Streszczenie – czyli na czym będę bazował i co będę rozwijał. Praca docelowo będzie pisana w LaTeXie. Każdy z rozdziałów zostanie podzielony na podrozdziały.

1. Wstęp.

Raytracing jest techniką służącą do generowania realistycznych obrazów scen 3D. Na przestrzeni lat technika ta ciągle się rozwijała. Doczekała się wielu modyfikacji, które usprawniają proces generowania realistycznej grafiki. Takimi technikami są między innymi PathTracing, PhotonMapping, Radiostity i wiele innych. Wszelkie rodzaje Raytracingu wykorzystuje się w grafice komputerowej jak i w kinematografii do generowania jak najwierniejszych foto realistycznych scen 3D. Czas generowania pojedynczej klatki/ujęcia takiej sceny niekiedy potrafi być liczony nawet w godzinach. Dlatego technika ta nie doczekała się jeszcze swojej wielkiej chwili w przemyśle rozrywkowym jakim są np. gry komputerowe oraz inne aplikacje generujące grafikę 3D w czasie rzeczywistym.

2. Cel pracy.

Celem niniejszej pracy jest przeniesienie a zarazem zrównoleglenie algorytmu śledzenia promieni na procesory graficzne (GPU). Celem także jest przyśpieszenie obliczeń standardowego Wstecznego Raytracingu w celu jak najszybszego generowania scen 3D. W ramach niniejszej pracy napisany został uniwersalny system Raytracingu działający na wielo rdzeniowych procesorach komputerowych (CPU), a także na kartach graficznych typu NVIDIA(GPU), które obsługują technologie NVIDIA CUDA.

3. Wprowadzenie do Raytracingu

W rzeczywistym świecie promienie świetlne rozchodzą się od źródła światła do obiektów znajdujących się w świecie. Gdyby zaadaptować tą metodę do generowania realistycznej grafiki komputerowej, otrzymalibyśmy nieskończenie dokładny obraz. Z racji jednak na to, że sprzęt komputerowy ma ograniczone możliwości, a metoda ta jest bardzo nie efektywną metodą pod względem obliczeniowym. Najszerzej stosowaną metodą śledzenia promieni jest wsteczne śledzenie promieni (backward raytracing). W odróżnieniu od

postępowego algorytmu śledzenia promieni (forward raytrcing), które opiera się na generowaniu jak największej liczby promieni dla każdego źródła światła. Algorytm wstecznego śledzenia promieni zakłada, że promienie śledzone są od obserwatora, poprzez scenę do obiektów z którymi kolidują.

4. NVIDIA CUDA jako znakomita platforma do zrównoleglenia obliczeń.

CUDA jest dość nową technologią wprowadzoną na rynek przez firmę NVIDIA. Udostępnia ona mechanizmy między innymi zrównoleglające obliczenia na ko-procesory znajdujące się w kartach graficznych firmy NVIDIA. Przy pomocy tego mechanizmu jesteśmy wstanie uzyskać znaczne przyśpieszenie w obliczeniach w stosunku do obliczeń na zwykłym procesorze CPU.

(… Ogólnie obszerniejszy opis technologii CUDY. Jak ją stosować, przykłady kodu, opisy zaczerpnięte z dokumentacji, jak pisać kod by szybko się wykonywał w CUDZIE…)

