

Automatyczne obliczenia fotometryczne przy użyciu oprogramowania Radiance - raport

Jakub Porzycki *porzycki@agh.edu.pl*

Robert Lubaś *rlubas@agh.edu.pl*

Marcin Mycek *mycek@agh.edu.pl*

17 grudnia 2013

1 Radiance - silnik do obliczeń fotometrycz- nych

Po wykonaniu analizy dostępnych rozwiązań, wytypowano silnik Radiance do przeprowadzenia obliczeń fotometrycznych w projekcie. Radiance jest to profesjonalne narzędzie do wizualizacji oświetlenia w wirtualnym środowisku. Narzędzie to jest rozwijane od 1985 roku do dzisiaj. Pierwszy raz zostało udostępnione w 1989 roku i od wielu lat jest wykorzystywane w komercyjnych projektach architektonicznych. Było prezentowane na licznych prestiżowych konferencjach m.in. SIGGRAPH oraz TVCG

Wszystkie powyżej wymienione fakty potwierdzają, że Radiance to wypróbowane i rozbudowane narzędzie do obliczeń fotometrycznych. Spośród innych dostępnych rozwiązań wyróżnia się szerokim i udokumentowanym użyciem w komercyjnych projektach - istnieje na rynku od 24 lat.

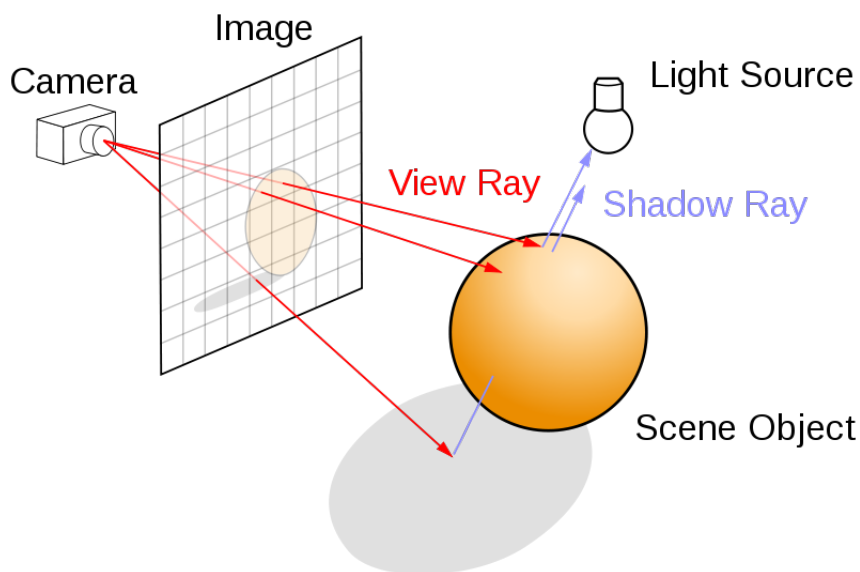
Obecnie Radiance jest rozwijany w Lawrence Berkeley National Laboratory, znajdującym się na University of California. Od dłuższego czasu jest również częścią OpenStudio - profesjonalnego środowiska do analizy i projektowania budynków. W ramach OpenStudio utrzymywana jest skompilowana aktualna wersja Radiance dla Windowsa[1].¹

¹Oryginalnie Radiance jest oprogramowaniem działającym w systemie operacyjnym linux.

2 Zasada działania

Działanie Radiance opiera się na wstecznym śledzeniu promieni. Metoda ta została przedstawiona w 1980 roku [2]. Od tego czasu była wielokrotnie udoskonalana - szczególnie ważna jest praca [3]. W Radiance używana jest poprawiona wersja algorytmu [4] ².

