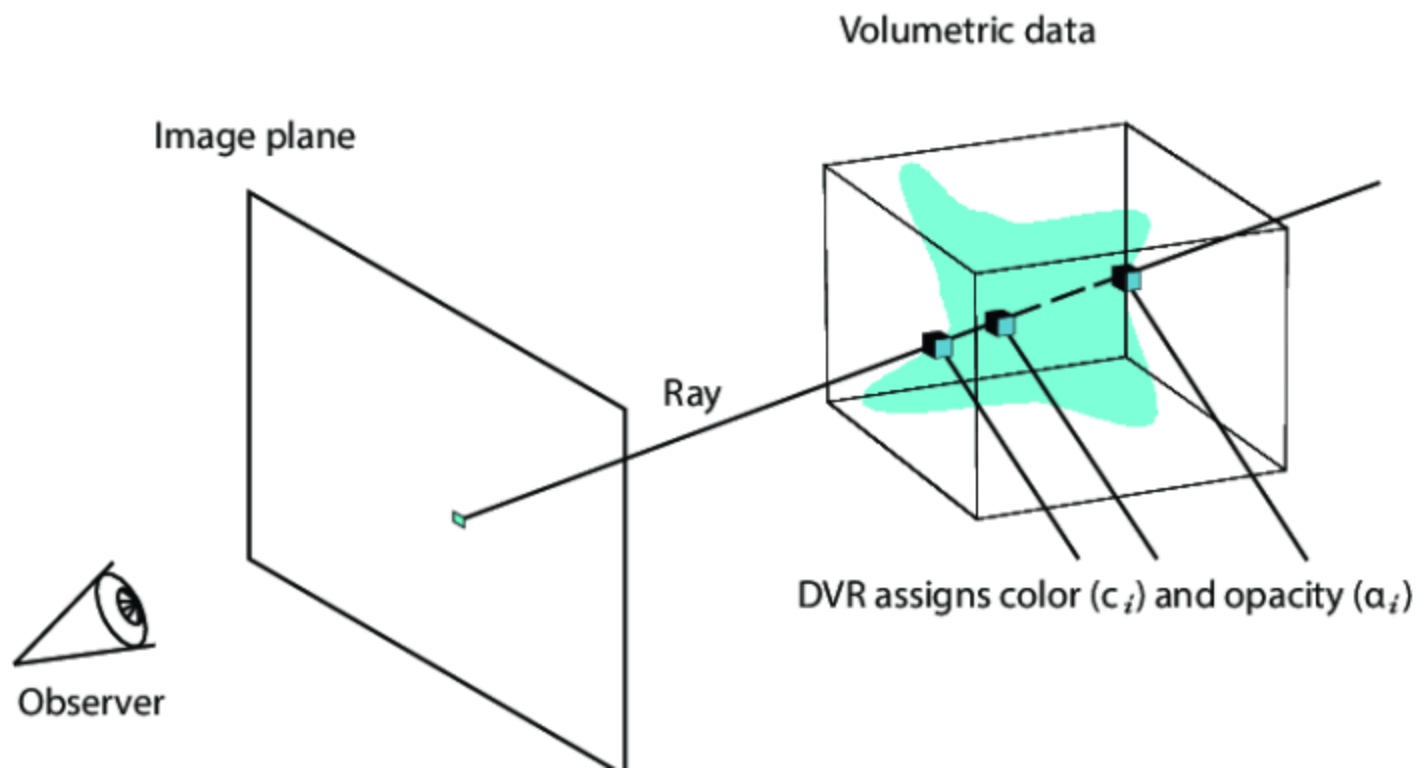


Magisterský Projekt

Dynamic Lightning in Direct Volume Rendering

Matej Lorinc



- Aktuálne používaná metóda DVR:

$$C = \int_{t_{near}}^{t_{far}} c(\mathbf{r}(t)) \cdot \sigma(\mathbf{r}(t)) \cdot T(t) dt$$

