

Algorytmy Przetwarzania Obrazów

Obraz i jego histogram

WYKŁAD 1

Dla studiów niestacjonarnych 2019/2020

Dr hab. Anna Korzyńska, prof. IBIB PAN

Przypomnienie

– Obraz to ...

Dwuwymiarowa funkcja intensywności wielkości pomiarowej $f(x, y)$, najczęściej wartość określa intensywność światła/luminancję lub składowych koloru w miejscu o współrzędnych x, y czyli na ograniczonej spójnej powierzchni dwuwymiarowej.



Obraz niesie informację, o odwzorowywanej rzeczywistości/wizji autora (najczęściej 3D) umieszczonej na ograniczonej dwuwymiarowej przestrzeni (2D)

2

Przypomnienie

Obraz cyfrowy to informacja podwójnie dyskretna:

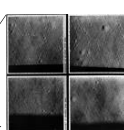
- odwzorowuje 3D na **dyskretną** i ograniczoną przestrzeń 2D



Księżyc



Akwizycja Ranger 7



31 czerwca 1964 o 13:09 UT (9:09 AM EDT)
Vidicon B
Rozmiar w pikselach:
1150x1150
Rozdzielczość
r. poziomów
szarości: 64
Pole widzenia
8.4x8.4
Kąt 78.3
Dystans
2445.97 km

- informacja o intensywności cechy pomiarowej jest **dyskretna** (skwantowana)

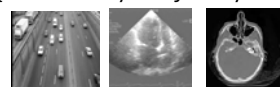
http://nssdc.gsfc.nasa.gov/imcat/html/object_pager/ra7_b001.html

3

Obraz cyfrowy w naukach technicznych i przyrodniczych to:

Zwarty, jednorodny i przestrzennie uporządkowany zbiór sygnałów:

- związanych z cechą/cechami pomiarowymi, na bazie których tworzymy obraz (natężenie fali elektromagnetycznej, akustycznej, wielkości nie falowe np. czas relaksacji)
- dostosowanych do materialnego nośnika obrazu (papieru, kliszy, dyskiety, pamięci dyskowej itp.)
- niosących informację o odwzorowywanej rzeczywistości



4

Podstawowa definicja obowiązująca na APO

Obraz to dyskretna dwuwymiarowa funkcja $f(x, y)$ określona na ograniczonym fragmencie płaszczyzny, której wartości f to **intensywność** (jasność, kolor) w tym punkcie (x, y) .

- Dla obrazów szaroodcieniowych wartość f to luminancja jest skalar
- Dla obrazów kolorowych wartość f to wektor o trzech składowych, określający kolor w wybranej przestrzeni koloru $f = \{f_r, f_g, f_b\}$
- Dla obrazów wielomodalnych i multispektralnych wartość f to wielowymiarowy wektor określający różne dane pomiarowe.

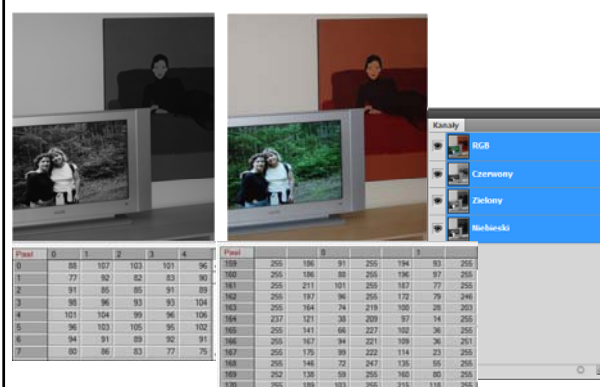
$f \in [L_{min}, L_{max}]$ - skala szarości/intensywności składowej pojedynczego kanału obrazu

$L_{min} = 0$, minimalna intensywność odpowiada czerni

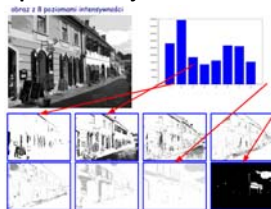
L_{max} = maksymalna intensywność odpowiada bieli

M - liczba poziomów szarości $M = L_{max} - L_{min} + 1$ $M = 2^k$

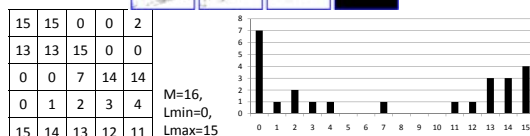
5



Histogram - rozkład ilości wystąpień pikseli o zadanych poziomach jasności w obrazie



M=8,
Lmin=0,
Lmax=7



Histogram definicja

Histogram to wykres słupkowy przedstawiający ilość pikseli o każdej potencjalnej wartości występującej w obrazie.

