
Inżyniera Obrazów

laboratorium numer 5

Autor sprawozdania: Michał Dziedziak 263901

Imię i nazwisko prowadzącego kurs: dr inż. Jan Nikodem

Dzień i godzina zajęć: czwartek, 11:15 - 14:15

Spis treści

1	Temat laboratorium	3
2	Zadanie 1	3
2.1	Zadania do wykonania	3
2.2	Teoria	3
2.3	Prezentacja wykonanego zadania	4
3	Zadanie 2	5
3.1	Zadania do wykonania	5
3.2	Prezentacja wykonanego zadania	6
4	Zadanie 3	9
4.1	Zadania do wykonania	9
4.2	Teoria	9
4.3	Prezentacja wykonanego zadania	10
5	Zadanie 4	11
5.1	Zadania do wykonania	11
5.2	Prezentacja wykonanego zadania	11
6	Zadanie 5	11
6.1	Zadania do wykonania	11
6.2	Prezentacja wykonanego zadania	12

Spis rysunków

1	Zdjęcie przedstawiające obraz oryginalny, obraz w skali szarości, obraz po redukcji kolorów i obraz po przetworzeniu.	4
2	Zdjęcie przedstawiające obraz oryginalny, obraz w skali szarości, przybliżony obraz po redukcji kolorów i przybliżony obraz po przetworzeniu.	5
3	Zdjęcie przedstawiające obraz oryginalny, obraz po redukcji kolorów i obraz po przetworzeniu (dla $k = 2$).	6
4	Zdjęcie przedstawiające obraz oryginalny, przybliżony obraz po redukcji kolorów i przybliżony obraz po przetworzeniu (dla $k = 2$).	6

5	Zdjęcie przedstawiające histogram składowych kolorów przetworzonego obrazu dla $k = 2$	7
6	Zdjęcie przedstawiające obraz oryginalny, obraz po redukcji kolorów i obraz po przetworzeniu (dla $k = 4$).	7
7	Zdjęcie przedstawiające obraz oryginalny, przybliżony obraz po redukcji kolorów i przybliżony obraz po przetworzeniu (dla $k = 4$).	8
8	Zdjęcie przedstawiające histogram składowych kolorów przetworzonego obrazu dla $k = 4$	8
9	Zdjęcie przedstawiające obraz z narysowaną linią i trójkątem.	10
10	Zdjęcie przedstawiające obraz z narysowaną linią i trójkątem z gradientem kolorów.	11
11	Zdjęcie przedstawiające obraz z wygładzonymi krawędziami.	12
12	Zdjęcie przedstawiające obraz z wygładzonymi krawędziami (przybliżenie).	12

1 Temat laboratorium

W ramach piątych zajęć laboratoryjnych:

- Poznaliśmy czym jest dithering, oraz jakie są jego zastosowania.
- Poznaliśmy algorytmy do rysowania linii i trójkątów w obrazie.
- Poznaliśmy czym jest SSAA

2 Zadanie 1

2.1 Zadania do wykonania

Zadanie pierwsze polegało na zaimplementowaniu ditheringu dla obrazu w skali szarości. W tym celu wykonałem następujące kroki:

1. Przeniosłem obraz z przestrzeni barw RGB na skalę szarości.
2. Użyłem algorytmu Floyda-Steinberga do nałożenia ditheringu na obraz.
3. Wyświetliłem obraz oryginalny, obraz po redukcji kolorów i obraz przetworzony.

2.2 Teoria

Redukcja palety barw

Redukcja palety kolorów obrazu polega na zmniejszeniu liczby unikalnych kolorów w celu uproszczenia danych obrazu, zmniejszenia zużycia pamięci lub uzyskania określonego efektu wizualnego. Typowo każdą składową koloru reprezentujemy jako jedną z 256 wartości, co daje 16 777 216 unikalnych odcieni. Niektóre urządzenia i formaty graficzne są ograniczone w tej kwestii, co skutkuje mniejszą liczbą dostępnych odcieni.