Algorytm wstecznego śledzenia w przeciwieństwie do śledzenia postępowego symuluje promienie światła poruszające się przeciwnie do kierunku (zwrotu) ich ruchu w rzeczywistym świecie. Śledzenie promieni zaczyna się od obserwatora. Przez każdy piksel wirtualnego ekranu prowadzone są promienie, które poruszają się po scenie. Gdy promień napotka jakiś obiekt, wtedy z promienia pierwotnego generowane są promienie wtórne: odbite i załamane - w zależności od materiału z jakiego jest zbudowany obiekt. W Radiance do obliczonego oświetlenia dodawany jest wpływ światła odbitego od najbliższych sąsiadów.



Rysunek 1: Ogólna zasada działania algorytmu wstecznego śledzenia promieni.

Radiance oblicza parametry oświetlenia w pomieszczeniu korzystając z reprezentacji przestrzeni w postaci drzewa ósemkowego. Jest to sposób podziału trójwymiarowej przestrzeni na mniejsze, regularne fragmenty. Obszar symulacji otaczany jest sześciannym, który dzielony jest rekurencyjnie na

²Autorom, nie udało nam się znaleźć informacji o późniejszej, znaczącej aktualizacji algorytmu ray-tracingu w przedstawianym systemie.

osiem mniejszych sześciątów. Przy czym podział dokonywany jest tylko w przypadku kiedy w danym sześciacie znajdują się jakieś obiekty.

3 Instalacja

Kolejne kroki instalacji Radiance dla Windowsa przedstawiają się następująco:

- Ściągnij skompilowaną wersję Radiance 4.2 ze strony:
<https://openstudio.nrel.gov/accept/file/1043>
Dostępna jest wersja dla systemów 32 oraz 64 bitowych. Pobranie oprogramowania wymaga utworzenia darmowego konta w serwisie.
- Uruchom instalator. Przejdź przez proces instalacji zgodnie z podanymi instrukcjami.
- Po instalacji pod ścieżką C:\Program Files\Radiance lub inną wybraną, znajdują się dwa katalogi:
 - bin
 - lib
- Ścieżkę do katalogu **bin** dodaj do zmiennej **PATH**
- Utwórz zmienną systemową **RAYPATH**, dodaj do niej ścieżkę do katalogu **lib**.
- Może być konieczne ponowne zalogowanie się lub restart systemu.

Rozszerzoną wersję instrukcji można znaleźć na stronie:
<https://openstudio.nrel.gov/getting-started-developer/getting-started-radiance>

4 Opis sceny

Opis sceny w Radiance tworzony jest przy użyciu plików zawierających opisy obiektów według zdefiniowanej składni. Znajduje się on w plikach tekstowych (ASCII text) i może zawierać cztery rodzaje elementów w następującym formacie [5]:

1. # komentarz

Komentarze mogą zawierać dowolny tekst, aby dana linia tekstu mogła zostać uznana za komentarz musi rozpoczynać się znakiem #.

```

2. modyfikator typ nazwa
   n S1 S2 S3 ... Sn
   0
   m R1 R2 R3 ... Rm

```

Podstawowe elementy opisu sceny mają wszystkie ten sam, przedstawiony powyżej format. Posiadają modyfikator, typ oraz nazwę. Modyfikator to albo wcześniej zdefiniowany element, albo *void*. Nazwą może być dowolna sekwencja znaków (bez znaków białych). Typ określa jakiego rodzaju element jest definiowany, spośród różnych powierzchni, materiałów, tekstur oraz wzorów. Kolejne 3 linie to argumenty specyficzne dla danego typu elementu. W kolejnych liniach znajdują się argumenty będące łańcuchem znaków, liczbami całkowitymi (nie używane) oraz liczbami rzeczywistymi. Pierwszy znak w linii oznacza liczbę argumentów.

```

3. modyfikator alias nazwa odnośnik

```

Możliwe jest również jednego elementu na bazie już zdefiniowanego. Można to zrobić definiując alias elementu z innym modyfikatorem.

```

4. ! komenda

```

Linie rozpoczynające się od znaku: *!*, są interpretowane jako komendy do wykonania. Są to podprogramy, których wynik działania jest umieszczany w opisie sceny. Pozwalają na przyspieszenie tworzenia jej opisu.

