

Histogram obrazu. Wyrównywanie histogramu.

Cel:

- zapoznanie z pojęciem histogramu obrazu (w odcieniach szarości i kolorze),
- zapoznanie z metodami modyfikacji histogramu (rozciąganie, wyrównywanie, dopasowywanie).
- dodatkowo: zaawansowane metody wyrównywania histogramu: BBHE i DSIHE

Histogram

- histogramem obrazu nazywamy wykres słupkowy zdefiniowany następującymi zależnościami:

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} p(i, (x, y)) \quad , \text{gdzie} \quad p(i) = \begin{cases} 1 & \text{gdy } f(x, y) = i; \\ 0 & \text{gdy } f(x, y) \neq i; \end{cases}$$

- inaczej mówić histogram zawiera informacje na temat tego ile pikseli o danym poziomie jasności występuje na obrazie (w przypadku obrazu w odcieniach szarości)
- często wykorzystuje się tzw. znormalizowaną postać histogramu - wszystkie wartości $h(i)$ są dzielone przez liczbę pikseli na obrazie. Otrzymana w ten sposób wielkość to gęstość prawdopodobieństwa wystąpienia na obrazie pikseli o odcieniu "i".
- histogram można zdefiniować dla obrazów kolorowych. Otrzymujemy wtedy 3 histogramy - po jednym dla danej składowej: R, G, B (lub HSV, YcrCb, itp.) lub histogram trójwymiarowy
- histogram jest bardzo użyteczny w przetwarzaniu obrazów. Wykorzystywany jest przy binaryzacji (szerzej na jednym z kolejnych laboratoriów) oraz do oceny jakości (dynamiki, kontrastu) obrazu. W idealnym przypadku wszystkie poziomy jasności w obrazie powinny być wykorzystane (i to najlepiej w miarę jednolicie) - obrazowo mówiąc histogram powinien rozciągać się od 0 - 255 (obraz w skali szarości)
- w przypadku gdy wykorzystujemy jedynie fragment dostępnego zakresu (wąski histogram) lub histogram nie jest jednolity (występują dominujące grupy pikseli) obraz ma dość słaby kontrast. Cechę tę można poprawić stosując tzw. rozciąganie albo wyrównywanie histogramu (*histogram equalization*).

A. Histogram dla obrazów w odcieniach szarości:

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Nazwij go i zapisz. Wykonaj polecenia `clearvars, close all, clc`. Wczytaj obrazy: "lena1.bmp", "lena2.bmp", "lena3.bmp" i "lena4.bmp" (wcześniej ściągnij archiwum ze strony www.roma2.unipr.it/~luciano/lenas.htm lub histogram trójwymiarowy katalogu). Wykorzystując polecenie `subplot` stwórz wykres składający się z 8 obrazków - 4 x "lena" + histogram. Do wyznaczenia i wyświetlenia histogramu wykorzystaj polecenie `imhist`. Uwaga należy zapoznać się z opisem funkcji w helpie, a następnie wykorzystać wersję "imhist(I, n)", gdzie $n = 256$ (liczba poziomów szarości).

2. Przeanalizuj (dokładnie) związek histogramu z jasnością i ostrością obrazu (tu rozumianą jako subiektywne odczucie).
3. Wczytaj obraz "hist1.bmp". Wyświetl go. Zwróć uwagę na ilość widocznych szczegółów. Wyświetl histogram rozpatrywanego obrazu (na wspólnym rysunku z obrazem).

4. Najprostszą metodą poprawienia jakości obrazu jest tzw. rozciągnięcie histogramu - przeskalowanie wartości pikseli w obrazie aby wykorzystać cały dostępny zakres (0-255). Operację taką realizuje funkcja `imadjust`. Zapoznaj się z jej dokumentacją oraz wykonaj rozciągnięcie histogramu dla rozpatrywanego obrazu. Rezultat operacji wyświetl (obraz i histogram). Czy ilość "widocznych" szczegółów uległa zmianie?

5. Ciekawostka: funkcja `imadjust` umożliwia przeprowadzenie korekcji jasności obrazu - w wersji z parametrem *gamma*, gdzie definiuje się kształt przekształcania - szczegóły dokumentacja.