Redukcja palety kolorów zmniejsza liczbę tych odcieni, mapując każdą wartość koloru do najbliższego odpowiednika w ograniczonej przestrzeni kolorów. Na przykład, przy redukcji do dwóch kolorów (czarny i biały), wartości kolorów poniżej 128 są zaokrąglane do 0 (czarny), a powyżej 128 do 255 (biały).

W obrazach RGB zmniejsza się paletę kolorów dla każdego z kanałów osobno, traktując każdy kanał jako oddzielny obraz w skali szarości. Rezultatem jest obraz używający mniej unikalnych kolorów, ale zachowujący ogólną strukturę i wygląd oryginału.

Dithering

Dithering to technika stosowana w grafice komputerowej w celu symulacji większej liczby kolorów i odcieni w obrazach o ograniczonej palecie kolorów. Polega na rozmieszczaniu pikseli o różnych kolorach w taki sposób, aby stworzyć iluzję dodatkowych kolorów i płynniejszych przejść tonalnych.

Dithering działa poprzez wprowadzanie drobnych, losowych wzorców lub regularnych układów pikseli, które ludzkie oko interpretuje jako nowe odcienie. Na przykład, na obrazie zredukowanym do czerni i bieli, dithering może mieszać czarne i białe piksele, aby zasymulować odcienie szarości.

Algorytm Floyda-Steinberga

Algorytm Floyda-Steinberga to jeden z najpopularniejszych algorytmów ditheringu, stosowany do poprawy jakości obrazów o ograniczonej liczbie kolorów. Algorytm ten jest techniką ditheringu rozproszonego błędu (error diffusion dithering), co oznacza, że błąd powstały podczas kwantyzacji koloru jest rozpraszany na sąsiednie piksele, co skutkuje bardziej naturalnym wyglądem obrazu.

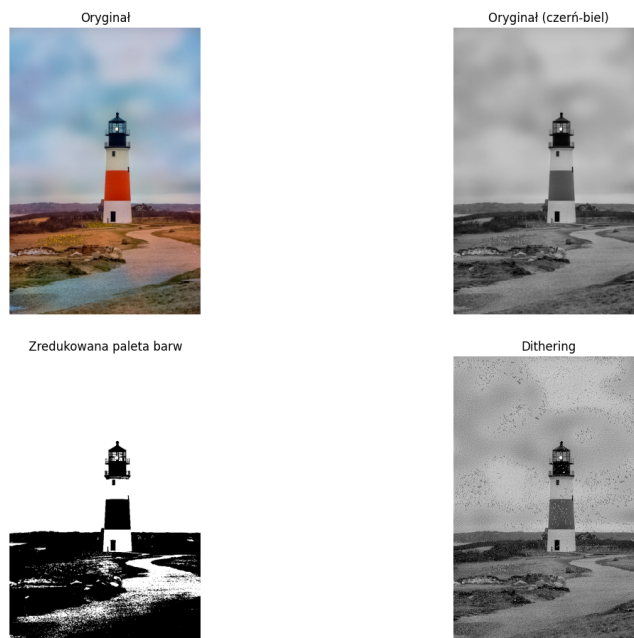
Algorytm Floyda-Steinberga działa w następujący sposób:

1. Kwantyzacja piksela: Każdy piksel w obrazie jest przetwarzany pojedynczo. Kolor aktualnego piksela jest zaokrąglany do najbliższego koloru w ograniczonej palecie.
2. Obliczenie błędu: Błąd kwantyzacji jest różnicą między oryginalnym kolorem piksela a nowym kolorem po kwantyzacji.
3. Rozproszenie błędu: Błąd ten jest następnie rozpraszany na sąsiednie piksele, które jeszcze nie zostały przetworzone. Rozproszenie błędu odbywa się według macierzy:

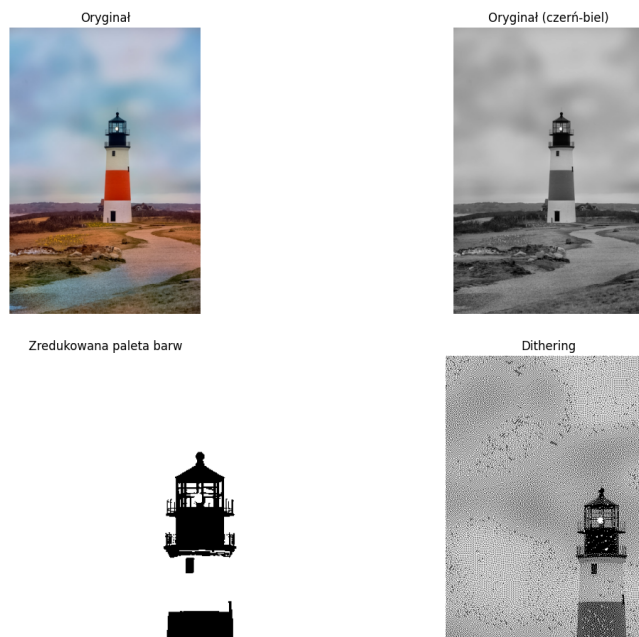
$$\begin{bmatrix} - & * & 7/17 \\ 3/16 & 5/16 & 1/16 \end{bmatrix}$$

Dodajemy błąd przeskalowany przez wartość z macierzy do pikseli sąsiednich (względem piksela oznaczonego).

2.3 Prezentacja wykonanego zadania



Rysunek 1: Zdjęcie przedstawiające obraz oryginalny, obraz w skali szarości, obraz po redukcji kolorów i obraz po przetworzeniu.



Rysunek 2: Zdjęcie przedstawiające obraz oryginalny, obraz w skali szarości, przybliżony obraz po redukcji kolorów i przybliżony obraz po przetworzeniu.

3 Zadanie 2

3.1 Zadania do wykonania

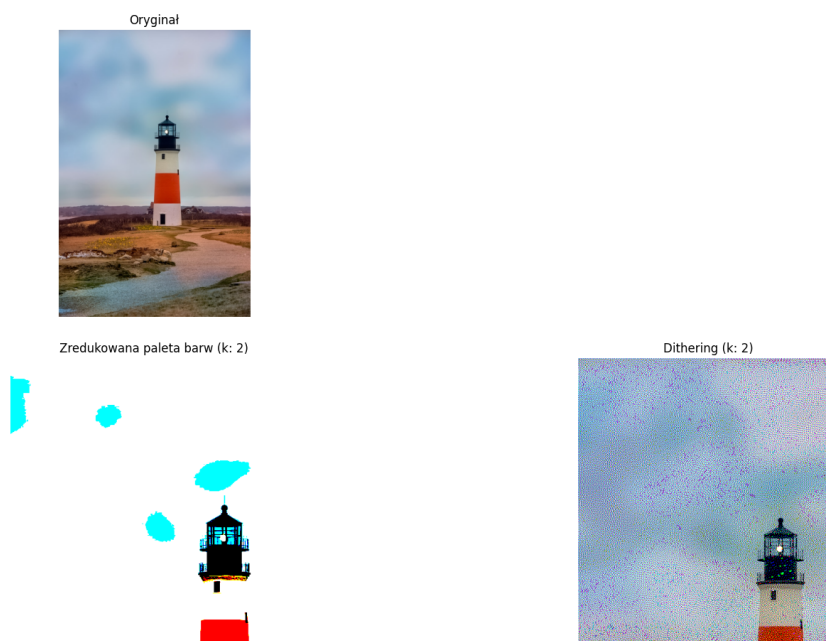
Zadanie drugie polegało na zaimplementowaniu ditheringu dla obrazu w kolorze. Wykonane kroki były analogiczne do poprzedniego zadania, z tą różnicą, że każdy kanał koloru był przetwarzany osobno.

