

Binaryzacja

Cel:

- zapoznanie z segmentacją poprzez binaryzację,
- zapoznanie z binaryzacją na podstawie histogramu (globalną),
- zapoznanie z metodą Ots'u, Kittler'a i Kapur'a,
- zapoznanie z binaryzacją lokalną (metoda Sauvola),
- zapoznanie z binaryzacją dwu-progową,
- zapoznanie z binaryzacją wykorzystującą średnią ruchomą.

Binaryzacja

Jednym z najważniejszych etapów podczas analizy obrazów jest segmentacja - podział obrazu na rejony według pewnego kryterium (jasność, kolor, tekstura). Najprostszą (i też najczęściej wykorzystywaną) metodą segmentacji jest **binaryzacja**. Do jej głównych zalet zalicza się: intuicyjność, prostotę, łatwość implementacji i szybkość wykonywania. Jest ona etapem wielu algorytmów analizy obrazów. Pozwala na znaczną redukcję informacji w obrazie (np. z zakresu 0-255 do 0-1).

Binaryzacja najczęściej realizowana jest poprzez progowanie. Na przykład: dla obrazu w odcieniach szarości ustala się próg na poziomie 'k'. Wszystkie piksele o wartości większej od 'k' zostają uznane za obiekty istotne, a pozostałe za tło. Podejście daje się zastosować wtedy, gdy obiekty mają istotnie różną jasność od otaczającego je tła.

Binaryzacja na podstawie histogramu

Podstawą określania progu binaryzacji jest zazwyczaj analiza histogramu ("ręczna" lub automatyczna).

1. Otwórz program **Matlab**. Ustal ścieżkę **Current Directory** na swój własny katalog na dysku D. Utwórz nowy m-plik (**File->New M-File**) lub (**New->Script**). Na początku wykonaj polecenia `clc; close all; clearvars;`

2. Wczytaj obraz "coins.png" (jest to obraz "wbudowany" w pakiet Matlab). Wyświetl go. Wyznacz jego histogram i wyświetl go (funkcja `imhist`) - najlepiej na wspólnym wykresie.

3. Wizualna analiza histogramu pozwala zauważyć dwa maksima - jedno odpowiadające poziomowi jasności tła (które jest względnie jednolite - ciemnoszare) i drugie odpowiadające monetom. Na podstawie histogramu wyznacz próg i wykonaj binaryzację:

- wykorzystaj funkcję `im2bw`, uwaga próg należy podać z zakresu 0-1, zatem wybrany na podstawie histogramu próg (np. 150) należy podzielić przez 255,
- wynik binaryzacji wyświetl,

- spróbuj dobrać najlepszy próg binaryzacji i zapisz go. Uwaga. Jako „kryterium jakości” przyjmij kształty monet. Idealnie powinny to być wypełnione koła.

4. Na łatwość przeprowadzenia binaryzacji największy wpływ mają dwa czynniki:
 - szum,
 - niejednorodność oświetlenia.

Wczytaj obraz "figura.png". Wyświetl jego histogram (wykorzystaj kod z punktu 2). Czy łatwo jest wyznaczyć próg binaryzacji. Wczytaj obraz "figura2.png" (dodany szum Gaussowski o średniej 0 i odchyleniu standardowym 10). Wyświetl jego histogram. Jak dodanie szumu wpłynęło na histogram? Czy dalej można łatwo wyznaczyć próg binaryzacji? Wczytaj obraz "figura3.png" (dodany szum Gaussowski o średniej 0 i odchyleniu standardowym 50). Wyświetl jego histogram. Czy da się na podstawie histogramu wyznaczyć próg binaryzacji ?

Wczytaj obraz "figura4.png" (dodany gradient oświetlenia - symulacja oświetlenia niejednorodnego). Wyświetl histogram. Czy da się wyznaczyć próg binaryzacji?

5. W automatycznym systemie analizy obrazów (działanie bez nadzoru operatora) konieczne jest zastosowanie metody binaryzacji, która automatycznie wyznacza próg. Jedną z części wykorzystywanych jest metoda zaproponowana przez Ots'u. Dokonuje ona podziału histogramu na dwa klasy, tak aby wariancja pomiędzy tymi klasami była minimalna. W Matlab'ie dostępna jest implementacja metody Otsu (`graythresh`). Funkcja zwraca próg. Wykonaj binaryzację obrazu "coins.png" metodą Otsu. Porównaj wyniki z "ręcznie" wyznaczonym progiem.

