

DIE HERKENNING EN VERWERKING VAN TWEEDIMENSIONELE BEELDE

G.H. LAMPRECHT

DIE HERKENNING EN VERWERKING VAN TWEEDIMENSIONELE BEELDE

deur

GERT HENDRIK LAMPRECHT

Verhandeling voorgelê ter voldoening aan die vereistes vir die graad

**MAGISTER TECHNOLOGIAE:
INGENIEURSWESE: ELEKTRIES**

in die

Skool vir Elektriese en Rekenaarstelsel-Ingenieurswese

aan die

Technikon Vrystaat

Studieleier: Mnr. P.E. Hertzog, M.Tech.(Ing.)

Medestudieleier: Mnr. J. Raath, M.Tech.(Ing.)

BLOEMFONTEIN

Desember 2002



VERKLARING

Ek, **GERT HENDRIK LAMPRECHT**, verklaar hiermee dat die navorsingsprojek wat vir die verwerwing van die graad **MAGISTER TECHNOLOGIAE: INGENIEURSWESE: ELEKTRIES** aan die Technikon Vrystaat deur my voorgelê word, my selfstandige werk is en nie voorheen deur my of enige ander persoon ter verwerwing van enige kwalifikasie voorgelê is nie.


G.H. Lamprecht
Student

04/12/2002
Datum

DANKBETUIGING

Die skryf van hierdie verhandeling was vir my `n groot leerskool. Gedurende hierdie tyd het ek baie mense betrek wat `n bydrae gelewer het om sodoende `n sukses van die navorsing te maak. Aan daardie persone wat hul allerbeste, hul ondersteuning, emosioneel en finansieel, `n spesiale woord van dank. In die besonder wil ek Johan Kok bedank vir sy hulp rakende die ontwikkeling van die sagteware. Dank aan Suzette vir die taalversorging en my broer vir die rondry in Bloemfontein. Dit word opreg waardeer. Graag bedank ek ook my studieleier, mnr. P.E. Hertzog, vir sy bystand, geduld en vriendskap tydens die studie en voltooiing van die projek.

Laastens dra ek graag hierdie verhandeling op aan my ouers en my gunstelingbeeld, Ans. Dankie vir al die hulp, ondersteuning, raad en leiding, dit word opreg waardeer. `n Besondere woord van dank aan my pa Johan, vir al sy motivering en ondersteuning deur my lewe!

Soli Deo Gloria

UITTREKSEL

Hipotese

Die massa van vryswemmende forelle kan deur elektroniese proessengeringstegnieke bepaal word, met behulp van 'n digitale kamera, 'n rekenaar en digitale seinverwerkings sagteware.

Inhoud van Navorsing

Hierdie navorsing is daarop gemik om deur effektiewe tweedimensionele beeldanalise van lengte, die massa van 'n forel (vis) te bepaal.

Die probleem is daarin geleë dat die massa van 'n forel nie bepaal kan word, sonder dat dit eers gevang en dan geweeg word nie. Die forelboer kan die voorgestelde data wat groei aandui gebruik, om nuttige inligting aangaande gewigseenvormigheid, siektes/kwale en die lewensvatbaarheid van sy forelteling te bepaal. Die studie kan die forelboer help met belangrike bestuurs- en bemarkingsbesluite. As gevolg hiervan is verskillende metodes en tegnieke ondersoek om die lewendige visse se massa ten opsigte van hul lengte te bepaal.

Die probleem kan aan die hand van die volgende stappe opgelos word:

- a) Die verkryging van geldige beelde (foto's) van forelle deur middel van 'n digitale kamera.
- b) Die opdeling van die tweedimensionele beelde in segmente, byvoorbeeld randbeelde.
- c) Die analise van dié beelde met behulp van digitale seinverwerkings-tegnieke.

Inligting is verkry deur 'n literatuurstudie, persoonlike onderhoude en eksperimentele navorsing. Die literatuurstudie sluit geselekteerde, relevante bronne oor digitale beeldverwerking, met die klem op tweedimensionele beeldverwerking, in.

Gevolgtrekking

Die studie het bewys dat die massa van 'n forel wel verkry kan word as die lengte bekend is. Die lengte is bepaal deur 'n duidelike digitale foto van die forel te neem. Hierdie gefotografeerde vis se beeld is deur 'n stiffie vanaf die kamera na 'n rekenaar oorgedra om geanaliseer te word. Die analise van die digitale beeld is gedoen deur 'n sagtewareprogram wat eers die lengte van die voorwerp bepaal het en daarna die lengte omgeskakel het na massa, met behulp van 'n lengtebepalingstelsel.

Dit is moontlik om die lengte van 'n voorwerp te bepaal indien die maksimum lengte van die vis op die digitale beeld sigbaar is. Onduidelike beelde is nie in enige analise gebruik nie.

Daar moet aan vereistes voldoen word om die optimum kwaliteit foto's te verkry, aangesien die foto's die belangrikste verwysingsraamwerk van die hele stelsel is.

SYNOPSIS

A mass of free moving trout can be determined by using a digital camera, a computer and special digital processing software (Troutedge).

The aim of this research is to calculate the mass of a trout with the application of effective two dimensional image analyses on the length of the trout. The problem is that the trout must first be caught in order to be weighed and is difficult as well as a time wasting process. This is why an alternative method is necessary.

If the trout farmer applies the proposed data for indicating the growth of the fish, he can group fish of the same mass together. A further benefit to the trout farmer is that the life expectancy can be determined, and disease can be controlled

This study can be of tremendous help with important management and marketing decisions. Various methods and techniques were researched to calculate a trout's mass by using its length.

The problem can be solved by the application of the following steps:

- a) A digital camera must be used to take valid images of photos of the fish.
- b) Two-dimensional images are divided into segments, for example images of the periphery of the photos.
- c) Digital signal processing techniques will be used to analyze the images.

Information was gathered by means of a literature study, personal interviews and experimental research. The literature study includes relevant, selected sources on digital image processing and the importance of applying two-dimensional image processing.

Conclusion

This study proves that the mass of a trout can be determined if the length is known. To determine the length of the fish, digital photos of fish are used. These photos are transferred by means of a stiffy to a computer analysis. A software program (Troutedge) determines the length of the fish and converts it into the mass with the help of a mass determining table.

The length can only be calculated if the maximum length is seen on a digital image. No indistinct photographs were used in the analyzing process.

INHOUD

	Bladsy
HOOFSTUK 1	1
INLEIDING	1
1.1 Oorsig	1
1.2 Hipotese	1
1.3 Doel van die ondersoek	2
1.4 Afbakening van studieterrein	2
HOOFSTUK 2	4
TWEE DIMENSIONELE BEELDHERKENNING	4
2.1 Menslike waarneming	4
2.1.1 Die persepsie van lig	4
2.1.1.1 Die “Mach band” -effek	5
2.2 Digitale beelddefinisies	6
2.2.1 Algemene waardes	7
2.2.2 Eienskappe van beeldfunksies	7
2.2.2.1 Verskillende soort beeldfunksies	7
2.2.2.2 Metodes van monsterneming	8
2.2.2.3 Videoparameters	9
2.3. Verwerking van digitale beelde	10
2.3.1 Konvolusie	11
2.3.2 Fourier transformasie	11
2.3.3 Beeldstatistieke	12
2.4 Ruis in beelde	13
2.4.1 Fotonruis	14
2.4.2 Termiese ruis	15
2.4.3 Seinruisverhouding	15
2.4.4 CCD-kameras en kameraruis	16
2.4.5 Sensitiwiteit	17
2.4.6 Die vorm van beeldelemente	18
2.4.7 Sluiterspoed	18
2.4.8 Videokameras	19
2.4.9 Uitleestempo	19
2.5 Kontoervoorstelling	19
HOOFSTUK 3	21
HISTOGRAMME EN DRUMPELBEPALING	21
3.1 Histogramme	21
3.1.1 Kontrasstrekking	21
3.2 Berekeninge	24
3.2.1 Binêre berekeninge	24

3.2.2	Ander bewerkinge	25
3.2.3	Berekeninge van 'n afgeleide	26
3.3	Segmentering	26
3.3.1	Drumpelbepaling	27
3.4	Randbepaling	29
3.4.1	Stensilm metode	29
HOOFSTUK 4		32
EKSPERIMENTELE OPSTELLING EN SAGTEWARE		32
4.1	Toerusting en integrasie van substelsels	34
4.1.1	Toerusting gebruik in die navorsing	34
4.1.2	Integrasie van substelsels	35
4.1.2.1	Kamera-opstelling	36
4.1.2.2	Fotografering van beelde	37
4.1.2.3	Oordrag van die gefotografeerde beelde na 'n rekenaar	37
4.1.3	Funksie van die sagteware	37
4.1.3.1	Kleur na grysvlakomskakeling	38
4.1.3.2	Randbepaling	41
4.1.3.3	Koördinaatbepaling	43
4.1.3.4	Lengtebepaling	45
4.1.4	Die wyse waarop die sagteware werk	46
4.1.4.1	Lengte-massaverhouding	52
HOOFSTUK 5		53
STELSELEVALUASIE EN RESULTATE		53
5.1	Stelselevaluasie	53
5.2	Resultate	53
5.3	Gevolgtrekking	58
HOOFSTUK 6		61
SAMEVATTING		61
	Navorsings moontlikhede	62
BYLAE A		63
LITERATUURLYS		1000

LYS VAN FIGURE

Figuur 1.1	Blokdiagram van die stappe om die lengte van die forel te bepaal	2
Figuur 2.1	Sensitiwiteitspektrum van 'n menslike waarnemer.	5
Figuur 2.2	Verskillende vlakke van helderheid is sigbaar.	5
Figuur 2.3	'n Voorstelling van twee optiese illusies.....	6
Figuur 2.4	Deurlopende digitalisering van die beeld.	6
Figuur 2.5	Beskrywing van verskillende tipes beeldfunksies.....	8
Figuur 2.6	Twee verskillende voorbeelde van monsterneming wat gebruik kan word vir beeldverwerking.....	9
Figuur 2.7	(a) Aanduiding van 'n helderheidverspreidingsfunksie van Figuur 2.4.	13
	(b) Aanduiding helderheidshistogram (x-as) teenoor digtheid (y-as) van Figuur 2.4a.	13
Figuur 2.8	Werking van digitale CCD-kameras	17
Figuur 2.9	Voorstelling van die verskille in beeldelement buitelyne tussen kamera en aftaster.....	18
Figuur 2.10	Voorstelling van gekleurde gebied wat gedigitaliseer is en waarvan die randbeeld verkry is	20
Figuur 2.11	Die kontoerbeeldelemente, soos wat dit gebruik word in die reekskode is diagonaal ingekleur.....	20
Figuur 3.1	Kontrasstrekking word in (a) verduidelik deur 'n transformasiefunksie. Beelde met verskillende kontraste is sigbaar in (b),(c) en (d).....	22
Figuur 3.2	Voorstelling van vier verskillende histogramme	23
Figuur 3.3	Voorstellings van binêre bewerkings op beelde.	25
Figuur 3.4	(a) Beeld waarvan die drumpelwaarde bepaal gaan word.....	29
	(b) Helderheidshistogram van die beeld waarmee die drumpelwaarde van die beeld bepaal word.	29

Figuur 3.5	Voorbeeld van 'n grys vlakbeelde wat deur die "Sobel Operator" randbepalingstegniek daarop toegepas is.....	31
Figuur 4.1	Diagram vir die eksperimentele opstelling soos dit gebruik is in die navorsingstudie.....	34
Figuur 4.2	Voorstelling van die integrasie van substelsels.	35
Figuur 4.3	Opstelling van die eksperimenteletoestel	36
Figuur 4.4	Drie kleure wat gebruik word vir beeldverwerking	38
Figuur 4.5	Die drie kleure word benodig vir die berekening van grys vlakbeelde.....	39
Figuur 4.6	'n Funksie soortgelyk aan konvolusie wat 'n kleurbeeldelement verander na 'n skakering van grys.	39
Figuur 4.7	Die helderheidspektrum wat strek van 0 tot 255.....	40
Figuur 4.8	Die bepaling van 'n grysskakering met die samevoeging van die drie kleure (rooi, groen en blou).	40
Figuur 4.9	Randbeeld word verkry deur drie beeldelemente met mekaar te vergelyk.	42
Figuur 4.10	Bepaling van 'n randbeeld met die toepassing van verskillende drumpelwaardes.	42
Figuur 4.11	Vier koördinate wat gebruik word om die lengtebepaling te vergemaklik.	44
Figuur 4.12	Voorstelling van moontlike koördinate tussen die vier hoofpunte waarvandaan die lengtes bepaal word.	45
Figuur 4.13	Pythagoras se stelling word gebruik om die lengte tussen twee beeldelemente te bepaal.	46
Figuur 4.14	Metode hoe 'n beeld in die program gelaai word.	47
Figuur 4.15	Kleur- na 'n grys vlakbeeldomskakeling.	48
Figuur 4.16	Bepaling van die randbeeld.	48
Figuur 4.17	Bepaling van die vier koördinate en die nodige beeldelemente	49

