

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Katedra Informatyki

Autoreferat rozprawy doktorskiej

Projekt i implementacja
równoległych algorytmów transportu światła
opartych o śledzenie promieni kwazi-Monte Carlo

mgr inż. Michał Radziszewski

Promotor:

dr hab. inż. Krzysztof Boryczko, prof. n. AGH

Recenzenci:

prof. dr hab. inż. Jan Zabrodzki, Politechnika Warszawska

dr hab. inż. Krzysztof Cetnarowicz, prof. n. AGH

Cel i zakres pracy

Rendering fotorealistyczny jest częścią grafiki komputerowej, koncentrującą się na tworzeniu zarówno statycznych obrazów, jak i animacji w oparciu o modele trójwymiarowe. Jego celem jest tworzenie ilustracji, które są nierozróżnialne od scen znanych ze świata rzeczywistego. Praca ta jest dedykowana szczególnej klasie algorytmów renderingu fotorealistycznego – globalnemu oświetleniu realizowanemu w oparciu o śledzenie promieni.

Globalne oświetlenie jest przydatną koncepcją w tworzeniu ilustracji realistycznie oświetlonych trójwymiarowych scen. Dzięki automatycznemu i w pełni poprawnemu symulowaniu olbrzymiej różnorodności zjawisk optycznych, tworzy solidną bazę dla oprogramowania do renderingu. Oprogramowanie to wymaga od użytkownika wyłącznie określenia, co przedstawia dana scena, zamiast podawania dokładnego opisu jak zrenderować tę scenę.

Obecne zastosowania tak rozumianego globalnego oświetlenia rozciągają się od wielu systemów CAD do efektów specjalnych w filmach. Kiedy zostaną skonstruowane wystarczająco szybkie komputery, globalne oświetlenie zastąpi inne, mniej dokładne pod względem fizycznym techniki. To stwierdzenie jest często popierane przez obserwację podobnego przełomu dokonującego się obecnie w dziedzinie modelowania, gdzie modele fizyczne skutecznie konkurują z bardziej tradycyjnym podejściem.

Głównym celem tej pracy jest zwiększenie efektywności renderingu poprawnego pod względem fizycznym, a także opracowanie oprogramowania wykorzystującego ściśle modele teoretyczne. Założeniem rozprawy jest uniknięcie dwóch ekstremów – pracy nadmiernie teoretycznej, bez opracowania sposobu i implementacji opracowanych algorytmów, a także implementacji tychże algorytmów metodą prób i błędów. W pracy uwzględniono trendy rozwoju sprzętu komputerowego, ponieważ ma to znaczący wpływ na wydajność opracowywanych algorytmów.

Rozprawa koncentruje się na poprawie niezawodności algorytmów renderingu, a także ich efektywnej, równoległej realizacji. Oba te elementy równocześnie mogą znacząco rozszerzyć zakres zastosowań globalnego oświetlenia, a także są krokiem w kierunku możliwości jego użycia w systemach czasu rzeczywistego. Poniżej przedstawione zostały najważniejsze oryginalne osiągnięcia, użyteczne w realizacji tych założeń:

- modyfikacja pewnych równań tak, aby renderowany obraz był zdefiniowany jako funkcja ciągła, a nie tablica pikseli,
- nowa technika renderingu przy pomocy pełnego widma zjawisk zależnych od długości fali światła,
- jednoprzebiegowy wariant algorytmu mapowania fotonów,
- algorytm łączony, wykorzystujący elementy dwukierunkowego śledzenia ścieżek oraz mapowania fotonów,
- model przetwarzania strumieniowego rozszerzony o pamięć podręczną do odczytu i zapisu,
- interaktywny podgląd częściowych wyników renderingu oparty o procesory GPU,
- interfejs pomiędzy obiektami sceny, kamerami i algorytmami renderingu ukierunkowany na próbkowanie,
- technika optymalizacji przecięć promieni z elementami sceny,
- język tekstuowania stosujący podejście funkcyjne,
- nowy model odbicia połyskliwego.

Transport światła

Obecnie stosowanych jest kilka teorii opisujących zachowanie się światła. Wybór teorii fizycznej dla potrzeb symulacji jest kompromisem pomiędzy dokładnością wyników a kosztem obliczeniowym. Na potrzeby grafiki komputerowej, na ogół wystarczająca jest najprostsza teoria – optyka geometryczna. Jednakże,

pomimo prostoty optyki geometrycznej, niewiele systemów renderingu zapewnia pełną jej realizację ze względu na koszt obliczeniowy. Model teoretyczny optyki geometrycznej, na której niniejsza rozprawa jest oparta, zakłada następujące założenia upraszczające:

- ilość fotonów jest ogromna, a energie przez nie przenoszone są niezmiernie małe – dowolny rozkład fotonów może być traktowany jako wartość ciągła,
- fotony nie oddziałują ze sobą – zjawiska takie jak interferencja nie mogą być symulowane,
- kolizje fotonów z powierzchniami i cząsteczkami nieprzejrzystych mediów (np. mgła albo kurz) są elastyczne – długość fali światła związana z fotonem nie ulega zmianie w trakcie zderzenia,
- dyfrakcja, ciągle zmieniający się współczynnik załamania i jakiegokolwiek inne zjawiska wpływające na ruch fotonów są pomijane – pomiędzy kolizjami fotony poruszają się po liniach prostych,
- prędkość fotonów jest nieskończenie duża, scena znajduje się zawsze w stanie równowagi,
- własności optyczne materiałów nie zależą od natężenia oświetlenia, więc oświetlenie jest liniowe – może być obliczone niezależnie dla każdego źródła światła, a potem zsumowane.

