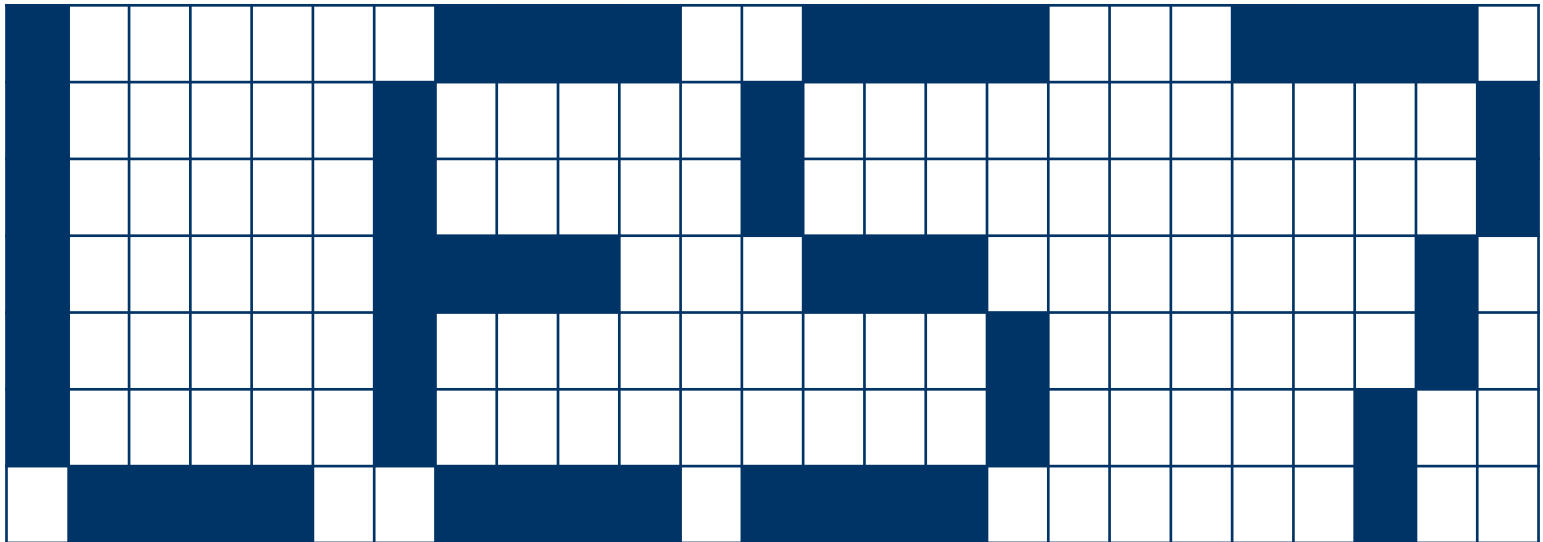


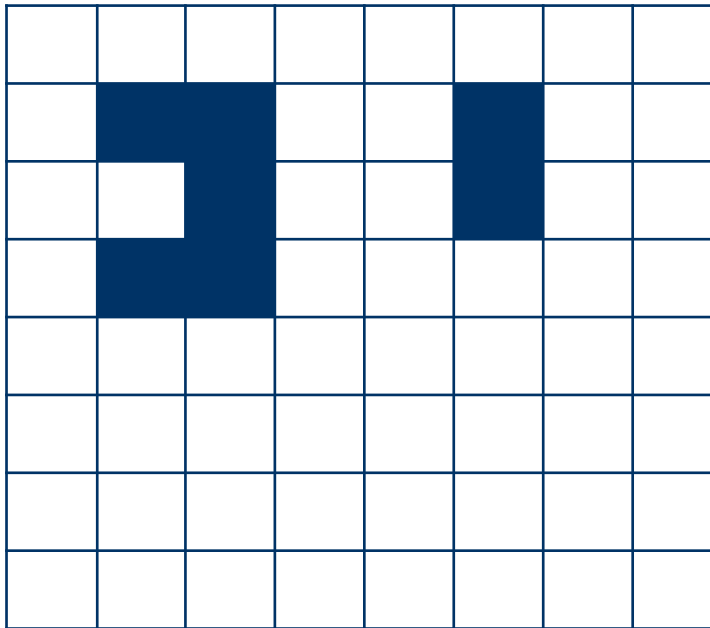


EVD1 – Vision operators

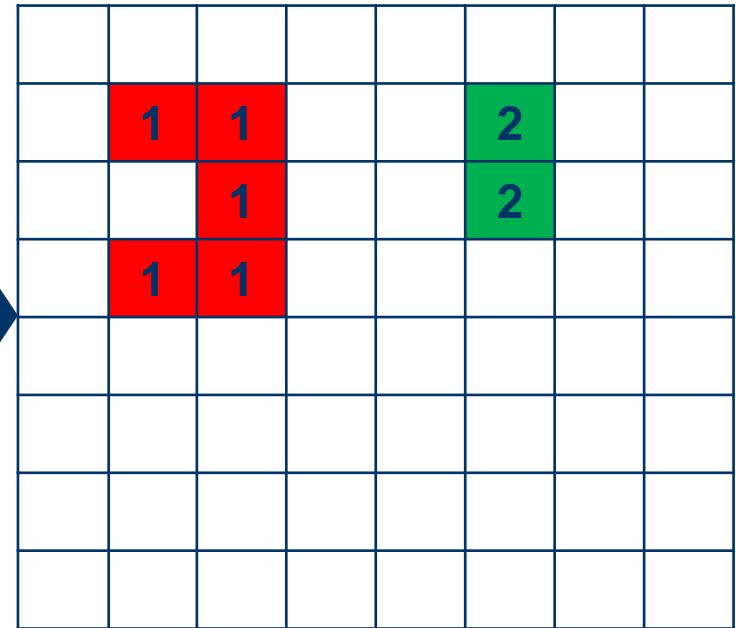




Label blob's



Source



Destination



Label blob's

	1	1			1		
		1			1		
	1	1					

Source



	1	2			3		
		4			5		
	6	7					

Destination



Label blob's

	1	2			3		
		4			5		
	6	7					

Source



	1	1			3		
		1			3		
	1	1					

Destination



Label blob's

	1	1			3		
		1			3		
	1	1					



	1	1			2		
		1			2		
	1	1					

Source

Destination

**blobs toekennen in juiste volgorde
(blob_cnt)**



Label blob's

	1	2			3		
		4			5		
	6	7					
				
				...	253	254	
					255	???	

Probleem:

Wat doen we met plaatjes die BLOB's hebben van meer dan 255 pixels?

Source



Label blob's

Afbakening alternatieve implementatie:

- beperkt aantal blobs detecteerbaar: maximaal 254
- return values:
 - 0: nul of meer dan 254 blob's
 - 1 t/m 254: aantal gevonden blobs
- marker: 255



Label blob's

	1	1			1		
		1			1		
	1	1					

Source



	255	255			255		
		255			255		
	255	255					

Destination



Label blob's

	255	255			255		
		255			255		
	255	255					

Source



	1	255			255		
		255			255		
	255	255					

Destination

blob_cnt = 1



Label blob's

	255	255			255		
		255			255		
	255	255					

Source



	1	1			255		
		255			255		
	255	255					

Destination

blob_cnt = 1



Label blob's

	255	255			255		
		255			255		
	255	255					

Source



	1	1			2		
		255			255		
	255	255					

