

Hisztogram alapú szegmentálás

A legegyszerűbb szegmentálási módszer a küszöbölés, amikor a képpontokat két osztályba soroljuk. A globális küszöbölés minden pont esetén ugyanazt a küszöbértéket tekinti, míg a lokális eljárások a vizsgált pont egy adott környezetében előforduló intenzitások alapján határozzák meg a küszöbértéket.

A küszöbölésen túl léteznek más szegmentálási módszerek, amelyek vagy intenzitás alapúak, vagy pedig régió alapú módszerek. Az intenzitás alapú módszerek közül a hisztogram alapú szegmentálást vizsgáljuk az alábbi példában. Hisztogram alapú szegmentálás olyan képek esetén alkalmazható, amelyeknél meghatározott intervallumtartományok egy-egy objektumhoz vagy objektum-osztályhoz tartoznak a képen.

Az alábbi példában a *haz_auto.jpg* képet fogjuk vizsgálni.

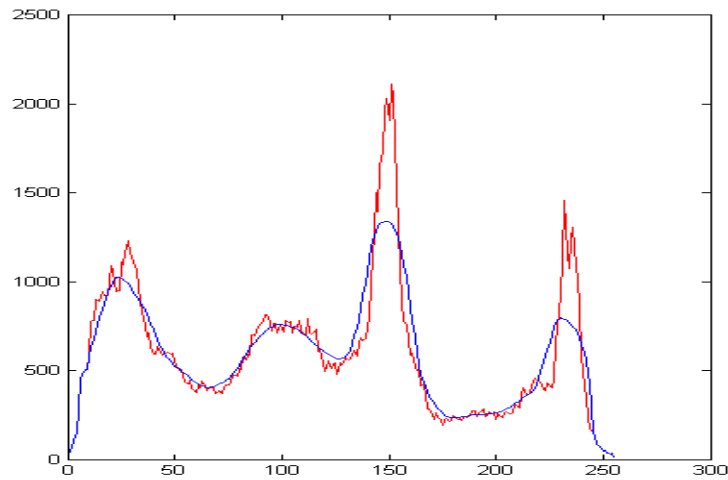
```
>> img = imread('haz_auto');  
>> imshow(img_gray);
```



Vizsgáljuk meg a kép hisztogramját. A hisztogramot ki is rajzolhatjuk, ha lekérdezzük a képen előforduló intenzitás értékeket valamint az előfordulásaik számát és eltároljuk őket egy-egy vektorba.

```
>> [count, x] = imhist(img_gray);  
>> plot(x, count, 'r-');
```

Az eredmény az ábrán látható piros görbe lesz.



A histogramon jól láthatóak lokális „púpok”, amelyek egy-egy intervallum-tartományt határolnak be. Azonban egy ilyen histogramon nehezen találhatók meg ezeket a tartományokat, ezért bevett szokás a histogram, mint görbe simítása. A simítás környezeti átlagolással történik, ahol a környezet mérete jelen példa szerint 21 érték. A környezeti átlagolást nem hajtjuk végre a histogram elején és végén, jelen példa esetén ez nem fontos. A simított histogramot a fenti ábrában késsel jelöltük.

```
>> count2 = count;
>> for i = 11:245;
    count2(i) = sum(count(i-10:i+10)) / 21;
end
```

A simított histogramon már jól kivehetőek azok az intervallumok, amelyeket meg akartunk határozni. A histogramon keressük meg a lokális minimum helyeket, és ezeket a szürkeintenzitásokat, ahol a minimumhelyek találhatóak, gyűjtjük ki egy tömbbe a **find()** parancs segítségével.

```
>> local_mini = 1:length(count);

>> local_mini(:) = 0;
>> for i= 10:250;
    if ( count2(i-1) > count2(i) && count2(i) < count2(i+1) )
        local_mini(i) = 1;
    end;
end

>> local_mins = find(local_mini>0);
```

Címkezzük meg a képet simított histogramon meghatározott intervallumok alapján (egy-egy intervallum két minimumpont között található). Ezt követően a címkézett képet színezzük át **label2rgb()** függvényvel.

```

>> labels =uint8(1:256);
>> for j=1:length(local_mins );
    if (j == 1)
        labels(1:local_mins (j)) = 1;
    else
        labels(local_mins (j-1):local_mins (j)) = j;
    end;
end
>> labels(local_mins(length(local_mins))+1:256) =
    labels(local_mins(length(local_mins))+1);

>> img_label = img_gray;
>> dims = size(img_gray);
>> x = 1:dims(1);
>> y = 1:dims(2);

>> for i = 1:dims(1);
    for j = 1:dims(2);
        img_label(i,j) = labels(img_gray(i,j)+1);
    end;
end;

>> img_color = label2rgb(img_label, 'jet');
>> figure;
>> subplot(1,2,1); imshow(img_gray);
>> subplot(1,2,2); imshow(img_color);

```



Hajtsuk végre ezt a műveletet a honlapon található többi képen is! Lehet, hogy simításhoz változtatni kell a környezet méretén.