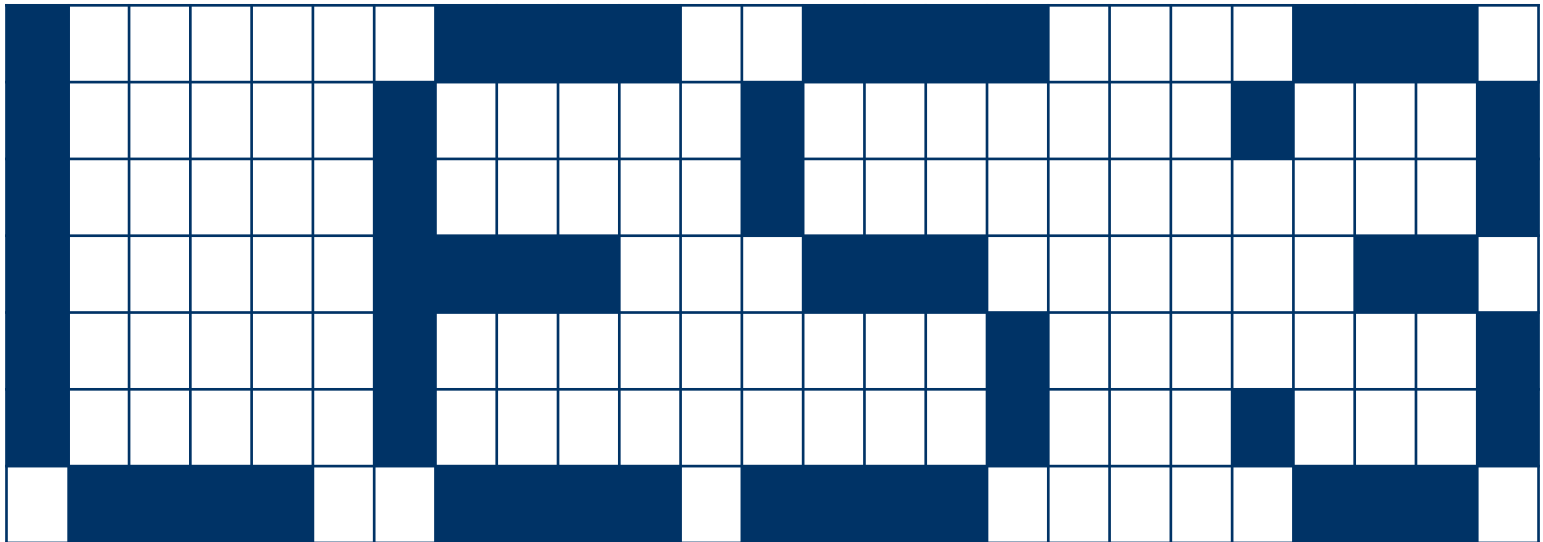


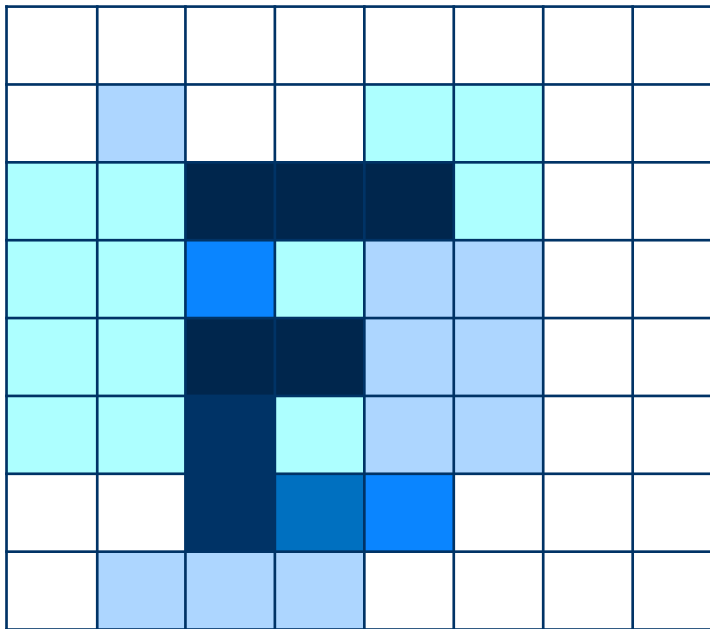


EVD1 – Vision operators

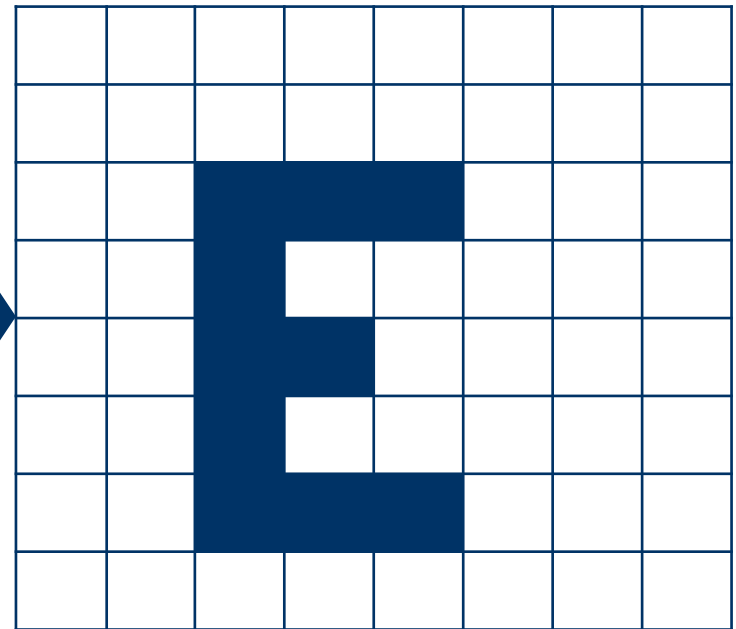




Threshold



Source



Destination



Automatic Threshold

```
void vHistogram(image_t *src,
                uint16_t *hist,
                uint32_t *sum);
```

Histogram

nof pixels

n

0



Object

Background

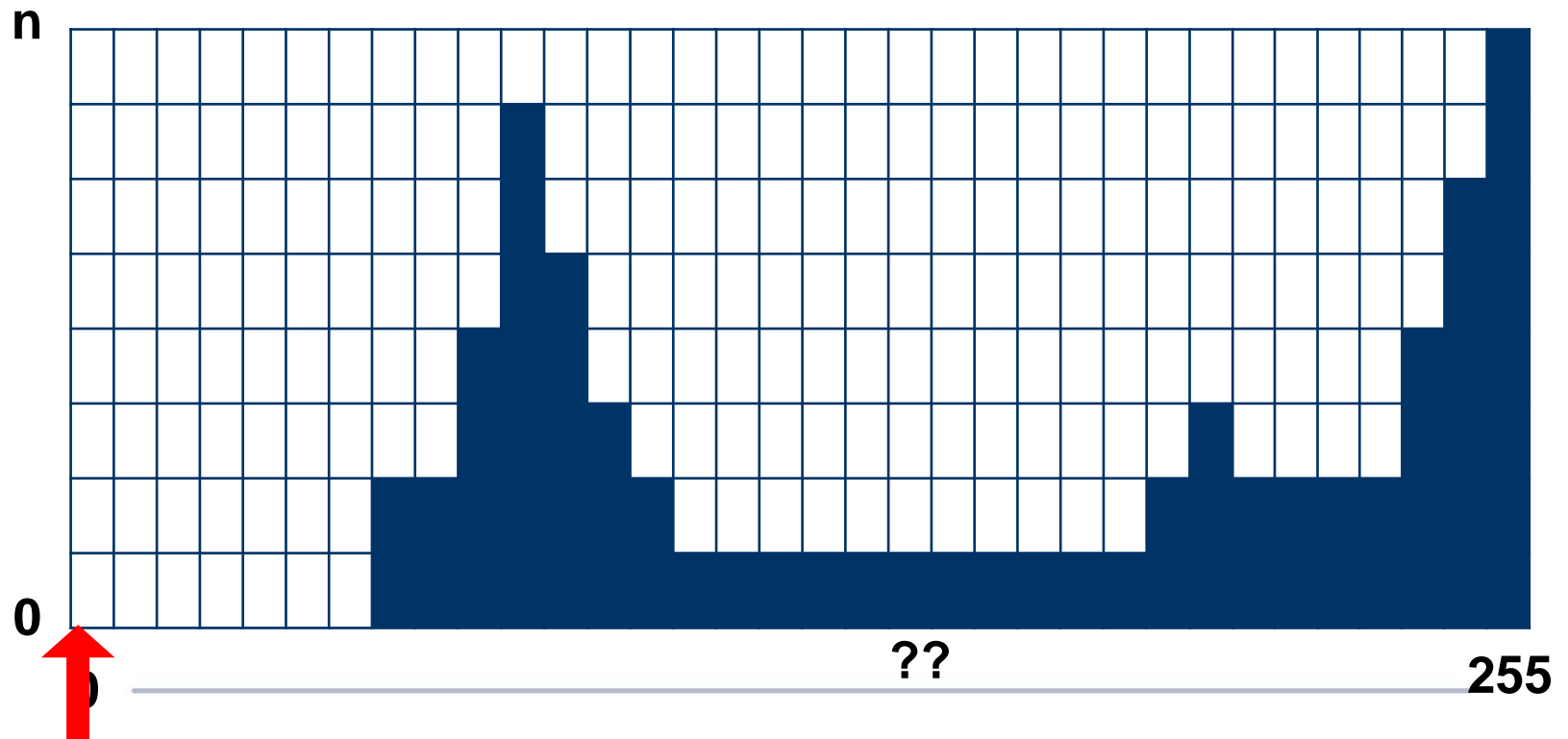
255

Grey value



Automatic Threshold – simple?