- Statystyka odzwierciedlająca rozkład jasności punktów w obrazie.
- Pewna estymata rozkładu jasności oryginalnego obrazu analogowego i rzeczywistości.



Tablica LUT

- Wykorzystywana do:
 - Przekodowania jasności;
 - Definiowania operacji punktowych (UOP)
 - Zapisu histogramu i działania operacji na histogramie

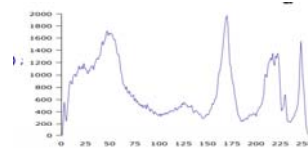
piksel	f(piksel)
0	13
1	34
2	234
...	...
254	11
255	255

13	14	5	13	6	14	8	6	9	8	11	12	7	10	7	7
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Histogram skumulacyjny

h - histogram

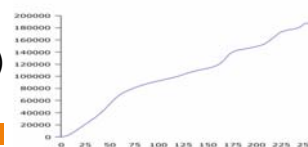
3 7 0 10 5 6 0 5



H - histogram skumulowany

- $H(0) = h(0)$
- $H(n) = H(n-1) + h(n)$

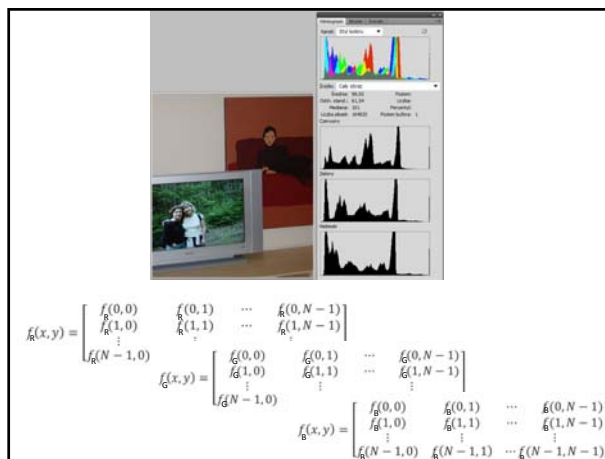
3 10 10 20 25 31 31 36



Obraz cyfrowy: $N \times N$ pikseli.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix}$$

11



$$f_h(x,y) = \begin{bmatrix} f_h(0,0) & f_h(0,1) & \dots & f_h(0,N-1) \\ f_h(1,0) & f_h(1,1) & \dots & f_h(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f_h(N-1,0) & f_h(N-1,1) & \dots & f_h(N-1,N-1) \end{bmatrix}$$

$$f_g(x,y) = \begin{bmatrix} f_g(0,0) & f_g(0,1) & \dots & f_g(0,N-1) \\ f_g(1,0) & f_g(1,1) & \dots & f_g(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f_g(N-1,0) & f_g(N-1,1) & \dots & f_g(N-1,N-1) \end{bmatrix}$$

$$f_b(x,y) = \begin{bmatrix} f_b(0,0) & f_b(0,1) & \dots & f_b(0,N-1) \\ f_b(1,0) & f_b(1,1) & \dots & f_b(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f_b(N-1,0) & f_b(N-1,1) & \dots & f_b(N-1,N-1) \end{bmatrix}$$

Algorytm przeszukiwanie obrazu

- Do operacji punktowych F
- Do operacji sąsiedztwa

```

For i=0 to X do:
  Begin.
    For j=0 to Y do
      Begin.
        fnew(i,j):= F(f(i,j))
      End.
    End.
  End.

For i=2 to X-2 do:
  Begin.
    For j=2 to X-2 do
      Begin.
        fnew(i,j):= F(f(i-1, j-1), f(i-1, j),
          f(i-1, j+1), f(i, j-1), f(i, j), f(i, j+1),
          f(i+1, j-1), f(i+1, j), f(i+1, j+1))
      End.
    End.
  End.

```

Algorytm tworzenia (wyznaczania wartości) histogramu

Algorytm wyznaczania histogramu jest prosty: analizuje każdy piksel obrazu i zlicza piksele o każdym możliwym poziomie jasności.

Oznaczenia: $f(P)$ jest wartością elementu P w przedziale $[0, L]$. h jest tablicą histogramu.

Wstaw do tablicy $h(Z)$ ($0 \leq Z \leq L$) wartości zero.

For wszystkich elementów P obrazu do:

Begin.

