

Akceleracja obliczeń wejściowego stopnia filtrującego hybrydowego algorytmu maskowania błędów transmisji obrazu stałego

Przetwarzanie cyfrowe danych wizyjnych znajduje coraz szersze zastosowanie. Jednym z ważnych aspektów przetwarzania jest zapewnienie żądanej jakości sygnału, co w praktyce oznacza konieczność stosowania metod i technik redukcji i zwalczania wpływu błędów, występujących w czasie transmisji tego rodzaju sygnałów. Pośród wielu technik eliminacji błędów kanału transmisyjnego ważne miejsce zajmują metody maskowania zakłóceń. Koncepcja ta ma szczególne znaczenie w systemach o rygorystycznych wymaganiach co do czasu przetwarzania, gdzie niedopuszczalne bądź niemożliwe jest wykorzystanie technik klasycznych (retransmisja, nadmiarowanie sygnału). Zasadniczą zaletą tej klasy metod to przyjęcie transmitowanego sygnału w takiej postaci, w jakiej jest dostępny (bez ingerencji w jego treść przed transmisją) i wykorzystanie różnorodnych mechanizmów do minimalizacji lub eliminacji wykrytych zakłóceń.

Prezentowane wyniki prac dotyczą akceleracji obliczeń wykonywanych przez wejściowy stopień filtrujący hybrydowego algorytmu maskowania błędów **HECA**, opisanego szczegółowo w [5, 6, 7], dzięki wyodrębnieniu i zrównolegleniu pewnych fragmentów algorytmu. Poddano dyskusji oczekiwane korzyści w postaci skrócenia czasu obliczeń wykonywanych przez filtr, w odniesieniu do poniesionych nakładów sprzętowych realizacji praktycznej.

ALGORYTM MASKOWANIA BŁĘDÓW

Omawiany algorytm jest szczegółowo w [5, 6, 7]. W niniejszym artykule przedstawione zostaną jego główne założenia oraz specyfikacja umożliwiająca wyeksponowanie niezależnych fragmentów algorytmu wejściowego stopnia filtrującego, które mogą być zrównoległone.

Koncepcja algorytmu

Hybrydowy algorytm maskowania błędów jest przeznaczony do polepszenia jakości sygnału obrazu stałego na drodze detekcji oraz korekcji możliwych do wykrycia usterek. Algorytm projektowano przy następujących założeniach:

- przetwarzanie obrazu odbywa się przy podziale na bloki o rozmiarach 8×8 punktów, w konsekwencji rozmiar pionowy i poziomy obrazu jest wielokrotnością liczby osiem,
- obraz jest monochromatyczny i składa się z punktów w jednym

Uzyskany z dekodera sygnał z błędami transmisji poddaje się przetwarzaniu. Jego celem jest eliminacja usterek, dających się wyeliminować oraz minimalizacja wpływu na odtworzony obraz tych usterek, których algorytm nie jest w stanie wyeliminować.

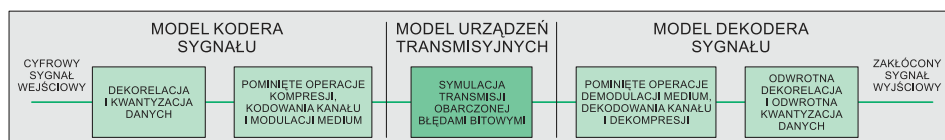
Przebieg sygnału

Schemat przetwarzania sygnału w algorytmie HECA przedstawiono na rys. 2. Pierwszy etap przetwarzania obejmuje eliminację danych wysokoczęstotliwościowych, które według badań przedstawionych w [5] nie niosą informacji. Drugi etap obejmuje wstępne przekształcenie obrazu z postaci współczynników **DCT** (postać częstotliwościowa, w której jest on transmitowany) do postaci przestrzennej. Po odtworzeniu obraz jest analizowany celem wykrycia tzw. wzorców błędów, charakterystycznych dla transformaty DCT. Uzyskane informacje są zapisywane w macierzy nazywanej mapą błędów. Trzeci etap przetwarzania operuje na danych w postaci współczynników DCT i dokonuje detekcji nieprawidłowych wartości poszczególnych współczynników. Szczegółowe działanie mechanizmu detekcji wzorców i nieprawidłowych współczynników opisano w [5]. Wynikiem działania etapów drugiego i trzeciego jest macierz, zawierająca informacje o współczynnikach uznanych za błędne. Mapa błędów jest podstawą realizacji etapu czwartego, który wykonuje zadanie korekcji współczynników DCT (mechanizm korekcji szczegółowo opisano w [5]) oraz właściwą operację przekształcenia skorygowanego obrazu do postaci przestrzennej. Ostatni, piąty etap przetwarzania to filtracja, korygująca usterki nieprawidłowego poziomu średniej jasności poszczególnych bloków (przyczynę powstawania tej usterki oraz działanie filtru opisano w [5]).

