Embedded Vision Design

EVD1 - Week 4

Segmentation

By Hugo Arends

HAN_UNIVERSITY OF APPLIED SCIENCES

Segmentation

- Segments an image into various components for object recognition
- Goal: separate object from background
- Based on the graylevel histogram of an image
- Simple if the graylevels of an object are significantly different from the background

- Threshold
- Threshold 2 means
- Threshold Otsu

Threshold

- Used to separate object from background
- Converts a graylevel image to a binary image
- Thresholding is defined as

$$p_{dst}(x,y) = \begin{cases} g_o\ (\mathbf{1}) & T_{min} \leq p_{src}\ (x,y) \leq T_{max} \\ g_b\ (\mathbf{0}) & elsewhere \end{cases}$$

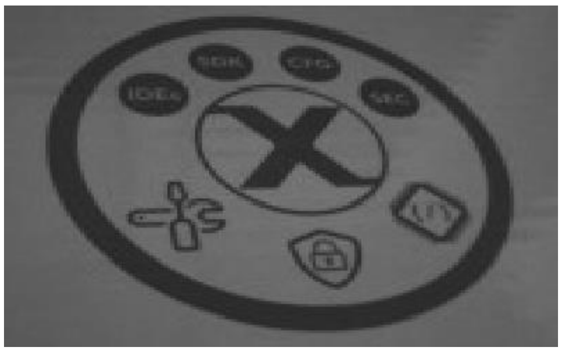
where

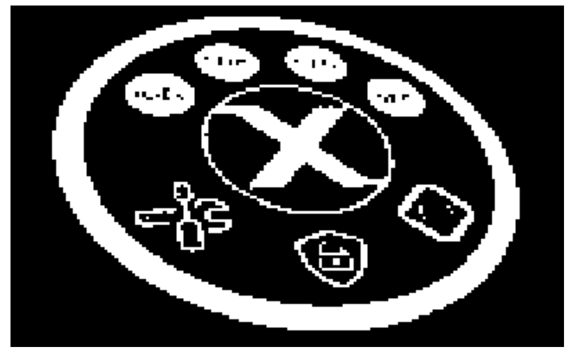
p_{src}: *original graylevel at position* (x,y)
p_{dst}: *new graylevel at position* (x,y)

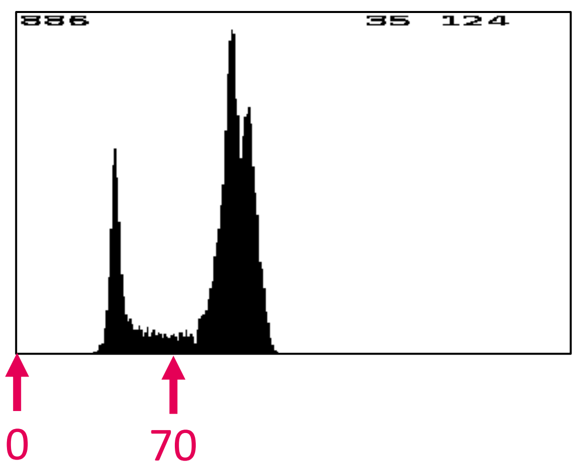
g_o: *object graylevel after thresholding:* ***1***
g_b: *background graylevel after thresholding:* ***0***
T_{min}: *minimum thresholding value*
T_{max}: *maximum thresholding value*

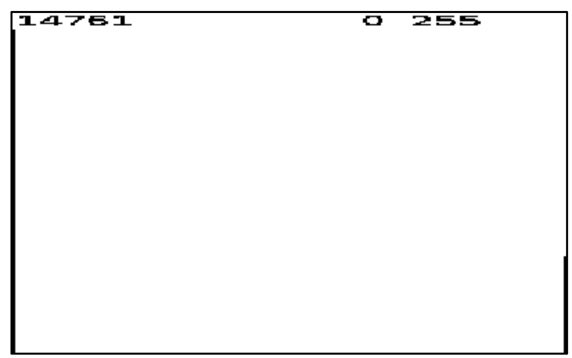
Threshold - example

$$T_{min} = 0$$
$$T_{max} = 70$$





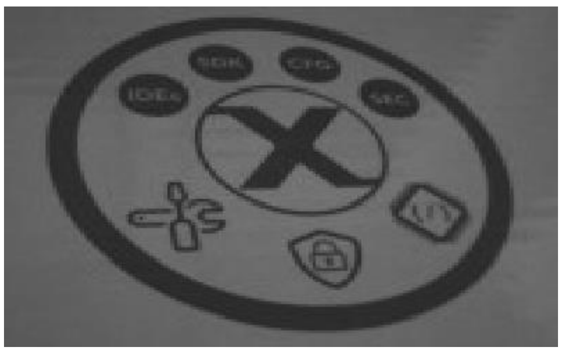


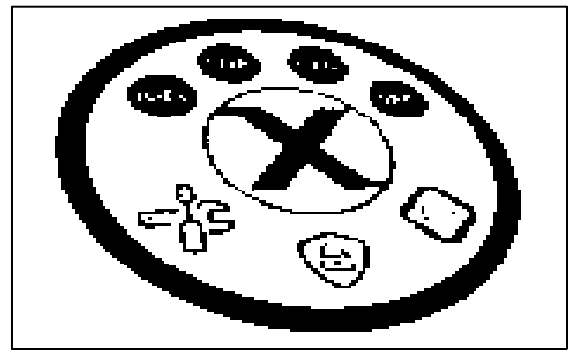


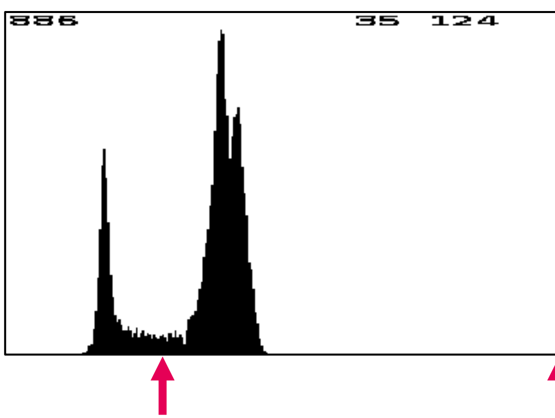
For visualization, the thresholded result is scaled!

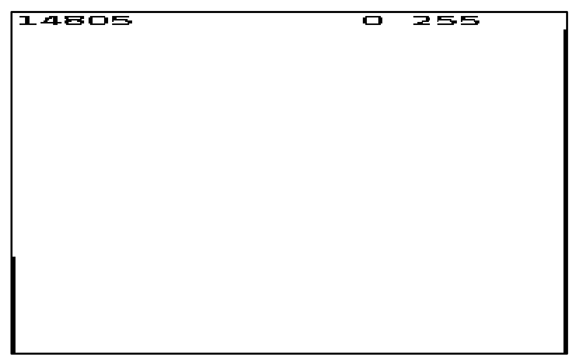
Threshold - example

$$T_{min} = 70$$
$$T_{max} = 255$$







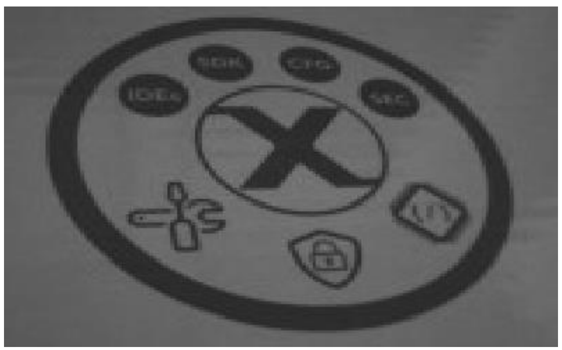


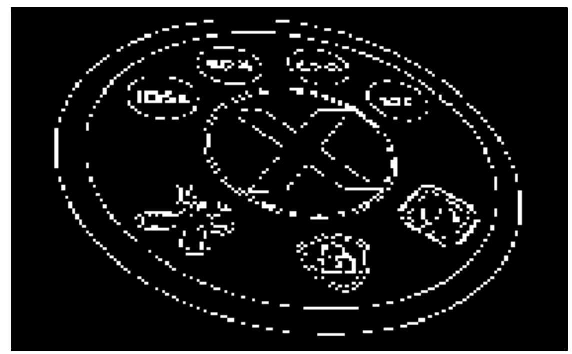
70 255

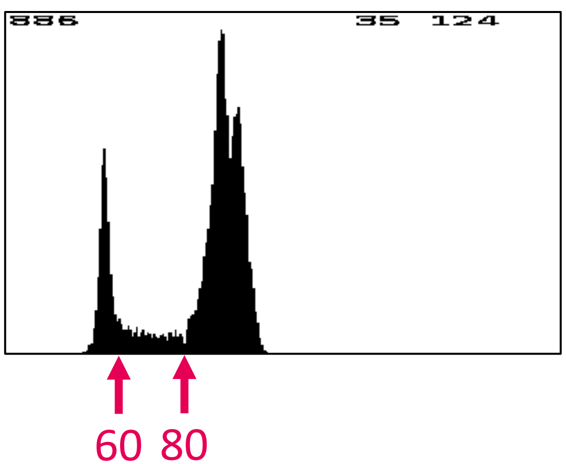
For visualization, the thresholded result is scaled!