1. Find two maxima
2. Find minimum in between

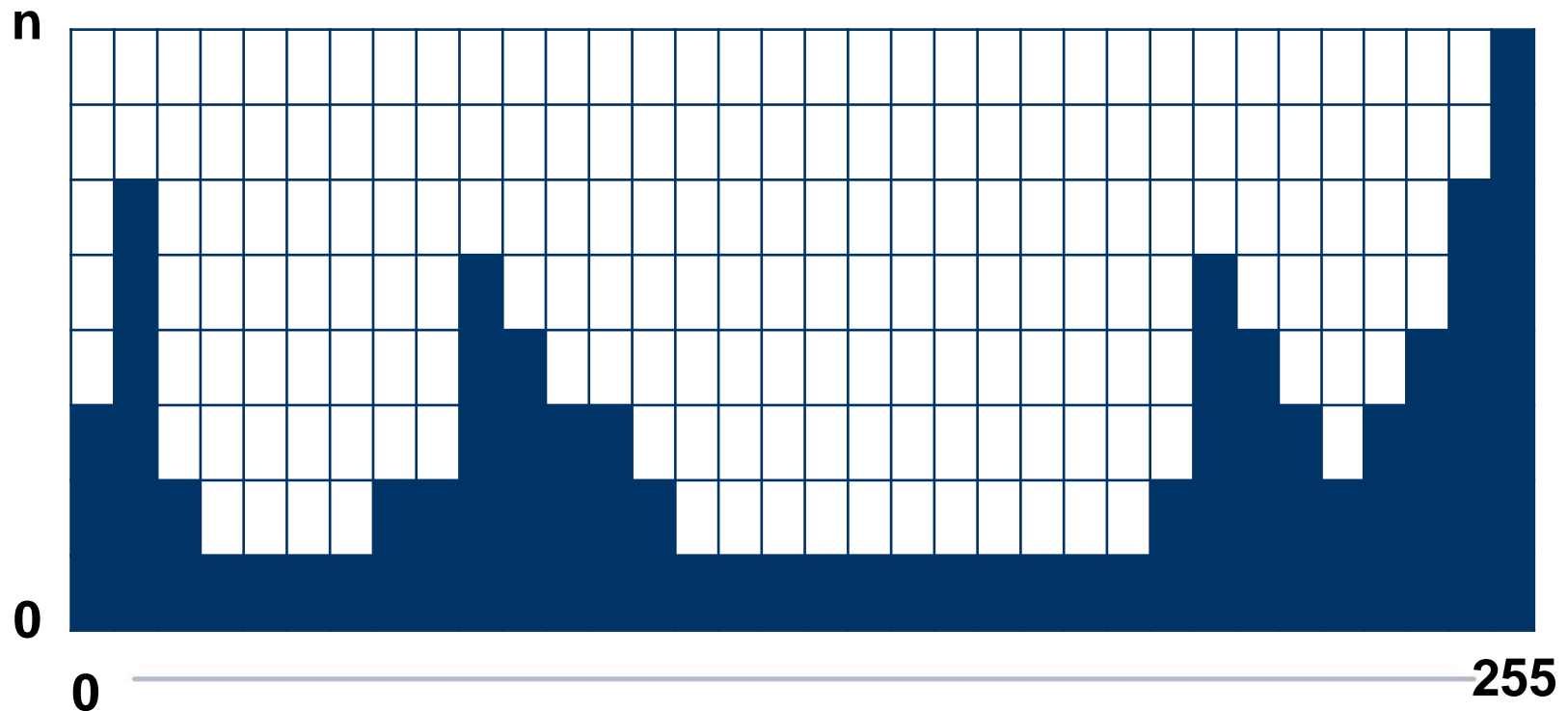




Automatic Threshold – simple?

Werkt niet bij:

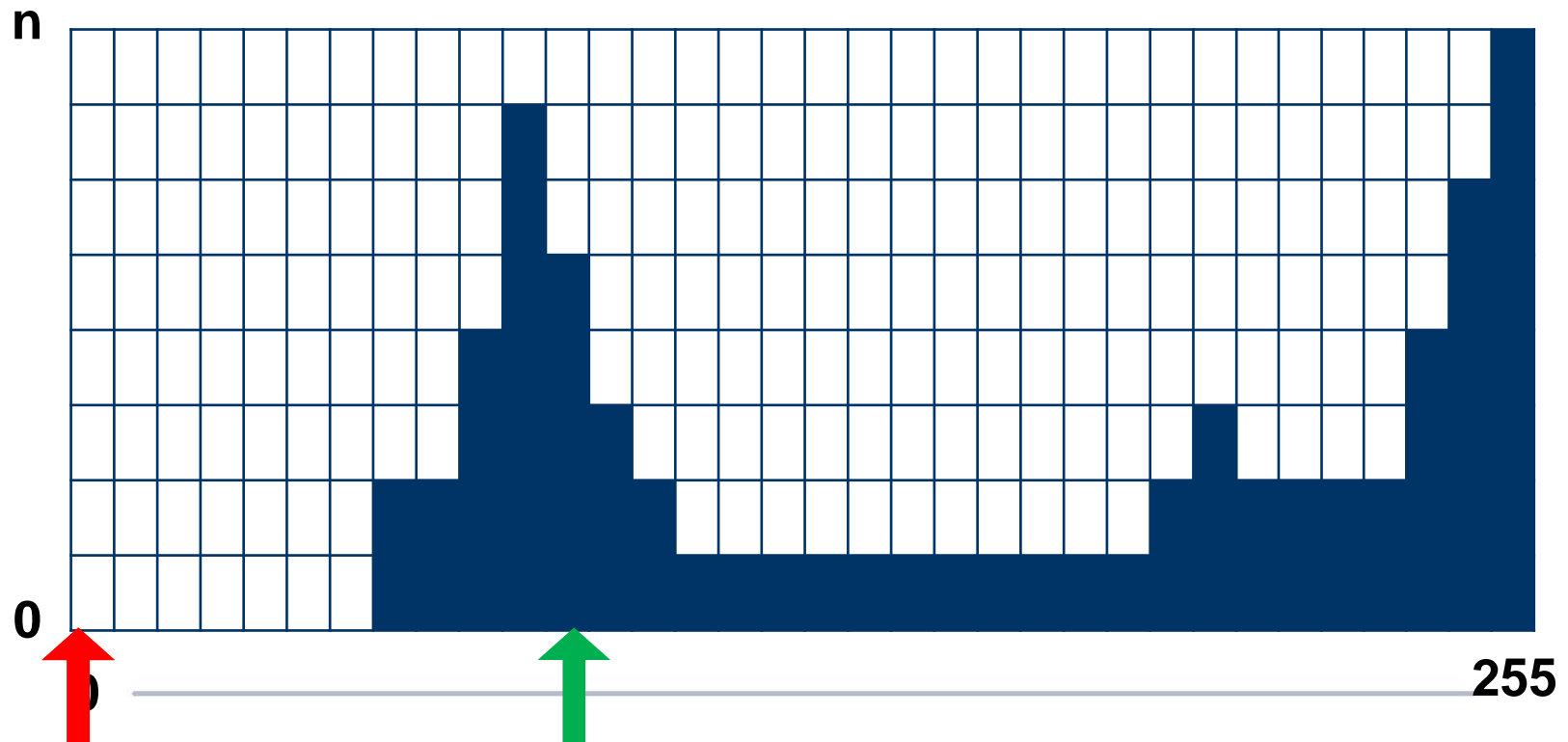
- (veel) ruis pixels (op te lossen door eerst een smooth filter toe te passen)
- 'vreemde' verdelingen





Automatic Threshold – 2 Means

- Find the mean value of the foreground pixels
- Find the mean value of the background pixels
- Find the mean of these two means

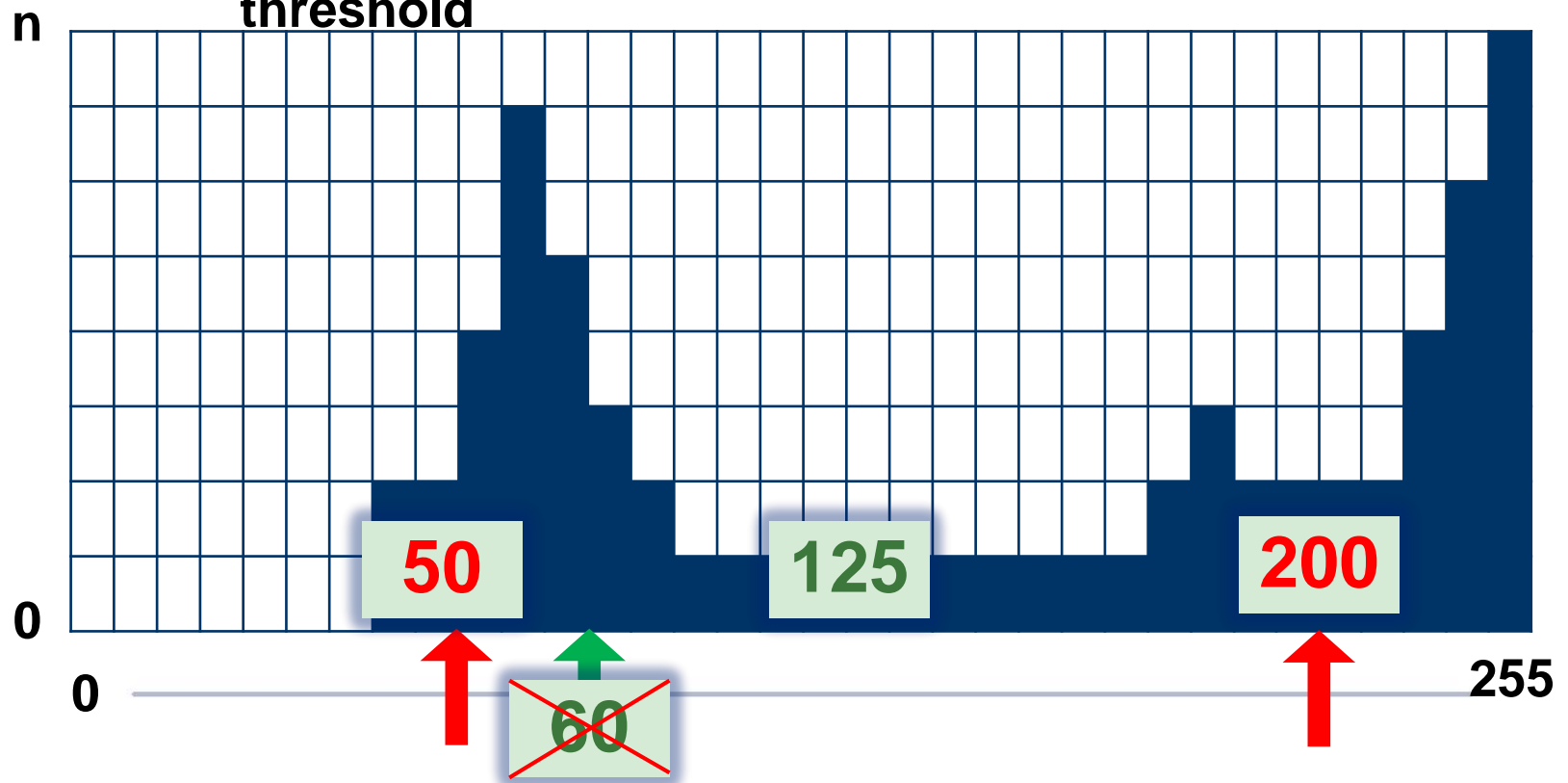


Automatic Threshold – 2 Means

Methode 1:

