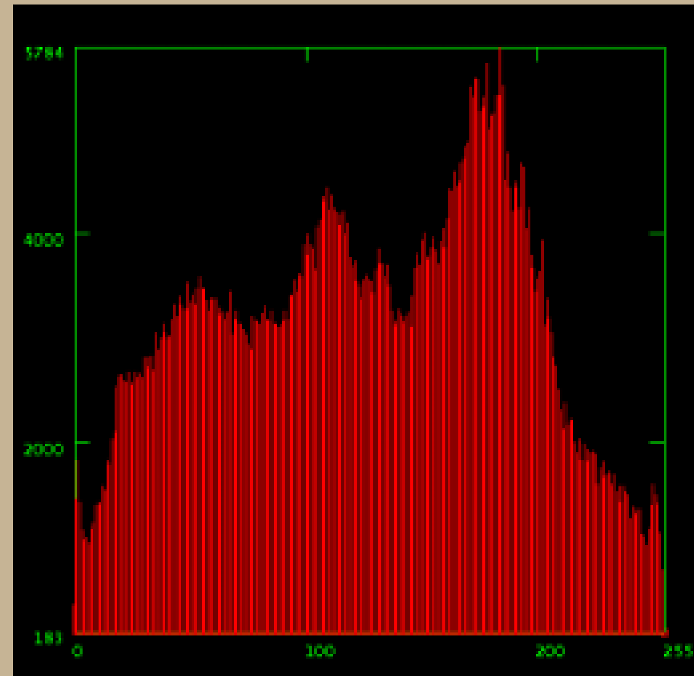
Digitális képek szegmentálása

2. Thresholding

Kató Zoltán

http://www.cab.u-szeged.hu/~kato/segmentation/

Hisztogram



- Olyan grafikon, amely minden lehetséges szürkeárnyalathoz megadja a képen az adott árnyalatú pixelek számát.

- Ha normalizáljuk (minden értéket elosztunk a kép méretével), akkor az egyes pixelértékek előfordulási valószínűségét kapjuk.

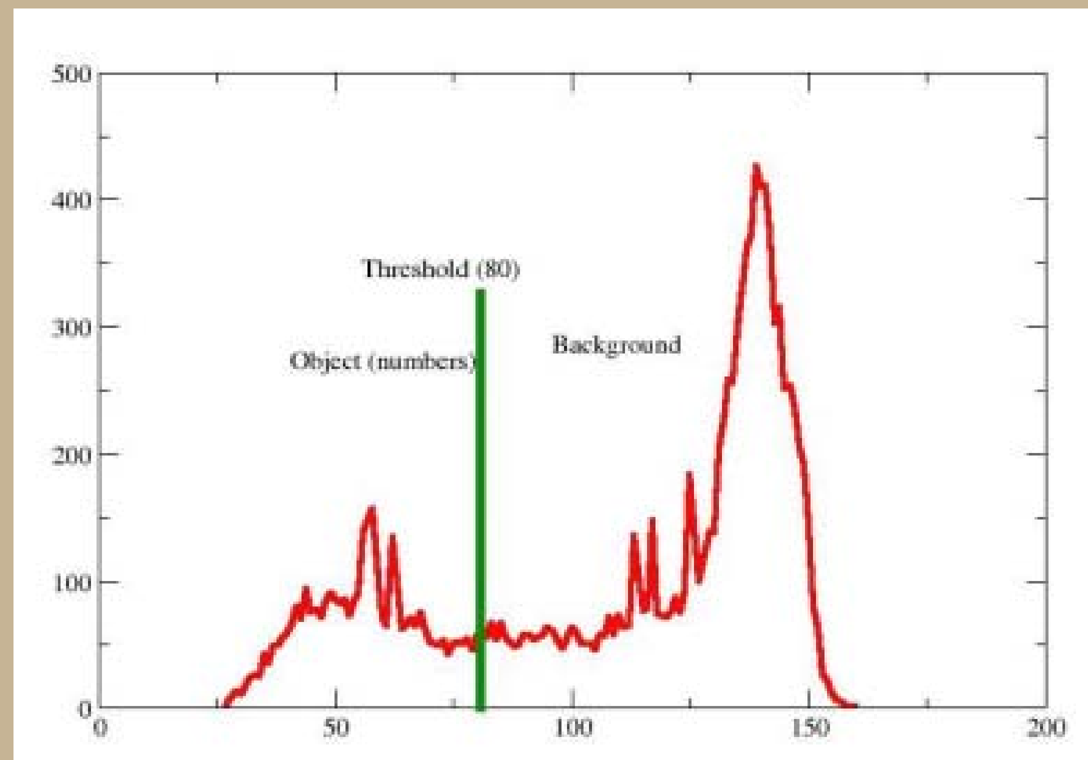
Intenzitás

- Egy szürkeárnyalatos kép fényintenzitást rögzít.
- A rögzített intenzitás egy ***(x,y)*** pontban lényegében két összetevő szorzata: ***f(x,y)=r(x,y)i(x,y)***
  - a felület visszaverődési tulajdonsága: ***r(x,y)***
  - illetve a megvilágítás erőssége: ***i(x,y)***
- Ha a megvilágítás **egyenletes** (vagyis ***i(x,y)*** konstans), akkor ***f(x,y)*** jól tükrözi a szegmentálandó felületek visszaverődési tulajdonságát ➔ a hisztogram jól használható
- Ha a megvilágítás **nem egyenletes**, akkor a ***f(x,y)*** torzítottan adja vissza ***r(x,y)***-t ➔ a hisztogramból nyert információ félrevezető lehet.

