

Jacek KONIECZNY, Adam ŁUCZAK

POLITECHNIKA POZNAŃSKA, KATEDRA TELEKOMUNIKACJI MULTIMEDIAŁNEJ I MIKROELEKTRONIKI

Skalowalny algorytm estymacji ruchu dla systemów rozproszonych

Mgr inż. Jacek KONIECZNY

Uzyskał tytuł mgr inż. na Politechnice Poznańskiej w roku 2008. Jest doktorantem w Katedrze Telekomunikacji Multimedialnej i Mikroelektroniki na Politechnice Poznańskiej. Jego głównym obszarem działalności naukowej jest kompresja sekwencji wizyjnych dla potrzeb telewizji wielowidokowej, a także algorytmy kompresji obrazu i dźwięku wraz z ich implementacją na PC oraz FPGA.



e-mail: jkoniczny@multimedia.edu.pl

Dr inż. Adam ŁUCZAK

Tytuł magistra uzyskał w 1997 na Politechnice Poznańskiej. Od roku 1997 prowadził badania w zespole przetwarzania obrazów. W roku 2001 uzyskał tytuł doktora. Od roku 2002 jest adiunktem w Katedrze Telekomunikacji Multimedialnej i Mikroelektroniki. Główna działalność badawcza związana jest ze sprzętową implementacją algorytmów kompresji i przetwarzania sekwencji wizyjnych na układach FPGA. Obecnie głównym nurtem zainteresowań są systemy przetwarzania rozproszonego wykorzystujące układy FPGA.



e-mail: aluczak@multimedia.edu.pl

Streszczenie

W pracy zaprezentowano autorski algorytm umożliwiający realizację procesu estymacji ruchu w czasie rzeczywistym przy wykorzystaniu systemów rozproszonych. Jest to algorytm należący do grupy tzw. szybkich algorytmów estymacji ruchu i umożliwia estymację ruchu z rozdzielczością poniżej jednego okresu próbkowania. Zaproponowany algorytm jest skalowalny i posiada budowę modułową, umożliwia elastyczne dostosowanie do rozmiaru układu oraz udostępnia mechanizm kontroli liczby cykli zegara niezbędnych do estymacji ruchu. W rezultacie możliwe jest dostosowanie częstotliwości zegara do możliwości układu przy zapewnieniu przetwarzania w czasie rzeczywistym.

Słowa kluczowe: algorytm skalowalny, estymacja ruchu, systemy rozproszone, przetwarzanie w czasie rzeczywistym.

A scalable motion estimation algorithm for distributed systems

Abstract

In this paper a novel scalable algorithm for real-time motion estimation dedicated for distributed systems is presented. The proposed algorithm has modular structure and provides ability to flexibly adjust the global clock-rate required for real-time processing. This is achieved using hierarchical structure of the algorithm, which assumes division of the whole motion estimation process into independent processing stages (Figs. 1, 3, 4) and introducing special mechanism for controlling the allowed number of computation cycles. The algorithm modularity provides additional profits like ability to choose various methods for each processing stage independently and scalability of the circuit structure resulting in more efficient hardware implementation. Flexible clock-rate adjustment enables real-time processing with various types of computational platforms, with special regards to distributed systems consisting of many low-performance units. An example of scalable performance of the algorithm in distributed systems is presented. In order to increase the image resolution in real-time processing the processed image is shared between many processing units (Fig. 5). An exemplary system used for evaluation was created using Digilent "Starter Boards" with Xilinx Spartan-3 XC3S1000 FPGA circuits connected with NOC (Network On Chip) (Fig. 6). In the presented implementation six Spartan-3 circuits were able to estimate motion vectors with half-pel accuracy in real-time for HD resolution (1920x1080) video sequence with 25 frames per second.

Keywords: scalable algorithm, motion estimation, distributed systems, real-time processing.