For every possible threshold:

1. Calculate meanLeft and meanRight
2. Calculate mean of meanLeft and meanRight
3. Optimum when calculated mean equals current threshold

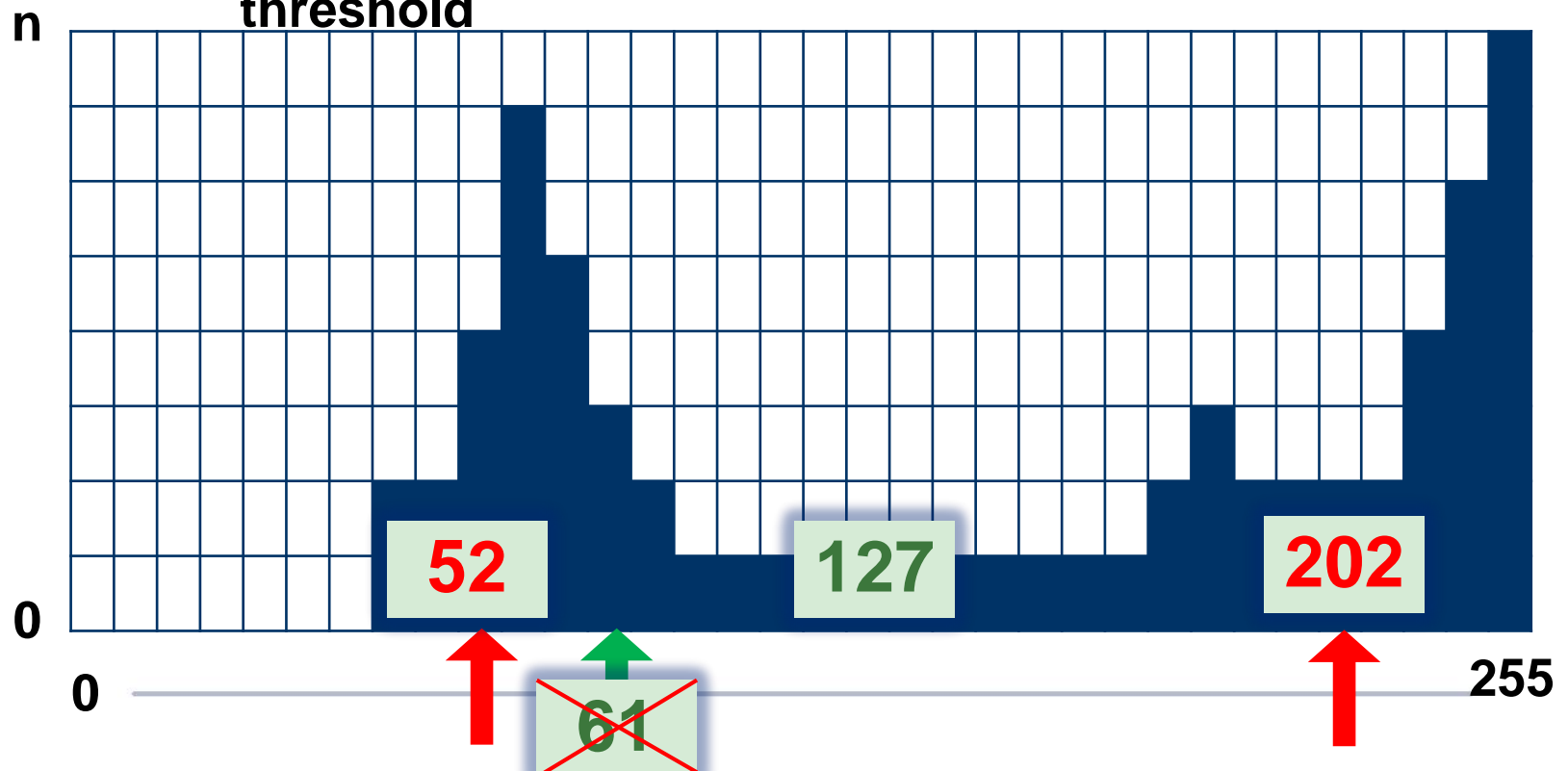


Automatic Threshold – 2 Means

Methode 1:

For every possible threshold:

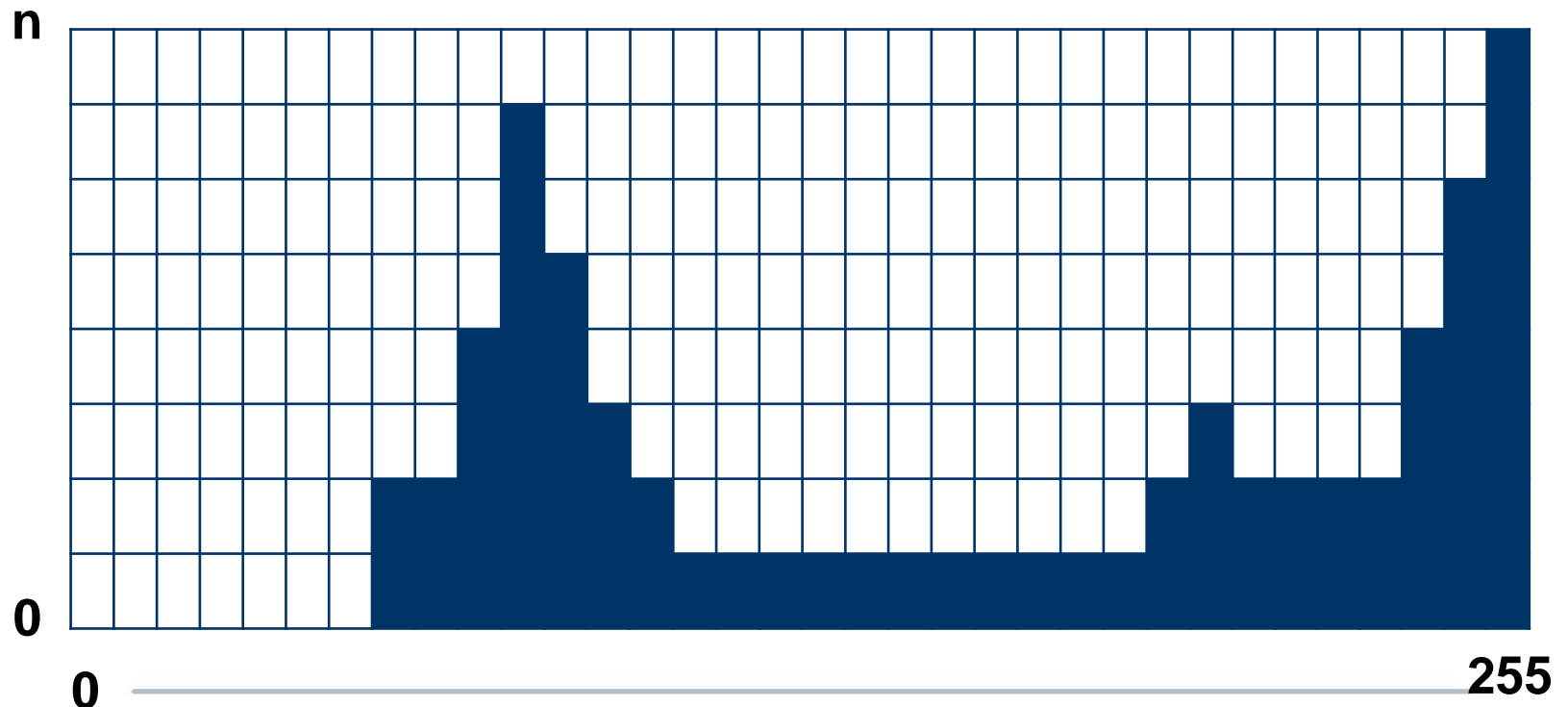
1. Calculate meanLeft and meanRight
2. Calculate mean of meanLeft and meanRight
3. Optimum when calculated mean equals current threshold





