Przetwarzanie i Kompresja Obrazów. Segmentacja

Aleksander Denisiuk (denisjuk@pja.edu.pl)
Polsko-Japońska Akademia Technik Komputerowych
Wydział Informatyki w Gdańsku
ul. Brzegi 55, 80-045 Gdańsk

5 czerwca 2016

Najnowsza wersja tego dokumentu dostępna jest pod adresem
http://users.pja.edu.pl/~denisjuk/

Zagadnienie Segmentacji

- Oddzielić obiekt od tła
  - $f(i,j)$ — funkcja obrazu w pikselu (i,j)
  - Ω — obszar spójny
  - $\partial\Omega$ — granica obszaru (ma sąsiednie piksele, należące do obszaru i sąsiednie piksele, nie należące do obszaru).
  - $e(x,y)$ — krawędź w pikselu (i,j).
    - wektor dwuwymiarowy, określany przez gradient:
      - norma: $M(i,j) = |\nabla f(i,j)| = \|\nabla f(i,j)\|$
      - kierunek: $\theta(i,j) = \arctan \frac{\partial f}{\partial y}(i,j) / \frac{\partial f}{\partial x}(i,j) - \frac{\pi}{2}$

Progowanie

- Podział na dwa segmenty
- Dany jest próg τ

$$g(i,j) = \begin{cases} 1, & f(i,j) \geqslant \tau, \\ 0, & f(i,j) < \tau. \end{cases}$$

- Może być na odwrót

$$g(i,j) = \begin{cases} 1, & f(i,j) < \tau, \\ 0, & f(i,j) \geqslant \tau. \end{cases}$$

- $f(i,j)$ może być kolorem, luminancją, gradientem, etc

- Nie zmienia obiekt
- Tło ustawia jako czarne

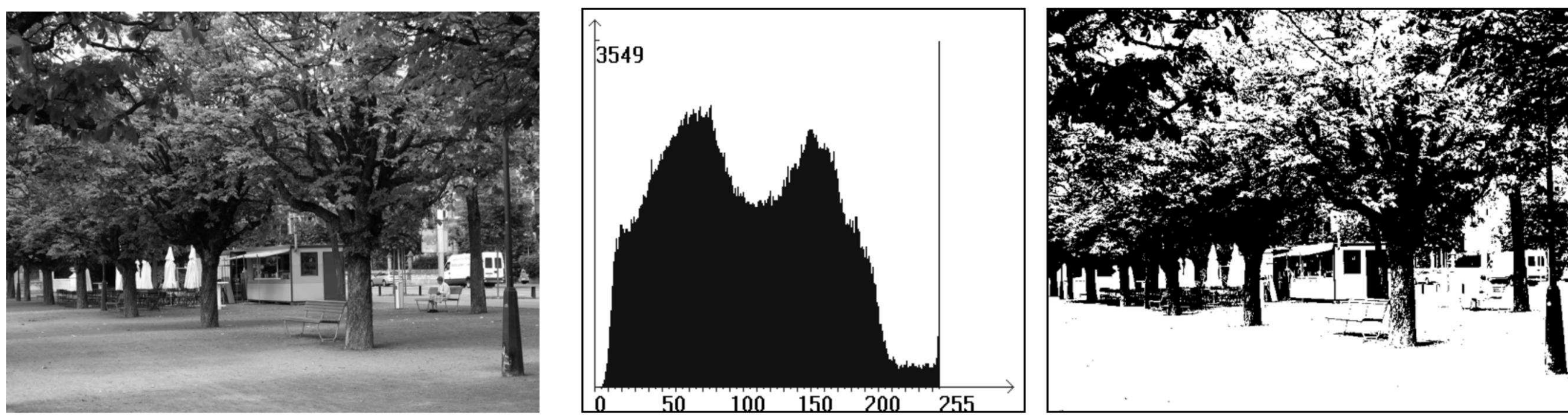
$$g(i,j) = \begin{cases} f(i,j), & f(i,j) \geqslant \tau, \\ 0, & f(i,j) < \tau. \end{cases}$$