Wejściowy stopień filtrujący

Zastosowanie transformaty jako stopnia dekorrelacji danych powoduje, że po pierwsze otrzymuje się dane o szczytkowym stopniu korelacji, dzięki czemu jest możliwa dobra ich kompresja, po drugie zaś liczba danych przeznaczonych do transmitowania zmniejsza się radykalnie. Właściwość ta została wykorzystana do zaprojektowania dolnoprzepustowego filtru wejściowego. Jego działanie opiera się na fakcie, że dane wysokiej częstotliwości (położone w macierzy danych DCT poniżej określonej odwrotnej przekątnej granicznej g) mają zerową wartość w poprawnie przetransmitowanym obrazie. Obecność niezerowej danej oznacza błąd transmisji, dlatego też wejściowe dane DCT są poddane wstępnej filtracji dolnoprzepustowej, w celu likwidacji możliwych przekła-

mań współczynników wysokiej częstotliwości. Szczegółowy algorytm działania filtru przedstawiono na rys. 3. Po pobraniu macierzy Y danych, dzielonej w trakcie działania algorytmu na $M \times N$ kwadratowych bloków po 8×8 punktów (liczb całkowitych) następuje etap analizy kolejnych bloków $y_{m,n}$, gdzie $1 < m < M$, $1 < n < N$. W poszczególnych blokach jest prowadzona analiza kolejnych współczynników w celu wyznaczenia tych z nich, których suma wewnętrznych współrzędnych (i, j) przekracza

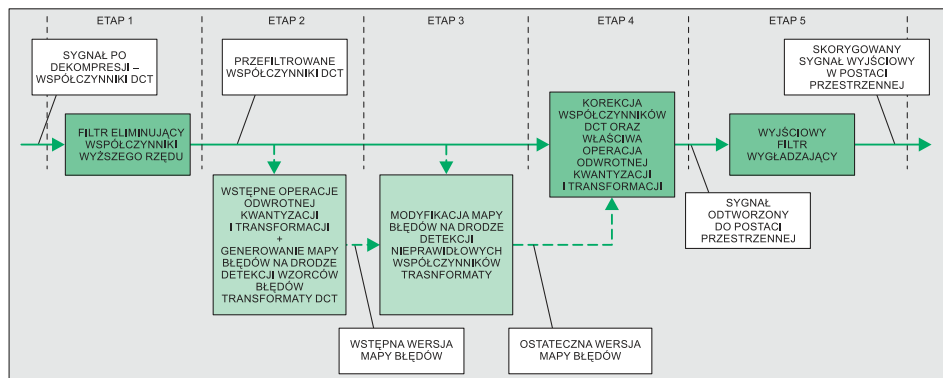


■ Rys. 1. Przyjęty w badaniach model systemu transmisji sygnału cyfrowego

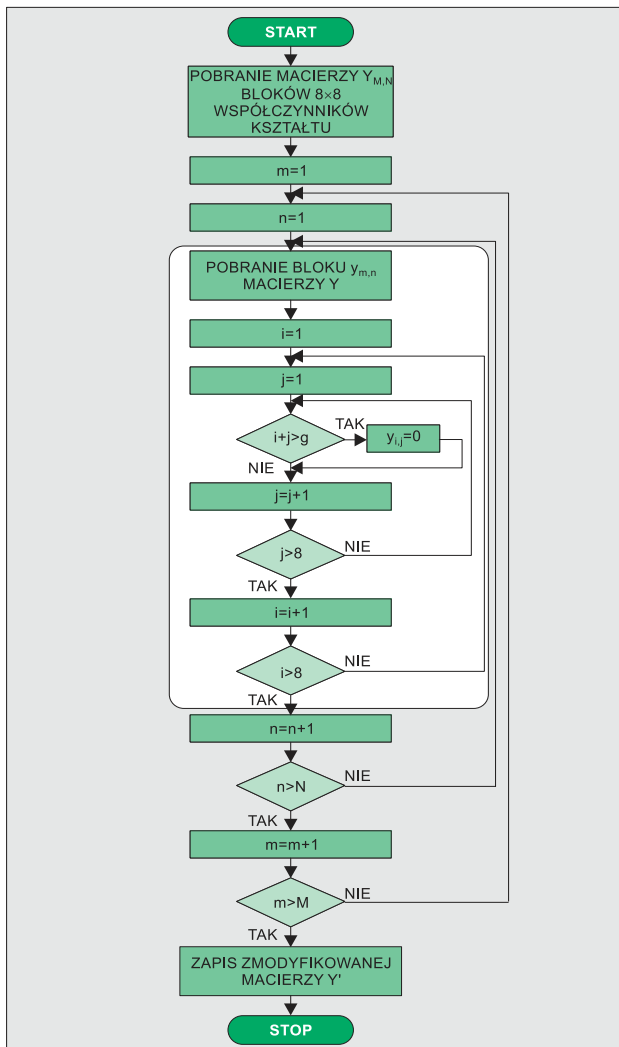
z 256 stopni szarości, a zatem jest reprezentowany przez macierz zawierającą liczby całkowite od 0 do 255,