Automatic Threshold – 2 Means

Methode 1:
Disadvantage: Slow

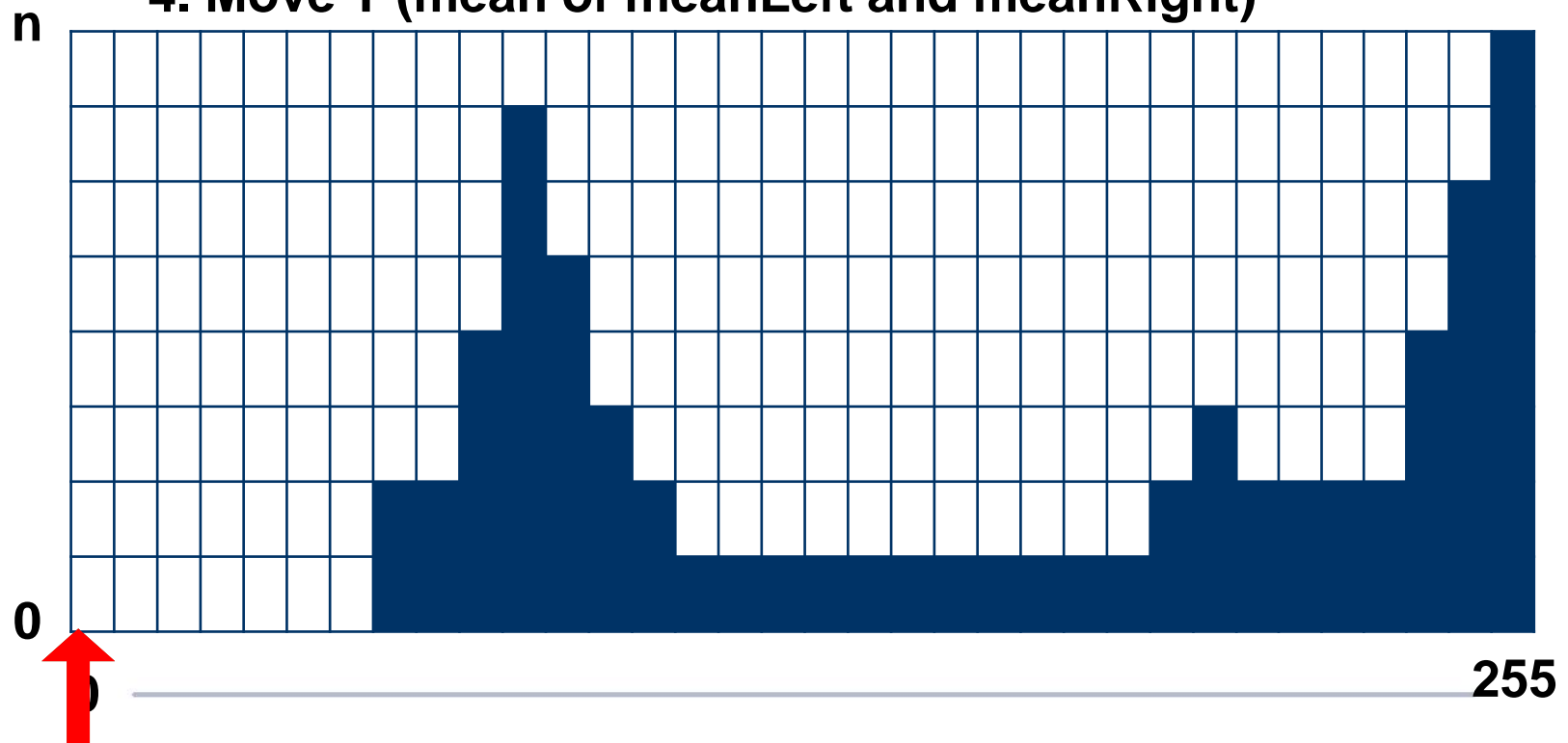




Automatic Threshold – 2 Means

Methode 2

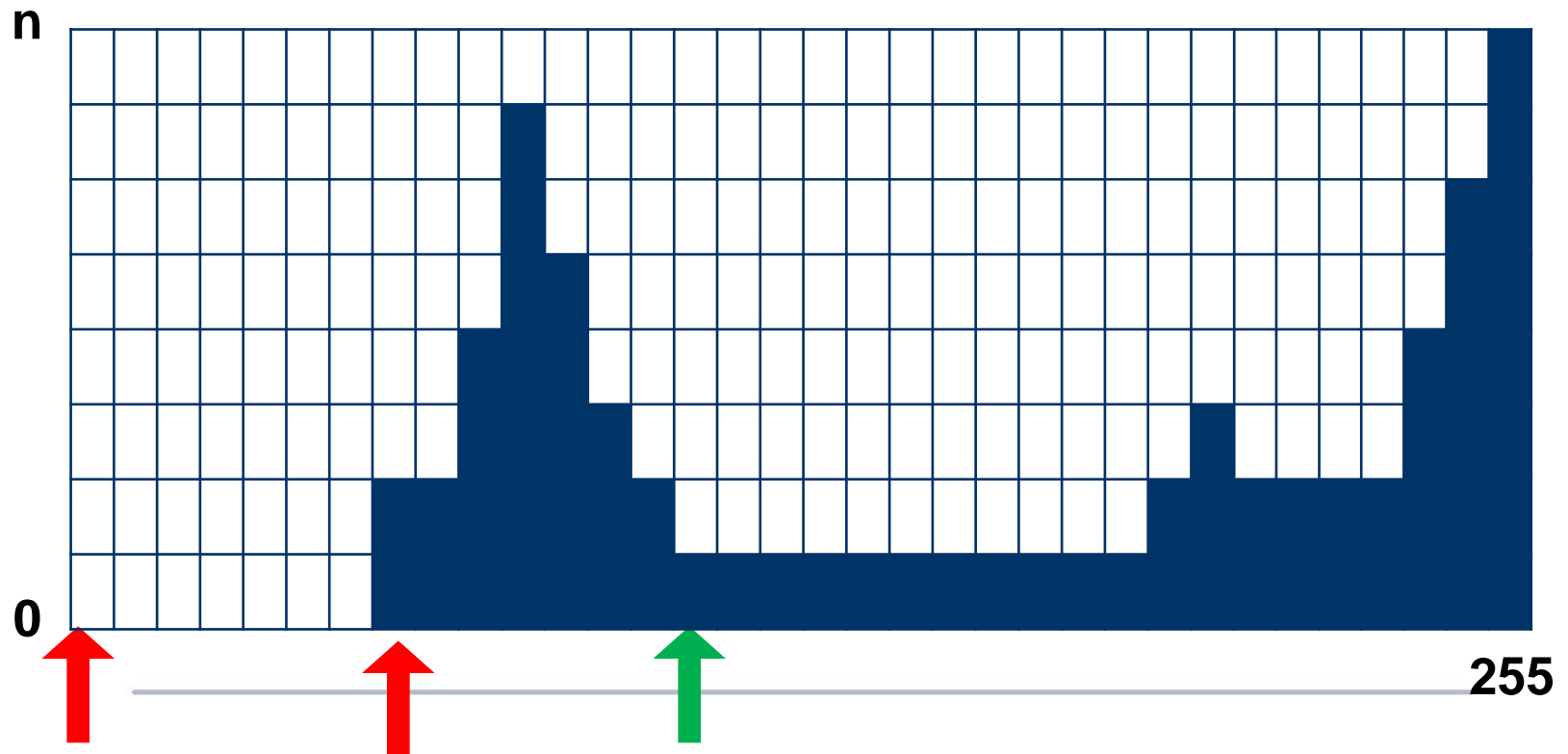
1. Set T to any value between lPixel and hPixel
2. Calculate meanLeft
3. Calculate meanRight
4. Move T (mean of meanLeft and meanRight)





Automatic Threshold – 2 Means

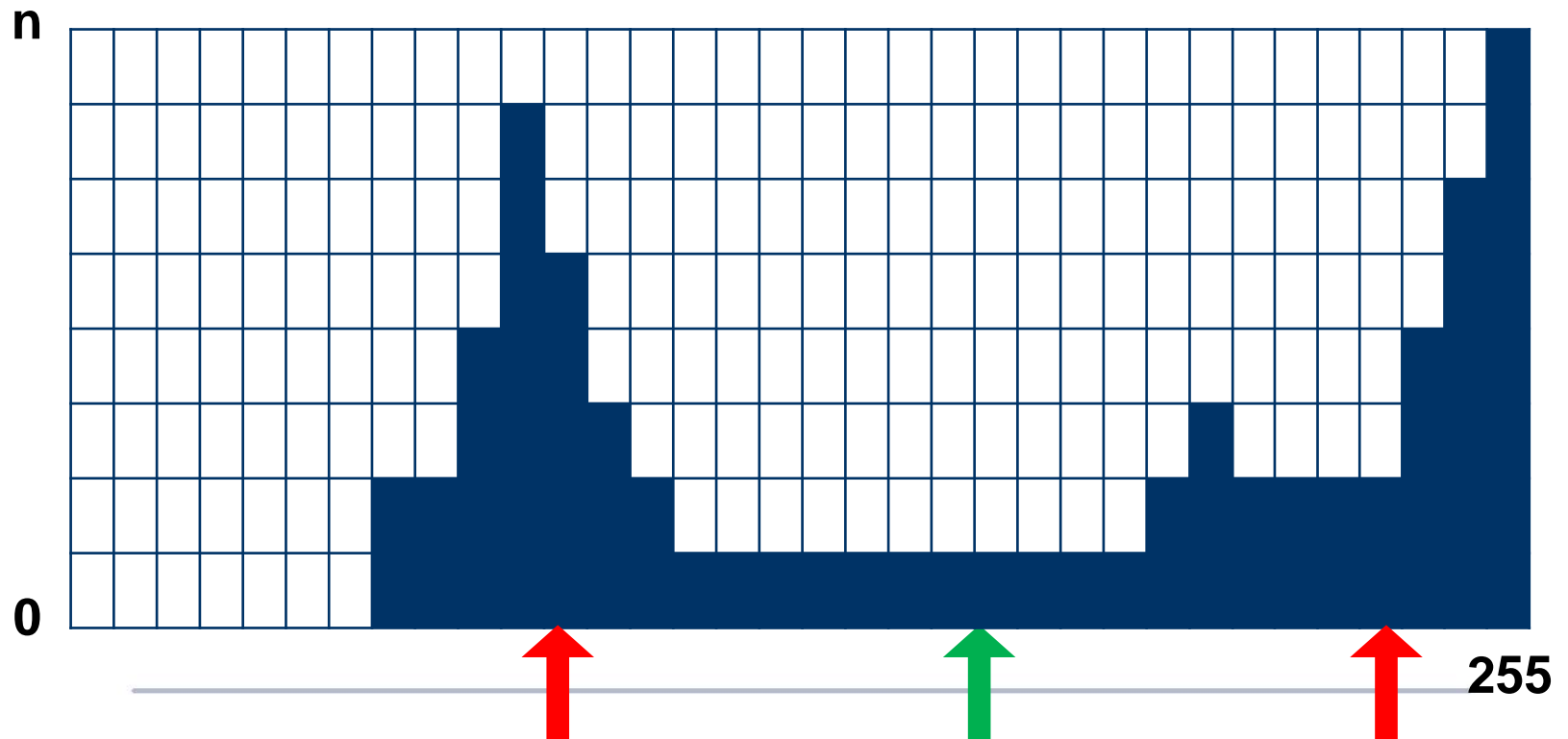
2. Calculate meanLeft
3. Calculate meanRight
4. Move T (mean of meanLeft and meanRight)





