

# Podstawy Programowania: Przetwarzanie obrazów cz.1"

Dawid Krekora

11 grudnia 2020

# 1 Opis programu

Program pozwala na wykonywanie podstawowych modyfikacji na obrazach monochromatycznych w formacie PGM. Opiera swoje działanie na podstawowych informacjach które możemy odczytać z obrazu uruchamiając go np. w edytorze tekstowym. W zapisie tych informacji ważna jest kolejność, prezentują się następująco:

- Numer magiczny (w przypadku obrazów w formacie PGM jest to 'P2')
- Szerokość oraz wysokość obrazu
- Skala szarości (informuje o tym ile odcieni szarości obejmuje obraz)
- Informacja o poszczególnych pikselach (rozmieszczonych zgodnie z informacją o wymiarach obrazu)

Program w momencie odczytania, interpretacji i zapisu tychże informacji przystępuje do obróbki obrazu zgodnie z wyborem użytkownika. Same obróbki polegają na modyfikacji pojedynczych pikseli które zmieniają im przypisaną wartość szarości by następnie zapisać całą zawartość w nowym pliku o wybranej nazwie.

# 2 Kod programu

Z racji obszerności kodu programu nie został on włączony do tego sprawozdania, jego treść zawarta zostanie w osobnym pliku tekstowym.

# 3 Wprowadzanie danych do programu. Błędy strumienia wejścia.

## 3.1 Uchwyty pliku

- Dane wejściowe: plik który nie istnieje
- Dane wyjściowe: komunikat o błędzie

```
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./test
Program operujący na obrazach PGM.
Podaj nazwę pliku:
plik_ktory_nie_istnieje.pgm
Błąd: Nie podano uchwytu do pliku
Segmentation fault (core dumped)
dawid@dawid-Aspire-E5-575T:~/Pulpit$
```

Rysunek 1: Błąd uchwytu pliku

### 3.2 Niewłaściwy typ pliku

- Dane wejściowe: plik który nie jest w formacie PGM
- Dane wyjściowe: komunikat o błędzie

```
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./test
Program operujący na obrazach PGM.
Podaj nazwę pliku:
1.png
Błąd: To nie jest plik PGM
dawid@dawid-Aspire-E5-575T:~/Pulpit$
```

Rysunek 2: Błąd typu pliku

### 3.3 Brak danych

- Dane wejściowe: plik który zawiera braki w informacjach o wymiarach obrazu lub liczbie odcieni szarości
- Dane wyjściowe: komunikat o błędzie

```
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./test
Program operujący na obrazach PGM.
Podaj nazwę pliku:
Kubus.pgm
Błąd: Brak wymiarów obrazu lub liczby stopni szarości
dawid@dawid-Aspire-E5-575T:~/Pulpit$
```

Rysunek 3: Brak odpowiednich danych w pliku

### 3.4 Błędne wymiary obrazu

- Dane wejściowe: plik który zawiera za
- Dane wyjściowe: komunikat o błędzie

```
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./test
Program operujacy na obrazach PGM.
Podaj nazwe pliku:
Kubus.pgm
Bład: Niewlasciwe wymiary obrazu
dawid@dawid-Aspire-E5-575T:~/Pulpit$
```

Rysunek 4: Niewłaściwe wymiary obrazu

### 3.5 Błędne wymiary obrazu

- Dane wejściowe: plik który zawiera za
- Dane wyjściowe: komunikat o błędzie

```
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./test
Program operujacy na obrazach PGM.
Podaj nazwe pliku:
Kubus.pgm
Bład: Niewlasciwe wymiary obrazu
dawid@dawid-Aspire-E5-575T:~/Pulpit$
```

