Vykreslování sledováním paprsku s antialiasingem

Jan Wozniak <xwozni00@stud.fit.vutbr.cz>

17. prosince 2012

1 Úvod

Úkolem projektu bylo vytvořit jednoduchý raytracer v libovolném programovacím jazyku využívající tři metody pro odstranění aliasu. Pro implementaci jsem zvolil jazyk Haskell, antialiasingové metody nadvzorkování, mip mapy a bilinearní interpolace.

2 Teorie

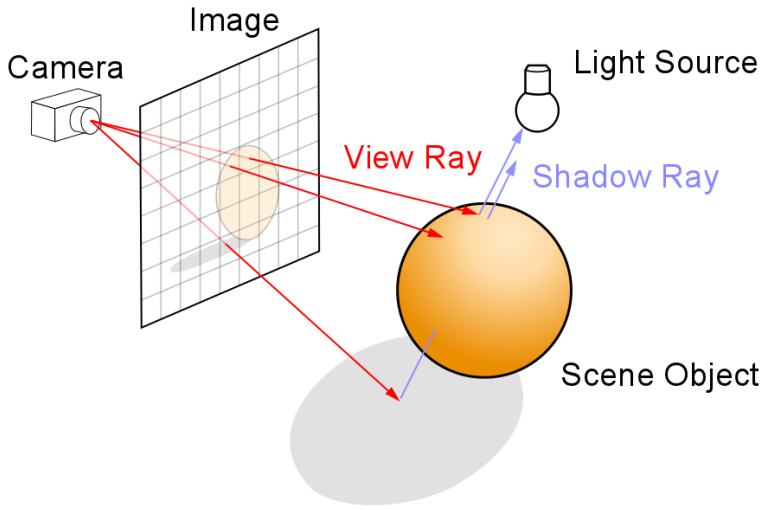
Raytracing je renderovací metoda jak z 3D scény udělat realistické 2D zobrazení. Rekurzivní princip kdy jsou z kamery vysílány paprsky přes zobrazovací plátno do scény s modely objektů[1]. Na obrázku 1 je znázorněno, jak tento princip funguje. V základní implementaci se dělí paprsky do tří skupin

- pohledový – z kamery do scény
- odražený – z objektu sceny do objektu scény
- stínový – z objektu scény ke světlu

Pro výpočet průsečíku paprsků s jednoduššími objekty je možno použít základní rovnice analytické geometrie a vektorové algebry, které jsou k dispozici například v dokumentu Vector Algebra for Ray Tracing [2].

Při použití textur může docházet v obraze k nežádoucímu efektu zvanému *aliasing*. V případě, že objekt má texturu obsahující příliš častou změnu barev, a objekt je ve velké vzdálenosti kde paprsky nevzorkují povrch objektu dostatečně často, jeví se pozorovateli textura velmi zkresleně. Pro zmírnění tohoto efektu lze použít mimo jiné použít [1]

- mip mapy – několik textur s ruzným rozlišením v jediné textuře
- supersampling – nadvzorkování pixelu posíláním paprsků v rozích pixelu
- bilinearní interpolaci – nalezení přibližné hodnoty barvy ve spojitém prostoru z diskrétní textury



Obrázek 1: Schéma sledování paprsků do scény, převzatý z wikipedie [3]

3 Popis řešení

Jelikož Haskell je ryze funkcionální jazyk, bez použití monád je kód velmi transparentní. Kód je rozdělen do modulů `Math.hs`, `Phong.hs`, `Raytracer.hs`, `Scene.hs`, `Data_types.hs` a `main.hs`. Jádro celého algoritmu tkví ve funkci `raytracer` a `ray_trace`.

```
raytracer :: [Color]
raytracer = map (rt) view_grid
    where
        rt point = ray_trace get_depth
            (Ray (vec point) camera_position point)
        vec point = normalize (point `sub` camera_position)
```

Funkce `raytracer` vrhá paprsky do scény a pro výpočet výsledné barvy volá funkcí `ray_trace`. Funkce vrací list barev, který je předán funkci `png`, která z listu vytvoří obrázek formátu png.

```
ray_trace :: Int -> Ray -> Color
ray_trace depth ray@(Ray vector source dest)
    | on_ray_intersections == [] = background_color
    | otherwise = (sum_phong (apply_shade get_phong))
                    + ref_color
    where
        all_intersections = concat (map
            (get_intersections ray) scene)
        intersections = map_filter_distance dest
            all_intersections
        on_ray_intersect = filter
            ((is_point_on_ray ray) . (point . snd))
            intersections
        closest_intersection = snd (minimum on_ray_intersect)
        get_phong = phong closest_intersection
            camera_position light_position
        shade = get_shade (point closest_intersection)
        apply_shade (PhongColor a d s) =
            (PhongColor a ((fst shade) `mul_color` d)
            ((snd shade) `mul_color` s))
        ref_color = reflection (depth-1) closest_intersection
```

Funkce `ray_trace` bere dva parametry, hloubku raytracingu a paprsek. Vrací výslednou barvu pro daný vržený paprsek.