Przy powyższych założeniach transport światła może zostać opisany przy pomocy wielkości radiometrycznych, takich jak strumień czy radiancja. Niektóre niegeometryczne zjawiska, takie jak np. dyfrakcja czy interferencja, nie dają się przedstawić przy pomocy tych wielkości, jednak pewne inne można łatwo uwzględnić poprzez proste rozszerzenia. W niniejszej pracy wszystkie wielkości zostały rozszerzone do wielkości spektralnych – uwzględniając długość fali światła wyłącznie do modelowania kolorów.

Transport światła przy założeniu stosowalności optyki geometrycznej prowadzi do następującego wyrażenia opisującego rozkład radiancji na scenie (zakładając brak zjawisk wolumetrycznych):

$$L(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_s(x, \omega_o, \omega_i) L(T(x, \omega_i), -\omega_i) d\sigma^{\perp}(\omega_i), \quad (1)$$

gdzie L oznacza radiancję, x dowolny punkt na scenie, ω kierunek, f_s funkcję odbicia materiału, T jest operatorem znajduwania najbliższego przecięcia promienia z geometrią sceny, a $\sigma^{\perp}(\omega_i)$ jest miarą rzutowanego kąta bryłowego. Równanie 1 można rozszerzyć o symulację zjawisk wolumetrycznych. Opracowany system renderingu opiera się na rozszerzonej wersji tego równania. Równanie podstawowe jak i rozszerzone można przekształcić do postaci całki oznaczonej, jednak całki te są określone na przestrzeni o nieograniczonej ilości wymiarów.

Kamery mogą zostać formalnie opisane jako urządzenia emitujące *ważność* (ang. importance). Ważność można traktować jako hipotetyczne cząstki podobne do fotonów, rozchodzące się od kamery, przeciwnie do zwrotu promieni światła. Intuicyjnie śledzenie ważności określa jak ważne dla renderowanego obrazu są poszczególne fragmenty sceny. Tworzenie obrazu jest zdefiniowane jako całka z iloczynu ważności (emitowanej z kamery) i radiancji (emitowanej ze źródeł światła). W pracy zmodyfikowano to równanie tak, aby obraz był zdefiniowany jako funkcja ciągła, a nie tablica pikseli.

Algorytmy renderingu

Równanie transportu światła ma rozwiązania analityczne tylko dla bardzo prostych scen, bezużytecznych w praktyce. Wymaga więc ono użycia odpowiednich metod numerycznych. Klasyczne kwadratury, np. Newtona-Cotesa albo Gaussa nie są efektywne w przypadku wielowymiarowych i nieciągłych funkcji. Okazuje się, że użycie algorytmu całkowania Monte Carlo (wykorzystującego liczby losowe i podejście statystyczne oparte o estymatory) daje o wiele lepsze rezultaty.

Prawie wszystkie implementacje całkowania Monte Carlo używają generatorów liczb pseudolosowych. Nie jest to ściśle poprawne, gdyż teoria opiera się na liczbach losowych. Jednak gdy generator spełnia pewne warunki, nie ma to istotnego wpływu na zbieżność algorytmu. Podejście związane z użyciem liczb pseudolosowych może być posunięte dalej. Tzw. metody kwazi-Monte Carlo wykorzystują starannie

zaprojektowane deterministyczne funkcje generujące liczby o wiele bardziej równomiernym rozkładzie niż liczby losowe, co zwykle zmniejsza wariancję estymatorów.

Algorytmy transportu światła zapewniają rozwiązania numeryczne równania transportu światła – są one odpowiedzialne za rendering. Obecnie wyłącznie śledzenie promieni technikami Monte Carlo albo kwazi-Monte Carlo umożliwia rozwiązanie równania transportu światła w jego pełnej formie, bez żadnych uproszczeń. Algorytmy śledzenia promieni są jedną z kilku popularnych metod renderingu. Nie zawsze są one najlepszym wyborem, ale gdy równanie transportu światła ma być rozwiązane dokładnie, a scena zawiera skomplikowaną geometrię i materiały, obecnie nie ma innej możliwości niż użycie śledzenia promieni.

Ścieżki transportu światła są podstawową koncepcją globalnego oświetlenia opartego o śledzenie promieni. Ścieżka taka jest sekwencją promieni łączących punkt na źródle światła z punktem na kamerze. Odbicia promieni są reprezentowane jako wierzchołki. Wszystkie obecnie używane metody nieobciążone opierają się na próbkowaniu lokalnym, które jest techniką polegającą na dodawaniu do konstruowanej ścieżki co najwyżej jednego segmentu na raz.