1. Wstęp

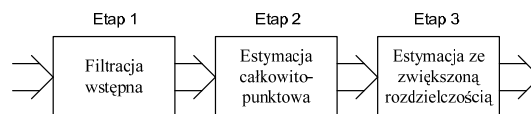
Estymacja ruchu jest nieodzownym etapem kodowania wykorzystywanym we wszystkich powszechnie używanych standardach kodowania obrazu ruchomego, takich jak MPEG-2 [1], czy AVC [2]. Jednocześnie stanowi najbardziej złożoną operację wykonywaną w koderze, zużywając niemal 70% jego mocy obliczeniowej. Wybór odpowiedniego algorytmu estymacji ruchu decyduje zatem w znacznym stopniu o efektywności całego procesu kodowania [3, 4]. Fakt ten ma szczególne znaczenie w przypadku rozwiązań wymagających przetwarzania w czasie rzeczywistym.

Rosnące wymagania dotyczące rozdzielczości kodowanych sekwencji wizyjnych powodują, iż częstotliwości taktowania produkowanych układów stają niewystarczające w przypadku przetwarzania w czasie rzeczywistym. Koniecznością staje się zatem stosowanie kilku bardzo silnych procesorów, dedykowanego akceleratora sprzętowego wspomagającego programowy proces kodowania lub też stosowanie rozwiązań czysto sprzętowych. W każdym z przypadków uzyskujemy zrównoleglenie procesu. Poszukiwania odpowiednich struktur systemów o wysokiej wydajności każą zwrócić szczególną uwagę na systemy rozproszone, w których zadania dzielone są pomiędzy jednostki tworzące system, przez co wymagania dotyczące częstotliwości taktowania oraz stopnia złożoności elementów systemu maleją wielokrotnie.

W artykule zaprezentowano nowy, skalowalny algorytm hierarchiczny estymacji. Algorytm ten należy do grupy tzw. szybkich algorytmów estymacji ruchu i wykorzystuje mechanizm pasowania bloków. Zaproponowany algorytm stanowi w pełni skalowalne rozwiązanie przeznaczone do efektywnej implementacji sprzętowej, umożliwiając przetwarzanie w czasie rzeczywistym z rozdzielczością wektorów ruchu przekraczającą jeden okres próbkowania. Z drugiej strony algorytm udostępnia funkcjonalność pozwalającą na łatwe dostosowanie złożoności oraz częstotliwości taktowania układu kosztem dokładności estymowanych wektorów ruchu. Możliwość elastycznego doboru częstotliwości taktowania wymaganej do pracy algorytmu w czasie rzeczywistym umożliwia jego wykorzystanie na różnych platformach obliczeniowych, w tym, w szczególności, w systemach rozproszonych składających się z wielu małych jednostek obliczeniowych.

2. Algorytm skalowalny

Autorskim rozwiązaniem jest wprowadzenie silnej modularności oraz skalowalności zegara w prezentowanym algorytmie estymacji ruchu. Z tego względu cały proces estymacji podzielony został na niezależne etapy. Ogólną koncepcję tego podziału przedstawiono na rys. 1. Dla każdej pary porównywanych bloków – bloku obrazu kodowanego oraz bloku obrazu referencyjnego – wykonywane jest przetwarzanie potokowe: pierwszy etap obejmujący filtrację obrazu, następnie estymacja całkowito-punktowa i wreszcie, wymagająca interpolacji obrazu, estymacja zwiększająca rozdzielczość wektorów ruchu. Dodatkowo etap estymacji całkowito-punktowej także posiada strukturę potokową.



Rys. 1. Główne etapy przetwarzania

Fig. 1. The main stages of processing pipeline

Druga z cech - skalowalność częstotliwości zegara taktującego układ, umożliwia kontrolę czasu przeznaczanego na estymację

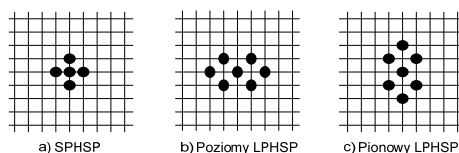
pojedynczego wektora ruchu, co z kolei pozwala na przerwanie procesu, gdy ustalony limit zostanie przekroczony. Oznacza to możliwość ograniczenia częstotliwości zegara taktującego układ przy zachowaniu zdolności do przetwarzania w czasie rzeczywistym. Należy jednak liczyć się z pogorszeniem dokładności estymowanych wektorów ruchu, chociaż układ nadal generował będzie poprawne wyniki przy wymaganej liczbie ramek na sekundę.

