

Warszawa, dnia

Student: Tomasz Jakubczyk

Numer albumu: 253093

Adres poczty elektronicznej: sigrond93@gmail.com

Numer telefonu: 666218894

RAPORT Z PRAKTYKI OBOWIĄZKOWEJ

(raport z praktyki obowiązkowej powinien zawierać:

- 1. krótką informację o placówce, w której praktyka była realizowana (nazwa, adres, działalność, produkcja, współpraca itp.)*
- 2. opis merytoryczny wykonanych prac z podaniem zadań jakie student miał do wykonania, charakterystykę aparatury i urządzeń, z których korzystał, opisem metod stosowanych do rozwiązywania zadań, prezentacją uzyskanych wyników,*
- 3. informację jaką wiedzę i umiejętności zdobyte na studiach student wykorzystał w trakcie praktyki, informację o wiedzy i umiejętnościach zdobytych przez studenta w trakcie praktyki,*
- 4. załączniki : publikacje dotyczące tematu praktyki lub odnośniki do nich, teksty opracowań wykonanych w czasie praktyki, inne podobne materiały mające bezpośredni związek z tematem praktyki,*
- 5. uwagi własne studenta, ciekawe spostrzeżenia, ewentualne trudności, informację o wymaganiach i oczekiwaniach pracodawcy w stosunku do nowych pracowników, wnioski dla młodszych kolegów itp.*

Raport ze praktyki obowiązkowej studenta Tomasza Jakubczyka Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej, zrealizowanej od dnia 29.02.2016 do dnia, w Laboratorium Klasterów i Nanocząstek
Zespół Spektroskopii Laserowej
Oddział Fizyki Promieniowania i Spektroskopii
Instytut Fizyki PAN

(nazwa i adres przedsiębiorstwa lub instytucji)

Główną techniką pomiarową używaną w Laboratorium Klasterów i Nanocząstek jest statyczne rozpraszanie światła. Wyniki eksperymentu są rejestrowane w postaci filmu. Kolejne klatki odpowiadają chwilowym obrazom rozproszonemu. Do opisu rozpraszania światła na obiektach kulistych wykorzystywana jest teoria Mie'go. Przy jej pomocy dla każdego obrazu rozwiązywane jest zagadnienie odwrotne i znajdowany promień (i współczynnik załamania) obserwowanego obiektu. Z punktu widzenia informatycznego przetwarzanie wyników eksperymentu odbywa się w dwóch etapach: (i) wydobywanie azymutalnego rozkładu kąтового natężenia światła, (ii) rozwiązanie właściwego zagadnienia odwrotnego. Rozwiązanie zagadnienia odwrotnego zostało zaimplementowane w CUDA C dramatycznie przyspieszając przetwarzanie danych z wielką korzyścią dla prowadzonych prac naukowych.

W ramach proponowanej praktyki studenckiej należało przyspieszyć istniejącą procedurę (Matlab, C) wydobywania azymutalnego rozkładu kąтового natężenia światła (przetwarzania klatek filmu zarejestrowanego w doświadczeniu) wykorzystując możliwość obliczeń na GPU. Będzie się to prawdopodobnie wiązało z przepisaniem jej do CUDA C++.

Początek praktyk pokrył się z uruchomieniem nowego komputera złożonego specjalnie z myślą o przetwarzaniu danych eksperymentu fizycznego. Zainstalowałem brakujące niezbędne sterowniki (Nvidia CUDA) i oprogramowanie (Visual Studio). Dokonałem konfiguracji Matlab'a niezbędnej do kompilowania kodu hybrydowego Visual C/C++ i CUDA C/C++ do plików .mex za pomocą skryptu nvmex. Jako że stanowisko działało pod 64 bitowym systemem w przeciwieństwie do poprzednich stanowisk musiałem dostosować i przekompilować część skryptów i bibliotek. Przeprowadziłem profiling dotychczasowych procedur wydobywania azymutalnego rozkładu kąтового natężenia światła i zidentyfikowałem wąskie gardła. Czytanie pliku odbywało się po jednym bajcie, co miało znaczący wpływ na wydajność (według profilera ~50%). Decodec, demosaic i

wygładzanie zakłóceń zajmowało drugą połowę czasu. Zacząłem od napisania współbieżnej wersji procedury ReducedMean w CUDA C++. Następnie po konsultacjach z fizykami zamieniłem ją na funkcję CUDA C++ liczącą średnią kroczącą na GPU dla danych z każdej klatki filmu. Kiedy było już widać, że główna część obliczeń została skutecznie przyspieszona rozpocząłem prace nad napisaniem funkcji .mex, która wykonałaby wszystkie obliczenia IntensityCalculation w pojedynczym wywołaniu dla całego filmu zamiast wywołań dla każdej klatki z osobna, gdyż okazuje się, że takie pojedyncze w dużej ilości dają spore dodatkowe obciążenie czasem. Uznałem, że plik powinno się czytać w osobnym wątku jednocześnie z obliczeniami, żeby nie marnować czasu kiedy można pobierać dane z dysku, oraz z zaleceniami MSDN, żeby uzyskać najlepszą wydajność odczytu z dysku należy czytać bloki po 64kB. Do przekazywania danych z wątku czytającego plik z dysku do wątku obliczeniowego postanowiłem użyć bufora cyklicznego zapakowanego w monitor. Sprawilo mi to trochę trudności przy debugowaniu ze względu na konieczność zapewnienia możliwie nieprzerwanego czytania pliku z dysku. Następnie do wątku obliczeniowego dodałem funkcje (kernele) CUDA obliczające równoległe rzeczywiste wartości pixeli (decodec), równoległy demosaic, równoległe dzielenie całej klatki przez obraz korekcyjny i nałożenie zadanej maski, dostosowałem wcześniej napisane wygładzanie średnią kroczącą i dopisałem wybieranie równomiernie rozłożonych punktów reprezentatywnych do dalszego wykorzystania w rozważaniu wyników doświadczenia. Trochę trudności sprawiło poprawne przekazywanie argumentów między funkcjami C++ i Matlabem ze względu na pionową orientację pierwszego wymiaru macierzy. Dość czasochłonnym i delikatnym okazało się zadanie znajdowania klatek w danych z dysku podzielonych po 10*64kB ze względu na wydajność odczytu, gdyż klatki mają rozmiary $640*480*2=614400$ co generuje wiele przypadków do rozważenia. Dodatkowo film z danymi z kamery Pike zapisany był przy użyciu nie opisanego CODEC'a. Rozmiar filmu przekracza zwykle 4,2GB (wcześniej wykorzystywane procedury wymagały, żeby film podzielić na wiele mniejszych plików), offset pomiędzy klatkami nie ma ustalonej długości (nie jest to wyrównanie naturalne), ale klatka jest zawsze dosunięta w prawo do nagłówka następnej klatki. Mimo, że film jest zapisany z rozszerzeniem .avi, to nie jest w całości zgodny ze specyfikacją RIFF. Wyniki końcowe w porównaniu do wyników ze starych procedur są zgodne. Czas obliczeń został skrócony kilkunastu godzin (zależnie od wybranych opcji) do 50 sekund dla filmu o rozmiarze ~10GB. Można uznać to za pełen sukces.

Szczególnie przydatną wiedzę zdobytą na studiach okazała się synchronizacja procesów. Inne wykozystane umiejętności które rozwijałem w trakcie studiów to programowanie w C, C++, CUDA C/C++, Matlab, oraz rozumienie assemblera.

.....
(data i podpis studenta)

OPINIA

.....
(data i podpis Zakładowego Opiekuna Praktyki)