

upel.agh.edu.pl

SW: Instrukcja - Współczynniki kształtu

7 — 9 minut

Tomasz Kryjak, Piotr Pawlik

PRZETWARZANIE OBRAZÓW CYFROWYCH

Rozpoznawanie obiektów za pomocą współczynników kształtu

Cel:

- zapoznanie z rozpoznawaniem obiektów za pomocą współczynników kształtu.

1. Indeksacja wbudowana w IPT Matlab'a

W Matlabie dostępna jest funkcja : `bwlabel`, która pozwala wykonać indeksację. Przy indeksacji ważne jest ustalenie sąsiedztwa: możliwości są dwie - sąsiedztwo 4-elementowe (za sąsiadów uznajemy piksele "stykające się" brzegami) i 8-elementowe (za sąsiada uznajemy piksele "stykające się" wierzchołkami - w ten sposób działa zaimplementowany wcześniej algorytm).

1. W nowym m-pliku wczytaj obraz "indeks2.bmp". Wyświetl go.

2. Przeprowadź indeksację za pomocą funkcji `bwlabel` z sąsiedztwem 4 i 8 elementowym. Wyniki porównaj (np. na wspólnym wykresie). Uwaga - w celu poprawnego wyświetlania obrazów poindeksowanych należy wykorzystać parametr `[]` funkcji `imshow`.

2. Współczynniki kształtu

Współczynniki kształtu są pewnymi parametrami liczbowymi opisującymi kształt obiektu. Pozwala to na użycie ich do automatycznego rozpoznawania obiektów. Teoretycznie współczynniki kształtu (dobre) powinny być niezależne od zmiany położenia obiektu, jego orientacji i wielkości.

W praktyce okazuje się jednak, że wartości współczynników zmieniają się w pewnym zakresie dla tych samych obiektów.

Wynika z tego konieczność uwzględnienia pewnej tolerancji wartości współczynników kształtu wykorzystywanych w praktycznych zastosowaniach analizy i rozpoznawania obrazu.

W ćwiczeniu wykorzystane zostaną następujące współczynniki:

- **Compactness** - stosunek pola obiektu do pola najmniejszego prostokąta w jakim się obiekt mieści
- **Rmin/Rmax** - pierwiastek stosunku minimalnej odległości konturu od środka ciężkości do maksymalnej odległości konturu od środka ciężkości
- **Blair - Bliss** -
$$S \sqrt{\frac{2\pi \sum_i r_i^2}{\sum_i r_i^2}}$$
 gdzie S - pole powierzchni obiektu, r_i - odległość piksela obrazka od środka ciężkości, i - numer piksela obiektu.

- **Haralick** -
$$\frac{(\sum_i d_i)^2}{2n \sum_i d_i^2 - 1} \sqrt{\frac{(\sum_i d_i)^2}{n \sum_i d_i^2 - 1}}$$
 gdzie d_i -

odległość piksela konturu od środka ciężkości, n - liczba pikseli konturu.

- **M7** - niezmiennik momentowy - $M7 = N_{20} \cdot N_{02} - N_{21}^2$ $M7 = N_{20} \cdot N_{02} - N_{11}^2$, N - moment znormalizowany: $N_{pq} = \frac{M_{pq}}{m_{00}^w}$ gdzie $w = p+q+1$ $w = \frac{p+q}{2} + 1$ M - moment centralny, m - moment zwykły.

Pierwszym etapem zadania będzie wybranie współczynników kształtu i przedziałów ich wartości - takich, które pozwolą na wykrycie konkretnych kształtów. Do wybrania jest kwadrat lub krzyżyk (koło jest za łatwe).

1. W m-pliku z punktu 1 (Indeksacja wbudowana w Matlabie) zmień wczytywany obraz na "kształty.bmp", a sąsiedztwo ustal na 8 elementowe.

2. Wykorzystując dostarczoną funkcję - obliczWspolczynniki.m wyznacz współczynniki kształtu dla wszystkich kształtów z obrazu "kształty.bmp". Funkcja jako argument przyjmuje obraz po indeksacji, a zwraca macierz dwuwymiarową - wiersze oznaczają konkretne kształty (kolejność taka jak numery indeksów przypisane podczas indeksacji), a kolumny kolejne współczynniki - Compactness, Rmin/Rmax, Blair - Bliss, Haralick, M7. Aby dobrze przyporządkować wiersze figurom warto użyć następującego kodu, umieszczając go PO funkcji imshow. Figury zostaną "podpisane" numerami, które uzyskały w indeksacji:

```
r = regionprops(labeled,'Centroid');
for i=1:length(r)
    text(r(i).Centroid(1),r(i).Centroid(2),['\color{magenta}',num2str(i)]);
```

end

gdzie, labeled -obraz po indeksacji.