Zwiększ $h(f(P))$ o 1.

End.

Koniec algorytmu.

Implementacja jest jednak bardziej skomplikowana ze względu na różnorodność typów obrazów: monochromatyczne i kolorowe, wielo-spektralne, itp..

Generowanie wykresu słupkowego z tablicy LUT

Wyszukaj maksymalną wartość zapisaną w tablicy h
Przygotuj ramkę histogramu i współrzędne początku wykresu (x, y)
For wszystkich elementów tablicy h (LUT) do:

Begin.

$h()$ unormuj wartość


Narysuj odcinek o długości odpowiadającej wartości unormowanej i punkcie zaczepienia w współrzędnych (x, y)

Przesuń współrzędne (x, y)

End.

Koniec algorytmu.

Implementacja w OpenCV(C++)



```

#include <cv.h>
#include <highgui.h>

//przeglądaj komentarz (lib, "cv210")
//przeglądaj komentarz (lib, "highgui210")

IplImage* image= 0;
IplImage* imgHistogram = 0;
IplImage* gray= 0;

CvHistogram* hist;

int main()
{
    image = cvLoadImage("XXXXX.jpg",1);

    //określenie rozmiaru histogramu (liczby koszy)
    int size[] = {256};

    //wartość kosza i jego znormalizowane wartości
    float val;
    int norm;

    //zakres jasności
    float range[] = { 0, 256 };
    float* ranges[] = { range };

    //wartość minimalna i maksymalna histogramu
    float min_val = 0, max_val = 0;

    //konwersja wejściowego obrazu kolorowego do 8-bitowego w skali szarości
    gray = cvCreateImage( cvGetSize(image), 8, 1 );
    cvCvtColor( image, gray, CV_BGR2GRAY );

    //tworzenie okien do pokazania rezultatu

```

```

//tworzenie okien do pokazania rezultatu
cvNamedWindow( "original",1);
cvNamedWindow( "gray",1);
cvNamedWindow( "histogram",1);

//obrazy z których zostanie obliczony histogram
IplImage* images[] = { gray };

//przeglądanie postępu histogramu
hist = cvCreateHist( 1, size, CV_HIST_ARRAY, ranges,1);
//obliczenie histogramu i obrazu
cvCalcHist( images, hist, 0, NULL);
//znalezienie minimalnej i maksymalnej wartości histo
cvGetMinMaxHistValue( hist, &min_val, &max_val);

//tworzenie 8-bitowego obrazu do pokazania histogramu
imgHistogram = cvCreateImage( cvSize(256, 50),8,1);
//wypełnienie go białym kolorem
cvRectangle( imgHistogram, cvPoint(0,0), cvPoint(256,50),
CV_RGB(255,255,255),CV_FILLED);

//rysowanie histogramu i wartości znormalizowanego od 0 do 50
for(int i=0; i < 256; i++)
{
    val = cvQueryHistValue_1D( hist, i);
    norm = cvRound( val*50/max_val);
    cvLine( imgHistogram, cvPoint(i,50), cvPoint(i,50-norm), CV_RGB(0,0,0));
}

//pokazanie okien z rezultatem
cvShowImage( "original", image );
cvShowImage( "gray", gray );
cvShowImage( "histogram", imgHistogram );
//zapis obrazu histogramu do pliku
cvSaveImage( "histogram.jpg",gray);
cvWaitKey();

return 0;
}

```

Operacje na histogramie

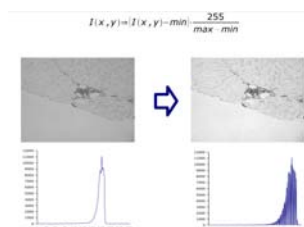
Obrazy bardzo często zawierają elementy, które są trudne do zauważenia głównie dlatego obiekty są mało zróżnicowane w stosunku do otoczenia.

Operacje na histogramach:

1. Rozciąganie
2. Wyrównywanie
3. Wyrównanie typu equalizacja

Rozciąganie histogramu

Obrazy w których nieefektywnie wykorzystujemy pełną dynamikę odcieni i barw dostępną w danym zakresie



- Efekt: wyostrzenie

Wyrównanie histogramu

- Metoda ta polega na wyrównaniu poziomów szarości, tak aby histogram był płaski (czyli wszystkie stopnie szarości miały dokładnie taką samą liczbę punktów). W praktyce nie jest to możliwe, więc otrzymany histogram często jest postrzępiony.
- Do takiego wyrównania histogramu wykorzystujemy tablicę LUT.
- W pierwszym kroku musimy stworzyć dystrybucję empiryczną (odpowiednik histogramu skumulowanego)

$$D[n] = (h_0 + h_1 + \dots + h_n) / \text{sum}$$

gdzie:

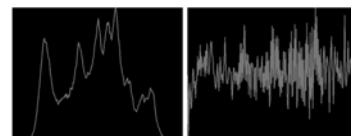
h_n - to ilość punktów na obrazie o n -tym poziomie szarości,
 sum - to liczba wszystkich punktów obrazu.

