

# Analiza i Implementacja Filtru Kuwahara

## Cel zadania

Głównym celem zadania była implementacja, a następnie analiza filtru Kuwahara. Jest to filtr stosowany w przetwarzaniu obrazów, który wygładza oraz zachowuje krawędzie modyfikowanego obrazu. Charakterystyczną cechą tego filtru jest podział maski na regiony, w których dla każdego regionu obliczana jest wartość średnia oraz wariancja względem tej wartości. Efektem filtracji jest wartość średnia dla tego obszaru, w którym wariancja osiąga minimum.

Podczas implementacji kodu należało posłużyć się wiedzą zdobytą na wcześniejszych zajęciach związane z dyrektywą `pragma parallel for`. Ostatnim etapem jest ocena działania zaimplementowanego filtru oraz pomiar czasu dla różnych obrazów i odpowiednio dobranych parametrów.

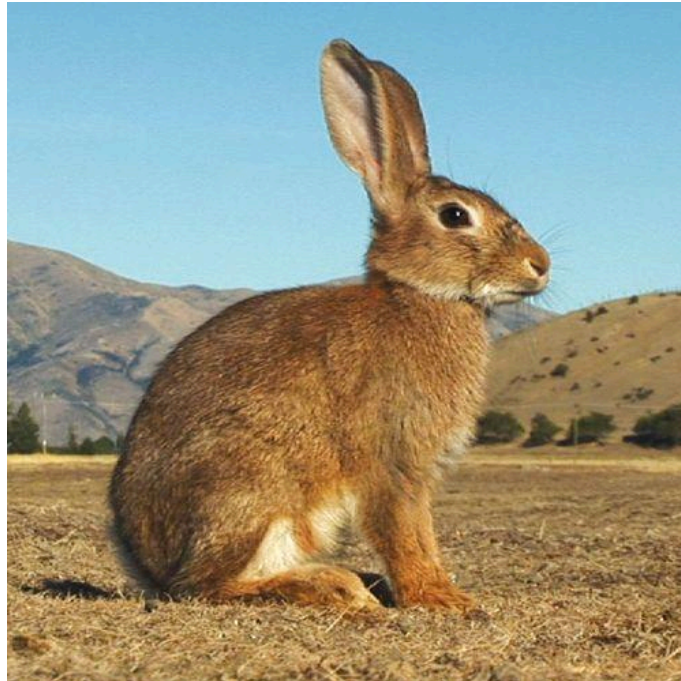
## Specyfikacja techniczna procesora

AMD Ryzen 5 2500U	
Szybkość podstawowa	2.00 GHZ
Rdzenie	4
Procesory logiczne	8
Architektura	Zen
Pamięć podręczna poziomu 1	384KB
Pamięć podręczna poziomu 2	2.0 MB
Pamięć podręczna poziomu 3	4.0 MB

## Przetwarzanie obrazów z wykorzystaniem filtra Kuwahara

1.

a. Obraz oryginalny



b. Obraz po wykorzystaniu filtra Kuwahara

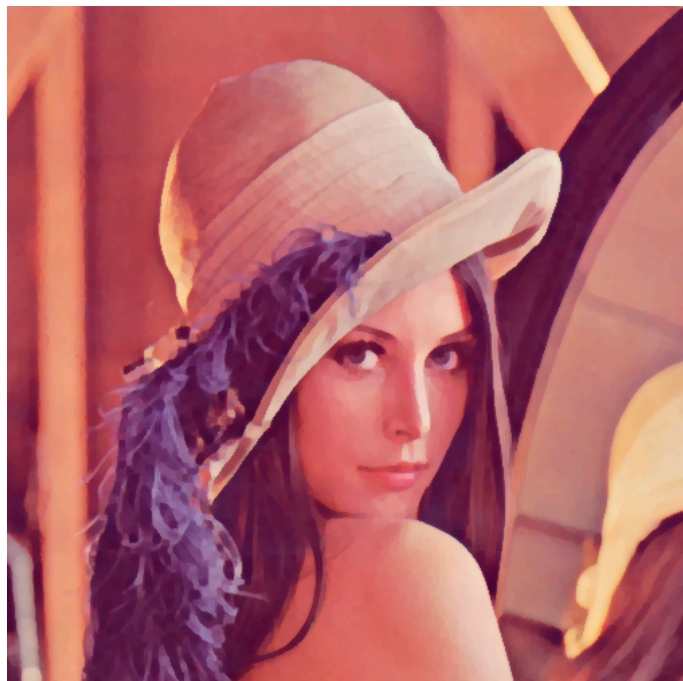


2.

a. Obraz oryginalny



b. Obraz po wykorzystaniu filtru Kuwahara



3.

a. Obraz oryginalny



b. Obraz po wykorzystaniu filtru Kuwahara





4.

a. Obraz oryginalny



b. Obraz po wykorzystaniu filtru Kuwahara

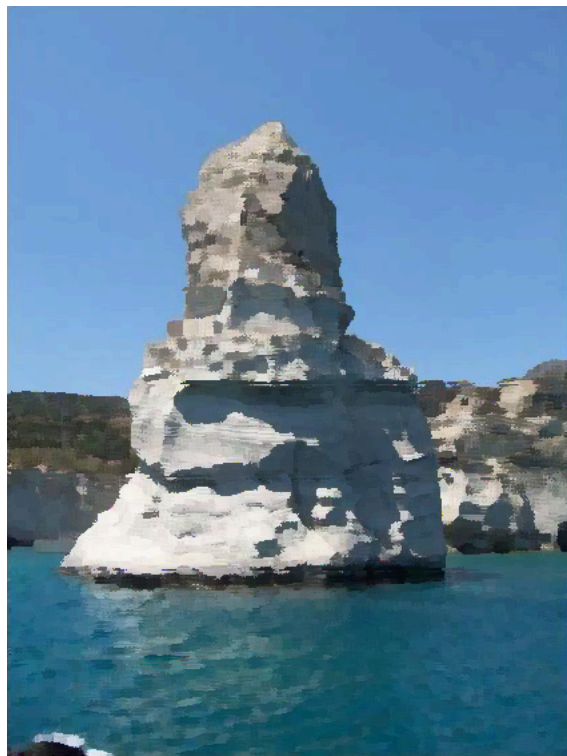


5.

a. Obraz oryginalny



b. Obraz po wykorzystaniu filtru Kuwahara



## Wyniki pomiarów czasu

Numer obrazu	Czas wykonania - przetwarzanie równoległe	Czas wykonania - przetwarzanie sekwencyjne
1	0.245072 s	1.086026 s
2	0.217366 s	1.005938 s
3	0.198379 s	0.988183 s
4	0.218197 s	0.823266 s
5	0.139908 s	0.847842 s

## Wnioski

Dokonując analizy obrazów przetworzonych przy pomocy filtru Kuwahara oraz ich czasów wykonania można sformułować kilka wniosków dotyczących ich efektywności. Filtr Kuwahara skutecznie wygładza obraz oraz redukuje szумы pozwalając jednocześnie na zachowanie krawędzi. Zastosowanie dyrektywy `pragma parallel for` umożliwia równoległe przetwarzanie kodu, co pozwoliło na znaczne przyspieszenie czasu wykonywania filtracji. W przypadku wykonywania programu sekwencyjnie bez dyrektywy czas przetwarzania był kilkukrotnie wyższy.