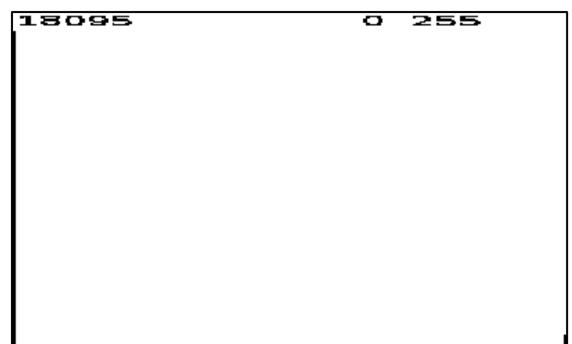
Threshold - example

$$T_{min} = 60$$
$$T_{max} = 80$$









For visualization, the thresholded result is scaled!

Threshold - algorithm

```
void threshold(    const image_t *src, image_t *dst,
                   const basic_pixel_t min, const basic_pixel_t max);
```

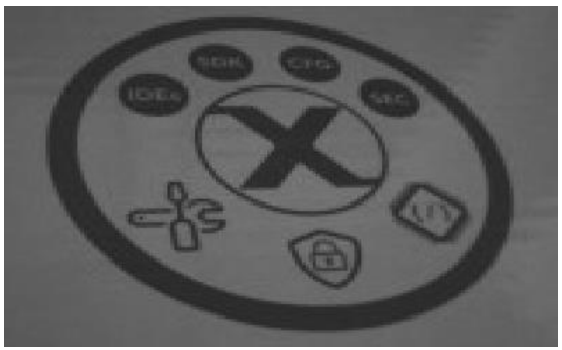
See file **EVDK_Operators\segmentation.c**

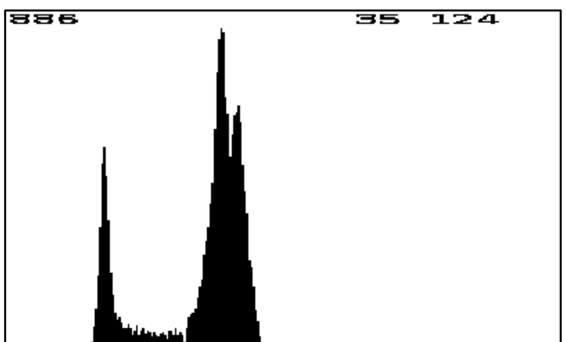
Threshold 2 Means

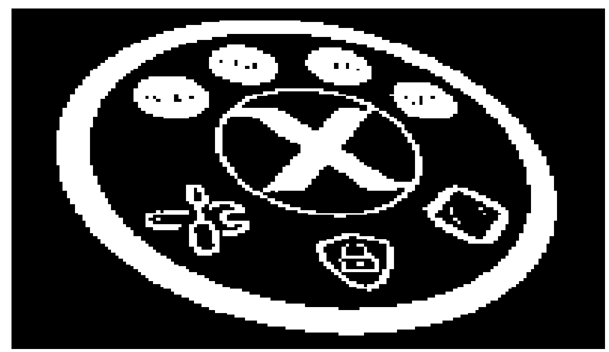
- Assumes that the histogram shows two clusters of pixels
- Where does Object stop and Background end?
- Goal: minimize the overlap
- Solution: iterative K-means algorithm

Threshold 2 Means - example

Automatically find the dark objects



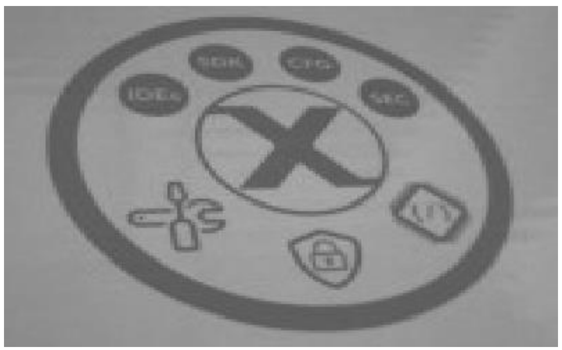


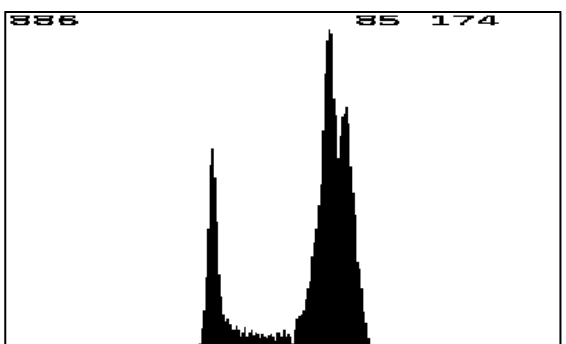


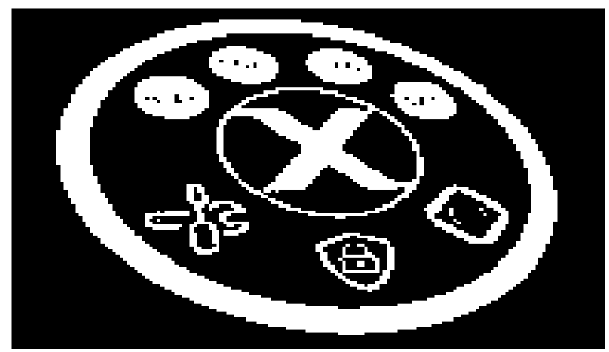
Threshold 2 Means - example

Automatically find the dark objects

Even when the lighting conditions change



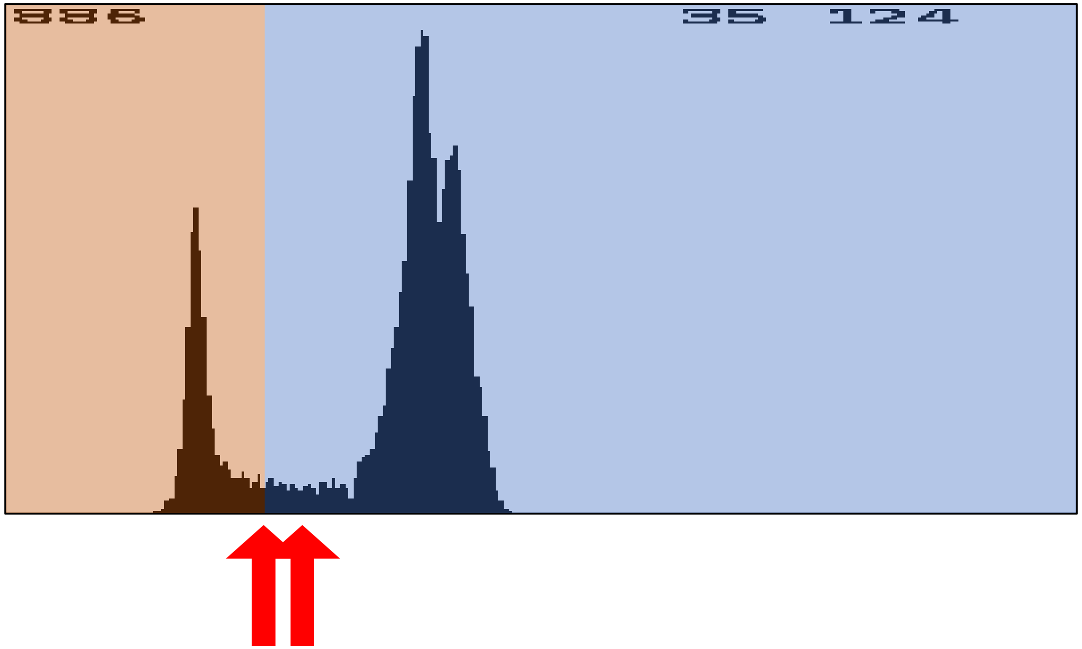




Threshold 2 Means - algorithm

Iterative 2-means algorithm:

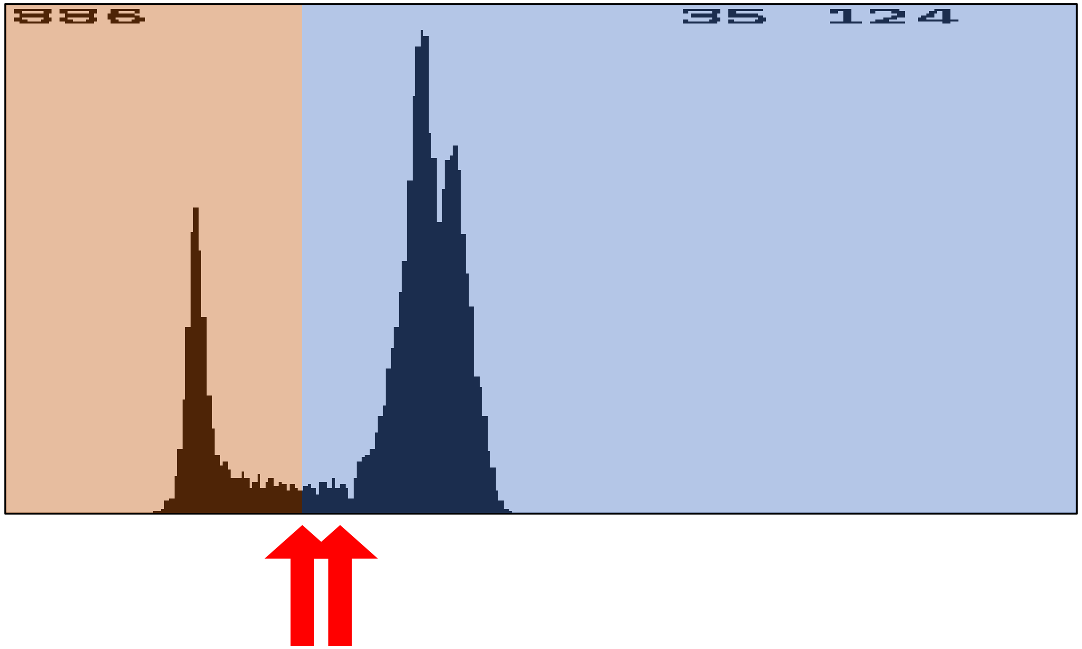
1. Set T to any value between low_pixel and high_pixel
2. Calculate mean value of all pixels to the left of T (mean left)
3. Calculate mean value of all pixels to the right of T (mean right)
4. Move T to the mean of mean left and mean right



