Jarosław Samelak
Cezary Korzeniewski
Łukasz Kowalski
Jakub Stankowski
Katedra Telekomunikacji Multimedialnej i Mikroelektroniki,
Politechnika Poznańska, ul. Polanka 3, 60-965 Poznań,
{jsamelak, ckorzeniewski, lkowalski, jstankowski}@multimedia.edu.pl

KKRRiT 2014
Warszawa, 11-13 czerwca 2014

ZOPTYMALIZOWANA IMPLEMENTACJA DEKODERA HEVC

Streszczenie: W niniejszym artykule zostały przedstawione metody optymalizacji niektórych elementów dekodera HEVC, opracowane przez Katedrę Telekomunikacji Multimedialnej i Mikroelektroniki Politechniki Poznańskiej. Zaprezentowane algorytmy przyczyniły się do blisko dwukrotnej redukcji czasu dekodowania w stosunku do oprogramowania modelowego.

1. WSTĘP

W kwietniu 2013 roku została przyjęta nowa norma kodowania sekwencji wizyjnych określana mianem HEVC (*High Efficiency Video Coding*). Jest to czwarta generacja technik kompresji, pozwalająca na dwukrotną redukcję prędkości bitowej w stosunku do popularnego standardu AVC (*Advanced Video Coding*) przy zachowaniu podobnej jakości przesyłanego materiału [1]. Zwiększenie współczynnika kompresji uzyskuje się kosztem wielokrotnego wzrostu złożoności obliczeniowej kodeka.

W ramach prac nad nowym standardem, grupa JCT-VC opracowała oprogramowanie modelowe kodeka HEVC [2]. Dekoder zawarty w tym oprogramowaniu cechuje się niską wydajnością i daleki jest od możliwości dekodowania strumienia w czasie rzeczywistym. Autorzy niniejszego artykułu podjęli się analizy i optymalizacji algorytmów zaimplementowanych w oprogramowaniu modelowym, jak również opracowali nowe, bardziej wydajne rozwiązania. Celem tych działań było osiągnięcie wyraźnej redukcji czasu dekodowania strumienia wizyjnego. Optymalizacji zostały poddane krytyczne bloki funkcjonalne dekodera. W artykule przedstawiono kluczowe modyfikacje dotyczące predykcji wewnątrzobrazowej, międzyobrazowej, wyznaczania odwrotnej transformaty kosinusowej i skalowania. Przyspieszenie procesu dekodowania uzyskano w głównej mierze dzięki optymalizacji stosowanych algorytmów oraz wykorzystaniu możliwości współczesnych procesorów, takich jak zrównoleglenie wykonywanych obliczeń. Wykorzystano w tym celu instrukcje SSE (*Streaming SIMD Extentions*) umożliwiające równoległe przetwarzanie wektora danych.

Opracowane techniki optymalizacji znalazły zastosowanie w implementacji dekodera HEVC zrealizowanej przez Katedrę Telekomunikacji Multimedialnej i Mikroelektroniki Politechniki Poznańskiej.

2. OPTYMALIZACJA PREDYKCJI

2.1. Predykcja wewnątrzobrazowa

Kodek HEVC charakteryzuje się dużą liczbą możliwych trybów predykcji wewnątrzobrazowej (*intraframe prediction*). Ich liczba została kilkukrotnie zwiększona w stosunku do kodeka AVC (9 trybów dla bloków 4×4, 8×8 oraz 4 tryby dla bloków 16×16) – do 35 trybów [3]. Większość z nich to tryby kierunkowe (33), wśród których wyróżnia się tryb poziomy (*Horizontal*) oraz pionowy (*Vertical*). Pozostałe to tryb uśredniający (*DC*) i planarny (*Planar*). Predykcja wewnątrzobrazowa wykonywana jest w kwadratowym bloku obrazu o rozmiarze 4×4, 8×8, 16×16, 32×32 lub 64×64 okresy próbkowania. Dobór rozmiaru bloku jest zależny od treści przewidywanego fragmentu obrazu. Zastosowanie mniejszego bloku pozwala zazwyczaj na uzyskanie mniejszego błędu predykcji. Kosztem wyboru bloku o małej liczbie próbek jest zwiększenie liczby bitów przesyłanej informacji dodatkowej. W celu zmniejszenia błędu predykcji podczas zastosowania trybu poziomego lub pionowego wykonywana jest filtracja krawędzi bloku.