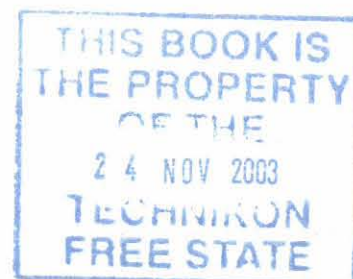
Figuur 4.18	Die resultate wat deur onsigbare beeldelemente beïnvloed is.	50
Figuur 4.19	Lengtebepaling en kalibrering van 'n voorwerp.....	51
Figuur 4.20	Tabel van lengte teenoor massa.....	51
Figuur 4.21	Grafiese voorstelling van die verband tussen die lengte die massa van forelle.	52
Figuur 5.1	Die grafieke stel die berekende lengte teenoor die fotograferingsafstand voor.	56
Figuur 5.2	Lengte berekening by verskillende horisontale oriëntasies.....	58



Central University of
Technology, Free State

LYS VAN TABELLE

Tabel 2.1	Algemene waardes vir digitale beelde	7
Tabel 2.2	Verskillende funksies wat op beelde toegepas kan word	8
Tabel 2.3	Standaard videoparameters van NTSC, PAL en SECAM.....	10
Tabel 2.4	Teoretiese en gemete SNR waardes van vyf verskillende kameras	15
Tabel 3.1	Bewerkinge wat op beeldverwerking gebruik word.....	26
Tabel 3.2	Die stensils wat gebruik word vir die Sobel- en Roberts- randbepalingstegnieke.....	30
Tabel 5.1	Kalibrering van die stelsels met die 50 mm voorwerp.	54
Tabel 5.2	Kalibrering van die stelsels met die 137 mm voorwerp.	54
Tabel 5.3	Kalibrering van die stelsels met die 175 mm voorwerp.	54
Tabel 5.4	Kalibrering van die stelsels met die 435 mm voorwerp.	55



KODELYS

Kodelys 3.1 : Die randbepaling van 'n ligte beeld.	27
Kodelys 3.2 : Die randbepaling van 'n donker beeld.	27



LYS VAN TERME

kontinue ruimte	Continuous space
diskrete ruimte	Discrete space
raamvangers	Frame grabbers
skandeer	Scan
vertooneenheid	Display
Sigsag	Zigzag
distributief	Distributive
diskreet	Discrete
stelling	Theorem
grootte	Magnitude
monster	Sample mean
gebied van belang	Region of interest
kraakkode	Crack code
rykode	Run code
stimulus	Simulus
skyfie	Chip
donker stroom	dark current
opstelling	Array
willekeurig	Arbitrary
drumpel	Threshold
algoritmes	Algorithms
Laplace transformasie	Laplace transform
stensil	Template
operateur	Operator
lading-gekoppelde toestelle	Charge coupled devices
aftaster	Desktop scanner
verkort dinamiese strek	Compresses dynamic range
vergrootte dinamiese strek	Expanded dynamic range
wisselpunt	Floating point

kleurfilteropstelling

Colour filter array

bis

Bit

druk

Click

Fourier-transformasie

Fourier Transform

keuselys

menu bar

Opsoektabel

Lookup Table

LYS VAN AKRONIEME EN AFKORTINGS

DSV	digitale seinverwerking
ROI	“region of interest”
2D	tweedimensioneel
FFT	“Fast Fourier Transform”
NTSC	“National Television Sub-committee”
PAL	“phase alternating line”
SECAM	“sequential chrominance signal and “memory”
SNR	“signal to noise ratio”
CCD	“charge coupled device”
ADC	“analog to digital convertor”
CID	“charge injection device”
CFA	“colour filter array”
CRT	katodestraalbuis
LoG	“Laplacian of Gaussian”
GUI	“graphical user interface”
ns	nanosekonde
ms	millisekonde
nm	nanometer

HOOFSTUK 1

INLEIDING

1.1 Oorsig

Wanneer daar na hierdie bladsy gekyk word, word elke klein swart voorwerp/vorm op die wit bladsy gevisualiseer, geanaliseer en geïnterpreteer. 'n Aantal van die swart voorwerpe/vorms word geassosieer met 'n bekende woord; 'n aantal bekende woorde saam vorm 'n betekenisvolle sin. Spelfoute kan waargeneem word, veranderings in letter-groottes is sigbaar of selfs spasies wat u i t v e r h o u d i n g i s, kan opgemerk word.

Die menslike oog kan veranderinge op 'n klein skaal baie vinnig waarneem. As hiërdie bladsy 'n armlengte ver weggehou word, sal die aandag eerste op die veranderinge van lettergroottes en -spasies in die vorige paragraaf val. Uit honderde letters en spasies word die sonderlinge/ongelykes eerste uitgeken [20, pp. 3-7].

In die verhandeling sal daar slegs op tweedimensionele (2D) beelde gekonsentreer word. Sommige konsepte en tegnieke wat bespreek word kan egter ook op driedimensionele beelde toegepas word.

1.2 Hipotese

Die massa van vryswemmende forelle kan deur elektroniese prosesseringstegnieke bepaal word met behulp van 'n digitale kamera, 'n rekenaar en digitale seinverwerkingsagteware.

Hierdie navorsing is daarop gemik om met effektiewe beeldanalise die massa van 'n forel(vis) te bepaal. Die probleem kan aan die hand van verskeie stappe uiteengesit word:

- a) Die verkryging van geldige beelde van forelle deur middel van 'n digitale kamera.
- b) Die opdeling van die tweedimensionele beelde in segmente.
- c) Die analise van die gesegmenteerde beelde met behulp van digitale seinverwerkingstegniese.

1.3 Doel van die ondersoek

Die oorkoepelende doelstelling van hierdie studie is om 'n doeltreffende tweedimensionele beeldverwerkingstelsel te ontwikkel. Om hierdie doelstelling te bereik, kan die volgende vier punte as spesifieke doelwitte vir die studie uiteengesit word:

- a) Konstruksie van 'n eksperimentele toestel vir die fotografering van visse.
- b) Bepaling van die verband tussen gemete afmetings en die massa van 'n vis.
- c) Ontwikkeling van sagteware vir beeldanalise ten einde 'n gefotografeerde vis te korreleer met 'n standaard lengtebepalingstelsel.
- d) Ontwikkeling van sagteware vir die omskakeling van die gefotografeerde afmetings na die massa van visse.
- e) Evaluering ten opsigte van die akkuraatheid van die stelsel.

1.4 Afbakening van studieterrein

Die blokdiagram in Figuur 1.1 is 'n voorstelling van die stappe wat gevolg is om die nodige resultate te verkry. Elk van die blokke word kortliks beskryf:



Figuur 1.1 Stappe om die lengte van die forel te bepaal

- Verkryging van gefotografeerde beelde van forelle:
Voor enige beeldanalise gedoen kan word, is dit belangrik om gefotografeerde beelde van forelle te verkry. Digitale foto's van verskillende grootte visse in 'n dam word versamel. Dië foto's word tydens die eksperiment vir analise gebruik.
- Omskakeling van gefotografeerde beelde in grysvlak- en randbeelde:
Die gefotografeerde beelde word van kleur na grysvlak omgeskakel, waarna die randbeeld bepaal sal word.
- Analise van grysvlak- en randbeelde om deur die berekende lengte die massa te verkry:
Hierdie proses behels die analise van die grysvlak- en randbeeld om die lengte van die forel te bepaal, waarna die massa deur middel van 'n standaard lengtebepalingstelsel hanteer word.

TWEEDIMENSIONELE BEELDHERKENNING

In hierdie hoofstuk word daar na die menslike visualiseringstelsel verwys. Verskeie digitale beelddefinisies, funksies wat op beelde toegepas word en verskeie vorms van ruis word ondersoek. Die werking van CCD-kameras en die voorstelling van kontoere binne 'n beeld word ook bespreek.

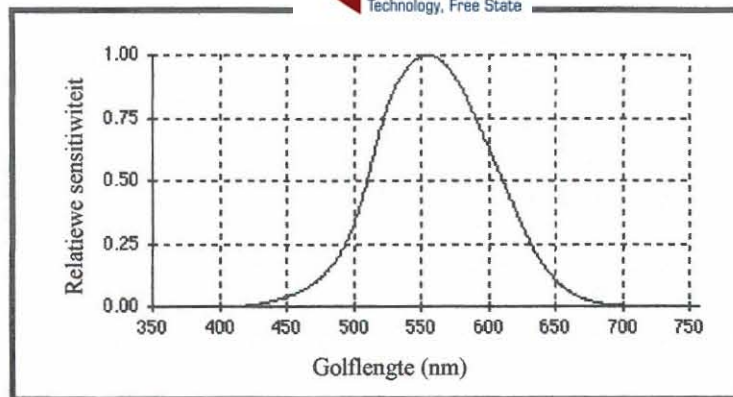
2.1 Menslike waarneming

Die mens se oog is baie belangrike, maar kompleks sintuig. Dit is belangrik om die eienskappe en beperkinge van die menslike visuele waarnemingstelsel te verstaan. 'n Gewone kamera funksioneer op 'n soortgelyke wyse as die menslike visualiseringstelsel. Navorsing wat deur sielkundiges gedoen is, verskaf belangrike insig rakende die menslike visualiseringstelsel [1, pp. 1-3].

2.1.1 Die persepsie van lig

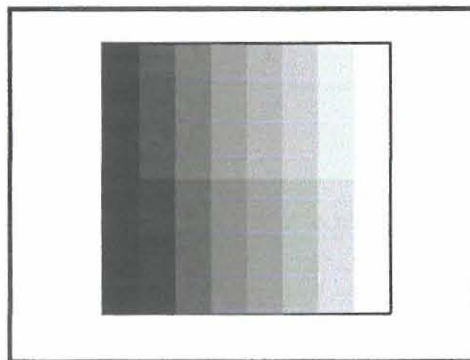
Volgens Webster [24] is lig: "Radiant energy which, by its action on the organs of vision, enables them to perform their function of sight." Sigbare lig is 'n vorm van elektromagnetiese radiasie wat in 'n relatiewe nou frekwensiespektrum van die elektromagnetiese spektrum voorkom (binne 'n band van ongeveer 350 – 780 nm). 'n Ligbron word gekarakteriseer deur sy radiasie-uitstralings tempo en verskaf lig met 'n spesifieke golflengte [20, p. 25].

Die sigsensitiwiteitspektrum van 'n menslike waarnemer word in Figuur 2.1 voorgestel [19, p. 32].



Figuur 2.1 Sensitieweitspektrum van 'n menslike waarnemer

2.1.1.1 Die “Mach band”-effek

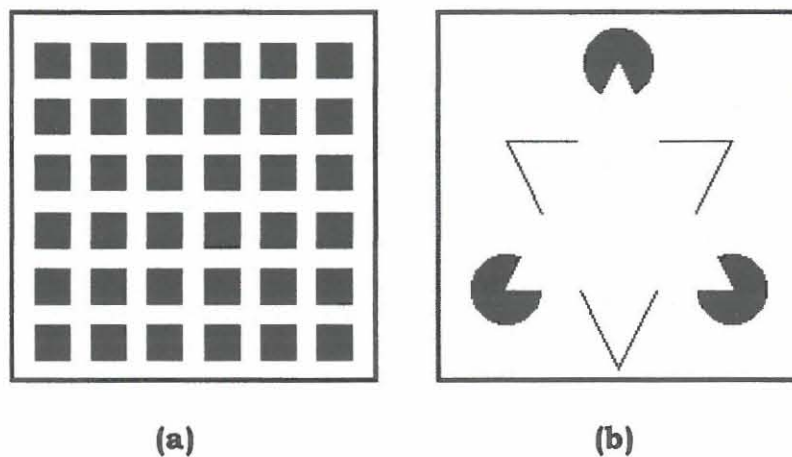


Figuur 2.2 Verskillende vlakke van helderheid is sigbaar

Die helderheid van die vertikale grys strepe in Figuur 2.2 is konstant oor die breedte van elke streep, met 'n eenvormige toename in helderheid in die aangrensende intervalle. Ten spyte van die feit dat die helderheid oor elke vertikale interval konstant is, word daar 'n oënskynlike klein vermindering in helderheid (links) ten opsigte van die ware intensiteit waargeneem, net voor die oorgang na 'n volgende interval. Met die oorgang na 'n volgende interval, word 'n oordrewe helderheid (regs), ten opsigte van die fisiese waarde van die helderheid van die interval waargeneem. Hierdie optiese illusie staan bekend as die “Mach band”-effek [10, pp. 53-54][19, pp. 33-35].