Destination

blob_cnt = 2



Label blob's

	255	255			255		
		255			255		
	255	255					



	1	1			2		
		1			255		
	255	255					

Source

Destination

blob_cnt = 2



Label blob's

	255	255			255		
		255			255		
	255	255					



	1	1			2		
		1			2		
	1	1					

Source

Destination

blob_cnt = 2



Label blob's

		1			1		
		1			1		
		1			1		
		1			1		
		1			1		
		1	1	1	1		

Probleem:

**Maar wat doen we met dit
soort plaatjes?**

Source



Label blob's

		1			1		
		1			1		
		1			1		
		1			1		
		1			1		
		1	1	1	1		

Source



		255			255		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		

Destination



Label blob's

		255			255		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		

Source



		1			255		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		

Destination

blob_cnt = 1



Label blob's

		255			255		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		



		1			2		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		

Source

Destination

blob_cnt = 2



Label blob's

		255			255		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		



		1			2		
		1			2		
		1			2		
		1			2		
		1			2		
		1	1	1	1		

Source

Destination

blob_cnt = 2



Label blob's

		255			255		
		255			255		
		255			255		
		255			255		
		255			255		
		255	255	255	255		



		1			1		
		1			1		
		1			1		
		1			1		
		1			1		
		1	1	1	1		

Source

Destination

oplossing: herhalen van RO->LB en LB->RO totdat er geen veranderingen meer zijn!