Z procedurálních textur je implementována pouze 3D šachovnice, a to funkcí funkce `checked_texture_procedural`, pro výpočet souřadnic neporcedurální textury a její namapování na objekt, což je převod 3D souřadnice do 2D, slouží funkce `uv_map`. Mip mapy a bilinearní filtrování jsou jako samostatný typ materiálu, stejně jako všechny ostatní typy textur. Při výpočtu supersamplingu je ve funkci `raytrace` před funkí `ray_trace` volána funkce `supersample`, která paprsek nadvzorkuje, počítá průměr a v případě potřeby nadvzorkuje znovu. Je možné použít kombinace více různých metod pro potlačení aliasu v rámci jednoho obrázku.

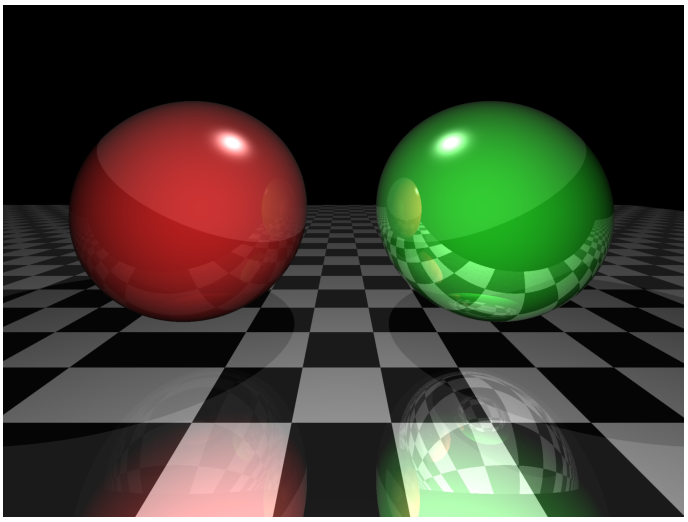
Implementováno byl phongův osvětlovací model, pomocí funkce se signaturou:

```
phong :: Intersection -> Dim3 -> Dim3 -> PhongColor
```

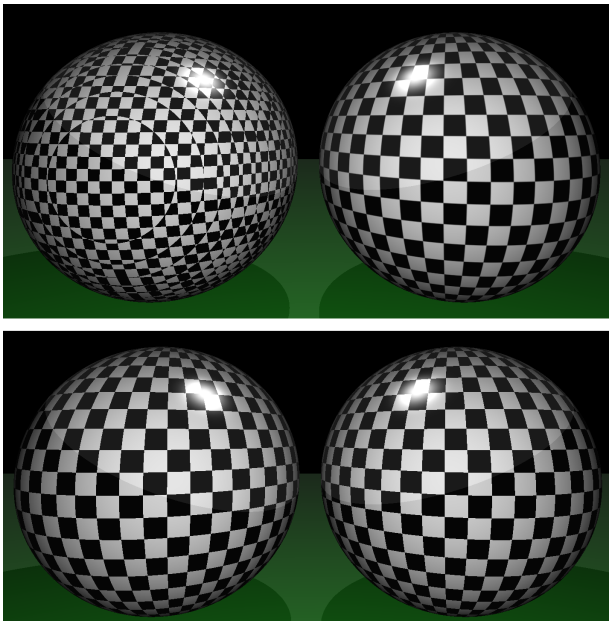
Jako parametry bere průsečík, pozici světla a pozici kamery, vrací barvu tohoto průsečíku.

4 Vyhodnocení

Zde je uvedeno pár vybraných obrázků demonstrujících implementované techniky a tabulka s časy renderování.

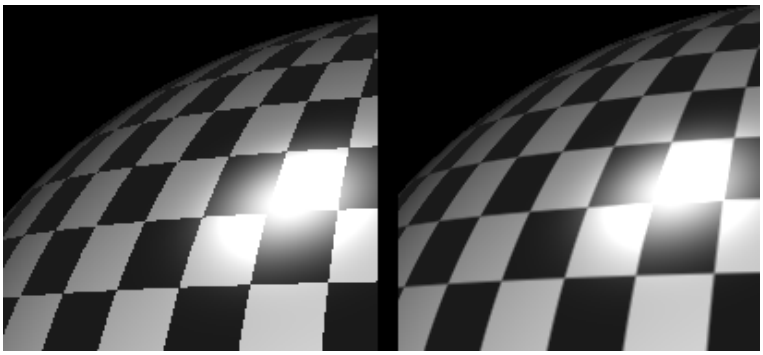


Obrázek 2: Renderovaný obrázek pomocí supersamplingu.



Obrázek 3: Srovnání čtyř textur procedurální, bilineární interpolace, běžná textura, mip mapa.

Obrázek 4: Srovnání mip mapy s běžnou texturou.



Obrázek 5: Bilineární interpolace.

Renderován byl obrázek 1000 x 1000 pixelů s hloubkou rekurze 5.

Metoda	Doba výpočtu
raytracing	1 min 23 sec
supersampling	2 min 5 sec
bilinear interpolation	1 min 28 sec

Tabulka 1: Srovnání základní metody, supersamplingu a bilineární interpolace.

5 Závěr

Použité metody pro potlačení aliasu jsou patrné i bez důkladného prozkoumání, zpomalení algoritmu je vzhledem k původní době výpočtu přijatelné.

Literatura

[1] A. Herout. *Počítačová grafika – studijní opora.* 2008-12-08.

[2] *Vector Algebra for Ray Tracing.* [cit. 2012-11-24] <http://tom.cs.byu.edu/~455/rt.pdf>

[3] *Wikipedia, The Free Encyclopedia.* [cit. 2012-11-24] <http://en.wikipedia.org/>