

Przetwarzanie Obrazów: Sprawozdanie

Damian Ubowski Maciej Tarach

Warszawa, 2019

Spis treści

1	Wstęp	7
1.1	Format obrazu	7
1.1.1	Struktura formatu	7
1.1.2	Przykładowa struktura IFF	8
1.1.3	Instrukcja obsługi programu	8
2	Operacje ujednolicania obrazów	9
2.1	Ujednolicenie obrazów szarych geometryczne	9
2.2	Ujednolicenie obrazów szarych rozdzielczościowe	12
2.3	Ujednolicenie obrazów RGB geometryczne	12
2.4	Ujednolicenie obrazów RGB rozdzielczościowe	12
3	Operacje sumowania arytmetycznego obrazów szarych	13
3.1	Sumowanie (określonej) stałej z obrazem	14
3.2	Sumowanie dwóch obrazów	14
3.3	Mnożenie obrazu przez zadaną liczbę	14
3.4	Mnożenie obrazu przez inny obraz	14
3.5	Mieszanie obrazów z określonym współczynnikiem	14
3.6	Potęgowanie obrazu (z zadaną potęgą)	14
3.7	Dzielenie obrazu przez (zadaną) liczbę	14
3.8	Dzielenie obrazu przez przez inny obraz	14
3.9	Pierwiastkowanie obrazu	14
3.10	Logarytmowanie obrazu	14
4	Operacje sumowania arytmetycznego obrazów barwowych	15
4.1	Sumowanie (określonej) stałej z obrazem	16
4.2	Sumowanie dwóch obrazów	16
4.3	Mnożenie obrazu przez zadaną liczbę	16
4.4	Mnożenie obrazu przez inny obraz	16
4.5	Mieszanie obrazów z określonym współczynnikiem	16
4.6	Potęgowanie obrazu (z zadaną potęgą)	16

4.7	Dzielenie obrazu przez (zadaną) liczbę	16
4.8	Dzielenie obrazu przez przez inny obraz	16
4.9	Pierwiastkowanie obrazu	16
4.10	Logarytmowanie obrazu	16
5	Operacje geometryczne na obrazie	17
5.1	Przemieszczenie obrazu o zadany wektor	17
5.2	Jednorodne skalowanie obrazu	17
5.3	Niejednorodne skalowanie obrazu	17
5.4	Obracanie obrazu o dowolny kąt	17
5.5	Symetrie względem osi układu	17
5.6	Symetrie względem zadanej prostej	17
5.7	Wycinanie fragmentów obrazu	17
5.8	Kopiowanie fragmentów obrazów	17
6	Operacje na histogramie obrazu szarego	19
6.1	Obliczanie histogramu	19
6.2	Przemieszczanie histogramu	19
6.3	Rozciąganie histogramu	19
6.4	Progowanie lokalne	19
6.5	Progowanie globalne	19
7	Operacje na histogramie obrazu barwowego	21
7.1	Obliczanie histogramu	21
7.2	Przemieszczanie histogramu	21
7.3	Rozciąganie histogramu	21
7.4	Progowanie 1-progowe lokalne	21
7.5	Progowanie wielo-progowe lokalne	21
7.6	Progowanie 1-progowe globalne	21
7.7	Progowanie wielo-progowe globalne	21
8	Operacje morfologiczne na obrazach binarnych	23
8.1	Okrawanie (erozja)	23
8.2	Nakładanie (dylatacja)	23
8.3	Otwarcie	23
8.4	Zamknięcie	23
9	Operacje morfologiczne na obrazach szarych	25
9.1	Okrawanie (erozja)	25
9.2	Nakładanie (dylatacja)	25
9.3	Otwarcie	25

9.4	Zamknięcie	25
10	Filtrowanie liniowe i nieliniowe	27
10.1	Filtr dolnoprzepustowy uśredniający	27
10.2	Filtr dolnoprzepustowy Gaussowski	27
10.3	Operator Roberts'a	27
10.4	Operator Prewitt'a	27
10.5	Operator Sobel'a	27
10.6	Filtr kompasowy	27
10.7	Gradient wektora kierunkowego	27
10.8	Filtr medianowy	27
10.9	Filtr maksymalny	27
10.10	Filtr minimalny	27
10.11	Filtr płaskorzeźbowy	27

Rozdział 1

Wstęp

1.1 Format obrazu

Wybrany przez nas formatem obrazów cyfrowych jest DjVu, który jest oparty na zaawansowanej metodzie segmentacji obrazu. Tworzenie pliku DjVu polega na rozdzieleniu dowolnie skomplikowanego obrazu na odrębne warstwy, a następnie poddaniu warst odrębnym optymalizacjom i kompresjom. Format ten stosuje ładowanie progresywne, kodowanie arytmetyczne, oraz kompresję stratną dzięki czemu przy minimalnej ilości przestrzeni dyskowej można delektować się obrazami i dokumentami w wysokiej jakości.

