



EVD1 – Vision operators

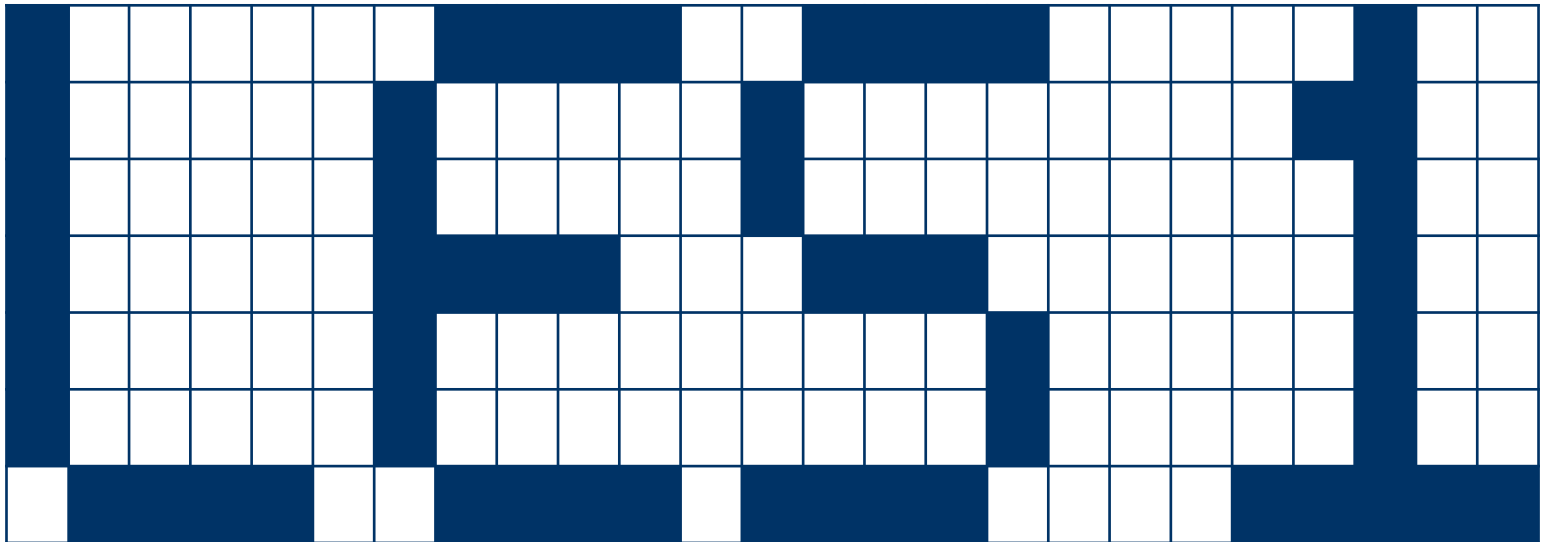




Image basics

Wij werken met uint8_t images:

```
// Image attributes  
#define IMG_HEIGHT (144)  
#define IMG_WIDTH (176)
```

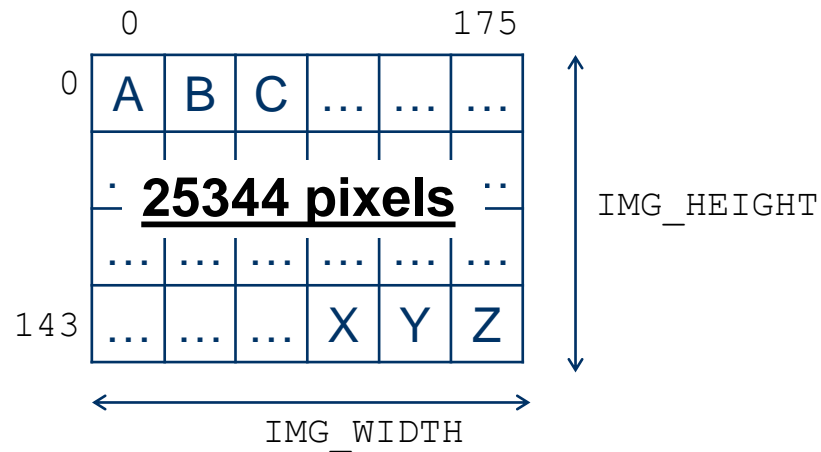




Image basics

Een image bestaat uit meer dan een afbeelding:


```
typedef struct
{
    uint16_t width;
    uint16_t height;
    uint16_t lut;
    uint16_t dummy; // power of 32 (for DMA)
    uint8_t data[IMG_HEIGHT][IMG_WIDTH];
}image_t;
```

	0					175
0	A	B	C

143	X	Y	Z

Geheugen:

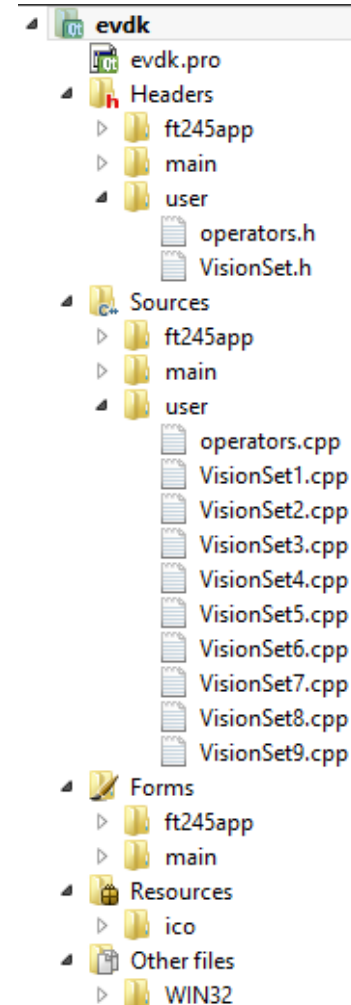
width		height		lut		dummy		data (25344 bytes)											
								A	B	C	X	Y	Z	