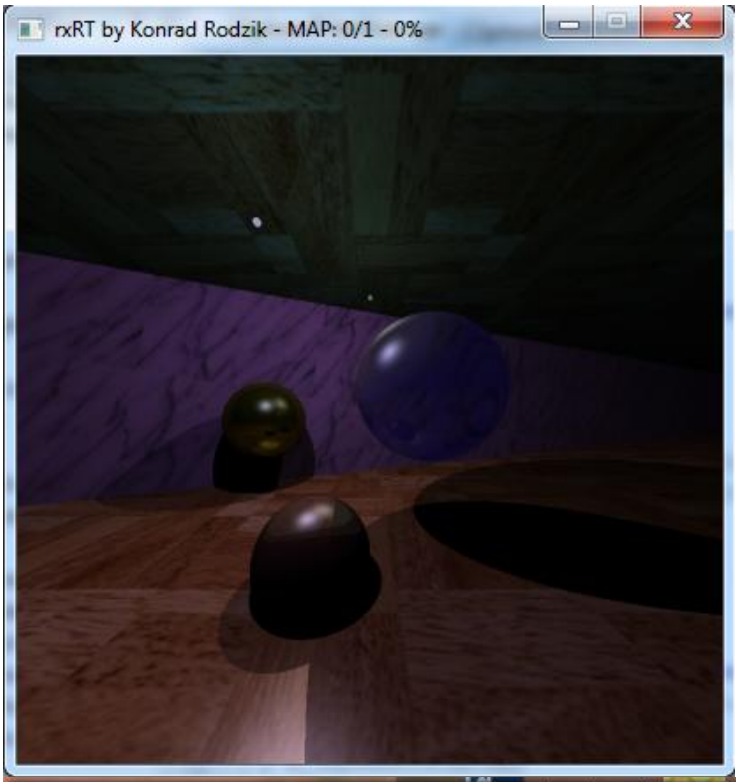
Dodatkowo opisane jak zaadaptowałem wsteczny raytracing przy użyciu technologii CUDA (wycinki kodów źródłowych…)

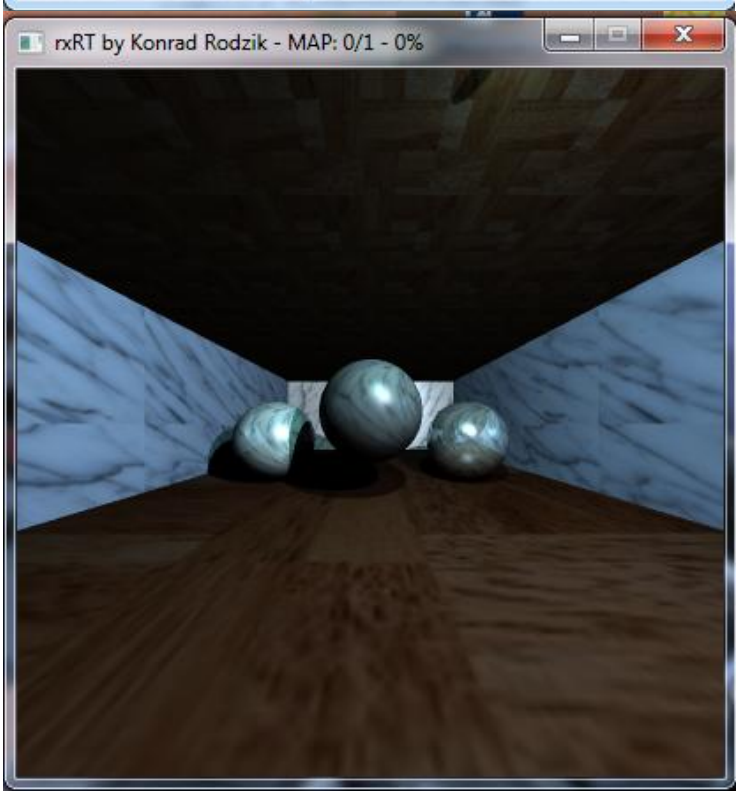
5. Omówienie aplikacji testowej.

