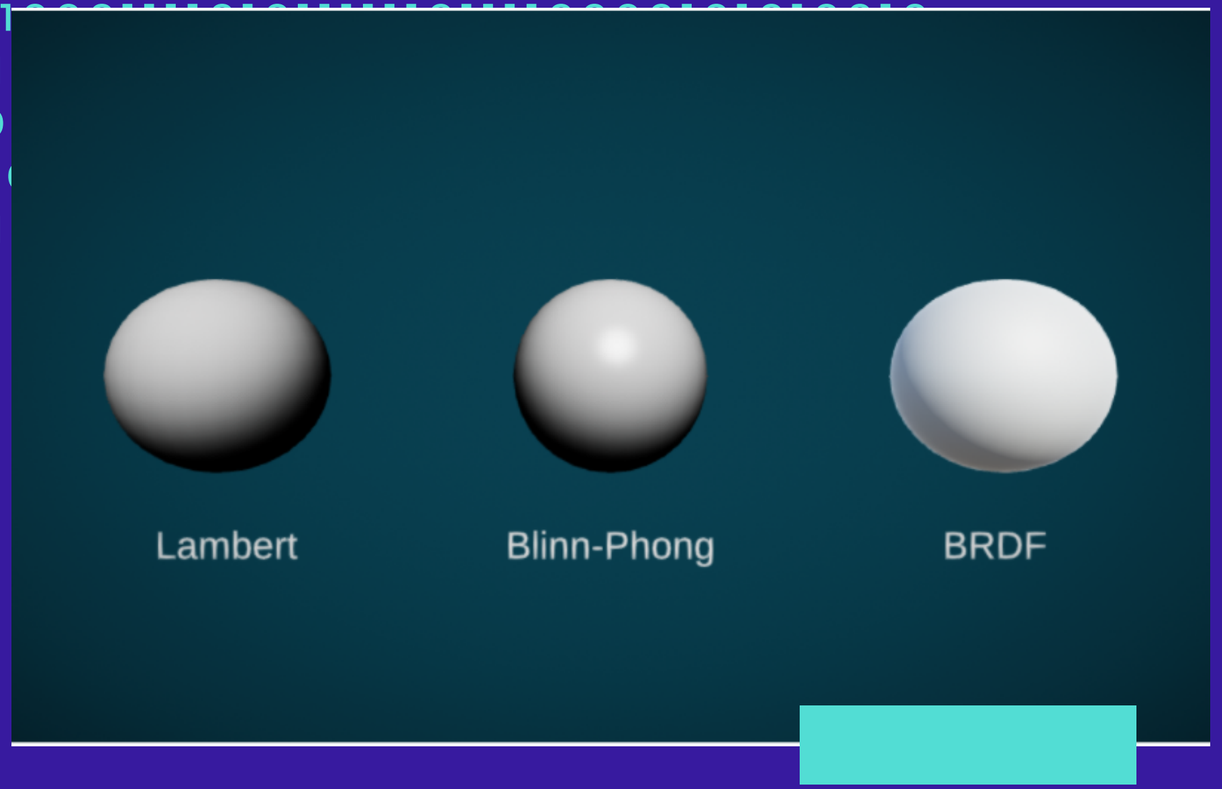
Sergio Peñaloza

Concepto General



A : Normal map en espacio de objeto
B : Modelo high poly
C : Modelo low poly usando face normals
D : Modelo low poly con normal map aplicado





Algunas Ecuaciones de Iluminación que utilizan normales

IMPLEMENTACION (VERTEX/FRAGMENT SHADERS)

```
//This function uses tex2D, use it only in fragment shader
//Assuming the normal map texture channels RGB match Normal Map XYZ
float3 CalculateNormals(sampler2D normalMapTexture, float2 textureSamplingUvs, float3 meshNormal, float3 meshTangent)
{
    //"Tangent, Binormal, Normal"
    //Tangent to mesh normal matching space
    //transformation matrix
    const float3x3 TBN = transpose(float3x3(meshTangent, cross(meshNormal, meshTangent), meshNormal));
    const float3 textureNormal = tex2D(normalMapTexture, textureSamplingUvs).xyz * 2.0 - 1.0;
    return mul(TBN, textureNormal);
}
```

RETOS

1

Implementar un normal map invertido (las protuberancias se convierten en hendiduras) con surface shaders o shader graph.

2

Agregar un parámetro que rote las normales después de ser calculadas con surface shaders o shader graph.