Stosując lokalne próbkowanie ścieżka transportu światła może zostać zbudowana tylko wtedy, gdy ma co najmniej dwa następujące po sobie optycznie nieidealne odbicia. Przykładem zjawiska opisywanego przez ścieżki, które nie mogą zostać zbudowane to połyski na dnie zbiornika wody od punktowego źródła światła widziany pośrednio przez powierzchnię wody za pomocą kamery o punktowym obiektywie. Takie zjawisko po prostu zniknie z renderowanego obrazu. Metody obciążone na ogół nie posiadają tej wady.

Model RGB jest często używany do renderingu. Jest to jednak nadużycie tego modelu, który projektowany był do przechowywania i wyświetlania obrazów. Model ten nie ma żadnych podstaw fizycznych. Efektem renderingu przy pomocy barw RGB może być np. przekłamanie kolorów, czy niewłaściwie symulowana dyspersja. Oba te problemy mogą zostać zminimalizowane poprzez umiejętne stosowanie różnych sztuczek, jednak wynik renderingu RGB nigdy nie będzie poprawny.

Najpopularniejsze techniki reprezentacji pełnego widma opierają się na kombinacjach liniowych funkcji bazowych albo na próbkowanych punktowo funkcjach ciągłych. Ta druga opcja okazuje się być lepsza przy śledzeniu promieni metodami Monte Carlo. Próbkowanie punktowe o losowych pozycjach próbek powoduje dodatkową wariancję. Jeżeli jednak jest dobrze zaprojektowane, to zbieżność tej techniki jest na ogół bardzo szybka. Poza tym, technika ta zapewnia poprawność wyników, nieosiągalną dla funkcji bazowych o skończonej rozciągłości w przestrzeni długości fal.

Najprostszym sposobem renderingu z użyciem pełnego widma jest symulacja jednej długości fali światła na jedną ścieżkę transportu. Szybko dostrzeżono, że ta technika jest nieefektywna i o wiele bardziej opłaca się śledzić całą grupę promieni o różnych długościach fal na jednej ścieżce. Zjawiska zależne od długości fali światła do tej pory symulowane były za pomocą nieefektywnej, pierwszej z ww. metod. W rozprawie zaproponowano metodę umożliwiającą próbkowanie ścieżek przy użyciu wielu długości fal jednocześnie, nawet przy takich zjawiskach. Proponowana metoda stosuje technikę wielokrotnego próbkowania ważnościowego do próbkowania pełnego widma.

Obecnie nie ma idealnego algorytmu renderingu. Każda ze współcześnie znanych metod jest efektywna w renderingu pewnych scen, ale nie sprawdza się w przypadku innych. W rozprawie przedstawiona jest analiza zachowania różnych algorytmów w zależności od renderowanej sceny – śledzenia ścieżek, dwukierunkowego śledzenia ścieżek oraz mapowania fotonów. Analiza ta stanowi podstawę zaproponowanego łącznego algorytmu transportu światła.

Proponowany algorytm łączy dwie różne techniki aby otrzymać najlepszy możliwy wynik. Użycie techniki nieobciążonej – dwukierunkowego śledzenia ścieżek zapewnia sprawny rendering wielu scen, ze względu na małe wymagania pamięciowe i błąd, który jest łatwy do oszacowania. Algorytm złożony powinien wykryć przypadki trudne dla metody nieobciążonej – wynikające głównie z ograniczenia próbkowania lokalnego, i w tych przypadkach przełączyć rendering na zastosowanie mapowania fotonów. Zatem mapowanie fotonów jest używane tylko wtedy, kiedy jest niezbędne, co minimalizuje zużycie pamięci i występowanie nieprzewidywalnego błędu spowodowanego obciążeniem estymatora.

Mapowanie fotonów jest dwuprzebiegowym algorytmem rozwiązywania równania transportu światła. W pierwszym kroku, śledzone są półścieżki startujące od źródeł światła, a przenoszona energia jest zapisywana w punktach przecięć promieni z geometrią sceny jako tzw. fotony. Następnie budowana jest struktura

specjalizowana do szybkiego wyszukiwania. Jest to tzw. mapa fotonów.

Okazuje się, że mapowanie fotonów można przeprowadzić jednoprzebiegowo, z minimalną utratą wydajności względem oryginalnego podejścia, co zostało zaproponowane w rozprawie. Algorytm jednoprzebiegowy jest o wiele bardziej użyteczny w połączeniu z interaktywnym podglądem wyników renderingu. Technika jednoprzebiegowa wylicza niezbędną minimalną ilość fotonów w mapie jako funkcję liniową ilości policzonych próbek, i dodaje nowe fotony w trakcie renderingu jeżeli okaże się to niezbędne. Zatem mapa fotonów nie jest już strukturą statyczną tylko do odczytu – wymaga synchronizacji w przypadku renderingu równoległego.

Rendering równoległy

Ze względu na trendy w rozwoju procesorów, aby zapewnić optymalną wydajność, obecnie rendering musi być wykonywany równoległe. Na szczęście, większość algorytmów renderingu posiada bardzo wiele części niemalże niezależnych od siebie.