Thresholding

- Alapfeltevések:
  - Az objektum és a háttér eltérő intenzitású (nagy kontraszt)
  - Az objektum és a háttér homogén intenzitású

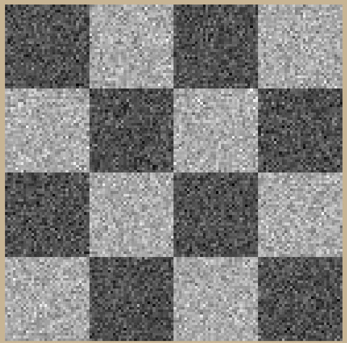
- Például: sötét objektum világos háttéren:
  - ***f(x,y)*** - input kép;
    t(x,y) - szegmentált kép;
    T=küszöbérték

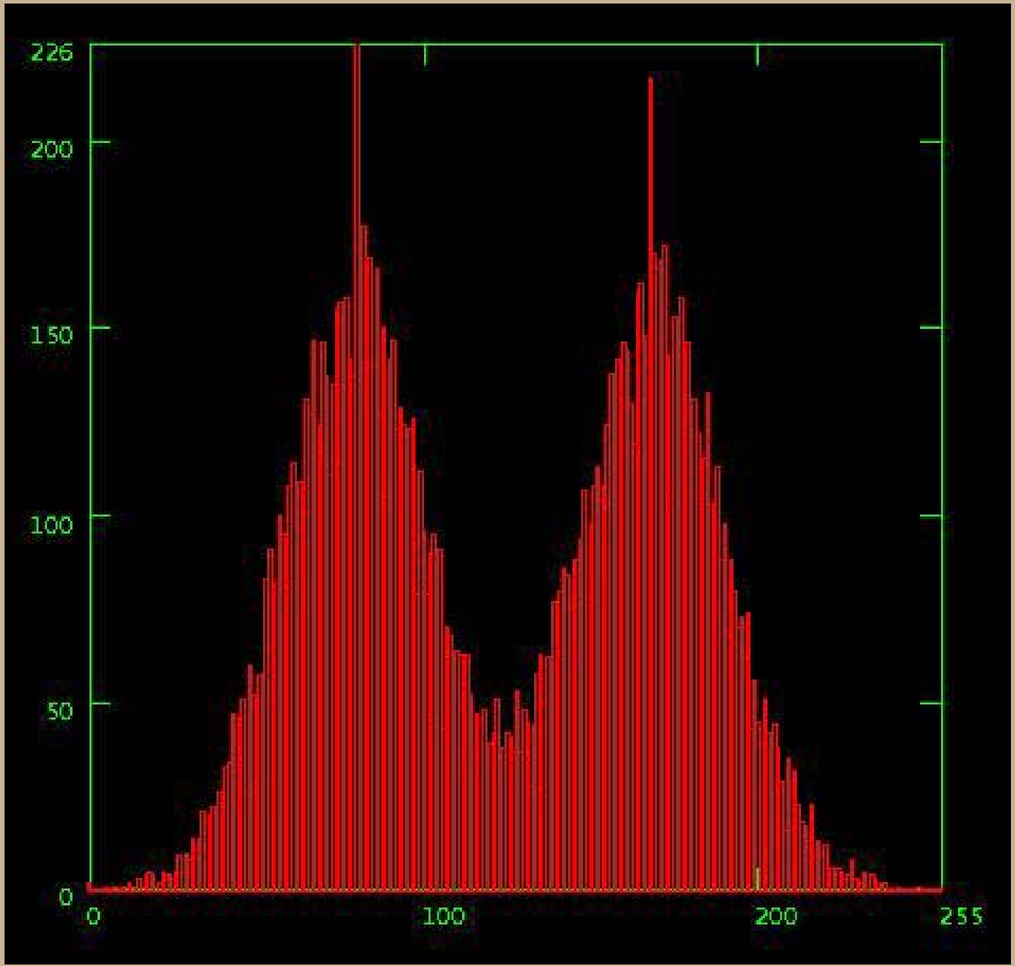


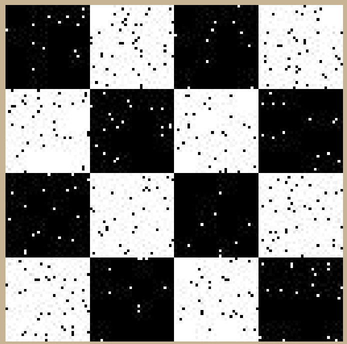
f(x,y)<=T ➔ ***t(x,y)=1*** (object)
f(x,y)>T ➔ ***t(x,y)=0*** (background)

Küszöbérték meghatározása

- Zajos kép esetében nehéz (~lehetetlen) kielégítő küszöbértéket meghatározni → a szegmentálás inhomogén lesz.
  - Zajszűrés jelenthet megoldást, ha nem túl nagy a zaj…