Label blob's

Maarrrr...

	1		1				
	1		1				
	1		1		1		
	1	1	1		1		
					1		
					1		

Source



	1		2				
	1		2				
	1		2		3		
	1	1	1		3		
					3		
					3		

Destination (1^e iteratie)



Label blob's

Maarrrr...

	1		2				
	1		2				
	1		2		3		
	1	1	1		3		
					3		
					3		



	1		1				
	1		1				
	1		1		3		
	1	1	1		3		
					3		
					3		

Source

Destination (2^e iteratie)

Dus: blobs toekennen in juiste volgorde (blob_cnt)



Label blob's

	1		2				
	1		2				
	1		2		3		
	1	1	1		3		
					3		
					3		

Source



	1		1				
	1		1				
	1		1		2		
	1	1	1		2		
					2		
					2		

Destination



Label blob's

blob_cnt=1

Markeer alle objecten met 255

// Verbind alle pixels met dezelfde (laagste) waarde

ZOLANG er veranderingen zijn

LB -> RO

ALS huidige pixel geen achtergrond is **EN** een buurpixel heeft die kleiner is

DAN maak huidige pixel gelijk aan kleinste buurpixel

ANDERS

ALS huidige pixel nog gemarkeerd is met 255

DAN maak huidige pixel gelijk aan blob_cnt

blob_cnt++

RO -> LB

IDEM

// Zet de labels in de juiste volgorde

blob_cnt=1

LB -> RO

ALS huidige pixel is gelabeld **EN** groter of gelijk aan blob_cnt

DAN **vSetSelectedToValue(dst, dst, huidige pixel value, blob_cnt)**

blob_cnt++



Blob analysis

	1		1				
	1	1	1				
	1	1	1				
					2	2	
					2	2	

Source

Blob info

Height

Width

Nof_pixels

Perimeter

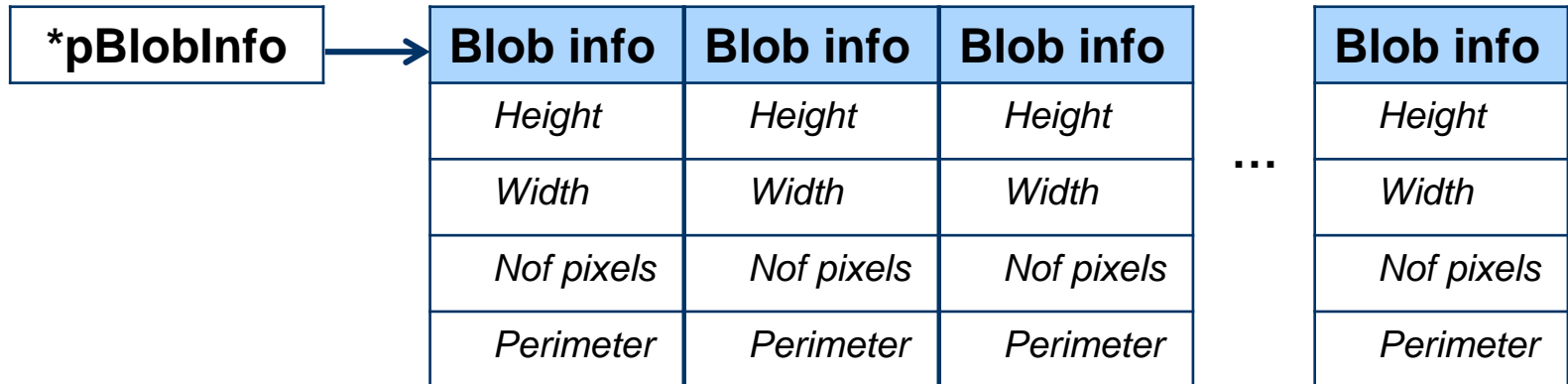
```
typedef struct blobinfo_t
{
    uint16_t height;
    uint16_t width;
    uint16_t nof_pixels;
    float    perimeter;
}blobinfo_t;
```




Blob analysis

Gebruik van de functie:

```
void vBlobAnalyse(image_t *img,
                  const uint8_t blobcount,
                  blobinfo_t *pBlobInfo);
```



Geheugen alloceren:

- malloc() + free() (runtime)
- fixed length (compile time)