W ramach prac nad implementacją kodeka HEVC dokonano optymalizacji procesu predykcji wewnątrzobrazowej w dwóch krokach. W pierwszym kroku wykonano optymalizację implementacji funkcji wykonujących predykcję wewnątrzobrazową w modelu testowym kodera, natomiast w drugim zastosowano technikę zrównoleglenia obliczeń.

W pierwszej kolejności optymalizacji poddano tryby kierunkowe. Dokonano redukcji liczby wykonywanych operacji mnożenia poprzez przekształcenie wzoru na wyznaczanie średniej ważonej z dwóch próbek odniesienia:

$$P = (a \cdot R + b \cdot R1 + 16) >> 5, \quad (1)$$

$$P = R + (b(R1 - R) + 16) >> 5, \quad (2)$$

gdzie:
P – wartość przewidywanej próbki,
a ,b – wagi próbek odniesienia,
R ,$R1$ – wartości próbek odniesienia,
>> – przesunięcie bitowe w prawo.

W powyższym przekształceniu wykorzystano fakt, iż suma wag próbek odniesienia jest stała i wynosi 32. Jak można łatwo zauważyć, we wzorze (1) występują dwa mnożenia i dwa dodawania, natomiast we wzorze przekształconym (2) tylko jedno mnożenie, dwa dodawania oraz jedno odejmowanie. Ponadto w omawianym kroku optymalizacji zauważono, że korzystnym jest stworzenie oddzielnych fragmentów oprogramowania, które realizują predykcję dla pewnych grup trybów kierunkowych. Tryby należące do jednej grupy łączą pewne cechy, takie jak zbiór próbek odniesienia. Podział ten pozwolił wye-

liminować zbędne operacje dla danej grupy trybów, które były wykonywane w oprogramowaniu modelowym. Efektywniejsza implementacja pozwoliła zredukować liczbę wykonywanych działań arytmetycznych. Na tym etapie optymalizacji zrealizowano także kopiowanie całego wiersza danych, które jest szybsze niż kopiowanie próbka po próbce.

Przedstawione rozwiązania pozwoliły na uzyskanie względem modelowego kodeka zysku wykonywania predykcji wewnątrzobrazowej rzędu kilkunastu procent.

W drugim kroku optymalizacji poddano wszystkie 35 trybów predykcji wewnątrzobrazowej. Zastosowano instrukcje SSE. Wykorzystują one 128-bitowe rejestry XMM pozwalające na zrównoleglenie obliczeń. W przypadku zastosowania reprezentacji 16 bitów na próbkę obrazu, rejestry XMM umożliwiają wykonanie danej operacji na 8 próbkach obrazu jednocześnie. Aby dokonać zrównoleglenia obliczeń wynik z kroku bieżącego musi być niezależny od wyniku z poprzedniego kroku algorytmu. Taka sytuacja często występuje w funkcjach realizujących predykcję wewnątrzobrazową, co skrupulatnie wykorzystano w czasie prac nad optymalizacją dekodera HEVC.

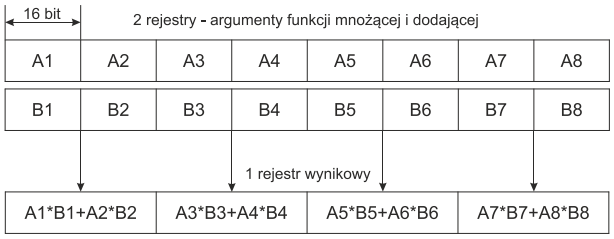
Zastosowanie zrównoleglenia obliczeń przyczyniło się do znacznej redukcji czasu potrzebnego do przeprowadzenia procesu predykcji wewnątrzobrazowej. Osiągnięta redukcja czasu jest zależna od rozmiaru bloku oraz trybu predykcji. Szczegółowy opis procesu optymalizacji predykcji wewnątrzobrazowej wraz z fragmentami oprogramowania oraz uzyskane wyniki można znaleźć w pracy [4].