Nog twee voorbeelde van optiese illusies word in Figure 2.3a en 2.3b getoon. Oor die algemeen word voorwerpe en lyne wat na aan mekaar

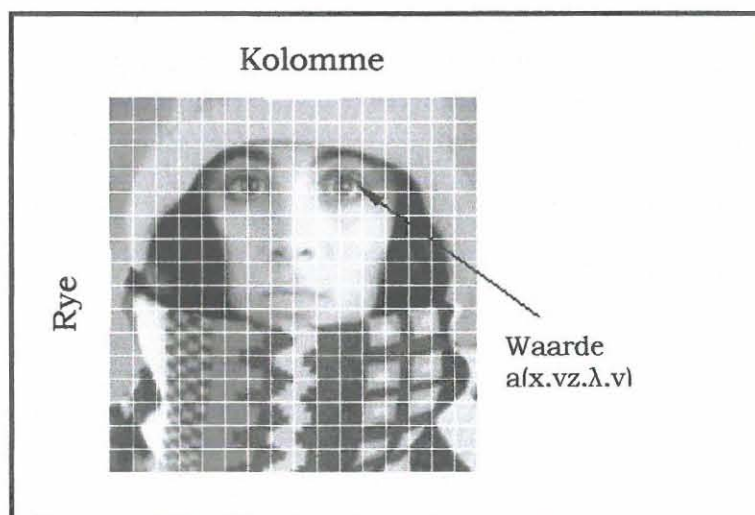
geplaas word, maklik deur die menslike oog saamgegroeper om die illusie van 'n ander voorwerp te vorm (Figuur 2.3(a)) [20, pp. 45-49]. Die tweede illusie (Figuur 2.3(b)); Kaniza se driehoek) beeld 'n uitstekende kontras en valse kontoere uit. Beide dié voorbeelde kan nie deur stelsel-georiënteerde aspekte van waarneembare persepsies beskryf word nie [1, pp. 208-212].



Figuur 2.3 Voorstelling van twee optiese illusies

2.2 Digitale beelddefinisies

'n Digitale beeld $a(m,n)$ in 'n 2D diskretegebied word afgelei van 'n analoog-beeld $a(x,y)$ in 'n 2D-deurlopende gebied deur middel van 'n steekproefproses wat as digitalisering bekend staan [2, p. 2][14, pp. 2-5]. Die effek van digitalisering kan aan die hand van Figuur 2.4 verduidelik word.



Figuur 2.4 Deurlopende digitalisering van die beeld [25]

Die beeld in Figuur 2.4 is verdeel in $N = 16$ rye en $M = 16$ kolomme. Die waarde van elke beeldelement word bereken deur die gemiddelde helderheid af te rond tot die naaste heelgetal.

2.2.1 Algemene waardes

Die relatiewe posisie en beeldinligting van 'n sekere beeldelement word gewoonlik as 'n numeriese waarde gesien. Tabel 2.1 toon tipiese waardes vir rye, kolomme en grysvlakke [1, pp. 59-61]. Sodra 'n grysvlakbeeld omgeskakel word na 'n binêre beeld is daar slegs twee grysvlakke: swart en wit, of te wel '0' of '1' [1, p. 61-62].

Tabel 2.1 Algemene waardes vir digitale beelde

Parameter	Simbool	Tipiese indelings
Rye	N	256, 512, 525, 625, 1024, 1035
Kolomme	M	256, 512, 768, 1024, 1320
Grysvlakke	L	2, 64, 256, 1024, 4096, 16384

2.2.2 Eienskappe van beeldfunksies

Daar is 'n verskeidenheid van maniere om beeldfunksies te karakteriseer. Die onderstaande uiteensetting verduidelik 'n paar eienskappe van tipiese funksies wat op 'n beeld toegepas kan word.

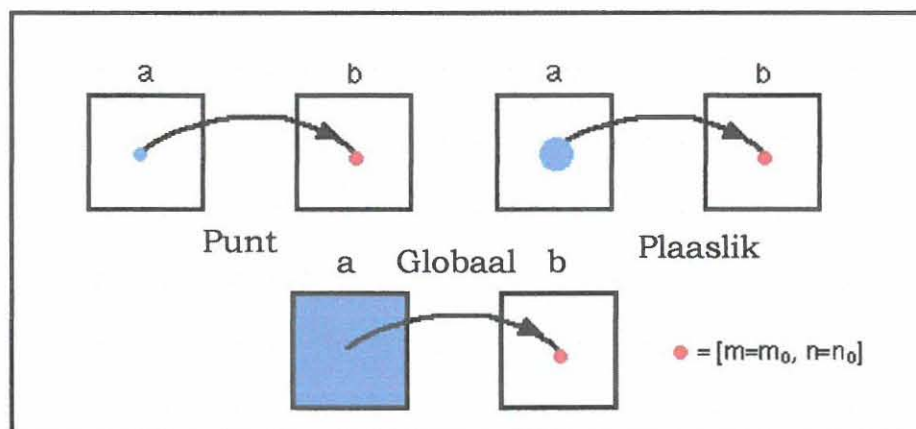
2.2.2.1 Verskillende soort beeldfunksies

Die verskillende funksies wat op digitale beelde toegepas kan word om 'n insetbeeld ($a(m,n)$) om te skakel na 'n uitsetbeeld ($b(m,n)$), word in Tabel 2.2 uiteengesit [1, pp. 99-125].

Tabel 2.2 Verskillende funksies wat op beelde toegepas kan word

Beeldfunksies	Karakterisering
* Punt	Die uitsetwaarde by 'n spesifieke koördinaat is slegs afhanklik van die insetwaarde by dieselfde koördinaat.
* Plaaslik	Die uitsetwaarde by 'n spesifieke koördinaat is afhanklik van die insetwaarde in die plaaslike omgewing van dieselfde koördinaat.
* Globaal	Die uitsetwaarde by 'n spesifieke koördinaat is afhanklik van al die waardes van die insetbeeld.

In die navorsingstudie is die plaaslike beeldfunksie gebruik om 'n rand van 'n beeld te verkry (sien afdeling 4.1.3.2). Tabel 2.2 kan ook soos volg deur Figuur 2.5 voorgestel word.



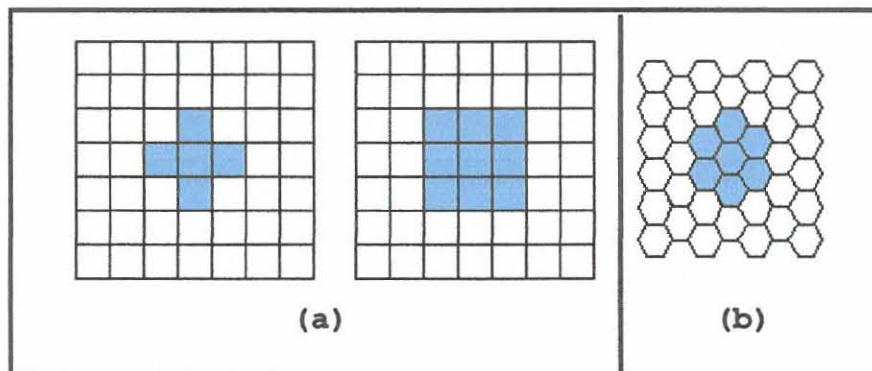
Figuur 2.5 Beskrywing van verskillende tipe beeldfunksies

2.2.2.2 Metodes van monsterneming

Wanneer daar besluit word hoe monsterneming op 'n digitale beeld toegepas kan word, moet die fisiese rangskikking van die beeldelemente binne die digitale beeld ondersoek word. Die mees algemene metode is die

reghoekige mosaïekblokkies, maar daar is beperkinge wat kan veroorsaak dat ander mosaïekblokkies meer van belang kan wees [1, p. 63]. Reghoekige en heksagonale monsters is geneem om die verband tussen belangrike punte wat in beeldverwerking gebruik word te vergelyk en te verstaan [7, pp. 30-31].

- *Reghoekige monsterneming* – In die meeste gevalle word 'n beeld verwerk deur 'n vierkantige rooster oor die beeld te plaas soos in Figuur 2.4 [18, pp. 18-21]. Dit verskaf die voorbeelde soos aangedui in Figure 2.69(a) en 2.6(b).
- *Heksagonale monsterneming* – Dit word gebruik vir funksies wat sirkulêr simmetries en/of groepsgebonde binne 'n bepaalde gebied is (sien Figuur 2.6(c)). Heksagonale patrone benodig 13.4% minder monsters as reghoekige monsterneming [10, p. 92]. Dit versnel die beeldverwerkingsproses.



Figuur 2.6 Twee verskillende voorbeelde van monsterneming wat gebruik kan word vir beeldverwerking

2.2.2.3 Videoparameters

Die verskillende videostelsels is kortliks ondersoek om die moontlike effek wat dit op beeldverwerking het te identifiseer. Vervolgens die algemene videostelsels:

Eerstens: NTSC, bekend as “National Television Subcommittee”, kom hoofsaaklik in Noord-Amerika en Japan voor.

Tweedens: PAL, bekend as “phase alternating line”, kom hoofsaaklik in dele van Europa, Asië, Suid-Amerika en Afrika voor.

Derdens: SECAM - “Sequential Couleur à Mêmorie (sequential chrominance signal and memory)”.

Tabel 2.3 verskaf eienskappe rakende die tegnieke [10, pp. 82-83].

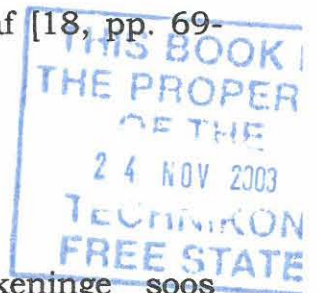
Tabel 2.3 Standaard videoparameters van NTSC, PAL en SECAM

Standaard eienskappe	NTSC	PAL	SECAM
beeld / sekonde	29.97	25	25
lyne / beeld	525	625	625
(horis./vert.) = voorkomsverhouding	4:3	4:3	4:3
vervlegging	2:1	2:1	2:1
us / lyn	63.56	64	64

In 'n vervlegte beeld word slegs onewe lyne (1,3,5,...) in die helfte van die toegelate tyd geskandeer. Die ewe lyne (2,4,6,...) word in die oorblywende tyd geskandeer. Die rede vir vervlegging is om die persepsie van flikkering op die vertooneenheid te verminder. Die analise van bewegende voorwerpe vereis spesiale aandag by die gebruik van video's met vervlegging, om sodoende sigsagrande te voorkom. Die hoeveelheid rye (N) van 'n videobron, stem 1:1 ooreen met die lyne in die videobeeld. Die hoeveelheid kolomme hang egter af van die elektroniese gedeelte wat verantwoordelik is vir die digitalisering van die beeld. Die toepassing van verskillende raamvangere op 'n beeld, met behulp van 'n enkele videokamera, kan $M = 384, 512, \text{ of } 768$ horisontale beeldelemente per lyn verskaf [18, pp. 69-79][20, pp. 108-111].

2.3 Verwerking van digitale beelde

Verwerking van digitale beelde sluit wiskundige berekeninge soos konvolusie, Fourier-analise en statistiese uiteensetting en toebehore vir



beeldmanipulasie in. Daar bestaan nog verskeie ander tegnieke, maar dit is nie in dié studie van toepassing op die verlangde resultate nie.

