

Załącznik nr 2a: Autoreferat

1. Dane osobowe

Imię i nazwisko: Grzegorz Pastuszak

2. Edukacja i uzyskane dyplomy

1996-2001 – jednolite studia magisterskie: Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska, specjalność: mikroelektronika

2003-2006 – studia doktoranckie: Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska, uzyskany stopień doktora nauk technicznych w zakresie elektroniki, specjalność: techniki multimedialne (z wyróżnieniem), Tytuł rozprawy doktorskiej: „Optimization of hardware architectures of binary codecs in compression of visual data”.

3. Informacje o dotychczasowym zatrudnieniu

2001-2002 - Fujitsu Group: FFC (Tokyo), Fujitsu Devices (Yokohama): projektant układów scalonych

2006-obecnie – Instytut Radioelektroniki, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska: adiunkt

4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 Ustawy z dnia 14 marca 2003r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki:

a) Tytuł osiągnięcia naukowego:

Jednotematyczny cykl publikacji pod zbiorczym tytułem „**Algorytmy i architektury koderów sprzętowych w kompresji danych wizyjnych w czasie rzeczywistym**”

b) Wykaz publikacji stanowiących osiągnięcie naukowe:

Publikacje w czasopismach indeksowanych w bazie ISI JCR (9)

- [1] M. Jakubowski, **G. Pastuszak**, “A New Multi-Path Scheme for Adaptive Computation-Aware Motion Estimation”, *Opto-Electronics Review*, vol. 15, no. 2, pp. 118-124, 2007. (IF=1.01, MNiSW:20, wkład:30%)
- [2] **G. Pastuszak**, „A High Performance Architecture of the Double-Mode Binary Coder for H.264.AVC”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 949-960, July 2008. (IF:2.95, MNiSW:24, wkład:100%)
- [3] M. Jakubowski, **G. Pastuszak**, “Block-based motion estimation algorithms - a survey”, *Opto-Electronics Review*, vol. 21, no. 1, pp. 86-102, 2013. (IF:1.279, MNiSW:20, wkład:30%)
- [4] **G. Pastuszak**, M. Jakubowski, “Adaptive Computationally-Scalable Motion Estimation for the Hardware H.264/AVC Encoder”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23 no. 5, pp. 802-812, May 2013. (IF:2.26, MNiSW:35, wkład:80%)
- [5] M. Roszkowski, **G. Pastuszak**, “Intra Prediction for the Hardware H.264/AVC High Profile Encoder,” *Journal of Signal Processing Systems*, vol. 76, no. 1, pp. 11-17, July 2014. (IF:0.564, MNiSW:20, wkład:45%)

- [6] **G. Pastuszak**, M. Trochimiuk, "Architecture Design of the High-Throughput Compensator and Interpolator for the H.265/HEVC Encoder," *Journal of Real-Time Image Processing*, vol. no. June 2014. (IF=1.111, MNiSW:25, wkład:55%)
- [7] **G. Pastuszak**, "Hardware Architectures for the H.265/HEVC Discrete Cosine Transform," *IET Image Processing*, E-First Articles, 2014. (IF:0.676, MNiSW:20, wkład:100%)
- [8] **G. Pastuszak**, "Flexible Architecture Design for H.265/HEVC Inverse Transform," *Circuits, Systems & Signal Processing*, Online First Articles, 2014. (IF:1.264, MNiSW:25, wkład:100%)
- [9] **G. Pastuszak**, "Architecture Design of the H.264/AVC Encoder based on Rate-Distortion Optimization", *IEEE Transactions on Circuits and Systems for Video Technology*, Early Access, 2015. (IF:2.259, MNiSW:35, wkład:100%)

Książki (1):

- [10] **G. Pastuszak**, „Video Compression from the Hardware Perspective,” chapter in: Constantin Volosencu (ed.), "Cutting Edge Research in New Technologies," pp. 233-256, Intech, ISBN 978-953-51-0463-6, 2012. (MNiSW:5, wkład:100%)

Publikacje recenzowane (konferencje zagraniczne) po uzyskaniu stopnia doktora (12)

- [11] M. Jakubowski, **G. Pastuszak**, „Multi-Path Adaptive Computation-Aware Search Strategy for Block-Based Motion Estimation” *The International Conference on Computer as a Tool: IEEE EUROCON 2007*, pp. 175-181, Warsaw 2007. (MNiSW:3, wkład:30%)
- [12] **G. Pastuszak**, "Architecture Design of the Double-Mode Binarization for High-Profile H.264/AVC Compression," *IEEE Workshop on Signal Processing Systems (SIPS'2007)*, Shanghai, China, 17-19 October, 2007. (MNiSW:3, wkład:100%)
- [13] **G. Pastuszak**, "Transforms and Quantization in the High-Throughput H.264/AVC Encoder Based on Advanced Mode Selection," *IEEE Annual Symposium on VLSI (ISVLSI 2008)*, Montpellier, France, 7-9 April 2008. (MNiSW:1, wkład:100%)
- [14] M. Jakubowski, **G. Pastuszak**, „A Hardware-Oriented Variable Block-Size Motion Estimation Method for H.264/AVC Video Coding,” *Signal Processing - Algorithms, Architectures, Arrangements, and Applications (SPA 2008) & New Trends in Audio and Video (NTAV 2008)*, Poznań, Poland, 25-27 September 2008. (MNiSW:1, wkład:30%)
- [15] M. Jakubowski, **G. Pastuszak**, „An Adaptive Computation-Aware Algorithm for Multi-Frame Variable Block-Size Motion Estimation in H.264/AVC,” *International Conference on Signal Processing and Multimedia Applications (SIGMAP 2009)*, pp. 122-125, Milan, Italy, 7-10 July 2009. (MNiSW:3, wkład:30%)
- [16] M. Jakubowski, **G. Pastuszak**, „Data Reuse in Two-Level Hierarchical Motion Estimation for High Resolution Video Coding,” *International Conference on Signal Processing and Multimedia Applications (SIGMAP 2010)*, pp. 159-162, Athens, Greece, 26-28 July 2010. (MNiSW:3, wkład:30%)
- [17] M. Roszkowski, **G. Pastuszak**, „Intra Prediction Hardware Module for High-Profile H.264/AVC Encoder,” *Signal Processing - Algorithms, Architectures, Arrangements, and Applications (SPA 2010)*, Poznań, Poland, 23-25 September 2010. (MNiSW:10, wkład:30%)
- [18] **G. Pastuszak**, M. Jakubowski, "Hardware Implementation of Adaptive Motion Estimation and Compensation for H.264/AVC," *Picture Coding Symposium (PCS 2012)*, pp. 369-372, Kraków, 7-9 May 2012. (MNiSW:10, wkład:80%)
- [19] A. Abramowski, **G. Pastuszak**, "A Novel Intra Prediction Architecture for the Hardware HEVC Encoder," *16th Euromicro Conference on Digital System Design*, pp. 429-436, Santander, Spain, Sept. 2013. (MNiSW:10, wkład:30%)
- [20] **G. Pastuszak**, M. Trochimiuk, "Architecture Design and Efficiency Evaluation for the High-Throughput Interpolation in the HEVC Encoder," *16th Euromicro Conference on Digital System Design*, pp. 423-428, Santander, Spain, Sept. 2013. (MNiSW:10, wkład:55%)