2.2. Predykcja międzyobrazowa

Zdecydowana większość obrazów sekwencji wizyjnych jest kodowana z wykorzystaniem predykcji międzyobrazowej z uwagi na możliwość uzyskania znacznie wyższego współczynnika kompresji, niż przy zastosowaniu kodowania wewnątrzobrazowego [5]. Na predykcję z kompensacją ruchu przypada około połowa czasu dekodowania [6]. Dekoder realizuje predykcję międzyobrazową na podstawie otrzymanych w strumieniu informacji o ruchu oraz identyfikatorów obrazów odniesienia. Standard HEVC przewiduje wykonywanie predykcji z kompensacją ruchu z dokładnością do ¼ okresu próbkowania [3]. Po stronie dekodera może się zatem okazać konieczne wykonanie interpolacji fragmentu obrazu odniesienia w celu osiągnięcia podpunktowej dokładności predykcji. Interpolacja jest dwuwymiarowym splotem obrazu odniesienia z maską odpowiedniego filtru. Ten element dekodera HEVC jest znacznie bardziej złożony obliczeniowo i wymaga większej liczby operacji dostępu do pamięci niż w technice AVC [7].

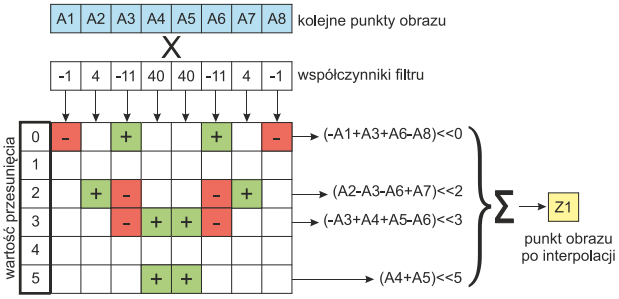
W opracowanym dekoderze zaproponowano trzy metody przyspieszenia obliczeń związanych z interpolacją obrazów odniesienia [8]. Dwie z nich wykorzystują rejestry 128-bitowe oraz instrukcje zrównoleglające obliczenia, operujące na tych rejestrach. Trzeci algorytm zaimplementowano dla procesorów nie wyposażonych w zestaw instrukcji SSE.

Pierwsza z opracowanych metod polega na zrównolegleniu funkcji dwuwymiarowego splotu. Wyznaczenie każdego punktu interpolowanego obrazu lumy wymaga obliczenia sumy ośmiu iloczynów. Zgodnie z normą HEVC interpolacja powinna być wykonywana na liczbach o 14-bitowej reprezentacji, w związku z czym wewnątrz 128-bitowego rejestru może zostać umieszczonych do ośmiu punktów obrazu, a w kolejnym odpowiadające im współczynniki filtru. Następnie wszystkie operacje mnożenia można wykonać za pomocą jednej instrukcji, co znacznie przyspiesza obliczenia. Ponieważ iloczyn dwóch liczb 16-bitowych jest liczbą 32-bitową, przechowanie wyniku wspomnianej instrukcji wymagałoby dwóch rejestrów. W stworzonym oprogramowaniu wykorzystano jednak specjalną instrukcję (rys. 1), która poza mnożeniem odpowiadających sobie 16-bitowych fragmentów rejestrów dodaje parami kolejne iloczyny. Dzięki takiemu rozwiązaniu zmniejsza się liczba używanych rejestrów oraz skraca się czas obliczeń. Po wykonaniu tej instrukcji należy dwukrotnie dodać 32-bitowe fragmenty wynikowego rejestru w celu otrzymania jednego punktu interpolowanego obrazu.



Rys. 1. Schemat działania instrukcji mnożącej i dodającej

Drugim ze sposobów na przyspieszenie wykonywania obliczeń funkcji interpolacji jest zastosowanie metody pozbawionej multiplikacji. Polega ona na zastąpieniu operacji mnożenia przez stałe liczby kombinacjami dodawania i przesunięć bitowych, które procesor wykonuje w krótszym czasie. Bazując na tym spostrzeżeniu przeprowadzono analizę filtrów interpolacyjnych zgodnie z algorytmem pokazanym na rysunku 2.



Rys. 2. Schemat algorytmu metody bez mnożeń - przykład dla filtru połówkowego luminancji.

W powyższym przykładzie każdy ze współczynników przedstawiono jako sumę liczb będących potęgami dwójki (np. -11 można zapisać jako 1-4-8), a następnie otrzymane składniki zastąpiono przesunięciami bitowymi w lewo (np. mnożeniu przez 8 odpowiada przesunięcie bitowe w lewo o 3). Uzyskanie pojedynczego punktu obrazu interpolowanego wymaga zsumowania ośmiu kolejnych punktów obrazu wejściowego, przesuniętych o odpowiednie wartości.