- sygnał jest transmitowany zgodnie z modelem, przedstawionym na rys. 1. Szerszą dyskusję modelu zawarto w [5, 6].

* Uniwersytet Zielonogórski, Instytut Informatyki i Elektroniki,
e-mail: W.Zajac@iie.uz.zgora.pl, G.Andrzejewski@iie.uz.zgora.pl



■ Rys. 2. Schemat przetwarzania danych w algorytmie HECA



■ Rys. 3. Algorytm działania wejściowego stopnia filtrującego

zdefiniowany w pracy [5] próg graniczny g . Jeśli tak jest, dany współczynnik jest zerowany. Po zakończeniu analizy powstaje zmodyfikowana macierz współczynników DCT Y' .

MODYFIKACJA ALGORYTMU

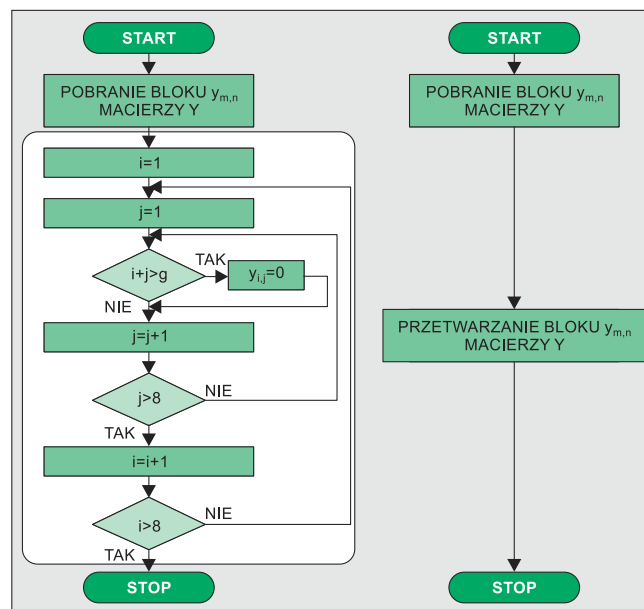
W celu przygotowania algorytmu przetwarzania danych wejściowego stopnia algorytmu HECA do implementacji częściowo równoległej jest konieczne wyznaczenie fragmentów algorytmów realizacji poszczególnych etapów procesu, których charakter umożliwia realizację współbieżną [3]. Na rys. 3 wyeksponowany fragment algorytmu odpowiada za przetwarzanie pojedynczego bloku obrazu. Ze względu

na fakt, że do przetwarzania dowolnego bloku nie są wymagane dane pochodzące z analizy jakiegokolwiek innego bloku, można ten fragment wyekstrahować z algorytmu głównego, tworząc podalgorytm przetwarzania pojedynczego bloku (rys. 4). W powyższym podalgorytmie wyeksponowano blok przetwarzania zastępując go uogólnionym blokiem funkcyjnym *przetwarzanie bloku* $y_{m,n}$ macierzy Y .

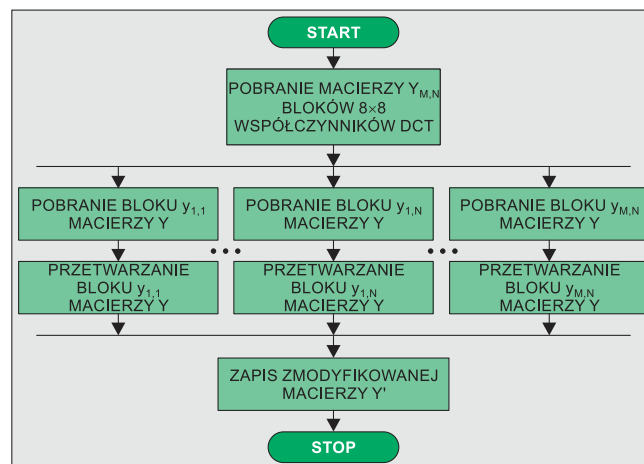
Na rys. 5 przedstawiono specyfikację uogólnionego modelu algorytmu obliczeń wykonywanych

przez wejściowy stopień filtrujący hybrydowego algorytmu maskowania błędów HECA w postaci współbieżnej. Tak przedstawiony algorytm nadaje się do implementacji w strukturach zapewniających współbieżność obliczeń.