- Dany jest przedział progowania U
- Przykładowo $[\tau_0, \tau_1]$

$$g(i,j) = \begin{cases} 1, & f(i,j) \in U, \\ 0, & f(i,j) \notin U. \end{cases}$$

Progowanie modalne

- Jeżeli obiekt ma luminancję, która różni się od luminancji tła, histogram będzie mieć dwa maksima (bimodalny histogram)
- Za próg wybiera się minimum lokalne między maksimami



(a) (b) (c)

- Jeżeli luminancja nie jest równomierna, histogram może mieć więcej niż dwa maksima
  - obraz dzieli się na fragmenty i do każdego fragmenty stosuje się powyższą metodę

Progowanie iteracyjne

- Za początkowy próg przyjmuje się średnia wartość
- Obraz dzieli się na dwa obszary
- Za kolejny próg przyjmuje pół-suma średnich obiektu i tła
- Warunek zakończenia iteracji: stabilizacja

(a) (b)

Wejście: $f(i,j)$ — $m \times n$ obraz, $\varepsilon > 0$ — dokładność
Wyjście: τ — próg optymalny z dokładnością ε

$\tau \leftarrow \frac{1}{n+m} \sum f(i,j)$
repeat
$\Omega_1 \leftarrow \{(i,j) | f(i,j) \geqslant \tau\}, \quad \Omega_2 \leftarrow \{(i,j) | f(i,j) < \tau\}$
$\mu_1 \leftarrow \frac{1}{|\Omega_1|} \sum_{\Omega_1} f(i,j), \quad \mu_2 \leftarrow \frac{1}{|\Omega_2|} \sum_{\Omega_2} f(i,j)$
$\tau_1 \leftarrow \frac{\mu_1 + \mu_2}{2}$
$\Delta\tau \leftarrow |\tau - \tau_1|$
$\tau \leftarrow \tau_1$
until $\Delta\tau > \varepsilon$

Segmentacja konturowa

- Operatory gradientowe Robertsa, Prewitta oraz Sobela pozwalają na wzmocnienie krawędzi, ale
  - Szerokośćkrawędzi powiększa się
  - Szum może spowodować artefakty
- Algorytm Canny’ego:
  1. rozmycie Gaussa
  2. obliczenie normy i kierunku gradientu w każdym pikselu
  3. detekcja pikseli granicznych
  4. progowanie z histerezą

- splot z filtrem $h = \frac{1}{1115}\begin{pmatrix} 1 & 4 & 7 & 10 & 7 & 4 & 1 \\ 4 & 12 & 26 & 33 & 26 & 12 & 4 \\ 7 & 26 & 55 & 71 & 55 & 26 & 7 \\ 10 & 33 & 71 & 91 & 71 & 33 & 10 \\ 7 & 26 & 55 & 71 & 55 & 26 & 7 \\ 4 & 12 & 26 & 33 & 26 & 12 & 4 \\ 1 & 4 & 7 & 10 & 7 & 4 & 1 \end{pmatrix}$
- Obraz wygładzony $S = h * f$

- $G(i,j) = \sqrt{\left(\frac{\partial S}{\partial x}(i,j)\right)^2 + \left(\frac{\partial S}{\partial y}(i,j)\right)^2}$
- $\theta(i,j) = \arctan\left(\frac{\partial S}{\partial y}(i,j) / \frac{\partial S}{\partial x}(i,j)\right)$
- gdzie
  - $\frac{\partial S}{\partial x}(i,j) = \frac{1}{2}\left(S(i+1,j) - S(i,j) + S(i+1,j+1) - S(i,j+1)\right)$
  - $\frac{\partial S}{\partial y}(i,j) = \frac{1}{2}\left(S(i,j+1) - S(i,j) + S(i+1,j+1) - S(i+1,j)\right)$