 1 byte



Development basics

Uitleg file structuur in Qt Creator





Development basics

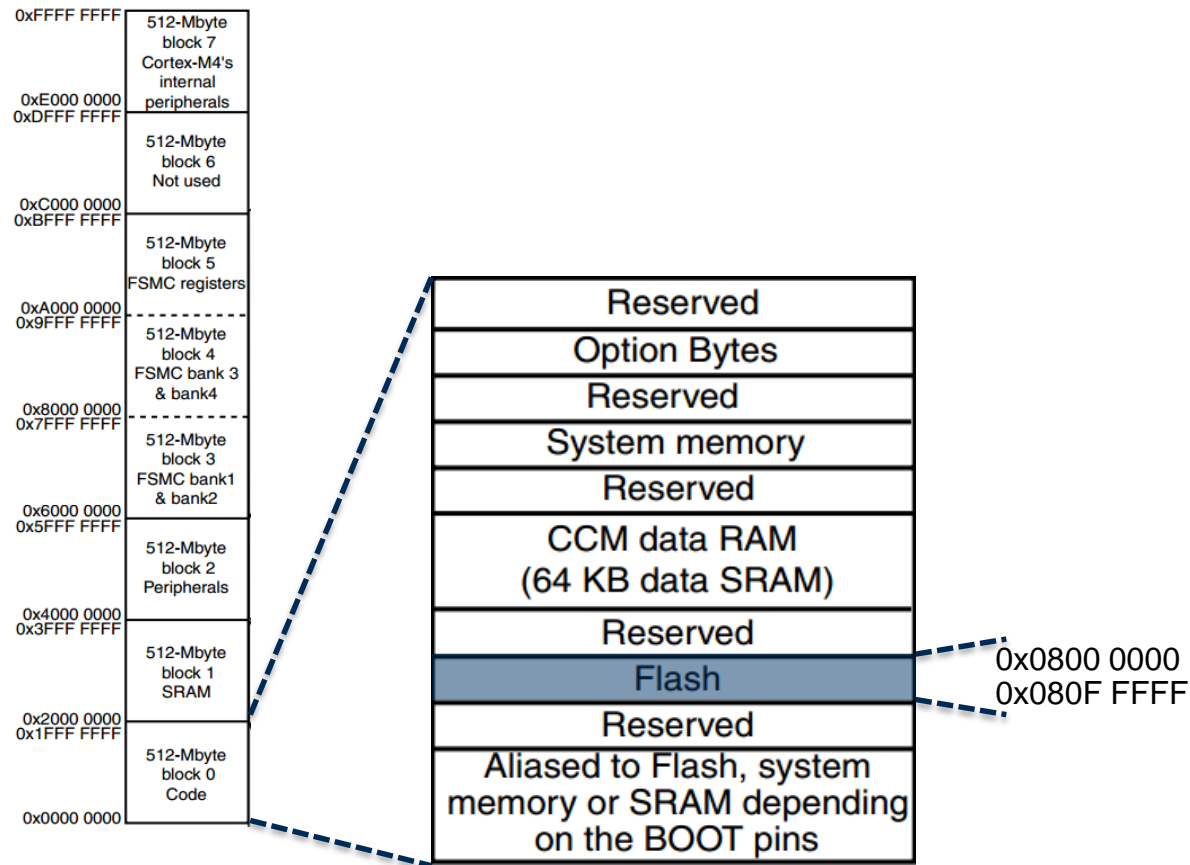
Keil MDK-ARM

- target is maximaal 32 kB (ivm Lite editie)
- Twee targets: RAM & FLASH



Development basics

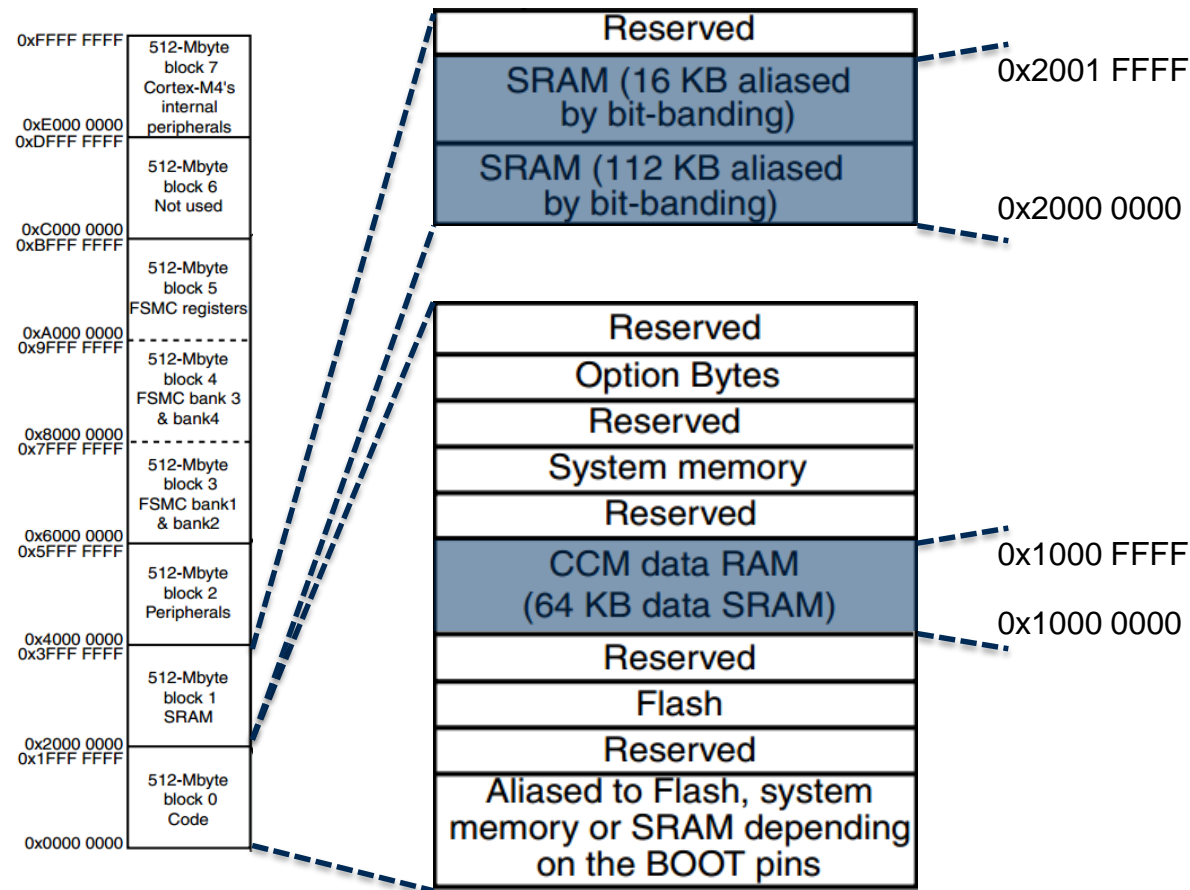
Target: FLASH (1 MB beschikbaar)