6. Bardziej zaawansowaną metodą jest tzw. wyrównywanie histogramu (HE). Idea jest następująca: z punktu widzenia lepszego wykorzystania dostępnych poziomów jasności pożądaną by było rozciągnięcie "szczytów" histogramu, a skompresowanie "dolin" tak aby taka sama liczba pikseli reprezentowana była przez każdy z dostępnych poziomów szarości (warto zwrócić uwagę, że takie przekształcenie powoduje częściową utratę informacji o szczegółach w obszarach "dolin") - inaczej mówiąc dążymy do sytuacji aby histogram był jednostajny. Operacją, która pozwala wykonać wyrównywanie histogramu jest przekształcenie LUT z funkcją przejścia w postaci histogramu skumulowanego danego obrazu.

7. **Histogram skumulowany** to funkcja obliczona na podstawie histogramu w następujący sposób. Pierwszy element to liczba pikseli o odcieniu '0', a kolejne wartości to liczba pikseli o odcieniu 'n' + suma pikseli o odcieniach 0 - n-1. Jeżeli histogram jest w postaci znormalizowanej (gęstość rozkładu prawdopodobieństwa) to histogram skumulowany stanowi dystrybucję rozkładu prawdopodobieństwa.

8. Wyznacz histogram skumulowany dla rozpatrywanego obrazka. Funkcja `imhist` ma możliwość zwrócenia wektorów opisujących histogram (zliczenia oraz odpowiadające poziomy jasności) `[H,x] = imhist(...)`. Do wyliczenia histogramu skumulowanego wykorzystaj funkcję `cumsum` (jak? - help). Histogram skumulowany (nazwany np. *C*) wyświetl razem z histogramem zwykłym. Uwagi:
 - wywołanie typu `[H,x] = imhist(...)` powoduje, że histogram nie jest wyświetlany. Trzeba albo pozostawić oryginalne wywołanie `imhist(...)` albo osobno wyświetlić histogram np. `bar(x,H)`;
 - histogram skumulowany należy przeskalować tak aby się "zmieścił" na wykresie. W tym celu należy wyliczyć maksimum z histogramu `max(H)` i histogramu skumulowanego `max(C)`, wyliczyć iloraz $k = \max(C) / \max(H)$; i podzielić wszystkie elementy *C* przez *k* ($C2 = C/k$);
 - aby wyświetlić histogram skumulowany wspólnie z histogramem należy wykorzystać polecenie `hold on`; - które powoduje, że rysowanie odbywa się na poprzednim wykresie oraz `plot(x,H)` i `plot(x,C2)`; - rysowanie wykresów.

9. [P] Wykorzystując kod stworzony w poprzednim punkcie wyświetl histogram i histogram skumulowany dla obrazka po rozciągnięciu histogramu. Dotychczasowe rezultaty zaprezentuj prowadzącemu.

10. W kolejnym kroku należy zaimplementować klasyczny algorytm wyrównywania histogramu (HE). Wykorzystujemy obliczony histogram skumulowany. Należy go przeskalować w taki sposób aby na jego podstawie zrealizować przekształcenie LUT - czyli do zakresu 0 - 255. Jak? - zadanie do samodzielnej realizacji (przypomnienie LUT to funkcja `intlut`). Konwersja do formatu `uint8`: `lut = uint8(lut)`;

11. Na kolejnym rysunku wyświetl obrazek po przekształceniu, jego histogram oraz histogram skumulowany. Co szczególnego można powiedzieć o jego histogramie i histogramie skumulowanym?