2.3.1 Konvolusie

Konvolusie word gebruik in beeldverwerking by die duideligmaking, verdowing, ruisvermindering en die uitlig van 'n randbeeld. In hierdie afdeling gaan daar kortliks gekyk word na die wiskundige uiteensetting van konvolusie. Daar bestaan verskeie moontlike notasies om die konvolusie van twee seine as 'n uitset uit te druk. Die mees algemene vergelyking is:

$$c = a \otimes b \quad (1)$$

waar: c = die konvolusie tussen twee seine a en b is

a = insetsein a

b = insetsein b

Belangrike wiskundige eienskappe aangaande konvolusie kan as volg uiteengesit word:

- Konvolusie is verwisselend.

$$c = a \otimes b = b \otimes a \quad (2)$$

- Konvolusie is assosiatief of kan met mekaar verbind word.

$$c = a \otimes b(a \otimes c) = (a \otimes b) \otimes c = a \otimes b \otimes c \quad (3)$$

- Konvolusie is distributief, met ander woorde verdelend of toedelend.

$$c = a \otimes (b + d) = (a \otimes b) + (a \otimes d) \quad (4)$$

Bogenoemde is waar indien a, b, c en d voortdurende of diskrete beelde is [2, pp. 149–159].

2.3.2 Fourier transformasie

Die formules vir die voorwaartse Fourier- en inverse Fourier-

transformasies kan soos volg uiteengesit word. Gestel 'n beeld word gegee as a , en sy Fourier-transformasie as A , dan sal die voorwaartse Fourier-transformasie die sein vanaf die ruimtelike strek omskakel na die frekwensiestrek, wat altyd 'n voortdurende ruimte is [7, pp. 61-71] [10, p. 141].

Die term ruimtelike strek verwys na die kombinasie van beeldelemente wat saam 'n beeld vorm. Ruimtelike strekmetodes is prosedures wat op die spesifieke beeldelemente toegepas kan word [7, pp. 61-71] [10, p. 141].

$$\text{Voorwaarts} \quad A = F(a) \quad (5)$$

waar a = 'n enkele beeld

A = Fourier-transformasie van a

Inverse Fourier-transformasie is die teenoorgestelde van die voorwaartse Fourier-transformasie. Deur middel van inverse Fourier-transformasie kan die frekwensiestrek van 'n sein omgeskakel word na die ruimtelike strek (voorgestel in vergelyking 6).

$$\text{Inverse} \quad a = F^{-1}\{A\} \quad (6)$$

waar a = 'n enkele beeld

A = Fourier-transformasie van a

Die Fourier-transformasie is dus 'n omkeerbare aksie, waar:

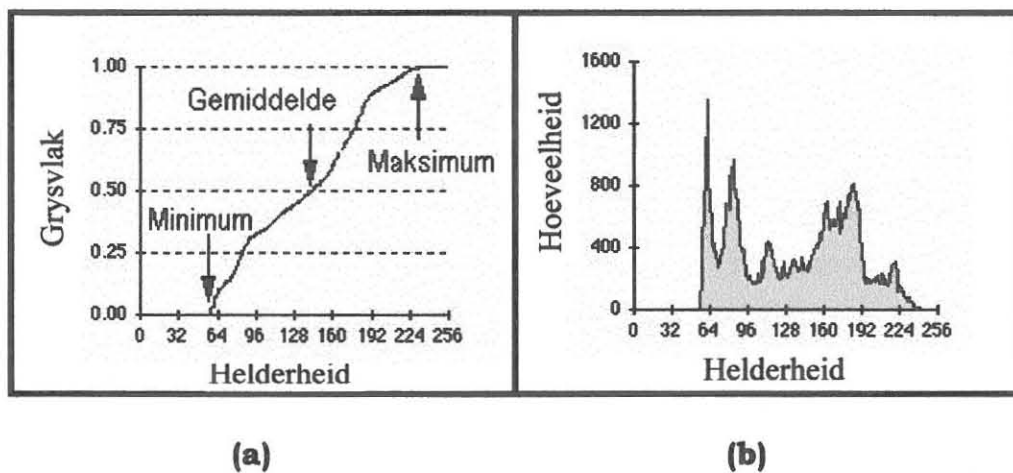
$$a = F^{-1}\{F\{a\}\} \quad \text{en} \quad A = F\{F^{-1}\{A\}\} \quad (7)$$

2.3.3 Beeldstatistieke

Die grafiese voorstelling van beeldstatistieke (beelde en subbeelde), is algemeen in beeldverwerking (sien Figuur 2.7). Die gebruik van beeldstatistieke is sorgelyk aan die konsep van opdeling van sein-amplitudes. Daar kan moontlik twee eienskappe in 'n gegewe gebied waargeneem word, naamlik [25, p.16]:

- a) Daar bestaan 'n verspreidingsfunksie rakende die helderheid binne die gebied.
- b) Daar bestaan 'n digtheidsfunksie van die helderheid binne die gebied.

Die helderheidsverspreidingsfunksie van die beeld in Figuur 2.4 word in Figuur 2.7(a) verskaf. Die helderheidshistogram van Figuur 2.4, wat proporsioneel is aan die moontlike helderheidsdigtheidsfunksie, word in Figuur 2.7(b) getoon. Die hoogte van die histogram stem ooreen met die hoeveelheid beeldelemente by 'n gegewe helderheid. Die inligting is verkry met behulp van "Scion Image"-sagtewareprogram [22].



Figuur 2.7 (a) Aanduiding van die helderheidsverspreidingsfunksie van Figuur 2.4
(b) Aanduiding van die helderheidshistogram (x-as) teenoor digtheid (y- as) van Figuur 2.4(a)

Beide die verspreidingsfunksie en die histogram, wat verteenwoordigend is van 'n spesifieke gebied binne Figuur 2.4, kan beskou word as statistiese beskrywings [25, pp. 17-19].

2.4 Ruis in beelde

Beelde wat deur moderne sensors verkry is, kan deur verskeie tipes ruis beïnvloed word. Ruis kan 'n verwronge beeld wees as gevolg van 'n skaduwee of 'n beeld wat uit fokus is en ook verskeie ander redes.

2.4.1 Fotonruis

Moderne CCD-kameras is sensitief genoeg om fotone individueel waar te neem. Fotonruis ontstaan as gevolg van die fundamentele oorsprong van foton-ontwikkeling. Foton-ontwikkeling word beheer deur die wette van kwantumfisika, wat veroorsaak dat daar ook nie verwys kan word na 'n gemiddelde hoeveelheid fotone binne 'n gegewe raam nie. Die moontlike verspreiding van fotone binne 'n gegewe raam met tyd (T in sekondes) staan bekend as vergiftiging. Indien 'n CCD verlig word met byvoorbeeld 100 fotone per beeldelement per sekonde, sal die ware aantal fotone wat enige beeldelement in 'n sekonde tref, 'n willekeurige hoeveelheid wees [2, p. 26].

Die sein- tot ruisverhouding (SNR) is baie belangrik, selfs al is daar geen ander ruisbronne binne die betrokke beeld nie. Die statistiese fluktuasie wat geassosieer word met fotonwaarneming oor 'n sekere tyd (T), sal veroorsaak dat daar 'n bepaalde seinruisverhouding (SNR) is. Die formule wat gebruik word om SNR te bepaal, is:

$$\text{SNR} = 10 \log_{10}(\rho T) \text{ dB} \quad (8)$$

waar ρT = gemiddelde waarde vir helderheid en tyd.

Dit is duidelik vanuit vergelyking 8 vir fotonruis, dat dit wel moontlik is om die SNR te verhoog deur die integrasietyd van 'n beeld te verleng en daarom meer fotone van 'n spesifieke beeld vas te vang. Daar is slegs 'n beperkte kapasiteit (hoeveelheid) beeldelemente wat binne 'n CCD-kamera vasgelê kan word. Die beperkte kapasiteit (c) beteken dat die maksimum SNR van 'n CCD-kamera per beeldelement gegee kan word deur [25, p. 41]:

$$\text{SNR} = 10 \log_{10}(c) \text{ dB} \quad (9)$$

waar c = kapasiteit (hoeveelheid beeldelemente per CCD)

Die teoretiese en gemete SNR-waardes van vyf verskillende kameras word in Tabel 2.8 verskaf.

2.4.1 Fotonruis

Moderne CCD-kameras is sensitief genoeg om fotone individueel waar te neem. Fotonruis ontstaan as gevolg van die fundamentele oorsprong van foton-ontwikkeling. Foton-ontwikkeling word beheer deur die wette van kwantumfisika, wat veroorsaak dat daar ook nie verwys kan word na 'n gemiddelde hoeveelheid fotone binne 'n gegewe raam nie. Die moontlike verspreiding van fotone binne 'n gegewe raam met tyd (T in sekondes) staan bekend as vergiftiging. Indien 'n CCD verlig word met byvoorbeeld 100 fotone per beeldelement per sekonde, sal die ware aantal fotone wat enige beeldelement in 'n sekonde tref, 'n willekeurige hoeveelheid wees [2, p. 26].

Die sein- tot ruisverhouding (SNR) is baie belangrik, selfs al is daar geen ander ruisbronne binne die betrokke beeld nie. Die statistiese fluktuasie wat geassosieer word met fotonwaarneming oor 'n sekere tyd (T), sal veroorsaak dat daar 'n bepaalde seinruisverhouding (SNR) is. Die formule wat gebruik word om SNR te bepaal, is:

$$\text{SNR} = 10 \log_{10} (\rho T) \text{ dB} \quad (8)$$

waar ρT = gemiddelde waarde vir helderheid en tyd.

Dit is duidelik vanuit vergelyking 8 vir fotonruis, dat dit wel moontlik is om die SNR te verhoog deur die integrasietyd van 'n beeld te verleng en daarom meer fotone van 'n spesifieke beeld vas te vang. Daar is slegs 'n beperkte kapasiteit (hoeveelheid) beeldelemente wat binne 'n CCD-kamera vasgelê kan word. Die beperkte kapasiteit (c) beteken dat die maksimum SNR van 'n CCD-kamera per beeldelement gegee kan word deur [25, p. 41]:

$$\text{SNR} = 10 \log_{10}(c) \text{ dB} \quad (9)$$

waar c = kapasiteit (hoeveelheid beeldelemente per CCD)

Die teoretiese en gemete SNR-waardes van vyf verskillende kameras word in Tabel 2.8 verskaf.

Tabel 2.4 Teoretiese en gemete SNR-waardes van vyf verskillende kameras [25]

Kamera	C(#)	Teoretiese SNR (dB)	Gemete SNR (dB)	Beeldelement-grootte ($\mu\text{m} \times \mu\text{m}$)	Diepte # / μm^2
C – 1	32	55	45	6.8 x 6.8	892
C – 2	340	55	55	22.0 x 22.0	702
C – 3	32	45	43	6.8 x 6.8	692
C – 4	400	56	52	23.0 x 23.0	756
C – 5	40	46	43	11.0 x 5.5	661

2.4.2 Termiese ruis

Elektrone kan vrygestel word van die CCD-materiaal deur termiese vibrasie. Die termiese elektrone is nie onderskeibaar van die 'ware' foto-elektrone nie. Deur verkoeling van die CCD-skyfie, is dit moontlik om die hoeveelheid termiese elektrone te verminder wat termiese ruis of 'n 'donker stroom' veroorsaak. 'n Toename in integrasietyd (T), veroorsaak 'n verhoging in die hoeveelheid termiese elektrone. Die waarskynlike verspreiding van termiese elektrone is ook 'n vergiftigingsproses, waar die "rate parameter" 'n verhogingsfunksie van die temperatuur is. Sommige optiese skyfies van kameras word deur middel van die Peltier verkoelingselement verkoel, ten einde die donker stroom te beperk.