6. Metod automatycznego wyznaczania progu binaryzacji (globalnego) jest bardzo dużo - w artykule "*Survey over image thresholding techniques and quantitative performance evaluation*" autorzy M. Sezgin i B.Sankur testują 40 różnych metod. Rozważmy jeszcze dwie metody:

- Kittler - minimalizacja funkcji błędu zbudowanej z dystrybucji prawdopodobieństwa i odchylenia standardowego, koncepcyjnie podobna do Otsu (funkcja `clusterKittler`)
- Yen - maksymalizacja entropii obiektów i tła (funkcja `entropyYen`)

7. Wykonaj binaryzację obrazu "coins.png" metodami Kittler i Yen. Argumentem dla obu funkcji jest obraz, a wynikiem próg binaryzacji (z zakresu 0 - 255). Wyniki wszystkich binaryzacji najwygodniej wyświetlić na wspólnym rysunku (`subplot`). Rysunek powinien składać się z 6 wykresów - obraz oryginalny, histogram oraz 4 obrazy zbinaryzowane. Wykresy opisz (`title`). W tytule, oprócz nazwy metody, wyświetl próg - funkcja `num2str` - szczegóły i przykład w opisie funkcji `title`.

8. Zmień obraz testowy na "rice.png" (wbudowany w Matlab), "tekst.bmp", "obiekty.bmp", "katalog.bmp". Przetestuj algorytm automatycznej binaryzacji. Za każdym razem spróbuj ręcznie dobrać najlepszy próg binaryzacji, zanotuj go i porównaj z uzyskanymi automatycznie.

9. [P] Zaprezentuj wyniki prowadzącemu.

Binaryzacja lokalna

Analiza wyników binaryzacji dla obrazów "rice.png" i "katalog.bmp" pokazuje, że globalna binaryzacja nie najlepiej działa dla obrazów o niejednorodnym oświetleniu. Dla obu obrazów trudno również wyznaczyć odpowiedni próg "ręcznie".

Metodą, która pozwala poprawić wyniki binaryzacji, jest binaryzacja lokalna (niekiedy zwana adaptacyjną). Polega ona na wyznaczeniu progu osobno dla każdego piksela na podstawie jego otoczenia.

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `clc; close all; clearvars;`

2. Dla uproszczenia zakładamy, że obraz ma rozmiar 256x256 pikseli. Przyjmijmy okno o rozmiarze 15 pikseli.

3. Najprostszą wersją binaryzacji lokalnej zakłada, że próg binaryzacji dla danego okna to średnia z pikseli w tym oknie. Wykorzystując funkcję `meanLT(i, j, W2, I, X, Y)`; (parametry: `ij` - współrzędne aktualnego piksela, `W2` - połowa rozmiaru okna, `I` - obrazek w odcieniach szarości, `X, Y` - rozmiar obrazka). Wykonaj binaryzację lokalną.

4. Wczytaj obraz "rice.png". Rozmiar obrazka (parametry `X, Y`) można uzyskać za pomocą funkcji `[X Y] = size(I)`; Wykonaj kopię wczytanego obrazka (`obrazBW = obraz;`) - wykorzystana zostanie przy binaryzacji.

5. Podstawą algorytmu są dwie pętle `for` iterujące po pikselach obrazka (tablice w Matlabie indeksowane są od 1):

```
for i = 1:X
    for j = 1:Y
        ....
    end
end
```

6. Wewnątrz pętli należy dla każdego piksela wyznaczyć średnią (za pomocą funkcji `meanLT`) i na jej podstawie dokonać binaryzacji - za pomocą instrukcji `if else` (przypisać odpowiednio wartości 255 i 0 do kopii obrazka).

7. Obraz oryginalny oraz zbinaryzowany wyświetl.

8. [P] Zaobserwuj rezultaty działania metody dla obrazów "rice.png" i "katalog.bmp". Poeksperymentuj z rozmiarem okna (bez przesady). Jaka jest podstawowa wada zaimplementowanej metody ? (pomijając złożoność obliczeniową). Wyniki zaprezentuj prowadzącemu.