Automatic Threshold – 2 Means

Repeat until T doesn't change





Automatic Threshold – 2 Means

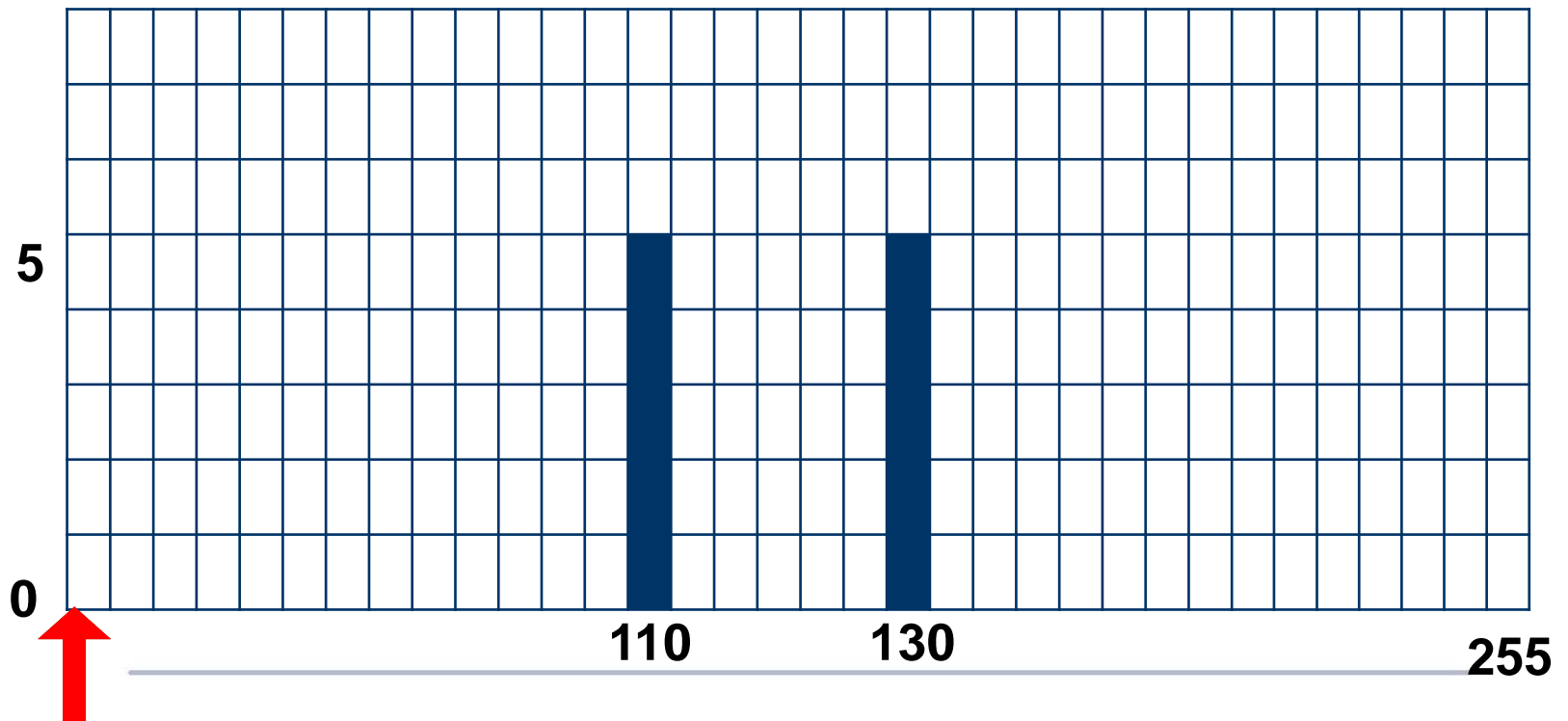
Getallenvoorbeeld

$$T_0 = 110$$

$$\text{MeanLeft} = (5 \times 110) / 5 = 110$$

$$\text{MeanRight} = (5 \times 130) / 5 = 130$$

$$T_1 = (110 + 130) / 2 = 120$$





Automatic Threshold – 2 Means

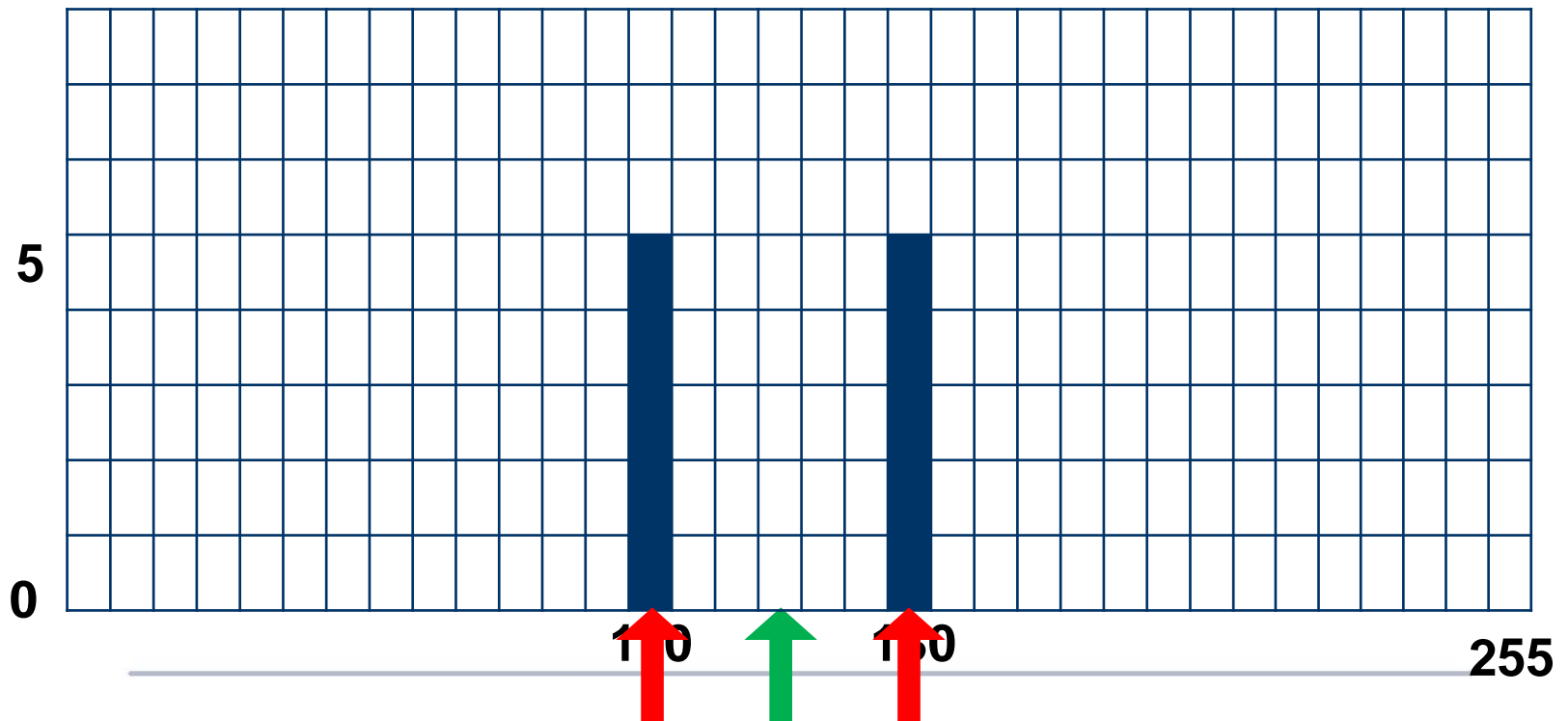
Getallenvoorbeeld

$$T_1 = 120$$

$$\text{MeanLeft} = (5 \times 110) / 5 = 110$$

$$\text{MeanRight} = (5 \times 130) / 5 = 130$$

$$T_2 = (110 + 130) / 2 = 120$$





Automatic Threshold – 2 Means

Opdracht

Implementeer de functies:

- `vThreshold()`
- `vHistogram()`
- `vThresholdIsoData()`

Hoeveel iteraties
zijn er nodig?