2.4.3 Seinruisverhouding

Die SNR van digitale kameras word in dB uitgedruk. Gestel die helderheid van 'n beeld varieer tussen bepaalde limiete, dan kan die SNR [25, pp. 18-19] soos volg gedefinieer word:

$$SNR = 20 \log_{10} \frac{(a_{maks} - a_{min})}{s_n} \text{ dB} \quad (10)$$

waar a_{maks} = maksimum helderheid
 a_{min} = minimum helderheid
 s_n = standaardafwyking

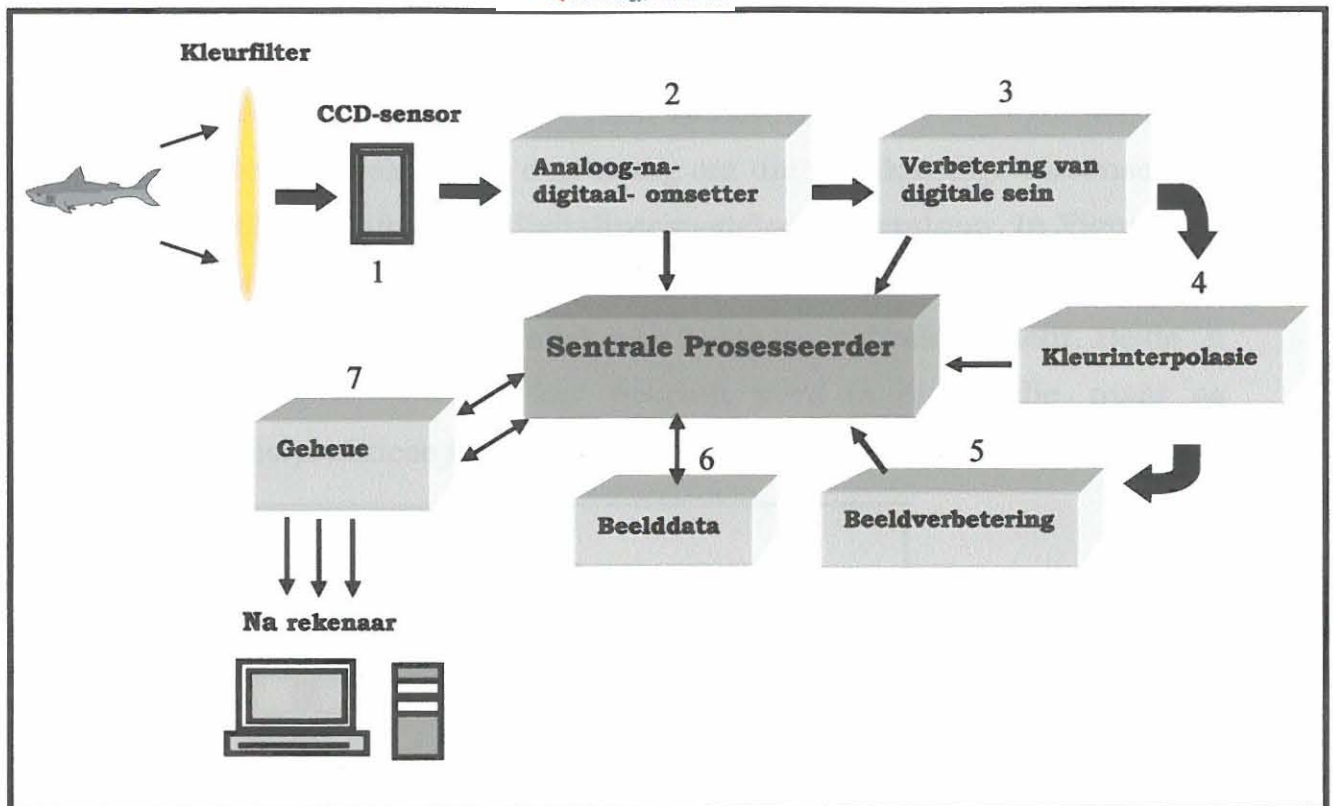
2.4.4 CCD-kameras en kameraruus

In 'n CCD-opstelling word lig deur 'n baie klein fotosensitiewe halfgeleier waargeneem. Soos lig op die halfgeleier skyn, word dit met energie gelaai [2, pp. 24-28]. Die hoeveelheid energie wat in 'n spesifieke halfgeleier oor 'n sekere tyd gelaai word, is direk eweredig aan die hoeveelheid lig wat daarop skyn. Dit is nodig dat elke halfgeleier se energie oorgeplaas word na 'n 'stoorarea'¹, waar die energie na 'n spanning omgeskakel word. Omdat die lading tussen die fotosensitiewe halfgeleier en die 'stoorarea' oorgeplaas kan word, staan dit bekend as 'n 'gekoppelde lading' [20, pp. 98-101].

In Figuur 2.8 word die werking van 'n enkel digitale CCD-kamera getoon:

- Die ware beeld gaan deur 'n kleurfilteropstelling (CFA) en word deur die CCD-sensor waargeneem.
- 'n Analoo-na-digitaalomsetter verander die analoo-sein na 'n digitale sein.
- Die verbetering van die digitale sein vind dan plaas.
- Kleurinterpolasie vind na die verbetering van die digitale sein plaas, met ander woorde dit is die byvoeging van kleur-beeldelemente wat vermis word.
- Algehele beeldverbetering om die beeld se kwaliteit te verhoog.
- Beelddata word verpak (saamgepers) en na die geheue gestuur om tydelik gestoor te word.
- Dit kan op 'n latere stadium na 'n rekenaar oorgedra word vir verdere beeldverwerking [12, pp. 380-381].

¹ Stoorarea is 'n gebied binne die CCD-skyfie wat soos 'n kapasitor werk om energie te stoor.



Figuur 2.8 Werking van digitale CCD-kameras [12, p. 380]

2.4.5 Sensitiwiteit

Daar bestaan twee moontlike tegnieke om die sensitiwiteit van 'n kamera te beskryf.

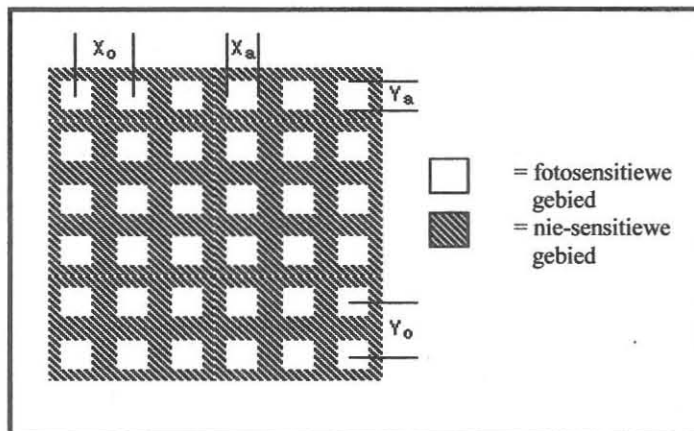
Eerstens: die absolute sensitiwiteit. Dit is waar die minimum aantal waarneembare foto-elektrone bepaal word.

Tweedens: die relatiewe sensitiwiteit is die hoeveelheid foto-elektrone wat benodig word om van een digitale helderheidsvlak na 'n volgende oor te skakel.

'n Wetenskaplike graad CCD-kamera (in Tabel 2.8), benodig slegs agt foto-elektrone (naastenby 16 fotone) om te onderskei tussen die twee grys vlakke binne 'n digitale voorstelling van 'n beeld. Vir 'n relatiewe goedkoper kamera, wat vir die studie gebruik is, word 256 foto-elektrone (naastenby 512 fotone) gebruik om twee grys vlakke van mekaar te skei [25, p. 39].

2.4.6 Die vorm van beeldelemente

Die beeldelemente in Figuur 2.4 kom vierkantig voor, ten einde die beeld deurlopende te bedek. Dit is dus nodig om die meetkundige uiteensetting van 'n spesifieke kamera of digitaliseringstelsel te verstaan. In Figuur 2.9 word 'n vergelyking getref tussen die parameters (buitelyne) van 'n kamera en 'n aftaster en die effek van beide op 'n beeldelement. Beide 'n aftaster en digitale kamera kan gebruik word in die studie, maar as gevolg van die praktiese toepassing is die gebruik van 'n kamera beter.



Figuur 2.9 Voorstelling van verskille in beeldelement buitelyne tussen kamera en aftaster

Die beeldelement se middelpunte word voorgestel deur parameters X_0 en Y_0 . Die ligsensitiewe oppervlak van die kamera word deur parameters X_a en Y_a aangedui.

2.4.7 Sluiterspoed

Die tydsduur wat 'n beeld blootgestel word wanneer fotone versamel word, kan varieer van kamera tot kamera of kan varieer as gevolg van die basis van die videoformaat (Tabel 2.3). Sluiterspoed² staan bekend as die blootstellingstyd wat met die terme van fotografie te doen het, alhoewel geïntegreerde tyd 'n meer toepaslike beskrywing sou wees [6, p. 74].

² Sluiterspoed is die tempo waarteen die sluiser van die kamera oop en toe maak.

2.4.8 Videokameras

Sluiterspoedwaardes so laag soos 500 ns is beskikbaar in kommersiële CCD-videokameras, alhoewel die konvensionele spoed vir videobeeld, 33.37 ms (NTSC) en 40.4 ms (PAL, SECAM) is. Die kosteverskil vir beter sluiterspoedspesifikasies is egter ook baie groot.

2.4.9 Uitleestempo

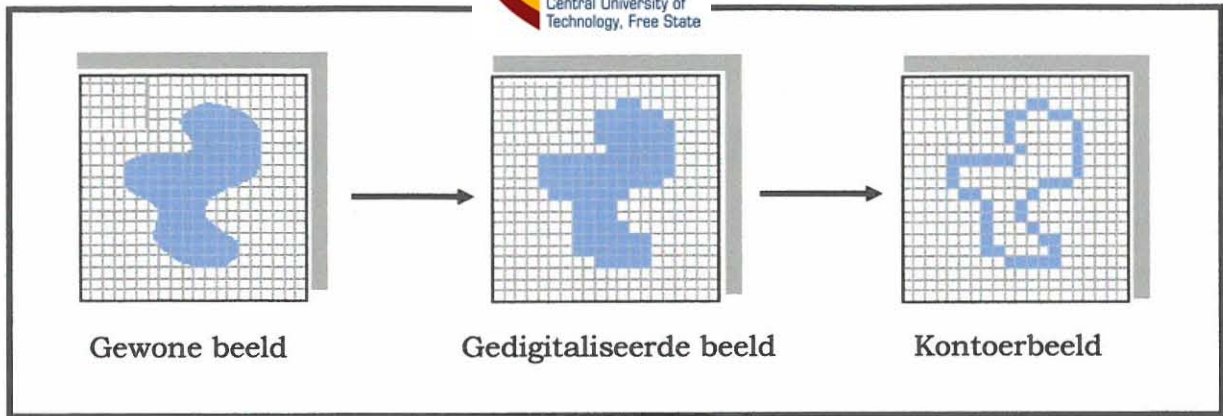
Die term uitleestempo staan bekend as die tyd waarteen data vanaf 'n sensorskyfie gelees word om 'n volledige beeld oor te dra (sek/beeld). Die uitleestempo vir standaard videokameras hang af van die parameters van die "frame grabber", asook van die kamera. Vir 'n standaard videobeeld (sien afdeling 2.2.2.3) word die uitleestyd bepaal deur:

$$R = \left(\frac{\text{sek}}{\text{Beeld}} \right) \times \left(\frac{\text{Lyne}}{\text{Beeld}} \right) \times \left(\frac{\text{Beeldelemente}}{\text{Lyne}} \right) \quad (11)$$

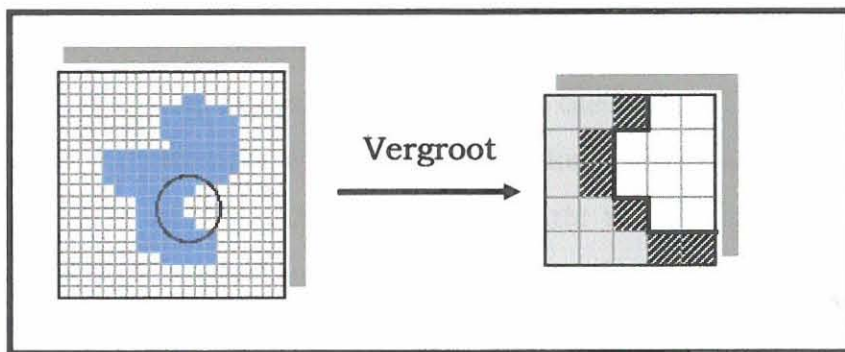
2.5 Kontoervoorstelling

Figuur 2.10 verwys na die simulasie van 'n beeld wat gedigitaliseer word en waarvan die rand verkry is. Tegnieke byvoorbeeld reekskode, kraakkode en rykode bestaan om 'n gebied of 'n voorwerp te vertoon, deur die beskrywing van sy kontoere [19, pp. 701-706][6, p. 356]. Die verskil tussen die kraakkode en die reekskode is die punt waarby die verbinding van beeldelemente van die voorwerp voorkom. Kraakkodering, soos die naam aandui, gebruik 'n punt op die grens tussen twee beeldelemente, eerder as die middelpunte van die beeldelemente vir verbindingsvektore. Die resultaat is dus 'n beter voorstelling van die ware kontoer van die voorwerp. Dit word voorgestel deur 'n vergroting van 'n gedeelte van Figuur 2.10 in Figuur 2.11.