1.1.1 Struktura formatu

Pliki DjVu rozpoczynają się od swojej “Magic number” potwierdzającej rodzaj pliku i mającej wartość `0x41 0x54 0x26 0x54`. Następnie czerpiąc inspirację ze struktury IFF (**I**nter**c**hange **F**ile **F**ormat) plik dzieli się na kawałki (*ang. chunks*) zawierające interesujące nas cenne dane. Takie jak szerokość lub wysokość obrazu, dpi, informacje o kolorach, rozmieszczeniu pikseli, etc. Każdy kawałek składając się z ID typu, długości zawartości i samej zawartości tworzy zwarty format. Identyfikator typu określa rolę w jakiej przyjdzie służyć kawałkowi. Do dyspozycji ma ich całkiem sporo, ale uwzględniając najbardziej przydatne w naszym kontekście to ograniczymy liczbę do:

- * BGjp - warstwa tylna przechowywana przy użyciu kodowania JPEG.
- * BFjp - warstwa przednia w formacie JPEG.
- * INFO - opisuje wysokość, szerokość, rozdzielczość, wersję kodera, oraz flagi wskazujące na obrót obrazu.

1.1.2 Przykładowa struktura IFF

FORM:DJVU [14260]

INFO [10]

Sjbx [13133]

FG44 [181]

BG44 [935]

Powyższa struktura przedstawia dokument składający się z jednej strony, na co wskazuje *FORM:DJVU*, wraz z grafiką. Ten znacznik informuje, że mamy do czynienia z kontenerem o długości 14260 bajtów, który może zawierać inne kawałki dokumentu. Zgodnie z konwencją, po identyfikatorze typu i informacji o długości znajduje się zawartość kawałka. W tym wypadku jak i w każdym innym po *FORM:DJVU* powinno znaleźć się *INFO* z podstawowymi informacjami. Jeśli konwencji i wymagań specyfikacyjnych stało się zadość wtedy czas nastał na jakieś wizualne atrakcje takie jak *Sjbx*, czyli masce wyboru pomiędzy kolorami z warstwy przedniej (*FG44*) i tylnej (*BG44*).

1.1.3 Instrukcja obsługi programu

W celu uruchomienia kodu źródłowego będzie niezbędny:

- * [DjVuLibre](#) ($\geq 3.5.21$)
- * [Python](#) (≥ 2.6 lub $3.X$)
- * [Cython](#) (≥ 0.19 , lub ≥ 0.20 dla Python 3)
- * [pkg-config](#) (POSIX)

Rozdział 2

Operacje ujednolicania obrazów

Ujednolicanie obrazów oznacza sprowadzenie ich do wspólnego gruntu pod względem określonego parametru. W tym wypadku będziemy ujednolicać obrazy pod względem geometrycznym (ilości kolumn i wierszy pikseli) i następnie rozdzielczościowym (wypełnienia pikselami). Sekwencyjność tych operacji jak i one same nie są w stanie spowodować spadku jakości obrazu.

2.1 Ujednolicenie obrazów szarych geometryczne

Algorytm

Algorytm geometrycznego ujednolicenia obrazów ma za zadanie sprowadzić oba obrazy do tej samej liczby pikseli w każdym wierszu i każdej kolumnie.

Kroki algorytmu

1. Porównaj szerokości i wysokości obu obrazów i wybierz największe.
2. Jeśli pierwszy lub drugi obraz mają szerokość lub wysokość mniejszą od największej dostępnej to:
 - (a) Utwórz czarne tło
 - (b) Przenieś z wyśrodkowaniem piksele na czarne tło
3. Jeśli żaden z warunków jest niespełniony to nie rób nic

Rysunek 2.1: Przed uruchomieniem algorytmu (od lewej): obraz 1 (1067x1067, 300dpi), obraz 2 (2133x2133, 300dpi)



Rysunek 2.2: Po uruchomieniu algorytmu (od lewej): obraz 1 (2133x2133, 300dpi), obraz 2 (2133x2133, 300dpi)