- C je výsledná farba pixelu
- c je farba daného samplu v priestore (transferová funkcia)
- σ je priehľadnosť daného bodu (transferová funkcia)
- T je naakumulovaná transmitancia
- s tým že pre diskkrétne samplovanie $T_i = 1 - \sigma_i$

```

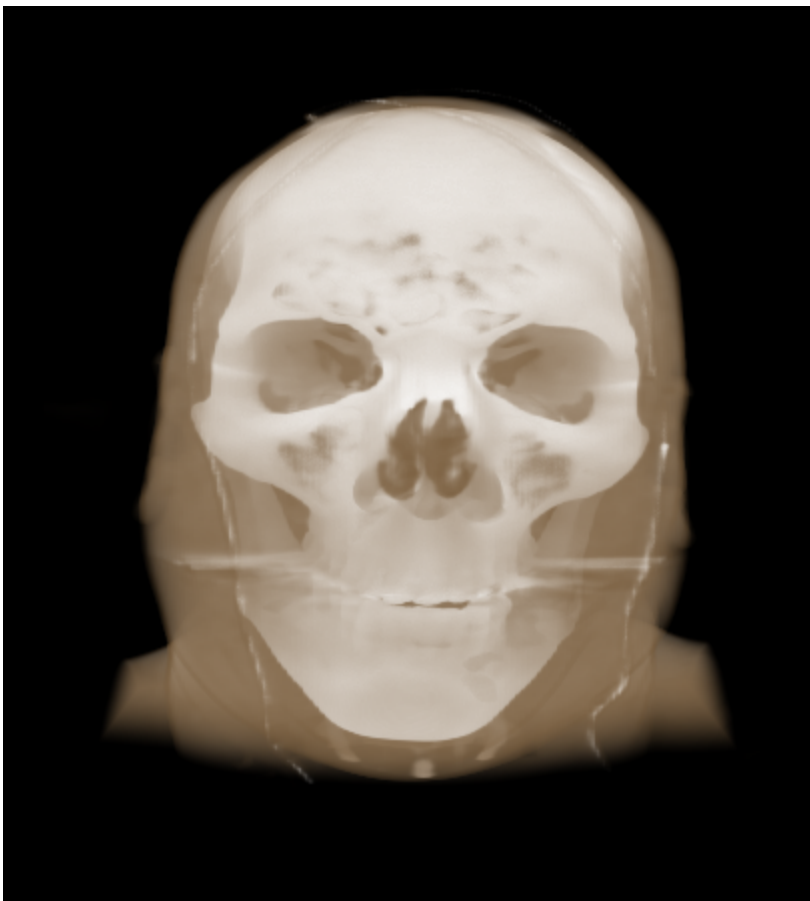
raymarchInfo.numStepsRecip = 1.0 / raymarchInfo.numSteps;
for (int iStep = 0; iStep < raymarchInfo.numSteps; iStep++)
{
    const float t = iStep * raymarchInfo.numStepsRecip;
    const float3 currPos = lerp(ray.startPos, ray.endPos, t);
    const float density = getDensity(currPos);
    float4 src = getTF1DColour(density);

    ...

    float blendFactor = 1.0f / _SamplingRateMultiplier;
    src.a = 1.0f - pow(1.0f - src.a, blendFactor);           // s. rate correction
    src.rgb *= src.a;                                         // b = c * sigma
    col = (1.0f - col.a) * src + col;                         // T * b
                                                            // T accumulation happens as
                                                            // (1.0f - col.a) * src

    ...
}
volrend_result output;
output.colour = col;

```



- Jediné zdroje svetla sú body skenu v priestore

- Modifikácia:
 - Externý zdroj svetla + Model interakcie svetla so skenom
- Optický model:
 - Ambientné osvetlenie (rovnomerné pre celý objem)
 - Difúzne odrazy (nezávislé od uhla pohľadu)
 - Spekulárne odrazy (závislé od uhla pohľadu)