Dodatkowo została zaimplementowana możliwość wyboru liczby kolorów, do których ma zostać zredukowana paleta.

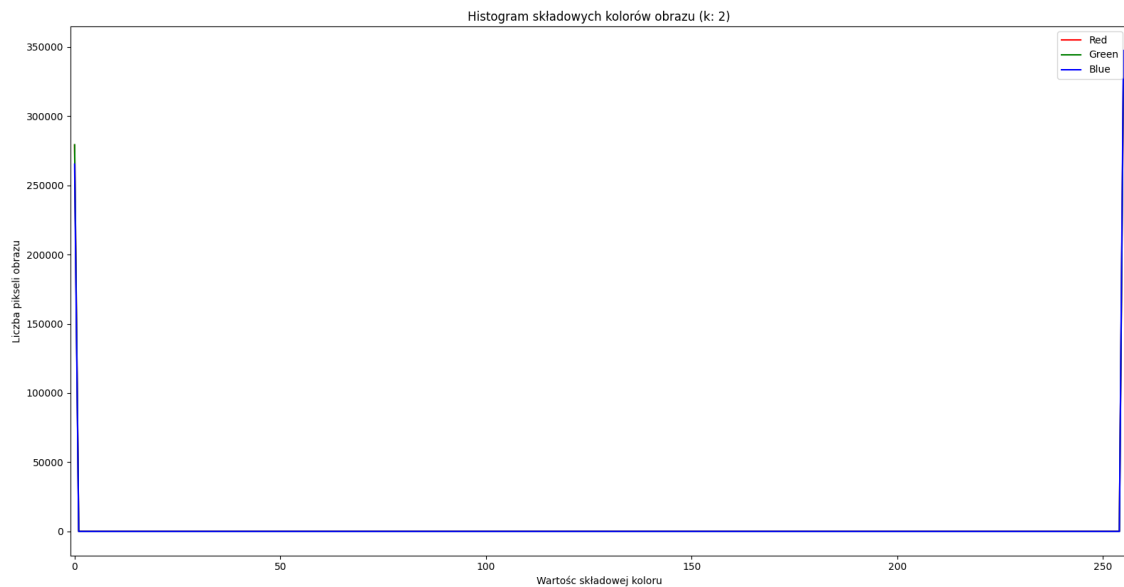
3.2 Prezentacja wykonanego zadania



Rysunek 3: Zdjęcie przedstawiające obraz oryginalny, obraz po redukcji kolorów i obraz po przetworzeniu (dla $k = 2$).



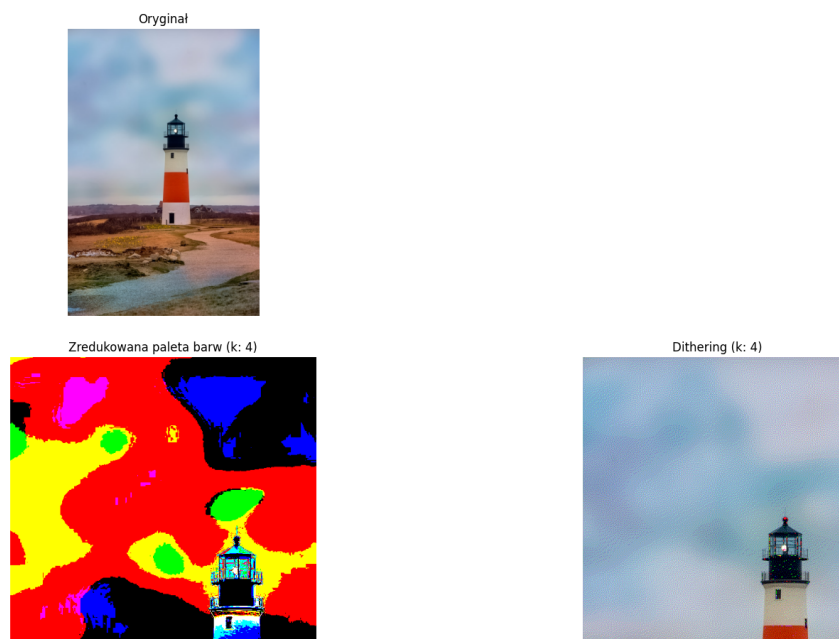
Rysunek 4: Zdjęcie przedstawiające obraz oryginalny, przybliżony obraz po redukcji kolorów i przybliżony obraz po przetworzeniu (dla $k = 2$).



