

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom-left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

3/20/2020

Meetrapport

Kernel en thresholding optimalisatie

Kevin Patist en Wietse ten Dam

Table of Contents

Doel..... 2

Hypothese 2

Werkwijze..... 2

Resultaten 3

 Test 1:..... 3

 Resultaten..... 3

 Test 2 4

 Resultaten..... 5

Verwerking 5

Conclusie 6

Evaluatie..... 6

Doel

Om te kunnen bepalen of onze nieuwe edge detection beter is dan de oude moeten we zeker weten dat wij deze optimaal hebben geïmplementeerd. Hiervoor moeten wij weten wat de beste combinatie van kernel en thresholding waarde is. Zodra deze is gevonden kunnen de edge detection implementaties vergeleken worden.

Hypothese

Wij denken dat de default Sobel kernel (staat hiernaast) het waarschijnlijk het best gaat doen. Dit omdat wij bij elke bron die wij voor ons onderzoek hebben gebruikt deze kernel terug zien komen. Omdat wij beide nog nooit op deze manier met thresholding hebben gewerkt kunnen wij hier geen verwachting voor bedenken.

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

Werkwijze

Om de beste combinatie van kernel en thresholding waarde te bepalen gaan wij 25 tests doen. Dit met elke mogelijke combinatie van vijf kernels en vijf thresholding waarden. Deze waarden hebben een gelijke afstand tot elkaar. De waarden zijn als volgt:

Naam	Kernel	Naam	Thresholding waarde
K1	1 0 -1 2 0 -2 1 0 -1	T1	100
K2	2 0 -2 3 0 -3 2 0 -2	T2	120
K3	3 0 -3 4 0 -4 3 0 -3	T3	140
K4	4 0 -4 5 0 -5 4 0 -4	T4	160
K5	5 0 -5 6 0 -6 5 0 -5	T5	180

Elke combinatie wordt getest met set van tien foto's die wij zelf hebben verzameld op het internet. Deze tien foto's zijn te vinden in de map testsets/found images/K&T test1

Per combinatie voeren we handmatig de 10 foto's in waaruit een gezicht moet worden herkend.

Na het testen met de set worden de resultaten genoteerd.

Deze stappen worden voor elke combinatie herhaald. De combinatie met de meeste goed herkende gezichten is in onze ogen de beste. Als wij deze tests hebben gedraaid en er is geen duidelijke winnaar uit gekomen dan gaan wij verder testen. Voor deze tests nemen wij de kernel waar de beste resultaten uit komen. De thresholding waarden waar we mee gaan testen liggen dan tussen te waarden die in de eerste test de beste resultaten gaven. De waarden zullen steeds twee van elkaar af liggen. De tweede testronde wordt ook uitgevoerd met twee test sets, het gaat hier om K&T test 1 en K&T test 2.

Resultaten

Test 1:

Resultaten

Kernel	Thresholding waarde	Aantal goed herkende gezichten	Gevonden gezichten* (met error in programma)	Herkende foto's (nummer) van set 1
K1	T1	0	2	2, 6
	T2	0	1	2
	T3	0	2	1, 2
	T4	0	0	
	T5	0	0	
K2	T1	0	0	
	T2	0	1	2
	T3	0	0	
	T4	0	2	2, 7
	T5	0	0	
K3	T1	0	0	
	T2	0	0	
	T3	0	0	
	T4	0	0	
	T5	0	1	2
K4	T1	0	0	
	T2	0	0	
	T3	0	0	
	T4	0	0	
	T5	0	1	1
K5	T1	0	0	
	T2	0	0	
	T3	0	0	
	T4	0	0	
	T5	0	0	

* = Deze kolom bevat het aantal gezichten die wel goed herkend zijn (dit wil zeggen dat alle features die worden gezocht goed gevonden zijn) maar waar het programma een exception of error geeft bij het verder verwerken. Dit laatste controleren wij door te kijken of wel elke feature is gevonden voor de exception of error (dit kan door in de gui te kijken).