In die studie is daar wel gebruik gemaak van kontoervoorstelling om die lengte van die forel te bepaal. Die tegniek wat gebruik is, word in afdeling 4.1.3.2 uiteengesit.



Figuur 2.10 Voorstelling van gekleurde gebied wat gedigitaliseer is en waarvan die randbeeld verkry is



Figuur 2.11 Die kontoerbeeldelemente soos wat dit gebruik word in die reekskode is diagonaal ingekleur.

HOOFSTUK 3

HISTOGRAMME EN DRUMPELBEPALING

In hierdie hoofstuk word daar verwys na bewerkinge wat die grondslag van digitale beeldverwerking daarstel. Hierdie bewerkinge kan in drie kategorieë opgedeel word, naamlik:

- a) Histogramme;
- b) Wiskundige berekeninge;
- c) Berekeninge met afgeleides.

3.1 Histogramme

'n Grysvlakhistogram is 'n funksie wat die hoeveelheid beeldelemente vir spesifieke grysvlakke van 'n betrokke beeld aandui. Een van die eenvoudigste en mees bruikbare elemente van digitale beeldverwerking is die grysvlakhistogram. Die histogram van enige beeld bevat nuttige inligting soos byvoorbeeld die hoeveelheid beeldelemente by 'n spesifieke grysvlak [2, pp. 71-79].

3.1.1 Kontrasstrekking

Kontras verwys na die amplitude waarmee grysvlakke verskil binne 'n beeld. Lae kontrasbeelde kan veroorsaak word deur swak beligting, 'n foutiewe opstelling van die fotograferingsapparaat of die lensopening kan verkeerdelik opgestel wees. Kontrasstrekking se hoofdoel is om die grysvlakke se dinamiese ruimte te verhoog ten opsigte van die beeld wat verwerk word.

Die negatiewe voorstelling van 'n digitale beeld kan verkry word deur die transformasiefunksie:

$$s = T(r) \tag{12}$$

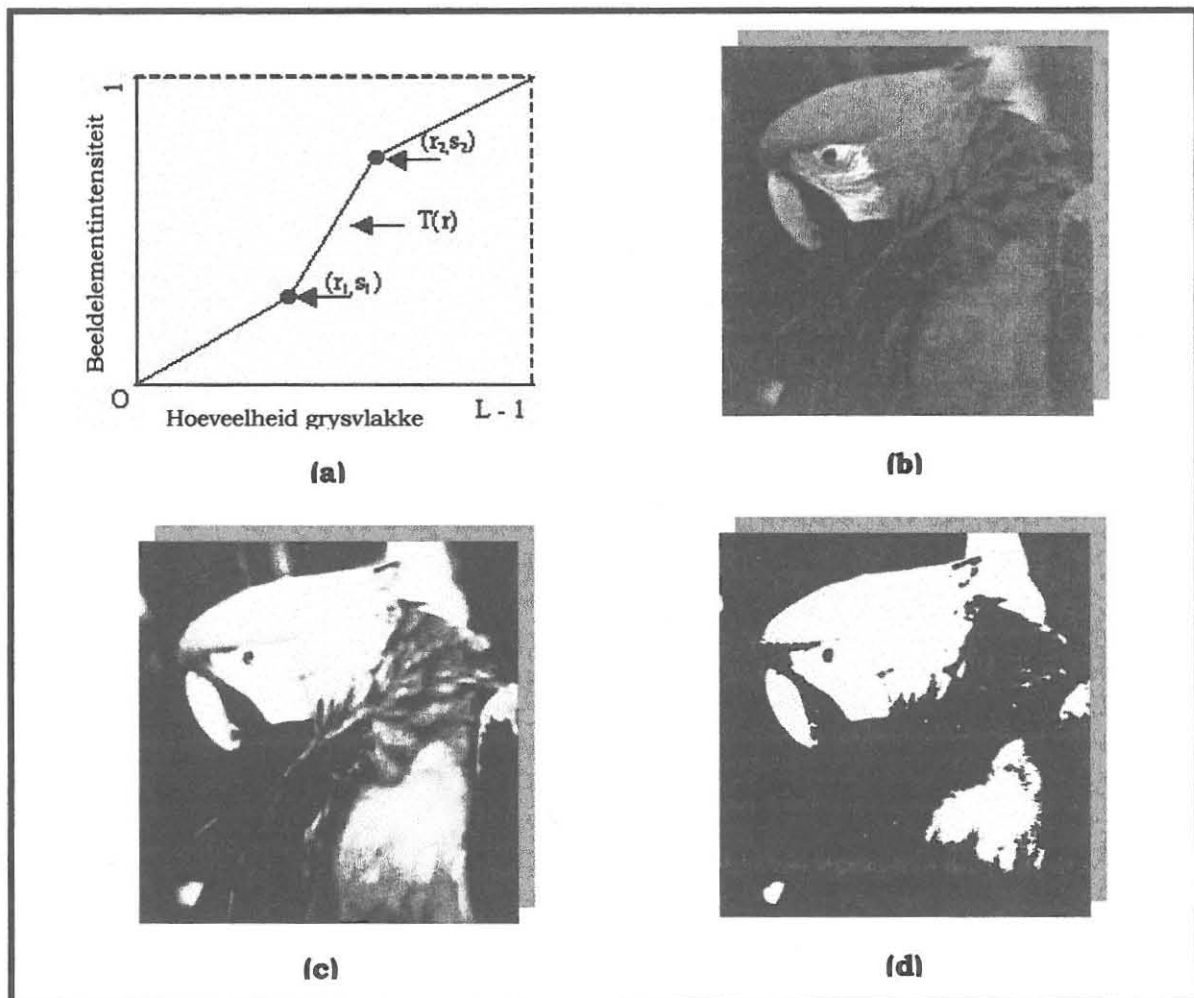
waar: s = intensiteit van die beeldelemente na verwerking.

T = grysvlaktransformasie funksie.

r = intensiteit van die beeldelemente voor verwerking.

Met verwysing na Figuur 3.1(a) beheer die posisie van (r_1, s_1) en (r_2, s_2) die vorm van die transformasiefunksie. Gestel $r_1 = s_1$ en $r_2 = s_2$, dan sal die transformasie lineêr wees wat geen veranderinge in die grysvalkke sal veroorsaak nie. As $r_1 = r_2$, $s_1 = 0$ en $s_2 = L - 1$, word die transformasie 'n drumpelfunksie wat 'n binêre beeld laat ontwikkel. 'n Binêre beeld is 'n beeld wat slegs uit swart en wit beeldelemente bestaan.

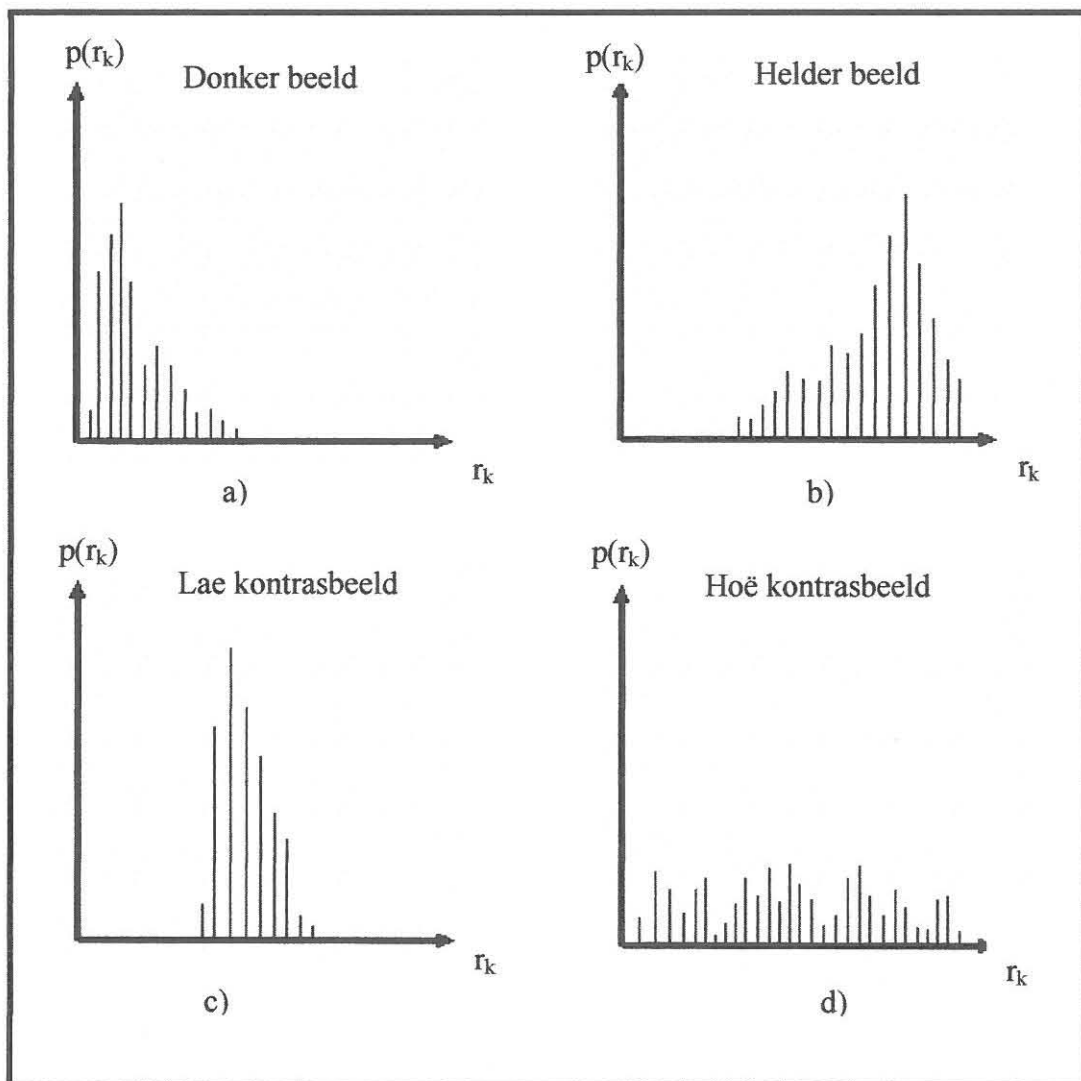
Figuur 3.1(b) stel 'n 8-bisbeeld met lae kontras voor. Figuur 3.1(c) dui op die effek van kontrasstrekking en Figuur 3.1(d) dui die drumpel van die beeld aan [7, p. 141][6, pp. 167-168]. In Figuur 3.1(a) is L die hoeveelheid grysvalkke, r is die intensiteit van die beeldelemente voor verwerking en s die intensiteit na verwerking.



Figuur 3.1 Kontrasstrekking word in (a) verduidelik deur 'n transformasiefunksie. Beelde met verskillende kontraste is sigbaar in (b), (c) en (d) [6, p. 168]

As die histogram in Figuur 3.2(a) ondersoek word, is dit duidelik dat die grys vlakke meer aan die donker kant van die grys skaal is. Die teenoorgestelde is waar vir die histogram in Figuur 3.2(b). Dit beteken dus dat die beeld in die eerste figuur 'n baie donker beeld sal wees.

In Figuur 3.2(c) is die vorm van die histogram baie smal en dus word 'n beeld met 'n lae kontras ten toon gestel. Figuur 3.2(d) dui op 'n histogram met 'n baie betekenisvolle vorm en is dus 'n beeld met 'n hoë kontras [6, p. 174]. 'n Beeld met 'n histogram wat 'n hoë kontras het se kwaliteit is van hoë gehalte en kan baie goed vir beeldanalise gebruik word. In die studie is daar gepoog om beelde te verkry met soortgelyke histogramme soos in Figuur 3.2(d).