Threshold 2 Means - algorithm

Iterative 2-means algorithm:

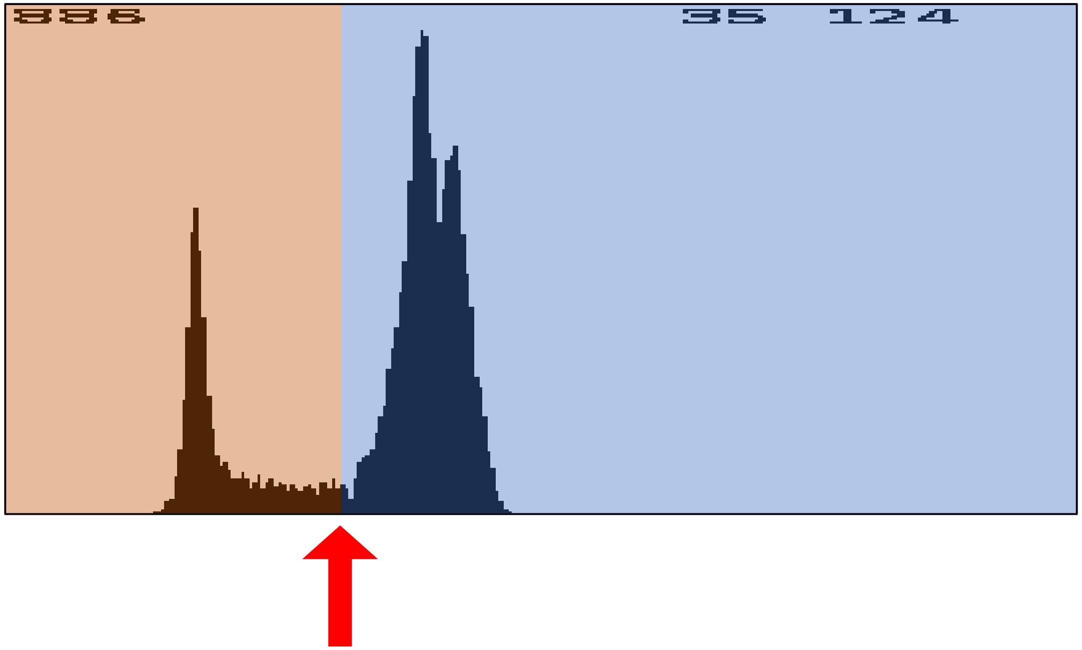
2. Calculate mean value of all pixels to the left of T (mean left)
3. Calculate mean value of all pixels to the right of T (mean right)
4. Move T to the mean of mean left and mean right



Threshold 2 Means - algorithm

Iterative 2-means algorithm:

5. Repeat as long as T changes



Threshold 2 Means - example

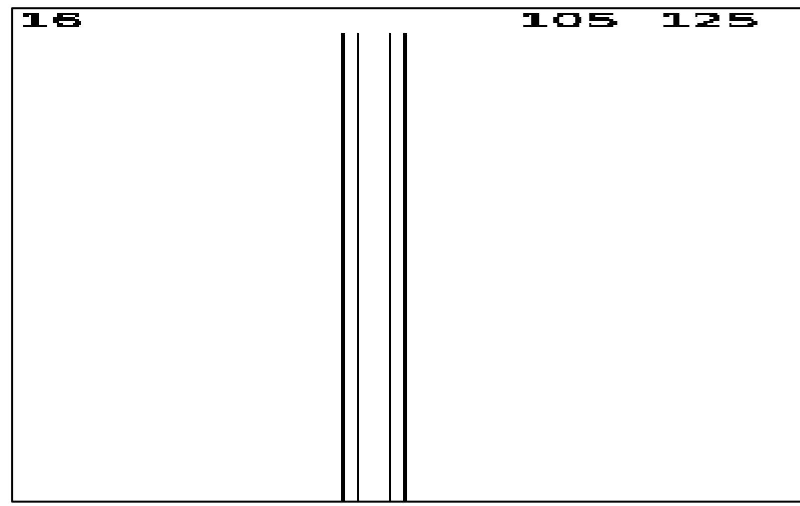
8x8 image

105	105	105	105	105	105	105	105
105	105	105	105	105	105	105	105
110	110	110	110	110	110	110	110
110	110	110	110	110	110	110	110
120	120	120	120	120	120	120	120
120	120	120	120	120	120	120	120
125	125	125	125	125	125	125	125
125	125	125	125	125	125	125	125

Threshold 2 Means - example

8x8 image

105	105	105	105	105	105	105	105
105	105	105	105	105	105	105	105
110	110	110	110	110	110	110	110
110	110	110	110	110	110	110	110
120	120	120	120	120	120	120	120
120	120	120	120	120	120	120	120
125	125	125	125	125	125	125	125
125	125	125	125	125	125	125	125



Threshold 2 Means - example

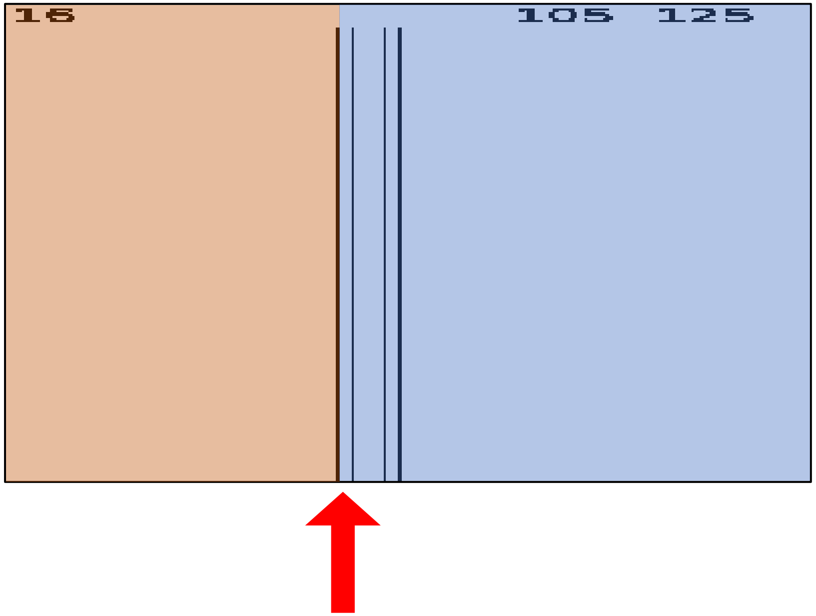
8x8 image

$T_0 = 105$

$$\text{mean left} = \frac{16 \times 105}{16} = 105.0$$

$$\text{mean right} = \frac{16 \times 110 + 16 \times 120 + 16 \times 125}{3 \times 16} = 118.3$$

$$T_1 = \frac{105.0 + 118.3}{2} = 112$$



HAN_UNIVERSITY
OF APPLIED SCIENCES

Threshold 2 Means - example

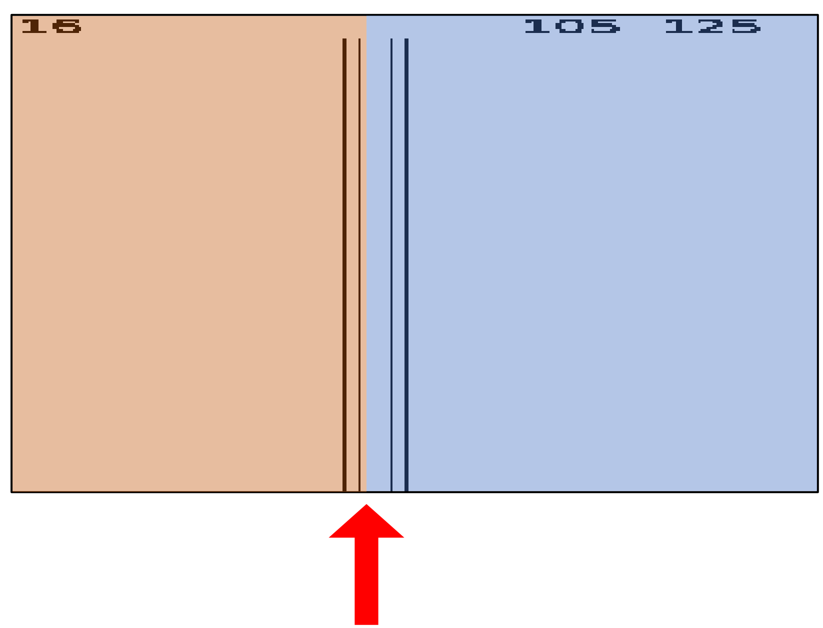
8x8 image

$T_1 = 112$

$$\text{mean left} = \frac{16 \times 105 + 16 \times 110}{2 \times 16} = 107.5$$

$$\text{mean right} = \frac{16 \times 120 + 16 \times 125}{2 \times 16} = 122.5$$