- W drugim kroku możemy stworzyć tablicę LUT:

$$\text{LUT}[j] = ((D[j] - D_0) / (1 - D_0)) * (M - 1)$$

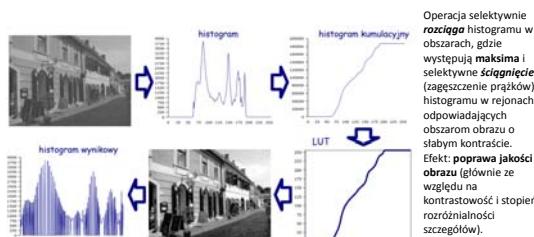
gdzie

D_0 - to pierwsza niezerowa wartość dystrybucji obrazu źródłowego,
 M - to liczba możliwych wartości jasności obrazu (zwykle 256).



Wyrównywanie typu equalizacja

- Wyrównanie histogramu (*ang. Equalization*) dąży do ujednolicenia rozkładu pikseli we wszystkich fragmentach osi poziomach jasności.
- Splaszczanie - zrównanie ilości wystąpień jasności i ich rozłożenie na osi dostępnych poziomów jasności.
- Wykorzystanie całki z histogramu (dystrybucji) do ustalania jasności docelowych (wyjściowych).
- Proste wyrównanie histogramu rzadko sprawia, że obraz wygląda naprawdę dobrze.



Algorytm wyrównywania histogramu

Oznaczenia: H jest tablicą histogramu, H_{nowe} jest całką histogramu. Z oznacza stare poziomy, a R nowe. Każde Z jest odwzorowane w przedziale $[\text{left}(Z), \text{right}(Z)]$.

Czytaj obraz, wyznacz wartości jego histogramu i przechowaj je w tablicy H . Niech H_{nowe} będzie wartością średnią.

Podstaw $R=0$ i $H_{\text{nowe}}=0$.

For $Z=0$ do L do:

Begin.

Podstaw $\text{left}(Z)=R$ i dodaj $H(Z)$ do H_{nowe} .

While H_{nowe} jest większe od H_{nowe} do:

Begin.

Odejmij H_{nowe} od H_{nowe} i zwiększ R .

End.

Podstaw $\text{right}(Z)=R$ i określ wartość $\text{new}(Z)$ zgodnie ze stosowaną regułą. Dla reguły 1 podstaw zamiast $\text{new}(Z)$ wartość średnią z $\text{left}(Z)$ i $\text{right}(Z)$. Dla reguły 2 podstaw zamiast $\text{new}(Z)$ różnicę $\text{right}(Z) - \text{left}(Z)$. Dla reguły 3 $\text{new}(Z)$ jest niezdefiniowane.

End.

For wszystkich elementów P obrazu do:

Begin.

If $\text{left}(P)$ jest równe $\text{right}(P)$, then podstaw $\text{left}(P)$ jako nową wartość elementu P .

Else:

Jeśli stosujesz regułę 1, podstaw wartość $\text{new}(P)$.

Jeśli stosujesz regułę 2, wybierz losowo punkt z przedziału $[\text{left}(P), \text{right}(P)]$, dodaj jego wartość do $\text{left}(P)$ i przyjmij wynik jako nową wartość P . Jeśli stosujesz regułę 3, oblicz średnią wartość sąsiadów P . Jeśli przekracza ona $\text{right}(Z)$, użyj $\text{right}(Z)$ jako nowej wartości. Jeśli jest mniejsza od $\text{left}(Z)$, użyj $\text{left}(Z)$ jako nowej wartości. W pozostałych przypadkach użyj średniej jako nowej wartości.

End.

10. Koniec algorytmu.

Niech w obrazie pierwotnym $[p(i, j)]$ poziomy szarości przyjmują dyskretne wartości obliczane ze wzoru:

gdzie:

L - liczba poziomów szarości

n_k - liczba wystąpień pikseli o r_k - tej jasności.