Development basics

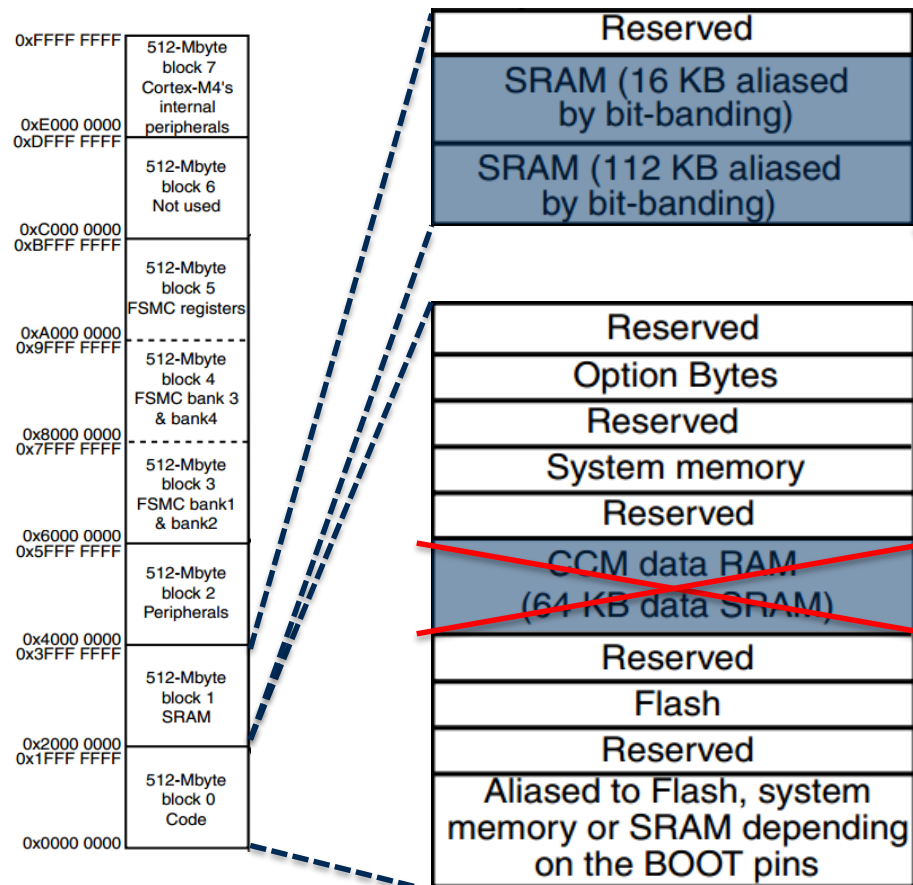
Target: RAM (192 kB beschikbaar)





Development basics

Target: RAM (192 kB)



100 kB reserveren voor vier images (ruim 177*144 bytes per image)

28 kB 'over' om te debuggen

Niet te gebruiken voor debugging



Development basics

Target in RAM

Voordelen

- sneller laden
- eenvoudig debuggen

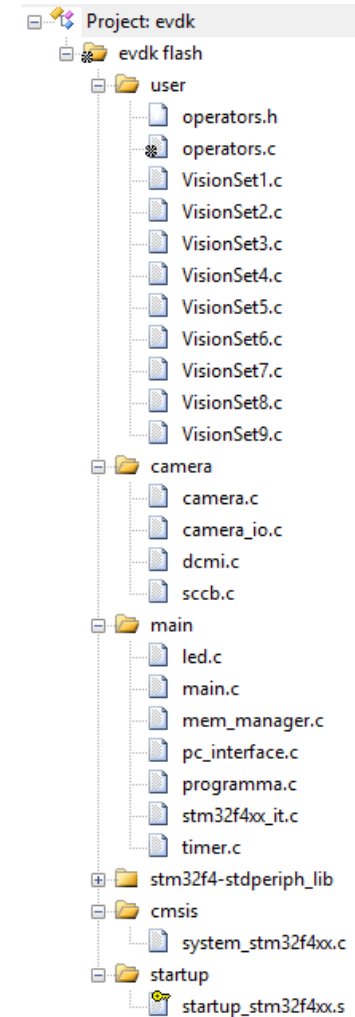
Nadelen

- 4 kB minder code space
- 28 kB minder RAM (van totaal 192 kB...)
- verloopt altijd via de debugger van MDK-ARM
- code executie is trager (door debugging)!



Development basics

Uitleg file structuur in Keil MDK-ARM





Development basics

Debuggen

```
#ifdef QDEBUG_ENABLE
```

```
#include <QDebug>
#define QDEBUG(x) qDebug()<<x

// Example usage:
// QDEBUG("Debug" << parameter << "value");
```