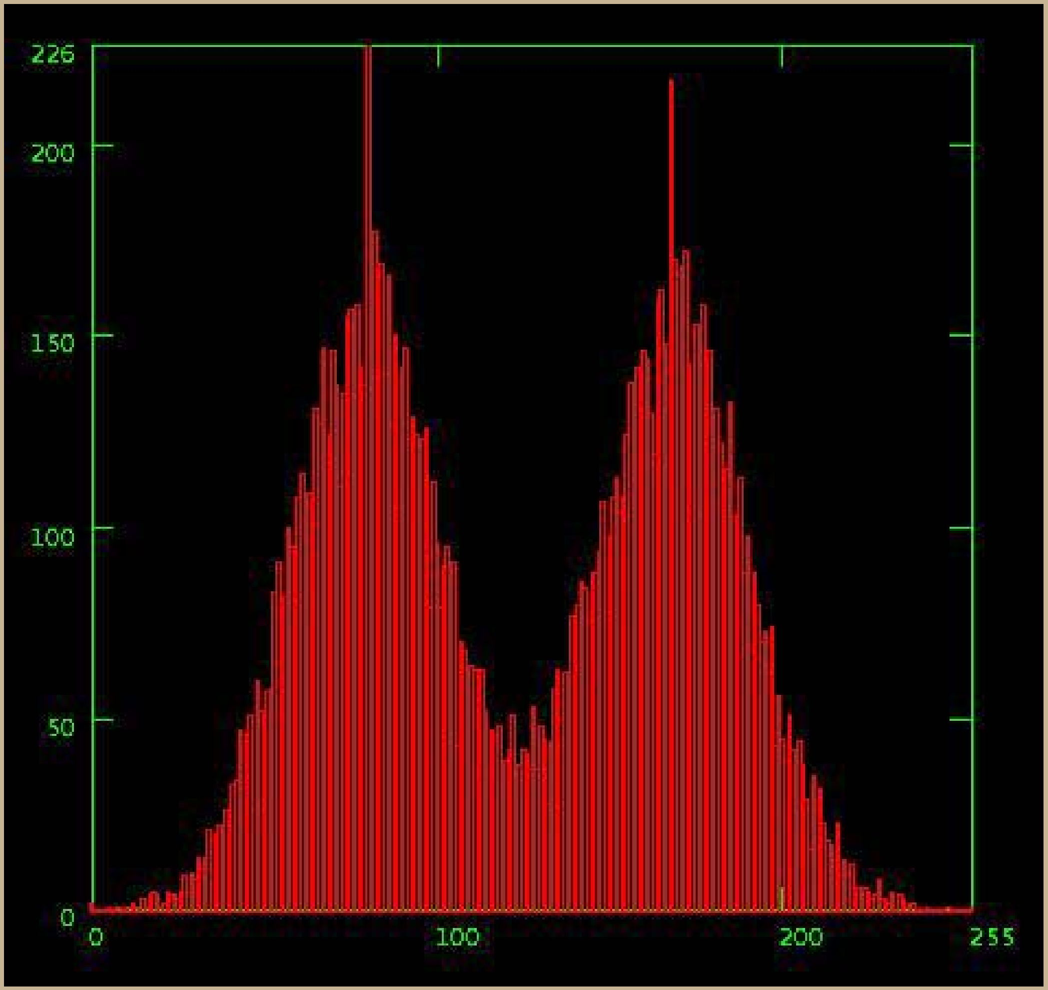


Küszöbérték meghatározása

- Lehet az input képtől függetlenül, manuálisan rögzített érték
  - Egyszerű
  - Kontrollált környezetben jól használható (ipari alkalmazások)
- Adaptív eljárások, melyek az input képhez automatikusan választják ki az optimális küszöbértéket:
  - Globális hisztogram-ból klaszterezéssel számított érték
    - Isodata algoritmus (Yanni)
    - Otsu algoritmus
  - Lokálisan változó érték (egyenetlen megvilágítás)
    - Niblack algoritmus

Isodata algoritmus (Yanni)

- Jól használható, ha az előtér és a háttér kb. ugyanannyi pixelből áll.
  - **Inicializálás**: a hisztogramot két részre osztjuk (célszerűen a felezőponton): T_0
  - kiszámítjuk az objektum valamint a háttér intenzitásának középértékét: M_i, m_i
  - Az új küszöbérték a két középérték átlaga: $T_i=(M_i+m_i)/2$
  - **Vége** ha a küszöbérték már nem változik: $T_{k+1}=T_k$



Otsu algoritmus

- A bemeneti kép ***L*** szürkeárnyalatot tartalmaz
- a normalizált hisztogram minden ***x*** szürkeértékhez megadja az előfordulási gyakoriságát (valószínűségét): p_x
- Az algoritmus lényege: keressük meg azt a ***T*** küszöbszámot, amely **maximalizálja** az objektum-háttér közötti varianciát.
- Az előtér/háttér pixelek gyakorisága (valószínűsége)

$$\text{háttér}: \quad B(T) = \sum_{x=1}^{T} p_x \qquad \text{objektum}: \quad 1 - B(T) = \sum_{x=t+1}^{L} p_x$$

$$\mu(T) \equiv \sum_{x=1}^{T} x p_x \;, \text{ a teljes kép középértéke}: \quad \mu \equiv \frac{\sum_{x=1}^{L} x p_x}{\sum_{x=1}^{l} p_x}$$

Otsu algoritmus

- Obkektum/háttér pixelek középértéke: $\mu_B = \frac{\mu_T}{B(T)}, \quad \mu_O = \frac{\mu - \mu_T}{1 - B(T)}$
- Hasonlóan a szórás: $\sigma_B^2 = \frac{1}{B(T)} \sum_{x=1}^{T} (x - \mu_B)^2 p_x, \quad \sigma_O^2 = \frac{1}{1 - B(T)} \sum_{x=T+1}^{L} (x - \mu_O)^2 p_x$
- A teljes szórás:

$$\sigma_T^2 = \sum_{x=1}^{T} (x - \mu)^2 p_x + \sum_{x=T+1}^{L} (x - \mu)^2 p_x = \ldots =$$

$$= \underbrace{B(T)\sigma_B^2 + (1 - B(T))\sigma_O^2}_{\text{osztályon belüli varianciától függ}} + \underbrace{(\mu_B - \mu)^2 B(T) + (\mu_O - \mu)^2 (1 - B(T))}_{\text{osztályok közötti varianciától függ}} \equiv \sigma_W^2(T) + \sigma_B^2(T)$$

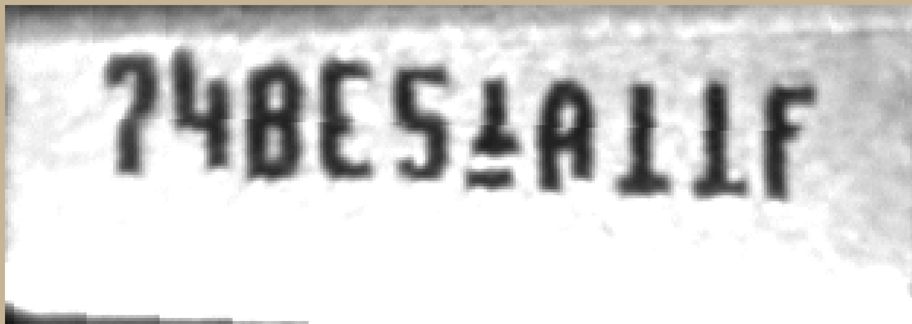
- nyilván σ_T^2 konstans, és **T**-t úgy kell beállítani, hogy $\sigma_B^2(T)$ a lehető legnagyobb legyen:

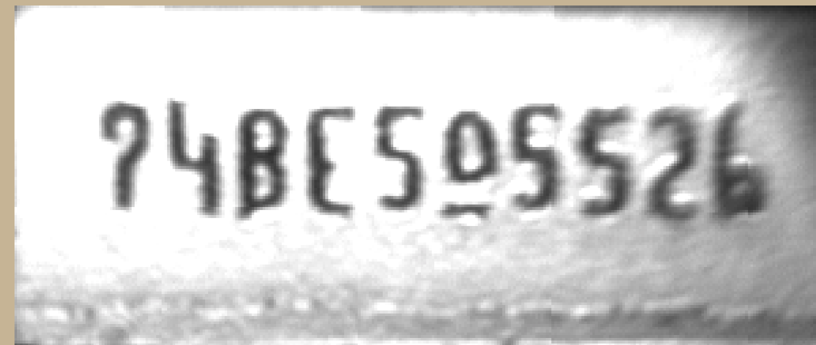
$$\sigma_B^2(T) = \ldots = \frac{(\mu(T) - \mu B(T))^2}{B(T)(1 - B(T))}$$