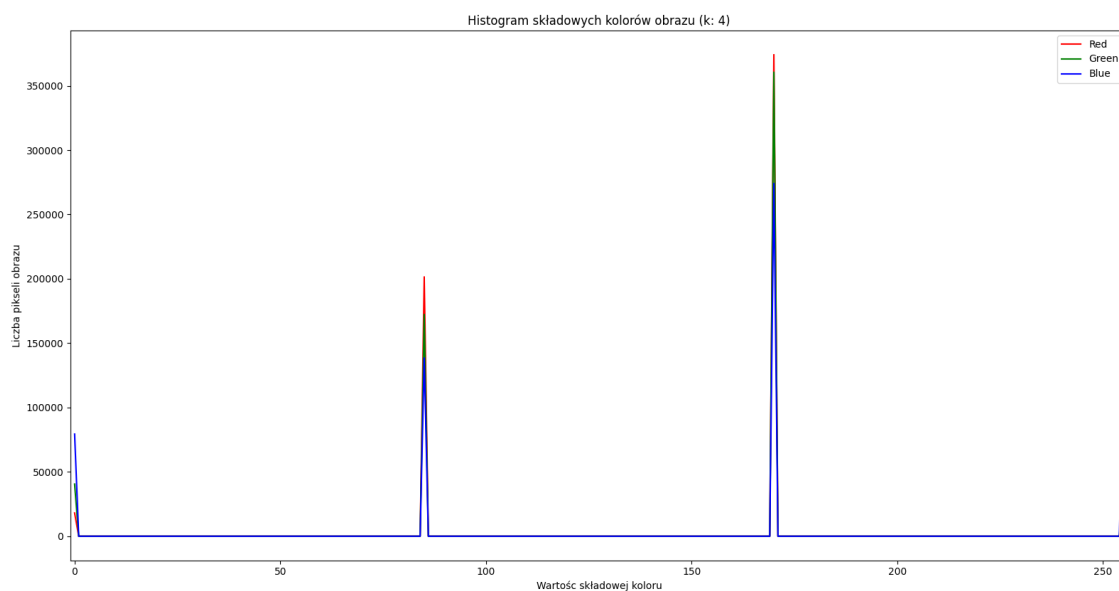
Rysunek 5: Zdjęcie przedstawiające histogram składowych kolorów przetworzonego obrazu dla $k = 2$



Rysunek 6: Zdjęcie przedstawiające obraz oryginalny, obraz po redukcji kolorów i obraz po przetworzeniu (dla $k = 4$).



Rysunek 7: Zdjęcie przedstawiające obraz oryginalny, przybliżony obraz po redukcji kolorów i przybliżony obraz po przetworzeniu (dla $k = 4$).



Rysunek 8: Zdjęcie przedstawiające histogram składowych kolorów przetworzonego obrazu dla $k = 4$

4 Zadanie 3

4.1 Zadania do wykonania

Zadanie trzecie polegało na zaimplementowaniu algorytmu Bresenhama do rysowania linii oraz algorytmu do rysowania trójkątów. Figury miały być jednolite kolorystycznie.