Analizę każdego z filtrów interpolacyjnych przeprowadzono w taki sposób, aby zredukować do minimum liczbę wykonywanych operacji. Interpolacja filtrem z rysunku 2 wymagałaby 8 mnożeń i 7 dodawań, jednak stosując metodę bez multiplikacji ten sam rezultat uzyskuje się za pomocą 3 przesunięć bitowych i 13 dodawań. W przeciwieństwie do mnożenia są to operacje, które procesor wykonuje w zaledwie jednym cyklu maszynowym, w związku z czym opisywana metoda efektywnie przyspiesza wykonywanie obliczeń związanych z predykcją międzyobrazową.

Trzeci algorytm optymalizacji interpolacji zaprojektowano w celu połączenia zalet poprzednich metod. Opracowane rozwiązanie pozwala z jednej strony na eliminację kosztownych obliczeniowo operacji mnożenia, a z drugiej umożliwia jednoczesną interpolację kilku punktów obrazu. Punktem wyjścia do implementacji tej techniki była metoda bez multiplikacji, w której operacje przesunięć bitowych, dodawania i odejmowania zastąpiono odpowiadającymi im instrukcjami SSE. Wszystkie użyte instrukcje wykonywane są przez procesor w jednym cyklu maszynowym i realizują równoczesną interpolację czterech punktów, co znacznie zwiększa wydajność algorytmu w porównaniu z przedstawioną wcześniej metodą bez multiplikacji.

Predykcja międzyobrazowa w technice HEVC bardzo często wykonywana jest na podstawie dwóch obrazów odniesienia – wówczas jednostka predykcji powstaje poprzez uśrednienie dwóch fragmentów tych obrazów. Przeprowadzona analiza dekodera pokazała, że funkcja uśredniająca jest drugim (po interpolacji) najbardziej czasochłonnym elementem predykcji międzyobrazowej. W opracowanym oprogramowaniu dokonano optymalizacji tej części dekodera, wykorzystując instrukcje zrównoleglające obliczenia.

Standardowa realizacja funkcji uśredniającej polega na obliczeniu sumy dwóch fragmentów obrazów odniesienia i podzieleniu wyniku przez 2. W tym algorytmie istnieje jednak ryzyko przekroczenia zakresu 16-bitowej zmiennej po dodaniu dwóch liczb i w efekcie błędnej interpretacji wyniku. Generuje to konieczność zastosowania reprezentacji 32-bitowej, przez co po wykonaniu zrównoleglenia obliczeń jednocześnie możliwe jest wyznaczanie jedynie 4 punktów wynikowych.

Powyższe ograniczenie zostało wyeliminowane poprzez przesunięcie zakresu we fragmentach obrazów odniesienia w taki sposób, aby po ich zsumowaniu nie przekroczył on reprezentacji 16-bitowej. Podzielenie wyniku przez 2 umożliwia przywrócenie pierwotnego zakresu bez obaw o błędną interpretację wyniku. Opisane rozwiązanie wprowadza kilka nadmiarowych operacji, jednak umożliwia równoczesne obliczanie aż 8 punktów obrazu. Zastosowanie przesuwania zakresu i zrównoleglenia obliczeń pozwoliło na niemal siedmiokrotne przyspieszenie funkcji uśredniającej fragmenty obrazu odniesienia, natomiast wydajność całego dekodera wzrosła o ok. 4,5%.

3. OPTYMALIZACJA ODWROTNEJ TRANSFORMACJI KOSINUSOWEJ I SKALOWANIA

Dyskretna transformacja kosinusowa (DCT – *discrete cosine transform*) jest przekształceniem szeroko stosowanym w kompresji sekwencji wizyjnych i obrazów. W standardzie HEVC zastosowano całkowitoliczbową transformację będącą aproksymacją DCT. W dekoderze otrzymuje się wartości próbek obrazu ze współczynników transformaty dokonując odwrotnej transformacji kosinusowej (IDCT – *inverse discrete cosine transform*).

