

Vykreslování sledováním paprsku s antialiasingem

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

9. 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 bilineární interpolace.

2 Teorie

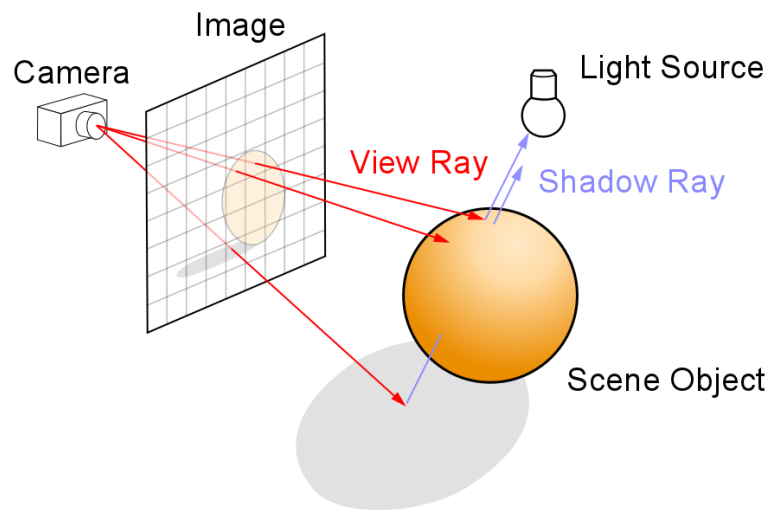
Raytracing je renderovací metoda jak z 3D scény udělat realistické 2D zobrazení. Rekursivní princip kdy jsou z kamery vysílány paprsky přes zobrazovací plátno do scény s modely objektů. Na obrázku 1 je znázorněno, jak tento princip funguje. V základní implementaci se dělí paprsky do tří skupin

- pohledový – z kamery do scény
- odražený – z objektu scény 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* [?].

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

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



Obrázek 1: Schéma sledování paprsků do scény, převzatý z wikipedie [2]

3 Popis řešení

Jelikož Haskell je ryze funkcionální jazyk, bez použití monád je kód velmi transparentní. 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 funkci `ray_trace`. Funkce vrátí 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)) + reflection_color
where
  all_intersections = concat (map
    (get_intersections ray) scene)
  intersections = map_filter_distance dest
    all_intersections
  on_ray_intersections = filter
    ((is_point_on_ray ray) . (point . snd))
    intersections
  closest_intersection = snd (minimum on_ray_intersections)
  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))
  reflection_color = reflection (depth-1) closest_intersection

```

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

Procedurální textury jsou implementovány pomocí funkce , pro výpočet souřadnic neporcedurální textury a její namapování na objekt, což je převod 3D souřadnice do 2D, slouží funkce `uv_map`. Mip mapy a bilineární filtrování jsou jako samostatný typ materiálu, stejně jako všechny ostatní typy textur. Při výpočtu supersamplingu je ve funkci `raytrace` před funkcí `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

Tady by mělo být napsané jak to funguje. Protože se jedná o počítačovou grafiku nebo vidění, tak by tady měl být screenshot, ze kterého bude poznat jak to funguje. K tomu by měla být ideálně tabulka s vyhodnocením jak přesně/rychle to funguje.

5 Závěr

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

Literatura

- [1] *Vector Algebra for Ray Tracing*. [cit. 2012-11-24]
<<http://tom.cs.byu.edu/~455/rt.pdf>>
- [2] *Wikipedia, The Free Encyclopedia*. [cit. 2012-11-24]
<<http://en.wikipedia.org/>>