

Laboratorium 01

Oskar Lewna

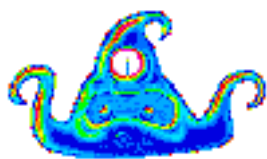
March 10, 2025

1 Pointylizm



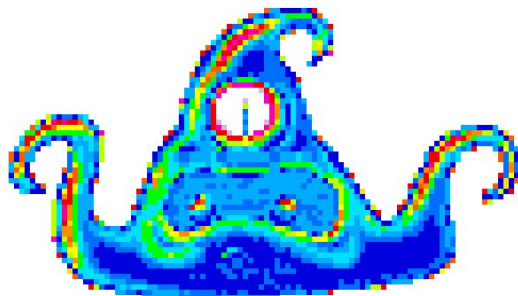
Obrazek po lewej stronie jest oryginalny. Ten po prawej został przerobiony w stylu kropkowanym. Algorytm, który go utworzył wybierał losowy piksel, następnie pobierał jego kolory i w zakresie od 1 do 20 pikseli tworzył kropkę. Widać, że obraz teraz nie jest tak dokładny, utracił szczegóły, jak np: brwi, które w pełni zniknęły.

2 Skalowanie



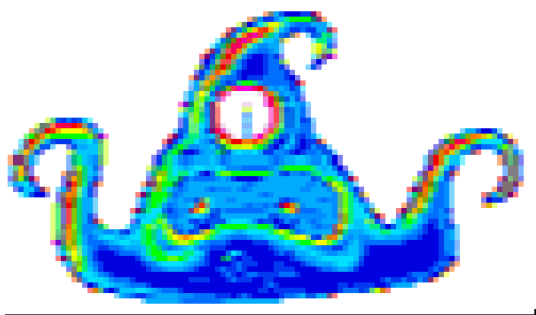
Oryginalny obrazek

2.1 a) algorytm Nearest Neighbor



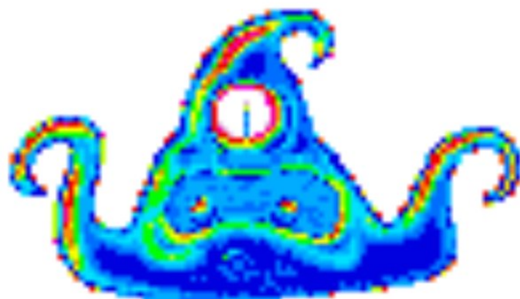
Według mnie, ten obrazek został dobrze przeskalowany, tylko dlatego, że jest duży i oko człowieka nie zauważy powtórzeń pojedynczych pikseli.

2.2 b) algorytm Nearest Neighbor ze średnią



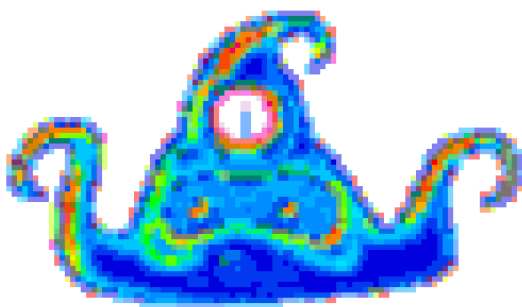
W tym przypadku, średnia wartość dwóch najbliższych pikseli spowodowała, że obraz jest rozmazany i nieostry. Dodatkowo obraz jest przesunięty w lewo. Ta technika mogłaby być dobra do wielkich obrazów.

2.3 c) algorytm interpolacji dwuliniowej



Ten obraz został przeskalowany z algorytmem wbudowanym w ImageJ. Wygląda mniej rozmyty niż poprzedni przykład. Piksele są mniej widoczne i kolorowy lepiej się łączą, nie widać krawędzi tych pikseli.

2.4 d) algorytm interpolacji dwuliniowej ze średnią



Tutaj obraz stracił niektóre szczegóły przez uśrednienie jasności z najbliższych pikseli. Wydaje się, że obraz się powiększył o jeden piksel dodatkowej ramki.