Przyjrzyjmy się opisowi kilku elementów:

- Materiał, czerwony plastik:

```

void plastic red_plastic
  0
  0
  5 .7 .05 .05 .05 .05
  # red green blue speculariry roughness #

```

- Kula wykonana z czerwonego plastiku:

```

red_plastic sphere ball
  0
  0
  4 1 1 .625 .2
# xcent ycent zcent radius #

```

- Materiał - światło. W Radiance źródło światła jest definiowane jako specjalny typ materiału pokrywający świecące obiekty.

```
void light bright
    0
    0
    3 200 200 200
    # red_emission green_emission blue_emission
#
```

- Lampa - kula wykonana z materiału *światło*:

```
bright sphere fixture
    0
    0
    4 1 5 2.75 .125
    # xcent ycent zcent radius
```

- Chrom - powierzchnia chromowana:

```
void metal chrome
    0
    0
    5 .8 .8 .8 .9 0
    # red green blue speculariry roughness
#
```

- Chromowany cylinder

```
chrome cylinder fixture_support
    0
    0
    7
    3 3 2.75
    3 3 3
    .05
```

- Deklaracja materiału - niebieskiego plastiku oraz wykorzystanie komendy *genbox* do wygenerowania prostopadłościanu o wymiarach 2 x 2 x 0.1 oraz przesuniętego o wektor 0,4,0.5.

```
void plastic blue_plastic
    0
```

```

0
5 .1 .1 .6 .1 .1

!genbox blue_plastic box 2 2 0.1 | xform -t
0 4 0.5

```

Wszystkie polecenia jakie mogą znaleźć się w opisie sceny są szczegółowo wyjaśnione w [5].

5 Podstawowe polecenia

Radiance jest w rzeczywistości zbiorem kilkudziesięciu programów. Po zainstalowaniu i odpowiednim ustawieniu zmiennych systemowych **PATH** oraz **RAYPATH** możemy je wywoływać z linii komend. Szczegółowy opis można znaleźć w [6].

Do podstawowych obliczeń potrzebna jest znajomość zaledwie kilku programów. Zakładając, że definicja sceny znajduje się w pliku *room.rad*, możemy wykonać następujące polecenie:

```
oconv room.rad > room.oct
```

Program *oconv* konwertuje opis sceny na reprezentację przestrzeni przy użyciu drzewa ósemkowego.

Mając już plik *room.oct*, przy użyciu programów *rpict* oraz *rvu* możemy podglądać widok symulacji. *rpict* produkuje obrazek w formacie .hdr³, *rvu* pozwala na interaktywny podgląd. *Rvu* możemy wywołać w następujący sposób:

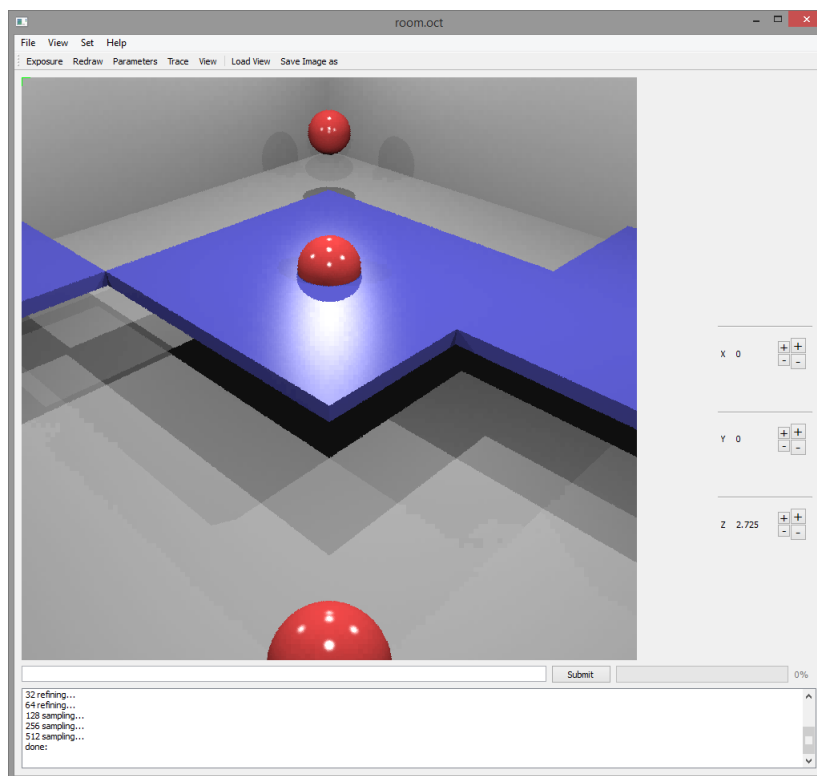
```
rvu -vp 2.25 .375 1 -vd .25 .125 -.125 -av .5 .5 .5 room.oct
```