Rysunek 5: Niewłaściwe wymiary obrazu

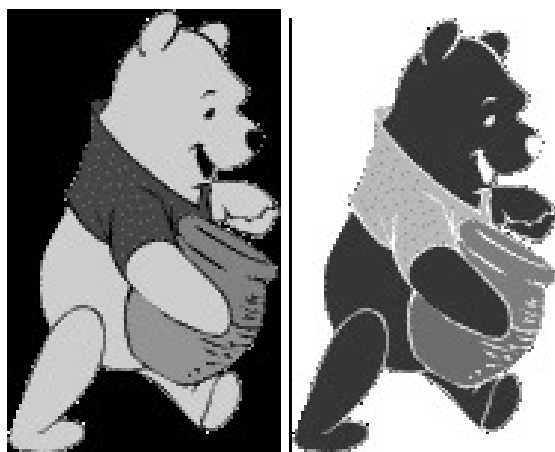
## 4 Funkcje operujące na obrazie. Operacje punktowe.

Operacje punktowe należą do najprostszej klasy operacji które możemy zastosować na obrazie. Przeprowadzane są tylko i wyłącznie na podstawie wartości każdego z pikseli z osobna, nie jest natomiast istotne jego sąsiedztwo. Do poprawnego działania funkcji przedstawionych w tym dziale niezbędne jest wprowadzenie pewnych wartości które zostaną przekierowane na strumień wejścia podczas dalszych operacji. Takie rozwiązanie generuje potrzebę implementacji dodatkowych zabezpieczeń w programie, które w przypadku podania błędnych wartości wygenerują odpowiednie komunikaty. Rodzaje komunikatów o błędach:

- Błąd wprowadzenia wartości niebędącej liczbą
- Błąd podania liczby spoza wymaganego zakresu
- Błąd wymagań co do zależności pomiędzy wartością dwóch parametrów (funkcja 'zmiana poziomów')

### 4.1 Funkcja negatywu

Funkcja 'negatyw' modyfikuje piksele tak, aby uzyskać efekt przeciwnych kolorów. Działanie funkcji zostało przedstawione na rysunkach poniżej:



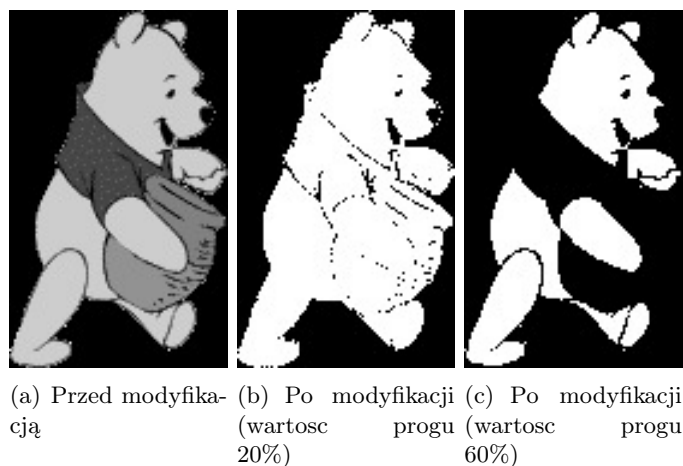
(a) Przed modyfikacją

(b) Po modyfikacji

Rysunek 6: Działanie funkcji negatyw

## 4.2 Funkcja progowania

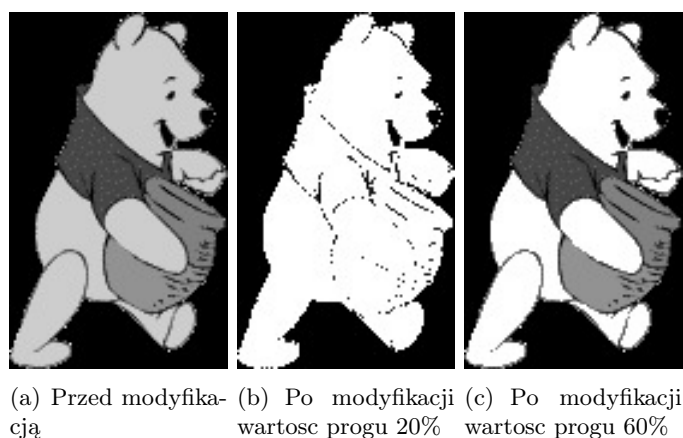
Funkcja 'progowanie' opiera swoje działanie na wstępnej klasyfikacji piksela do grupy zbiorów punktów czarnych lub białych. Wszystkie piksele które znajdują się poniżej wartości progu zostaną maksymalnie przyciemnione, a te powyżej - maksymalnie rozjaśnione.