12. W pakiecie Matlab dostępne są dwie funkcje umożliwiające manipulację histogramem:
 - `histeq` - wyrównywanie histogramu
 - `adaphisteq` - adaptacyjne wyrównywanie histogramu algorytmem CLAHE (Contrast Limited Adaptive Histogram Equalization)

13. Zapoznaj się z ich opisem w pomocy, przetestuj ich działanie na rozpatrywanym obrazku. Zwróć uwagę, że funkcja `histeq(obraz,256)` daje identyczne rezultaty jak metoda zaimplementowana w p. 11.
14. Wykorzystywany dotychczas obraz jest obrazem sztucznym. W kolejnym etapie należy przetestować operacje na histogramie (rozciąganie, wyrównywanie (HE) i adaptacyjne wyrównywanie CLAHE) dla obrazów rzeczywistych: "hist2.bmp", "hist3.bmp", "hist4.jpg" - cztery wykresy: obraz oryginalny, rozciąganie, wyrównywanie HE oraz wyrównywanie CLAHE.

15. [P] Wyniki zaprezentuj prowadzącemu.

B. Dopasowywanie histogramu (*Histogram Matching*)

1. Klasyczne rozszerzanie histogramu nie zawsze jest najlepszym rozwiązaniem. Czasami lepiej jest wykorzystać technikę dopasowywania histogramu - podajemy wtedy jaki kształt ma mieć docelowy histogram przetwarzanego obrazu.

2. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Nazwij go i zapisz. Wykonaj polecenia `clearvars, close all, clc`. Wczytaj obraz "phobos.bmp" - jest to zdjęcie księżycy Marsa - Phobos - wykonane przez sondę Mars Global Surveyor. Wykonaj klasyczne wyrównywanie histogramu (HE) - wykorzystaj funkcję `histeq`. Czy wynik operacji jest satysfakcjonujący?

3. Funkcja `histeq` ma możliwość podania jako argument oczekiwanego histogramu (*hgram* - patrz help). Histogram ten trzeba dobrać "ręcznie" (zatem metoda ta nie nadaje się do systemów pracujących automatycznie). Wczytaj przygotowany oczekiwany histogram - `load histogramZadany`; i wyświetl go. Wykonaj dopasowywanie histogramu. Wyświetl obraz wynikowy i jego histogram.

4. [P] Na obrazie "phobos.bmp" spróbuj wykonać rozciąganie histogramu - `imadjust` oraz adaptacyjne wyrównywanie histogramu CLAHE. Zastanów się chwilę nad uzyskanymi rezultatami - na którym obrazie widoczne jest najwięcej szczegółów. Wyniki zaprezentuj prowadzącemu.

C. Histogram dla obrazów kolorowych i jego wyrównywanie

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Nazwij go i zapisz. Wykonaj polecenia `clearvars, close all, clc`. Wczytaj obraz "lenaRGB.bmp". Wyświetl wczytany obraz. Wydziel poszczególne składowe R, G, B. Wyświetl ich histogramy. Wskazówki:
 - wydzielenie składowych zostało zaprezentowane na ćwiczeniu wprowadzającym. Przypomnienie: `lenaR = lena(:, :, 1)`;
 - wykorzystaj polecenie `subplot`

2. Wyrównanie histogramu dla obrazu kolorowego jest zagadnieniem złożonym. Postępowanie zależy od celu jaki chcemy osiągnąć. Na początku wykonamy wyrównywanie w przestrzeni barw RGB - każdą składową osobno:
 - wykorzystując wydzielone wcześniej składowe oraz funkcję `histeq(...,256)` wykonaj wyrównywanie. Połącz otrzymane składowe w nowy obraz (np. w ten sposób:


```
obrazEQ(:,:,1) = Re;
obrazEQ(:,:,2) = Ge;
obrazEQ(:,:,3) = Be;
```

 i wyświetl go. Jaka jest zasadnicza wada takiego podejścia?

3. Przekształć obraz "lena" do przestrzeni HSV. Wyświetl histogramy poszczególnych składowych. Manipulacji dokonujemy na składowej odpowiadającej za jasność - czyli V. Wykonujemy wyrównywanie histogramu dla tej składowej (najlepiej poleceniem `histeq(...,256)`). W dokonaj podmiiany składowej V i wyświetl rezultat operacji (uprzednio przeprowadź konwersję HSV->RGB, ponieważ funkcja `imshow` niepoprawnie wyświetla obrazy w HSV).
4. Podmień obraz "lenaRGB.bmp" na "jezioro.jpg" zaobserwuj rezultaty działania rozpatrywanych metod.
5. [P] Wyniki zaprezentuj prowadzącemu.