W zaproponowanym rozwiązaniu kierowano się przede wszystkim chęcią zapewnienia możliwie dużej modularności całego procesu. Umożliwiło to niezależne modyfikowanie dowolnej części procesu przetwarzania bez ingerencji w pozostałe elementy łańcucha, a osiągnięta w ten sposób separacja kolejnych etapów przetwarzania pozwala na elastyczny dobór wykorzystywanych metod i wariantów niezależnie dla każdego etapu. Łańcuch przetwarzania składa się następujących etapów:

Etap 1. Filtracja wstępna stanowi niezwykle ważny krok poprawiający dokładność całego procesu estymacji ruchu. W zaproponowanym algorytmie istnieje możliwość elastycznego wyboru stosowanej metody filtracji wstępnej. W zależności od wybranego wariantu przetwarzaniu wstępnemu poddawana jest tylko obraz referencyjny, tylko obraz kodowany lub filtrowane są oba obrazy.

Etap 2. Estymacja całkowito-punktowa obejmuje wszystkie operacje związane z estymacją ruchu wykonywaną zarówno na obrazach zdycymowanych, jak i obrazach o oryginalnej rozdzielczości. W celu zapewnienia efektywnej implementacji sprzętowej algorytmu, autorzy oparli swe rozwiązanie na algorytmie estymacji ruchu zwanego PHS (ang. Predict Hexagon Search) [5].

Algorytm PHS umożliwia wyznaczenie wektora ruchu w niewielkiej liczbie korek oraz posiada regularną strukturę, co sprawia, iż nadaje się dobrze do implementacji sprzętowej. Przeszukiwanie pola ruchu w algorytmie PHS odbywa się z wykorzystaniem specjalnych schematów przeszukiwania, których kształt oraz rozmiar nie są przypadkowe [5]. Schematami tymi są mały oraz duży sześciokąt – tzw. SPHSP oraz LPHSP, przedstawione na rys. 2.



Rys. 2. Schematy przeszukiwania PHS: a) SPHSP, b) Poziomy LPHSP oraz c) Pionowy LPHSP

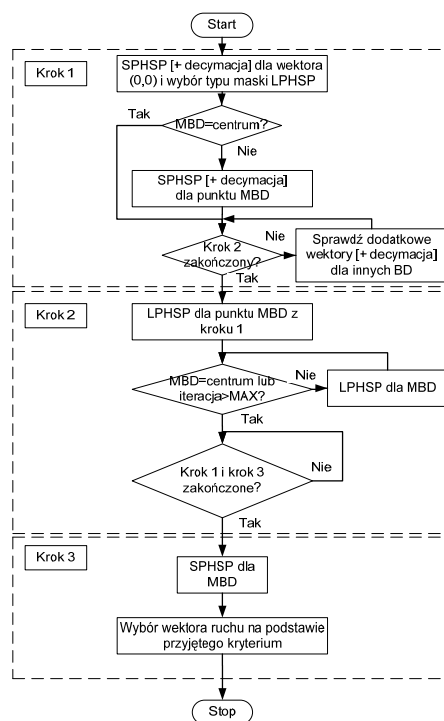
Fig. 2. PHS search patterns: a) SPHSP, b) Horizontal LPHSP and c) Vertical LPHSP

Schemat SPHSP, ze względu na duże skupienie punktów przeszukiwania, używany jest do dokładnego przebadania obszaru wokół punktu centralnego. Z kolei schemat LPHSP umożliwia sprawdzenie szerszego otoczenia w pojedynczej iteracji, co przyspiesza proces estymacji ruchu przy kodowaniu bardziej dynamicznych sekwencji.