$$T_2 = \frac{107.5 \times 122.5}{2} = 115$$



Threshold 2 Means - example

8x8 image

$T_2 = 115$

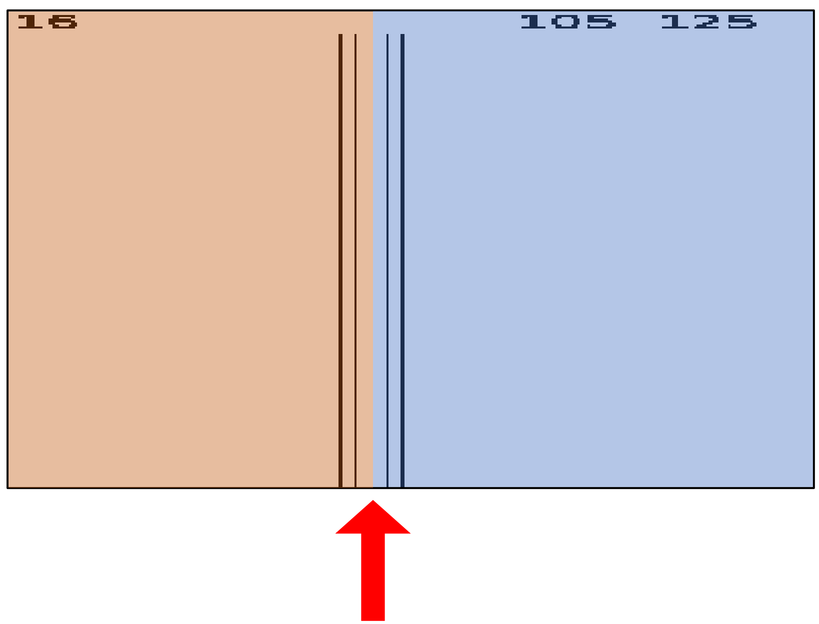
$$\text{mean left} = \frac{16 \times 105 + 16 \times 110}{2 \times 16} = 107.5$$

$$\text{mean right} = \frac{16 \times 120 + 16 \times 125}{2 \times 16} = 122.5$$

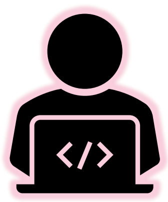
$$T_3 = \frac{107.5 \times 122.5}{2} = 115$$

Done!

Did not change



EVD1 – Assignment



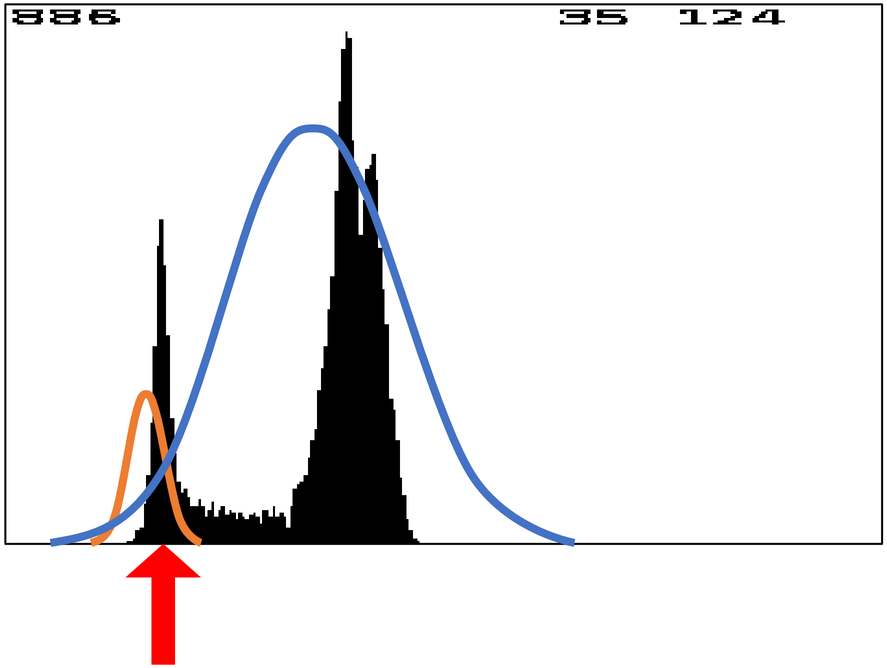
Study guide

Week 4

1 Segmentation – threshold2Means()

Threshold Otsu

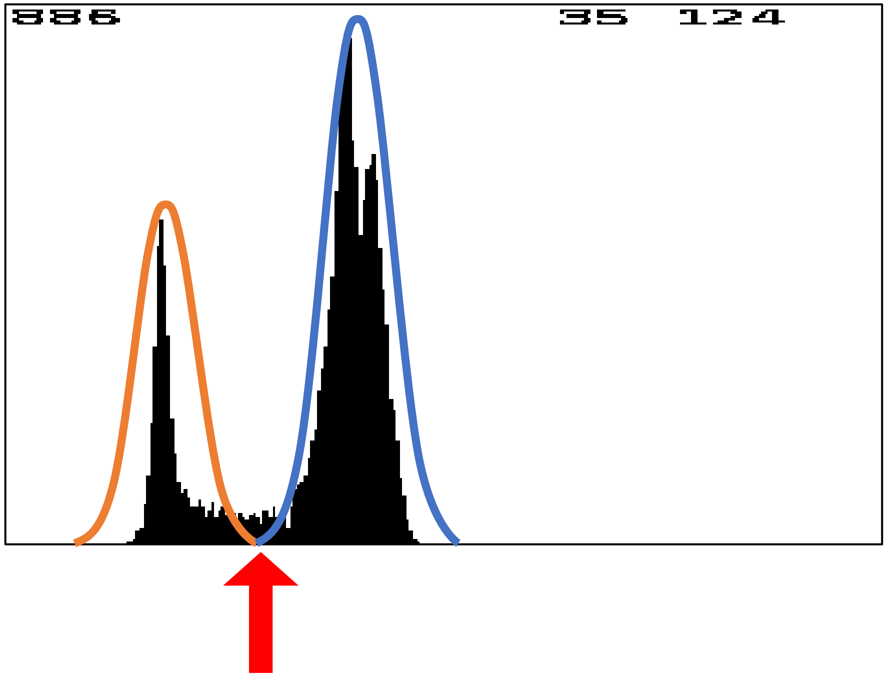
- Assumes that the histogram shows two clusters and that these clusters are normal distributions
- The threshold with the two ‘best’ normal distributions gives the optimum threshold



HAN_UNIVERSITY OF APPLIED SCIENCES

Threshold Otsu

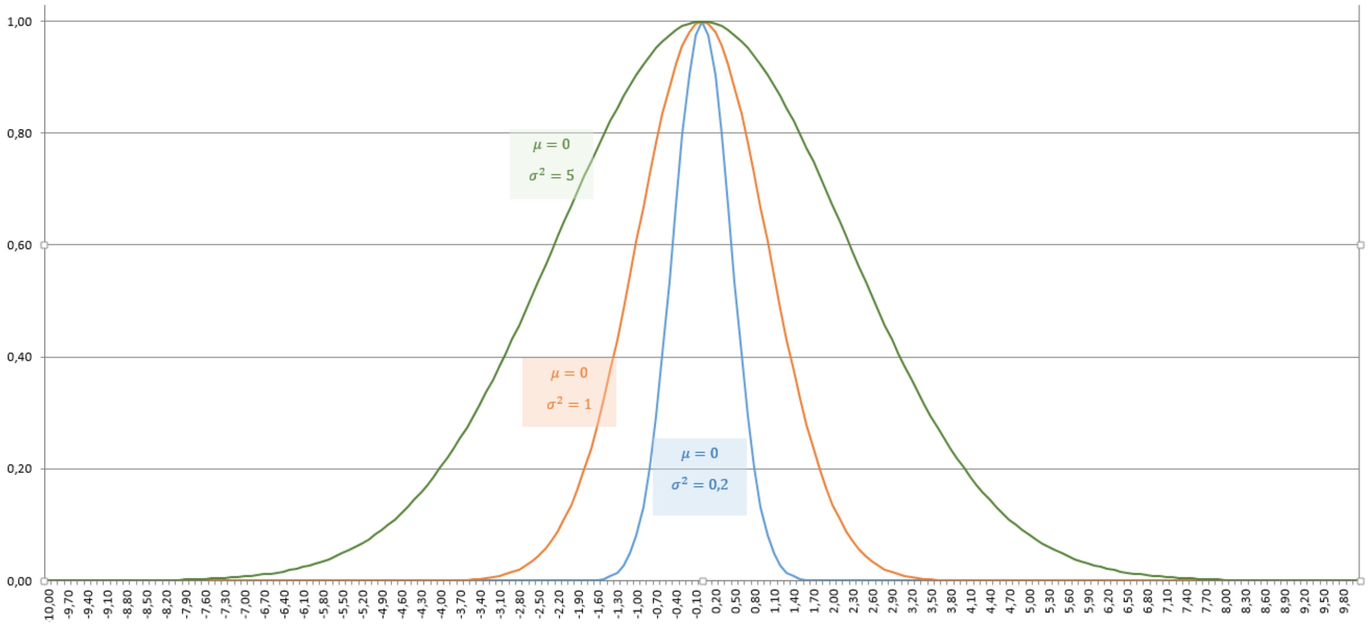
- Assumes that the histogram shows two clusters and that these clusters are normal distributions
- The threshold with the two ‘best’ normal distributions gives the optimum threshold



Threshold Otsu

Ideal normal distribution:

- μ - median value (mean value)
- δ - standard deviation
- δ^2 - variance (degree to which values differ)
- n - weight, number of pixels



Threshold Otsu - algorithm

The two ‘best’ normal distributions have the lowest sum of variances

Otsu:
“The Between Class Variance (BCV) is as high as possible:”

$$BCV(T) = \underbrace{n_{left}(T) \cdot n_{right}(T)}_{\textit{Distribution of the pixels}} \cdot \underbrace{(\mu_{left}(T) - \mu_{right}(T))^2}_{\textit{Distance between means}}$$

An equal distribution has the highest value:

The further the distance, the higher this product.