Otsu algoritmus

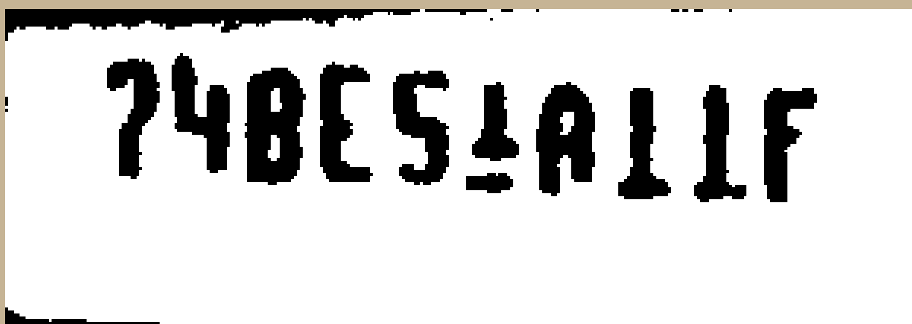
- A hisztogram elejéről kezdve nézzük meg minden szürkeértéket, mint lehetséges küszöböt:
  - Számoljuk ki $\sigma_B^2(T)$ értékét $\mu(T)$ és $B(T)$ segítségével
  - mindaddig növeljük T értékét, amíg $\sigma_B^2(T)$ növekszik.
- Ez az algoritmus feltételezi, hogy $\sigma_B^2(T)$ –nek egyetlen maximuma van!

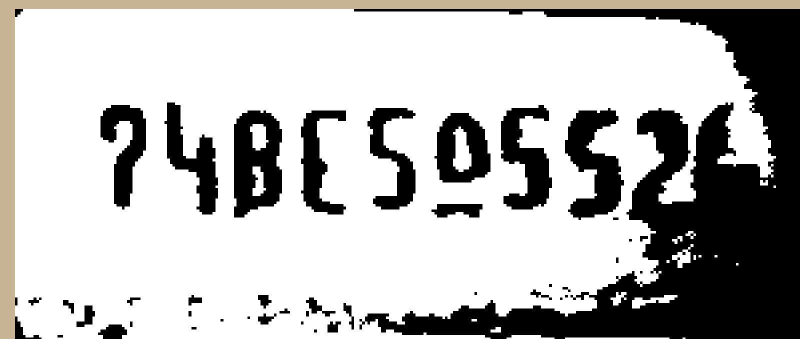
Példák



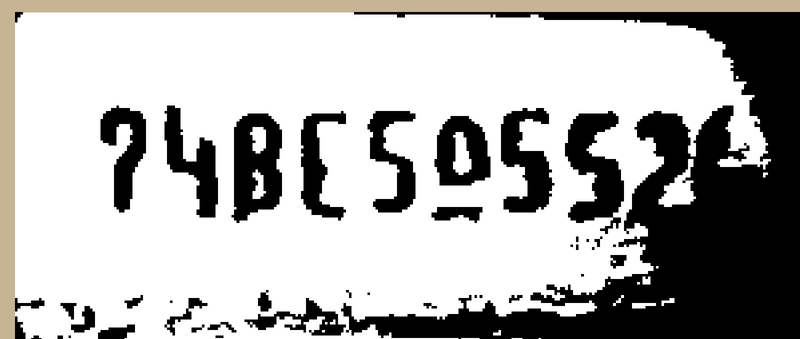


Original





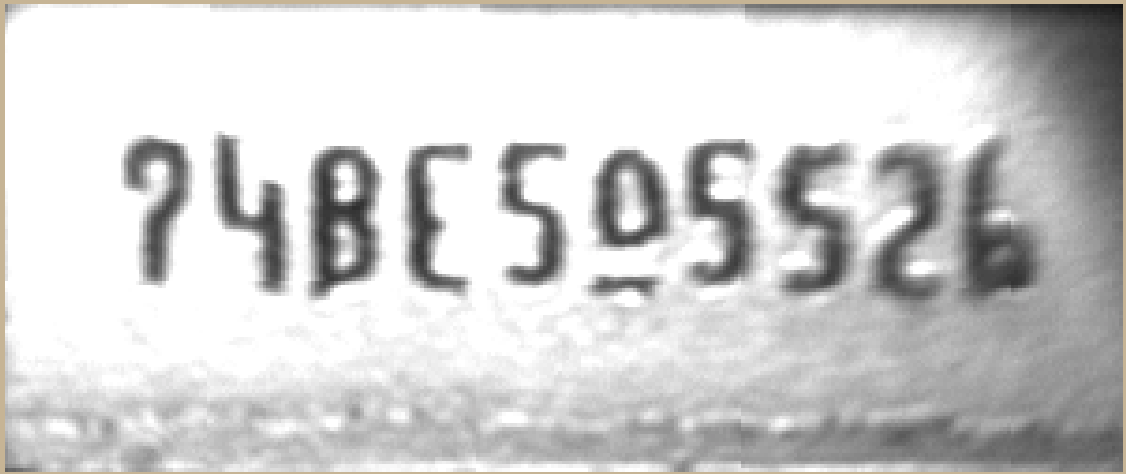
Isodata (Yanni)