$p_k(r_k)$ - prawdopodobieństwo wystąpienia k - tego poziomu szarości

N - całkowita ilość pikseli w obrazie.

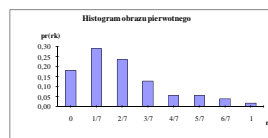
Metodę wyrównywania histogramu opisuje wzór:

gdzie:

$T(r_k)$ - funkcja transformująca

Przykład

Załóżmy obraz o rozmiarze $N \times N = 100 \times 100$, $M=8$.



r_k	n_k	$p(r_k)$
$r_0 = 0$	1800	0.18
$r_1 = 1/7$	2900	0.29
$r_2 = 2/7$	2350	0.24
$r_3 = 3/7$	1281	0.13
$r_4 = 4/7$	555	0.06
$r_5 = 5/7$	554	0.06
$r_6 = 6/7$	400	0.04
$r_7 = 1$	160	0.02

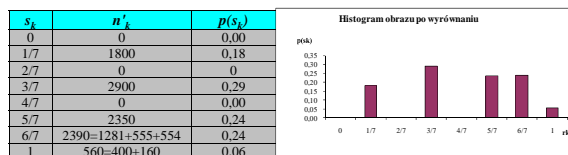
$$s_0 = T(r_0) =$$

$$s_1 = T(r_1) =$$

$$s_2 = 0,71; s_3 = 0,83; s_4 = 0,89; s_5 = 0,94; s_6 = 0,98; s_7 = 1$$

Po zaokrągleniu do możliwych poziomów szarości:

$$s_0 \approx 1/7; s_1 \approx 3/7; s_2 \approx 5/7; s_3 \approx 6/7; s_4 \approx 6/7; s_5 \approx 6/7; s_6 \approx 1; s_7 \approx 1;$$



- **OpenCV** (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then by firm Itseez (which was later acquired by Intel).
- The library is **cross-platform** and free for use under the **open-source BSD license**.



Biblioteka OpenCV

Uwagi ogólne dotyczące algorytmów

- Dyskusja nad algorytmami obejmuje m.in. zagadnienie złożoności obliczeniowej.
- Podając algorytm próbujemy określić czas i pamięć potrzebne do jego wykonania.
- Typowy błąd: mylenie złożoności obliczeniowej i programowej.
- Generalnie, długość programu realizującego algorytm ma mało wspólnego z szybkością wykonania, a nawet z wymaganą wielkością pamięci.
- Jeśli jakaś relacja istnieje, to jest ona wręcz odwrotna.
- Algorytmy „złożone” są zwykle szybsze niż „proste”. Np. program dla FFT (nierekurencyjny) jest dłuższy i bardziej złożony niż program realizujący wzór sumy dla transformaty. Jednak wykonuje się znacznie szybciej. Podobnych przykładów dostarczają algorytmy sortowania.
- Często atrakcyjniejsze wydaje się użycie rekurencyjnej formy algorytmu, jako dużo krótszej niż nierekurencyjna, a liczba operacji w obu formach jest taka sama. W takich przypadkach należy pamiętać o kosztach wywołań rekurencyjnych, potrzebie przechowywania wartości rejestrów w pamięci itp. Jeśli liczba wywołań jest mała w porównaniu z liczbą innych operacji, to ich koszt może być opłacalny z powodu prostoty programu. W innych przypadkach algorytmy w formie nierekurencyjnej dają programy wydajniejsze.
- O ile **prostota programowania** może wydawać się atrakcyjna programiście, który jest ograniczony czasem i planuje uruchomienie programu z niewielką ilością danych, o tyle jest **szkodliwa** w przypadkach zastosowań użytkowych przy dużych zbiorach danych.

• Materiał:

- M.Doros, Przetwarzanie obrazów, skrypt WSISIZ
- Materiały wykładowe z POB na UBIKu
- (katalog **doros\dor17_18\POBZ\Wykl**)
- T.Pavlidis, Grafika i Przetwarzanie Obrazów, WNT Warszawa 1987.
- I.Pitas, Digital image processing, algorithms and applications, John Wiley & Sons, Inc. 2000, pp. 162-166 (w katalogu ... \APOZ\Materiały na UBIKu).
- Plik na UBIKu: .. \dor18_19\APOZ\Wykl\Wykl1Przyklad

Omówienie tematów projektów

• Materiał w katalogach na UBIKu:

... \ APOZ \ Materiały

... \ APOZ \ Projekty \ TematyProjektow

... \ APOZ \ Projekty \ Przyklady