Poznań, 07.08.2020

# Raport – Praktyki studenckie 2020

**Imię i nazwisko praktykanta:** Patryk Śpitalniak  
**Termin odbycia praktyki:** 1-30.07.2020  
**Nazwa przedsiębiorstwa:** Instytut Telekomunikacji Multimedialnej Politechniki Poznańskiej  
**Opiekun praktyk:** dr inż. Tomasz Grajek  
**Temat praktyk:** Koder sekwencji wizyjnej JPEG 3D

1. **Wstęp teoretyczny:**

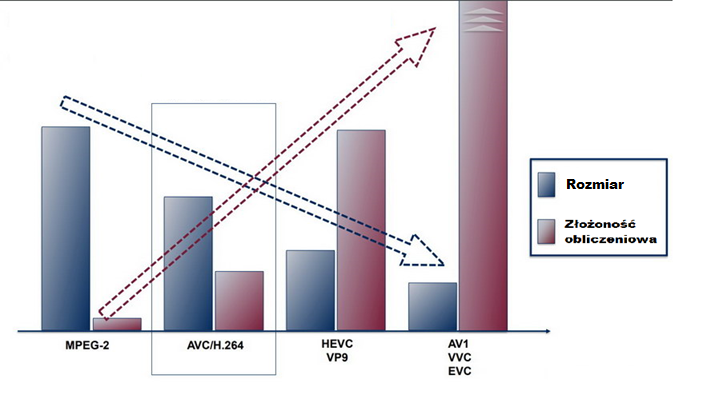
JPEG (ang. Joint Photographic Experts Group) to standard kompresji grafiki rastrowej, który został opublikowany w 1991 roku. Do dnia dzisiejszego jest to najpopularniejsze i najpowszechniejsze narzędzie stosowane do kompresji obrazów (w szczególności dla naturalnych zdjęć). W pierwotnej wersji koder ten dokonuje podziału obrazu na bloki o rozmiarach 8x8 próbek, a następnie na każdym bloku przeprowadza operacje matematyczne tj. DCT (dyskretna transformata kosinusowa), kwantowanie współczynników DCT, specjalne uszeregowanie współczynników tzw. zig-zag, kodowanie współczynników, kodowanie entropijne.

W późniejszym okresie stworzono format kompresji M-JPEG (ang. Motion – JPEG), który posłużył do kodowania sekwencji wizyjnych, gdzie każda klatka kodowanego materiału była kompresowana samodzielnie (wewnątrzobrazowo), przy użyciu zwykłego JPEG. Ze względu na niską złożoność obliczeniową, rozwiązanie to znalazło zastosowanie m.in. w urządzeniach o słabej mocy obliczeniowej lub tam, gdzie wydajna kompresja nie była potrzebna.

1. **Opis problemu:**

Jak wspomniano wyżej, w formacie M-JPEG, każda klatka sekwencji wizyjnej jest kodowana jako osobny, niezależny od siebie obraz. Nie trudno się domyślić, że przekłada się to na względnie duże rozmiary plików po kompresji. Jeśli jedna klatka filmu Full HD (1920x1080) ma rozmiar 162kB, a jedna sekunda zawiera takich klatek trzydzieści, to na przechowanie godziny takiego filmu potrzebujemy prawie 17.5GB pamięci, dlatego przechowywanie dłuższych nagrań w takim formacie jest problematyczne i wymaga posiadania dużej ilości pamięci zewnętrznej. Można zatem stwierdzić, że M-JPEG w swojej oryginalnej wersji jest stosunkowo mało wydajnym standardem kompresji sekwencji wizyjnych.

Oczywiście, dysponujemy obecnie znacznie bardziej wydajnymi standardami kompresji np. AVC (ang. Advanced Video Coding) lub HEVC (ang. High Efficiency Video Coding), jednakże wraz z siłą kompresji danego kodera, wzrasta również jego złożoność obliczeniowa (czasowa jak i pamięciowa), co oznacza, że potrzebujemy urządzeń o większej mocy obliczeniowej.



Porównanie rozmiaru (wydajności kompresji) i złożoności obliczeniowej koderów sekwencji wizyjnej.

Podsumowując, mamy do czynienia ze stosunkowo słabą kompresją, a zastosowanie lepszych i wydajniejszych koderów sekwencji wizyjnej często nie jest możliwe. W szczególności mówimy tutaj o urządzeniach takich jak kamery monitoringu i wideo rejestratory jazdy, ponieważ bardzo często używają one M-JPEG i posiadają procesory o słabej mocy obliczeniowej.