Rysunek 7: Działanie funkcji progowanie

## 4.3 Funkcja półprogowania bieli

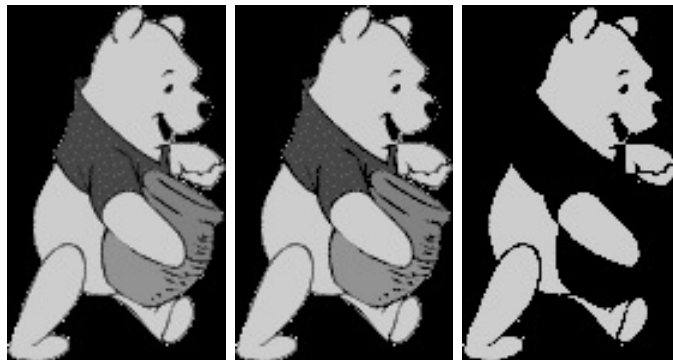
Funkcja 'progowanie bieli' tak jak i 'progowanie czerni' działają bardzo podobnie jak samo 'progowanie'. Różnica tym razem tkwi w braku modyfikacji wartości pikseli po jednej, wybranej stronie progu. Dla 'progowanie bieli' niezmodyfikowanymi pikselami będą te poniżej wartości progu, te powyżej otrzymają maksymalną możliwą wartość szarości.



Rysunek 8: Działanie funkcji półprogowanie bieli

#### 4.4 Funkcja półprogowania czerni

Funkcja działa przeciwnie do 'progowanie bieli'. Wszystkim wartościom poniżej progu przypisuje wartość 0, piksele powyżej progu pozostają niezmodyfikowane.

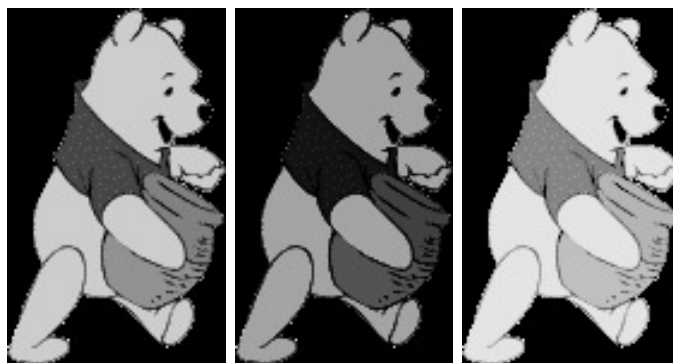


(a) Przed modyfikacją (b) Po modyfikacji (c) Po modyfikacji  
wartość progu 20% wartość progu 60%

Rysunek 9: Działanie funkcji półprogowanie czerni

#### 4.5 Funkcja korekcji gamma

Funkcja 'korekcja gamma' pozwala na przeskalowanie wartości pikseli na całym obrazie w rezultacie rozjaśniając bądź przyciemniając ich wartość. Wartości parametru gamma poniżej 1 przyciemniają obraz natomiast te powyżej 1 go rozjaśniają.