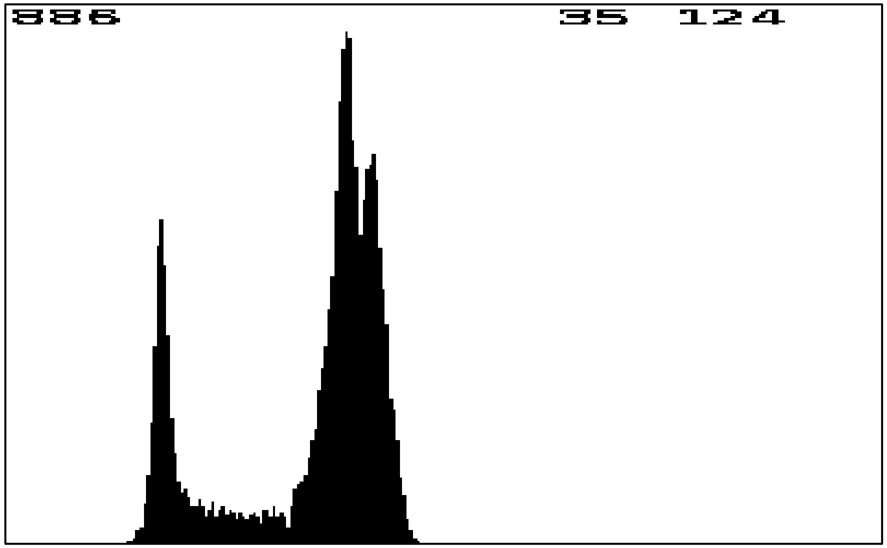
$$n \cdot n > (n-i)(n+i)$$

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

Threshold Otsu - algorithm

Strategy:

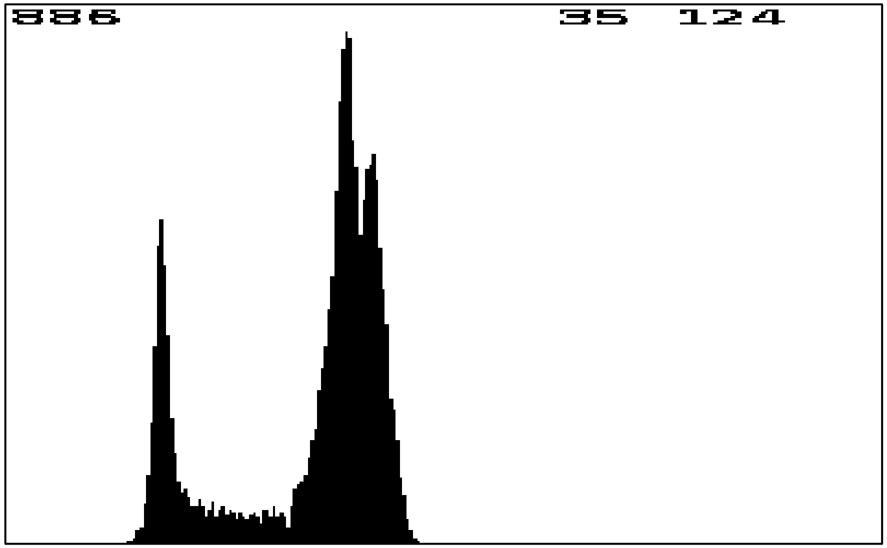
- Calculate for every possible threshold (T), which is every grayscale value between 0-255, the BCV
- The highest BCV gives the optimum threshold



Threshold Otsu - algorithm

This requires a lot calculations for each possible threshold (T):

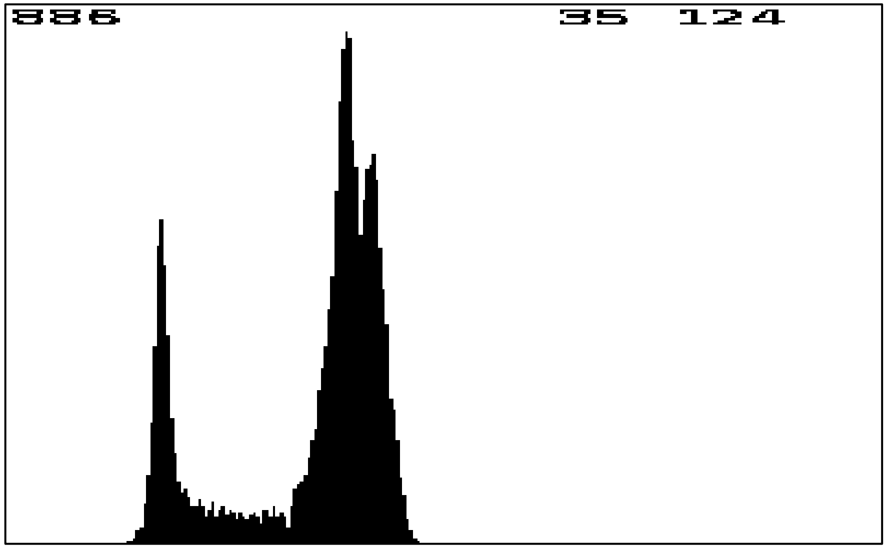
- Determine number of pixels to the left of T
- Calculate mean pixel value to the left of T
- Determine number of pixels to the right of T
- Calculate mean pixel value to the right of T
- Calculate BCV (using Otsu’s equation)



Threshold Otsu - algorithm

Instead, we'll use an iterative algorithm that uses the following properties:

- The total number of pixels in the image does not change
- The total sum of pixels in the image does not change
- Values added to the left, are removed from the right



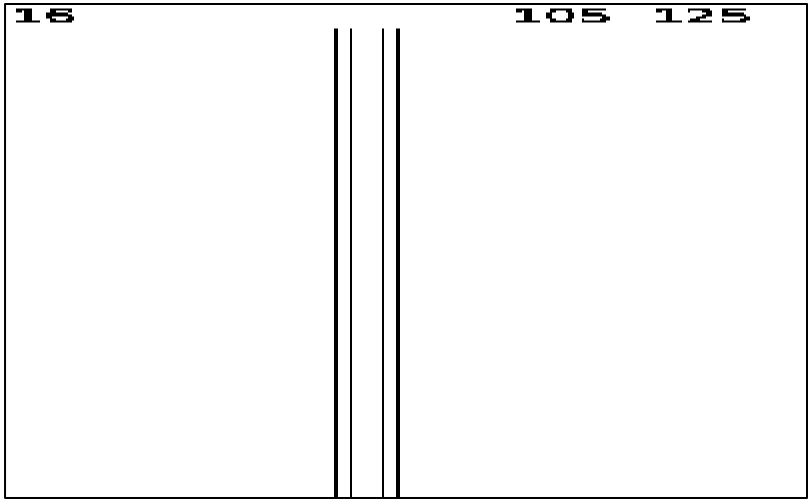
HAN_UNIVERSITY OF APPLIED SCIENCES

Threshold Otsu - example

8x8 image

$n_{total} = 8 \times 8 = 64$
$sum_{total} = 16 \times 105 + 16 \times 110 + 16 \times 120 + 16 \times 125 = 7360$

105	105	105	105	105	105	105	105
105	105	105	105	105	105	105	105
110	110	110	110	110	110	110	110
110	110	110	110	110	110	110	110
120	120	120	120	120	120	120	120
120	120	120	120	120	120	120	120
125	125	125	125	125	125	125	125
125	125	125	125	125	125	125	125