1. **Zaproponowane rozwiązanie problemu:**

Szukając złotego środka pomiędzy wydajniejszą kompresją, a jednocześnie niewiele większą złożonością obliczeniową, pojawia się pomysł modyfikacji oryginalnego M-JPEG, w taki sposób, aby bazując na rozwiązaniach stosowanych w zwykłym JPEG, zredukować nadmiarowość informacji.  
W tym celu powstały trzy propozycje modyfikacji:

* 1. JPEG 3D 8x8x8 DCT – Dodanie czasu, jako trzeciego wymiaru do bloków współczynników 8x8, otrzymując w ten sposób bloki 8x8x8, a następnie przeprowadzając DCT na każdych ośmiu współczynnikach następujących po sobie obrazów.
  2. JPEG 3D z predykcją – Porównanie wartości współczynników dla każdych następujących po sobie w czasie ośmiu bloków 8x8. W przypadku bardzo niewielkiej różnicy, blok następnego obrazu nie jest kodowany, a jedynie przesyłana jest informacja o tym, żeby dekoder skopiował go z obrazu poprzedniego.
  3. JPEG 16x16 DCT – Konwersja współczynników DCT z bloków 8x8, do bloków DCT 16x16.

Wyżej wymienione modyfikacje powinny zapewnić lepszą kompresję od zwykłego M-JPEG, ponieważ zredukują nadmiarowość informacji w czasie, jaka pojawia się w sekwencji wizyjnej, a także nie powinny wnosić dużej złożoności obliczeniowej, ze względu na wykorzystanie w większości algorytmiki oryginalnego standardu JPEG.

1. **Środowisko pracy:**

Do implementacji zmodyfikowanego kodera wykorzystano następujące narzędzia:

* Implementację kodera JPEG, „TooJPEG”: <https://github.com/stbrumme/toojpeg>
* Implementację dekodera JPEG, „picoJPEG”: <https://github.com/richgel999/picojpeg>
* Język programowania C++11
* Środowisko programistyczne CLion

1. **JPEG 3D 8x8x8 DCT:**

Na wejście programu podajemy sekwencję klatek kodowanego wideo w formacie JPEG. Każde osiem następujących po sobie klatek zostaje zdekodowanych do miejsca, w którym uzyskujemy nieprzeskalowane wartości współczynników DCT, następnie dla każdych ośmiu współczynników odpowiadających sobie położeniem, wyliczona zostaje jednowymiarowa transformata DCT (tzw. DCT w czasie). Uzyskane w ten sposób współczynniki, podobnie jak w zwykłym dwuwymiarowym JPEG zostają uszeregowane w specjalnej kolejność tzw. zig-zag, jednakże tym razem są one uszeregowane w blokach 8x8x8, gdzie pierwszy i drugi wymiar to odpowiednio długość i szerokość, natomiast trzeci wymiar to czas (następujące po sobie obrazy). Dalsze kroki to oczywiście odpowiednie kodowanie współczynników i kodowanie entropijne, które odbywają się w blokach 8x8x8, a efektem końcowym kodowania jest otrzymanie pliku JPEG dla każdych ośmiu klatek kodowanej sekwencji.