Parametr *-vp* określa punkt, w którym znajduje się obserwator, natomiast *-vd* kierunek patrzenia. Przykładowy widok z programu *rvu* jest przedstawiony na rys.: 2

Powyżej wymienione programy: *rpict* oraz *rvu* pozwalają na wizualną ocenę oświetlenia. Aby uzyskać dane liczbowe niezbędne jest skorzystanie z programu *rtrace*, śledzi on zadane wektory. Przykładowe wywołanie ma postać:

```
rtrace -ov room.oct < samples.inp > pomiar.out
```

³Do obsługi tego formatu a XnView oraz XnConvert



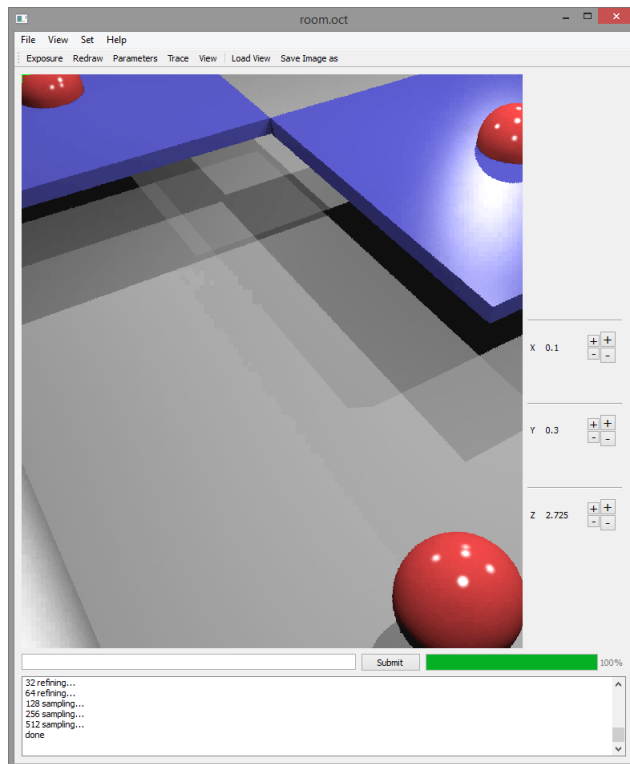
Rysunek 2: Testowe pomieszczenie. Widok z programu *rvu*.

W pliku `samples.inp` znajdują się definicje wektorów, a w pomiar `out` otrzymujemy odpowiadające im natężenie światła lub inne wartości zgodnie z parametrami wywołania programu. Szczegóły obliczania natężenia światła w następnym rozdziale.

6 Obliczanie natężenia światła - przykład

Aby przeprowadzić obliczenia natężenia światła, zdefiniowano pokój o wymiarach 6 x 6 x 23 metry. W tym pokoju znajduje się pięć identycznych lamp, pięć czerwonych kul oraz trzy niebieskie prostopadłości, patrz. rys ??