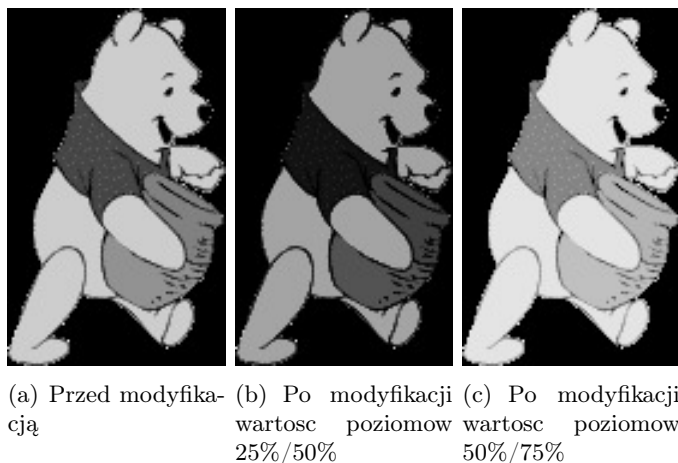


(a) Przed modyfikacją (b) Po modyfikacji (c) Po modyfikacji  
wartość progu 20% wartość progu 60%

Rysunek 10: Działanie funkcji korekcji gamma

## 4.6 Funkcja zmiany poziomów bieli i czerni

Funkcja 'zmiana poziomów' jest połączeniem funkcji 'progowanie bieli' oraz 'progowanie czerni'. Na wejściu przyjmuje ona dwa argumenty będące wartościami progów. Wszystkie piksele których wartość nie przekracza wartości mniejszego argumentu zostają zmniejszone do 0, a te powyżej wartości większego argumentu zwiększone do maksymalnego poziomu odcieni szarości w obrazie. To co znajduje się pomiędzy tymi dwoma wartościami zostanie natomiast przeskalowane na całym dostępnym w programie zakresie szarości.



Rysunek 11: Działanie funkcji zmiany poziomów

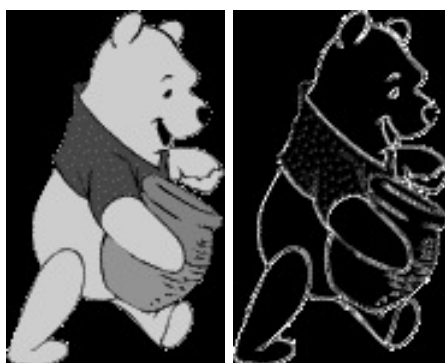


## 5 Funkcje operujące na obrazie. Operacje lokalne

Klasa funkcji operacji lokalnych łączy w sobie cechy operacji punktowych dokładając do końcowej wartości informacje o zależności pomiędzy sąsiednimi pikselami.

### 5.1 Funkcja konturowania

Funkcja 'konturowanie' opiera swoje działanie na metodzie gradientowej, co oznacza, że doszukuje się różnic pomiędzy sąsiednimi pikselami. W rezultacie piksele które różnią się od otoczenia zostaną rozjaśnione, a te które są podobne - przyciemnione.

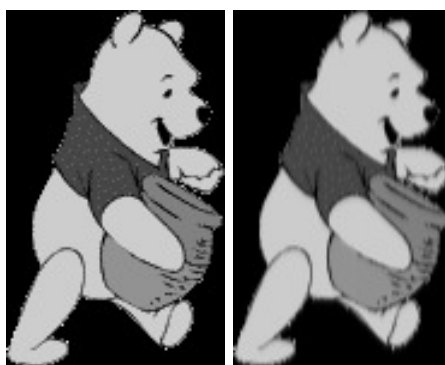


(a) Przed modyfikacją (b) Po modyfikacji

Rysunek 12: Działanie funkcji konturowania

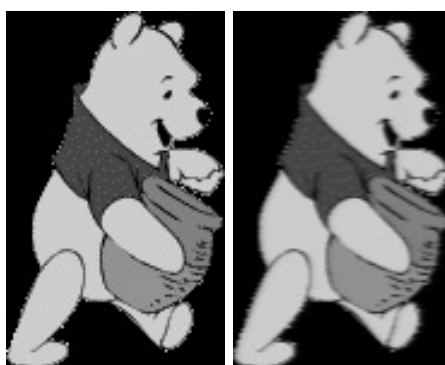
## 5.2 Funkcje rozmywania poziomego i pionowego

Funkcje te są identyczne w swoim działaniu, różnica przejawia się w orientacji osi wzdłuż której prowadzone są zmiany. W przypadku 'rozmywania pionowego' pod uwagę bierzemy sąsiedztwo pikseli leżących nad i pod docelowym punktem. 'Rozmywanie poziome' interpretuje z kolei piksele znajdujące się po bokach. Całość procesu ma za zadanie zmniejszenie ostrości obrazu zgodnie z kierunkiem podanym w nazwie funkcji.