[21]G. Pastuszak, "FPGA architectures of the quantization and the dequantization for video encoders," *17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 290-293, Apr. 2014. (MNiSW:10, wkład:100%)

[22]A. Abramowski, G. Pastuszak, "A double-path intra prediction architecture for the hardware H.265/HEVC encoder," *17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 27-32, Apr. 2014. (MNiSW:10, wkład:30%)

c) Omówienie celu naukowego ww. prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania

Ze względu na dużą złożoność algorytmów kompresji kodowanie sekwencji wizyjnych w warunkach czasu rzeczywistego wymaga zazwyczaj dedykowanych implementacji sprzętowych koderów. W literaturze naukowej opisano wiele rozwiązań całościowych i cząstkowych umożliwiających obsługę rozdzielczości do 1080p@30fps lub nawet większych. Na rynku dostępnych jest również wiele implementacji komercyjnych. Istniejące rozwiązania stosują jednak wiele uproszczeń, co skutkuje znacznym zmniejszeniem efektywności kompresji. Nie są one również efektywne z punktu widzenia zużywanych zasobów sprzętowych. W szczególności duże zużycie zasobów logicznych znacznie zwiększa koszt ich implementacji w układach FPGA. Innym ograniczeniem jest brak skalowalności pomiędzy efektywnością kompresji i szybkością.

Głównym celem prowadzonych przeze mnie prac badawczych i rozwojowych było opracowanie efektywnych implementacji sprzętowych koderów stosowanych w kompresji wideo. Efektywność odnosi się do minimalizacji potrzebnych zasobów sprzętowych przy danej przepustowości, maksymalizacji przepustowości przy ograniczeniach na ilość zasobów, maksymalizacji jakości przy stałej przepływności, minimalizacji przepływności bitowej przy stałej jakości, oraz minimalizacji poboru energii. W ramach prowadzonych prac zaproponowano szereg metod optymalizacji sprzętowej koderów wizyjnych. Bazując na tych metodach opracowano architektury poszczególnych modułów stanowiących części koderów sprzętowych standardów kompresji wizyjnej H.264/AVC oraz H.265/HEVC. W przypadku kodera H.264/AVC przeprowadzono integrację modułów i uruchomiono cały koder w układzie FPGA w warunkach czasu rzeczywistego. Efektywne implementacje sprzętowe algorytmów kompresji wizyjnej wymagały w wielu przypadkach modyfikacji tych algorytmów w stosunku do wersji programowych. Poniższe punkty omawiają poszczególne prace badawczo-rozwojowe z punktu widzenia opracowanych modułów: kodera entropijnego, estymacji ruchu, predykcji INTRA, pętli rekonstrukcji i wyboru trybu kodowania.

Koder entropijny

Prace nad koderem H.264/AVC rozpoczęły się jeszcze w trakcie studiów doktoranckich i dotyczyły implementacji kodera entropijnego/binarnego wykorzystującego koder arytmetyczny (CABAC – *Context Adaptive Binary Arithmetic Coder*). Opracowany koder entropijny umożliwiał przetwarzanie dwóch lub więcej binarnych symboli w jednym taktie zegara. W rozprawie doktorskiej udowodniono zostało, że podejście to pozwala na zwiększenie przepustowości i efektywności sprzętowej mierzonej jako stosunek przepustowości do ilości zasobów sprzętowych.

Prace badawcze bezpośrednio po doktoracie dotyczyły opracowania kodera entropijnego umożliwiającego kodowanie dla dwóch trybów wyspecyfikowanych w standardzie H.264/AVC. Oprócz trybu CABAC zaimplementowano tryb CAVLC (*Context Adaptive Variable Length Coder*), który cechuje się mniejszą złożonością obliczeniową i jednocześnie mniejszą efektywnością kompresji (9-14%). Przy projektowaniu kodera obsługującego oba tryby wykorzystano technikę współdzielenia zasobów sprzętowych na etapie binaryzacji oraz wspólnej pamięci kontekstu (sąsiedztwa). Takie podejście było możliwe dzięki temu, że koder wideo generuje strumień stosując jeden lub drugi tryb. W przypadku niezależnych ścieżek obliczeniowych przypisanych do tych trybów efektywność sprzętowa byłaby zmniejszona, gdyż jedna z nich jest zawsze nieaktywna. Opracowana architektura kodera entropijnego pozwala również na generowanie nagłówków plastrów (*slice*) i parametrów

sekwencji/obrazu przy użyciu współdzielonego modułu binaryzacji. Wyjściem kodera entropijnego jest strumień elementarny na poziomie warstwy NAL (*Network Abstraction Layer*). Wyniki implementacji pokazały, że stosując współdzielenie zredukowano ilość zasobów konieczną przy implementacji niezależnych ścieżek kodowania entropijnego dla dwóch trybów. Prace dotyczące kodera entropijnego opisałem w artykułach [2, 12].