W tak zdefiniowanym pomieszczeniu, przeprowadzono obliczenia natężenia światła. Przy pomocy programu *rtrace* próbkowano natężenie światła w wybranych punktach. Zdecydowano się użyć wektorów skierowanych prostopadłe w dół. Punkty obserwacji znajdowały się na wysokości 2.5 metra. Z tych punktów, śledzono natężenie światła w punkcie, gdzie wektora skierowany prostopadłe w dół z tego punktu, przecinał jakiś obiekt. Punkty rozmieszczono na kwadratowej siatce, w odstępach 10 cm.

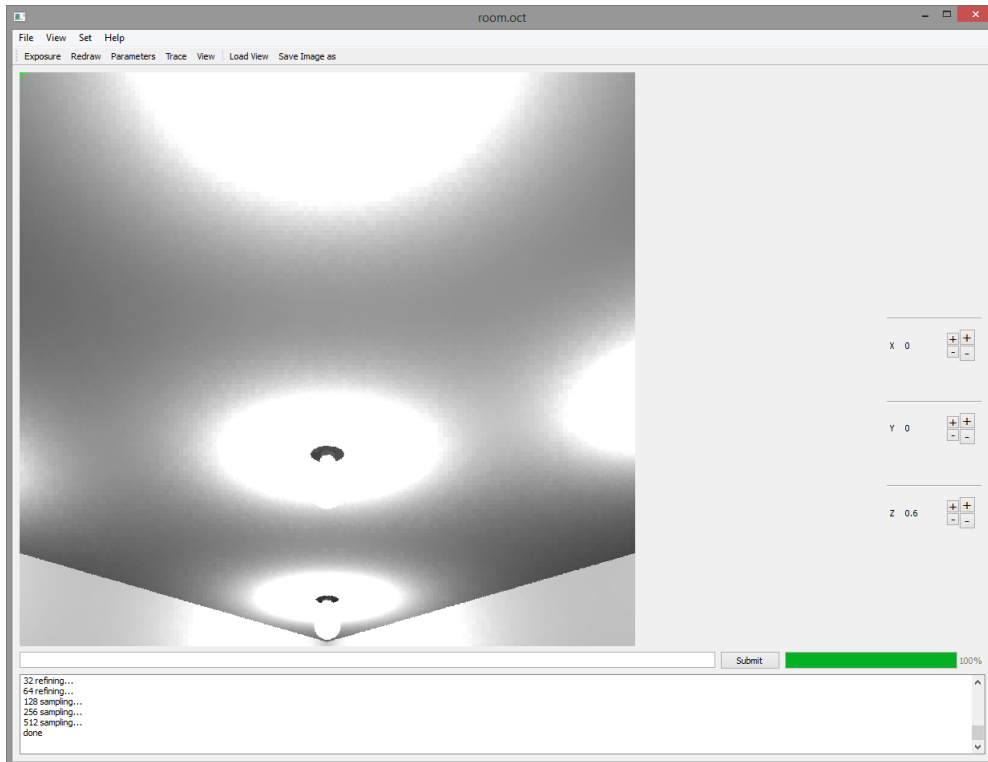


Rysunek 3: Testowe pomieszczenie. Widok z programu *rvu*.

Poniżej znajduje się fragment pliku *samples.inp*, który definiuje całość danych wejściowych do programu *rtace*. Pierwsze trzy wartości definiują wektor położenia, kolejne trzy kierunek śledzonego promienia.⁴.

```
0 5.5 2.5 0.00000001 0.00000001 -1
0 5.6 2.5 0.00000001 0.00000001 -1
0 5.7 2.5 0.00000001 0.00000001 -1
0 5.8 2.5 0.00000001 0.00000001 -1
0 5.9 2.5 0.00000001 0.00000001 -1
0 6 2.5 0.00000001 0.00000001 -1
0.1 0 2.5 0.00000001 0.00000001 -1
0.1 0.1 2.5 0.00000001 0.00000001 -1
0.1 0.2 2.5 0.00000001 0.00000001 -1
0.1 0.3 2.5 0.00000001 0.00000001 -1
0.1 0.4 2.5 0.00000001 0.00000001 -1
```

⁴Radiance nie pozwala śledzenie promieni prostopadłych do osi, dla tego użyto wartości 0.00000001, co nie powinno powodować znaczących różnic



Rysunek 4: Testowe pomieszczenie. Widok z programu *rvu*.

