

Aspekty czasowe algorytmu SURF w wersji sekwencyjnej i równoległej zaimplementowanej w technologii CUDA

Magdalena Szymczyk, Piotr Szymczyk

AGH Akademia Górniczo-Hutnicza, Wydział EAIiE, Katedra Automatyki

Streszczenie: W artykule przedstawiono wyniki prac, których celem było zbadanie możliwości implementacji algorytmu wyznaczania punktów charakterystycznych za pomocą metody SURF na platformie CUDA oraz porównanie czasów obliczeń sekwencyjnej i równoległej implementacji tego algorytmu.

Słowa kluczowe: SURF, CUDA, GPGPU, rozpoznawanie obrazów, programowanie równoległe

1. Wprowadzenie

Rozpoznawanie obrazów jest czynnością złożoną i pracochłonną – wymaga dużych mocy obliczeniowych, w szczególności w przypadku gdy operacje te chcemy wykonywać na bieżąco na strumieniu danych pochodzących z kamery. W celu przyspieszenia obliczeń można wykorzystać coraz bardziej popularną i tanią technologię przetwarzania równoległego przy pomocy kart graficznych i oprogramowania CUDA (GPGPU).

Istnieje szereg znanych z literatury metod rozpoznawania obrazów, jedną z najbardziej obiecujących ze względu na prędkość jest metoda SURF [6].

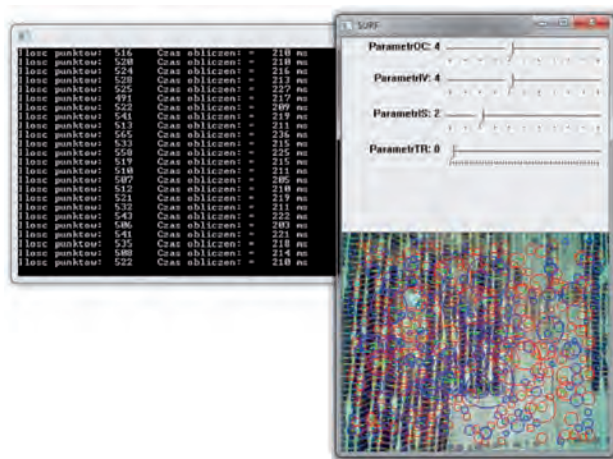
2. Technologia CUDA

CUDA jest technologią opracowaną przez firmę NVidia. Opiera się ona na wykorzystaniu procesorów GPU do prowadzenia obliczeń równoległych w modelu SIMT [5]. Jest to technologia stosunkowo tania i powszechnie dostępna, gdyż można wykorzystywać karty graficzne NVidia wraz z dostępnym pakietem oprogramowania CUDA wzbogaconym o liczne biblioteki. W przeprowadzonych badaniach wykorzystano dwa rodzaje kart: GeForce 9600M GT i GeForce GTX 280.

Karta GeForce 9600M GT ma 32 rdzenie pracujące z częstotliwością 1,25 GHz oraz 977 MB pamięci RAM. Natomiast karta GeForce GTX 280 ma 240 rdzeni pracujących z częstotliwością 1,3 GHz i 1008 MB pamięci RAM.

3. Algorytm SURF

SURF (*Speeded Up Robust Features*) jest algorytmem detekcji i opisu obrazu poprzez punkty charakterystyczne [2–4]. Po raz pierwszy został zaprezentowany przez Herberta Baya