Estymacja i kompensacja ruchu

Estymacja ruchu jest najbardziej złożonym etapem w kodowania wideo. Tradycyjne podejście do problemu estymacji ruchu w implementacjach sprzętowych polega na sprawdzeniu wszystkich możliwych wektorów ruchu w zadanym zakresie przeszukiwania przy użyciu różnych wariantów tablic systolicznych. Takie podejście wiąże się jednak z użyciem dużej ilości zasobów sprzętowych i/lub zawężeniem zakresu przeszukiwania. Choć znane są algorytmy szybkiego wyszukiwania nie są one zazwyczaj stosowane w realizacjach sprzętowych z uwagi na nieregularność przepływu danych. W ramach prowadzonych przeze mnie prac zaprojektowano i zrealizowano architektury modułu estymacji i kompensacji ruchu stosujące nową organizację przepływu danych. Zamiast tablic systolicznych zaproponowano macierz (8×8) pamięci pozwalającą na dostęp do bloku próbek 8×8 w jednym cyklu zegara. Odpowiednie adresowanie poszczególnych pamięci oraz dwuwymiarowe rotacje pozycji odczytanych próbek w ramach bloku pozwalają uzyskać predykcję 8×8 odpowiadającą dowolnemu wektorowi ruchu. Powyższe podejście jest wspólne dla architektur zaprezentowanych w kolejnych publikacjach dotyczących sprzętowych realizacji estymacji ruchu [4, 6, 18]. Pierwsza z nich dotyczy implementacji dla kodera standardu H.264/AVC, natomiast druga opisuje architekturę opracowaną dla standardu H.265/HEVC. W porównaniu do powszechnie stosowanych architektur estymacji ruchu opracowane implementacje uzyskują znacznie lepszą efektywność sprzętową mierzoną stosunkiem szybkości do ilości zasobów logicznych.

W implementacji standardu H.264/AVC [4, 18] zastosowano nowe podejście do problemu estymacji pod-pikselowej. Polega ono na interpolacji próbek w pozycjach ułamkowych ($1/2$, $1/4$ i $3/4$) przed ich zapisem do macierzy pamięci (czyli przed estymacją ruchu na poziomie całkowitych pikseli). Interpolacji podlega obszar referencyjny o wymiarze 32×32 piksele używany przy dokładnej estymacji ruchu dla danego makrobloku. W efekcie możliwe jest sprawdzanie w jednym cyklu zegara dowolnego wektora ruchu z dokładnością $1/4$ piksela. Cecha ta umożliwia unifikację dwóch etapów estymacji. Ponieważ estymacja pod-pikselowa wprowadza w tym zunifikowanym podejściu niewielkie opóźnienie w stosunku do estymacji całkowito-pikselowej, cała estymacja ruchu wykonywana jest na tym samym etapie/stopniu potoku przetwarzającego makrobloki. Kolejną nową cechą architektury jest możliwość skierowania sygnału różnicowego/rezydualnego (różnica między oryginalnym blokiem i jego predykcją) uzyskanego bezpośrednio po kompensacji do dalszego przetwarzania w pętli rekonstrukcji i wyboru trybu. Bloki sygnału różnicowego zapisywane są do bufora. Bloki odpowiadające najlepszym wektorom czytane są z bufora i kierowane do dalszego przetwarzania w miarę gotowości pętli rekonstrukcji. W konsekwencji uzyskano jeden etap/stopień przetwarzania makroblokowego, który w tradycyjnym podejściu rozbijany jest na trzy.

Zmniejszenie opóźnienia w przetwarzaniu kolejnych makrobloków pozwala obliczyć predykcję wektora ruchu na etapie estymacji ruchu, ponieważ warunkiem jest znajomość trybu wybranego dla poprzedniego (lewego) makrobloku. Ma to istotne znaczenie z punktu widzenia efektywności kompresji wideo. Po pierwsze predykcja wektora ruchu pozwala na sprawdzenie trybu SKIP. Po drugie bitowy koszt różnicowego zakodowania wektora ruchu jest dodawany do kosztu SAD kolejno sprawdzanych wektorów ruchu. Stosowane w innych rozwiązaniach sprzętowych rozbieżne przetwarzania na trzy stopnie potoku makroblokowego prowadzi do dużych strat efektywności kompresji. Bez dedykowanej analizy trybu SKIP, jakość zakodowanych ramek INTER zmniejsza się średnio o 1.5 dB. Oprócz zysków na efektywności kompresji opracowana architektura zużywa najmniej zasobów logicznych w porównaniu do alternatywnych rozwiązań o zbliżonej przepustowości. Z drugiej strony koszt pamięci wewnętrznych jest duży (128kB). Nie stanowi to problemu w przypadku układów FPGA, które mają znacząco większą ilość zasobów pamięciowych. Aby zminimalizować zasoby sprzętowe przy zapewnieniu szerokiego zakresu estymacji ruchu, zastosowano dwa poziomy

dokładności przeszukiwania – zgrubny i dokładny. Pierwszy z nich ma szerszy zakres przeszukiwania (od -64 do +63 w obu wymiarach) i znajduje mniejszy obszar do przeszukiwania dokładnego (od -8 do +7 w obu wymiarach). Przy estymacji zgrubnej obrazu sekwencji wideo reprezentowane są w zredukowanej rozdzielczości (16:1). Hierarchiczna estymacja ruchu rozszerza zakres przeszukiwania przy niewielkim wzroście złożoności i niewielkich stratach efektywności kompresji (0.1dB).