W ramach prac nad dekoderem HEVC dokonano optymalizacji procesu odwrotnej transformacji kosinusowej. Właściwości IDCT zgodne ze standardem HEVC pozwalają na częściową faktoryzację transformacji dzięki wielu symetriom i antysymetriom występującym w macierzy transformacji. Skutkuje to znaczącą redukcją liczby wykonywanych operacji mnożenia podczas wyznaczania wartości próbek obrazu ze współczynników transformaty. Dokonano i przetestowano również kilka różnych implementacji IDCT z wykorzystaniem instrukcji SSE, z których każda w różnym stopniu wykorzystywała możliwość faktoryzacji transformacji. Najbardziej efektywna z nich przyniosła zysk czasu wyznaczania odwrotnej transformacji ponad 50%, co jest bardzo satysfakcjonującym wynikiem.

W celu redukcji nieistotnych dla odbiorcy informacji podczas kodowania sekwencji wizyjnych współczynniki transformaty poddawane są kwantowaniu [5]. Wartość każdego współczynnika przybliżana jest do jednego z określonych poziomów kwantyzacji. W trakcie dekodowania wykorzystywany jest proces odwrotny do kwantyzacji zwany skalowaniem. Ponieważ dane przetwarzane podczas skalowania są od siebie niezależne, proces ten można zrównoleglić. Implementacja z wykorzystaniem instrukcji SSE pozwala w jednej iteracji uzyskać aż 8 współczynników. Efektem optymalizacji jest blisko 5-krotnie przyspieszenie obliczeń związanych ze skalowaniem.

4. OCENA OPROGRAMOWANIA

Ocena stworzonego oprogramowania polegała na pomiarze czasu potrzebnego do zdekodowania 7 strumieni wizyjnych w rozdzielczości 1920×1080, zakodowanych z czterema różnymi parametrami kwantowania. Odpowiednie fragmenty przedstawionych modyfikacji zostały poddane indywidualnej ocenie. Otrzymane wyniki zestawiono z czasami dekodowania sekwencji przy użyciu oprogramowania modelowego i na tej podstawie wyznaczono średnią redukcję czasu przetwarzania danych.

Tabela 1 przedstawia redukcję czasu wykonywania trybu wewnątrzobrazowego w stosunku do fragmentów oprogramowania modelowego, uzyskaną dzięki opisanym modyfikacjom. Zaprezentowane wyniki zostały przedstawione z podziałem na dostępne rozmiary jednostek predykcji oraz kolejne kroki opisanych ulepszeń. W tabeli 2 zestawiono wyniki przyspieszenia predykcji międzyobrazowej osiągniętego przy użyciu różnych metod wyznaczania obrazów interpolowanych, które w głównej mierze zależy od kierunku interpolacji. Tabela 3 prezentuje redukcję czasu wykonywanych obliczeń

uzyskaną poprzez zrównoleglenie obliczeń wykonywanych w procesach skalowania współczynników transformaty oraz odwrotnej transformacji kosinusowej w zależności od rozmiaru bloku próbek transformaty.

Tab. 1. Uzyskana redukcja czasu wykonywania predykcji wewnątrzobrazowej

Rozmiar bloku	Optymalizacja algorytmiczna	Optymalizacja algorytmiczna + zrównoleglenie obliczeń
4x4	15,1%	54,6%
8x8	9,5%	42,4%
16x16	13,6%	48,0%
32x32	13,6%	46,0%
64x64	9,7%	54,6%
Średnia	**12,3%**	**49,1%**

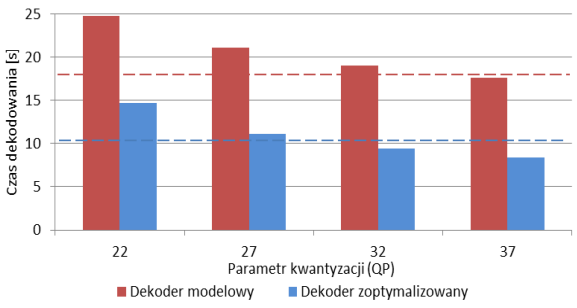
Tab. 2. Uzyskana redukcja czasu wykonywania predykcji międzyobrazowej

Kierunek interpolacji	Metoda interpolacji		
	Bez multiplikacji	Zrównoleglenie obliczeń	Zrównol. obliczeń metody bez multiplikacji
Poziomy	32,6%	59,6%	44,7%
Pionowy	36,9%	40,6%	41,0%
Mieszany	36,3%	42,9%	41,4%

