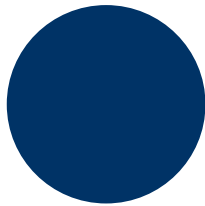


Blue	White	Blue	White	Blue	White	Blue	White	Blue	White
White	Blue	White	Blue	White	Blue	White	Blue	White	Blue
Blue	White	Blue	White	Blue	White	Blue	White	Blue	White
White	Blue	White	Blue	White	Blue	White	Blue	White	Blue
Blue	White	Blue	White	Blue	White	Blue	White	Blue	White
White	Blue	White	Blue	White	Blue	White	Blue	White	Blue
Blue	White	Blue	White	Blue	White	Blue	White	Blue	White
White	Blue	White	Blue	White	Blue	White	Blue	White	Blue
Blue	White	Blue	White	Blue	White	Blue	White	Blue	White
White	Blue	White	Blue	White	Blue	White	Blue	White	Blue



# Vormen classificeren

---





# Vormen classificeren – eenvoudig

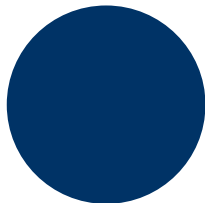
Oppervlak, hoogte en breedte is bekend uit blob analyse.  
Dus...



oppervlak = hoogte x breedte



oppervlak = (hoogte x breedte) / 2



oppervlak =  $\pi r^2 = \pi(\text{breedte} / 2)^2$

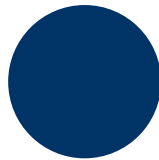


# Vormen classificeren - Form factor

Om onafhankelijk van rotatie te kunnen werken, rekenen we met de zgn. form factor.

De form factor is een getal dat aangeeft welke vorm een object heeft.

Het wordt berekend aan de hand van het oppervlak en de omtrek.

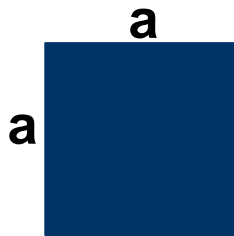


$$V = \text{omtrek}^2 / \text{opp}$$



# Form factor - square

---



$$opp = a^2$$

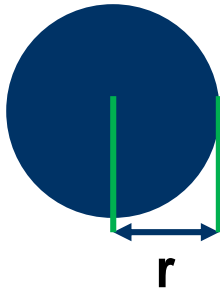
$$omtrek = 4a$$

$$V = \frac{(4a)^2}{a^2}$$

$$V = ?$$



# Form factor - circle



$$opp = \pi r^2$$

$$omtrek = 2\pi r$$

$$V = \frac{(2\pi r)^2}{\pi r^2}$$

$$V = ?$$

VisionLab draait het om:

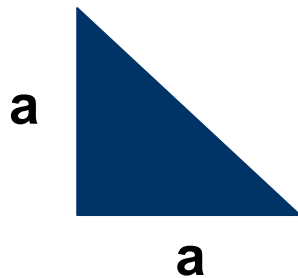
$$\text{FormFactor} = 4\pi * ((opp) / (omtrek^2))$$

1 = circle

0 = line



# Form factor - triangle



$$opp = \frac{1}{2} a^2$$

$$omtrek = 2a + \sqrt{a^2 + a^2}$$

$$omtrek = 2a + a\sqrt{2}$$

$$V = \frac{(2a + a\sqrt{2})^2}{\frac{1}{2} a^2}$$

$$V = ?$$



# Form factor – opp bepalen

vBlobAnalyse resultaat gebruiken om Form Factor te berekenen

Immers, is de Form Factor bekend, dan kunnen we de vorm classificeren: cirkel / driehoek / vierkant / onbekend

Oppervlak is eenvoudig: size van de blob

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

size: 25





# Form factor – omtrek bepalen

Omtrek bepalen is lastiger, want we kunnen niet gewoon het aantal pixels in de rand tellen:

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

omtrek berekenen:  $4a = 20$

pixels tellen: 16



# Form factor – omtrek bepalen

Oplossing: tel het aantal randen van de rand pixels

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

omtrek berekenen:  $4a = 20$

randen tellen: 20



# Form factor – omtrek bepalen

Maar wat gebeurt er bij andere vormen?

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

omtrek berekenen:  $2a + a \cdot \sqrt{2} = 17,07$

randen tellen: 20



# Form factor – omtrek bepalen

Oplossing: reken met schuine randen!