Test 2

Vanwege het grote succes met K1 zullen we met deze kernel verder werken. De nieuwe thresholding waarden waarmee we gaan testen zullen tussen T1 en T2 liggen. Dit omdat hier het meeste gezichten wordt herkend. Een groot deel van de waarden zal ook onder T1 liggen omdat wij er tijdens test 1 achter zijn gekomen dat lagere waarden het bij deze kernel goed doen.

De thresholding waarden verfijnen we als volgt:

Naam	Thresholding waarde
T10	80
T11	82
T12	84
T13	86
T14	88
T15	90
T16	92
T17	94
T18	96
T19	98
T20	100
T21	102
T22	104
T23	106
T24	108
T25	110
T26	112
T27	114
T28	116
T29	118
T30	120

Resultaten

Naam	Aantal goed herkende gezichten	Gevonden gezichten* (met error in programma)	Herkende foto's (nummer)
T10	1	0	2.10
T11	1	1	1.7 2.10
T12	1	0	2.10
T13	1	1	1.2 2.10
T14	1	1	1.2 2.10
T15	1	3	1.2 1.7 2.3 2.10
T16	1	2	1.2 2.3 2.10
T17	1	3	1.2 1.6 2.3 2.10
T18	1	3	1.2 1.6 2.3 2.10
T19	1	3	1.2 1.6 1.7 2.10
T20	1	1	2.10, 1.2, 1.6
T21	1	1	2.10, 1.2
T22	1	1	2.10, 1.2
T23	1	1	2.10, 1.2
T24	1	1	2.10, 1.2
T25	0	1	1.2
T26	0	1	1.2
T27	0	1	1.2
T28	1	1	2.10, 1.2
T29	1	1	2.10, 1.2
T30	1	1	2.10, 1.2

Bij de herkende foto's zetten we een 1 of 2 voor het getal. Hiermee bedoelen we de foto set waar de foto's in staan. 1.2 is dan dus de tweede foto van set 1.

Uit deze tests is gebleken dat de combinaties K1 met T15, T17, T18 en T19 de beste resultaten opleveren. Van deze thresholding waarden kiezen wij T19 als beste om de volgende reden. Omdat T19 1.6 én 1.7 heeft herkend naast een paar andere foto's denken wij dat deze het beter zal doen met andere foto's. Dit omdat 1.6 en 1.7 tot de moeilijkere foto's vallen om te herkennen binnen onze testsets.

Verwerking

De uitkomst van onze resultaten laat zien dat de combinatie van K1 en T19 de beste resultaten levert. Dit komt ook goed overeen met onze hypothese. Hierin hebben wij aangegeven dat wij de kernel K1 als uitkomst verwachtte maar verder geen idee hadden over welke thresholding waarde hierbij zou horen. Er is een mogelijkheid dat een andere kernel het bij andere foto's beter had kunnen doen omdat de performance naar ons idee ook heel afhankelijk is van de verlichting in de foto.

Conclusie

Uit de resultaten en conclusie halen wij dat de beste resultaten behaald worden met de combinatie K1-T19. Waarom we voor deze resultaten hebben gekozen is uitgelegd onder het kopje resultaten. Deze combinatie heeft als thresholding waarde 98 en de kernel van x en y is als volgt:

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

Evaluatie

Over het algemeen is het hele proces goed verlopen. De samenwerking, het maken van het meetrappport en planning zijn allemaal super goed gegaan naar onze mening. Wat iets beter had gekund is het vooruit denken voor we aan de slag gingen. We zijn er namelijk halverwege het schrijven van dit testrapport achter gekomen dat we eigenlijk nog een test wilde doen voor betere en specifiekere resultaten. Ons meetrappport hier op aanpassen heeft dus wat tijd gekost die we anders in het andere meetrappport hadden kunnen steken. Verder zijn we veel te snel bezig gegaan met het opzetten van een betere edge detection waardoor we eigenlijk veel te lang hebben besteed aan niet systematische tests met willekeurige waarden. Wat we dus een volgende keer beter kunnen doen is het beter vooruit denken over eventuele tests die gedaan moeten worden voordat we beginnen met het testen. Ook het overmatig los testen van onze methoden kan minder wanneer we weer in een dergelijke situatie terecht komen.