

STRESZCZENIE ROZPRAWY DOKTORSKIEJ

Metodyka optymalizacji czasowej algorytmu fastICA w elektroencefalografii gęstej matrycy z wykorzystaniem możliwości wielordzeniowych architektur

Autor: Anna Gajos-Balińska

Promotor: dr hab. Grzegorz Marcin Wójcik

Słowa kluczowe: elektroencefalografia, analiza niezależnych składowych,
programowanie równoległe, CUDA

Elektroencefalografia jest techniką badania ludzkiego mózgu. To metoda diagnostyczna oraz badawcza wykorzystywana między innymi w psychologii czy medycynie. Jednak sygnał EEG jest podatny na zakłócenia, które nie pochodzą z mózgu, dlatego istotną częścią obróbki takiego sygnału jest jego oczyszczenie. Jedną z najpowszechniejszych metod wykorzystywanych w tym celu jest Analiza Niezależnych Składowych (ICA), a jedną z najczęściej spotykanych implementacji jest algorytm fastICA. Jednak z uwagi na charakter algorytmu oraz wykorzystanie elektroencefalografów gęstej matrycy (powyżej 128 elektrod), generujących znacznie więcej danych może być on kosztowny czasowo. Ponadto dostępne dla badaczy implementacje algorytmu typu ICA rzadko nastawione są na zadowalającą optymalizację czasową.

Niniejsza praca jest próbą odpowiedzenia na pytanie, czy przy użyciu obliczeń równoległych i odpowiedniej implementacji, wykorzystującej efektywną wektoryzację kodu, możliwa jest lepsza skalowalność czasu obliczeń algorytmu fastICA. Na początku przedstawione zostały podstawy elektroencefalografii oraz metody pracy nad sygnałem, wraz z opisem sprzętu, na którym oparto badania. W dalszej części znalazł się opis algorytmu analizy niezależnych składowych, jego główne założenia i podstawy teoretyczne w szczególności w kontekście wykorzystania danych EEG. W pracy przybliżone zostały motywacje podjęcia tematu i ścieżka dotarcia do finalnej wersji programu realizującego algorytm. Implementacja algorytmu fastICA została dostosowana dla potrzeb architektury wielordzeniowej oraz danych EEG, pochodzących z elektroencefalografu o dużej liczbie elektrod. Wykorzystane zostały możliwości wielordzeniowych architektur Intel, dostępnych bibliotek i rozszerzeń takich jak MKL, BLAS czy Cilk Plus oraz możliwości bibliotek CUDA (CUBLAS).

Rozwiązane zostało przetestowane na architekturach o różnych parametrach sprzętowych. Zmierzony został czas wykonania algorytmu dla danych różnego rozmiaru. Porównany został czas wykonania implementacji algorytmu fastICA z ogólnodostępnej biblioteki z własną implementacją z użyciem obliczeń równoległych oraz z tą samą implementacją z dodatkowym wsparciem technologii CUDA.

Anna Gajos-Balińska