Przetwarzanie strumieniowe jest ograniczoną formą ogólnego obliczenia równoległego. Procesory strumieniowe mają uproszczoną architekturę, co zwiększa ich wydajność, jednak za cenę mniejszej elastyczności. Powoduje to, że nie każdy algorytm równoległy może zostać wykonany na procesorze strumieniowym.

Koncepcja przetwarzania strumieniowego oparta jest o *strumienie* i *jądra*. Strumienie są zbiorami struktur danych, na których operacje są wykonywane przez jądra. Jądro jest dowolną funkcją, przyjmującą jeden strumień wejściowy i posiadającą co najmniej jeden strumień wyjściowy. Jądro przetwarza struktury danych wejściowych niezależnie od siebie, zapisując zero lub więcej struktur wyjściowych do odpowiednich strumieni. Dodatkowo, jądro ma dostęp do specjalnej pamięci tylko do odczytu.

Zasadniczą trudnością ograniczającą różnorodność algorytmów strumieniowych jest całkowity brak możliwości komunikacji pomiędzy przetwarzaniem różnych elementów strumienia danych. W rozprawie zaproponowano dołączenie dodatkowej pamięci do odczytu i zapisu, zwanej dalej pamięcią podręczną. Pamięć ta zapewnia dostępne programowo niepodzielne operacje odczytu i zapisu. Proponowana synchronizacja nie zapewnia jednak zachowania kolejności indywidualnych operacji.

Wyrażenie algorytmu śledzenia promieni jako obliczenia strumieniowego jest możliwe na wiele sposobów. Zaproponowano już podział algorytmu na kilka prostszych jąder, co przy dodatkowym znacznym ograniczeniu możliwości renderingu pozwoliło na wykorzystanie karty graficznej jako procesora strumieniowego. Jeżeli jednak celem nie jest uruchomienie śledzenia promieni na procesorze graficznym, tylko jego efektywne wykonanie na dowolnej maszynie, o wiele lepszym rozwiązaniem jest użycie jednego jądra. Dwukierunkowe mapowanie ścieżek może zostać wykonane w ten sposób na podstawowej maszynie, natomiast jednoprzebiegowe mapowanie fotonów i algorytm łączony wymagają procesora z pamięcią podręczną.

Rozwiązanie z pojedynczym jądrem jest bardzo proste i jednocześnie efektywne. Strumień wejściowy zawiera tylko numery próbek (kolejne liczby całkowite). Jądro na podstawie tych numerów generuje liczby pseudolosowe, których następnie używa do generacji i śledzenia promieni. Opis sceny jest w pamięci tylko do odczytu, natomiast opcjonalnie mapa fotonów w pamięci podręcznej. Wyjściami są strumienie zawierające wartości pełnego widma. Wartości te muszą zostać jeszcze przetworzone tak, aby wygenerować gotowy do wyświetlenia obraz. Przetwarzanie to jest jednak o wiele prostsze niż rendering.

W dziedzinie obliczeń dużej skali, z procesorami ogólnego przeznaczenia od niedawna konkurują procesory graficzne. Procesory te mają o wiele większą moc obliczeniową, jednak dostępną wyłącznie dla wąskiej klasy algorytmów – operujących na regularnych strukturach danych i wykonujących dużo obliczeń zmienoprzecinkowych przy małej ilości rozgałęzień warunkowych. Niestety, najbardziej czasochłonną operacją w śledzeniu promieni jest przeszukiwanie drzew. Algorytmy te mają charakterystykę zupełnie odmienną od tej oczekiwanej przez procesory graficzne. Zdaniem autora, zupełnie nie sprawdzają się one przy obliczaniu globalnego oświetlenia, do którego o wiele lepsze są wieloprocessorowe stacje robocze. Procesory graficzne natomiast bardzo dobrze radzą sobie z obróbką strumieni wyjściowych renderingu.

Przetwarzanie wynikowych strumieni renderingu do obrazu, który jest wyświetlany na ekranie, o wiele lepiej przekazać na specjalizowany procesor graficzny, niż wykonywać je na procesorze głównym, będą-

cym już obciążonym renderingiem. Podstawową ideą proponowanego podziału zadań pomiędzy CPU a GPU jest zastosowanie odpowiedniego procesora do odpowiedniego algorytmu, zamiast niepotrzebnego przenoszenia wszystkiego na procesor graficzny.

Proponowany klient wizualizacji większość obliczeń wykonuje przy pomocy programów GLSL (OpenGL Shading Language) w wersji co najmniej 3.3. Podstawowym zadaniem klienta jest przekształcenie próbek w obraz rastrowy, z dynamicznym dostosowaniem rozdzielczości do ilości otrzymanych próbek i szacowanej ich wariancji. Ilość ta zależy od wydajności renderingu. Dopóki nie nastąpi zmiana opisu sceny, proces renderingu generuje kolejne próbki, przy pomocy których klient dynamicznie poprawia jakość obrazu. Dodatkowo klient może zmieniać pewne parametry obrazu, jak np. jasność czy kontrast, nie przerywając procesu renderingu.