simulatie

```
#else
```

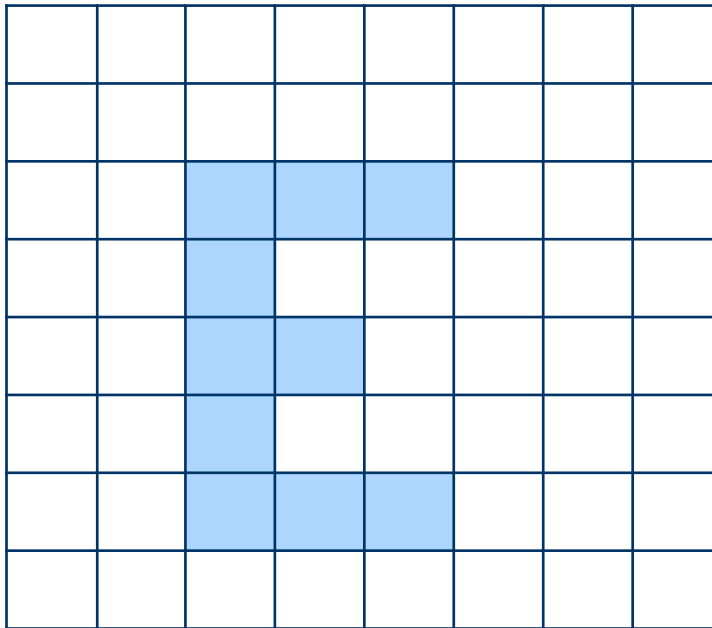
```
#define QDEBUG(x) //x
```

target

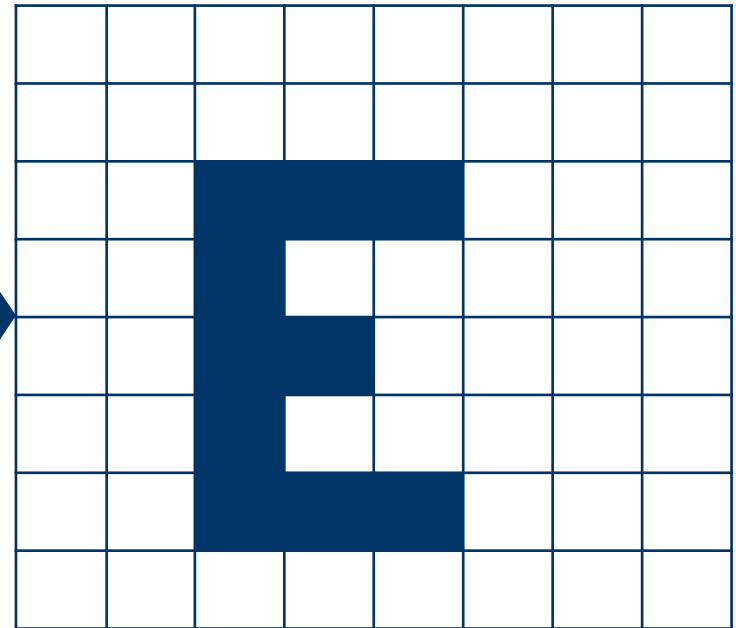
```
#endif
```



Contrast Stretch



Source



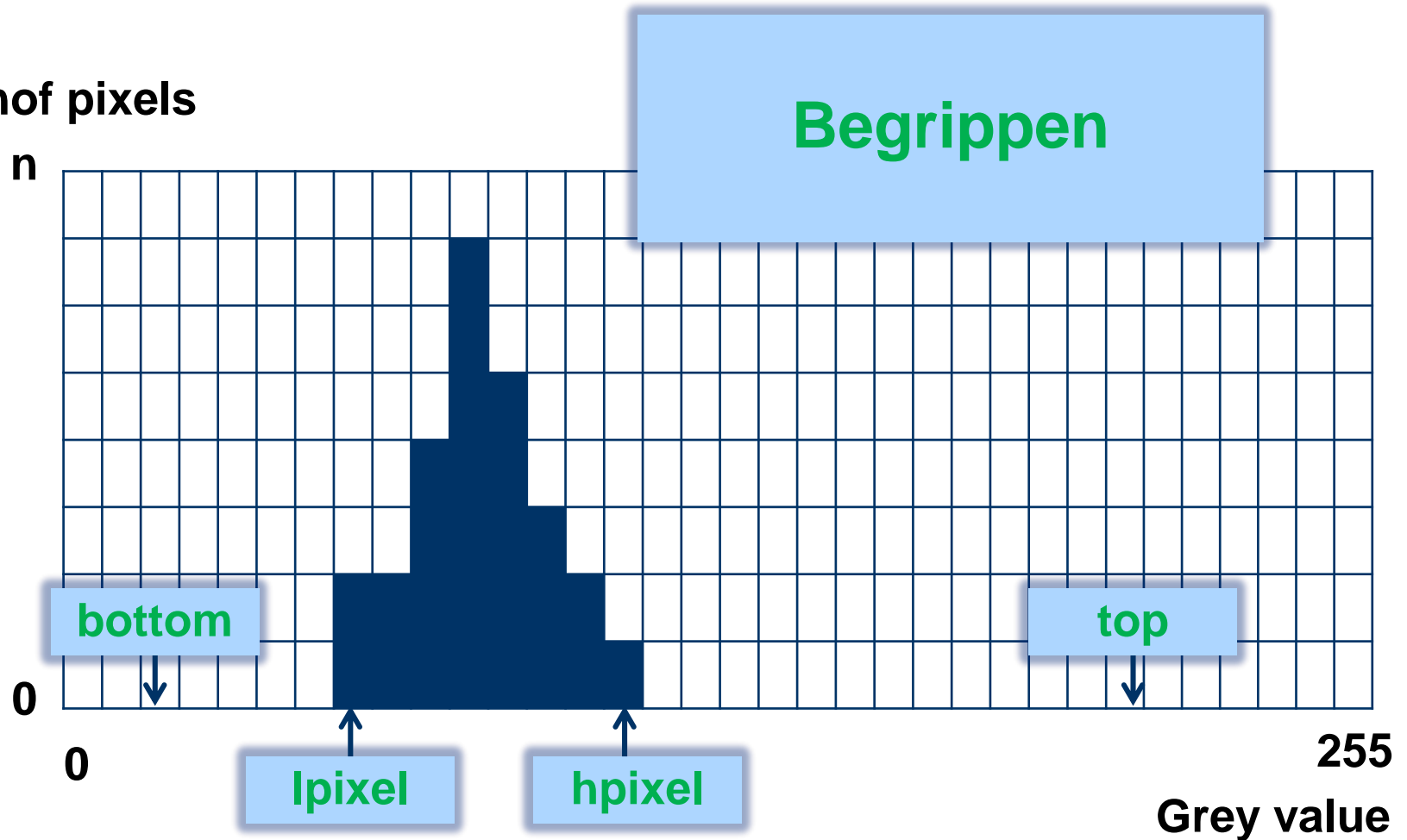
Destination



Contrast Stretch

nof pixels

n

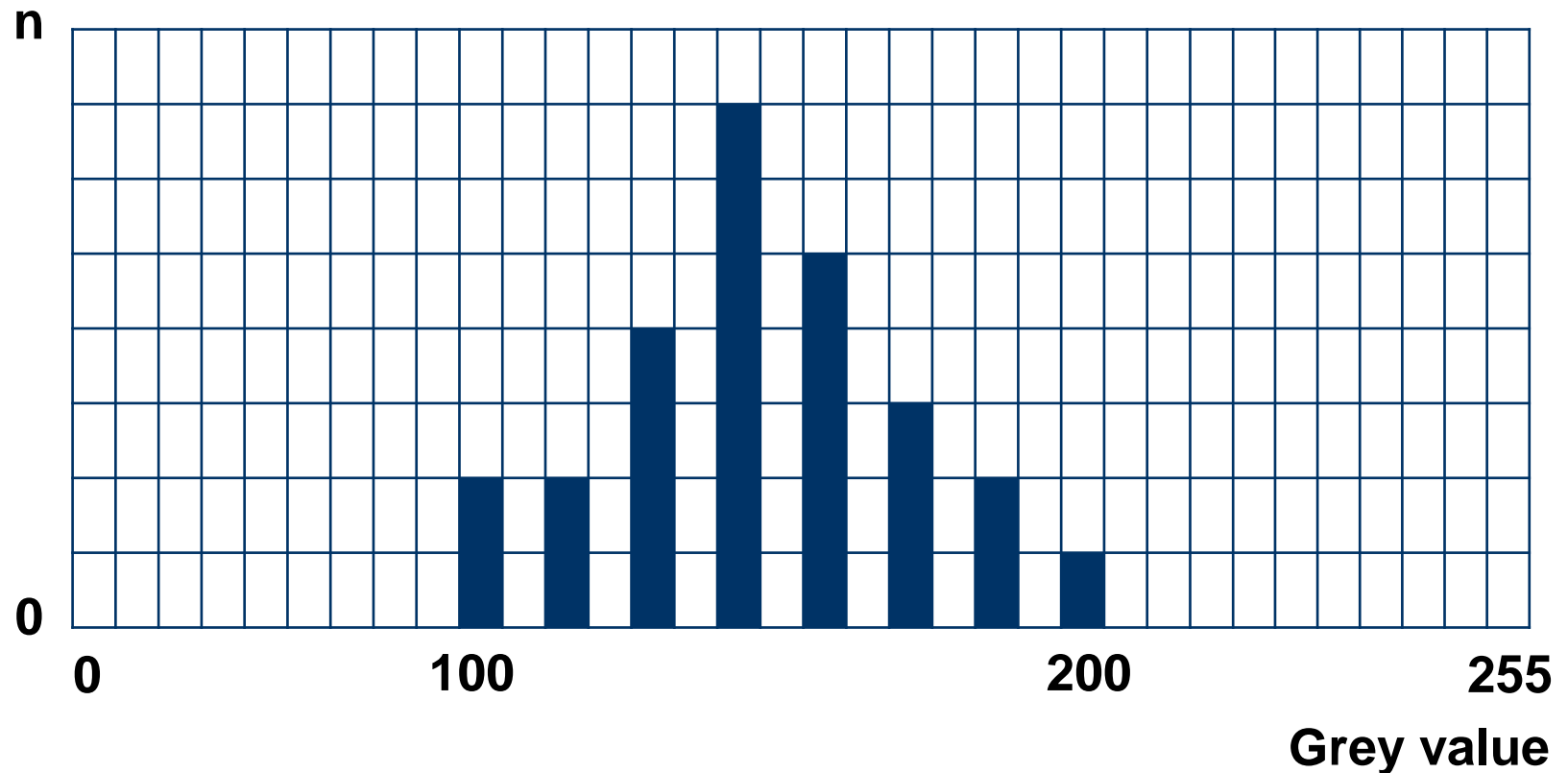




Contrast Stretch

Destination (i.e. bottom=100 & top=200)