Ze względu na duży współczynnik współbieżności rozpatrywanego modelu (jednocześnie można wykonywać obliczenia na $M \times N$ blokach), wydaje się nieefektywna implementacja na platformie programowej, w której realizacją zajmuje się wielordzeniowy system mikroprocesorowy. Przetwarzane obrazy testowe miały rozmiar 32×32



■ Rys. 4. Podalgorytm przetwarzania pojedynczego bloku danych



■ Rys. 5. Algorytm wejściowego stopnia filtrującego w wersji współbieżnej

bloków danych. Dla prezentowanego modelu w systemie musiało by funkcjonować $R=M*N+1=1025$ rdzeni procesorowych.

Dużo bardziej efektywnymi platformami wydają się platformy sprzętowa oraz sprzętowo-programowa. Typowym przykładem platformy sprzętowej są reprogramowalne struktury **FPGA** bądź **CPLD**. Obecnie dostępne zasoby takich układów odpowiadają milionom bramek logicznych (np. 5 milionów bramek w układzie Spartan-3 FPGA firmy Xilinx) przy ich znikomym małych kosztach (w granicach 3 USD) [8].

Alternatywą mogą być również struktury sprzętowo-programowe, nazywane mikrosystemami cyfrowymi, w których programowalna matryca jest również zintegrowana z rdzeniem mikroprocesorowym. Struktury takie zapewniają uzyskanie dobrych efektów implementacyjnych, wykorzystując atuty elastyczności realizacji programowych (wykonywanych przez rdzeń mikroprocesorowy) oraz szybkości realizacji sprzętowych (wykonywanych przez struktury reprogramowalne FPGA) [1, 2].

Można przypuszczać, że zrównoleglenie modelu algorytmu obliczeń wykonywanych przez wejściowy stopień filtrujący hybrydowego algorytmu maskowania błędów HECA zaowocuje nawet tysiąc-krotnym przyspieszeniem obliczeń. Wiąże się to niestety również z tysiąc-krotnym wzrostem zapotrzebowania na zasoby sprzętowe realizacji praktycznej. Nie musi być to jednak znaczący problem wobec dużych zasobów i możliwości dynamicznej rekonfiguracji systemów sprzętowych. W szczególności własność dynamicznej rekonfiguracji zwalnia nieużywane zasoby sprzętowe i umożliwia wykorzystanie ich do innych celów na dalszych etapach obliczeń [4].

* * *

W artykule przedstawiono propozycję metody przyspieszenia obliczeń wykonywanych przez blok wejściowego stopnia filtrującego

hybrydowego algorytmu maskowania błędów HECA. Prezentowana metoda opiera się na zrównolegleniu pewnych fragmentów algorytmów, których charakter zapewnia ich realizację współbieżną. Poddano dyskusji zalety proponowanych modyfikacji oraz oszacowano koszty realizacji rozumiane jako nakłady sprzętowe. Jako platformę realizacyjną zaproponowano dynamicznie rekonfigurowalne matryce sprzętowe klasy FPGA. Dalsze badania zostaną ukierunkowane na praktyczne aspekty realizacji sprzętowej omówionych systemów maskowania zakłóceń w obrazach wizyjnych.

LITERATURA

- [1] Andrzejewski G.: *Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych*, Zielona Góra, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, 2003
- [2] Andrzejewski G.: *Hierarchical Petri nets for digital controller design, in Design of embedded control systems*, New York, Springer, 2005
- [3] Andrzejewski G., Skowroński Z.: *Zrównoleglenie algorytmów sterowania w systemach klasy PLC*, *Pomiary Automatyka Kontrola*, 2006, nr 6, wyd. spec.
- [4] Czarniecki R., Deniziak S.: *Resource Constrained Cosynthesis of Self-reconfigurable SOPCs*, *Design and Diagnostics of Electronic Circuits and Systems, DDECS'07*, IEEE, Volume, Issue, 11-13 April 2007
- [5] Zając W.: *Korekcja błędów transmisji cyfrowego obrazu monochromatycznego z wykorzystaniem algorytmu ukrywania błędów*, Rozprawa doktorska. Politechnika Wrocławska, 2001.
- [6] Zając W.: *An Error Concealment Algorithm for Digital Image Transmission*, *Proceedings of XIV International Symposium on Computer and Information Sciences*, Kusadasi Turcja, October 1999
- [7] Zając W.: *A Hybrid Method of Error Elimination in Digital Image Transmission*, *Proceedings of the 5TH International Conference MIXDES'98*, Łódź 1998
- [8] <http://www.xilinx.com>