Wyniki

Aby móc precyzyjnie oceniać jakość obrazów wygenerowanych przez różne algorytmy, musi zostać zastosowana norma umożliwiająca porównywanie tych obrazów z obrazem referencyjnym. W przypadku symulacji optyki geometrycznej, najlepszym możliwym obrazem referencyjnym byłaby fotografia rzeczywistej sceny. Niestety, aby takie porównanie miało sens, dane wejściowe do procesu renderingu – np. geometria sceny, odbijalność powierzchni czy radiancja źródeł światła – muszą zostać precyzyjnie zmierzone. Bez laboratorium wyposażonego w specjalistyczną aparaturę pomiarową nie jest to możliwe.

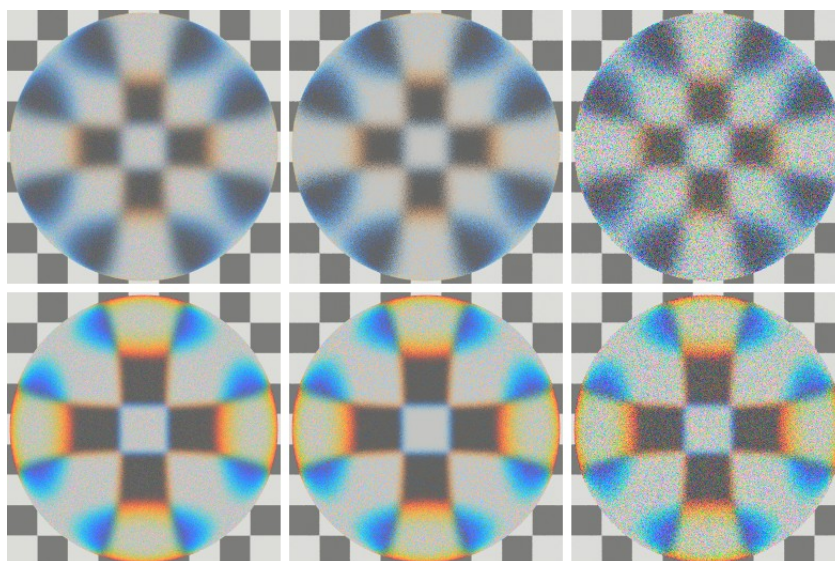
W większości prac poświęconych renderingowi, a także w niniejszej rozprawie, stosowane jest o wiele prostsze podejście. Obraz referencyjny jest tworzony przy pomocy dobrze znanego algorytmu (np. dwukierunkowego śledzenia promieni), którego poprawność została udowodniona, w bardzo długim czasie, tak aby wyeliminować wpływ błędu metody. Ponieważ dwukierunkowe śledzenie promieni jest algorytmem nieobciążonym, jeżeli wyprodukowany obraz nie ma widocznego szumu (skutek wariancji), to z dużym prawdopodobieństwem jest on całkowicie poprawny. Następnie testowane algorytmy uruchamiane są na tym samym wejściu co algorytm referencyjny. Powyższa metoda jest w stanie wykryć wszystkie błędy w renderingu, poza tymi wynikającymi z założenia o stosowalności optyki geometrycznej.

Bardzo ważna jest metoda użyta do porównywania obrazów testowych z obrazem referencyjnym. Opracowano już pewne metody do tego celu, biorące pod uwagę różne właściwości ludzkiego wzroku. Jednakże żadna z tych metod do tej pory nie uzyskała powszechnej akceptacji. Z tego powodu wszystkie porównania obrazów w niniejszej pracy stosują normę opartą o średni błąd kwadratowy.

Porównanie wyników renderingu w pełnym widmie przy użyciu poprzedniej techniki i proponowanej na scenie testowej z jednokrotnym odbiciem jest przedstawione na rysunku 1. W górnym wierszu: połyskliwość $n = 1000$ i liniowo malejący współczynnik załamania η od 1.35 dla $380nm$ do 1.20 dla $830nm$, a w dolnym: $n = 4000$, $\eta \in [1.50, 1.20]$. Zysk z zastosowania proponowanej techniki jest mniejszy, jeżeli zwiększona zostanie połyskliwość lub dyspersyjność. Błąd numeryczny dla tej sceny jest przedstawiony w tabeli 1.

Ustawienia	C	A1	A2
$n = 1000$ $\eta = [1.35, 1.20]$	1	$1.26 \cdot 10^{-1}$	$2.47 \cdot 10^{-1}$
	4	$6.67 \cdot 10^{-2}$	$2.02 \cdot 10^{-1}$
	16	$2.63 \cdot 10^{-2}$	$1.34 \cdot 10^{-1}$
	64	$1.22 \cdot 10^{-2}$	$7.56 \cdot 10^{-2}$
	256	$5.32 \cdot 10^{-3}$	$3.83 \cdot 10^{-2}$
$n = 4000$ $\eta = [1.50, 1.20]$	1	$2.07 \cdot 10^{-1}$	$2.46 \cdot 10^{-1}$
	4	$1.33 \cdot 10^{-1}$	$1.96 \cdot 10^{-1}$
	16	$7.39 \cdot 10^{-2}$	$1.29 \cdot 10^{-1}$
	64	$3.84 \cdot 10^{-2}$	$7.37 \cdot 10^{-2}$
	256	$1.74 \cdot 10^{-2}$	$3.72 \cdot 10^{-2}$

Tabela 1: Porównanie błędu numerycznego proponowanej techniki (kolumna A1) i starej (A2), dla C próbek.



