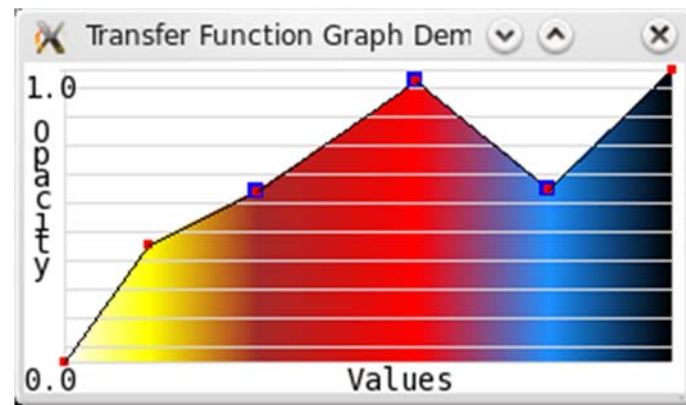


PRE-INTEGRATION

2011. 5. 2.

Transfer Function Design

- C^0 continuity
 - Color values between control points are interpolated linearly



- Exercise
 - 64, (1.0, 0.384313, 0.384313, 0.0);
 - 128, (1.0, 1.0, 0.705882, 1.0);
 - 192, (1.0, 1.0, 1.0, 1.0);
 - 255, (1.0, 1.0, 1.0, 0.0);

2D Look-up Table

```
// for each element of LUT
float4 BuildPreIntLUT2D(front, back)
{
    float4 output = 0;
    float distance = back-front;

    for(s=front; s<back; s++)
    {
        float4 color = ColorFromTransferFunction(s);
        color.a = 1 - pow(1-color.a, 1/distance);
        color.rgb *= color.a;

        output += (1-output.a)*color;
    }

    return output;
}
```

2D Look-up Table

```
// CPU side (pseudo code)
ID3D11Texture2D *pPreIntLUT2DTexture;
MakeTexture(pPreIntLUT2DTexture);

float4 *pLUTData;
BuildPreIntLUT(pLUTData);

Update(pPreIntLUT2DTexture, pLUTData);
SetShaderResource(pPreIntLUT2DTexture);

Render();
```

2D Look-up Table

```
// GPU side
...
Texture2D preintTexture: register(t2);
...
float4 PreIntPS(float4 Position : SV_POSITION, float2 TexCoord : TEXCOORD0) : SV_TARGET
{
    // set start, end, raydirection, etc
    ...

    int sf, sb;
    sf = GetValue(start);
    float4 color = 0;
    float4 output = 0;

    for(int n=1; n<rayCount; n++)
    {
        float3 pos = start + n*rayUnit;
        sb = GetValue(pos);
        color = preintTexture.Load(int3(sf, sb, 0));

        // shading
        color = Shading(color, normal, rayDirection);
        output += (1-output.a)*color;

        ERT();
        sf = sb;        // to use next ray segment
    }

    return output;
}
```

1D Look-up Table

```
// CPU side (pseudo code)
ID3D11Texture1D *pPreIntLUT1DTexture;
MakeTexture(pPreIntLUT1DTexture);

float4 *pLUTData;
// build via accumulating
... (assignment)

Update(pPreIntLUT1DTexture, pLUTData);
SetShaderResource(pPreIntLUT1DTexture);

Render();
```

1D Look-up Table

// GPU side

```
...
Texture1D preintTexture: register(t2);
...
float4 PreIntPS(float4 Position : SV_POSITION, float2 TexCoord : TEXCOORD0) : SV_TARGET
{
    // set start, end, raydirection, etc
    ...

    int sf, sb;
    sf = GetValue(start);
    float4 color0 = 0;
    float4 color1 = 0;
    float4 output = 0;

    color0 = preintTexture.Load(int2(sf, 0));

    for(int n=1; n<rayCount; n++)
    {
        float3 pos = start + n*rayUnit;
        sb = GetValue(pos);
        color1 = preintTexture.Load(int2(sb, 0));

        float4 color = (color1-color0)/(float)(sb-sf);

        // shading
        color = Shading(color, normal, rayDirection);
        output += (1-output.a)*color;

        ERT();
        sf = sb;          // to use next ray segment
        color0 = color1;
    }

    return output;
}
```

1D Look-up Table

- What if s_f is equal to s_b ?
 - Do the same as post-classification