Number of pixels remain the same

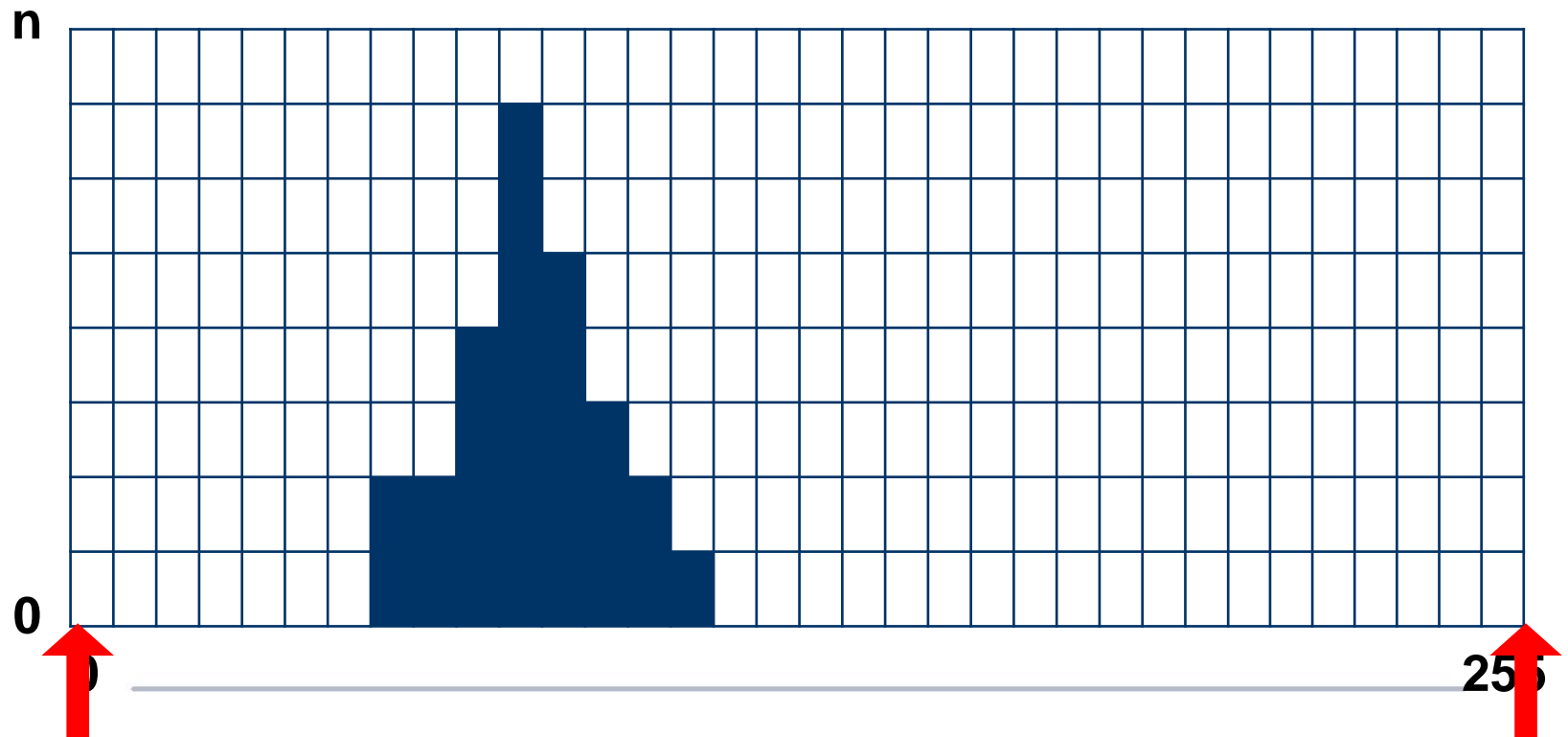




Contrast Stretch

Algorithm:

1. find l_{pixel} & h_{pixel}



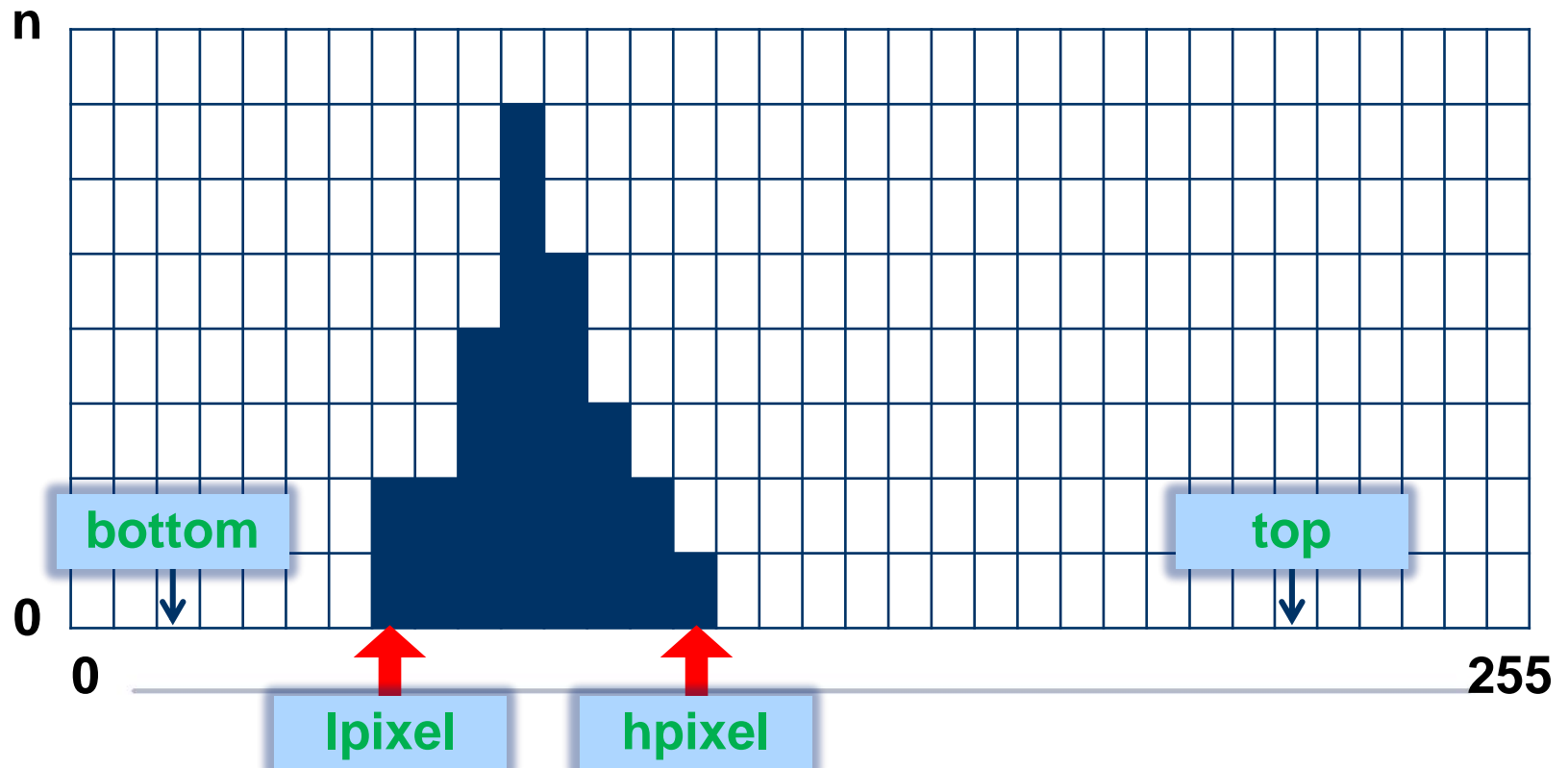


Contrast Stretch

Algorithm:

1. find lpixel & hpixel
2. calculate stretch factor

$$\frac{(\text{top} - \text{bottom})}{(\text{hpixel} - \text{lpixel})}$$

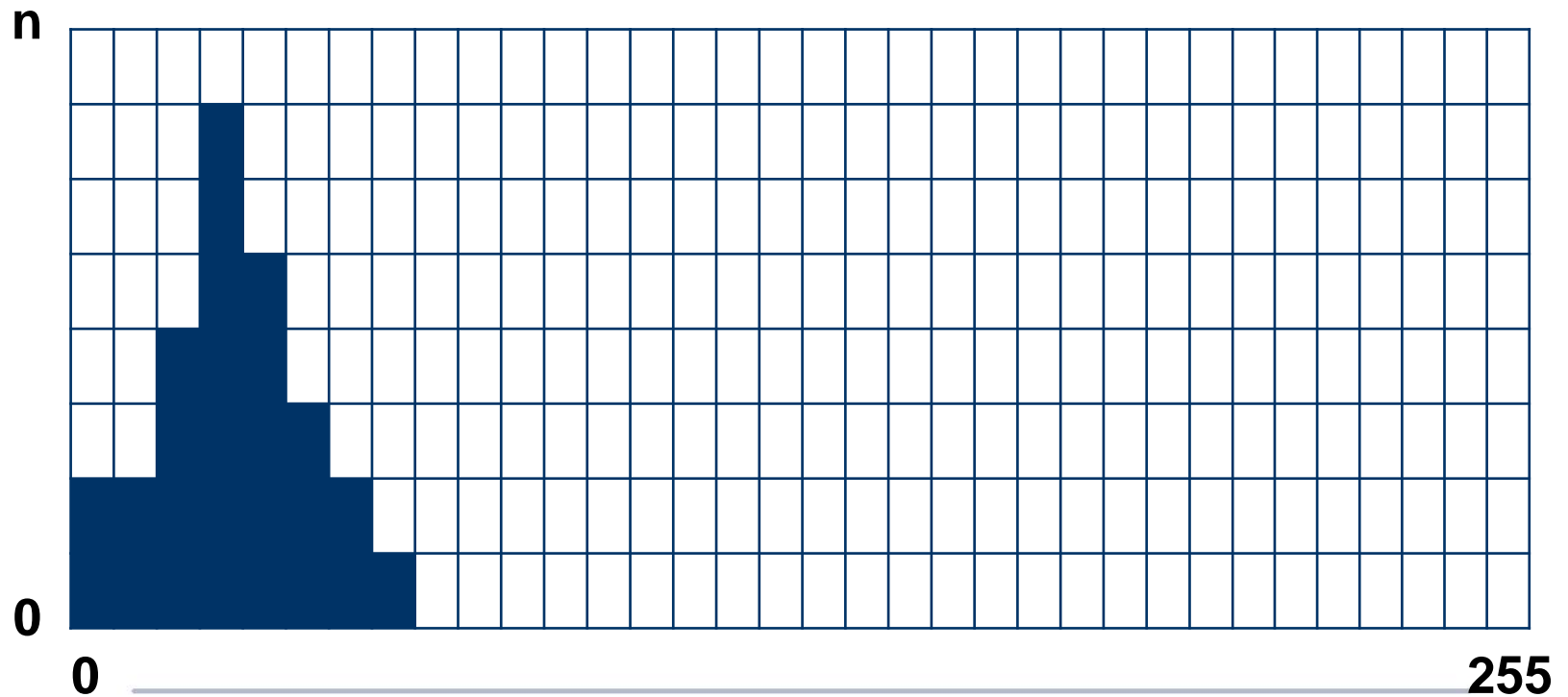




Contrast Stretch

Algorithm:

1. find l_{pixel} & h_{pixel}
2. calculate stretch factor
3. move all pixels to 0