Kod źródłowy algorytmu

```
def geometricGray(self):
    max_width = max([self.firstDecoder.width, self.
                     secondDecoder.width])
    max_height = max([self.firstDecoder.height, self.
                      secondDecoder.height])
    print('max size: ' + str(max_width) + 'x' + str(max_height)
          )

    width, height = self.firstDecoder.width, self.firstDecoder.
                     height
    if width < max_width or height < max_height:
        # Create black background
        firstResult = numpy.zeros((max_height, max_width), numpy.
                                  uint8)

        # Copy smaller image to bigger
        startWidthIndex = int(round((max_width - width) / 2))
        startHeightIndex = int(round((max_height - height) / 2))
        pixelsBuffer = self.firstDecoder.getPixels()
        for h in range(0, height):
            for w in range(0, width):
                firstResult[h + startHeightIndex, w + startWidthIndex] =
                    pixelsBuffer[h, w]

        img = Image.fromarray(firstResult, mode='L')
        img.save('Resources/output_1.png')
        print('first image done')

    width, height = self.secondDecoder.width, self.
                     secondDecoder.height
    if width < max_width or height < max_height:
        # Create black background
        secondResult = numpy.zeros((max_height, max_width), numpy.
                                   .uint8)

        # Copy smaller image to bigger
        startWidthIndex = int(round((max_width - width) / 2))
        startHeightIndex = int(round((max_height - height) / 2))
        pixelsBuffer = self.secondDecoder.getPixels()
        for h in range(0, height):
            for w in range(0, width):
                secondResult[h + startHeightIndex, w + startWidthIndex] =
                    pixelsBuffer[h, w]

        img = Image.fromarray(secondResult, mode='L')
        img.save('Resources/output_2.png')
        print('second image done')
    print('unification done')
```

- 2.2 Ujednolicenie obrazów szarych rozdzielczościowe**
- 2.3 Ujednolicenie obrazów RGB geometryczne**
- 2.4 Ujednolicenie obrazów RGB rozdzielczościowe**

Rozdział 3

Operacje sumowania arytmetycznego obrazów szarych

- 3.1 Sumowanie (określonej) stałej z obrazem
- 3.2 Sumowanie dwóch obrazów
- 3.3 Mnożenie obrazu przez zadaną liczbę
- 3.4 Mnożenie obrazu przez inny obraz
- 3.5 Mieszanie obrazów z określonym współczynnikiem
- 3.6 Potęgowanie obrazu (z zadaną potęgą)
- 3.7 Dzielenie obrazu przez (zadaną) liczbę
- 3.8 Dzielenie obrazu przez przez inny obraz
- 3.9 Pierwiastkowanie obrazu
- 3.10 Logarytmowanie obrazu

Rozdział 4

Operacje sumowania arytmetycznego obrazów barwowych

- 4.1 Sumowanie (określonej) stałej z obrazem
- 4.2 Sumowanie dwóch obrazów
- 4.3 Mnożenie obrazu przez zadaną liczbę
- 4.4 Mnożenie obrazu przez inny obraz
- 4.5 Mieszanie obrazów z określonym współczynnikiem
- 4.6 Potęgowanie obrazu (z zadaną potęgą)
- 4.7 Dzielenie obrazu przez (zadaną) liczbę
- 4.8 Dzielenie obrazu przez przez inny obraz
- 4.9 Pierwiastkowanie obrazu
- 4.10 Logarytmowanie obrazu

Rozdział 5

Operacje geometryczne na obrazie

- 5.1 Przemieszczenie obrazu o zadany wektor
- 5.2 Jednorodne skalowanie obrazu
- 5.3 Niejednorodne skalowanie obrazu
- 5.4 Obracanie obrazu o dowolny kąt
- 5.5 Symetrie względem osi układu
- 5.6 Symetrie względem zadanej prostej
- 5.7 Wycinanie fragmentów obrazu
- 5.8 Kopiowanie fragmentów obrazów

Rozdział 6

Operacje na histogramie obrazu szarego

6.1 Obliczanie histogramu

6.2 Przemieszczanie histogramu

6.3 Rozciąganie histogramu

6.4 Progowanie lokalne

6.5 Progowanie globalne

Rozdział 7

Operacje na histogramie obrazu barwowego

- 7.1 Obliczanie histogramu
- 7.2 Przemieszczanie histogramu
- 7.3 Rozciąganie histogramu
- 7.4 Progowanie 1-progowe lokalne
- 7.5 Progowanie wielo-progowe lokalne
- 7.6 Progowanie 1-progowe globalne
- 7.7 Progowanie wielo-progowe globalne

Rozdział 8

Operacje morfologiczne na obrazach binarnych

8.1 Okrawanie (erozja)

8.2 Nakładanie (dylatacja)

8.3 Otwarcie

8.4 Zamknięcie

Rozdział 9

Operacje morfologiczne na obrazach szarych

9.1 Okrawanie (erozja)

9.2 Nakładanie (dylatacja)

9.3 Otwarcie

9.4 Zamknięcie

Rozdział 10

Filtrowanie liniowe i nieliniowe

10.1 Filtr dolnoprzepustowy uśredniający

10.2 Filtr dolnoprzepustowy Gaussowski

10.3 Operator Roberts'a

10.4 Operator Prewitt'a

10.5 Operator Sobel'a

10.6 Filtr Kompasowy

10.7 Gradient wektora kierunkowego

10.8 Filtr medianowy

10.9 Filtr maksymalny

10.10 Filtr minimalny

10.11 Filtr płaskorzeźbowy