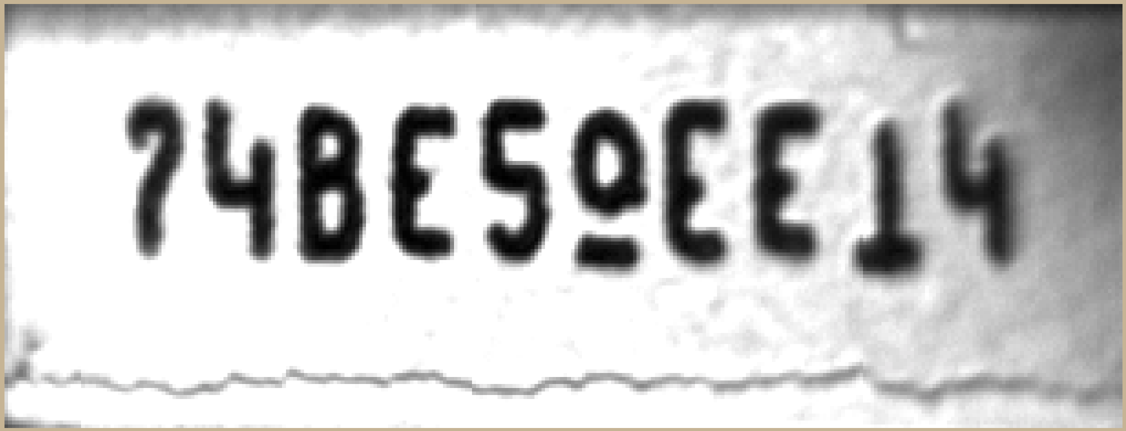


Otsu

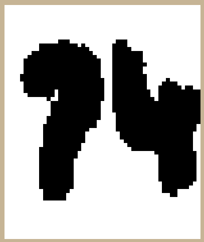
Lokálisan változó küszöbérték

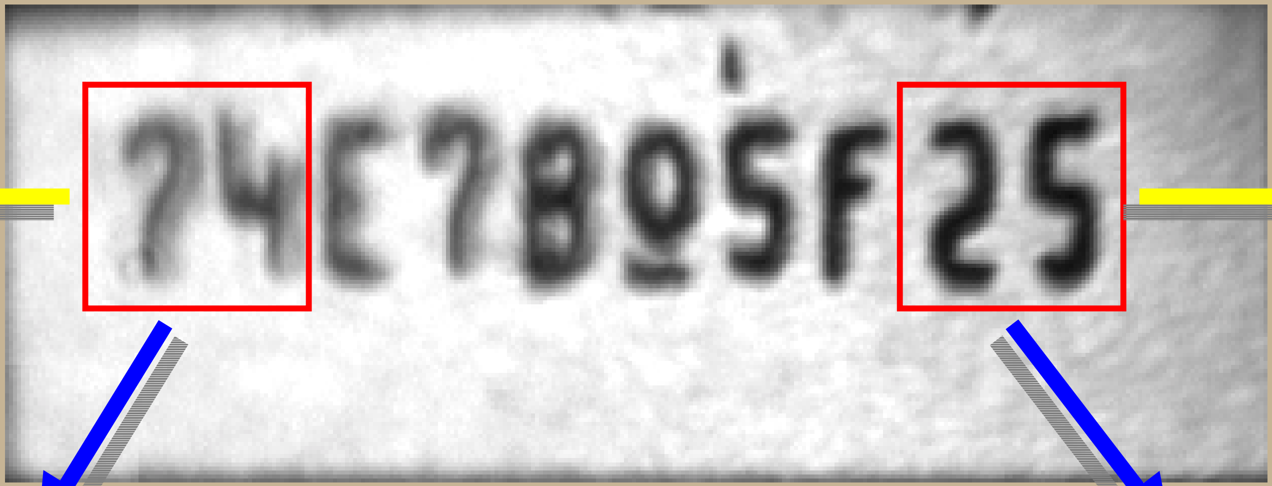
- Mit tehetünk abban az esetben, ha az objektum vagy a háttér nem homogén?
- Amennyiben az objektum és a háttér kontrasztja lokálisan továbbra is nagy, akkor alkalmazhatunk **lokális küszöbölést**.