(a) Przed modyfikacją (b) Po modyfikacji

Rysunek 13: Działanie rozmywania poziomego o promieniu 1



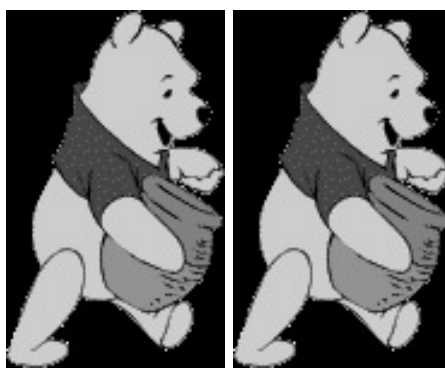
(a) Przed modyfikacją (b) Po modyfikacji

Rysunek 14: Działanie rozmywania pionowego o promieniu 1

## 6 Operacje globalne. Funkcja rozciągania histogramu

Funkcja 'rozciągania histogramu' przypomina swoją funkcjonalnością funkcję zmiany poziomów bieli i czerni. Różnica między nimi jest taka, że tym razem progi operacji nie będą wybrane przez nas ale przeszukiwane w tablicy zawierającej informacje o wartościach poszczególnych pikseli. Tym samym progowi dolnemu zostanie przypisana wartość najciemniejszego piksela, a górnemu tego najjaśniejszego. W etapie końcowym wszystkie piksele zawarte w pliku zostaną przeskalowane tak, aby wykoszystać cały zakres odcieni szarości dostępny w programie

Uwaga: implementacja funkcji na pliku kubus.pgm nie przyniosła rezultatów zgodnych z tymi przedstawionymi w dokumencie zamieszczonym w materiałach. Wynika to z faktu, że wykorzystano w nim pełny zakres szarości dostępny w obrazie, w wyniku czego wartości MAX i MIN równoważyły się wraz z pozostałymi zmiennymi. W związku z tym zastosowano również funkcję na obrazie Lena.pgm co spowodowało przyciemnienie obrazu



(a) Przed modyfikacją (b) Po modyfikacji

Rysunek 15: Działanie funkcji rozciągania na pliku kubus.jpg



(a) Przed modyfikacją (b) Po modyfikacji

Rysunek 16: Działanie funkcji rozciągania na pliku Lena.pgm

## 7 Podsumowanie i wnioski

Projekt "Przetwarzanie obrazów cz.1" choć wydawał się trudny w początkowym etapie tworzenia, udowodnił prawdziwość reguły mówiącej o tym, że 'nie ma rzeczy niemożliwych - wszystko zależy od tego ile czasu czemuś poświęcimy'. Dzięki rozłożeniu problemu na części pierwsze, jego zrozumieniu, a następnie ponownej syntezie rozwiązanie wręcz samo zaczęło się nasuwać. Tak jak i w tym projekcie, stosowanie się do tych zasad przyniesie niewątpliwie korzyści również w przyszłych, stając się swoistą etyką pracy i dobrym nawykiem.

Zaimplementowane funkcje są domyślnie zaprojektowane dla pliku w formacie PGM jednak mogą stanowić dobrą bazę przy pracy nad rozszerzeniem także na inne formaty. Operacje przeprowadzane na plikach bardzo przypominają te które obserwujemy przy obróbce zdjęć chociażby za pomocą telefonu co pokazuje jak niewielka droga dzieli kod programu od rzeczywistej, życiowej funkcjonalności

Program został w pełni przetestowany i zabezpieczony przed możliwymi nieprawidłowościami w działaniu.