Crash Prevention App

EC601 - Product Design

Henry Xia & Zihao Dai

What we do on sprint 4

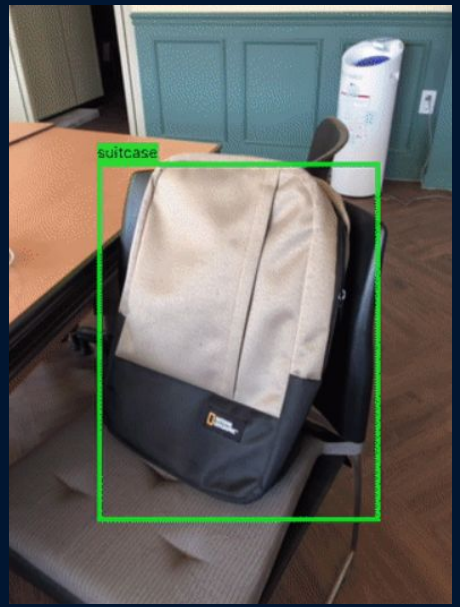
- Demo an application, which can display the distance between the object and the camera.

Achievements

- We have initially realized the effect of displaying the distance on the depth map, even though it is a black and white image.

Displace(code part)

```
class ViewController: UIViewController, ARSessionDelegate {
    @IBOutlet weak var sceneView: ARSCNView!
    @IBOutlet weak var imageView: UIImageView!
    let configuration = ARWorldTrackingConfiguration()
```



```
// Display a 2D texture.
vertex ColorInOut planeVertexShader(Vertex in [[stage_in]])
{
    ColorInOut out;
    out.position = float4(in.position, 0.0f, 1.0f);
    out.texCoord = in.texCoord;
    return out;
}

// Shade a 2D plane by passing through the texture inputs.
fragment float4 planeFragmentShader(ColorInOut in [[stage_in]], texture2d<float,
    access::sample> textureIn [[ texture(0) ]])
{
    constexpr sampler colorSampler(address::clamp_to_edge, filter::linear);
    float4 sample = textureIn.sample(colorSampler, in.texCoord);
    return sample;
}

// Shade a 2D plane by using the length of the values that are encoded in the RGBA channels.
fragment half4 planeFragmentShaderCoefs(ColorInOut in [[stage_in]], texture2d<float,
    access::sample> textureIn [[ texture(0) ]])
{
    constexpr sampler colorSampler(address::clamp_to_edge, filter::linear);
    float4 sample = textureIn.sample(colorSampler, in.texCoord);
    half a = length(sample.rgb);
    half b = abs(sample.a);
    return half4(a+b, b, b, 1);
}
```

```
// Shade the point cloud points by using quad particles.
fragment half4 pointCloudFragmentShader(
    ParticleVertexInOut in [[stage_in]])
{
    // Avoid drawing particles that are too close, or filtered particles that
    // have zero depth.
    if (in.depth < 1.0f)
        discard_fragment();
    else
    {
        return in.color;
    }
    return half4();
}


// Convert the Y and CbCr textures into a single RGBA texture.
kernel void convertYCbCrToRGBA(texture2d<float, access::read> colorYtexture [[texture(0)]],
                                texture2d<float, access::read> colorCbCrtexture
                                    [[texture(1)]],
                                texture2d<float, access::write> colorRGBTexture
                                    [[texture(2)]],
                                uint2 gid [[thread_position_in_grid]])
{
    float y = colorYtexture.read(gid).r;
    float2 uv = colorCbCrtexture.read(gid / 2).rg;
```

The process is:

Display a 2D texture

→ Giving the distance

→ Convert color by Jet color scheme

→ Color the depth value of the texture

→ Shading Point Cloud Points by Using Quadrilateral Particles

What did not work

- We haven't been able to blend all the modules together yet, so the current image is black and white.
- Alerter has not been added yet

RESOURCES

- Smart Traffic Management System | Smart Traffic system | FaststreamTech
- ARKit 6 - Augmented Reality - Apple Developer
- How a Laser Rangefinder Works (Explained!) | Outdoor Empire
- Artificial Intelligence in Tesla Vehicles | Xaltius
- What is Automatic Emergency Braking (AEB)? - Basic Guide (caradas.com)
- Xcode 14 Overview - Apple Developer
- https://github.com/TokyoYoshida/ExampleOfiOSLiDAR
- https://developer.apple.com/documentation/arkit/arframe/3566299-scenedepth
- https://www.wwdcnotes.com/notes/wwdc20/10611/
- Why does the human eye see more shades of green than any other colour? - Quora
- MATLAB | Converting a Grayscale Image to Binary Image using Thresholding - Image-processing (topitanswers.com)
- tucan9389/ObjectDetection-CoreML: An example running Object Detection using Core ML (github.com)

Thanks

Crash Prevention App

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.