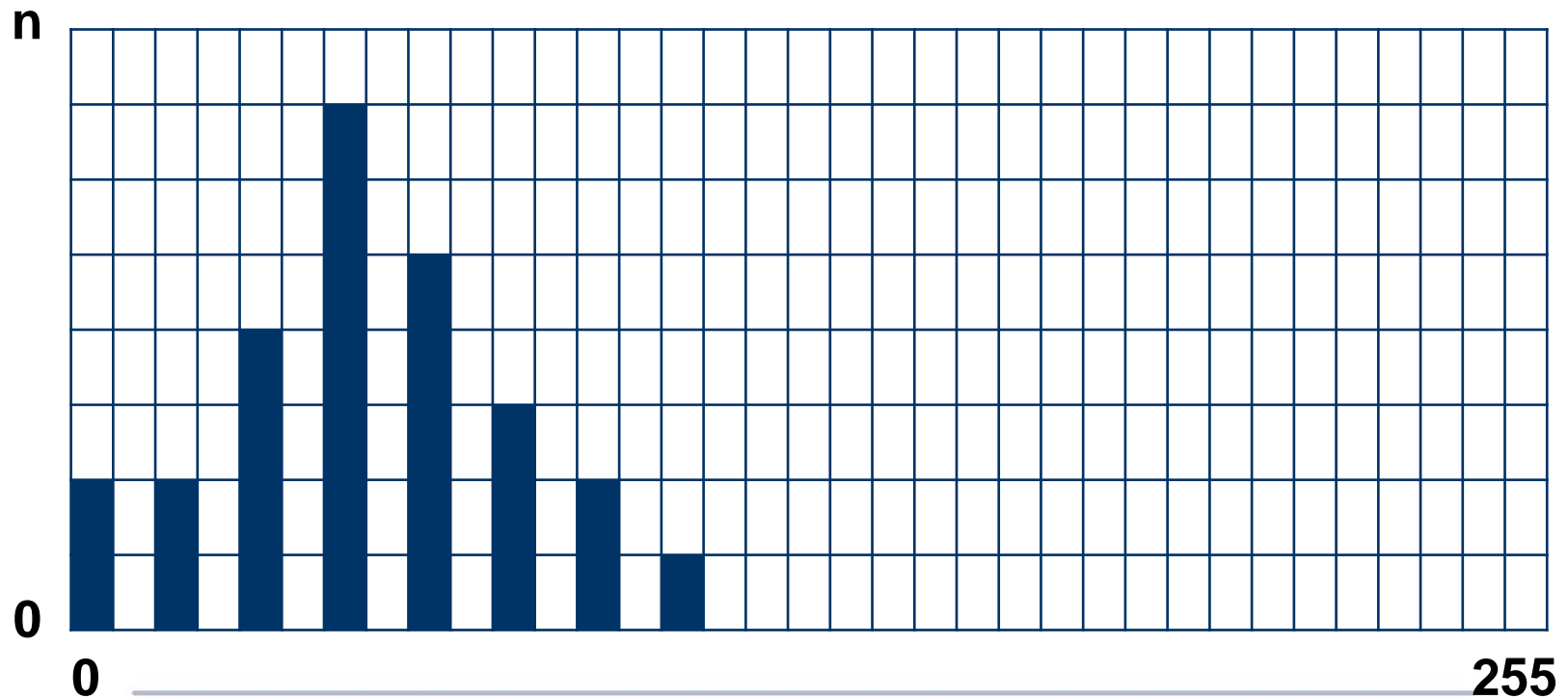




Contrast Stretch

Algorithm:

1. find lpixel & hpixel
2. calculate stretch factor
3. move all pixels to 0
4. apply stretch factor to each pixel





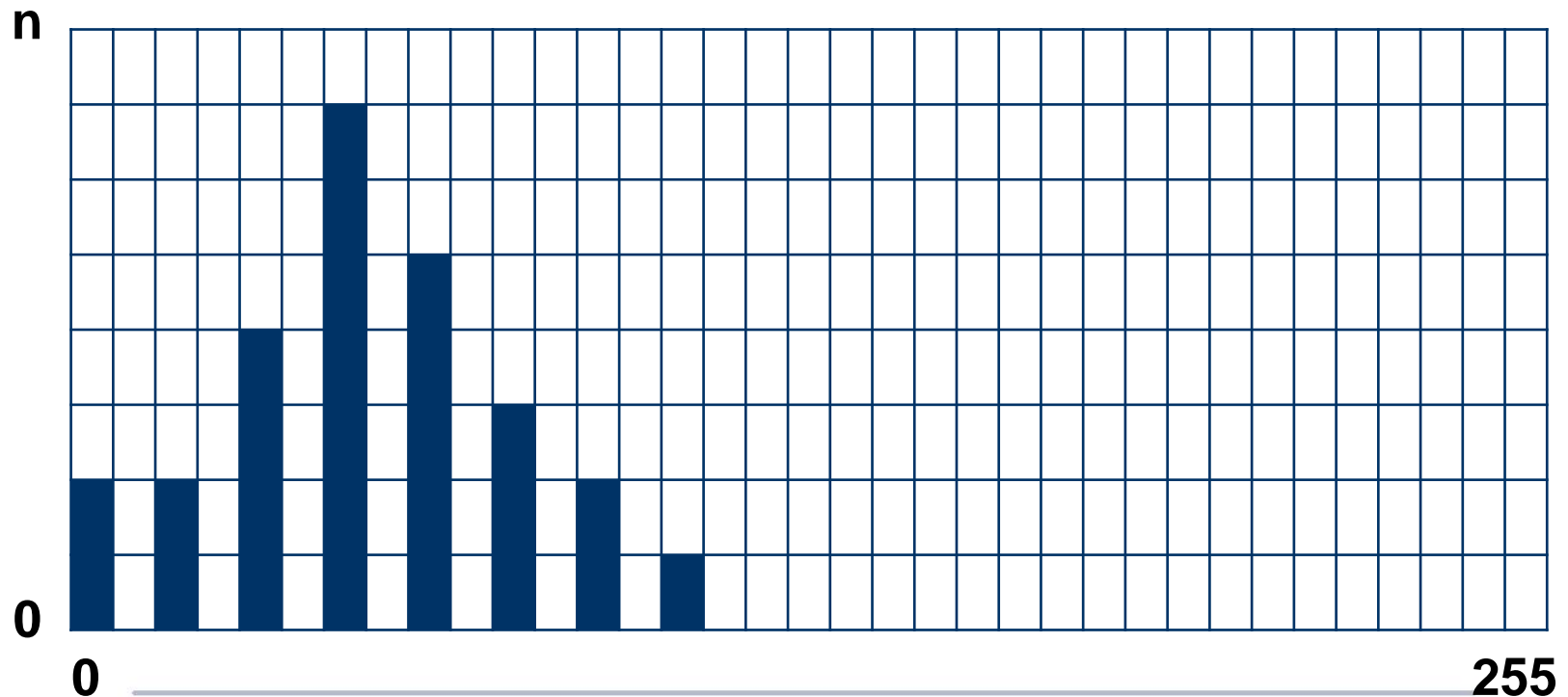
Contrast Stretch

Algorithm:

1. find lpixel & hpixel
2. calculate stretch factor
3. move all pixels to 0
4. apply stretch factor to each pixel
5. rond af naar boven (add 0,5 and typecast to uint8_t)

```
float f = 254,99995;
uint8_t i = (uint8_t)f;
```

i = 254!!!