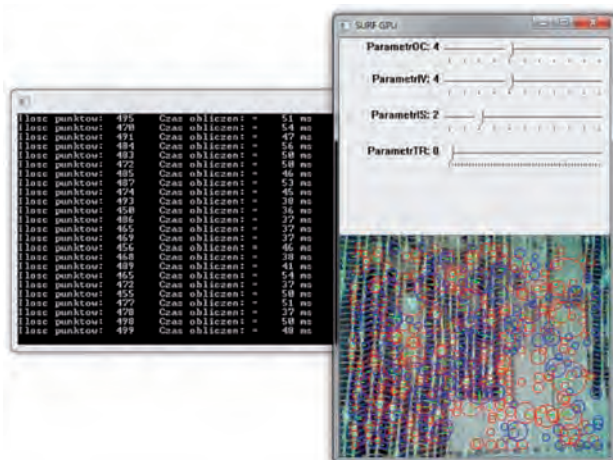


Rys. 1. Wyznaczanie punktów SURF na CPU

Fig. 1. Determination of SURF key points on the CPU

w 2006 r. [1]. Jest używany do rozpoznawania obiektów i rekonstrukcji scen 3D. Jest częściowo wzorowany na SIFT [2]. Standardowa wersja SURF jest kilka razy szybsza od SIFT [6]. SURF jest inwariantny ze względu na skalę i obrót. Algorytm działa w dwóch fazach. W pierwszej wyznaczane są punkty charakterystyczne, a w drugiej dla każdego punktu obliczany jest 64-elementowy wektor będący jego deskryptorem.

Szybkość działania algorytmu uzyskano między innymi poprzez zastosowanie skalowanego obrazu oraz aproksymacji filtrami blokowymi wyznaczania wyznacznika Hessianu.



Rys. 2. Wyznaczanie punktów SURF na GPU

Fig. 2. Determination of SURF key points on the GPU

Jak widać na rys. 1 oraz 2 obydwie (sekwencyjna na CPU i równoległa na GPU) implementacje działały prawidłowo oraz czego należało się spodziewać implementacja równoległa działała znacznie szybciej. Dla około 500 punktów charakterystycznych SURF czas obliczeń na CPU wynosił ponad 200 ms, a na GPU około 50 ms.

4. Porównanie czasu obliczeń punktów charakterystycznych dla różnych implementacji algorytmu SURF uruchamianych na różnym sprzęcie

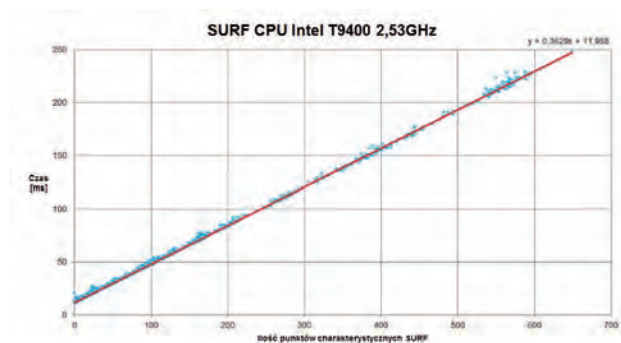
Do przeprowadzenia testów równoległych wykorzystano karty GeForce 9600M GT i GeForce GTX 280. Natomiast do testów sekwencyjnych procesory: Intel T9400 pracujący z częstotliwością 2,53 GHz oraz Intel E8400 pracujący z częstotliwością 3 GHz. Poniżej w formie wykresów przedstawiono otrzymane wyniki.

Z rys. 3 i 4 wynika oczywiście, że im szybszy procesor pracujący sekwencyjnie, tym obliczenia realizowane w krótszym czasie.

Z rys. 7 można wywnioskować, że wyznaczanie dużej liczby punktów charakterystycznych na GPU jest znacznie

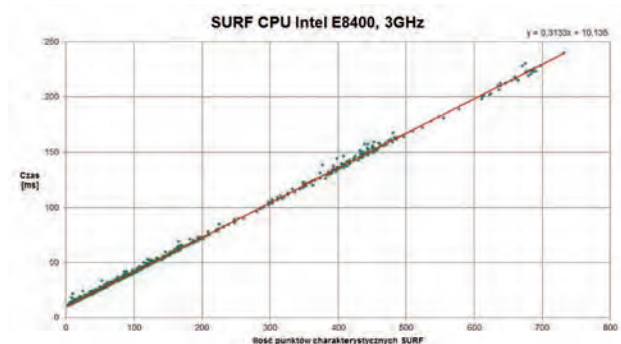
szybsze. Obserwuje się również mały przyrost czasu wykonania aplikacji na GPU przy znacznym przyroście punktów charakterystycznych. Przy małych liczbach punktów charakterystycznych SURF metody sekwencyjne są szybsze niż równoległe. Charakterystyki dla GPU są w porównaniu z CPU płaskie.

Z uzyskanych wyników dla problemu wyznaczania punktów charakterystycznych SURF wynika, że metoda ta ma bardzo duży potencjał dla szybkiego rozpoznawania obiektów na obrazie otrzymany z kamery.