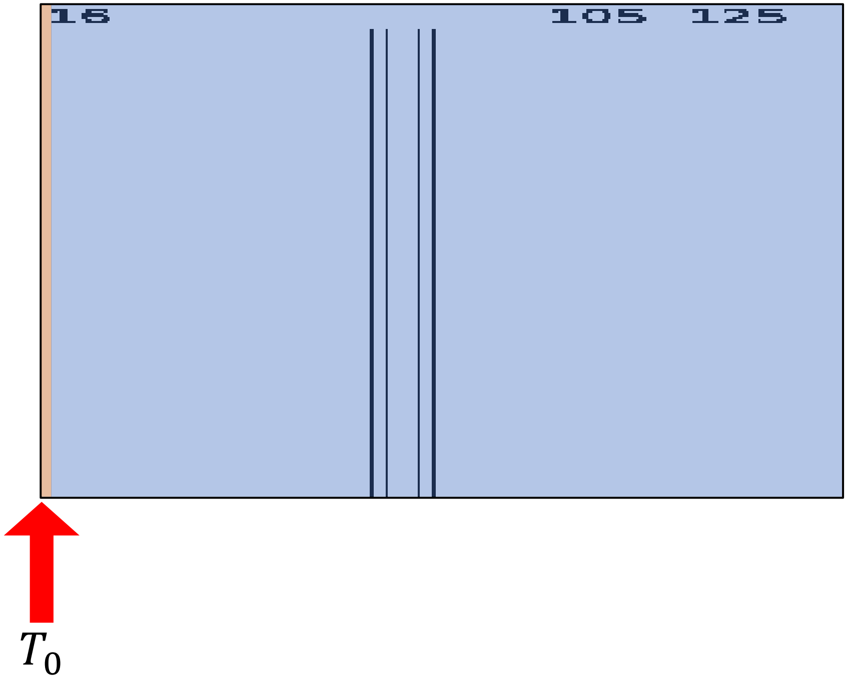
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$$\boldsymbol{BCV = 0.0}$$

$n_{left} = 0$
$sum_{left} = 0$
$mean_{left} = 0$



$n_{right} = 64 - 0 = 64$
$sum_{right} = 7360 - 0 = 7360$
$mean_{right} = 115$

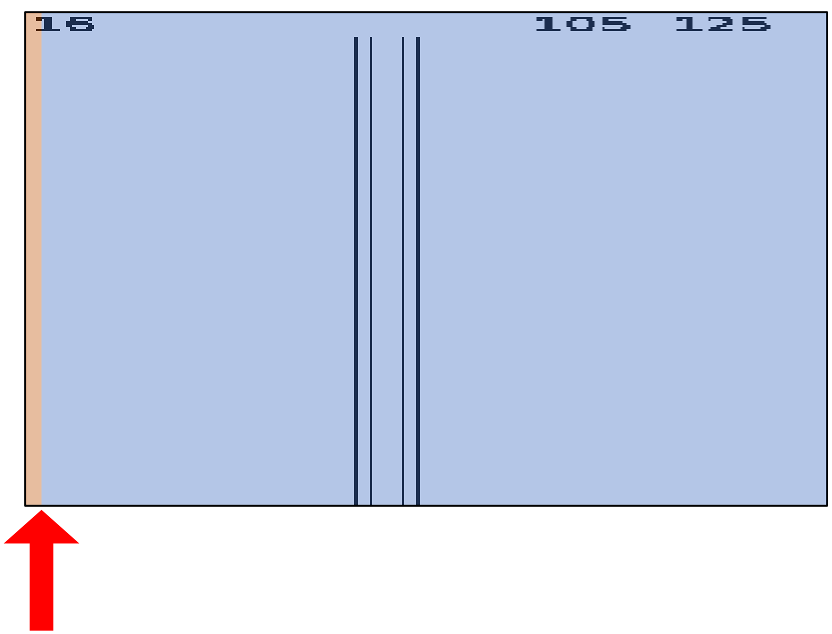
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 0.0}$

$n_{left} = 0$
$sum_{left} = 0$
$mean_{left} = 0$



T_1

$n_{right} = 64 - 0 = 64$
$sum_{right} = 7360 - 0 = 7360$
$mean_{right} = 115$

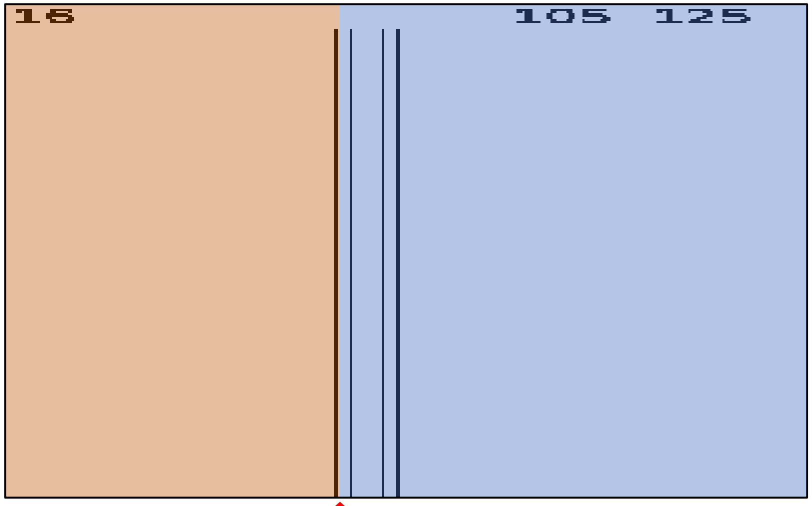
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 136533.39}$

$n_{left} = 16$
$sum_{left} = 1680$
$mean_{left} = 105$



T_{105}

$n_{right} = 64 - 16 = 48$
$sum_{right} = 7360 - 1680 = 5680$
$mean_{right} = 118.3$

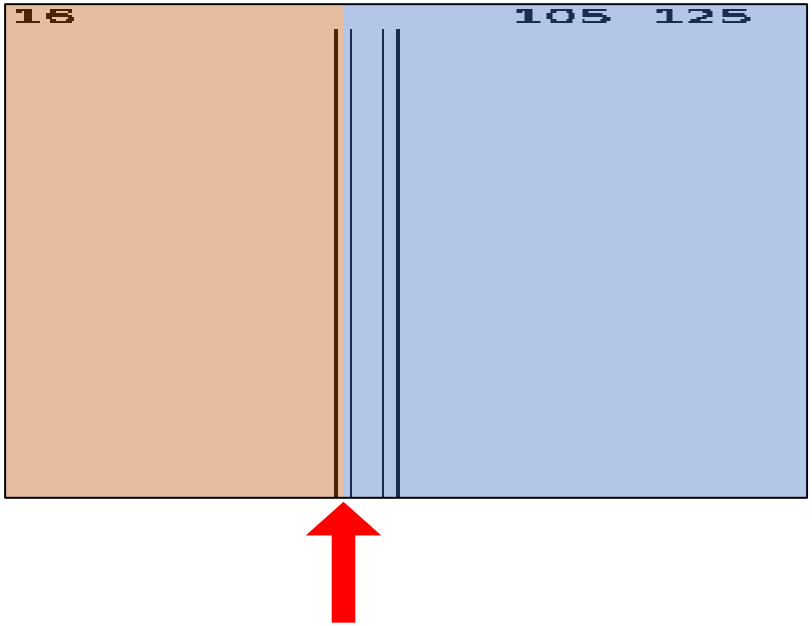
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 136533.39}$

$n_{left} = 16$
$sum_{left} = 1680$
$mean_{left} = 105$



T_{106}

$n_{right} = 64 - 16 = 48$
$sum_{right} = 7360 - 1680 = 5680$
$mean_{right} = 118.3$

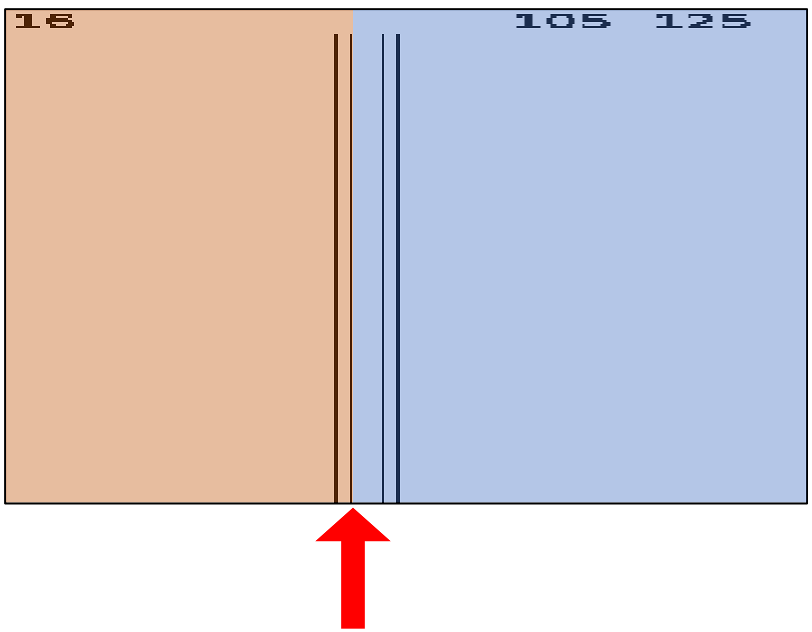
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 230400.0}$

$n_{left} = 32$
$sum_{left} = 3440$
$mean_{left} = 107.5$



$n_{right} = 64 - 32 = 32$
$sum_{right} = 7360 - 3440 = 3920$
$mean_{right} = 122.5$