1. **JPEG 3D z predykcją:**

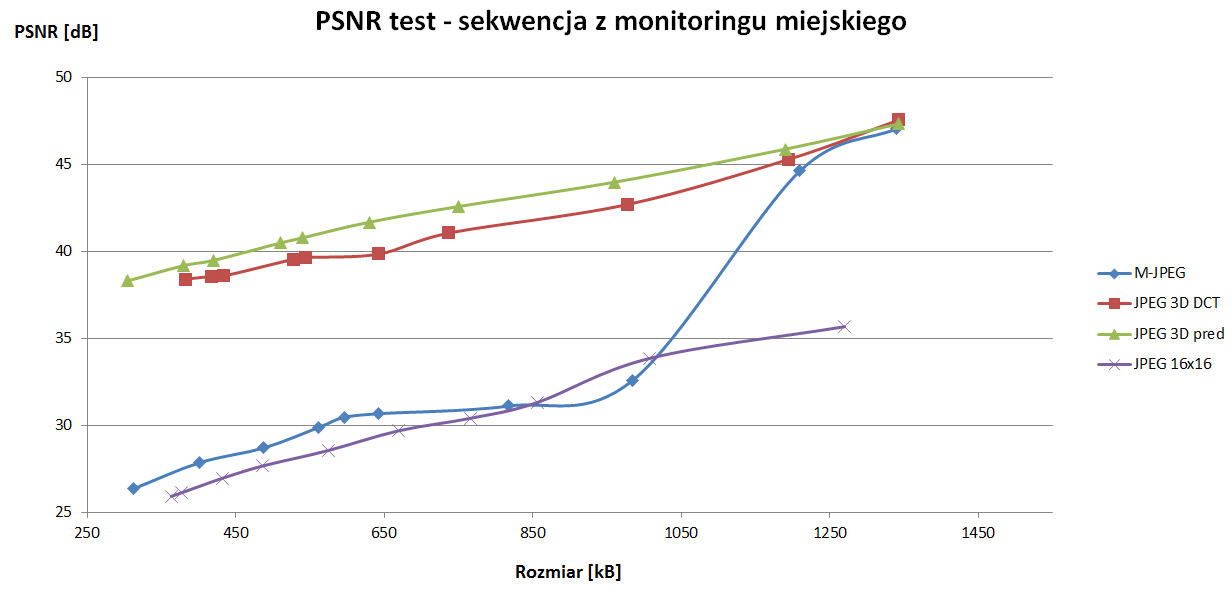
Podobnie jak w poprzedniej wersji, osiem klatek zostaje zdekodowanych, tym razem do miejsca w którym uzyskujemy przeskalowane wartości współczynników DCT, następnie każdy blok 8x8 obrazu (z pominięciem obrazu pierwszego, który jest pierwszym obrazem odniesienia) jest porównywany z odpowiadającym mu blokiem 8x8 obrazu poprzedzającego. Jeśli bloki te są podobne (niewielka zmiana wartości współczynników), zostaje zakodowana informacja, aby blok ten przepisać (skopiować) z obrazu poprzedzającego. W dalszym etapie kodowania współczynniki zostają skwantowane, ponownie uszeregowane w blokach 8x8x8 i zakodowane, a efektem końcowym jest otrzymanie pliku JPEG dla każdych ośmiu klatek kodowanej sekwencji.

1. **JPEG 16x16 DCT:**

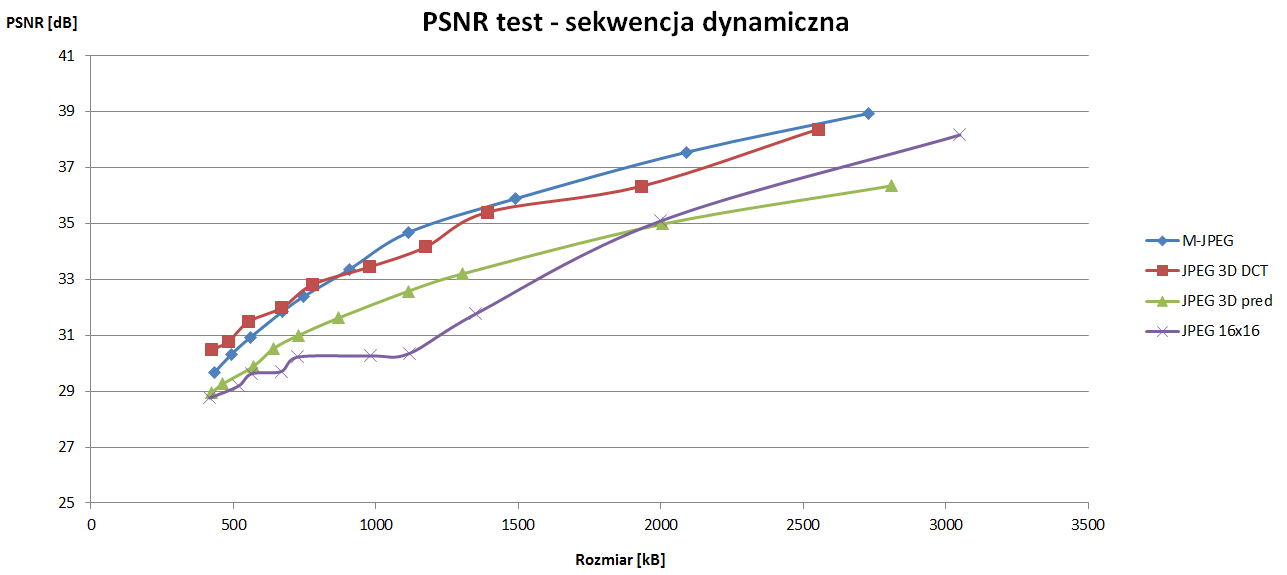
Każda klatka kodowanej sekwencji zostaje zdekodowana do miejsca, w którym uzyskujemy przeskalowane wartości współczynników DCT, następnie cztery kolejne (w kolejności kodowania) bloki 8x8 luminacji zostają odpowiednio zmodyfikowane i połączone w bloki 16x16. Modyfikacja ta ma na celu przejście współczynników z dwuwymiarowej transformaty DCT o rozmiarze 8x8, do dwuwymiarowej transformaty DCT o rozmiarze 16x16, bez konieczności wykonywania wcześniejszego IDCT na blokach 8x8. W dalszym etapie kodowania, współczynniki zostają skwantowane, uszeregowane w blokach 16x16 i zakodowane, a efektem końcowym jest otrzymanie pliku JPEG dla każdej kodowanej klatki.  
Podczas dekodowania, bloki 16x16 nie są już zamieniane na bloki 8x8, lecz wykonywana zostaje na nich odwrotna transformata dwuwymiarowa IDCT 16x16.