W rozwiązaniu zaproponowanym przez autorów oryginalny algorytm PHS został zmodyfikowany tak, by zapewnił on skalowalność struktury oraz umożliwił przetwarzanie równoległe dla wielu par porównywanych bloków. Nowa, skalowalna struktura algorytmu pozwala na podział procesu całkowito-punktowej estymacji ruchu na kroki, co z kolei umożliwia bardziej efektywną implementację sprzętową. Na rys. 3 przedstawiono graf decyzyjny zaproponowanego algorytmu całkowito-punktowej estymacji ruchu. Algorytm korzysta z mechanizmu podobnego do przetwarzania potokowego, co pozwala w większym stopniu wykorzystywać równoległość jakie oferuje przetwarzanie sprzętowe. Bez wprowadzenia takiego podziału część zasobów byłaby wykorzystywana tylko raz przez cały okres estymacji pojedynczego wektora ruchu, co byłoby zwyczajnym marnotrawstwem zasobów.

Podobnie jak miało to miejsce w oryginalnym algorytmie PHS, decyzje o zmianie położenia schematu przeszukiwania podejmowane są na podstawie położenia punktu MBD (ang. Minimum Block Distortion), czyli współrzędnych wektora o najmniejszym znalezionym dotąd niedopasowaniu. Niedopasowanie bloku BD

(ang. Block Distortion) obliczane jest na podstawie różnicy pomiędzy dwoma blokami porównywanych obrazów z użyciem wybranej miary podobieństwa, np. MAE (średnia wartość bezwzględna różnic) lub MSE (średnia wartość kwadratów różnic).

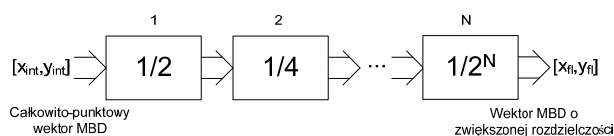


Rys. 3. Graf algorytmu etapu estymacji całkowito-punktowej
Fig. 3. Hierarchical full-pel motion estimation algorithm graph

Odrębnym zagadnieniem jest opcjonalna decymacja porównywanych obrazów w kroku 1 estymacji całkowito-punktowej. Prezentowany algorytm umożliwia taką decymację, co pozostaje w zgodzie z ogólną koncepcją algorytmów hierarchicznych.

Dla potrzeb implementacji sprzętowej w algorytmie wprowadzono także mechanizm zliczania iteracji wykonanych w kroku 2. Pozwala to na kontrolę długości całego cyklu przetwarzania dla pojedynczego wektora ruchu związanej z częstotliwością taktowania układu. Zaproponowane rozwiązanie umożliwia zatem przetwarzanie w czasie rzeczywistym z wybraną częstotliwością taktowania układu, powodując jednak, iż dokładność obliczanych wektorów ruchu zależy będzie od charakteru kodowanej treści.

Etap 3. Ostatnim etapem zaproponowanego algorytmu jest estymacja ze zwiększoną rozdzielczością, umożliwiającą dostosowanie rozdzielczości obliczanych wektorów ruchu do wymagań użytkownika oraz dostępnych zasobów sprzętowych. Zaimplementowanie takiej funkcjonalności wymaga zastosowania specyficznej, skalowalnej struktury algorytmu. W prezentowanym rozwiązaniu osiągnięto ten cel poprzez podział etapu estymacji ze zwiększoną rozdzielczością na podetapy, których liczba zależy od docelowej rozdzielczości wektorów ruchu. Schemat blokowy dla tego etapu estymacji ruchu przedstawiono na rys. 4.