T_{110}

HAN_UNIVERSITY OF APPLIED SCIENCES

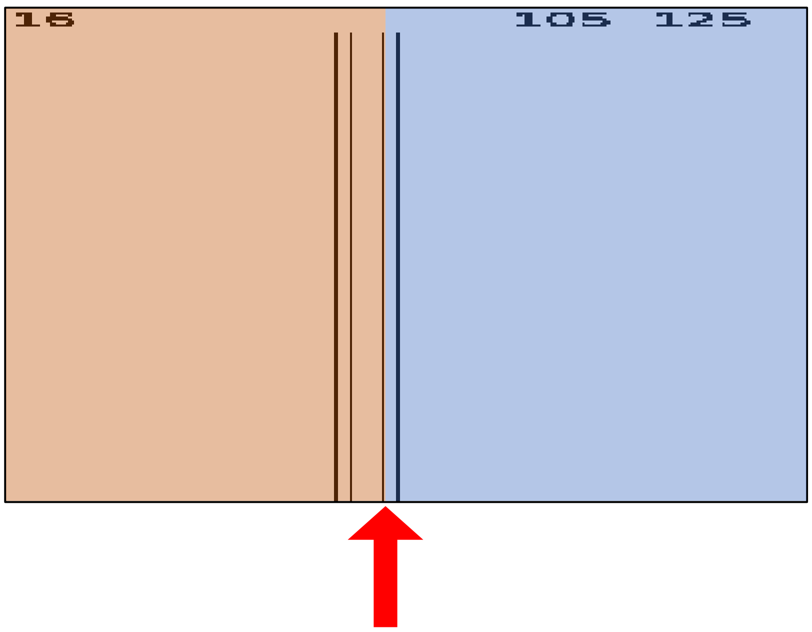
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 136533.39}$

$n_{left} = 48$
$sum_{left} = 5360$
$mean_{left} = 111.7$



T_{120}

$n_{right} = 64 - 48 = 16$
$sum_{right} = 7360 - 5360 = 2000$
$mean_{right} = 125$

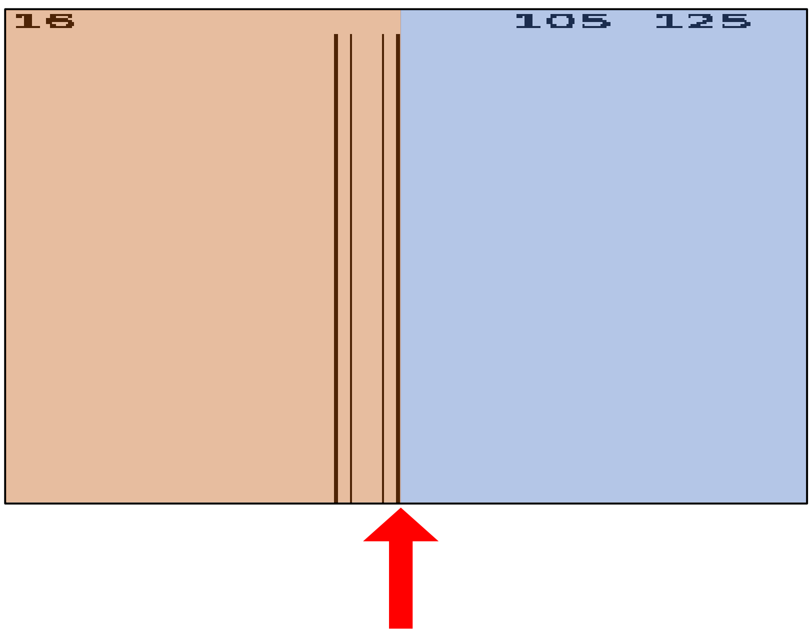
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 0.0}$

$n_{left} = 64$
$sum_{left} = 7360$
$mean_{left} = 115$



T_{125}

$n_{right} = 64 - 64 = 0$
$sum_{right} = 7360 - 7360 = 0$
$mean_{right} = 0$

HAN_UNIVERSITY OF APPLIED SCIENCES

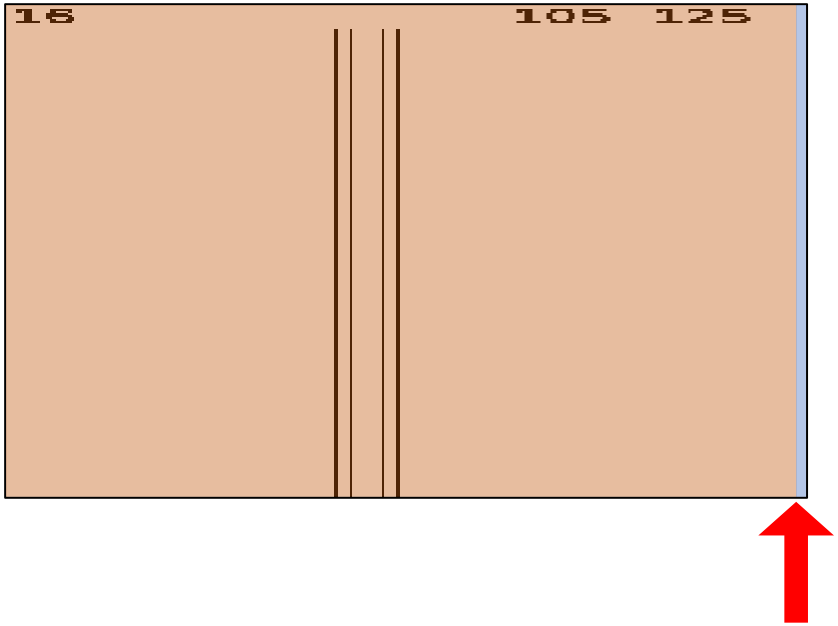
Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 0.0}$

$n_{left} = 64$
$sum_{left} = 7360$
$mean_{left} = 115$



T_{255}

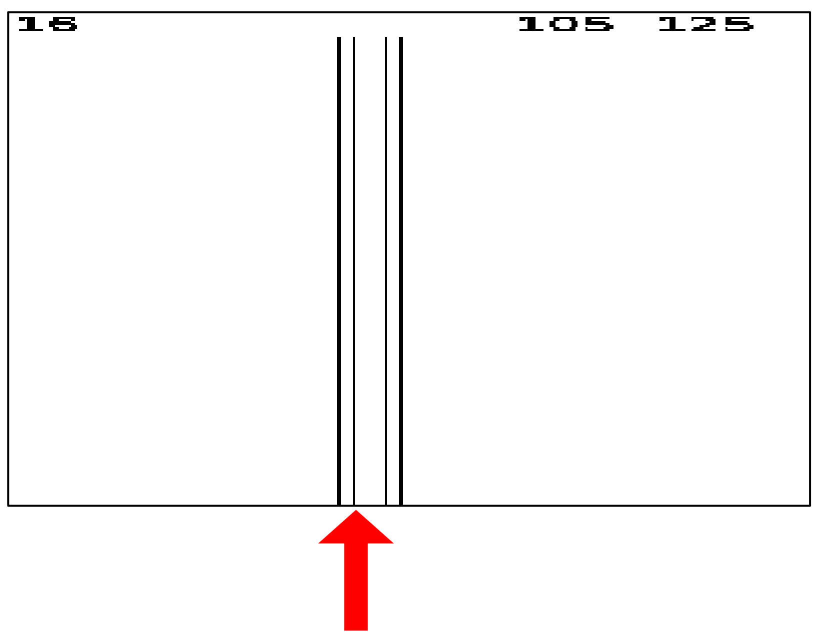
$n_{right} = 64 - 64 = 0$
$sum_{right} = 7360 - 7360 = 0$
$mean_{right} = 0$

Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 230400.0}$



T_{110}

Note.

$\boldsymbol{T_{110}}$ to $\boldsymbol{T_{119}}$ all have the same, highest BCV

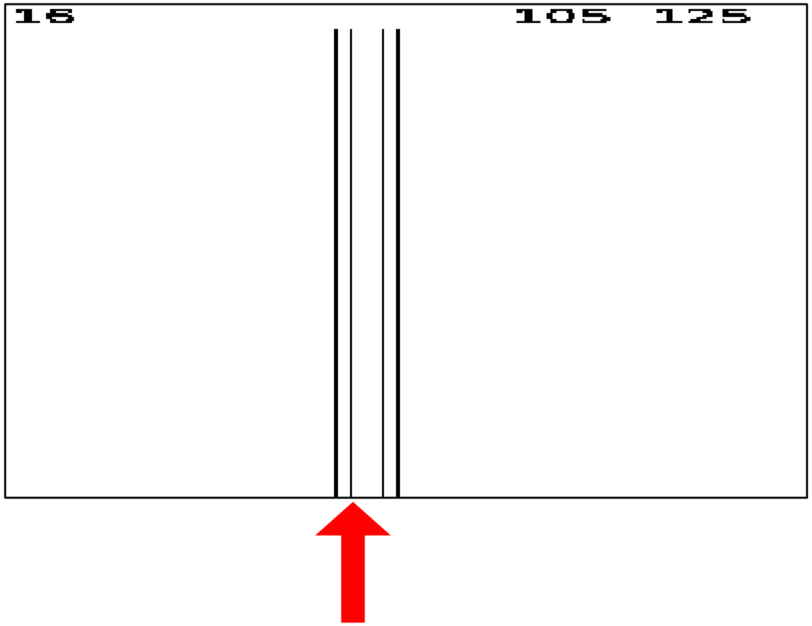
*$\boldsymbol{T_{110}}$ was the **first** with the highest BCV value*

Threshold Otsu - example

8x8 image

$n_{total} = 64$
$sum_{total} = 7360$

$\boldsymbol{BCV = 230400.0}$



T_{110}

Note.