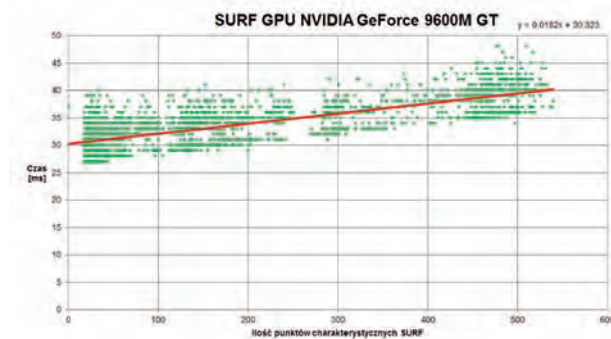
Rys. 3. Zależność czasu obliczeń od liczby punktów charakterystycznych dla implementacji sekwencyjnej pracującej na procesorze Intel T9400

Fig. 3. Dependence between calculation time and number of key points of the sequential implementation working on Intel T9400



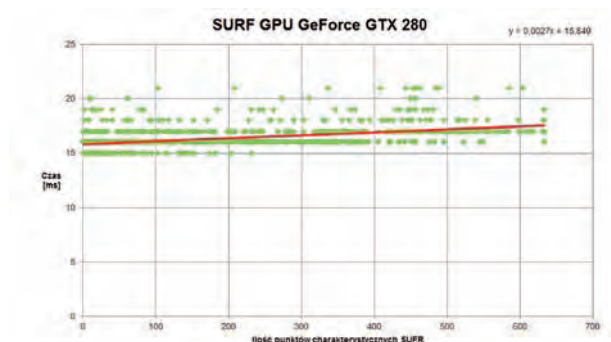
Rys. 4. Zależność czasu obliczeń od liczby punktów charakterystycznych dla implementacji sekwencyjnej pracującej na procesorze Intel E8400

Fig. 4. Dependence between the calculation time and number of key points of the sequential implementation working on Intel E8400



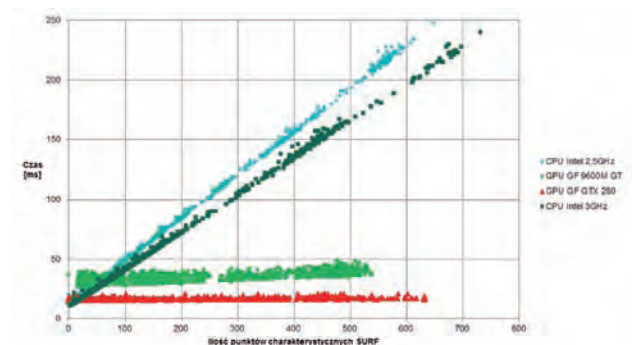
Rys. 5. Zależność czasu obliczeń od liczby punktów charakterystycznych dla implementacji równoległej pracującej na karcie GeForce 9600M GT

Fig. 5. Dependence between calculation time and number of key points for the parallel implementation working on GeForce 9600M GT



Rys. 6. Zależność czasu obliczeń od liczby punktów charakterystycznych dla implementacji równoległej pracującej na karcie GeForce GTX 280

Fig. 6. Dependence between calculation time and number of key points for the parallel implementation working on GeForce GTX 280



Rys. 7. Zbiorcze porównanie czasów obliczeń
Fig. 7. Summary comparison of computation times

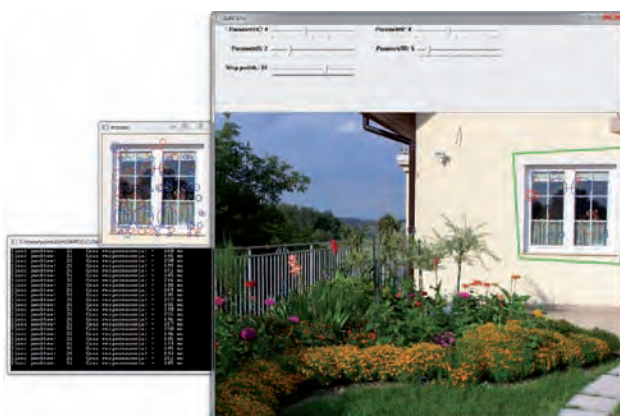
5. Implementacja rozpoznawania obrazu za pomocą algorytmu SURF

Na rys. 8 i 9 pokazano działające implementacje rozpoznawania pojedynczego obiektu na obrazie uzyskiwanym z kamery.