Opracowana architektura modułu estymacji i kompensacji ruchu może implementować bardzo różne algorytmy wyszukiwania wektorów ruchu. Choć możliwe jest zastosowanie algorytmu pełnego przeszukiwania (full search), główną zaletą jest użycie algorytmów szybkiego wyszukiwania. W celu znalezienia najlepszego rozwiązania szereg algorytmów było analizowanych z punktu widzenia szybkości znajdowania wyniku jak również stopnia dopasowania między blokami uzyskanego dla znalezionej wektora ruchu. Przegląd wszystkich analizowanych algorytmów został zamieszczony w artykule [3]. Na podstawie analizy wyselekcjonowano trzy najskuteczniejsze schematy wyszukiwania. Schematy te są w dużym stopniu komplementarne gdyż o ich skuteczności decyduje charakter ruchu w sekwencji wideo oraz liczba dostępnych wektorów ruchu możliwych do sprawdzenia. W toku dalszych prac opracowano adaptacyjny algorytm estymacji ruchu, który wybiera jeden z trzech schematów wyszukiwania na podstawie analizy wektorów ruchu poprzednio zakodowanych makrobloków. Ze względu na korelację ruchu wektory te sprawdzane są również na początku przetwarzania bieżącego makrobloku. Dodatkowo sprawdzany jest wektor zerowy oraz predyktor obliczany jako mediana z trzech sąsiednich wektorów. Po wyszukiwaniu jednym z trzech wybranych schematów znajduje się najlepszy wektor z dokładnością całkowito-pikselową. W dalszej kolejności następuje pod-pikselowa estymacja ruchu. Jeżeli możliwe jest sprawdzenie większej liczby wektorów wykonywane jest pełne przeszukiwanie pozostałych pozycji.

Nowością w przedstawionym algorytmie jest to, że osiąga on szybką zbieżność do minimalnego poziomu błędu dopasowania przy sprawdzaniu kolejnych wektorów. Przerwanie wyszukiwania pociąga za sobą względnie niewielkie straty w efektywności kompresji. Jednakże straty zależą od liczby pominiętych wektorów. Zależność ta umożliwia skalowanie obliczeń. W szczególności złożoność obliczeniowa wymieniana może być na błąd dopasowania, co przekłada się to na efektywność kompresji. Opis całego algorytmu wraz z wynikami efektywności dopasowania zaprezentowany został w artykułach [1, 11]. W kolejnych artykułach zaproponowano rozszerzenia algorytmu umożliwiające obsługę różnych rozmiarów bloków [14], wielu ramek referencyjnych [15] oraz większy zakres wyszukiwania (dodanie poziomu zgrubnego) [16]. Algorytm w wersji podstawowej został zaimplementowany w generatorze wektorów ruchu, który jest częścią sprzętowego systemu estymacji i kompensacji ruchu omawianego wcześniej [4, 18]. Choć generacja wektorów ruchu przeprowadzana jest w odniesieniu do makrobloków 16x16, również mniejsze bloki sprawdzane są dla tych samych wektorów [14]. Estymacja na poziomie zgrubnym (zaproponowana w [16]) umożliwiającą szerszy zakres wyszukiwania zaimplementowana została jako oddzielny moduł.

Kolejne prace dotyczące estymacji ruchu rozpoczęły się wraz z pojawieniem się kolejnych wersji roboczych nowego standardu kompresji wideo. Standard H.265/HEVC stosuje jednostki predykcji PU (Prediction Unit) o rozmiarach od $8 \times 4/4 \times 8$ do 64×64 . Kodowane są kolejne jednostki CTU (Coding Tree Unit), których rozmiar jest równy zazwyczaj 64×64 . Większe rozmiary oraz wzrost liczby dopuszczalnych trybów podziału na bloki zwiększają wielokrotnie złożoność obliczeniową estymacji ruchu w stosunku do kodera H.264/AVC. Dodatkowo interpolacja pikseli na pozycjach ułamkowych wykonywana jest przy pomocy filtrów wyższych rzędów. W konsekwencji implementacje sprzętowe opisane w literaturze wymagają dużych ilości zasobów w stosunku do uzyskiwanych przepustowości.

W ramach prowadzonych prac badawczo-rozwojowych opracowałem interpolator stosujący podobne założenia projektowe jak przypadku architektury dla H.264/AVC [20]. W dalszej kolejności zaprojektowano i zrealizowano implementację, która uzyskuje znacznie lepszą efektywność sprzętową. W architekturze tej predykcja w blokach 8×8 uzyskiwana jest podobnie jak w przypadku estymacji ruchu kodera H.264/AVC. Jednakże zrezygnowano z interpolacji pozycji ułamkowych przed zapisem do macierzy pamięci. Interpolacja przed zapisem jest nieefektywna z uwagi na 16-krotny wzrost pojemności pamięci wynikający z rozmiarów CTU. W ścieżce przetwarzania danych interpolacja następuje po estymacji na poziomie całkowito-pikselowym. Choć kolejność ta jest tożsama z tradycyjnym podejściem do implementacji estymacji ruchu, obliczenia próbek w pozycjach

pod-pikselowych wykonywane są w blokach 8×8 natychmiast po fazie estymacji całkowito-pikselowej. W efekcie opóźnienie estymacji pod-pikselowej jest niewiele większe niż w przypadku architektury opracowanej dla kodera standardu H.264/AVC. W nowej architekturze (H.265/HEVC) zastosowane zostało przeplatanie sprawdzania wektorów ruchu fazy całkowito-pikselowej oraz pod-pikselowej. Przeplot dokonywany jest pomiędzy blokami należącymi do różnych PU. W efekcie w każdym cyklu zegara sprawdzany jest jeden wektor ruchu dla bloku 8×8. Interpolacja pozycji pod-pikselowych wykonywana jest równoległe ze sprawdzaniem wektorów ruchu. Predykcje próbek w kolejnych pozycjach obliczane są przy pomocy macierzy rekonfigurowalnych filtrów. Rekonfiguracja pozwala na zmiany współczynników filtrów i użycie tych samych zasobów sprzętowych do obliczeń dla poszczególnych pozycji ułamkowych. W przypadku interpolacji dwuwymiarowej wynik po interpolacji jednowymiarowej jest kierowany ponownie na wejście filtrów. Opracowany moduł estymacji ruchu może pracować przy częstotliwości zegara 400 MHz przy implementacji technologii ASIC 90nm. Umożliwia to sprawdzenie 100 wektorów ruchu dla każdego bloku 8×8, gdy kodowane są sekwencje wideo o rozdzielczości 3840×2160@30fps.