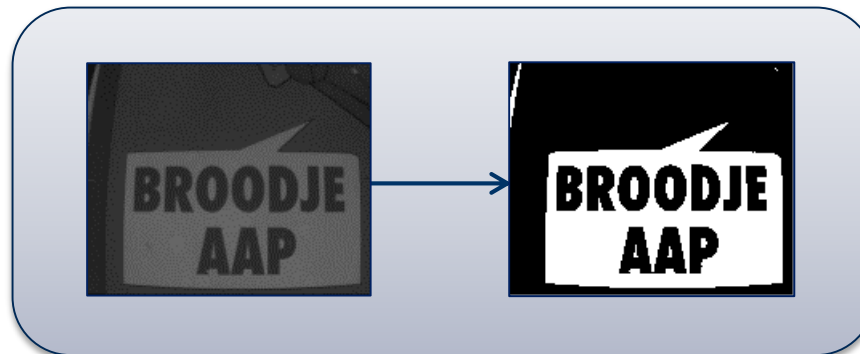


broodje_aap.bmp

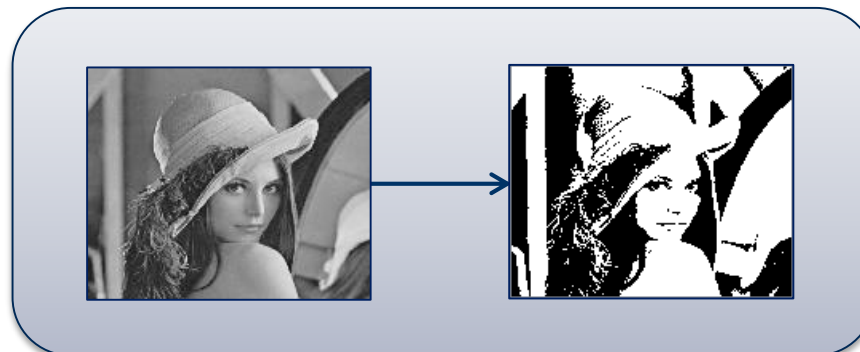


Automatic Threshold – 2 Means

Voorbeelden (zonder ContrastStretch()) ervoor:



$T_{ISODATA} = 69$



$T_{ISODATA} = 116$



Automatic Threshold – 2 Means

Conclusie:

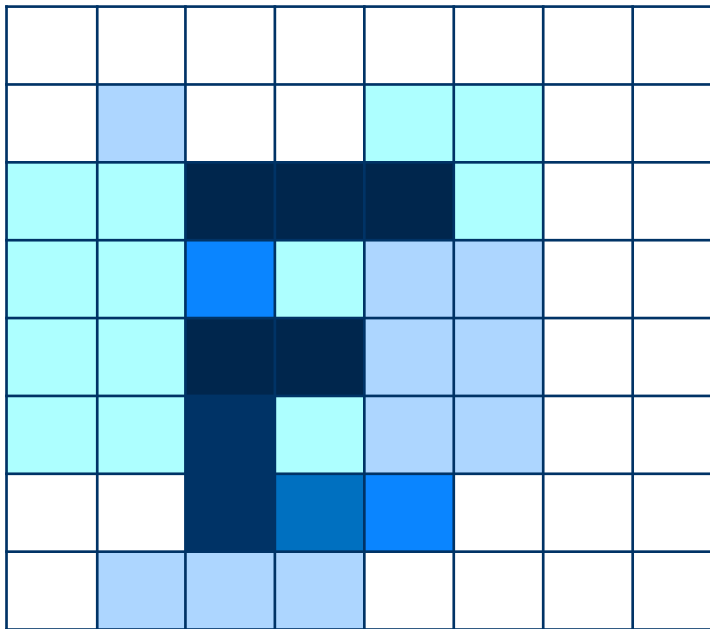
- Bewerkelijke operatie
- Werk op basis van het histogram
- Het aantal iteraties en dus de performance is afhankelijk van image

Vraag:

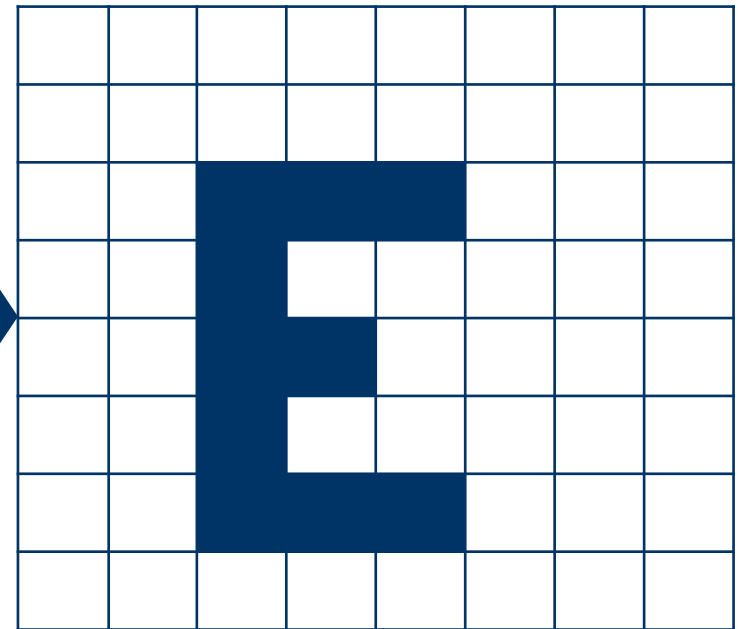
- Als de 'optimale' threshold value gevonden is, hoe kunnen dan donkere objecten binair worden gemaakt? En hoe de lichte objecten?



Automatic Threshold – Otsu



Source



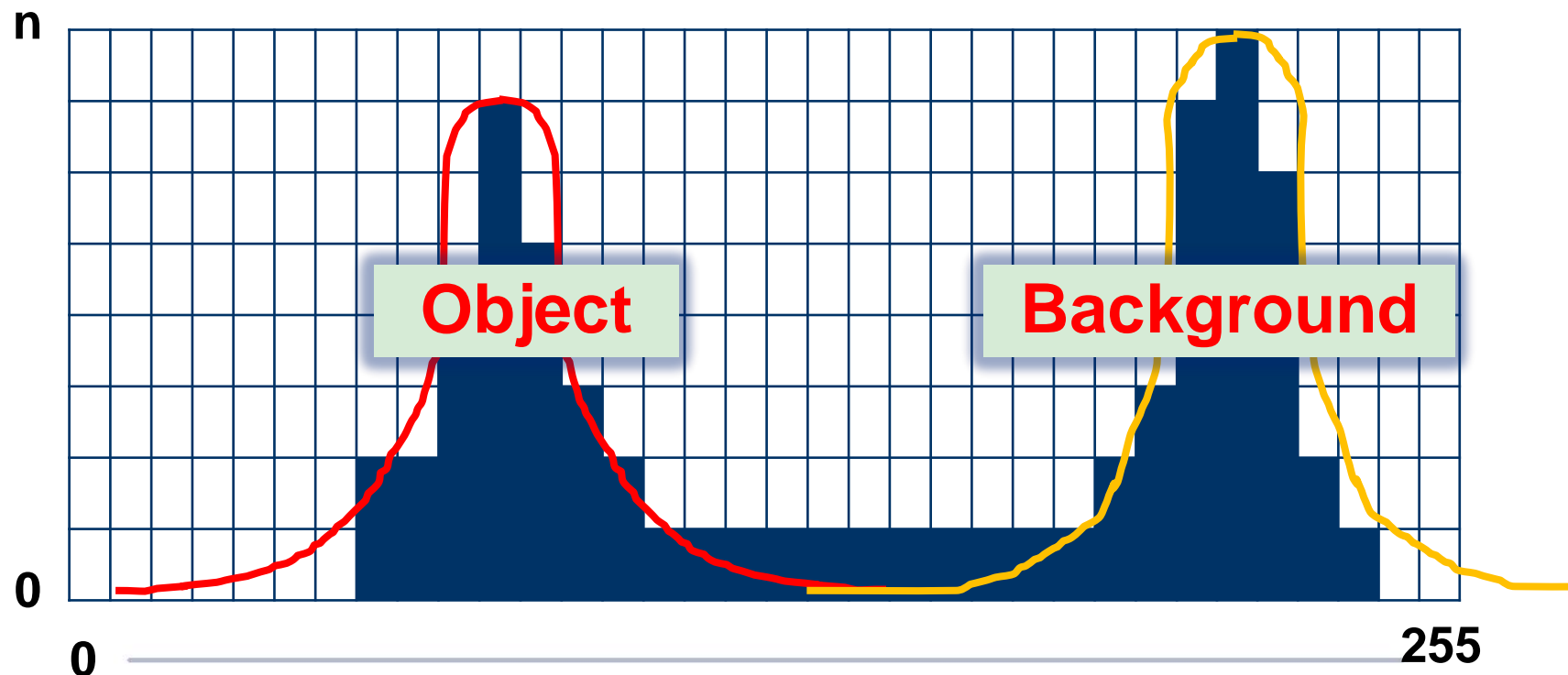
Destination



Automatic Threshold – Otsu

Uitgangspunt:

- object en background worden gezien als normaalverdelingen
- de twee 'beste' normaalverdelingen geven de 'beste' threshold