Artykuł recenzowany



Instytut Informatyki i Elektroniki Uniwersytetu Zielonogórskiego

Instytut Informatyki i Elektroniki Uniwersytetu Zielonogórskiego został utworzony 13 maja 1991 roku. Inicjatorami jego powołania byli: **doc. dr inż. Antoni Wysocki** oraz **dr hab. inż. Janusz Szajna**, który został pierwszym dyrektorem Instytutu. Od 1997 roku dyrektorem jest prof. zw. dr hab. inż. Marian Adamski.

Obecnie w Instytucie jest zatrudnionych 39 osób na stanowiskach dydaktycznych oraz 6 osób na stanowiskach administracyjnych.

Instytut tworzą trzy zakłady:

- Zakład Elektroniki i Układów Mikroprocesorowych – kierownik: dr hab. inż. Andrzej Olencki, prof. UZ;
- Zakład Inżynierii Komputerowej – kierownik: dr inż. Marek Węgrzyn;
- Zakład Technik Informatycznych – kierownik: dr inż. Wojciech Zając.

Przy Instytucie funkcjonują trzy koła naukowe:

- fantASTIC, które zostało utworzone na początku lat 90., opiekunowie: dr inż. Marek Węgrzyn, dr inż. Agnieszka Węgrzyn, mgr inż. Arkadiusz Bukowiec, mgr inż. Michał Doligalski, mgr inż. Tomasz Gratkowski, mgr inż. Jacek Tkacz, mgr inż. Remigiusz Wiśniewski;
- Studenckie Naukowe Koło Grafiki Komputerowej, które zostało utworzone na początku roku akademickiego 2002/2003, opiekunem jego jest dr inż. Wojciech Zając, dr inż. Andrzej Popławski oraz mgr inż. Sebastian Pawlak;
- Studenckie Koło Naukowe Informatyki UZ. NET, które zostało utworzone w styczniu 2005 roku, opiekunami jego są: prof. zw. dr hab. inż. Marian Adamski, dr inż. Grzegorz Łabiak, mgr inż. Piotr Bubacz.

Studenci, zaangażowani w prace kół naukowych, mają okazję

rozвивać swoje zainteresowania i umiejętności, mogą także pochwalić się sukcesami. Katarzyna Skrzypek ze Studenckiego Koła Grafiki Komputerowej została laureatką w konkursie na grafikę do kalendarza reklamowego, Łukasz Chodorski z koła UZ. NET wziął udział w programie praktyk Microsoft Internship i odbył trzymiesięczną praktykę w głównej siedzibie firmy Microsoft w Redmond (USA).

Instytut realizuje badania naukowe m. in. w ramach projektów: ACEP (1991–2008, międzynarodowy projekt w ramach programu TEMPUS), program COHERENCE, projekt 5. Programu Ramowego Unii Europejskiej REASON pt.: *Research and Training Action for System on Chip Design* (2001–2005), projekty i granty KBN.

Od roku 2003 Instytut organizuje cykliczne konferencje naukowe, połączone z warsztatami szkoleniowymi KNWS. Ideą konferencji jest połączenie klasycznej formuły forum wymiany informacji naukowej z zajęciami, podnoszącymi kwalifikacje i umiejętności pracowników naukowych oraz nauczycieli akademickich. Konferencje od wielu lat są objęte patronatem Polskiego Towarzystwa Informatycznego, Oddział Wielkopolski oraz J. M. Rektora Uniwersytetu Zielonogórskiego.

Instytut organizuje także międzynarodowe konferencje *International Workshop on Discrete – event System Design*, DESDes'01. Pierwsza odbyła się w 2001 roku. Wśród uczestników byli naukowcy z Niemiec, Portugalii, Wielkiej Brytanii, Czech, Białorusi, Japonii, Hiszpanii, Ukrainy i Polski.

Praca naukowa i dydaktyczna nie byłaby pełna bez kooperacji z partnerami przemysłowymi regionu województwa lubuskiego. Od wielu lat z dobrym skutkiem realizujemy wspólne przedsięwzięcia z takimi firmami, jak: 4sytem Polska, ADB Polska, Calmet, Gedia Poland, IBM, LUMEL oraz Max Elektronik.