Predykcja wewnątrzramkowa INTRA

Poza predykcją międzyramkową INTER alternatywną metodą służącą do zwiększenia efektywności kompresji jest predykcja wewnątrzramkowa INTRA. Choć jest ona mniej złożona obliczeniowo niż estymacja ruchu to może ograniczać przepustowość kodera sprzętowego. Ograniczenia wynikają z wzajemnych zależności pomiędzy kolejno kodowanymi blokami o różnych rozmiarach predykcji (np. 4×4 i 8×8). Innym wyzwaniem spotykanym przy implementacji modułu predykcji INTRA jest sprawdzenie wszystkich trybów wyspecyfikowanych w danym standardzie. Pozwala to osiągnąć najlepszą efektywność kompresji. Implementacje opisane w literaturze zazwyczaj nie obsługują wszystkich trybów i stosują schematy o mniejszej efektywności sprzętowej mierzonej stosunkiem przepustowości do ilości zasobów.

W ramach prac prowadzonych z moim udziałem zaprojektowano i zrealizowano moduł predykcji INTRA kodera H.264/AVC. Moduł obsługuje wszystkie tryby wyspecyfikowane w standardzie. W szczególności dotyczy to ośmiu kierunków predykcji, trybu DC oraz trybu Plane dla trzech rozmiarów bloków (4×4, 8×8 i 16×16). Moduł predykcji wykorzystuje współdzielenie tych samych zasobów do obliczenia predykcji dla różnych bloków i kierunków. Uzyskana przepustowość pozwala generować w każdym cyklu zegara blok predykcji o rozmiarze 4×4 przy użyciu macierzy elementów przetwarzających. Jednakże równoległość generacji próbek nie wystarcza na pełne wykorzystanie przepustowości modułu ze względu na opóźnienie wprowadzane przez pętlę rekonstrukcji. W szczególności do wyznaczenia predykcji bieżącego bloku konieczny jest dostęp do zrekonstruowanych próbek brzegowych bloków poprzednio przetwarzanych. Aby zniwelować opóźnienie pętli rekonstrukcji zastosowano przeplatanie bloków o różnych rozmiarach i typach (luma/chroma). Najbardziej krytyczne jest przetwarzanie bloków 4×4, które w bezpośrednim podejściu wymusza 16 iteracji. Każda z nich składa się z predykcji i rekonstrukcji. Pomimo przeplotu z blokami o większych rozmiarach istnieją odcinki czasu, kiedy moduł predykcji INTRA musi czekać na rekonstrukcję. W związku z tym zmieniono kolejność przetwarzania bloków 4×4. W szczególności pary wzajemnie niezależnych bloków są przetwarzane kolejno bez czekania na rekonstrukcję. W efekcie liczba zależnych iteracji zmniejszyła się do 10. Weryfikacja w koderze sprzętowym pokazała, że liczba cykli czekania na rekonstrukcję jest znacznie mniejsza niż liczba cykli wykorzystywanych na generację próbek predykcji (60%-90%). Architektura i harmonogramowanie opracowanego modułu predykcji INTRA kodera H.264/AVC opisane zostały w dwóch artykułach [5, 17].

Prace badawczo-rozwojowe dotyczące predykcji INTRA prowadzone były również dla kodera standardu H.265/HEVC. Ze względu na zwiększenie liczby dostępnych trybów (rozmiary bloków i kierunki predykcji) wymagania na przepustowość modułu predykcji INTRA dla takich koderów są znacznie większe. Zaproponowane zostało rozwiązanie wykorzystujące macierz elementów przetwarzających obliczającą blok 8×8 próbek predykcji w jednym taktie zegara [19]. Macierz umożliwia obliczenia dla różnych rozmiarów bloków (4×4, 8×8, 16×16, 32×32) i wszystkich 35 kierunków predykcji. Rozwiązanie to umożliwia obsługę rozdzielczości 1080p@15fps przy

implementacji w technologii ASIC 130nm. Wyłączenie trybów 4x4 zwiększa dwukrotnie przepustowość przy pewnej stracie efektywności kompresji. Architektura umożliwia również skalowanie obliczeniowe w przypadku bloków o większych rozmiarach. W kolejnym artykule [22] zaprezentowano optymalizację polegającą na wydzieleniu ścieżki przetwarzania bloków 4x4. W efekcie uzyskano zwiększenie przepustowości do 1080p@35fps przy obsłudze wszystkich trybów.

Pętla Rekonstrukcji

Pętla rekonstrukcji składa się z potoku kolejno następujących modułów transformacji, kwantyzacji, dekwantyzacji, odwrotnej transformacji i rekonstrukcji. Opóźnienie pętli wpływa w dużym stopniu na szybkość sprawdzania trybów INTRA. Przepustowość decyduje o liczbie trybów kodowania możliwych do sprawdzenia według kryterium kosztu bitowego i zniekształceń po rekonstrukcji. Każdy tryb określony jest przez metodę predykcji (INTRA/INTER), kierunek/wektor predykcji, rozmiar bloku, rozmiar transformacji i parametr kwantyzacji. Lepszą efektywność kompresji uzyskuje się, gdy liczba sprawdzanych trybów jest duża. Implementacje sprzętowe opisane w literaturze stosują pętle rekonstrukcji o mniejszych przepustowościach. W przypadku transformacji oznacza to konieczność buforowania wyników pośrednich. Często transformacje dla bloków o różnych rozmiarach są wykonywane w oddzielnych modułach, co zwiększa ilość zasobów sprzętowych.