4.2 Teoria

Algorytm Bresenhama

Algorytm Bresenhama to wydajna metoda rysowania linii prostych na rastrowym wyświetlaczu komputerowym. Algorytm działa poprzez iteracyjne wyznaczanie pikseli najbliższych idealnej linii prostej między dwoma punktami.

Kroki algorytmu Bresenhama:

1. Wyznaczenie wielkości pomocniczych:

- Oblicz różnice w pozycjach końcowych punktów linii:

$$dx = |x_2 - x_1|$$

$$dy = |y_2 - y_1|$$

- Określ kierunki kroków

$$Xi = 0 \text{ jeżeli } dx = 0 \text{ w innym przypadku } \text{znak}(x_2 - x_1)$$

$$Yi = 0 \text{ jeżeli } dy = 0 \text{ w innym przypadku } \text{znak}(y_2 - y_1)$$

2. Określenie początkowej wartości błędu:

- Jeżeli $dx > dy$ to $d = 2 * dy - dx$
- Jeżeli $dy > dx$ to $d = 2 * dx - dy$

3. Rysowanie w punkcie początkowym:

- Narysuj piksel w punkcie (x_0, y_0)

4. Powtarzanie w pętli aż do osiągnięcia punktu docelowego:

- Jeżeli $dx > dy$:

- $x_0+ = Xi$

- $d+ = 2 * dy$

- Jeżeli $d \geq 0$:

- * $y_0+ = Yi$

- * $d- = 2 * dx$

- Jeżeli $dy > dx$:

- $y_0+ = Yi$

- $d+ = 2 * dx$

- Jeżeli $d \geq 0$:

- * $x_0+ = Xi$

- * $d- = 2 * dy$

gdzie $\text{znak}()$ - funkcja zwracająca znak liczby (+1/ - 1)

Rysowanie trójkątów

Trójkąt jest powszechnie używaną figurą w grafice komputerowej. Na jego podstawie budują się bardziej skomplikowane kształty, takie jak wielokąty, krzywe czy powierzchnie.

Do rysowania trójkątów można użyć różnych algorytmów, na potrzeby zadania zaimplementowałem algorytm, którego kroki przedstawiają się następująco:

1. Wyznaczenie pomocniczych wartości:

- Wyznaczenie czworokąta otaczającego trójkąt (Trójkąt ma boki: a , b , c . Każdy bok ma składową x i y):

$$x_{\min} = \min(a, b, c)$$

$$y_{\min} = \min(a, b, c)$$

$$x_{\max} = \max(a, b, c)$$

$$y_{\max} = \max(a, b, c)$$

2. Dla każdego punktu $(p.x, p.y)$ wewnątrz czworokąta:

- Sprawdź, czy punkt znajduje się wewnątrz trójkąta:

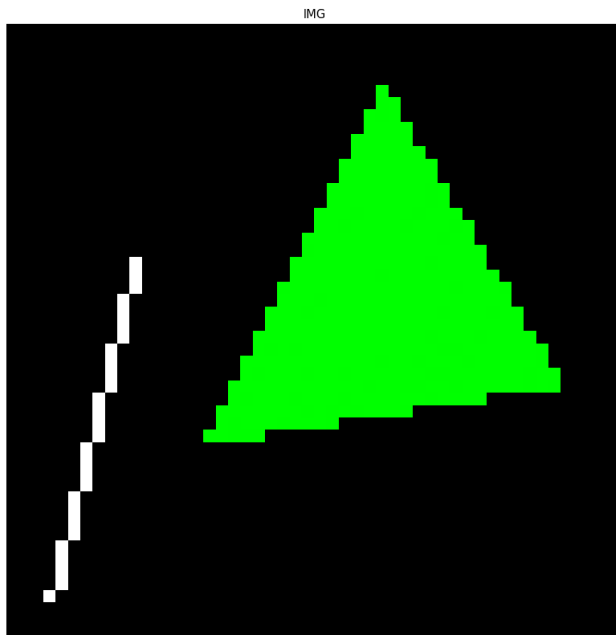
$$znak_1 = (p.x - a.x)(b.y - a.y)(p.y - a.y)(b.x - a.x)$$