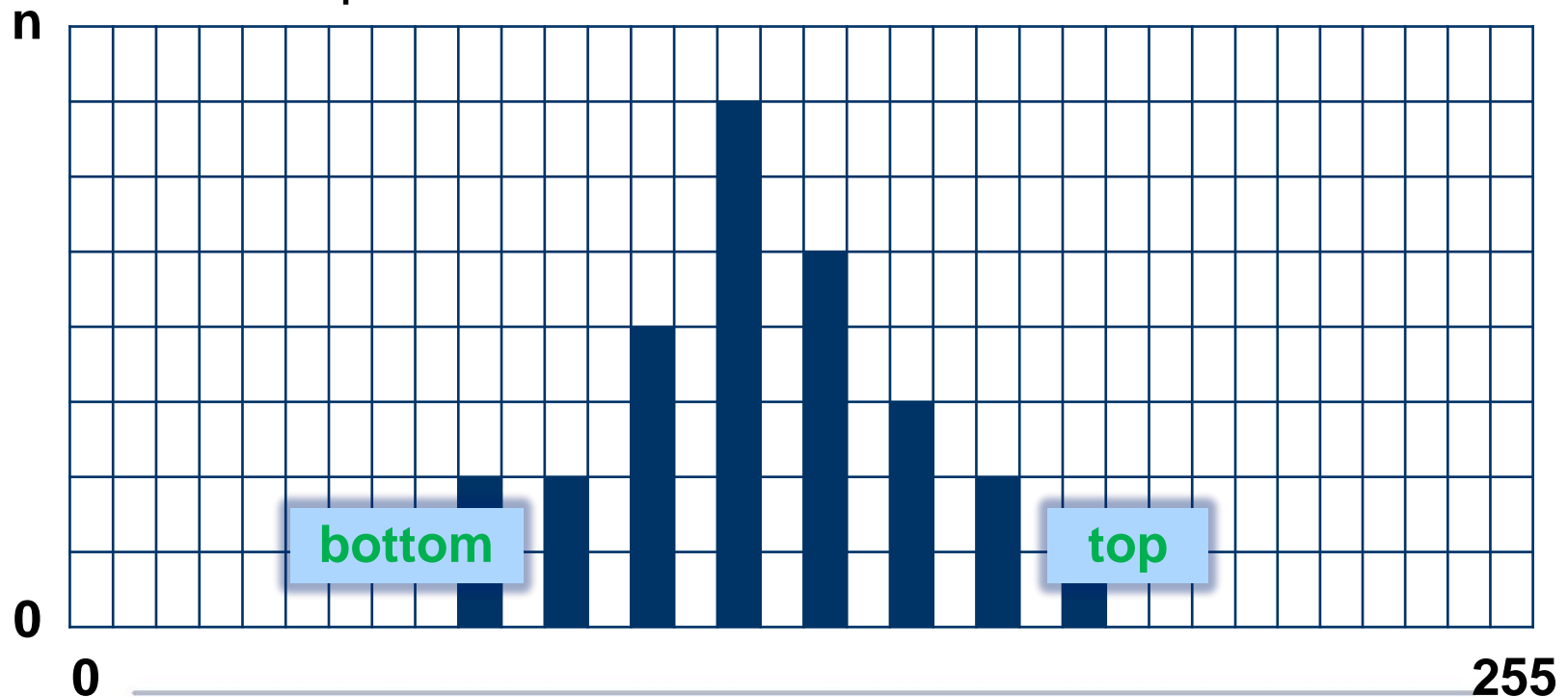




Contrast Stretch

Algorithm:

1. find lpixel & hpixel
2. calculate stretch factor
3. move all pixels to 0
4. apply stretch factor to each pixel
5. rond af naar boven (add 0,5 and typecast to uint8_t)
6. move all pixels to bottom

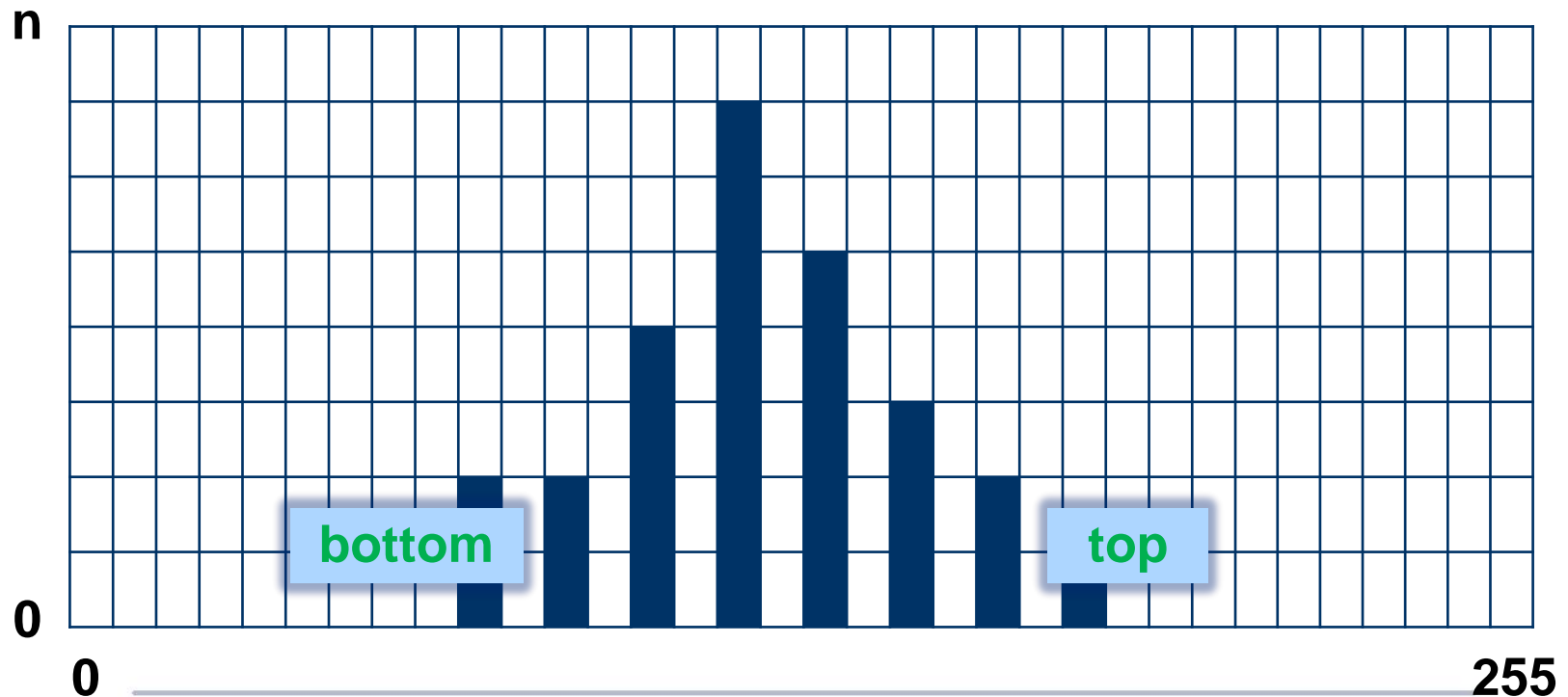




Contrast Stretch

Formula:

$$p_{n+1} = \left((p_n - lpixel) \cdot \frac{top - bottom}{hpixel - lpixel} \right) + 0,5 + bottom$$





Contrast Stretch

Homogene plaatjes

- Wat is het probleem?
- Wat is het gewenste resultaat?