W ramach prac badawczo-rozwojowych nad koderem H.264/AVC zaimplementowałem wszystkie moduły pętli rekonstrukcji o przepustowości 32 próbki/współczynniki na takt zegara o częstotliwości 100MHz (FPGA Arria II) lub 200 MHz (ASIC 90nm). Przepustowość ta jest wystarczająca, aby sprawdzić kilkanaście trybów INTER i wszystkie INTRA przy kodowaniu sekwencji wideo 1080p@30Hz (ASIC). Każdy z pięciu wymienionych wcześniej modułów wprowadza opóźnienie dwóch taktów zegara. Pętla obejmuje też dodatkowe moduły transformacji Hadamard'a oraz bufora skwantowanych współczynników. Moduły te konieczne są do obsługi hierarchicznej transformacji stosowanej w blokach chromy oraz lummy Intra 16x16. Uzyskane opóźnienie pętli jest niewielkie i wynosi 14-22 cykli. Jest ono większe niż 14 cykli, gdy przetwarzane są bloki wymagające hierarchicznej transformacji. Moduły transformacji wykorzystują nowe schematy współdzielenia zasobów do obsługi różnych rozmiarów transformacji. W szczególności wykorzystano kilkanaście multiplexerów, aby rekonfigurować połączenia pomiędzy elementami arytmetycznymi. W efekcie uzyskano redukcję potrzebnych zasobów sprzętowych. W porównaniu z innymi rozwiązaniami opisanymi w literaturze efektywność sprzętowa opracowanych modułów transformacji jest większa również ze względu na brak dodatkowego buforowania. W szczególności przetwarzanie całych bloków 8x8 sprawia, że każdy stopień rejestrowy w potoku odpowiada rozmiarowi bufora stosowanego między dwoma stopniami transformacji. Architektura pętli rekonstrukcji kodera H.264/AVC opisana została w artykule [13].

Najnowsze prace dotyczą implementacji modułów transformacji w koderze i dekodek H.265/HEVC. Tradycyjne podejście do problemu implementacji transformacji bazuje na dekompozycji jednowymiarowej N-punktowej transformacji na sieć operacji dodawania i odejmowania wykonywanych jednocześnie dla wszystkich N wyjść. Podejście to wymaga bufora transpozycji o rozmiarze NxN. Ponieważ standard H.265/HEVC specyfikuje transformacje o N równym nawet 32, rozmiar bufora może wynieść 1024 16-bitowych współczynników transformaty. Implementacja takiego bufora wymaga znacznej ilości zasobów sprzętowych. Obsługa różnych rozmiarów transformacji zwiększa dodatkowo złożoność obliczeniową. Inną wadą tradycyjnego podejścia jest znaczne opóźnienie wnoszone przez moduł transformacji. W dwóch artykułach [7, 8] zaprezentowałem nowe architektury modułów transformacji i odwrotnej transformacji. Wykorzystują one dekompozycję mnożenia macierzy na fragmenty mapowane na jednostki obliczeniowe w kolejnych cyklach zegarowych. W szczególności dekompozycja dzieli wejściowy i wyjściowy blok NxN na podbloki. Podbloki wejściowe przeliczane są na sumy cząstkowe (poprzez mnożenie i sumowanie w ramach podbloku wyjściowego). W kolejnych cyklach zegarowych sumy cząstkowe są akumulowane dla danego podbloku wyjściowego. Podblok wyjściowy jest gotowy, gdy wszystkie sumy cząstkowe są zakumulowane.

Choć zastosowana dekompozycja nie eliminuje mnożeń, ma kilka ważnych zalet. Po pierwsze unifikacja jednostek obliczeniowych pozwala na efektywną ich implementację w układach scalonych (użycie jednostek DSP). Po drugie możliwe jest konfigurowanie architektury do wymaganych przepustowości/zasobów/interfejsów. Po trzecie różne rozmiary/typy transformacji mogą być obsługiwane przez zmianę mnożników. W opracowanych modułach zmieniona została kolejność obliczania podbloków wyjściowych na pierwszym stopniu transformacji. Zamiast obliczać kolejno podbloki wyjściowe odwołujące się do tych samych podbloków wejściowych (np. linia podbloków w transformacji poziomej), kolejne podbloki wyjściowe są obliczane wzdłuż drugiego wymiaru (tworzą kolumnę w transformacji poziomej). Wykonanie transformacji w tym drugim wymiarze zaczyna się natychmiast, gdy jedna kolumna/linia podbloków jest gotowa. Podstawową zaletą zmodyfikowanego porządku obliczeń jest redukcja rozmiaru bufora transpozycji do $N \times M$ 16-bitowych współczynników transformaty, gdzie M to jeden z wymiarów podbloku wyjściowego. Dodatkowym zaletą jest zmniejszenie opóźnienia, gdyż drugi stopień transformacji nie musi czekać na skończenie obliczeń na pierwszym.

Opracowane moduły transformacji wykorzystują operacje motylkowe (jeden stopień) ze względu na symetrię macierzy transformacji wyspecyfikowaną w standardzie H.265/HEVC. Umożliwia to zmniejszenie liczby jednostek obliczeniowych o połowę. Wyniki implementacji pokazują, że ilość zasobów sprzętowych koniecznych do realizacji modułów transformacji jest mniejsza w porównaniu z innymi architekturami opisanymi w literaturze pomimo zastosowania potoku w jednostkach obliczeniowych. Uzyskiwane częstotliwości zegara są zdecydowanie wyższe w przypadku opracowanych modułów. Pewną ich słabością jest zmniejszona przepustowość przy większych rozmiarach transformacji (zwiększona liczba akumulacji sum cząstkowych dla każdego podbloku wyjściowego). W porównaniu z najbardziej zaawansowanymi rozwiązaniami o zbliżonej ilości zasobów logicznych przepustowość jest mniejsza tylko dla $N=32$.