Rysunek 1: Złamanie z dyspersją w ośrodku niedoskonałym optycznie. Lewy górny obraz: poprzednia technika, 900 próbek na piksel. Środkowy górny obraz: proponowana technika, 50 próbek na piksel, podobna jakość. Prawy górny obraz: poprzednia technika, też 50 próbek na piksel. Obrazki w dolnym wierszu zostały wygenerowane używając odpowiednio 900, 100, i 100 próbek na piksel.



Rysunek 2: Porównanie technik renderingu pełnego widma. Lewo: proponowana technika, 1 próbka/piksel. Środek: poprzednia technika, 1 próbka/piksel. Prawo: obraz referencyjny, 256 próbek/piksel.

Na rysunku 2 przedstawione jest porównanie wyników na rzeczywistej scenie (rysunek 3) renderowanej przy pomocy dwukierunkowego śledzenia ścieżek, skonfigurowanego do współpracy z poprzednią oraz proponowaną techniką renderingu pełnego widma. Błąd numeryczny obliczony na prostokącie zawierającym figurkę jest pokazany w tabeli 2.

algorytm	Ilość próbek				
	1M	2M	4M	8M	16M
A1	0.203	0.156	0.121	0.103	0.085
A2	0.376	0.285	0.229	0.175	0.132

Tabela 2: Porównanie błędu numerycznego proponowanej techniki (wiersz A1) i starej (A2) dla fragmentu obrazu z rysunku 2.



Rysunek 3: Rendering w pełnym widmie sceny z niedoskonałym załamaniem światła na szkłe. Obraz renderowany dwukierunkowym śledzeniem ścieżek, z użyciem 0.5G próbek, w rozdzielczości 1920x1080, w czasie 3h 27min na 2.93GHz procesorze Intel Core i7. W prawym dolnym rogu jest miniatura pokazująca względną wariancję fragmentów obrazu.

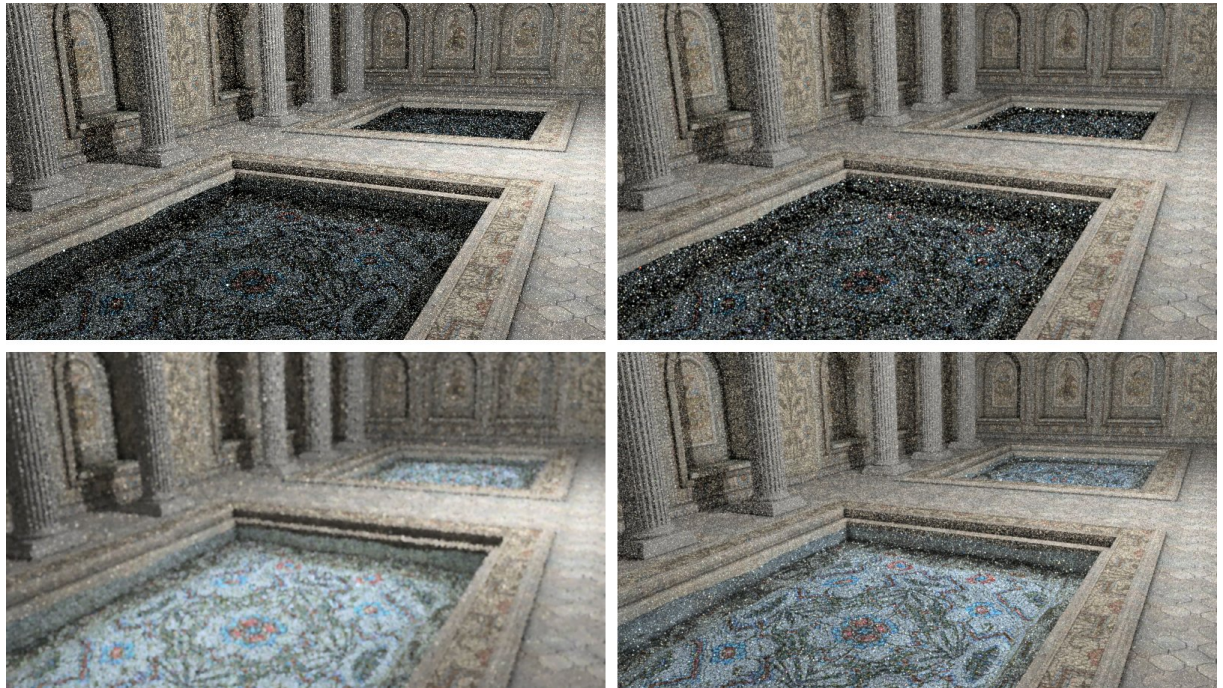
Porównanie obrazów generowanych przez proponowany algorytm łączony z obrazami generowanymi przez śledzenie ścieżek, dwukierunkowe śledzenie ścieżek oraz mapowanie fotonów przedstawione jest na rysunku 4. Ograniczenie próbkowania lokalnego uniemożliwia zastosowanie obu wariantów śledzenia ścieżek na tej scenie. Proponowany algorytm, ze względu na ograniczoną rolę mapowania fotonów, zapewnia mniejsze rozmycie w tym samym czasie renderingu niż mapowanie fotonów użyte jako samodzielna technika. Obraz referencyjny znajduje się na rysunku 5. Numeryczne porównanie wyników przedstawione jest w tabeli 3.