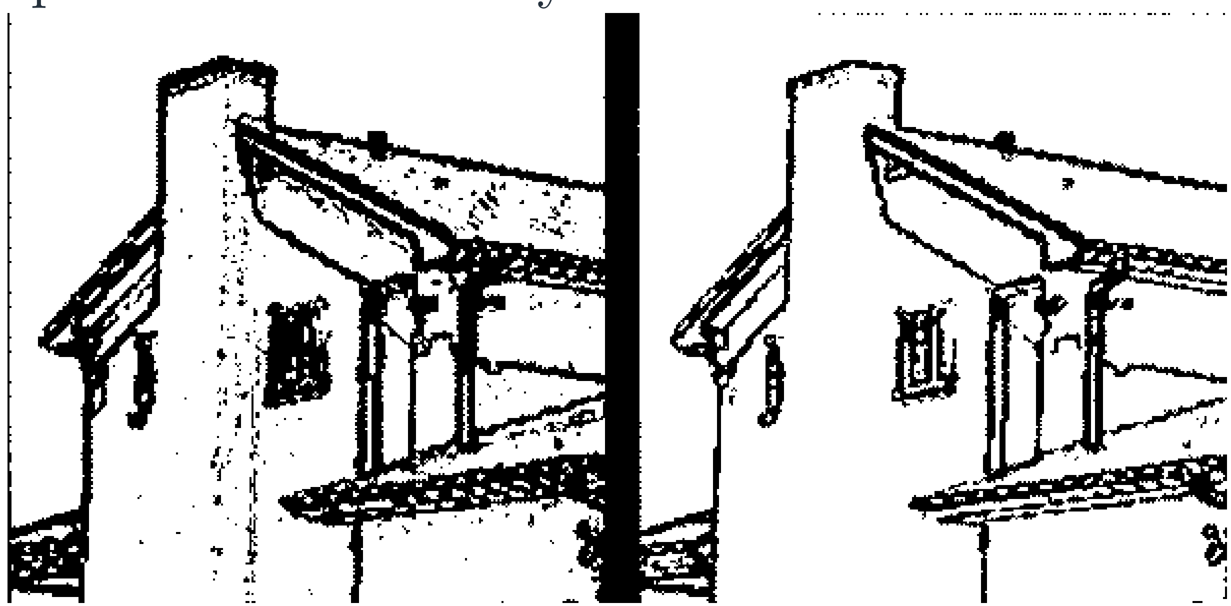
- Piksel (i, j) jest granicznym, jeżeli $G(i, j)$ jest lokalnym maksimum w kierunku gragientowym
  - niech $p_1 = (i_1, j_1)$ oraz $p_2 = (i_2, j_2)$ będą dwoma sąsiednimi pikselami w kierunku gradientu
  - obraz wynikowy określa się w sposób następujący:
    $\phi(i, j) =$
    $$\begin{cases} G(i, j), & \text{jeżeli } G(i, j) \geqslant G(i_1, j_1) \& G(i, j) \geqslant G(i_2, j_2), \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

Piksele, sąsiednie w kierunku gradientu

- Wybiera się z 8-otoczenia piksela $p = (i, j)$
- Kwantyzacja kierunku gragientu $\theta(i, j)$

1. jeżeli $-\frac{\pi}{8} < \theta(i,j) \leqslant \frac{\pi}{8}$, to $\theta = 0$ oraz $(i_1, j_1) = (i, j-1)$, $(i_2, j_2) = (i, j+1)$
2. jeżeli $\frac{\pi}{8} < \theta(i,j) \leqslant \frac{3}{8}\pi$, to $\theta = \frac{\pi}{4}$ oraz $(i_1, j_1) = (i+1, j-1)$, $(i_2, j_2) = (i-1, j+1)$
3. jeżeli $-\frac{3}{8}\pi < \theta(i,j) \leqslant -\frac{\pi}{8}$, to $\theta = -\frac{\pi}{4}$ oraz $(i_1, j_1) = (i-1, j-1)$, $(i_2, j_2) = (i+1, j+1)$
4. jeżeli $\frac{3}{8}\pi < \theta(i,j) < \frac{3}{2}\pi$ lub $-\frac{\pi}{2} < \theta(i,j) < -\frac{3}{8}\pi$, to $\theta = \frac{\pi}{2}$ oraz $(i_1, j_1) = (i-1, j)$, $(i_2, j_2) = (i+1, j)$