Rys. 4. Schemat blokowy etapu estymacji ze zwiększoną rozdzielczością
Fig. 4. Block diagram of the sub-pel motion estimation algorithm

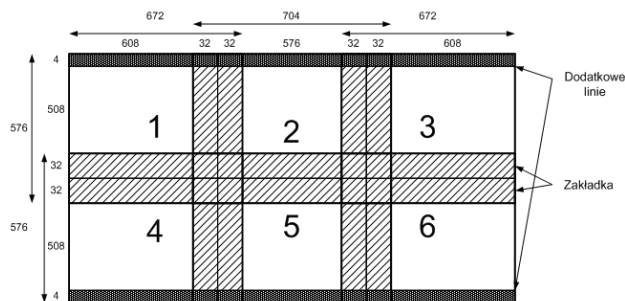
Podczas estymacji ze zwiększoną rozdzielczością sprawdzane jest bezpośrednio otoczenie punktu MBD wyznaczonego w etapie estymacji całkowito-punktowej. Pierwszy z podetapów sprawdza wektory o rozdzielczości równej $1/2$ okresu próbkowania i wybiera spośród nich nowy punkt MBD. Proces ten jest następnie powta-

rzany w kolejnych podetapach, których liczba zależy od wymaganej docelowej rozdzielczości wektorów ruchu, przy czym każdy z podetapów zwiększa ich rozdzielczość dwukrotnie. W rezultacie rozdzielczość kolejnych podetapów wynosi odpowiednio: $1/2$, $1/4$, ..., $1/2^N$, gdzie N jest liczbą podetapów.

Podobnie jak w etapie estymacji całkowito-punktowej, w każdej chwili czasowej w każdym podetapie przetwarzana jest inna para porównywanych bloków, co pozwala znacznie zwiększyć częstotliwość wykorzystania zasobów. Z kolei modularność algorytmu umożliwia użycie różnych schematów interpolacji, zwiększających rozdzielczość obrazu referencyjnego do wymaganej wartości. W prezentowanym rozwiązaniu zdecydowano się na użycie schematów interpolacji opartych na standardzie MPEG-2 [1] oraz AVC [2], zmodyfikowanych jednak w taki sposób, by umożliwiły one interpolację do dowolnej rozdzielczości. Należy również zauważyć, iż zaprezentowana struktura algorytmu pozwala także na zastosowanie szerokiej gamy metod niewymagających interpolacji. Jedynym warunkiem jaki owe metody powinny spełniać jest możliwość ich dekompozycji na niezależne etapy, co pozwoli na efektywną implementację sprzętową (np. [6]).

3. Implementacja algorytmu w środowisku rozproszonym

W przypadku implementacji algorytmu dla potrzeb przetwarzania w czasie rzeczywistym z użyciem niewielkiej jednostki obliczeniowej głównym problemem jest maksymalna częstotliwość taktowania takiego układu. Pomimo mechanizmu kontroli czasu przetwarzania, jaki udostępnia omawiany algorytm, dla każdej częstotliwości taktowania układu istnieje maksymalna rozdzielczość przetwarzanego obrazu, powyżej której warunek przetwarzania w czasie rzeczywistym nie jest spełniony.



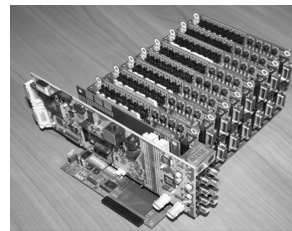
Rys. 5. Przykładowy podział obrazu o rozdzielczości 1920x1080 na sześć części dla potrzeb estymacji ruchu z użyciem systemu rozproszonego (rozmiar makrobloku: 16, obszar przeszukiwania: 32)

Fig. 5. Example of dividing 1920x1080 resolution image into six parts for motion estimation with distributed system (macroblock size: 16, search area: 32)

Możliwym rozwiązaniem wspomnianego problemu jest zastosowanie systemu rozproszonego składającego się z wielu jednostek obliczeniowych, w którym każda z jednostek wykonuje estymację ruchu dla pewnego obszaru przetwarzanego obrazu. Przykładowy podział obrazu na 6 nakładających się części przedstawiono na rys. 5. Wektory ruchu obliczone w ten sposób dla obszarów znajdujących się w obrębie zakładki danego fragmentu obrazu są odrzucane. W przypadku, gdy rozdzielczość przetwarzanego obrazu nie jest wielokrotnością rozmiaru makrobloku używanego przez algorytm estymacji ruchu, obraz o oryginalnej rozdzielczości należy rozszerzyć o dodatkowe linie – pionowe lub poziome.