Algorytm	Czas renderingu			
	15sec	1min	4min	16min
PT	13.9	13.2	12.4	11.1
BDPT	9.97	9.64	8.95	7.94
PM	9.93	9.94	8.86	8.28
Combined	9.51	8.75	8.14	7.19

Tabela 3: Porównanie szybkości zbieżności różnych algorytmów dla sceny z rysunku 5. Tabela zawiera błąd numeryczny skalowany o czynnik $f = 10^2$.

Rendering równoległy nie ma wpływu na jakość obrazu. Istotne jest jednak określenie osiąganego przyspieszenia względem renderingu sekwencyjnego w zależności od stopnia równoległości, stosowanego algorytmu i złożoności sceny. Do testów zastosowany został czterordzeniowy procesor Intel Core i7, mogący przetwarzać równoległe do ośmiu wątków wykorzystując technologię SMT (każdy rdzeń obsługuje do dwóch wątków jednocześnie). Dwie sceny testowe zostały przedstawione na rysunku 6. Scena a) jest złożona z około miliona prymitywów, natomiast scena b) zaledwie siedmiu. Scena b) mieści się w całości w jednym węźle drzewa kd, a także w pamięci cache L3 procesora. Obie sceny były renderowane za pomocą wybranych algorytmów transportu światła, ograniczając maszynę strumieniową do stosowania z góry zadanej liczby wątków. Wyniki przedstawiono na rysunku 7.

Podejście strumieniowe do śledzenia promieni jest dogodną techniką wykorzystania ich potencjału do pracy równoległej. Dwie najważniejsze korzyści to skalowalność tego podejścia wobec zwiększania ilości



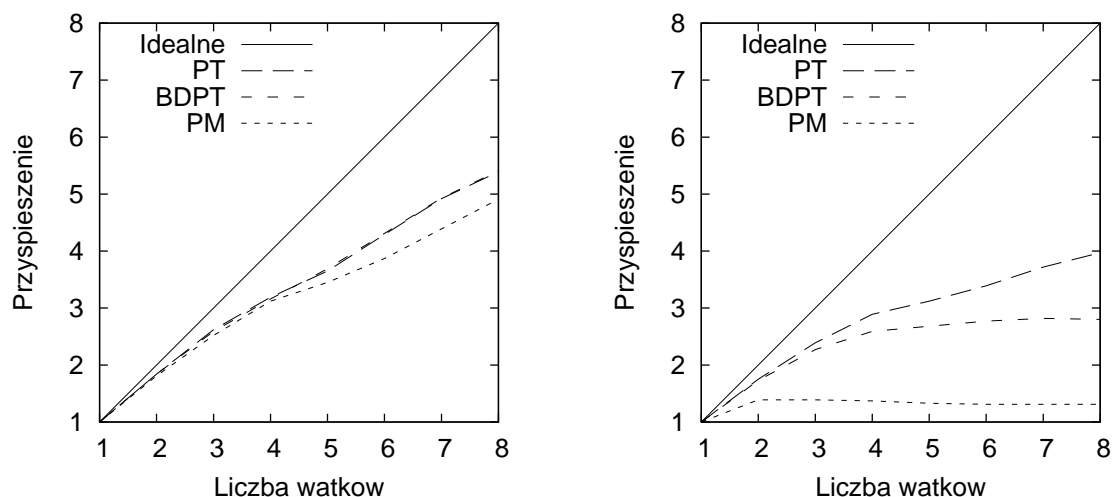
Rysunek 4: Porównanie różnych algorytmów renderingu. Lewy górny obraz: śledzenie ścieżek, 2M próbek. Prawy górny obraz: dwukierunkowe śledzenie ścieżek, 1M próbek. Lewy dolny obraz: mapowanie fotonów, 256k próbek, 128k fotonów. Prawy dolny obraz: algorytm łączony, 1M próbek, 32k fotonów. Wszystkie obrazy zostały wygenerowane w przybliżeniu w tym samym czasie.



Rysunek 5: Scena ilustrująca transport światła przez powierzchnię eody, renderowana proponowanym algorytmem łączonym, przy użyciu 64M próbek i 8M fotonów, w rozdzielczości 1920x1080, w czasie 1h 50min na 2.93GHz procesorze Intel Core i7. W prawym dolnym rogu jest miniatura pokazująca względną wariację fragmentów obrazu.



Rysunek 6: Sceny testowe dla renderingu równoległego. Lewo: scena a) zbudowana z ok. 1M prymitywów. Prawo: scena b) z siedmiu prymitywów.