Choć architektury modułów kwantyzacji i dekwantyzacji są znane i łatwe w implementacji możliwe są optymalizacje prowadzące do redukcji zasobów. W szczególności w artykule [21] zaproponowałem redukcję zasobów logicznych w przypadku implementacji w układach FPGA dla standardów H.264/AVC i H.265/HEVC. Optymalizacja wykorzystuje fakt, że dostępne jednostki DSP mają większą dokładność bitową mnożeń niż potrzeba przy bezpośredniej implementacji modułów kwantyzacji i dekwantyzacji. Następująca po mnożeniu operacja zmiennego przesuwania bitowego jest wyeliminowana lub ograniczona poprzez przesunięcia bitowe wartości mnożników. Ponieważ przy równoległym kwantowaniu/dekwantowaniu wielu współczynników (np. 4×4) wykorzystuje się te same mnożniki uzyskuje się znaczną redukcję zużywanych elementów logicznych.

Wybór trybu kodowania

Największą efektywność kompresji uzyskują kodery wideo wykorzystujące analizę kosztów zakodowania różnych trybów na bazie estymacji zniekształceń (D-distortion) oraz kosztu bitowego (R-Rate). Analiza RD wiąże się jednak z dużą złożonością obliczeniową, szczególnie w przypadku estymacji kosztu bitowego w trybie CABAC (kodowanie arytmetyczne). Rozwiązania koderów sprzętowych opisane w literaturze stosują alternatywne metody estymacji kosztów takie jak suma wartości bezwzględnych różnic (SAD) oraz suma wartości bezwzględnych transformowanych różnic (SATD). W konsekwencji jakość zakodowanego materiału wideo jest mniejsza o ok. 0.3-0.5 dB (przy tej samej przepływności) w stosunku do koderów używających analizy RD. W zaproponowanym przeze mnie podejściu wykorzystuję analizę RD z uproszczeniami, które umożliwiają efektywną implementację sprzętową. W przypadku trybu CAVLC estymacja kosztu bitowego nie bierze pod uwagę międzyblokowych zależności schematów kodowania liczby niezerowych współczynników w blokach 4×4 . W przypadku trybu CABAC estymacja kosztu bitowego wykorzystuje fakt, że liczba bitów wyjściowych koderu arytmetycznego jest silnie skorelowana z liczbą bitów/symboli wejściowych. W związku z tym koszt bitowy obliczany jest jako liczba bitów/symboli wejściowych po fazie binaryzacji. Estymacja energii zniekształceń wymaga układów mnożących do obliczenia funkcji kwadratowej. Zastosowane podejście wykorzystuje tablice bitowe LUT zamiast układów mnożących. Każde z

opisanych uproszczeń wprowadza błędy estymacji, jednakże odpowiadające im straty jakości zakodowanych sekwencji wizyjnych są stosunkowo niewielkie (ok. 0.1dB).

Oprócz kryterium kosztu efektywność kompresji zależy również od liczby kandydujących trybów skierowanych do analizy i metody ich preselekcji. Programowy model referencyjny JM koder standardu H.264/AVC wyznacza na podstawie miary SAD/SATD po jednym kandydacie dla każdego rozmiaru bloku w trybie INTER. Implementacje sprzętowe opisane w literaturze powielają te podejście, ale bez analizy RD. W opracowanym przeze mnie podejściu preselekcja zakłada wybór zbioru kilku (minimum 8) najlepszych kandydatów dla bloków 16x16 i skierowanie ich do analizy RD. Sama analiza RD umożliwia sprawdzenie mniejszych bloków zawartych w każdym z bloków 16x16. Do preselekcji najlepszych kandydatów użyta została lista rankingowa, która analizuje na bieżąco koszty SAD (suma wartości bezwzględnych) odpowiadających kolejnym wektorom i wybiera osiem dotychczas najmniejszych dla danego makrobloku. W przypadku predykcji INTRA wszystkie tryby sprawdzane są według kryterium RD, podobnie jak w koderze JM.

Algorytm wyboru trybu z użyciem uproszczonej analizy RD został zaimplementowany w koderze sprzętowym H.264/AVC [9, 10]. W szczególności zaprojektowano i zrealizowano moduły estymacji kosztu bitowego i wyboru trybu. Estymacja zniekształceń włączona została do modułu rekonstrukcji. Przepustowość tych modułów została dopasowana do zaprojektowanej wcześniej pętli rekonstrukcji. Umożliwia to na analizę dużej liczby różnych trybów. Liczba ta jest większa niż w przypadku programowego koder referencyjnego JM. W rezultacie uzyskiwane jest dodatkowe zwiększenie efektywności kompresji, które częściowo kompensuje straty wynikłe z uproszczeń analizy RD. Dla trybów INTER, moduł wyboru trybu analizuje koszty zarówno dla całych bloków 16x16 jak również dla ich partycji 16x8, 8x16 i 8x8. Estymacja kosztu bitowego obejmuje dane pikselowe (współczynniki) oraz kontrolne (tryby, wektory ruchu, kierunki predykcji). Algorytm preselekcji zmapowany został na oddzielny moduł w koderze sprzętowym. Umożliwia on zbuforowanie kilkunastu kandydujących predykcji w blokach 16x16, które są otrzymywane z modułu kompensatora trybów INTRA i INTER. Moduł zawiera listę rankingową, która na bieżąco odrzuca najslabszych kandydatów i kieruje do pętli rekonstrukcji najlepszych. Tryby INTRA 4x4 i 8x8 nie podlegają preselekcji i wszystkie kierowane są do pętli rekonstrukcji. Wszystkie tryby skierowane do pętli rekonstrukcji podlegają analizie RD.