3 Dithering

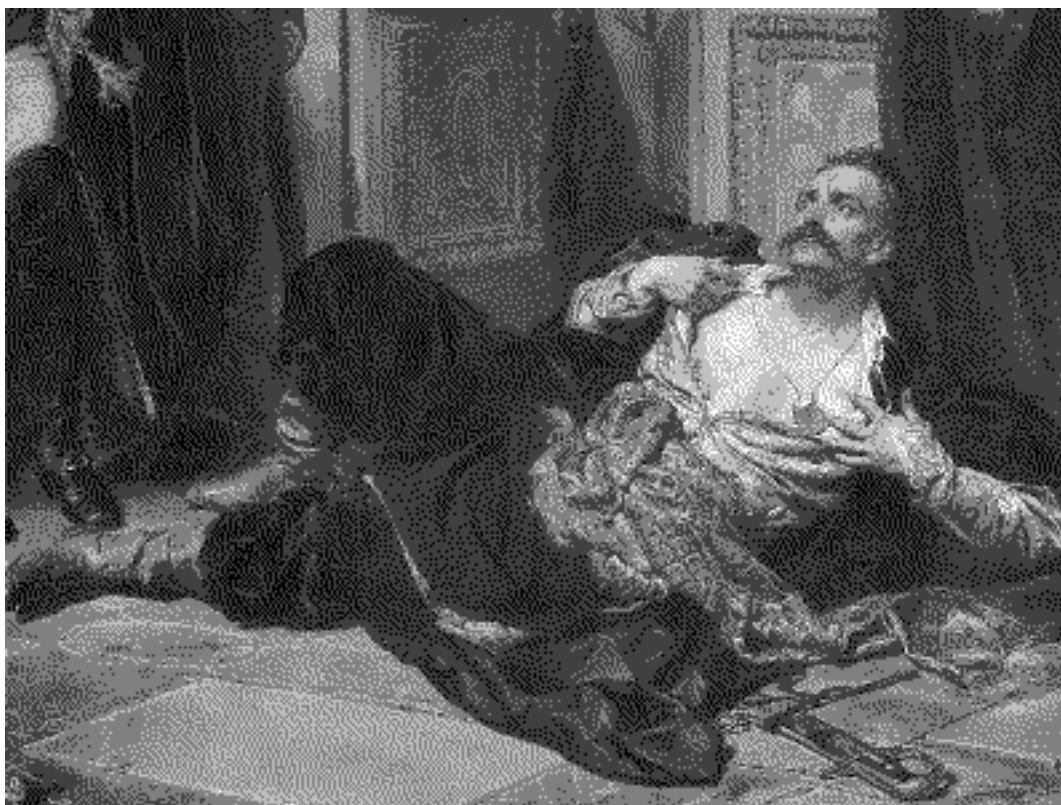
3.1 algorytm Floyda-Steinberga

3.1.1 a) progowanie $T = 128$



Progowanie spowodowało, że piksele są albo czarne albo białe. Po przybliżeniu można dokładnie zauważyć piksele w tych kolorach, które z większej odległości powodują złudzenie, jakby obraz był stworzony z różnych odcieni szarości.