1. **Analiza i interpretacja wyników:**

Poniższe wykresy przedstawiają porównanie rozmiaru i jakości (PSNR) dla 8 klatek sekwencji wizyjnej, kompresowanej z różną siłą, czterema metodami tj. M-JPEG, JPEG 3D 8x8x8 DCT, JPEG 3D z predykcją i JPEG 16x16 DCT. Dla każdej metody kompresji wykonano 10 pomiarów.  
Pierwszy wykres został wykonany dla sekwencji z monitoringu miejskiego, na której obserwujemy względnie mało zmian. Drugi wykres natomiast został wykonany dla sekwencji bardzo dynamicznej, na której zaobserwować możemy bardzo dużo ruchu.



Porównanie PSNR od rozmiaru dla sekwencji z monitoringu miejskiego.



Porównanie PSNR od rozmiaru dla dynamicznej sekwencji wizyjnej.

Jak widać na powyższych wykresach, JPEG 3D 8x8x8 DCT i JPEG 3D z predykcją zapewniają znacznie lepszą kompresję niż oryginalny M-JPEG, jeśli chodzi o sekwencję pochodzącą z monitoringu miejskiego. Przy takiej samej jakości obrazu (wartości PSNR) uzyskujemy około trzy razy mniejszy rozmiar kodowanej sekwencji. Dla sekwencji dynamicznej zaproponowane rozwiązania radzą sobie gorzej niż oryginalny M-JPEG, jednak różnica nie jest względnie duża, a należy też zwrócić uwagę, że proponowane rozwiązania nie powstały z myślą o kodowaniu trudnych sekwencji dynamicznych.

1. **Podsumowanie:**

Celem odbytej praktyki studenckiej było stworzenie kodera sekwencji wizyjnej, który bazując na formacie M-JPEG zapewniał by lepszą kompresję, a jednocześnie jego złożoność obliczeniowa (czasowa i pamięciowa) utrzymywała się na względnie niskim poziomie. W tym celu powstały trzy propozycję modyfikacji oryginalnego standardu M-JPEG, tj. JPEG 3D 8x8x8 DCT – wykorzystanie transformaty DCT do zredukowania nadmiarowości informacji w czasie, JPEG 3D z predykcją – predykcja podobnych bloków obrazu z obrazów poprzednich w czasie, JPEG 16x16 DCT – przejście z dwuwymiarowej transformaty DCT 8x8 do 16x16.  
Wstępne wyniki PSNR od rozmiaru dla sekwencji pochodzącej z monitoringu miejskiego pokazują, że zaproponowane rozwiązania tj. JPEG 3D 8x8x8 DCT i JPEG 3D z predykcją osiągają około trzykrotnie lepszą kompresję niż oryginalny M-JPEG. Dla dynamicznej sekwencji wizyjnej z dużą ilością ruchu wyniki okazały się być mniej zadowalające, jednakże zaproponowany koder nie powstał z myślą o kodowaniu takowych sekwencji, a wykres ma jedynie charakter czysto poglądowy.  
Podsumowując, udało się zrealizować postawiony cel. Nie jest to oczywiście wersja ostateczna. Projekt będzie dalej rozwijany poza praktyką studencką, wprowadzając modyfikacje takie jak np. zmiana sposobu kwantyzacji współczynników czy też poprawa predykcji bloków obrazu, aby otrzymać jak najwydajniejsze rozwiązanie.