$$I = k_a c + k_d(\hat{n} \cdot \hat{l})c + k_s(\hat{r} \cdot \hat{v})^\alpha$$

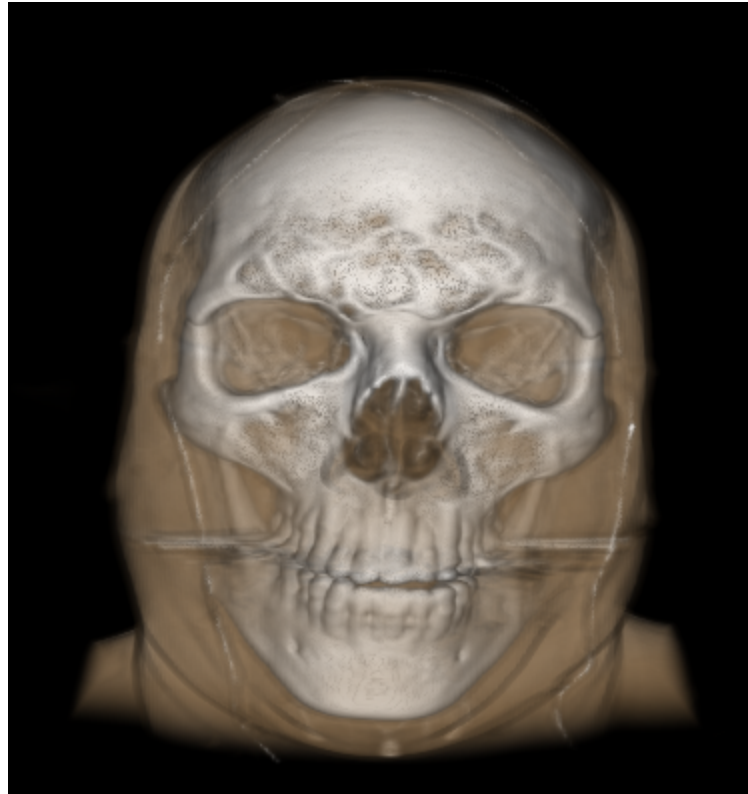
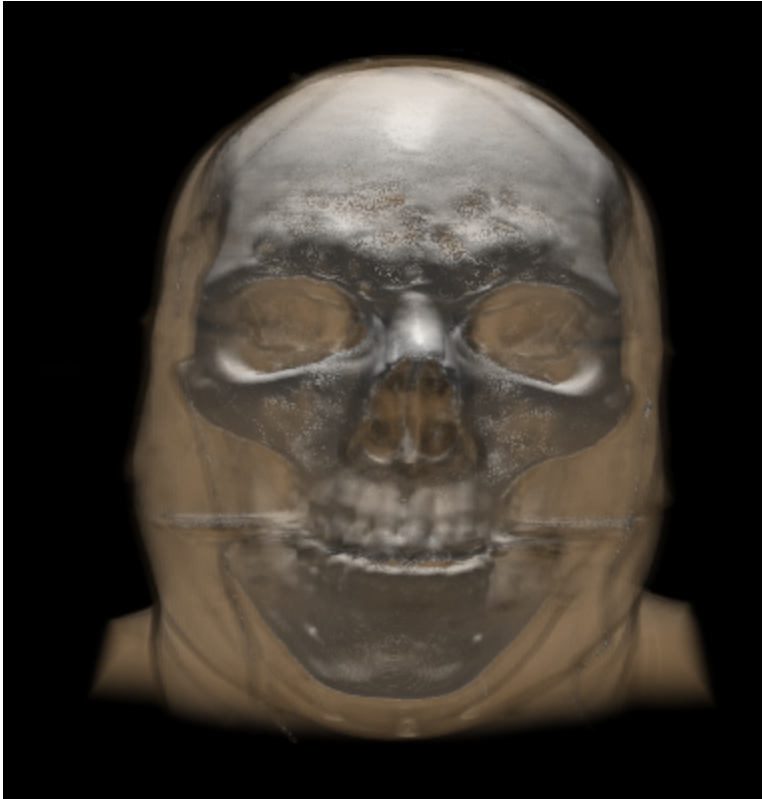
- k ovládajú silu každého komponentu
- n, l, r, v sú normála, uhol zdroju svetla, odraz svetla, uhol pohľadu
- na výpočet n používame disk. priestorový gradient

- Aktuálna verzia

```
float3 calculateLighting(float3 col, float3 normal, float3 lightDir, float3 eyeDir, float specularIntensity)
{
    normal *= (step(0.0, dot(normal, eyeDir)) * 2.0 - 1.0);    // handles and flips normals facing away
                                                                // can be either 1 or -1

    float ndotl = max(dot(normal, lightDir), 0.0f); // normals facing away contribute zero diffuse
    float3 diffuse = ndotl * col;
    float3 ambient = AMBIENT_LIGHTING_FACTOR * col;
    float3 v = eyeDir;
    float3 r = normalize(reflect(-lightDir, normal));
    float rdotv = max( dot( r, v ), 0.0 );
    // fixed 32 shininess factor, float3(1.0f, 1.0f, 1.0f) controls color of the shine
    float3 specular = pow(rdotv, 32.0f) * float3(1.0f, 1.0f, 1.0f) * specularIntensity;
    float3 result = diffuse + ambient + specular;
    return float3(min(result.r, 1.0f), min(result.g, 1.0f), min(result.b, 1.0f));
}
```

- Dva rôzne uhly zdroja svetla



Ďakujem za pozornosť