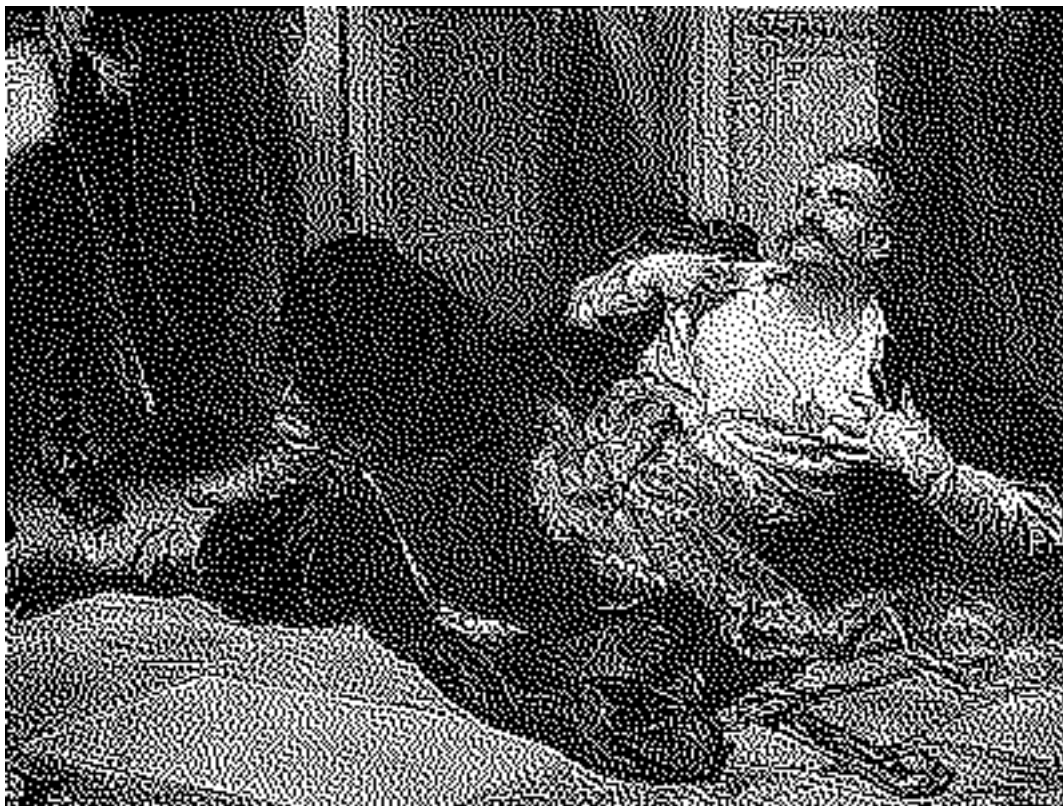
3.1.2 a) progowanie pięcioma wartościami [0,64,128,192,255]



Progowanie 5 wartościami szarości wygląda jeszcze lepiej. Z większej odległości nie widać różnicy między oryginalny obrazem, a tym po wykonaniu ditcheringu. Ten algorytm z tym progowaniem wygląda według mnie najlepiej. Detale są widoczne, ale gdyby było jeszcze kilka progów to ten obraz prawdopodobnie byłby bardzo bliski oryginałowi.

3.2 algorytm Jarvis-Judice-Ninke

3.2.1 a) progowanie $T = 128$



Moim zdaniem, ten algorytm nie działa dobrze z tak surowym progowaniem. Z pewnością lepiej pokazuje szczegóły niż algorytm Floyda-Steinberga z tym samym progowaniem. Przykładowo naszyjnik na szyji jest lepiej widoczny lub nawet pas na głównej postaci.

3.2.2 a) progowanie pięcioma wartościami [0,64,128,192,255]



Algorytm Jarvis-Judice-Ninke działa lepiej, z większą ilością wartości w progowaniu. Dokładniej są pokazane cienie oraz światło. Dodatkowo piksele nie występują pojedynczo tak często jak w algorytmie Floyda-Steinberga. Są bardziej zbite po kilka, kilkanaście, a nawet kilkadziesiąt niż po jednym.

4 Dithering metodą zmiennego progu



Obraz po ditheringu metodą zmiennego progu jest powiększony o wielkość macierzy (w tym przypadku $\times 4$). Wszystko jest doskonale widoczne, jednak kosztem pamięci.

5 Dithering z macierzą Bayer'a

5.1 a) paleta 1-bitowa



Na obrazie po zastosowaniu ditheringu z macierzą Bayer'a dla czarno-białej palety widać krzyżyki na obrazie. Stracił on już szczegółowość, jednak zarys głównej postaci jest nadal widoczny.

5.2 a) paleta 4 wartości [50, 100, 150, 200]



Po przeprowadzeniu ditheringu z macierzą Bayera wielkości 8 oraz z użyciem palety 4 wartości szarości, widać że obraz jest ciemniejszy. Dodatkowo są powtarzające się symbole, tylko w różnych wartościach szarości. Obraz jest jednak nadal widoczny.