Ciekawostki:

- histogram występuje w popularnych programach graficznych (Corel Photo-Paint, Adobe Photoshop oraz Gimp)
- w wymienionych programach występuje również opcja wyrównywania histogramu:
 - **Gimp** - działa tak jak zrealizowana w podpunkcie 2 (osobno kanały RGB)
 - **Photoshop** - algorytm daje wyniki inne niż omawiane metody - brak możliwości konfiguracji
 - **Photo-Paint** - istnieje możliwość wyboru opcji: wyrównywanie w RGB, każdy kanał z osobną obróbką zachowania kolorów

Zadanie dodatkowe: BBHE i DSIHE

Klasyczne wyrównywanie histogramu HE ma jedną zasadniczą wadę: po wykonaniu operacji jasność obrazu ulega zmianie (dalo się to zaobserwować podczas przeprowadzonych eksperymentów, a jak nie to należy uruchomić skrypt z sekcji A i zwrócić na to uwagę) - średnia jasność dąży do środkowego poziomu szarości. Dlatego klasyczne HE ma ograniczone zastosowanie.

Powstało sporo metod, które eliminują to niekorzystne zjawisko. Najprostsze z nich polegają na dekompozycji obrazu wejsciowego na dwa podobrazy (wg. pewnego kryterium) i wykonania operacji HE dla tych podobrazów.

Dwie znane z literatury metody to:

- **BBHE** - Bi-Histogram Equalization
- **DSIHE** - Dualistic Sub-Image Histogram Equalization

W metodzie BBHE za kryterium podziału przyjmujemy się średnią jasność w obrazie, a w DSIHE obraz dzieli się na dwa podobrazy o takiej samej ilości pikseli (jaśniejszych i ciemniejszych).

Zadanie: zaimplementować wybraną metodę: BBHE lub DSIHE (ew. obie)

1. Utwórz nowy m-plik, wykonaj polecenia `clearvars, close all, clc`. Wczytaj obraz "jet.bmp" i wylicz jego histogram.
2. W kolejnym kroku należy zaimplementować próg podziału obrazu na dwa podobrazy (*lm*).
 - dla BBHE wykorzystujemy funkcję `mean` (trzeba uzyskać jedną liczbę, czyli obliczyć średnią ze średniej z obrazu, lub zamienić macierz obrazu na wektor `obraz(:)` oraz zaokrąglić wynik `round`).
 - dla DSIHE można wykorzystać histogram skumulowany:
 - obliczamy histogram skumulowany (`cumsum`)
 - określamy rozmiar obrazka `[X Y] = size(obraz)`;
 - na podstawie histogramu skumulowanego da się wyznaczyć próg podziału np. `[value lm] = min(abs(C - (X*Y/2)))`; - szukamy poziomu jasności który znajduje się "w połowie" histogramu skumulowanego

3. Następnie należy podzielić histogram oryginalnego obrazu na dwa histogramy H1 i H2, wykonać normalizację histogramów H1 i H2 (czyli podzielić histogramy przez sumę ich elementów: $H1 = H1 / \sum(H1)$;) wyznaczyć histogramy skumulowane dla H1 i H2 (*C1* i *C2*).

4. Na podstawie *C1* i *C2* tworzymy przekształcenie LUT. Idea jest następująca: należy tak przeskalować *C1* i *C2* aby uzyskać jednorodne przekształcenie. Wartość *C1* wystarczy pomnożyć przez próg podziału *lm*, wartość *C2* należy przeskalować do przedziału ($lm+1$; 255)


```
C1n = (lm)*C1;
C2n = lm+1 + (255-lm+1)*C2;
lut = [C1n; C2n];
```
5. Ostatecznie należy wykonać operację `lut(intlut)` i wyświetlić wynik wyrównywania histogramu. Porównać wynik operacji BBHE lub DSIHE z klasycznym HE.
6. [P] Rezultaty pracy zaprezentuj prowadzącemu.