$$znak_2 = (p.x - b.x)(c.y - b.y)(p.y - b.y)(c.x - b.x)$$

$$znak_3 = (p.x - c.x)(a.y - c.y)(p.y - c.y)(a.x - c.x)$$

- Jeżeli $znak_1 = znak_2 = znak_3$ to punkt znajduje się wewnątrz trójkąta

4.3 Prezentacja wykonanego zadania



Rysunek 9: Zdjęcie przedstawiające obraz z narysowaną linią i trójkątem.

5 Zadanie 4

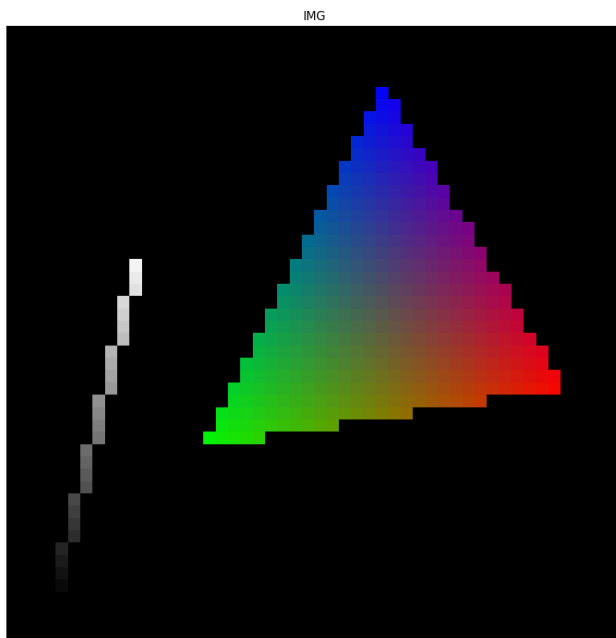
5.1 Zadania do wykonania

W tym zadaniu tak samo jak poprzednio mieliśmy narysować linię i trójkąt, ale tym razem kolory nie miały być jednolite, a miały być gradientem.

Dla linii wykorzystałem interpolację liniową koloru w zależności od położenia piksela na linii. Im bliżej punktu początkowego linii tym kolor był bardziej zbliżony do koloru początkowego, a im bliżej punktu końcowego tym bardziej zbliżony do koloru końcowego.

Dla trójkąta wykorzystałem interpolację barycentryczną. W tym przypadku kolor piksela był obliczany na podstawie współrzędnych barycentrycznych punktu w trójkącie. Im bliżej jednego z wierzchołków tym kolor był bardziej zbliżony do koloru tego wierzchołka.

5.2 Prezentacja wykonanego zadania



Rysunek 10: Zdjęcie przedstawiające obraz z narysowaną linią i trójkątem z gradientem kolorów.