Automatic Threshold – Otsu

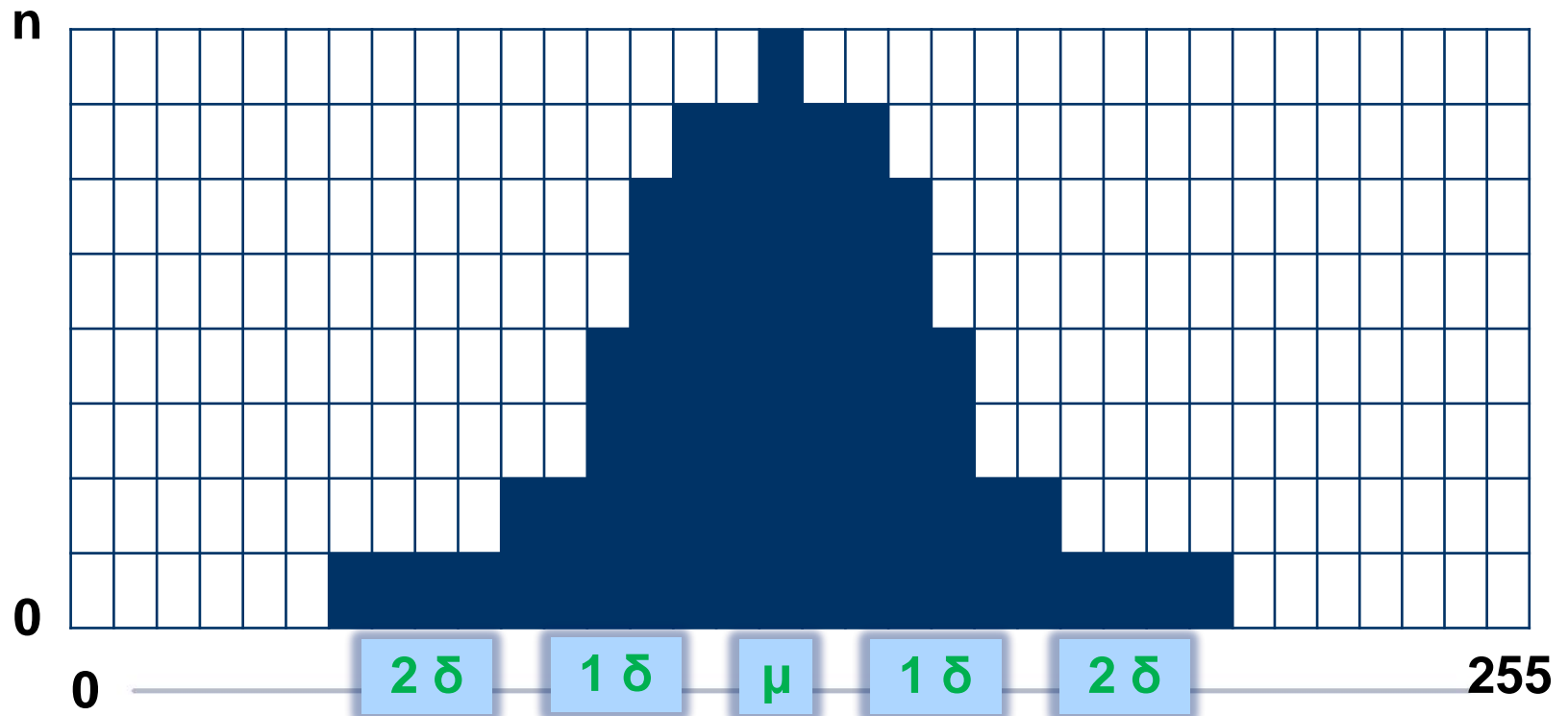
Optimale normaalverdeling:

μ - verwachtingswaarde (gemiddelde waarde)

δ - standaardafwijking

δ^2 - variantie is klein (mate waarin waarden onderling verschillen)

n - gewicht, aantal pixels





Automatic Threshold – Otsu

Bij de twee ‘beste’ normaalverdelingen is de som van de twee varianties zo klein mogelijk.

Otsu:

“De Between Class Variance (BCV) is dan zo hoog mogelijk:”

$$BCV(T) = \underbrace{n_{back}(T) \cdot n_{object}(T)}_{\text{Verdeling van de pixels}} \cdot \underbrace{(\mu_{back}(T) - \mu_{object}(T))^2}_{\text{Afstand tussen gemiddelden}}$$

Verdeling van de pixels

Bij gelijke verdeling is dit product zo hoog mogelijk, want

$$n \cdot n > (n - i)(n + i)$$

$$n^2 > n^2 - i^2$$

Afstand tussen gemiddelden

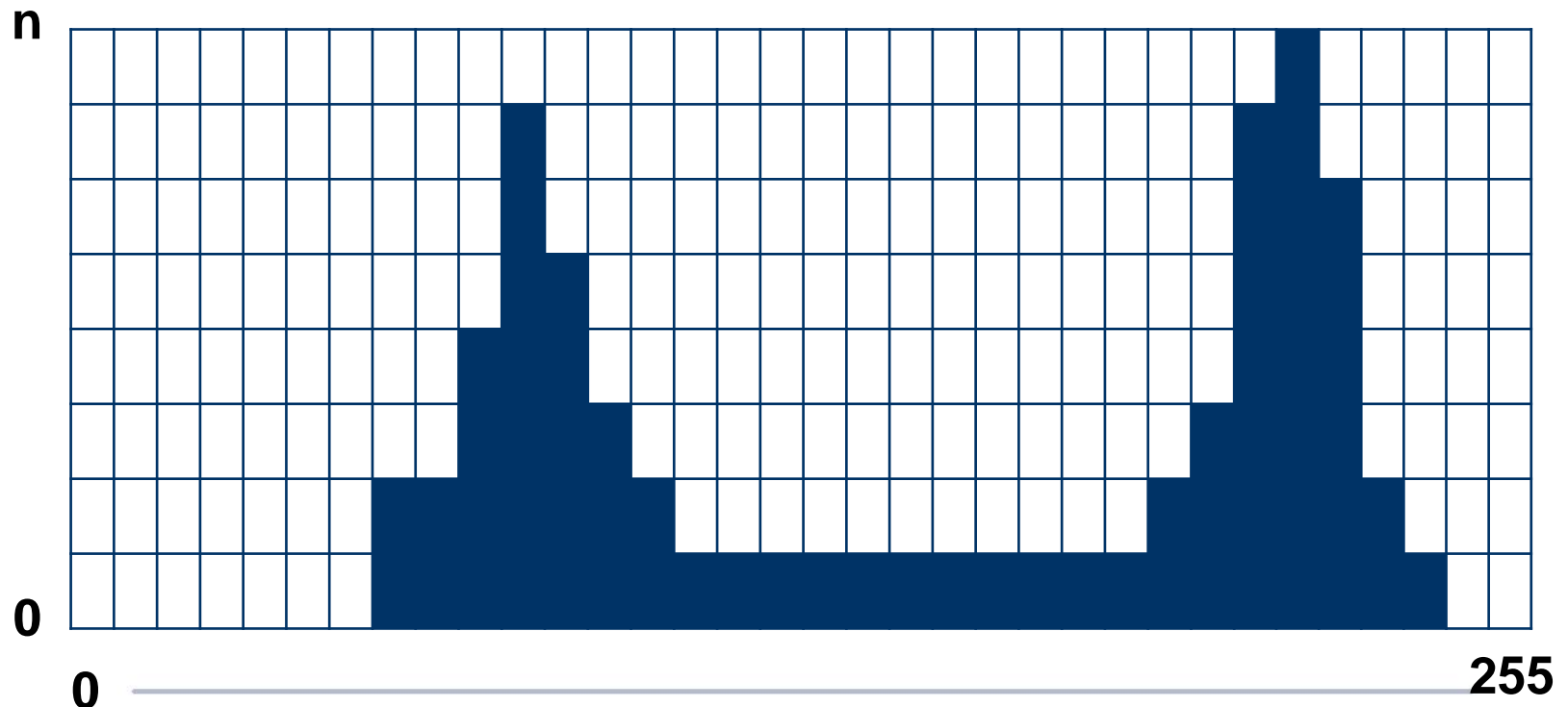
*Uitgedrukt in variantie,
Hoe groter de afstand, des te groter het getal*



Automatic Threshold – Otsu

Strategie:

- Bereken voor iedere mogelijke threshold (0-255) de BCV
- De hoogste BCV geeft de gewenste threshold

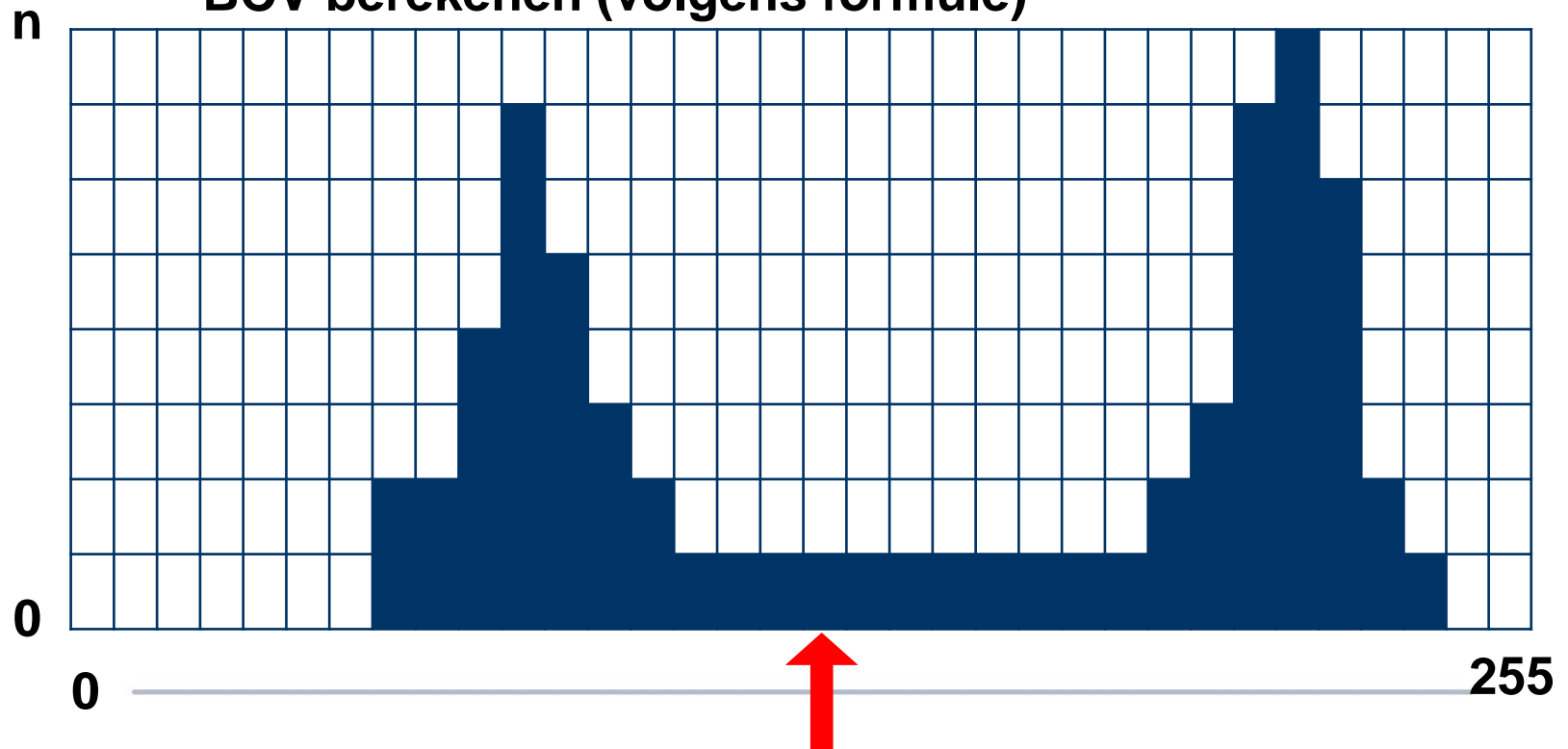




Automatic Threshold – Otsu

Veel rekenwerk, want voor iedere mogelijke threshold:

- Aantal pixels object bepalen
- Gem. waarde object berekenen
- Aantal pixels back bepalen
- Gem. waarde back berekenen
- BCV berekenen (volgens formule)

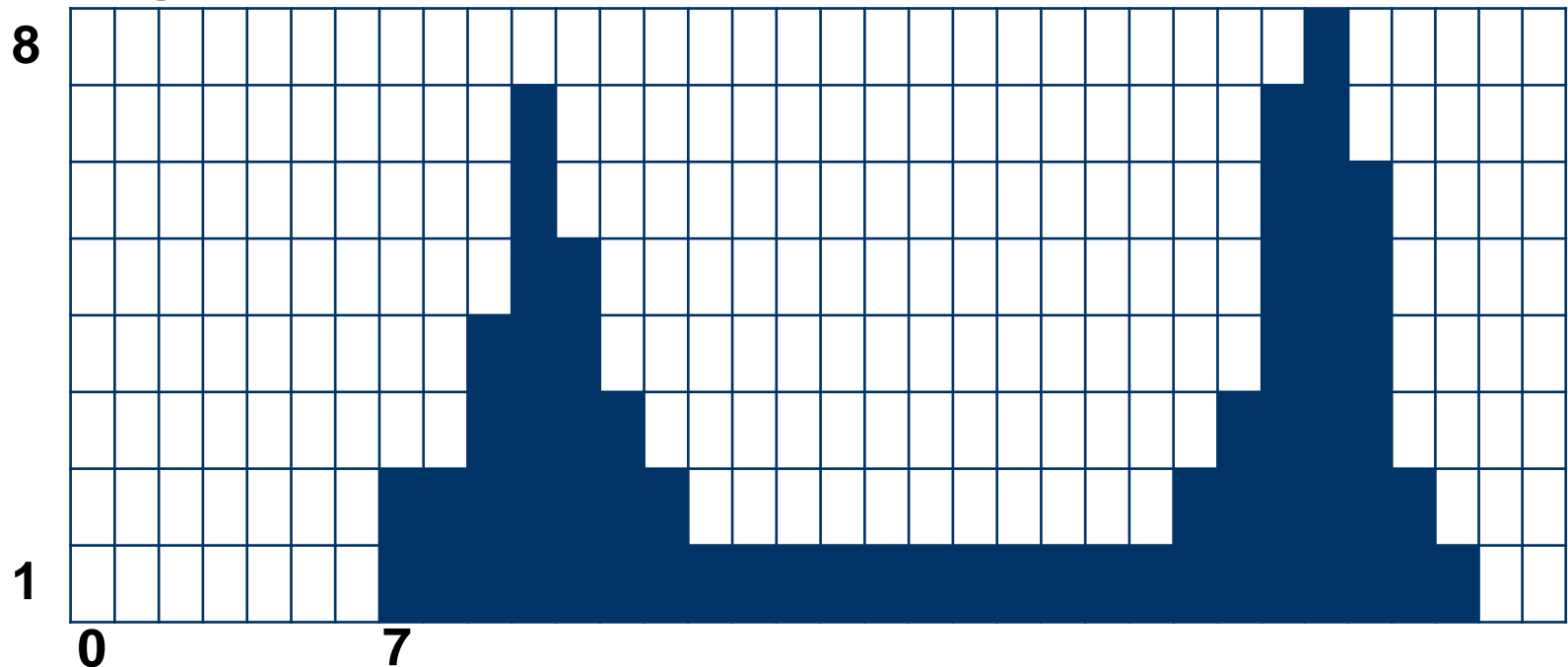




Automatic Threshold – Otsu

Uitgangspunt iteratief algoritme:

- Werken op basis van het histogram
- Het totaal aantal pixels verandert niet
- De totale som van de pixels verandert niet
- Wat er bij het object aan aantal en som bij komt, gaat van het totaal af

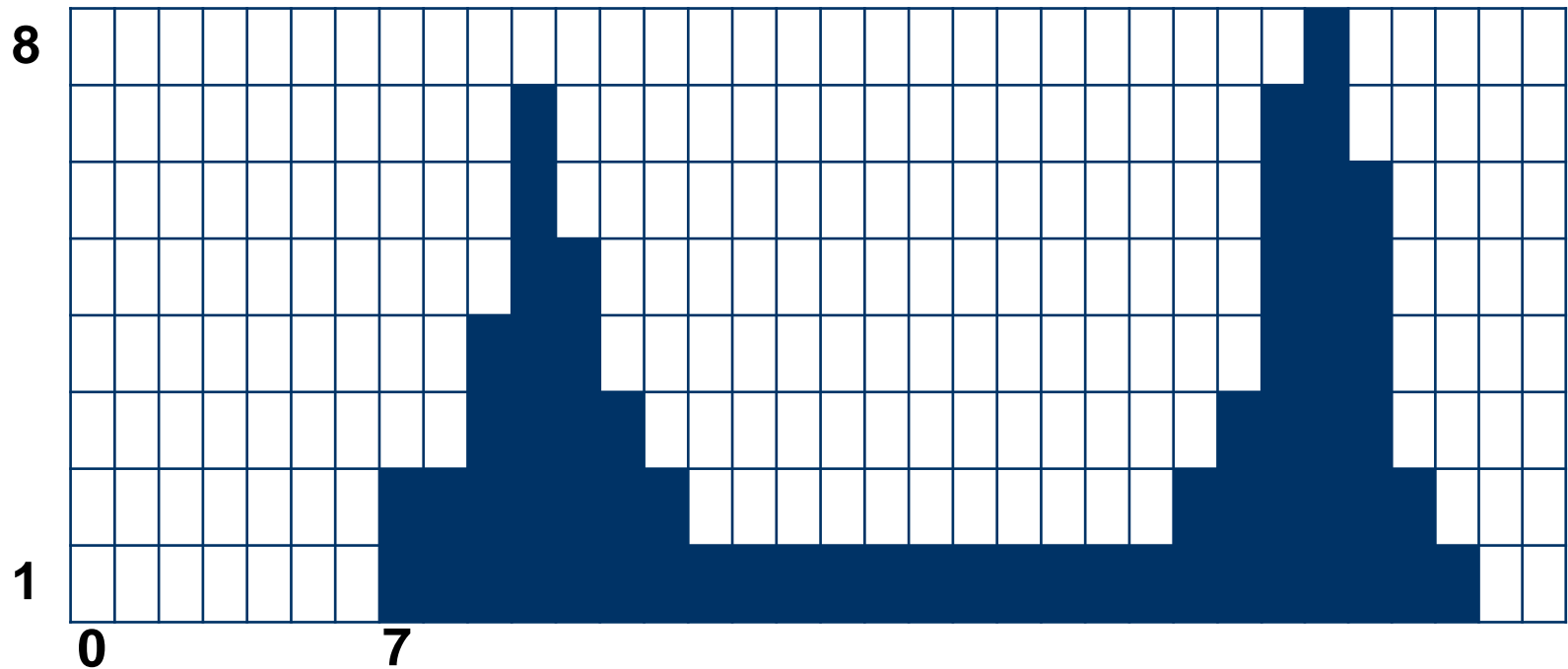




Automatic Threshold – Otsu

n totaal = 65

Som totaal = 1268





Automatic Threshold – Otsu

n totaal = 65

Som totaal = 1268

8

n object = 2

Som object = 14

Gem. object = 7

n back = 65-2 = 63

Som back = 1268-14 = 1254

Gem. back = 19.9

BCV = 20967.7

1

0

7



Automatic Threshold – Otsu

n totaal = 65

Som totaal = 1268

8

n object = 4

Som object = 30

Gem. object = 7.5

n back = 65-4 = 61

Som back = 1268-30 = 1238

Gem. back = 20.3

BCV = 39977.0

1

0

7





Automatic Threshold – Otsu

n totaal = 65

Som totaal = 1268

8

n object = 8

Som object = 66

Gem. object = 8.3

n back = 65-8 = 57

Som back = 1268-66 = 1202

Gem. back = 21.1

BCV = 74711.0

1

0

7





Automatic Threshold – Otsu

Iteratief algoritme:

Maak het histogram

Bereken het totale aantal pixels

Bereken de totale som van alle pixels

FOR Threshold = 0 t/m 255

aantal pixels van het object bepalen

som pixels van het object berekenen

gem. waarde pixels van het object berekenen

aantal pixels van de back bepalen

som pixels van de back berekenen

gem. waarde pixels van de back berekenen

BCV uitrekenen en max waarde controleren



Automatic Threshold – Otsu

Conclusie:

- Bewerkelijke operatie
- Werk op basis van het histogram
- Het aantal iteraties en dus de performance is voorspelbaar, aangezien alle mogelijke threshold values berekend worden

Opdracht

Implementeer de functie `vThresholdOtsu()`
Welke functie werkt sneller op de target?

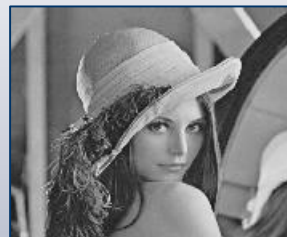


Automatic Threshold – Otsu

Voorbeelden (zonder ContrastStretch()) ervoor:



$$T_{\text{Otsu}} = 71$$



$$T_{\text{Otsu}} = 118$$