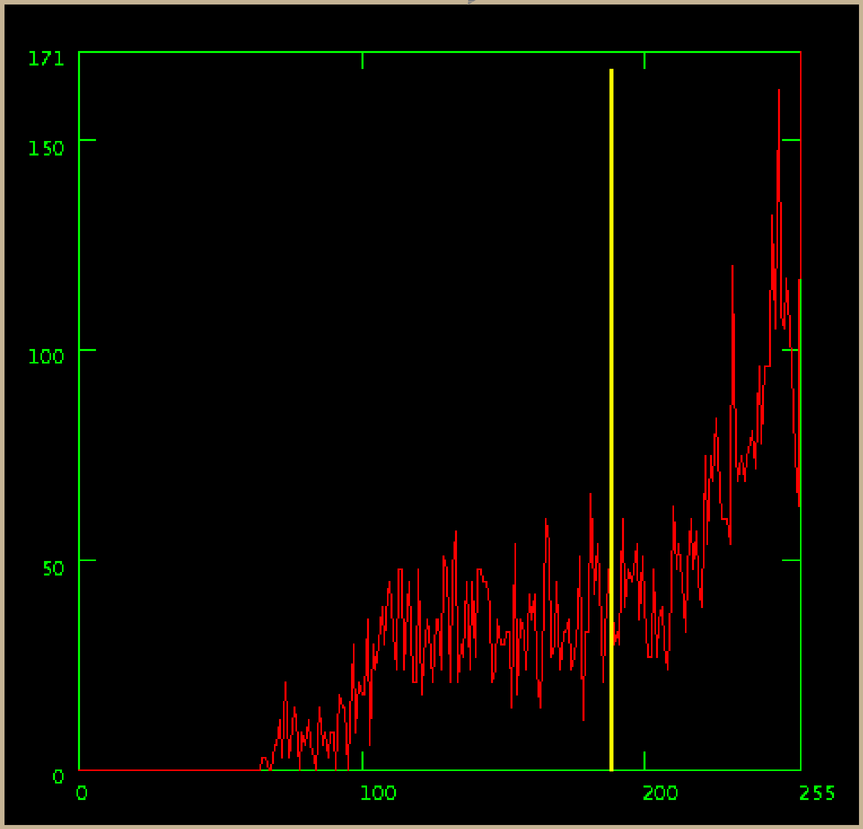
Lokális hisztogram + Otsu

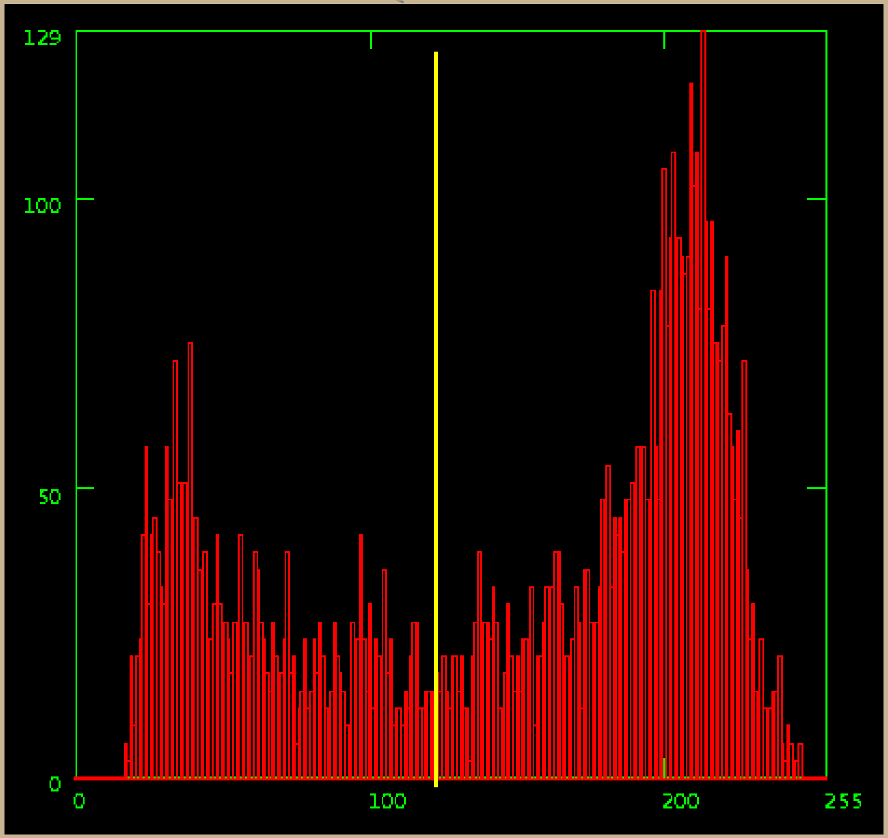




T=186

T=133



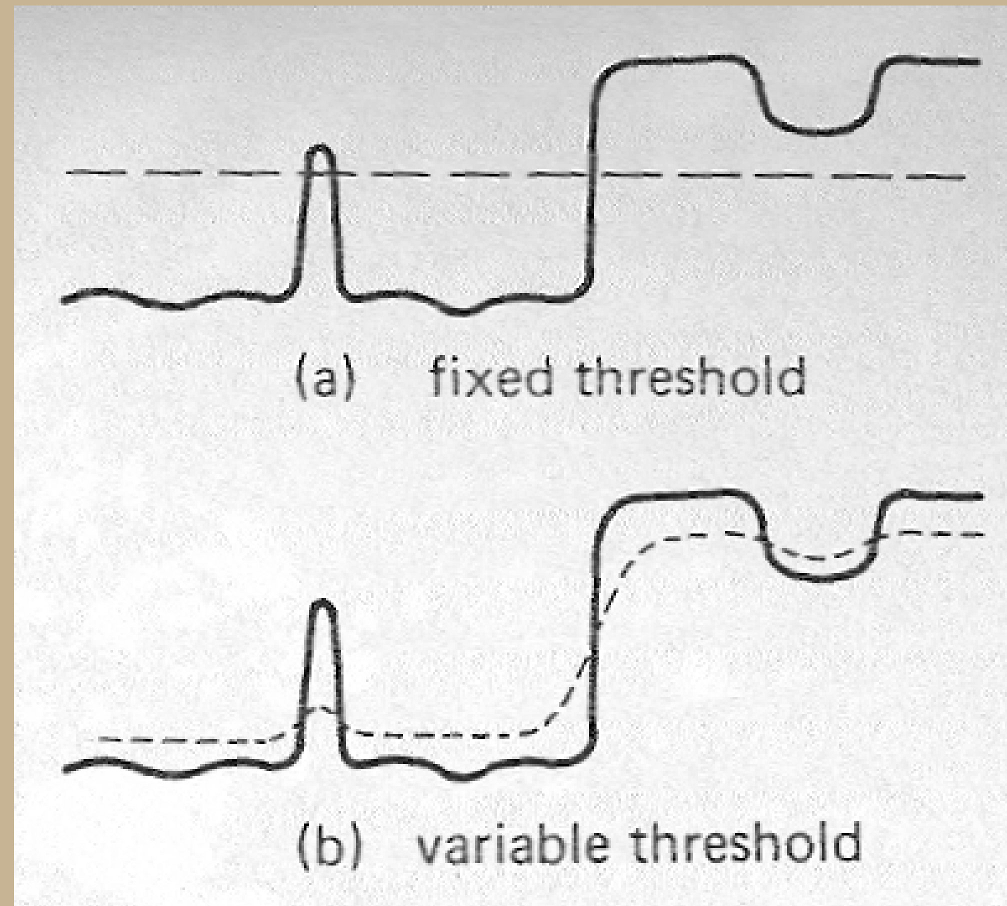


Niblack algoritmus

- Egyetlen küszöb nem elegendő az objektum és háttér szétválasztásához
- ➔változó küszöbérték (***T(i,j)***) kell, amely követi az intenzitásváltozásokat:

$$T(i,j) = \mu(i,j) + k\sigma(i,j)$$

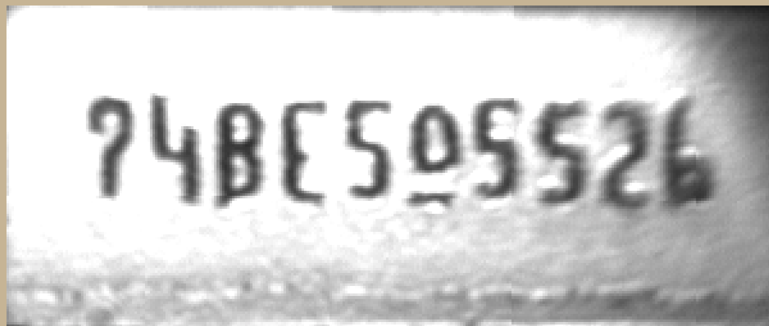
  - ***(i,j)*** adott környezetében:
    - ***μ(i,j)*** - középérték
    - ***σ(i,j)*** - szórás
  - ***k*** – mennyire vegyük figyelembe a szórást
    - Sötét objektum ➔ k< 0, világos objektum ➔ k>0
    - Általában ***|k| ~ 0.2***