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

omtrek berekenen:  $2a + a \cdot \sqrt{2} = 17,07$

randen berekenen  $2a + a \cdot \sqrt{2} = 17,07$

*Maar wanneer doen we dat dan,  
rekenen met schuine randen??*



# Form factor – omtrek bepalen

Er zijn drie opties:

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

pixels met één rand

$$\text{omtrek} = \text{omtrek} + 1$$



# Form factor – omtrek bepalen

Er zijn drie opties:

1				
1	1	—		
1	1	1		
1	1	1	1	
1	1	1	1	1

pixels met twee randen

$$\text{omtrek} = \text{omtrek} + \text{sqrt}(2)$$



# Form factor – omtrek bepalen

Er zijn drie opties:

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

pixels met drie randen

$$\text{omtrek} = \text{omtrek} + \text{sqrt}(5)$$



# Form factor – omtrek bepalen

Dat betekent dat we fouten introduceren:

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

omtrek berekenen:  $2a + a \cdot \sqrt{2} = 17,07$

omtrek door randen tellen:

$6 + 4 \cdot \sqrt{2} + 2 \cdot \sqrt{5} = 16,31$

$$V = 17,73 \quad (23,31)$$





# Form factor – omtrek bepalen

En:

<del>1</del>	1	1	1	<del>1</del>
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
<del>1</del>	1	1	1	<del>1</del>

omtrek berekenen:  $4a = 20$

omtrek door randen tellen:

$12 + 4 * \text{sqrt}(2) = 17,66$

$$V = 12,48 \quad (16)$$

Deze fouten lijken behoorlijk veel impact te hebben,  
maar dit effect wordt snel minder naarmate de  
size van de blob toeneemt!



# Form factor – omtrek bepalen

Fouten die we kunnen verwachten:

16

Door het 'afronden' van de hoeken wordt de omtrek korter

Kortere omtrek = lagere form factor

Let op met overlap!!

23,31

Door het 'afronden' van de hoeken wordt de omtrek korter

Kortere omtrek = lagere form factor

12,57

Door 'grilligheid' van de pixels wordt de omtrek langer dan  $2\pi r$

Langere omtrek = hogere form factor



# Form factor

---

## Opdracht

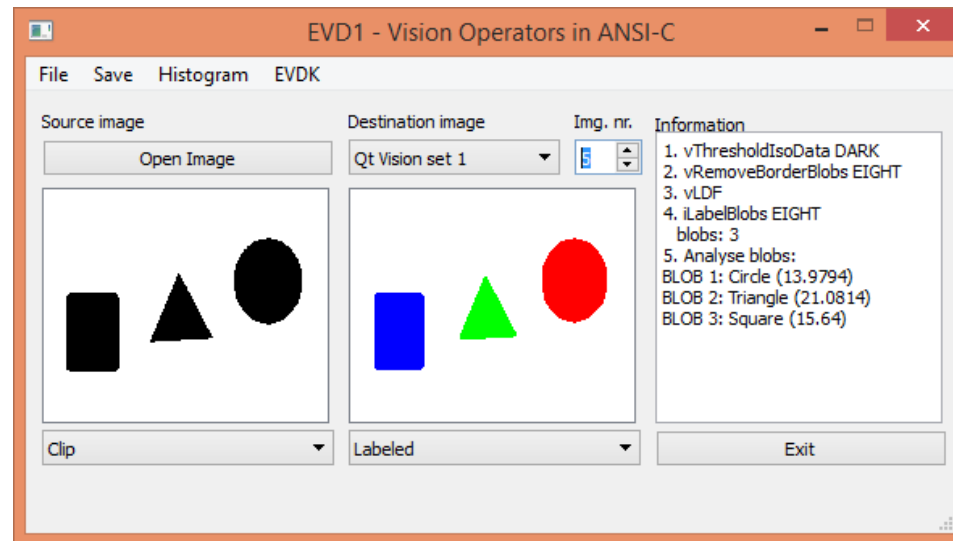
Pas de functie `vBlobAnalyse` zo aan dat het veld 'perimeter' in `blobinfo_t` struct volgens dit algoritme de omtrek berekent!



# Form factor

## Opdracht

Gebruik de uitkomst van de functie `vBlobAnalyse` om de form factor te berekenen in de simulatie...

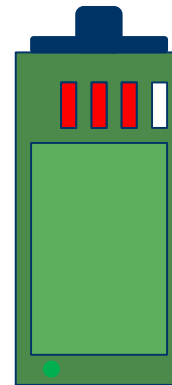
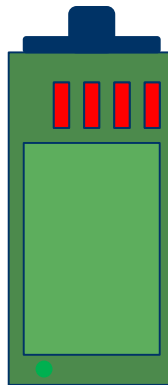
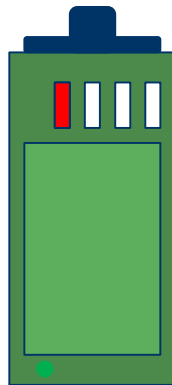
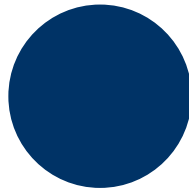




# Form factor

---

... en op het target!



→ User LED's