

TRAK – Projekty

Autor: Łukasz Dąbała

Wymagania projektu

W ramach projektu należy stworzyć program, który będzie realizował opisane w temacie funkcje. Projekt jest zadaniem zespołowym, gdzie każdy zespół składa się z 4 osób.

Rekomendowanym językiem programowania jest język Python lub C++.

Za projekt można uzyskać maksymalnie $x \times 25p.$, gdzie x to liczba osób w zespole. Każdy z członków zespołu może dostać maksymalnie 25 punktów.

Ocenie w ramach projektu podlegają:

1. Działanie programu - realizacja funkcji oraz wytłumaczenie algorytmów stojących za implementacją w zrozumiały sposób (dotyczy projektów powiązanych z artykułem) (19 p.)
2. Efekty wizualne - prezentacja działania programu oraz kroku algorytmu w przyjemnie wizualny sposób (przygotowanie modeli, scenerii itd.) (2 p.)
3. Jakość kodu (3 p.)
4. Prezentacja wykonana na wykładzie (1 p.)

Projekt musi być pokazany odpowiedniemu prowadzącemu przed terminem ostatniego wykładu. Dodatkowo, brak prezentacji na wykładzie skutkuje niezaliczeniem projektu.

Terminy

Projekt oddawany jest w 2 etapach:

1. prezentacja wstępnego programu / założeń algorytmu
2. prezentacja ostatecznej wersji programu.

Konieczne jest zaprezentowanie projektu odpowiedniemu prowadzącemu 2 razy - brak pierwszej prezentacji uniemożliwia oddanie projektu.

Zadanie	Ostateczny termin
Deklaracja zespołów projektowych	10.11.2024
Przydział projektów	12.11.2024
Prezentacja projektów związanych z artykułami	26.11.2024
Prezentacja projektów związanych z artykułami	3.12.2024
Oddanie pierwszego etapu projektu	13.12.2024
Oddanie ostatecznej wersji programu	24.01.2025
Prezentacja projektów nie związanych z artykułami	28.01.2025

Tabela 1: Terminy projektowe

1 Refrakcja w czasie rzeczywistym

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering refrakcji w przestrzeni obrazu w czasie rzeczywistym.

W programie powinny znaleźć się m.in.:

1. stały podgląd z kamery
2. obsługa mapy środowiska
3. interfejs konsolowy/graficzny, który umożliwi wczytanie siatki trójkątów z pliku, podanie współczynnika załamania oraz wczytanie mapy środowiska
4. implementacja algorytmu z artykułu: Interactive Image-Space Refraction of Nearby Geometry:
http://cwyman.org/papers/graphite05_InteractiveNearbyRefraction.pdf

2 Badanie struktur przyspieszających

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał badanie działania struktur przyspieszających w kontekście śledzenia promieni.

W programie powinny znaleźć się m.in.:

1. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku, wybór algorytmu oraz wybór struktury przyspieszającej
2. implementacja algorytmu śledzenia ścieżek
3. implementacja struktur przyspieszających: BVH (*ang. Bounding Volume Hierarchy*), drzewo-KD oraz siatki jednorodnej

3 Rendering spektralny

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering spektralny (wykorzystanie różnych długości fal).

W programie powinny znaleźć się m.in.:

1. obsługa mapy środowiska
2. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku, zmianę właściwości algorytmu (np. włączenie/wyłączenie wykorzystania spektrum) oraz wczytanie mapy środowiska
3. bazowa implementacja algorytmu śledzenia promieni
4. modyfikacja polegająca na wykorzystaniu fal (sprawdź: *spectral power distribution*).

4 Foveated rendering

Prowadzący: dr inż. Michał Chwesiuk

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering scen algorytmem śledzenia promieni przy uwzględnieniu ustalonego kierunku patrzenia obserwatora.

W programie powinny znaleźć się m.in.:

1. możliwość generowania plików graficznych zawierających render wygenerowanych scen.
2. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku, zmian metod renderingu i uruchomienie renderingu.
3. pomiar czasu renderingu sceny.
4. implementacja dowolnej metody foveated renderingu, t.j. renderingu uwzględniającego kierunek patrzenia.
 - (a) obszar generowanego obrazu powinien być podzielony na trzy obszary:
 - i. obszar centralny (pełna jakość renderingu).
 - ii. obszar peryferyjny (ograniczona jakość renderingu).
 - iii. dalszy obszar peryferyjny (ograniczona bardziej jakość renderingu).
 - (b) każdy obszar powinien charakteryzować się inną ilością śledzonych promieni.
 - (c) wielkość obszarów powinna zostać sparametryzowana (np. ilość pikseli, przeliczenie kątów widzenia na ilość pikseli).
5. Możliwość zmiany kierunku/pozycji patrzenia obserwatora.
6. Opcjonalne: komunikacja z urządzeniem śledzącym wzrok użytkownika.

5 Porównanie metod renderingu

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering różnymi metodami.

W programie powinny znaleźć się m.in.:

1. obsługa mapy środowiska
2. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku, podanie algorytmu renderowania wraz z parametrami oraz wczytanie mapy środowiska
3. implementacja algorytmu śledzenia promieni oraz mapowania fotonów
4. statystyka na temat renderingu np. ilość wyszukiwanych przecięć, ilość wygenerowanych promieni cienia itd.

6 Bokeh rendering

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering efektu typu Bokeh.

W programie powinny znaleźć się m.in.:

1. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku, podanie różnych właściwości algorytmu
2. implementacja algorytmu z artykułu: Real-Time Dynamic Bokeh Rendering with Efficient Look-Up Table Sampling:
<http://cg.skku.edu/pub/papers/2022-jeong-tvcg-bokeh-preprint.pdf>

7 Ray portals

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał edycję przebiegu ścieżki światła z wykorzystaniem portali.

W programie powinny znaleźć się m.in.:

1. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku, podanie różnych właściwości algorytmu, edycję portali
2. implementacja algorytmu z artykułu: RayPortals: a light transport editing framework:
<https://hal.science/hal-01295281v1/preview/2017-TVC-Portal-author-version.pdf>

8 Real-Time Ray-Traced Soft Shadows

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering miękkich cieni z wykorzystaniem metody opisanej w niżej wymienionym artykule.

W programie powinny znaleźć się m.in.:

1. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku
2. implementacja algorytmu z artykułu: Real-Time Ray-Traced Soft Shadows of Environmental Lighting by Conical Ray Culling:
<https://sapphiresoul.github.io/conicalrayculling.pdf>