3. Przypisanie numerów do figur ułatwi analizę wyniku funkcji

obliczWspolczynniki - numery figur to numery wierszy w matlabowym podglądzie zmiennej przechowującej ten wynik.

4. W analizowanym wyniku proszę wyszukać taki współczynnik (współczynniki), które pozwolą odróżnić kwadraty od krzyżyków - i ustalić pewien przedział wartości współczynnika. Dodatkowo trzeba zapewnić, żeby przypadkowo nie zostały wykryte koła.

5. Zaimplementuj logikę, która pozwoli wyeliminować z obrazka niepożądane obiekty. Wykonaj iterację po całym poindeksowanym obrazku. Odczytaj aktualną wartość piksela, jeżeli jest różna od zera to wykorzystując macierz zwracaną przez funkcję obliczWspolczynniki wyzeruj dany piksel lub nie. Przykład:

```
piksel = labeled(i,j);
```

```
if piksel ~= 0 && ~(wsp(piksel,1) > 0.73 && wsp(piksel,1) < 0.86)
```

```
    labeled(i,j) = 0;
```

```
end
```

Zapis ten oznacza, że jeżeli pierwszy współczynnik (Compactness) nie zawiera się w przedziale (0.73:0.86) to piksel należy wyzerować. W efekcie na obrazie labeled powinny pozostać tylko koła.

6. Wczytaj obraz "kszałtyReal.bmp". Wyświetl go. Wykorzystując poznane metody przetwarzania obrazów doprowadź obraz do postaci binarnej (usuń ew zakłócenia itp.). Następnie rozpoznaj na obrazie wybrany wcześniej kształt (postępowanie analogiczne jak w punktach 2-5).

7. Uwagi:

- należy z obrazu wyeliminować małe grupki pikseli - tak aby nie zostały poindeksowane
- obiektów powinno być 13.
- może się okazać, że zajdzie potrzeba modyfikacji warunków na współczynniki albo wręcz wprowadzenie dodatkowych współczynników.

8. Pokaż rezultaty prowadzącemu.

9. Klasycznym sposobem rozpoznawania za pomocą wektorów cech (a takim wektorem może być zbiór rozważanych przez nas współczynników) jest użycie klasyfikatora. Jednym z najprostszych jest klasyfikator knn. Spróbujemy go użyć, aby ponownie rozpoznać na obrazie wybrany kształt. Obraz z pliku "kształty.bmp" posłuży do 'trenowania' klasyfikatora - czyli wskazania jakie klasy rozpatrujemy ('krzyz';'kolo';'kwadrat') i jak wyglądają przedstawiciele tych klas, natomiast figury z "kształtyReal.bmp" będą przyporządkowywane do jednej z w/w klas (czyli 'rozpoznawane').

10. Aby 'wytrenować' klasyfikator posłużymy się funkcją **fitcknn**. Otrzymuje ona macierz wektorów cech oraz macierz z nazwami klas - każdy wektor cech jest przyporządkowany do jakiejś klasy. W naszym przypadku macierzą z wektorami cech jest po prostu wynik funkcji obliczWspolczynniki. Dla obrazu "kształty.bmp" tym wektorom powinna towarzyszyć macierz klas:

```
klasy={'kolo';'krzyz';'kwadrat';'kwadrat';'krzyz';'kolo';'kwadrat'; 'krzyz'}
```

Rezultatem funkcji **fitcknn** jest klasyfikator, który zostanie użyty w rozpoznawaniu. Dokona go funkcja **predict** - otrzymuje ona jako parametry klasyfikator i macierz wektorów cech do rozpoznania (czyli znów wynik obliczWspolczynniki tym razem dla drugiego obrazu). Rezultat to lista klas do których przyporządkowane zostały

kolejne wektory. Dla lepszego zorientowania się w tych wynikach można użyć funkcji **celldisp** - poda ona numery wektorów, które są zgodne z numerami wypisanymi na wyświetlanych figurach (jak w punkcie 5).

11. Wynik zapewne nie będzie zgodny z Państwa oczekiwaniem. Problemem jest to, że jedna z cech jest dominująca pod względem wartości (jest dużo większa od pozostałych) Wektory cech powinny być znormalizowane lub cechy powinny mieć porównywalną 'siłę'. W naszym wypadku wystarczające będzie sprowadzenie tej cechy do wartości 0-1 - proszę ją podzielić przez 10 w oby macierzach wektorów cech i dokonać ponownego rozpoznawania.

12. Pokaż rezultaty prowadzącemu.