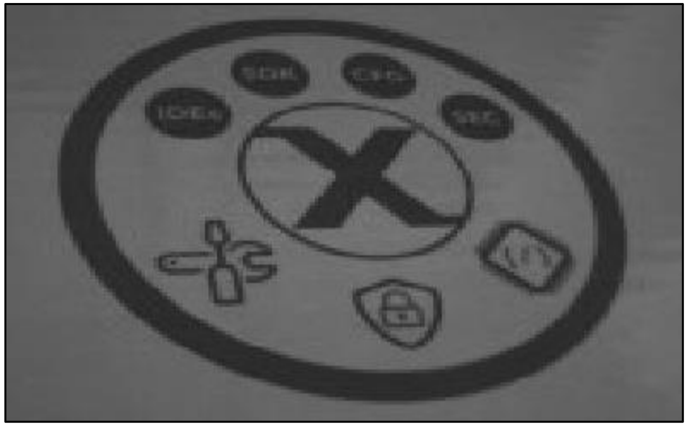
$\boldsymbol{T_{110}}$ to $\boldsymbol{T_{119}}$ all have the same, highest BCV

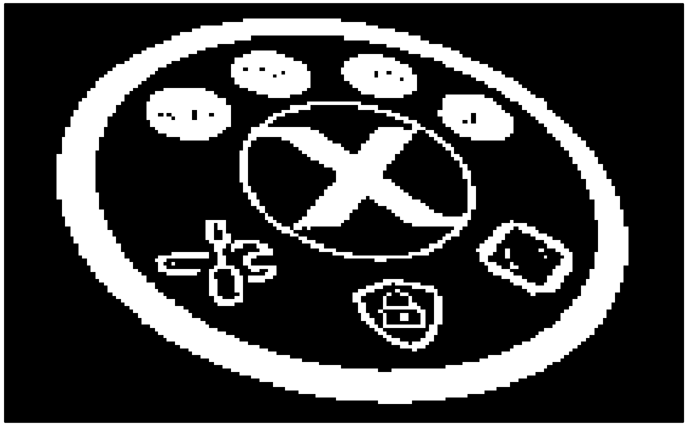
*$\boldsymbol{T_{110}}$ was the **first** with the highest BCV value*

And it is different from T_{115} from the threshold 2 means method

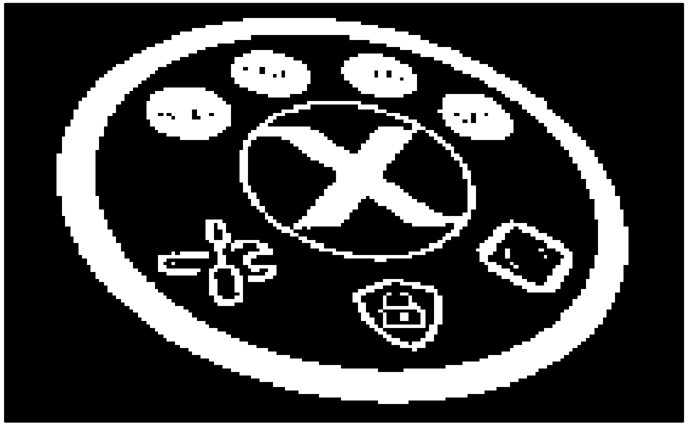
HAN_UNIVERSITY OF APPLIED SCIENCES

Threshold Otsu vs Threshold 2 Means



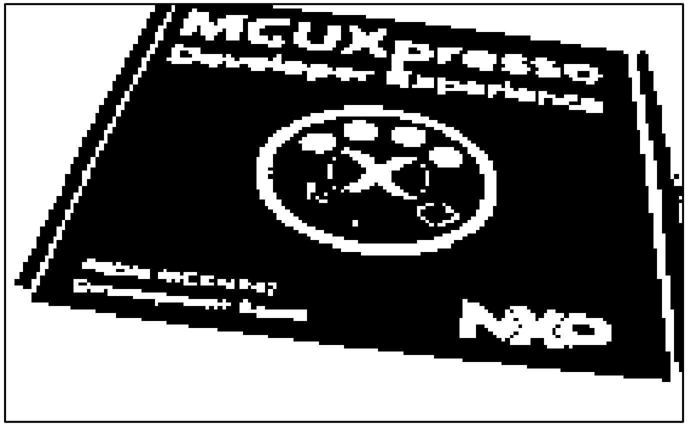


threshold2Means()

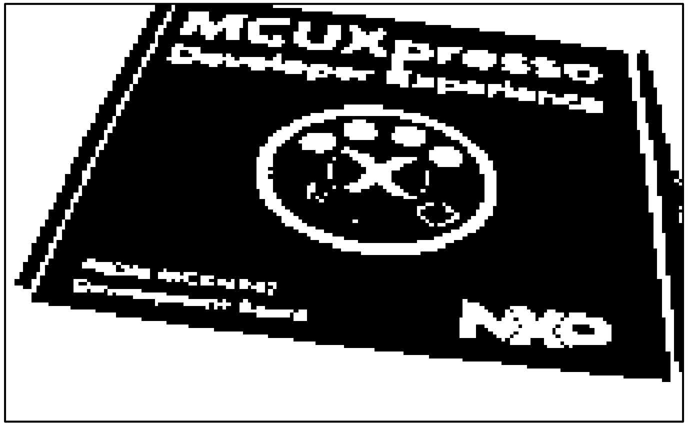


thresholdOtsu()

Threshold Otsu vs Threshold 2 Means



threshold2Means()



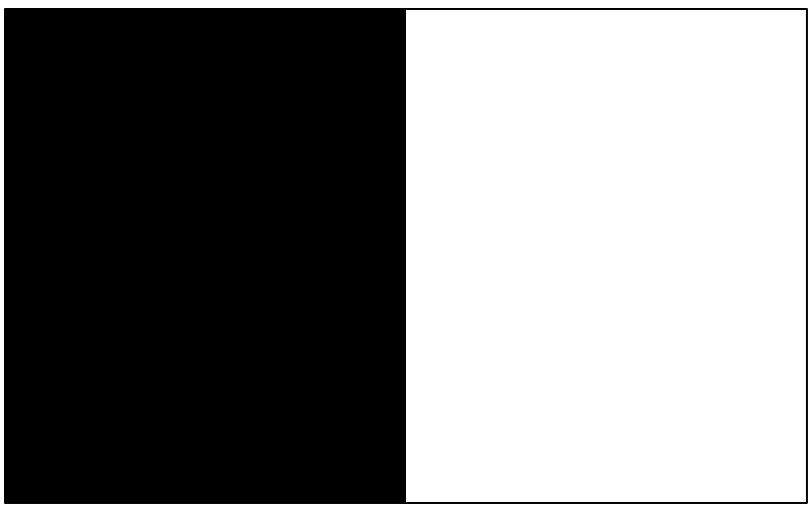
thresholdOtsu()

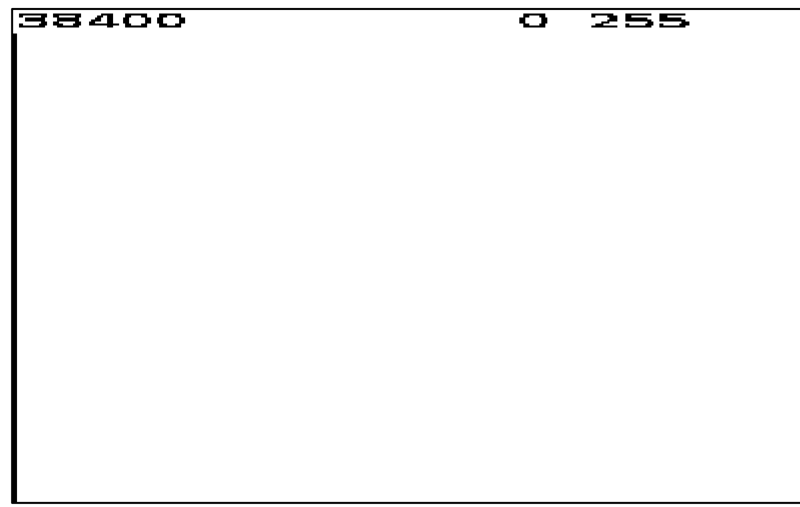
HAN_UNIVERSITY
OF APPLIED SCIENCES

Threshold Otsu - algorithm

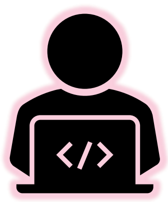
- What datatypes will you be using?

  Hint: Have a look at the Otsu thresholded image. What is the theoretic EVDK hardware maximum value for the calculated BCV value?





EVD1 – Assignment



Study guide

Week 4

2 Segmentation – thresholdOtsu()

HAN_UNIVERSITY OF APPLIED SCIENCES

References

- Myler, H. R., & Weeks, A. R. (2009). *The pocket handbook of image processing algorithms in C*. Prentice Hall Press.

- Morse, B. (2000). *Lecture 4: Thresholding*. Retrieved June 17, 2020, from http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf

- Gonzales, R. C., & Woods, R. E. (2002). *Digital image processing.*

HAN_UNIVERSITY OF APPLIED SCIENCES