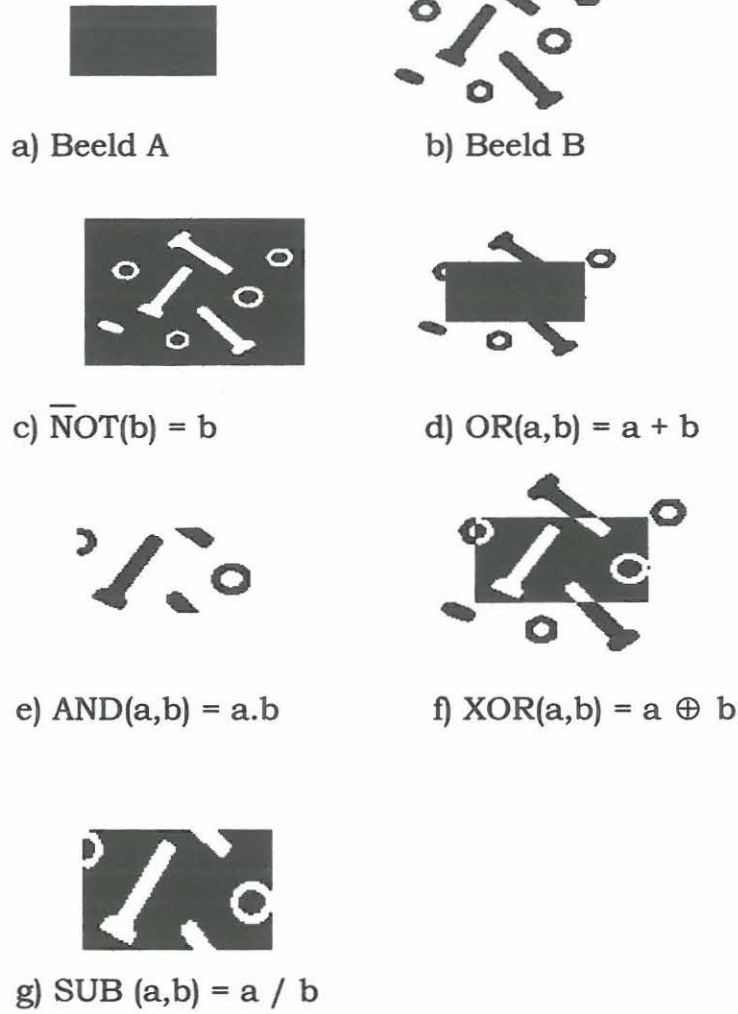
Figuur 3.2 Voorstelling van vier verskillende histogramme
[6, p. 174]

3.2 Berekeninge

In binêre bewerkinge is daar twee helderheidswaardes: “0” en “1”. Wiskundig word daar verwys na 2^B (hoeveelheid bisse) helderheidswaardes of vlakke, maar met beeldverwerking kan daar baie meer vlakke ontstaan. As gevolg van dié stelling maak sagtewarepakette voorsiening vir 16- en 32-bisvoorstellings van beeldelement-helderheidsvlakke om enige berekeningsprobleme te voorkom.

3.2.1 Binêre berekeninge

Morfologie word algemeen voorgestel as `n afdeling van biologie wat handel oor die vorm en struktuur van diere en plante. Bewerkinge gebaseer op binêre (Boolean) rekenkunde, vorm die basis van wiskundige morfologie. Wiskundige morfologie is `n hulpbron vir die ontleding van bruikbare beeldkomponente wat vorms soos buitelyne en grense insluit [6, p. 518][21, pp. 407-411] Simulasies van binêre bewerkinge word in Figuur 3.3 verduidelik, waar die binêre waarde “1” as swart voorgestel word en die binêre waarde “0” as wit.



Figuur 3.3 Voorstelling van binêre bewerkings op beelde

3.2.2 Ander bewerkinge

Grysvlakbeeldverwerking sluit onder andere die bewerkinge in Tabel 3.1 in:

Tabel 3.1 Bewerkinge wat vir beeldverwerking gebruik word

Bewerking:	Definisie:	Data tipe:
ADD	$c = a + b$	heelgetal
SUB	$c = a - b$	heelgetal
MUL	$c = a * b$	heelgetal of wisselpunt
DIV	$c = \frac{a}{b}$	wisselpunt
LOG	$c = \log(a)$	wisselpunt
EXP	$c = \exp(a)$	wisselpunt
SQRT	$c = \text{sqrt}(a)$	wisselpunt
TRIG	$c = \sin + \cos + \tan(a)$	wisselpunt
INVERT	$c = (2^B - 1) - a$	heelgetal

3.2.3 Berekeninge van 'n afgeleide

Verskeie bewerkinge wat uit een of meer afgeleides bestaan, kan op beelde toegepas word. Omdat 'n beeld 'n funksie van twee of meer veranderlikes is, is dit nodig om die rigting van die afgeleide te bepaal.

In 'n tweedimensionele geval bestaan daar 'n horisontale rigting, vertikale rigting of 'n kombinasie van dié twee rigtings, wat as 'n willekeurige rigting bekend staan.

3.3 Segmentering

Gedurende die verwerking van 'n voorwerp binne 'n beeld is dit belangrik om te onderskei tussen die voorwerp van belang en 'die res'. Daar word ook na laasgenoemde verwys as die agtergrond [12, pp. 210-214].

Segmenteringstegnieke word gebruik om die voorwerp van belang te vind, met ander woorde om 'n voorgrond van 'n agtergrond te onderskei.

Twee algemene en mees bruikbare segmenteringstegnieke is drumpelbepaling en randbepaling. Dit wil voorkom of daar nie 'n universiële toepaslike of bruikbare segmenteringstegniek bestaan wat op alle beelde toegepas kan word nie en dat segmenteringstegnieke nie perfek is nie [25, p. 103].

3.3.1 Drumpelbepaling

Drumpelbepaling is gebaseer op 'n konsep waar twee moontlike probleemstellings bestaan om die rand van 'n voorwerp te bepaal. Indien die rand van 'n ligte beeld met 'n donker agtergrond verkry moet word, kan die stelling, in Kodelys 3.1 gebruik word. 'n Helderheidsdrumpelparameter (q) word gekies en op 'n beeld $a[m,n]$ toegepas.

$$\begin{array}{ll} \text{Indien } a[m,n] > q & a[m,n] = \text{voorwerp} = 1 \\ \text{anders} & a[m,n] = \text{agtergrond} = 0 \end{array}$$

met q as die helderheidsdrumpelwaarde.

Kodelys 3.1 Die randbepaling van 'n ligte beeld

Sou die rand van 'n donker beeld met 'n ligte agtergrond verkry moet word, kan die volgende stelling gebruik word.

$$\begin{array}{ll} \text{Indien } a[m,n] < q & a[m,n] = \text{voorwerp} = 1 \\ \text{anders} & a[m,n] = \text{agtergrond} = 0 \end{array}$$

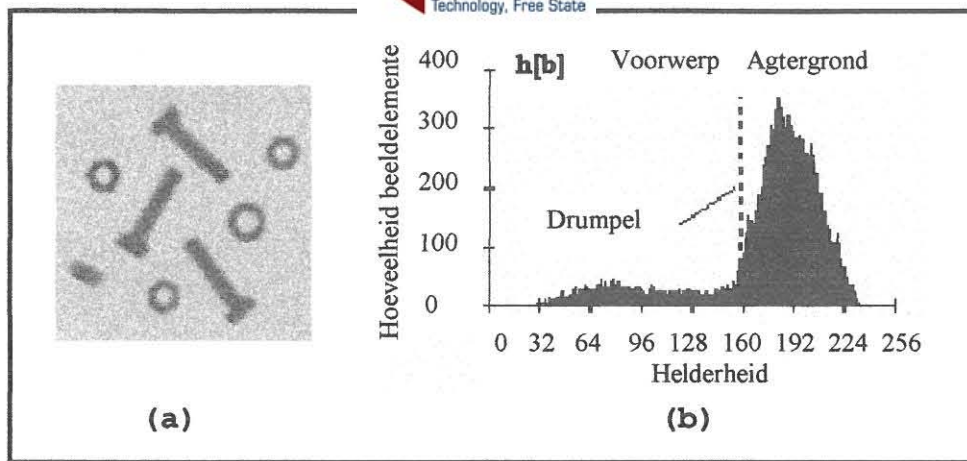
met q as die helderheidsdrumpelwaarde.

Kodelys 3.2 Die randbepaling van 'n donker beeld

Die uitset kan dus as 'n voorwerp of 'n agtergrond deur 'n Boolese veranderlike '1' of '0' voorgestel word. As dit by die bepaling van die drumpel kom, ontstaan die vraag: Hoe word die drumpel (q) gekies? Daar is nie 'n algemene prosedure vir die bepaling van 'n drumpel wat op 'n verskeidenheid van beelde toegepas kan word nie, maar daar bestaan egter verskeie metodes. Die volgende is voorbeelde van drumpelbepalings-tegnieke:

a) Vasgestelde drumpel: Die eerste metode is om 'n drumpel te kies wat onafhanklik van die beeld se data is. As dit bekend is dat 'n beeld met hoë kontras gebruik word, met ander woorde waar die voorwerp baie donker is en die agtergrond baie lig, kan 'n konstante drumpel van 128 (gemiddelde waarde) op 'n skaal van 0 tot 255 redelik akkuraat wees. Met akkuraat word bedoel dat die hoeveelheid vals-geklassifiseerde beeldelemente tot die minimum beperk sal word [25, p. 103].

b) Histogram van 'n afgeleide drumpel: In meeste gevalle word die drumpel gekies vanaf 'n helderheidshistogram. Die histogram is van die gebied of beeld wat gesegmenteer moet word. 'n Helderheidshistogram van 'n beeld is die helderheid van beeldelemente, teenoor die hoeveelheid beeldelemente by 'n sekere helderheid (grysvlakwaarde) en dit kan uitgestippel word. 'n Beeld met sy toepaslike helderheidshistogram word in Figuur 3.4 getoon. Dit is met behulp van 'n sagtewareprogram "Scion image" [22] verkry.



Figuur 3.4 (a) 'n Beeld waarvan die drumpelwaarde bepaal gaan word

(b) Helderheidshistogram van die beeld in (a) waarmee die drumpelwaarde van die beeld bepaal word

Beeldelemente wat kleiner is as die drumpel ($a(m,n) < q$), kan bestempel word as die voorwerpbeeldelemente. Dit wat bo die drumpel is kan geklassifiseer word as agtergrondbeeldelemente [2, pp. 452-454]. Dus word onderskei tussen die voorwerp van belang en die agtergrond (sien afdeling 4.1.3.1).

3.4 Randbepaling

Wanneer randbepaling op 'n beeld toegepas word, is die ideaal om 'n 'lynsketsvoorstelling' van die oorspronklike beeld te verkry met die belangrikste rante en lyne. Een metode rakende randbepaling is die stensilm metode.

3.4.1 Stensilm metode

Elke beeldelement en sy omliggende omgewing word saamgevoeg in 'n vierkantige stensil wat bestaan uit sekere waardes. Die resultate hiervan dui aan of daar wel 'n rand by die spesifieke punt aanwesig is, al dan nie. Die twee mees populêre randbepalingsmetodes wat op meer as een stensil gebaseer is, staan bekend as die: "Sobel Operator" en die "Roberts Cross". Die patrone wat gebruik word vir dié twee bewerkinge word in Tabel 3.2 uitgebeeld.

Tabel 3.2 Stensils wat gebruik word vir die Sobel- en Roberts-randbepalingstegnieke

-1	0
0	-1

0	-1
-1	0

“Roberts Cross”

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

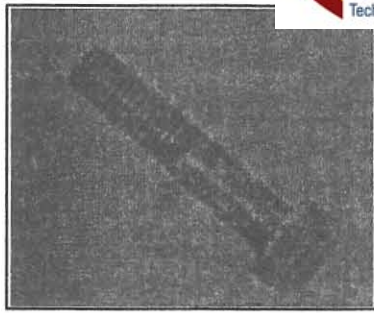
-1	-2	-1
0	0	0
1	2	1

“Sobel Operator”

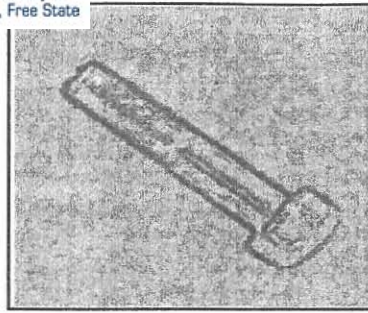
Elke bewerking bestaan uit twee stensils. 'n Stensil word oor 'n beeld geskuif en die graad van ooreenstemming word by elke punt bepaal. Die waardes verkry vanaf elke stensil word dan saamgevoeg om die algehele randwaarde by die spesifieke punt van belang te verkry. Dit word gewoonlik verkry deur die som van die absolute waardes.