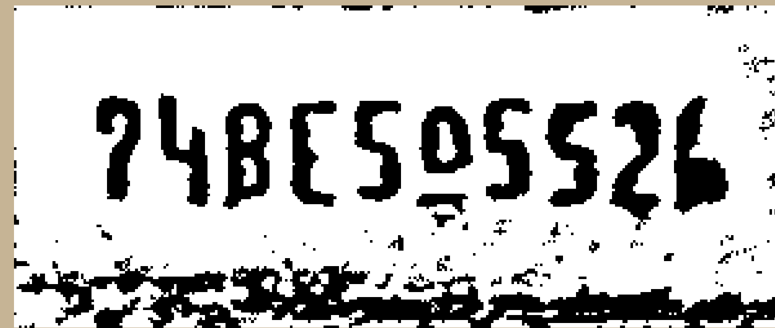


A környezet mérete: elég kicsi ahhoz, hogy a lokális részleteket megőrizze, de elég nagy ahhoz, hogy a zajt elnyomja (általában ***15X15***).

Niblack algoritmus: példák

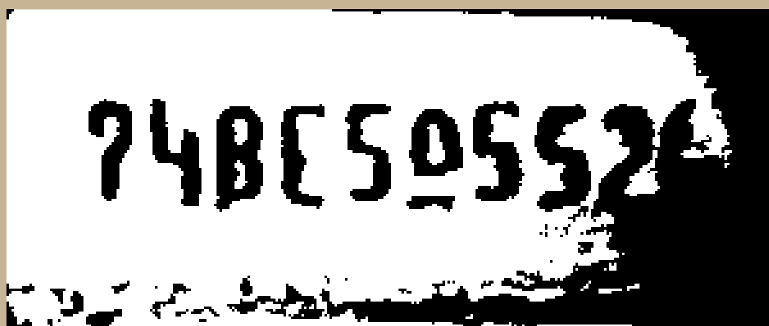
original

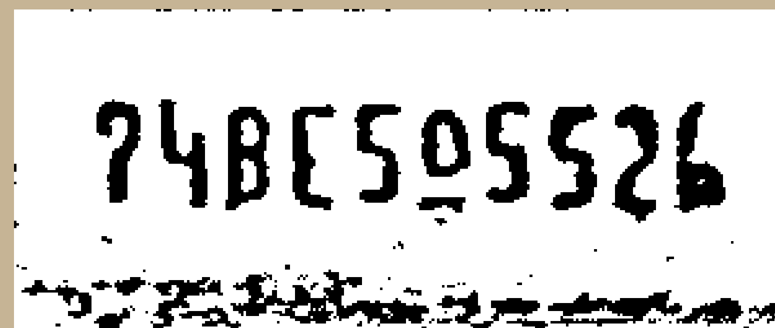




Niblack
k=-0.2
30X30

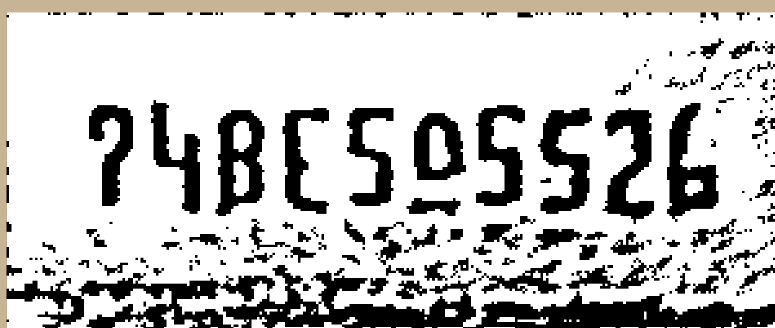
Otsu

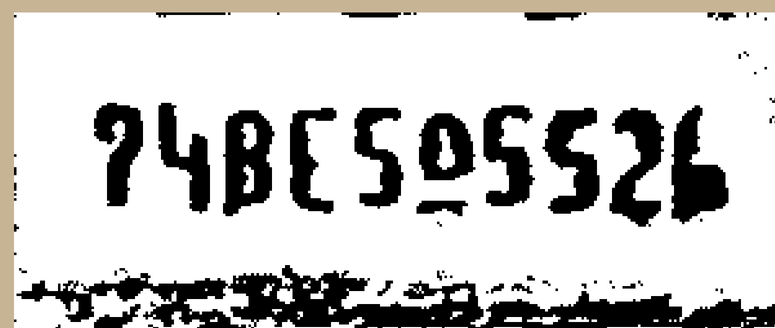




Niblack
k=-0.5
30X30

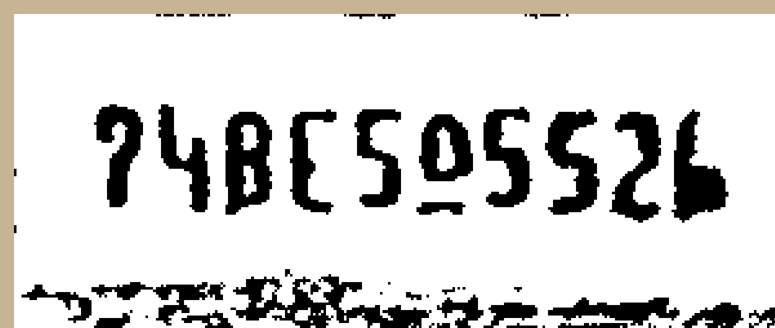
Niblack
k=-0.2
15X15





Niblack
k=-0.2
60X60

Niblack
k=-0.5
15X15



Niblack
k=-0.5
60X60

Postprocessing