6 Zadanie 5

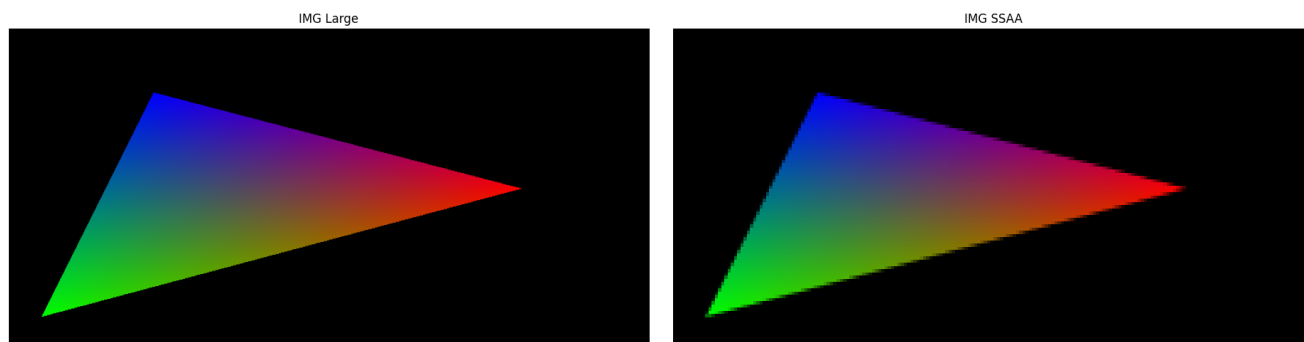
6.1 Zadania do wykonania

W ramach zadania piątego mieliśmy zaimplementować wygładzanie krawędzi w generowanym obrazie. W tym celu wykorzystaliśmy algorytm SSAA (Super Sampling Anti Aliasing).

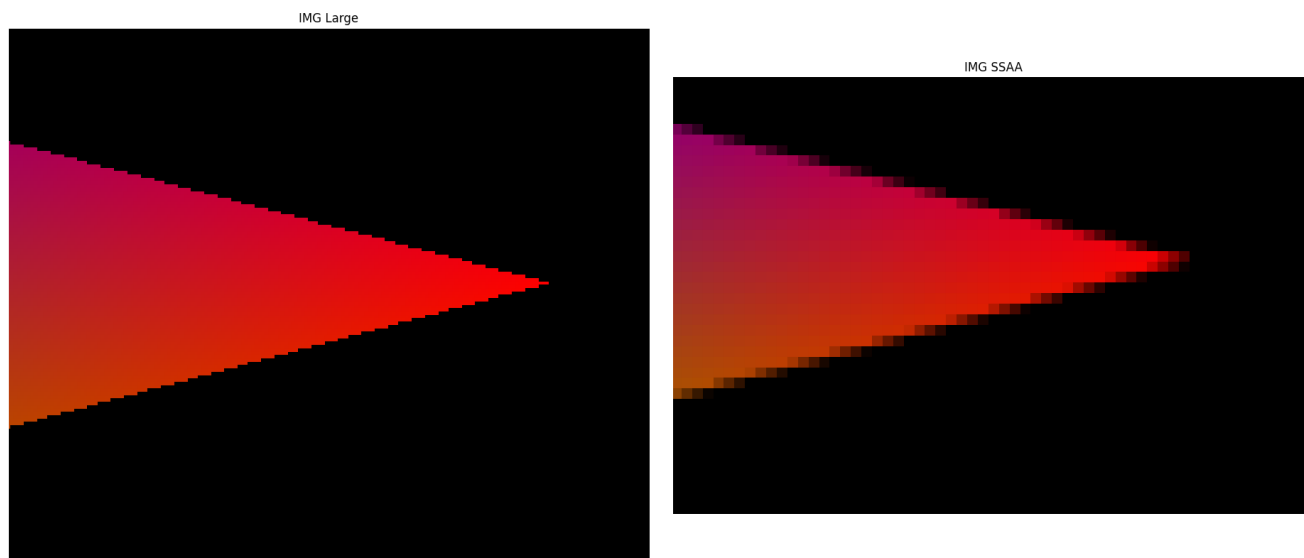
Algorytm SSAA polega na generowaniu obrazu o większej rozdzielczości niż docelowa, a następnie zmniejszeniu go do docelowej rozdzielczości. W ten sposób uzyskujemy wygładzenie krawędzi.

Zaimplementowany algorytm najpierw rysuje obraz dwukrotnie większy niż docelowy, a następnie z każdych czterech pikseli oblicza średni kolor i rysuje go na obrazie docelowym.

6.2 Prezentacja wykonanego zadania



Rysunek 11: Zdjęcie przedstawiające obraz z wygładzonymi krawędziami.



Rysunek 12: Zdjęcie przedstawiające obraz z wygładzonymi krawędziami (przybliżenie).