An Open Source Framework for Photo-Realistic Render of Synthetic Elements in Captured Panoramas

Dalai Felinto*, Aldo Zang† and Luiz Velho†

* Fisheries Centre, UBC - Vancouver, Canada  † Visgraf Laboratory, IMPA - Rio De Janeiro, Brazil

Blender Conference, Amsterdam, The Netherlands - October, 2012
w3.impa.br/~zang/blenderconf
Problems of panoramas-only solutions
So, how to make those wonderful Panoramas?
So, how to make those wonderful Panoramas?

Environment capture
Panorama calibration
Scene reconstruction
Scene depth computation
Illumination setup
Synthetic elements
Integration and rendering
Environment capture
Calibration system
Advantages of a calibration system

- Allow to move important sampling regions off the poles
- Less concern on tripod alignment for picture capturing
- It works with panoramas obtained from the internet
- Optimal aligned axis for the world reconstruction
Scene Reconstruction (IBL Toolkit)
Floor Reconstruction

IBL Reconstruction
- Draw Polygon
- Draw Square
- Draw Circle
- Draw Rectangle
vec3 glUnprojectGL( vec2 coords) {
    float u = coords.s * 2.0, 1.0 ;
    float v = coords.t * 2.0, 1.0 ;
    vec4 view = vec4 (u , v , 1.0 , 1.0 ) ;
    vec4 world = projectionmodelviewinverse * vec4 (view . x , -view . y, view . z , 1.0 ) ;

    return vec3 (world [ 0 ]* world [ 3 ], world [ 1 ]* world [ 3 ], -world [ 2 ]* world [ 3 ] ) ;
}

vec2 equirectangular( vec3 vert)
{

    float theta = asin (vert.z) ;
    float phi = atan (vert . x , vert . y) ;
    float u = 0.5 * (phi / PI) + 0.25 ;
    float v = 0.5 + theta / PI ;

    return vec2 (u , v) ;
}

void main( void )
{
    vec2 coords = gl_TexCoord [ 0 ] . st ;
    vec4 foreground = texture2D (color_buffer , coords) ;
    vec3 world = glUnprojectGL(coords) ;
    vec4 background = texture2D (texture_buffer , -equirectangular( normalize (world) )) ;
    float depth = texture2D (depth_buffer, coords).s ;
    if (depth > 0.99995 ) foreground = background ;

    gl_FragColor = foreground ;
}
Results
Conclusions

The Light-positional approach gives the correct world based accounts for rendering.

One pass rendering solution. We don’t need to make compositions or work with layers.

Extensible to Cycles render or other renderers.

Light-depth map idea can be used for real time rendering in other contexts (games, interactive visualization, etc.)

We can handle camera rotations and translations!

It’s possible apply deformation of mesh and texture deformation to the modeled environment.
Extra 1 - Environment Texture
Extra 2 - Camera Travelling
Extra 3 - Fisheye
Acknowledgements

Djalma Lucio
Juliano Kestenberg

Blender and Luxrender Developers

Sam Schad
www.blendedskies.com
Questions?
An Open Source Framework for Photo-Realistic Render of Synthetic Elements in Captured Panoramas

Dalai Felinto*, Aldo Zang† and Luiz Velho†

* Fisheries Centre, UBC - Vancouver, Canada † Visgraf Laboratory, IMPA - Rio De Janeiro, Brazil

Blender Conference, Amsterdam, The Netherlands - October, 2012