Michał Kaczara *181132*

Grafika komputerowa i komunikacja człowiek - komputer

Projekt:
Metoda śledzenia promieni (Ray Tracing)

Prowadzący: dr inż. Marek Woda
Grupa: czwartek, godz. 10:15-13:00

POLITECHNIKA WROCŁAWSKA – STYCZEŃ 2012

1. Cel i założenia projektu

W ramach projektu należało zaimplementować algorytm rekursywnego śledzenia promieni (ang. Recursive Ray Tracing). Wcześniej należało przeanalizować działanie programu realizującego najprostszą metodę śledzenia promieni – *Ray Casting*. Podczas realizacji projektu obowiązywały następujące założenia:

- scena składa się ze sfer i źródeł światła,
- jako sposób rzutowania przyjmuje się rzutowanie równoległe,
- współczynniki a, b i c określające wpływ odległości źródła światła na oświetlenie punktu powierzchni mogą być wpisane w kodzie programu,
- opis sceny zadany jest w pliku tekstowym złożonym z kolejnych linii. Każda z linii pliku zawiera słowo kluczowe i odpowiednie dane.

2. Algorytm rekursywnego śledzenia promieni

W porównaniu z przedstawionym w przykładowym programie algorytmem *Ray Casting*, algorytm rekursywnego śledzenia promieni ma inny zakres śledzenia analizowanego promienia. W prostym algorytmie promień biegnący od obserwatora, przez punkt na rzutni w głąb sceny, śledzony był tylko do ewentualnego pierwszego przecięcia z obiektem sceny. W rekursywnej metodzie śledzenia promieni analizowany promień po trafieniu w pierwszy obiekt sceny śledzony jest dalej. Po trafieniu promienia w obiekt, wylicza się kierunek promienia odbitego i sprawdza, czy nie trafia on w kolejny obiekt itd. Oczywiście w implementacjach śledzenie takie musi być w jakiś sposób ograniczone, aby nie dopuścić do zapętlenia się algorytmu w sytuacji dla przykładu, gdy jedno lustro odbija się w drugim. Śledzenie (wyliczanie kolejnych kierunków biegu promienia) prowadzone jest w oparciu o geometrię analizowanej sceny i określony z góry sposób oddziaływania obiektów i światła.

Algorytm śledzenia promieni opisywany jest w różny sposób. Dla potrzeb projektu programistycznego najlepiej jest wykorzystać zapis algorytmu w postaci rekurencji.

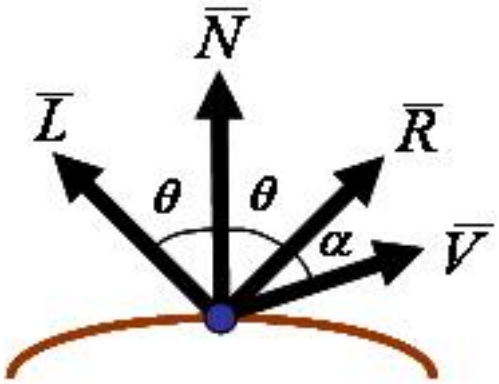
3. Opis zaimplementowanego rozwiązania

3.1 Opis aplikacji

W zaimplementowanym rozwiązaniu funkcją nadrzędną i realizującą wspomnianą rekurencję jest funkcja `Trace()`. Wyznacza ona kolor piksela dla promienia zaczynającego się w punkcie `p` (pierwszy parametr funkcji) i biegnącego w kierunku wyznaczonym przez wektor `v` (drugi parametr funkcji). Rekurencja wykonywana jest `step` razy (trzeci parametr funkcji). W ciele funkcji najpierw sprawdzany jest fakt przecięcia promienia z jakimkolwiek obiektem na scenie(funkcja `Intersect()` i warunek). Jeśli takie przecięcie nastąpiło, to następuje wyznaczenie wektora normalnego do powierzchni obiektu w danym punkcie (funkcja `Normal()`), następnie wyliczenie wektora kierunku odbicia promienia (funkcja `Reflect()`) i w końcu wyznaczenie oświetlenia punktu powierzchni.

Ostatnią funkcjonalność realizuje funkcja `Phong()`, wyliczająca oświetlenie punktu za pomocą modelu Phonga. Przyjmuje ona dwa parametry – numer obiektu i wektor kierunkowy promienia. Model Phonga służy do obliczania oświetlenia punktu leżącego na powierzchni obiektu 3-D dla dość ogólnego przypadku, zarówno w sensie oświetlenia, jak i charakterystyki powierzchni obiektu. Pozwala on na połączenie własności rozpraszania

światła i odbicia kierunkowego oraz na intuicyjne powiązanie zależności otrzymywanego efektu oświetlenia z wartościami liczbowymi parametrów. Model Phonga zadany jest wzorami pozwalającymi na obliczenie trzech składowych (R, G, B) intensywności oświetlenia analizowanego punktu powierzchni obiektu. Wzory te ściśle zależą od wartości parametrów i wektorów jednostkowych zaprezentowanych na Rysunku 1.



Rysunek 1. Wektory i parametry w modelu Phonga

Wektor `N` jest wektorem normalnym do powierzchni w danym punkcie. Wektor `L` oznacza kierunek padania światła na oświetlany punkt. Wektor `R` określa kierunek odbicia promienia dla idealnego zwierciadła, a wektor `V` – kierunek obserwacji punktu.

