Generowanie obrazu metodą śledzenia promieni w czasie rzeczywistym z wykorzystaniem obliczeń równoległych

Politechnika Wrocławska

Mateusz Gniewkowski

DD:MM:RR

Streszczenie

Streszczenie...

Spis treści

Rozdział 1

Wstęp

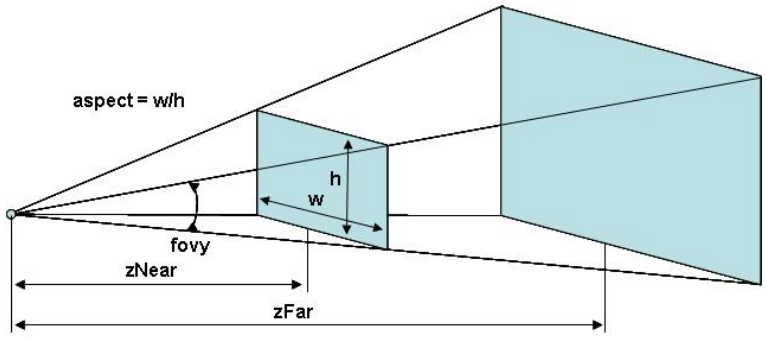
Rozdział 2

Analiza problemu

2.1 Śledzenie promieni

2.1.1 Podstawowy algorytm śledzenia promieni

Metoda śledzenia promieni pozwala określić widoczność obiektów znajdujących się na scenie (a tym samym na generowanie obrazu) na zasadzie śledzenia umownych promieni świetlnych biegnących od obserwatora w scenę. W perspektywicznym rozumieniu sceny (a takiego dotyczy algorytm zaimplementowany na potrzeby tej pracy), pierwszym krokiem algorytmu jest wybranie środka rzutowania (nazywanego okiem obserwatora) oraz rzutni (powierzchnia na której zostanie odwzorowana trójwymiarowa scena). Rzutnię (a właściwie interesujący nas wycinek rzutni - abstrakcyjne okno obserwatora) można podzielić na regularną siatkę, w której każde pole odpowiada jednemu pikselowi ekranie urządzenia (tzw. układ urządzenia). Kolejnym krokiem algorytmu jest wypuszczenie promienia wychodzącego z oka obserwatora, przechodzącego przez dany piksel ekranu i lecącego dalej - w scenę. Kolor piksela jest ustalany na podstawie barwy i oświetlenia najbliższego obiektu (więcej o metodach oświetlenia można przeczytać w rozdziale !TU WSTAW ROZDZIAŁ!), który został przecięty przez wysłany promień. W przypadku braku kolizji piksel przybiera barwę otoczenia.



Poniżej przedstawiono pseudokod podstawowego śledzenia promieni

piksele, obiekty
obj = null
dist = max

wybór środka rzutowania i rzutni

for piksel in piksele **do**
    wyznacz promień
    for obiekt in obiekty **do**
        if promień przecina obiekt i dystans < dist **then**
            obj = obiekt
            dist = dystans
        end if
    end for
end for

ustal kolor piksela na podstawie obj

2.1.2 Rekursywny algorytm śledzenia promieni

2.1.3 Równoległa wersja algorytmu śledzenia promieni

2.2 Wybór technologii

Rozdział 3

Projekt systemu

3.1 Projekt klastra

3.1.1 Master

3.1.2 Slave

3.2 Opis programu

Rozdział 4

Opis wybranych technologii

4.1 C++

4.2 QT

4.3 Standard MPI

Rozdział 5

Implementacja

5.1 Szczegółowy opis klas

5.2 Plik wejściowy

5.3 Warstwa prezentacji

5.4 Warstwa logiki biznesowej

Rozdział 6

Opis funkcjonalny

Rozdział 7

Rezultaty

7.1 Testy wydajnościowe

7.2 Omówienie wyników

7.2.1 Przyspieszenie obliczeń

7.2.2 Obliczenia w czasie rzeczywistym

7.3 Przykładowe obrazy

Rozdział 8

Podsumowanie

Rozdział 9

Dodatek A - obliczenia równoległe

Rozdział 10

Dodatek B - matematyka i algorytmy