

**Zachodniopomorski Uniwersytet
Technologiczny w Szczecinie
Wydział Elektryczny**



Radosław Rajczyk

nr albumu: 23804

**Implementacja algorytmu Viterbiego
z wykorzystaniem biblioteki OpenCL**

Praca dyplomowa magisterska
kierunek: Automatyka i Robotyka
specjalność: Systemy sterowania procesami przemysłowymi

Opiekun pracy:
dr hab. inż. Przemysław Mazurek
Katedra Przetwarzania Sygnałów i Inżynierii Multimedialnej
Wydział Elektryczny

Szczecin, 2016

Spis treści

1	Streszczenie	3
2	Wstęp	4
2.1	Przetwarzanie obrazu i jego rola w automatyce przemysłowej	4
2.2	Istotność szybkości obliczeń w problemach wizji // maszynowej	4
2.3	Cel, zakres i zastosowania pracy	5
3	Metody równoległego przetwarzania danych	6
3.1	Wielowątkowość CPU dla aplikacji C/C++	6
3.1.1	Biblioteka POSIX dla systemów Unix	6
3.1.2	OpenMP - wieloplatformowe API	6
3.1.3	Wielowątkowość w standardzie C++11	6
3.2	Programowanie równoległe z wykorzystaniem GPU	6
3.2.1	Architektura GPU i porównanie względem CPU	6
3.2.2	Biblioteka OpenCL	6
4	Algorytm Viterbiego	7
4.1	Opis działania i zastosowania	7
4.2	Implementacja w języku C++	7
4.2.1	Wersja szeregową	7
4.2.2	Wersja równoległa - C++11	7
4.2.3	Wersja równoległa - OpenCL	7
5	Wyniki badań doświadczalnych implementacji algorytmu Viterbiego	8
5.1	Porównanie czasu działania dla implementacji szeregowej, wielowątkowej oraz z wykorzystaniem biblioteki OpenCL	8
5.2	Porównanie szybkości algorytmów dla różnych konfiguracji sprzętowych	8
6	Wnioski końcowe	9
7	Załącznik B	10
8	Załącznik A	11
9	Bibliografia	12
	Spis rysunków	13

Rozdział 1

Streszczenie

To jest streszczenie

Rozdział 2

Wstęp

2.1 Przetwarzanie obrazu i jego rola w automatyce przemysłowej

W zagadnieniach technik pomiarowych oraz analizy otoczenia coraz częściej stosowane są rozwiązania wykorzystujące systemy wizyjne. Do najpopularniejszych zastosowań przemysłowych wizji maszynowej należą [1]:

- inspekcja elementów na linii technologicznej
- określanie właściwej orientacji i położenia elementów
- identyfikacja produktów
- pomiary metrologiczne

W automatyce przemysłowej gdzie do zagadnień inspekcji wcześniej niezbędna była ocena wizualna człowieka, obecnie powszechnie stosuje się systemy wizyjne, w których skład wchodzi kamera przemysłowa, czujniki wyzwalające (np. na bazie pozycji) oraz urządzenie odpowiadające za proces decyzyjny. Występują również rozwiązania w postaci systemów wbudowanych, gdzie inteligentna kamera oprócz akwizycji obrazu zajmuje się jego przetwarzaniem i analizą, wykorzystując własny procesor.[1][2]

Sprawdzanie orientacji i położenia elementów w przemyśle jest wykorzystywane między innymi w technologii montażu, gdzie informacje z urządzeń wizyjnych są wykorzystywane przez manipulatory przemysłowe do zautomatyzowanego montażu, sortowania oraz paletyzacji wyrobów.[1]

Identyfikowanie produktów na bazie obrazu cyfrowego jest wykorzystywane przy sortowaniu oraz monitorowaniu przepływu elementów i lokalizacji wąskich gardeł. Przykładowe metody identyfikacji to stosowanie kodów kreskowych i kodów DataMatrix.[1]

2.2 Istotność szybkości obliczeń w problemach wizji maszynowej

Większość praktycznych zastosowań przetwarzania obrazu jako dodatkowej informacji w sterowaniu jednym bądź grupą urządzeń, wymaga akwizycji oraz wykonywania obliczeń w czasie rzeczywistym. Oznacza to, że wybrany algorytm wykorzystywany do analizy obrazu cyfrowego, wraz z resztą niezbędnego kodu, musi posiadać czas wykonania spełniający narzucone przez sterowany system.

Dla zastosowań przemysłowych, gdzie monitorowane obiekty poruszają się z dużą prędkością, szybkość podjęcia decyzji przez system wizyjny może być wąskim gardłem dla danej gałęzi linii produkcyjnej. Do wykonania decyzji o, np. usunięciu wadliwego produktu z przenośnika taśmowego, niezbędna jest podrozdział 2 [4]

2.3 Cel, zakres i zastosowania pracy

podrozdział 3

Rozdział 3

Metody równoległego przetwarzania danych

3.1 Wielowątkowość CPU dla aplikacji C/C++

To jest rozdział 1 [6],[5]

3.1.1 Biblioteka POSIX dla systemów Unix

To jest podrozdział 1 rozdziału 1

3.1.2 OpenMP - wieloplatformowe API

To jest podrozdział 2 rozdziału 1

3.1.3 Wielowątkowość w standardzie C++11

To jest podrozdział 3 rozdziału 1

3.2 Programowanie równoległe z wykorzystaniem GPU

To jest rozdział 2[6],[3] [5]

3.2.1 Architektura GPU i porównanie względem CPU

To jest podrozdział 1 rozdziału 2

3.2.2 Biblioteka OpenCL

To jest podrozdział 2 rozdziału 2

Rozdział 4

Algorytm Viterbiego

4.1 Opis działania i zastosowania

To jest rozdział 1

4.2 Implementacja w języku C++

To jest rozdział 2

4.2.1 Wersja szeregową

To jest podrozdział 1 rozdziału 2

4.2.2 Wersja równoległa - C++11

To jest podrozdział 2 rozdziału 2

4.2.3 Wersja równoległa - OpenCL

To jest podrozdział 3 rozdziału 2

Rozdział 5

Wyniki badań doświadczalnych implementacji algorytmu Viterbiego

5.1 Porównanie czasu działania dla implementacji szeregowej, wielowątkowej oraz z wykorzystaniem biblioteki OpenCL

To jest rozdział 1

5.2 Porównanie szybkości algorytmów dla różnych konfigura- cji sprzętowych

To jest rozdział 2

Rozdział 6

Wnioski końcowe

To jest zakończenie

Rozdział 7

Załącznik B

To jest załącznik B

Rozdział 8

Załącznik A

To jest załącznik A

Rozdział 9

Bibliografia

- [1] Cognex. Introduction to machine vision. http://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf, 2016.
- [2] E.R. Davies. *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Elsevier, 225 WYman Street, Waltham, 02451, USA, 2012.
- [3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [4] Przemysław Mazurek Grzegorz Matczak. Line following with real-time viterbi trac-before-detect algorithm. *Przegląd Elektrotechniczny*, 1/2017:69–72, 2017.
- [5] Wikipedia. Acriting language. http://en.wikipedia.org/wiki/Scripting_language, 2014.
- [6] Wikipedia. Scripting language. http://en.wikipedia.org/wiki/Scripting_language, 2014.

Spis rysunków