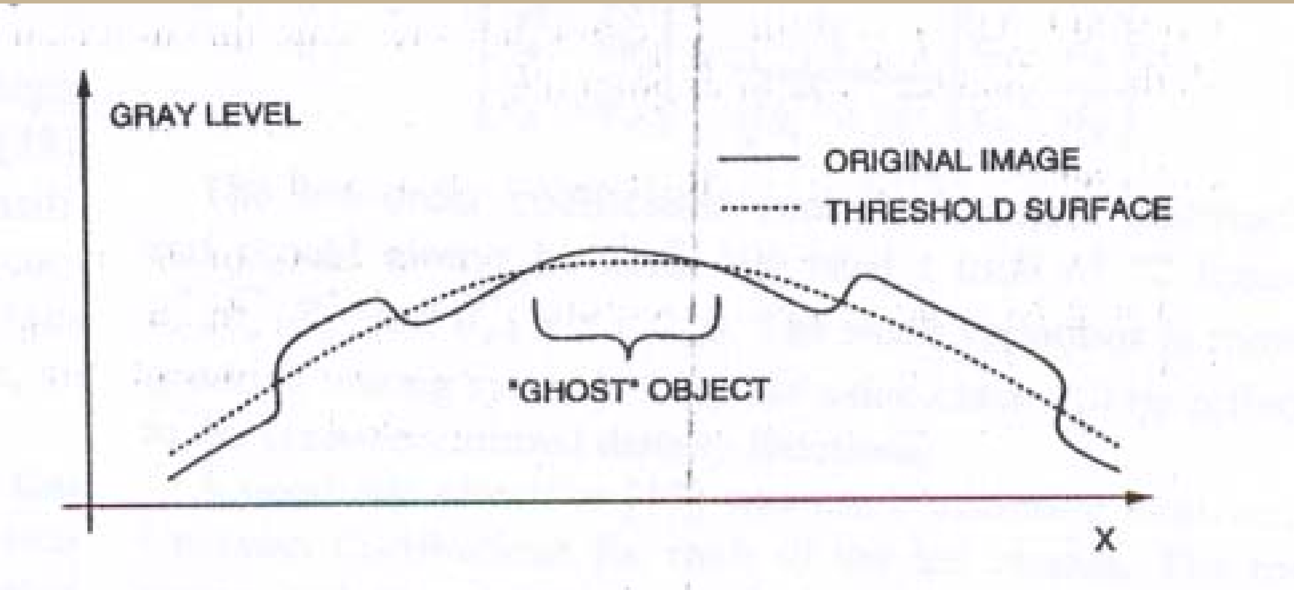
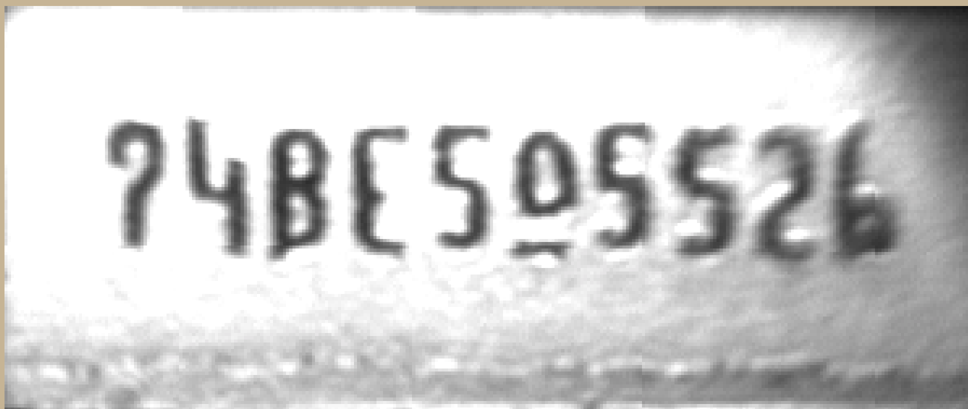


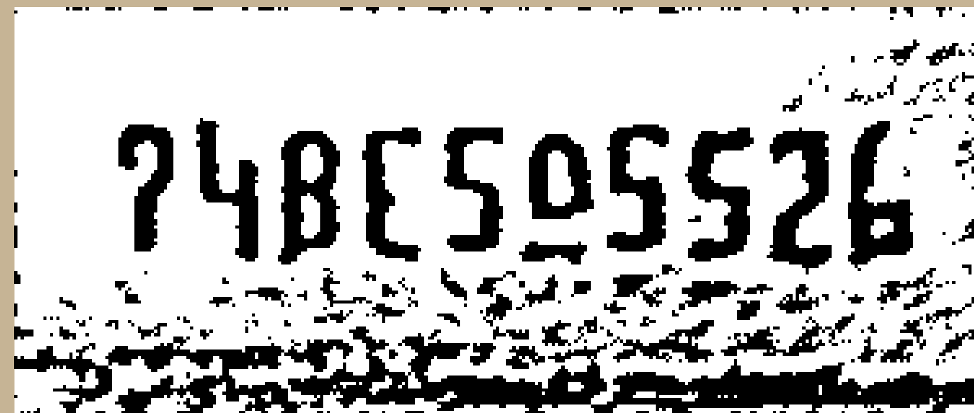
Fig. 4. Cross-section through an image, illustrating how "ghost" objects might occur. Based on a figure in Yanowitz and Bruckstein [24].

- "ghost" objektumok eltüntetése
  - Számitsuk ki az átlagos gradiens értékét az objektumok élei mentén
  - Töröljük le azokat az objektumokat, amelyeknek az átlagos gradiense egy adott küszöbérték alatt van

Postprocessing példák

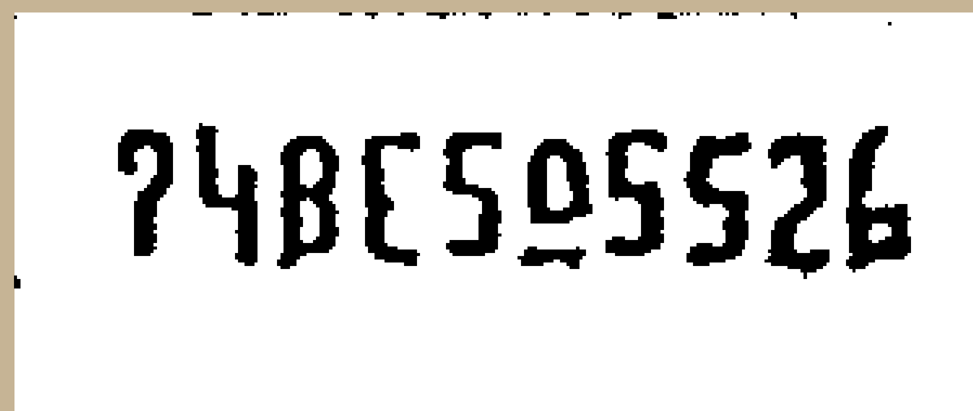


original



Niblack k=-0.2 15X15

gradiens



postprocess után