9. Jakość działania binaryzacji lokalnej można poprawić wyznaczając próg za pomocą metody Sauvola'. Wykorzystuje ona, oprócz średniej, informację o odchyleniu standardowym w danym oknie.

$$T = \text{średnia} \cdot [1 \pm k \left(\frac{\text{odchStd}}{R} - 1 \right)]$$

; gdzie `k` i `R` to parametry (`R` zwykle 128, a `k` na początek przyjmij 0.15).

10. Zaimplementuj algorytm Sauvola - wykorzystaj funkcję `stddevLT(i, j, W2, I, m, X, Y)`; (parametry: `ij` - współrzędne aktualnego piksela, `W2` - połowa rozmiaru okna, `I` - obrazek w odcieniach szarości, `m` - średnia w danym oknie, `X, Y` - rozmiar obrazka).

11. Zastanów się nad znaczeniem symbolu \pm we wzorze na próg. Kiedy należy zastosować znak $+$, a kiedy $-$.

12. [P] Porównaj jakość binaryzacji lokalnej metodą Sauvola' i z progiem na podstawie średniej. Poeksperymentuj z rozmiarem okna i parametrem `k` (dla obrazów "rice.png" i "katalog.bmp"). Wyniki zaprezentuj prowadzącemu.

Binaryzacja dwuprogową.

Binaryzację można przeprowadzić wykorzystując dwa progi - wybieramy w ten sposób przedział jasności (piksele w nim zawarte klasyfikujemy jako obiekty).

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `clc; close all; clearvars;`

2. Wczytaj obraz "bart.bmp". Wyświetl go, wyznacz i wyświetl jego histogram. Oceń, który fragment histogramu odpowiada kolorowi skóry Barta Simpsona. (Podpowiedź: Okno Figure posiada narzędzie Tools->Data Cursor, które pozwala podglądać wartość poszczególnych pikseli na obrazkach - dla odcieni szarości wartość Index)

3. Przeprowadź segmentację na podstawie koloru skóry (binaryzację dwuprogową). Wykorzystaj składnię:


```
obrazBW = obraz > progDolny & obraz < progGorny;
obrazBW = uint8 (obrazBW, []);
```

4. [P] Wynik wyświetl i zaprezentuj prowadzącemu.

***Zadanie dodatkowe - binaryzacja z wykorzystaniem średniej ruchomej.

Metoda ta jest niejako szczególnym przypadkiem binaryzacji lokalnej. Średnia ruchoma wyznaczana jest dla kolejnych linii (np. podczas skanowania obrazu). Jeżeli przez z_{k+1} oznaczmy jasność podczas $k+1$ kroku skanowania, to średnią ruchomą można wyznaczyć jako:

$$m(k+1) = \frac{1}{n} \cdot \sum_{i=k+1-n}^k z_i = m(k) + \frac{1}{n} \cdot (z_{k+1} - z_{k-n})$$

gdzie `n` oznacza liczbę punktów z których wylicza się średnią. Ponadto przyjmujemy, że $m(1) = \frac{z_1}{n}$ (dla uproszczenia obliczeń). Jako próg binaryzacji wykorzystujemy wartość $T = a \cdot m$, gdzie `a` oznacza współczynnik, a `m` wyznaczoną średnią ruchomą.

Zadanie: zaimplementować opisaną metodę:

1. Utwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `close all; clearvars; clc;`

2. Wczytaj obraz "tekstRecznz.png". Wyświetl go, wyznacz i wyświetl jego histogram. Spróbuj wykonać binaryzację - ręcznie oraz metodą Otsu.

3. Zaimplementuj binaryzację za pomocą średniej ruchomej. Wskazówki:

- ustal `N = 20`
- `a = 0.5`
- należy wykonać iterację po wszystkich pikselach w obrazie (pętla `for`)
- wyznaczanie średniej ruchomej wymaga zapamiętania elementów, które wchodzi w skład średniej - dobrze do tego nadaje się kolejka FIFO zaimplementowana na tablicy jednowymiarowej (tablica + dwa wskaźniki na początek i koniec kolejki).
- obliczenie średniej implementujemy zgodnie z podanym wzorem, dla uproszczenia nie przejmujemy się pierwszymi 20 pikselami (inicjujemy kolejkę wartościami 0).
- w każdej iteracji jeden element opuszcza kolejkę (piksel z_{k-n}) oraz jeden do kolejki dochodzi (piksel z_{k+1})

4. [P]. Wyświetl wyniki binaryzacji. Pokaż implementację i wyniki prowadzącemu.