Tab. 3. Uzyskana redukcja czasu wykonywania odwrotnej transformacji kosinusowej oraz skalowania

Rozmiar transformaty	Zrównoleglenie obliczeń	
	IDCT	Skalowanie
4x4	71,4%	56,1%
8x8	37,7%	81,0%
16x16	46,4%	83,9%
32x32	45,9%	87,3%
Średnia	**50,4%**	**77,1%**

Opisane w tym artykule ulepszenia w dużej mierze przyczyniły się do zwiększenia wydajności opracowanej implementacji dekodera. Przeprowadzone badanie czasu dekodowania sekwencji wykazało jego średnie blisko dwukrotne przyspieszenie (**47,7%**) względem oprogramowania modelowego. Im wyższy jest parametr kwantyzacji dekodowanej sekwencji, tym uzyskana redukcja czasu dekodowania jest większa (rys. 3).



Rys. 3. Średni czas dekodowania sekwencji Full HD z różnym stopniem kwantowania

5. PODSUMOWANIE

Jak pokazały przeprowadzone badania, stworzone oprogramowanie pozwala na dekodowanie w czasie rzeczywistym sekwencji 1080p (Full HD), zakodowanych z wysokimi parametrami kwantyzacji, jakie często przyjmuje się w systemach telewizji cyfrowej (powyżej 30). Zastosowanie opisanych w tym artykule ulepszeń pozwoliło na znaczne, blisko dwukrotne przyspieszenie procesu dekodowania w stosunku do oprogramowania referencyjnego. Co istotne, przedstawione propozycje wpływają jedynie na zmniejszenie czasu dekodowania, a nie oddziałują na wyniki otrzymywane w tym procesie.

Przeprowadzone prace nie zamykają możliwości dalszych usprawnień mających na celu szybsze wykonywanie procesu dekodowania. Autorzy zauważają jeszcze kilka możliwości usprawnienia stworzonego oprogramowania. Przede wszystkim zastosowanie wielowątkowości pozwoliłoby na zrównoleglenie wykonywania poszczególnych podprocesów. Jako alternatywę dla instrukcji SSE można by zastosować instrukcje AVX (*Advanced Vector Extensions*), które wykonują operacje na rejestrach 256-bitowych. Wykorzystując tego typu zestaw rozkazów, procesor byłby w stanie operować jednocześnie na 16 punktach obrazu zamiast 8, jak obecnie. Celem, który należałoby osiągnąć podczas dalszych prac byłoby dekodowanie sekwencji Ultra HD (4K, 8K) w czasie rzeczywistym.

Sfinansowano ze środków publicznych na naukę w ramach projektu badań statutowych.

SPIS LITERATURY

[1] Domański M. „HEVC – nowa generacja technik kompresji obrazu” Przegląd Telekomunikacyjny, Nr 4, 2012.

[2] Joint Collaborative Team on Video Coding (JCT-VC), HM 10: High Efficiency Video Coding (HEVC) Test Model 10 Encoder Description, JCTVC-L1002, Geneva, Jan. 2013.

[3] International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), *Recommendation H.265: High Efficiency Video Coding*, Apr. 2013.

[4] Korzeniewski C.: *Szybka implementacja trybu wewnątrzobrazowego w koderze HEVC*, Praca dyplomowa inżynierska, Wydział Elektroniki i Telekomunikacji Politechniki Poznańskiej, Poznań 2013.

[5] Domański M., *Obraz cyfrowy*, WKiŁ, Warszawa 2010.

[6] Bossen F., Bross B., Sühring K., Flynn D. „HEVC Complexity and Implementation Analysis” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, Dec. 2012, s. 1685-1696.

[7] Hao Lv, Ronggang Wang, Xiaodong Xie, Huizhu Jia, Wen Gao „A comparison of fractional-pel interpolation filters in HEVC and H.264/AVC” 2012 IEEE Visual Communications and Image Processing (VCIP), San Diego, 27-30 Nov. 2012.

[8] Samelak J.: *Optymalizacja predykcji z kompensacją ruchu w kodeku HEVC*, Praca dyplomowa inżynierska, Wydział Elektroniki i Telekomunikacji Politechniki Poznańskiej, Poznań 2014.