- Szum oraz tekstury (powtarzające się fragmenty) mogą spowodować artefakty

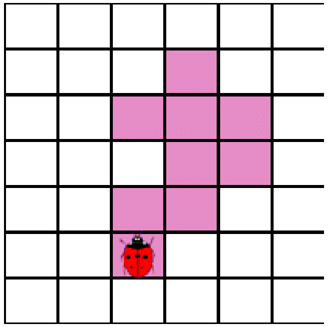


- Przeprowadza się progowanie o dwóch progach $\tau_1 < \tau_2$
  - jeżeli $\phi(i, j) > \tau_2$, to piksel zaliczamy do granicznych
  - każdy sąsiedni piksel, który ma $\phi(i, j) > \tau_1$ też zaliczamy do pikseli granicznych
  - pozostałe piksele wyłączamy z granicznych

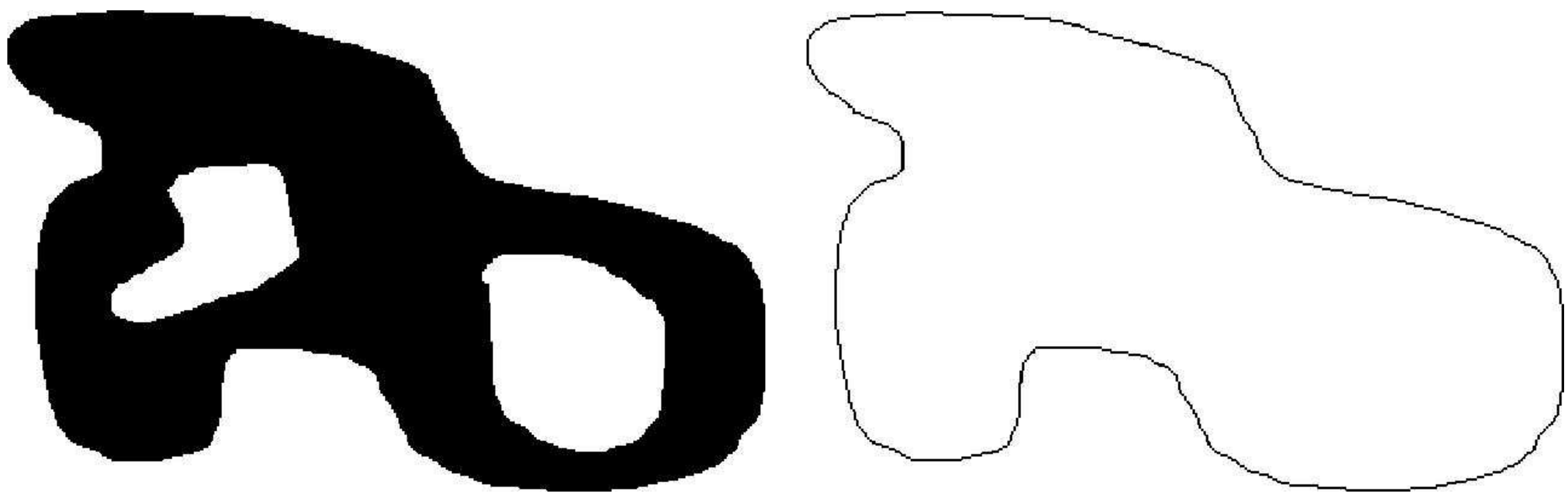
Wejście: $\phi(i,j)$ — $m \times n$ obraz, $\tau_1 < \tau_2$ — dwa progi,
$E(i,j) = 0$ — wypełniony zerami $m \times n$ obraz,
$\Omega(i,j)$ przedstawia 8-otoczenie piksela
Wyjście: $E(i,j)$ zawiera obraz segmentowany

```
repeat
  continue ← false
  for all (i, j) ∈ [0 ... m − 1] × [0 ... n − 1] do
    if φ(i, j) > τ2 then
      E(i, j) ← 1; φ(i, j) ← 0; continue ← true
    else if φ(i, j) > τ1 then
      if Ω(i, j) zawiera piksel (k, l), taki że E(k, l) == 1
      then
        E(i, j) ← 1; φ(i, j) ← 0; continue ← true
      end if
    end if
  end for
until continue
```