Opracowany koder sprzętowy [9, 10] standardu H.264/AVC posiada szereg zalet w porównaniu do architektur opisanych w literaturze. Po pierwsze, uzyskiwane efektywności kompresji są większe o 0.25-0.6 dB (faktycznie nawet o 1.5dB z uwagi na pominięcie sprawdzania trybu SKIP w niektórych architekturach). Po drugie, zużycie zasobów logicznych jest najmniejsze, co umożliwia efektywne implementacje w układach FPGA. Po trzecie, obsługiwany jest profil wysoki standardu, kodowanie CAVLC i CABAC, kodowanie ramkowe (progresywne) i polowe (z przeplotem), formaty 4:2:2 i 4:2:0 oraz kontrola stopnia kompresji. Po czwarte, przepustowość koder należy do największych i umożliwia obsługę rozdzielczości 1080p@60fps i 1080p@30fps odpowiednio dla technologii ASIC 90nm i FPGA Arria II (układy o średnim koszcie). Opracowany koder sprzętowy został zweryfikowany w warunkach czasu rzeczywistego w układzie FPGA.

Podsumowując, opisane osiągnięcia naukowe dotyczą rozwoju nowych algorytmów i architektur sprzętowych koderów wizyjnych. Do najważniejszych elementów wkładu przedłożonego jako osiągnięcie naukowe należy zaliczyć :

- Zintegrowaną architekturę koder entropijnego obsługującego tryby CAVLC i CABAC, ze współdzieleniem zasobów i wielo-symbolowym kodowaniem arytmetycznym.
- Adaptacyjny i skalowalny algorytm estymacji ruchu wraz odpowiadającą mu architekturą sprzętową unifikującą estymację na poziomie całkowito-pikselowym i pod-pikselowym.
- Wydajne architektury predykcji wewnątrzramkowej INTRA wraz schematami efektywnego harmonogramowania obliczeń.
- Metodologię dekompozycji transformacji DCT/IDCT umożliwiającą implementacje konfigurowanych architektur i zredukowaną ilość zasobów sprzętowych.
- Wydajną architekturę koder H.264/AVC z wyborem trybu kodowania na podstawie analizy RD.

5. Omówienie pozostałych osiągnięć naukowo badawczych:

Oprócz prac mieszczących się w zakresie głównego osiągnięcia naukowego zajmowałem się również innymi zagadnieniami naukowymi w zakresie technik multimedialnych. Zaliczać do nich można prace dotyczące:

- Sprzętowego kodera JPEG 2000,
- Sprzętowego dekodera H.264/AVC,
- Systemu komunikacji Audio-Video-Dane (AVD),
- Algorytmów kontroli stopnia kompresji koderów wizyjnych.

Prace dotyczące sprzętowego kodera JPEG 2000 były kontynuacją tych zaczętych w trakcie doktoratu. Po doktoracie były one prowadzone we współpracy z firmą Evatronix. W ich efekcie zrealizowano kompletny koder umożliwiający kompresję obrazów/wideo w czasie rzeczywistym przy rozdzielczości do 1080p@30fps. Koder został zweryfikowany w technologii FPGA i jest dostępny jako komponent wirtualny.

Realizacja dekodera standardu H.264/AVC prowadzona była jako praca zespołowa. Oprócz mnie zespół obejmował moich dyplomantów, którzy następnie zostali doktorantami ściśle ze mną współpracującymi. Opracowany dekodek został wyspecyfikowany od podstaw w języku VHDL i zweryfikowany w technologii FPGA w warunkach czasu rzeczywistego. Umożliwia on dekodowanie przy rozdzielczości 720p@30fps. Dekoder posłużył w projekcie PROTEUS (POIG) jako element składowy systemu komunikacji AVD.

System AVD był częścią składową systemu PROTEUS przeznaczonego do wspomaganiania działań antyterrorystycznych i antykrzysowych. Głównym zadaniem systemu AVD była kompresja i multipleksacja danych przesyłanych pomiędzy robotami i konsolami sterującymi operatorów. Oprócz kodera i dekodera wideo H.264/AVC system AVD obejmował koder i dekodek audio AAC, multiplekser i demultiplekser strumienia transportowego MPEG-2. Przy moim dużym udziale opracowane zostały moduły elektroniczne integrujące układ FPGA, mikrokontroler ARM, pamięci oraz przetworniki AC/CA. W układzie FPGA integrowane były moduły odbiorcze (dekodery i demultiplekser) lub nadawcze (kodery i multiplekser). Całość została uruchomiana i zintegrowana z urządzeniami radiowymi zapewniającymi wzajemną komunikację. Łączne opóźnienie w torze AVD jest niewielkie i wynosi 50-60 ms.

Kodery wideo pracujące w rzeczywistych systemach komunikacyjnych wymagają kontroli stopnia kompresji celem dopasowania przepływności skompresowanego strumienia wideo do pasma kanału transmisyjnego. W tym celu opracowałem algorytm kontroli stopnia kompresji przeznaczony do transmisji z małym opóźnieniem. Algorytm został zaimplementowany w koderze sprzętowym i zweryfikowany w całym systemie AVD. W ramach wcześniejszego projektu europejskiego VISNET opracowany został również algorytm kontroli stopnia kompresji stosowany w równoległym kodowaniu kilku sekwencji wizyjnych. Algorytm umożliwia dopasowanie łącznego stopnia kompresji przetwarzanych sekwencji do zadanej przepływności kanału transmisyjnego. Równocześnie rozdział budżetu bitowego pozwala na zrównoważenie jakości w poszczególnych sekwencjach.

Gregor Pastuszek