Wspomniana wcześniej funkcja `Intersect()` wyliczająca współrzędne punktu przecięcia promienia z obiektem przyjmuje dwa parametry – punkt `p` będący początkiem promienia i wektor kierunkowy promienia. Funkcja zwraca numer obiektu, który jest przecięty przez promień. Ponieważ promień może przecinać kilka sfer, wprowadzono zmienną kontrolną `length`, która przechowuje odległość między punktem początkowym promienia, a punktem przecięcia ze sferą. Dzięki temu funkcja zwraca numer obiektu położonego najbliżej punktu początkowego promienia. Zwrócony numer obiektu przyjmuje jako parametr funkcja `Normal()`.

Implementacja algorytmu wymagała zdefiniowania funkcji pomocniczych, np. metody `Scalar()` obliczającej iloczyn skalarny dwóch wektorów, metody `Normalization()` odpowiedzialnej za normalizację wektora, czy funkcji `ReadSceneFromFile()` wczytującej plik z opisem sceny.

3.2 Opis formatu pliku z definicją sceny

Plik z opisem sceny, która ma zostać wygenerowana przez program jest plikiem tekstowym. Każda z linii zawiera definicję jednego parametru sceny, poprzedzoną odpowiednim słowem kluczowym:

- *dimensions* – definicja rozmiarów sceny, szerokość i wysokość, np.:
  `dimensions 400, 400`

- *background* – definicja koloru tła, składowe: R, G i B, np.:
  `background 0.2, 0.2, 0.2`

- *global* – dane opisujące globalne światło rozproszone. Składowe: R, G i B określające intensywności świecenia źródła, np.:
  `global 0.1, 0.1, 0.1`

- *sphere* – dane opisujące sferę. Promień obiektu i współrzędne środka, następnie współczynniki materiałowe powierzchni dla otoczenia, światła rozproszonego,

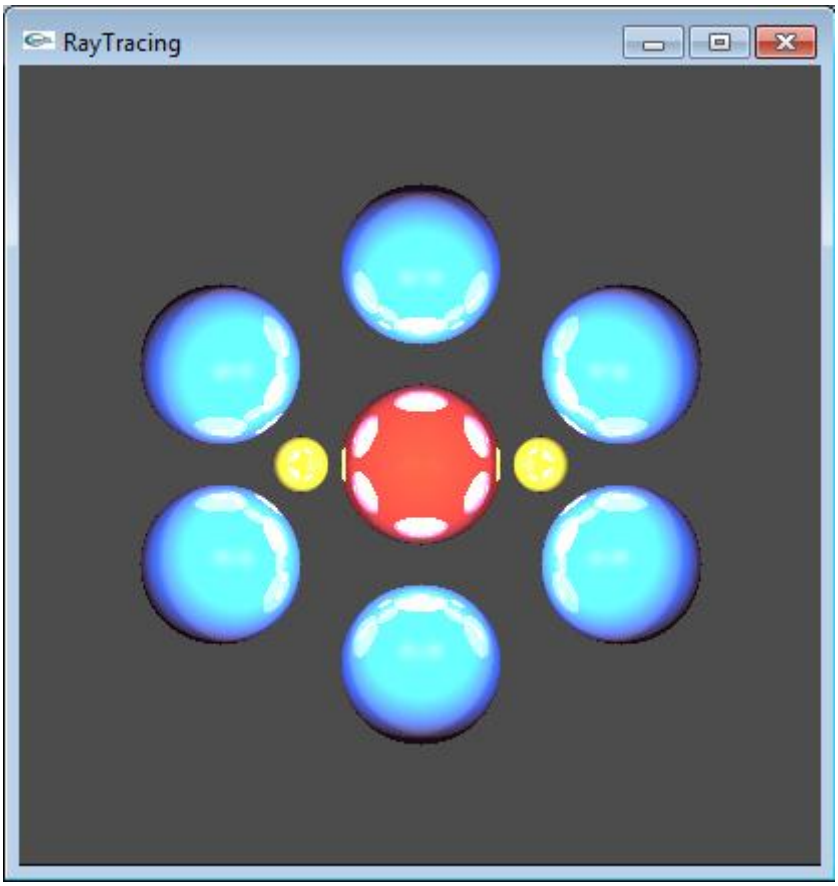
kierunkowego; współczynnik połysku, np.:
```
sphere 0.7 3.0 0.0 -5.0 0.8 0.2 0.0 0.7 1.0 0.0 0.3 0.2 0.2 40
```

- *source* – dane opisujące źródło światła. Współrzędne punktu źródła, składowe: R, G i B określające intensywności świecenia źródła dla światła otoczenia, światła rozproszonego i kierunkowego, np.:
```
source -5.0 0.0 12.0 0.2 0.2 0.2 0.0 1.0 1.0 0.4 0.5 0.3
```

Program po uruchomieniu pozwala na wybór nazwy pliku ze sceną.

3.3 Wynik działania programu

Wynik działania zaimplementowanego algorytmu przedstawia Rysunek 2. Scena wczytana do programu jest identyczna jak udostępniona przez prowadzącego na stronie ZSK.



Rysunek 2. Wynik działania programu RayTracing

4. Wnioski

Metoda rekurencyjnego śledzenia promieni *Ray Tracing* jest obecnie najlepszym znanym mechanizmem oświetlania scen 3-D. Zaimplementowanie aplikacji było ciekawym zdaniem, a dzięki dobremu wprowadzeniu z jego realizacją nie miałem problemów. Myślę, że we wnioskach warto wymienić wady wspomnianej metody. Pierwszą i dosyć ważną jest duża złożoność obliczeniowa. Druga wada to brak rozpatrywania wszystkich kierunków padania światła na powierzchnię obiektów, co może powodować błędy w wyznaczaniu oświetlenia. Trzecią wadą są występujące w przypadku śledzenia nieskończenie wąskich promieni efekty aliasingowe.