- Binaryzacja: obiekt jest czarnym, tło białym
- Wybór piertwszego piksela na krawędzi
  - zacząć od lewego dolnego rogu
  - pierwszy napotkany czarny piksel jest na krawędzi
- Obejście krawędzi (wzdłuż czarnych pikseli)
  - Po każdym kroku spróbować przejść w lewo i w dół



- Koniec iteracji: osiągnięcie początkowego piksela po raz drugi
- Następny obszar: czarny piksel, sąsiedni do białego, który nie należy go granicy poprzedich obiektów



Segmentacja obszarowa

- Poszukiwanie ziarna — piksela, należącego do obiektu
- Wyszukiwanie pikseli z 4-sąsiedztwa, podobnych do wyznaczonego
  - funkcja podobieństwa: luminancja, kolor, tekstura, etc
  - kryterium podobieństwa pikseli p_1 i p_2:

$$s(p_1, p_2) = \begin{cases} 1, & \text{jeżeli } |f(p_1) - f(p_2)| < \tau, \\ 0, & \text{jeżeli } |f(p_1) - f(p_2)| \geqslant \tau, \end{cases}$$

gdzie τ jest progiem podobieństwa

Zagadnienie Segmentacji
Progowanie
Segmentacja konturowa
Segmentacja obszarowa
Rozrost Obszarów
Łączenie Obszarów (SRM)

Wejście: punkt (i, j) zawiera się w obszarze
Wyjście: wyznaczony jest cały obszar

```
Stos S ← ∅
zaznacz (i, j); S ← S ∪ (i, j)
while S ≠ ∅ do
    S ← S \ (i, j)
    if niezaznaczony podobny piksel (i − 1, j) then
        zaznacz (i − 1, j); S ← S ∪ (i − 1, j)
    end if
    if niezaznaczony podobny piksel (i, j − 1) then
        zaznacz (i, j − 1); S ← S ∪ (i, j − 1)
    end if
    if niezaznaczony podobny piksel (i + 1, j) then
        zaznacz (i + 1, j); Stos S ← S ∪ (i + 1, j)
    end if
    if niezaznaczony podobny piksel (i, j + 1) then
        zaznacz (i, j + 1); S ← S ∪ (i, j + 1)
    end if
end while
```

Łączenie Obszarów (SRM)

- Obraz dzieli się na małe obszary
- Małe sąsiadujące obszary łączy się na podstawie średniej (lub innej statystycznej wielkości) funkcji podobieństwa
  - po połączeniu małych obszarów odświeża się statystyki
- Koniec obliczeń: nie można połączyć żadnych obszarów

Wejście: Dany jest $m \times n$ obraz, $f(i,j)$ jest funkcją podobieństwa, τ jest progiem podobieństwa, piksele są niezaznaczone, $k = 0$
Wyjście: Obraz podzielony na k obszarów

```
for all (i, j) ∈ [0 ... m − 1] × [0 ... n − 1] do
  if piksel (i, j) nie jest zaznaczony then
    zaznacz (i, j);
    for all u ∈ Ω(i, j) do
      if |f(u) − f(i, j)| < τ then
        if piksel u jest zaznaczony then
          zalicz (i, j) do obszaru piksela u
        else
          zalicz (i, j) oraz u do obszaru R_k
          k ← k + 1; zaznacz piksel u
        end if
      else
        zalicz (i, j) do obszaru R_k; k ← k + 1
      end if
    end for
  end if
end for
```

Wejście: $R(s)$ jest tablicą obszarów, $s \in [0 \ldots k-1]$, $r(s)$ jest średnią funkcji podobieństwa dla obszaru s, T jest progiem podobieństwa dla obszarów

Wyjście: sąsiadujące podobne obszary zostały połączone

```
merged ← false
repeat
    for all obszar R(s) do
        for all obszar sąsiadujący R(t) do
            if |r(s) − r(t)| < T then
                połącz obszary R(s) oraz R(t)
                merged ← true
            end if
        end for
    end for
until merged
```