

TRAK – Projekty

Autor: Łukasz Dąbała

1 Refrakcja w czasie rzeczywistym

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 Mapy półcieni

W ramach projektu należy stworzyć program, który będzie w stanie porównać mapy cieni oraz mapy półcieni (*ang. Penumbra Maps*).

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 wykorzystanej metody oraz wczytanie mapy środowiska
4. implementacja map cieni w podstawowej wersji
5. implementacja algorytmu z artykułu: Penumbra Maps: Approximate Soft Shadows in Real-Time:

http://cwymen.org/papers/egsr03_PenumbraMaps.pdf

3 Badanie struktur przyspieszających

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 promieni
3. implementacja struktur przyspieszających: BVH (*ang. Bounding Volume Hierarchy*) oraz siatki jednorodnej

4 Badanie struktur przyspieszających

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 promieni
3. implementacja struktur przyspieszających: KD-Tree oraz siatki jednorodnej

5 Badanie możliwości oświetlenia otoczenia w przestrzeni ekranu

W ramach projektu należy stworzyć program, który będzie umożliwiał porównanie renderingu z wykorzystaniem techniki SSAO (*Screen Space Ambient Occlusion*), zwykłego Ambient Occlusion oraz bez nich.

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 wykorzystanej metody oraz wczytanie mapy środowiska
4. implementacja techniki SSAO
5. implementacja Ambient Occlusion

6 Interaktywne oświetlenie z wykorzystaniem spójnych map cieni

W ramach projektu należy stworzyć program, który będzie implementował algorytm: Interactive Illumination with Coherent Shadow Maps.

W programie powinny znaleźć się m.in.:

1. stały podgląd z kamery
2. interfejs konsolowy/graficzny, który umożliwi wczytanie sceny (modele wraz z materiałami oraz ich właściwościami) z pliku
3. implementacja algorytmu z artykułu: Interactive Illumination with Coherent Shadow Maps:
<https://jankautz.com/publications/coherentShadowMapsEGSR07.pdf>

7 Dwukierunkowa natychmiastowa metoda energetyczna

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering sceny z wykorzystaniem algorytmu Bidirectional Instant Radiosity.

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: Bidirectional Instant Radiosity:
<http://artis.imag.fr/Projets/Cyber-II/Publications/SIMP06a.pdf>

8 Rendering spektralny

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*).

9 Badanie możliwości próbkowania

W ramach projektu należy stworzyć program, który będzie umożliwiał rendering z wykorzystaniem algorytmu śledzenia ścieżek wraz z różnymi algorytmami próbkowania.

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 próbkowania oraz wczytanie mapy środowiska
3. implementacja algorytmu śledzenia ścieżek
4. różne algorytmy próbkowania: całkowicie losowy, stratified sampling, multi-jittered sampling, szeregi małej rozbieżności (*ang. low-discrepancy series*)