4. Przykład implementacji sprzętowej

Przykładem implementacji sprzętowej prezentowanego algorytmu w systemie rozproszonym jest układ składający się z 8 płyt Digilent „Starter Board” z układami FPGA Spartan 3 XC3S1000 połączonych siecią NOC (ang. Network On Chip) pokazany na rys. 6.



Rys. 6. System przetwarzania rozproszonego (8 płyt Spartan 3 + płyta Video Starter Kit z układem Virtex II firmy Xilinx)

Fig. 6. Distributed processing system (8 Spartan 3 boards + Video Starter Kit board with Xilinx Virtex II circuit)

Z uwagi na niewielkie zasoby zastosowanych układów FPGA, w tym zwłaszcza liczbę dostępnych pamięci BRAM, zaimplementowane zostały następujące etapy algorytmu skalowalnego: estymacja całkowito-punktowa oraz estymacja ze zwiększoną rozdzielczością do $1/2$ okresu próbkowania obrazu z użyciem schematu interpolacji opartego na standardzie MPEG-2.

Pojedynczy układ FPGA z rodziny Spartan 3 umożliwia estymację ruchu w czasie rzeczywistym z użyciem wspomnianej implementacji algorytmu jedynie dla obrazów o standardowej rozdzielczości telewizyjnej, a więc 720x576. Dzięki zastosowaniu metody podziału obrazu pomiędzy jednostki rozproszone w sposób opisany w punkcie 3 możliwa stała się estymacja ruchu w czasie rzeczywistym dla obrazów o rozdzielczości HD. Przykładowo sześć takich układów umożliwia estymację ruchu dla obrazów o rozdzielczości 1920x1080 przy 25 ramkach na sekundę. W tym celu obraz wejściowy dzielony jest na 6 nakładających się części o rozdzielczościach nie większych niż 720x576.

5. Podsumowanie

W pracy zaprezentowano autorski algorytm umożliwiający realizację procesu estymacji ruchu w czasie rzeczywistym przy wykorzystaniu systemów rozproszonych. Autorzy przedstawili rozwiązanie cechujące się silnie modułową strukturą oraz możliwością skalowalnego doboru częstotliwości taktowania układu dla potrzeb przetwarzania w czasie rzeczywistym. Cel ten osiągnięto bazując na strukturze hierarchicznej algorytmu, zakładającej podział procesu estymacji ruchu na niezależne etapy przetwarzania oraz wprowadzając specjalny mechanizm kontroli liczby dozwolonych cykli obliczeniowych. Powyższe cechy umożliwiają jego wykorzystanie na różnych platformach obliczeniowych, w tym, w szczególności, w systemach rozproszonych składających się z wielu małych jednostek obliczeniowych.

6. Literatura

- [1] ITU-T Rec.H.262/ISO/IEC 13818-2, Information technology - Generic coding of moving pictures and associated audio information: Video - Part 2.
- [2] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, ISO/IEC 14496 10 Advanced Video Coding 3rd Edition, July 2004.
- [3] Ce Zhu, Xiao Lin, Lap-Pui Chau: Hexagon-based search pattern for fast block motion estimation, IEEE Trans. Circuits and Systems for Video Techn., Vol. 12, May 2002, pp.: 349-355.
- [4] Yilong L., Oraintara S.: Complexity comparison of fast block-matching motion estimation algorithms, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, 17-21 May 2004, pp.: 341-344.
- [5] Tsung-Han T., Yu-Nan P.: A Novel Predict Hexagon Search Algorithm for Fast Block Motion Estimation on H.264 Video Coding, The 2004 IEEE Asia-Pacific Conference on Circuits and Systems, Dec. 6-9, 2004, pp.: 609-612.
- [6] Kwon D. N., Agathoklis P., Driessen P.: Sub-pixel accuracy motion estimation using linear approximate model of the error criterion function, Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on 24-26 Aug. 2005, pp.: 261-264.