

TRAK – Projekty

Autor: Łukasz Dąbała

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 Mapy półcieni

Prowadzący: dr inż. Łukasz Dąbała

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://cwyman.org/papers/egsr03_PenumbraMaps.pdf

3 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

4 Microfacet materials

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć program, który będzie umożliwiał wyrenderowanie obiektów z wykorzystaniem mikrościankowych BRDF

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 oraz mapy środowiska
3. możliwość zmiany/definicji materiału dla danego obiektu
4. implementacja algorytmu śledzenia ścieżek
5. wsparcie dla modeli: Cook-Torrance, Beckman, GGX.

5 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.

6 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, śledzenia ścieżek ścieżek, mapowania fotonów
4. statystyka na temat renderingu np. ilość wyszukiwanych przecięć, ilość wygenerowanych promieni cienia itd.

7 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>

8 Wielokamerowy rendering w programie Blender

Prowadzący: dr inż. Łukasz Dąbała

W ramach projektu należy stworzyć wtyczkę do programu Blender, która będzie umożliwiać tworzenie zestawu kamer do różnych potrzeb.

Wtyczka powinna umożliwiać:

1. tworzenie kamery stereo w różnych trybach
 - (a) tryb równoległy - kamery mają równoległe osie widoku
 - (b) tryb zbieżny - osie widoku kamer przetną się. Miejsce przecięcia powinno być możliwe do ustawienia
2. tworzenie kamery typu light-field
 - (a) wiele kamer ustawionych w macierz z różnym oddaleniem od siebie (zarówno w osi pionowej jak i poziomej)
 - (b) kamera plenoptyczna - obrazy są od siebie oddalone o 1-2 piksele
3. tworzenie sieci kamer wokół zadanego obiektu i sceny
 - (a) ułożenie równomierne według zadanego pasa wokół obiektu
 - (b) ułożenie równomierne na kuli wokół obiektu
 - (c) ułożenie optymalne wokół obiektu - staramy się minimalizować liczbę kamer tzn. każda kolejna kamera powinna dostarczać jakąś informację na temat obiektu
4. ustawienie renderingu
 - (a) każda kamera ma swój folder na obrazy
 - (b) aplikacja zadanych właściwości kamer do wszystkich kamer utworzonych z wykorzystaniem wtyczki
5. rendering ze wszystkich kamer - rozpoczęcie renderingu powinno renderować według zadanego trybu tzn. kamera po kamerze lub klatka po klatce

9 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>