Rysunek 7: Przyspieszenie śledzenia ścieżek (PT), dwukierunkowego śledzenia ścieżek (BDPT) i mapowania fotonów (PM). Lewo: wyniki dla sceny a). Prawo: wyniki dla sceny b).

rdzeni obliczeniowych i ukrywanie opóźnień w dostępie do pamięci.

Ukrywanie opóźnień jest bardzo dobrze widoczne w przyspieszeniu renderingu sceny a). Maksymalne przyspieszenie śledzenia ścieżek osiągnięte na 8 wątkach przy 4 rdzeniach wyniosło ok. 5.5. Jest to spowodowane tym, że rdzeń może przetwarzać ten wątek, który ma dostępne dane. Oczekiwanie na dostęp do pamięci jednego wątku nie blokuje rdzenia. Przyspieszenie śledzenia ścieżek ok. 3.2 dla 4 wątków wydaje się być spowodowane niedoskonałością sprzętu. Algorytmy te nie używają struktur wymagających synchronizacji, więc ograniczenie przyspieszenia jest spowodowane synchronizacją wewnątrz procesora – prawdopodobnie współdzielonych pamięci cache L3 i kontrolera pamięci. Przy renderingu sceny b) dwukierunkowe śledzenie ścieżek osiąga gorsze przyspieszenie niż zwykle śledzenie ścieżek. Dla tak prostej sceny wąskim gardłem okazuje się być transfer strumienia wyjściowego do procesora graficznego. Zwykle śledzenie ścieżek generuje kilkukrotnie mniej danych wyjściowych w tej samej jednostce czasu – procesor testowy jest zbyt wolny aby transfer danych znacząco ograniczał ten algorytm.

Całkiem inne wyniki otrzymano dla mapowania fotonów. W testach używany był jednorobiegowy wariant tej metody, wymagający synchronizacji. Rendering równoległy jest opłacalny dla złożonych scen, natomiast nie sprawdza się w przypadku scen bardzo prostych. W zależności od złożoności sceny, istnieje

je graniczna liczba wątków, przy której synchronizacja mapy fotonów ogranicza wzrost przyspieszenia. Dalsze zwiększanie liczby wątków umożliwia rendering bardziej skomplikowanych scen w tym samym czasie, zamiast przyspieszenia renderingu tej samej sceny. Przyspieszenie algorytmu łączonego znajduje się pomiędzy przyspieszeniem dwukierunkowego śledzenia ścieżek, a przyspieszeniem mapowania fotonów. Dokładny wynik silnie zależy od tego, co dana scena przedstawia, gdyż algorytm ten wybiera używaną technikę w zależności od rodzaju odbić na każdej ścieżce transportu światła.

Podsumowanie

Jednym z podstawowych założeń projektowych prezentowanych algorytmów renderingu była ich poprawność przy założeniu obowiązywania zasad optyki geometrycznej. Zbieżność renderowanego obrazu do wyniku idealnego wraz z wydłużaniem czasu renderingu jest więc ściśle wymagana. Badania opisane w rozprawie stanowią krok naprzód w poprawie niezawodności algorytmów spełniających to wymaganie. Co więcej, dołączona została interaktywna przeglądarka cząstkowych rezultatów renderingu. Jednakże, ze względu na ściśle wymagania poprawności, przedstawione tutaj algorytmy nie są w stanie zapewnić wydajności oferowanej przez inne algorytmy zaprojektowane tak, aby uzyskać możliwie dobre rezultaty w jak najkrótszym czasie, nie zwracając uwagi na zbieżność do rozwiązania całkowicie poprawnego.

Po kilku latach spędzonych na badaniach w tej dziedzinie, dla autora stało się oczywiste, że mimo przeprowadzenia dużej ilości badań algorytmów renderingu, rendering czasu rzeczywistego, który wciąż bazuje na rasteryzacji linii skanu, obecnie nie może oprzeć się na globalnym oświetleniu i śledzeniu promieni, ze względu na ograniczenia w wydajności współczesnych komputerów. Jednakże rozwój sprzętu, a zwłaszcza procesorów graficznych, przebiega z zadziwiającą prędkością, całkowicie zmieniając sposoby programowania w czasie zaledwie kilku lat. Dwie takie znaczące zmiany zaszły w trakcie opracowywania algorytmów do prezentowanych badań. Było to wprowadzenie częściowo programowalnych procesorów graficznych oraz rozpowszechnienie wielordzeniowych procesorów głównych. Dziś nie jest jeszcze oczywiste, który model – procesor centralny, czy graficzny będzie w niedalekiej przyszłości lepszy dla algorytmów śledzenia promieni. Wydaje się jednak, że wkrótce przynajmniej jeden z nich będzie wystarczająco wydajny, aby zapewnić obliczanie pełnego globalnego oświetlenia w czasie rzeczywistym. Wyniki badań opisane w niniejszej rozprawie mogą ten dzień przybliżyć, czyniąc algorytmy globalnego oświetlenia bardziej wydajnymi i niezawodnymi.