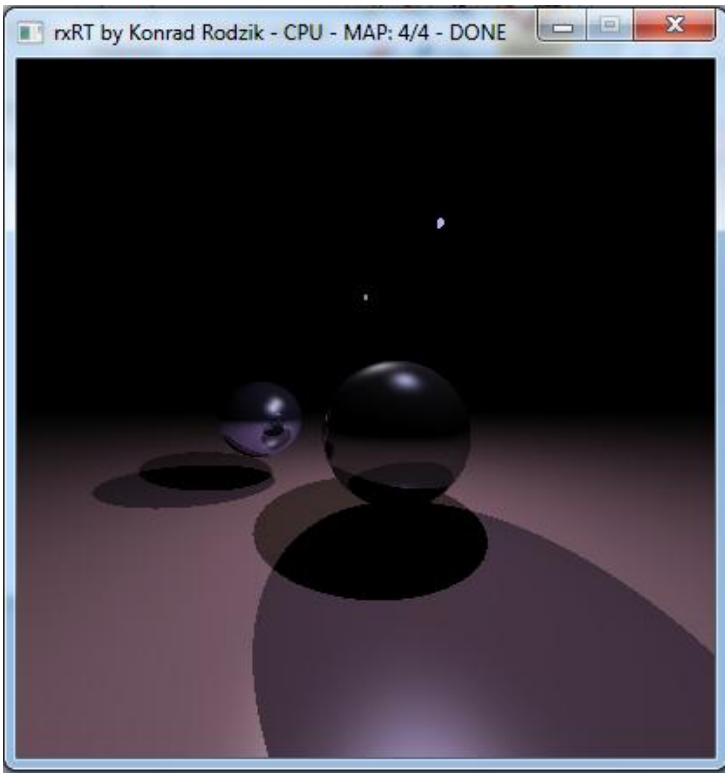
Na potrzeby niniejszej pracy zostało opracowane autorskie rozwiązanie uniwersalnego wstecznego Raytracera działającego zarówno na procesorze CPU jak i również na ko-procesorach graficznych GPU firmy NVIDIA. Aplikacja testowa jest wstanie generować wynikowe obrazy scen 3D składających się z kul, prostopadłościanów oraz płaszczyzn. Na każdy z elementów sceny jest możliwość nałożenia dowolnej tekstury oraz doboru odpowiednich parametrów materiału. Dodatkowo na scenie możliwe jest umieszczanie świateł punktowych. Aplikacja sama w sobie jest benchmarkiem, który potrafi przetestować zadaną liczbe scen 3D na komputerze użytkownika. Zebrane wyniki z obliczeń jest wstanie przesłać na wybrany adres e-mail (w tym przypadku za zgodą użytkownika do developera). Aplikacja przy generowaniu obrazu

sceny 3D bierze pod uwagi różne właściwości materiału danego obiektu. Docelowo generowane są takie efekty jak: oświetlenie, odblask, cienie, wielokrotne odbicia i załamania, tekstury. Przy użyciu materiałów o różnych parametrach jesteśmy wstanie uzyskać bardzo ciekawie wyglądające obiekty np: lustro, szkło, metale i wiele innych.

Przykładowe wygenerowane obrazy:







Aplikacja testowa napisana jest w języku C++ z użyciem biblioteki graficznej DirectX9 przeznaczona jest na systemy z rodziny Windows.

6. Porównanie wydajności Raytracera działającego na CPU oraz na GPU.

Tutaj przedstawię wszelkie wyniki generowania obrazów ze scen oraz wszelkie wartości czasowe im odpowiadające. Porównanie wyników dla raytracingu CPU vs GPU. Z każdych wynikowych danych badań przeprowadzonych na różnych konfiguracjach sprzętowych wygenerowany zostanie wykres przedstawiający różnice czasowe w generowaniu scen a tym samym przyspieszenie wstecznego raytracingu jakie udało się uzyskać.

Zakładam wstępnie testy 10 scen. Na każdej z nich różna konfiguracja obiektów oraz materiałów im nadanych. Chce pokazać jakie właściwości materiałów i jaka ilość prymitywów oraz ich rodzajów wpływa na zmniejszanie/zwiększanie wydajności raytracingu.

7. Podsumowanie i wnioski.

Podsumuje jasna dla wszystkich rzecz, że obliczenia na kartach graficznych przewyższają wydajnością obliczenia na zwykłych wielordzeniowych procesorach komputerowych. Dodatkowo opisze moje przemyślenia i wnioski które objawiły mi się podczas pisania uniwersalnej aplikacji wstecznego raytracingu.
Na koniec opowiem trochę o tym co udało się zrobić, osiągnąć a czego nie i w jaki sposób chciałbym dalej rozwijać prace nad raytracingiem.