W wyniku wykonania programu *trace* z tak zdefiniowanym wejściem, dla każdego punktu otrzymaliśmy plik, który dla każdego wcześniej zdefiniowanego wektora, zawiera trzy wartości - strumień światła, czerwonego, zielonego i niebieskiego wychodzący z punktu, w który „trafił” wektor.

Aby otrzymać ostateczną wartość luminancji w danym punkcie konieczne jest przeliczenie luminancji zgodnie ze wzorem:

$$L = 47.4 * Rr + 120 * Rg + 11.6 * Rb \quad (1)$$

gdzie Rr , Rg oraz Rb to otrzymane wcześniej wartości.

Przy użyciu programu *rcalc*, możemy bezpośrednio przeliczyć otrzymane wartości na natężenie światła.

```
rtrace -ov room.oct < samples.inp
| rcalc -e $1=47.4*$1+120*$2+11.6*$3
```

W efekcie otrzymujemy listę wartości natężenia światła w próbkowanych punktach.

Literatura

- [1] <http://www.radiance-online.org/download-install/installation-information/windows-installation>.
- [2] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, June 1980.
- [3] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *SIGGRAPH Comput. Graph.*, 18(3):137–145, 1984.
- [4] Gregory J. Ward. The radiance lighting simulation and rendering system. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 459–472, New York, NY, USA, 1994. ACM.
- [5] *The Radiance 4.1 Synthetic Imaging System*, 2011.
- [6] *Radiance Manual*. <http://radsite.lbl.gov/radiance/whatis.html>.

7 Apendyks A - generowanie opisu sceny przy użyciu plików .obj

Możliwe jest wygenerowanie opisu sceny na podstawie modelu sceny z programów do tworzenia grafiki 3D, eksportujących wyniki do formatu .obj (np. Blender). Taka forma tworzenia może przydać się do szybkiego testowania, lub tworzenia reprezentacji obiektów, które ze względu na stopień skomplikowania są trudne do opisanie za pomocą tekstu.

Poniżej przedstawiono kolejne kroki jakie należy wykonać, aby utworzyć opis sceny lub elementów sceny przy użyciu Blendera lub innego programu do tworzenia grafiki trójwymiarowej.

Proces tworzenia sceny (Blender):

- W programie Blender tworzymy lub otwieramy istniejącą scenę/element sceny (*.obj).
- Jeżeli zachodzi potrzeba nanosimy poprawki: skala, ułożenie, ... itd.
- Eksportujemy projekt do formatu Wavefront „obj”.

Proces tworzenia formatu wejściowego do Radiance:

- Posiadamy plik *.obj.
- obj2rad konwertuje plik obj do opisu sceny w radiance.
- Otwieramy konsolę w katalogu z plikiem *.obj i wykonujemy polecenie:

```
> obj2rad nazwa.obj > nazwa.rad
```

W efekcie otrzymujemy opis sceny w znanym formacie, z którym możemy postępować dokładnie tak samo jak z innymi plikami opisu sceny. Mogą pojawić się błędy spowodowane brakiem zdefiniowania materiałów (należy dopisać definicje materiałów w utworzonym pliku). Jeżeli wczytujemy całą scenę należy także zdefiniować światło.

8 Apendyks B - zawartość płyty

1. Folder *Wyniki* - arkusz z obliczonymi wartościami luminacji dla przykładowego pomieszczenia, zrzuty ekranu, plik ze zdefiniowanymi wartościami punktu zaczepienia i kierunkami wektora
2. Folder *Instalacja* - plik instalacyjny pakietu Radiance.

3. Folder *Raport* - zawiera dwa dokumenty w formie elektronicznej: Radiance - silnik do obliczeń fotometrycznych, oraz Unabiguous mapping between hypergraph and grid representations of architectural objects.