Rys. 8. Rozpoznawanie obrazu za pomocą sekwencyjnego algorytmu SURF uruchamianego na CPU

Fig. 8. Image recognition using sequential SURF algorithm running on the CPU



Rys. 9. Rozpoznawanie obrazu za pomocą równoległego algorytmu SURF uruchamianego na GPU

Fig. 9. Image recognition using parallel SURF algorithm running on the GPU

6. Podsumowanie

Wyniki przeprowadzonych badań pokazały, że algorytm w wersji równoległej jest zdolny przetwarzać bardzo duże ilości punktów charakterystycznych w metodzie SURF w bardzo krótkich odcinkach czasowych, a zwiększanie ich liczby w nieznaczny sposób zwiększa czas obliczeń. Dla małych ilości tych punktów algorytm sekwencyjny jest szybszy. Zastosowana technologia CUDA jest jednym z najlepszych spośród znanych do przetwarzania obrazów ze względu na zoptymalizowaną architekturę dla tego typu danych. W przyszłości planowane jest przeniesienie implementacji algorytmu równoległego metody SURF na platformę OpenCL i WebCL.

Bibliografia

1. Bay H., Ess A., Tuytelaars T., Luc Van Gool: *SURF: Speeded Up Robust Features*, Computer Vision and

Image Understanding (CVIU), vol. 110, no. 3, 2008, 346–359.

2. Evans Ch.: *Notes on the OpenSURF Library*, 2009.
3. Hess R.: *An Open-Source SIFT Library*, MM'10 October 25–29, Firenze Italy, 2010.
4. [www.d2.mpi-inf.mpg.de/surf?q=surf] – CUDA SURF – A real-time implementation for SURF | D2.
5. Sanders J., Kandrot E.: *CUDA by Examples: An Introduction to General-Purpose GPU Programming*. NVIDIA Corporation, Addison-Wesley, 2010.
6. Schweiger F., Zeisl B., Georgel P et al.: *Maximum Detector Response Markers for SIFT and SURF*, VMV, 2009.

Time aspects of SURF algorithm in sequential and parallel version

Abstract: This article presents results of our work concerned possibility of implementation of algorithm for assigning key points using SURF algorithm and CUDA technology. The work also compares time of execution of these applications.

Keywords: SURF, CUDA, GPGPU, image recognition, parallel programming

dr inż. Magdalena Szymczyk

Ukończyła studia na Wydziale Elektrotechniki Automatyki i Elektroniki (obecnie Elektrotechniki Automatyki Informatyki i Elektroniki) Akademii Górniczo-Hutniczej w Krakowie w 1988 r. i uzyskała tytuł magistra inżyniera elektronika (specjalność automatyka). Po ukończeniu studiów doktoranckich na Wydziale EAIiE AGH w 1999 r. otrzymała tytuł doktora nauk technicznych w dyscyplinie informatyka. W swoich badaniach naukowych zajmuje się programowaniem równoległym, bezpieczeństwem i niezawodnością systemów informatycznych i systemów wbudowanych. Opublikowała 60 artykułów naukowych.

e-mail: Magdalena.Szymczyk@agh.edu.pl



dr inż. Piotr Szymczyk

Ukończył studia na Wydziale Elektrotechniki Automatyki i Elektroniki (obecnie Elektrotechniki Automatyki Informatyki i Elektroniki) Akademii Górniczo-Hutniczej w Krakowie w 1988 r. i uzyskał tytuł magistra inżyniera elektronika (specjalność automatyka). Po ukończeniu studiów doktoranckich na Wydziale EAIiE AGH w 1997 r. obronił pracę doktorską i otrzymał tytuł doktora nauk technicznych w dyscyplinie informatyka. W swoich badaniach naukowych zajmuje się systemami wbudowanymi, systemami czasu rzeczywistego, systemami operacyjnymi, systemami operacyjnymi czasu rzeczywistego oraz rozpoznawaniem obrazu. Opublikował 60 artykułów naukowych i jedną książkę.

e-mail: Piotr.Szymczyk@agh.edu.pl