Die skrywer, Rosandich [20, pp. 134-136] het verskeie metodes ondersoek en gevind dat die "Roberts Cross" nie so akkuraat op 'n verskeidenheid van beelde is, soos die "Sobel Operator" nie. Die probleem wat ondervind is met die "Roberts Cross", is dat dit die vermoë het om ruis of ongewenste beeldelemente op die beeld te verhoog [20, pp. 193-196], wat tot gevolg het dat baie vals rante ontstaan wat nie ooreenstem met die voor die hand liggende rante nie. Daarteenoor kan die "Sobel Operator" baie meer suksesvol in randbepaling gebruik word.

'n Voorbeeld van die "Sobel Operator" toegepas op 'n ware beeld is sigbaar in Figuur 3.5(a) en is verkry met behulp van "Scion Image"[22].



(a)



(b)

Figuur 3.5 Voorbeeld van 'n grysvlakbeeld en sy randbeeld, nadat die “Sobel Operator”-randbepalingstegniek daarop toegepas is

EKSPERIMENTELE OPSTELLING EN SAGTEWARE

In hierdie hoofstuk word eksperimentele opstelling in totaliteit uiteengesit. Dit sluit die toerusting wat gebruik is in, asook hoe die substelsels met mekaar geïntegreer word. Die werking en beskrywing van die sagteware word ook verduidelik soos uiteengesit in Figuur 4.1.

Tydens die studie is ook soortgelyke probleme ondersoek, soos byvoorbeeld om die lengte en deursnee van plantwortels vas te stel. Dit bepaal die plant se water- en nutriëntinname wat sy toekomstige groeivermoë beïnvloed. Daar is gevind dat dit ook met groot sukses gebruik word [11, pp. 267-273]. Die meeste beeldverwerkingstelsels maak gebruik van 'n videokamera om 'n beeld te verkry. Die resolusie en grootte rakende die geskandeerde area van kameragebaseerde beeldverwerking, word deur die volgende aspekte beïnvloed:

- Die grootte van die beeldinname-opstelling.
- Die lensgrootte en afstand tussen die kamera en skanderingsoppervlakte.

Met die gebruik van ouer beeldverwerkingstelsels (gewoonlik met 'n 512 x 512 beeldinname-opstelling) is resolusies van 0.3 tot 0.7 mm verkry. Ewing en Kaspar [4], wat 'n verbeterde, hoër resolusie digitale videokamera (2800 x 1728-beeldinname-opstelling) gebruik het, beweer dat worteldiktes van 0.16 mm waargeneem [4] is teen 'n skanderingsoppervlak van 0.317 by 0.235 m (0.074 m²). In vergelyking met digitale videokameras het 'n aftaster die voordeel van beter resolusie oor 'n groter area. Die aftaster gebruik deur Pan en Bolton [16] het 'n resolusie van 23.6-beeldelemente mm⁻¹(0.042 by 0.042 mm-beeldelementgrootte): dus was dit vir hulle moontlik om 'n menslike haar te meet met 'n diameter van 0.13 mm wat op die skandeerderoppervlak geplaas is. Die aftaster wat hulle gebruik het, het 'n skandeerarea van 0.216 by 0.297 m (0.064 m²) wat ongeveer 14% kleiner is as die aftaster wat gebruik word deur Dowdy en ander [4][15].

Prosedures vir die verkryging van digitale beelde van forelle is op grysvlakbeelde, sowel as kleurbeelde van toepassing. Die gebruiker moet die drumpelwaarde van elke beeld afsonderlik verstel om die voorwerp van die agtergrond te onderskei en sodoende 'n beter swart-en-wit digitale randbeeld te verkry. Aangesien die bepaling van 'n beeld se drumpel 'n lang en uitgerekte proses is, is dit baie moeilik om twee keer na mekaar vanaf 'n beeld, dieselfde drumpel te verkry [5]. Dit is dus voor die hand liggend dat verskillende gebruikers verskillende drumpelwaardes sal verkry.

Volgens Ewing en Kaspar [4] en ander is bewys dat die meting van "soil pores" vanaf 'n digitale beeld uiters sensitief is vir die drumpelbepaling vanaf die oorspronklike beeld. In die studie wat hulle gedoen het, is ook bewys dat een persoon twee verskillende metodes/prosedures van drumpelbepaling gebruik het en dat die metingsresultate grootliks gevarieer het, ten spyte daarvan dat daar geen verskille op die beelde sigbaar was nie.

Nadat die beeld gedigitaliseer en die drumpel bepaal is, is daar twee moontlike berekeninge wat gebruik kan word om die lengte van 'n voorwerp op 'n digitale beeld te bepaal. Eerstens die lyn-kruisingmetode. Harris en Campbell [8] het deur middel van 'n gerekenariseerde weergawe van die lynkruisingmetode ook die lengtes van voorwerpe (wortels) vanaf digitale beelde bepaal. Die tweede opsie bestaan uit die kettingmetode, insluitend 'n paar variasies daarvan. Hierdie metode tel die lengtes van elke beeldelement binne 'n ketting bymekaar, met voorsiening vir beeldelemente wat diagonaal mag voorkom [23][4].

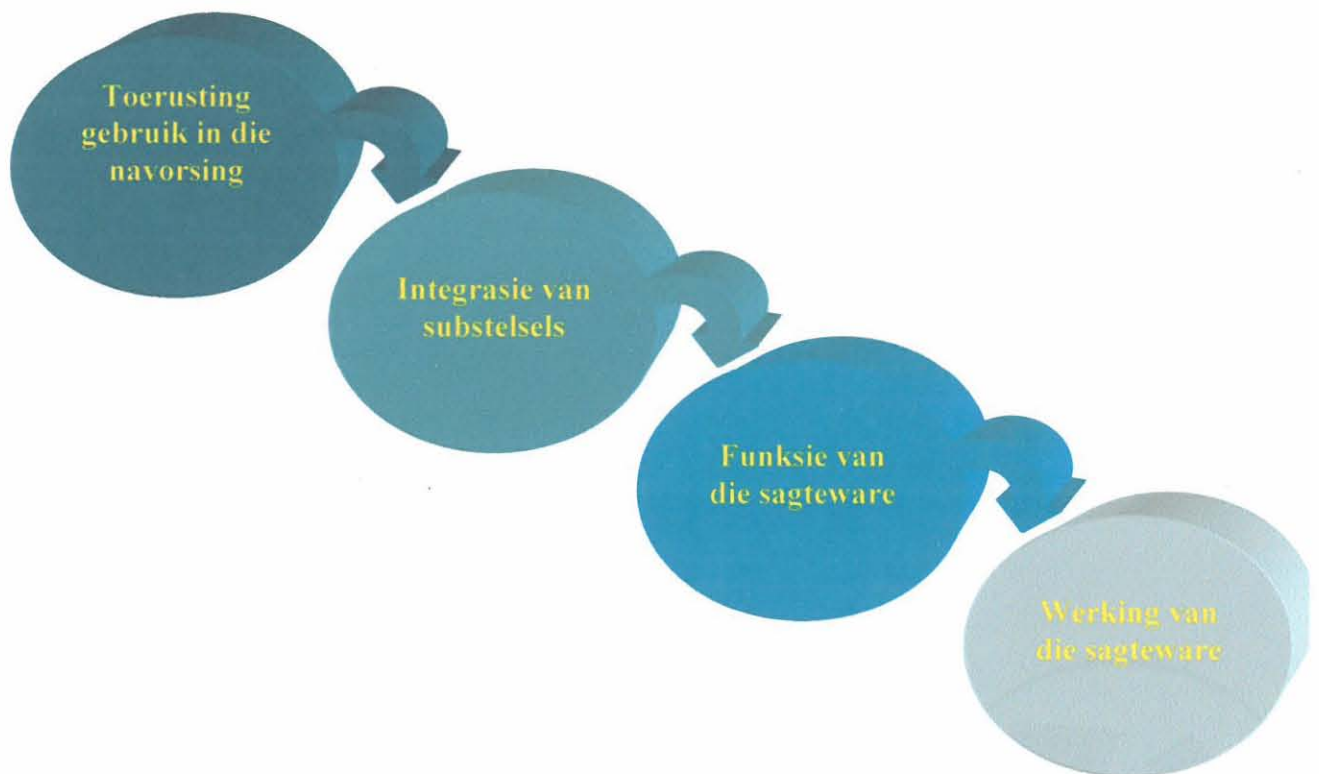
Beide die lynkruising- en die kettingmetode het groot nadele. Hierdie twee metodes gaan van die veronderstelling uit dat voorwerpe in 'n beeld perfek gerangskik is. Sodra die aanname geïgnoreer word, kan dit 'n ernstige oor- of onderskatting tot gevolg hê [4][27].

Na oorweging van die voor- en nadele, is 'n nuwe program geskryf: Troutedge. Troutedge is so ontwikkel dat kompensasie vir beeldelemente wat diagonaal verbind is, onnodig is soos in die geval van Travis et al. [23]

en Ewing en Kaspar [4] nie. Die toegepaste formule vir lengtebepaling kompenseer outomaties daarvoor, aangesien bewerkinge vanaf die middel van beeldelement X tot die middel van beeldelement Y uitgevoer word. Die lengte word bepaal vanaf 'n randbeeld van 'n belangrike voorwerp en die massa word dan uit Tabel 4.1 verkry.

4.1 Toerusting en integrasie van substelsels

Die volgende diagram stel die eksperimentele opstelling voor, ten einde bruikbare en akkurate resultate vir verdere beeldverwerking te verkry.



Figuur 4.1 Diagram vir die eksperimentele opstelling soos dit gebruik is in die navorsingstudie

4.1.1 Toerusting gebruik in die navorsing

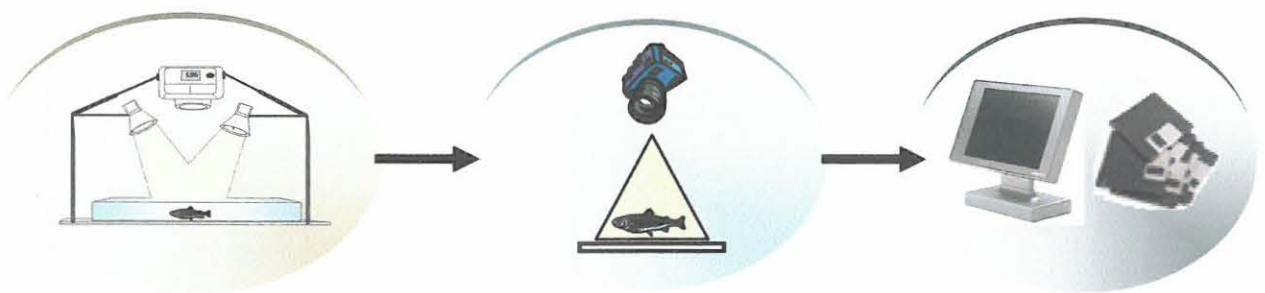
Een van die mees belangrike hardeware artikels wat gebruik is, is 'n kamera. 'n SONY (DSC-P51) is gebruik vir die fotografering van die nodige beelde. Die kamera se spesifikasies is die volgende: 2 miljoen beeldelemente, 6 x digitale vergroting en 32 x optiese vergroting.

’n Staalstruktuur is opgestel waarin die kamera geplaas is, om sodoende op konstante afstande foto’s vanaf die voorwerp te kan neem. Die struktuur kan die kamera in ’n vertikale rigting laat verstel, om verder of nader aan die grondoppervlak of voorwerp te beweeg. Saam met die kamera is daar aan die struktuur twee tungstram gloeilampe van 100 watt elk (220V) geheg. Die twee gloeilampe wys afwaarts op ’n wit 1500 x 800 x 500 mm-porseleinbad met ongeveer 160 mm water daarin. Verskeie lewendige visse (forelle) is afsonderlik gebruik om in die bad met water te swem, sodat foto’s geneem kon word.

’n Pentium II, 350 MHz-prosesseerder, met 20 GHz-hardeskyf en ’n 64 MHz videokaart is gebruik om die nodige beeldverwerking te behartig en inligting te stoor. Al die beeldverwerking wat gedoen is, is op ’n 14 duim-monitor vertoon.

4.1.2 Die integrasie van substelsels

Die integrasie van substelsels behels die kamera-opstelling, die fotografering van beelde en die oordrag van die gefotografeerde beelde in digitale formaat na die rekenaar vir analise. Dit word grafies voorgestel deur Figuur 4.2 .

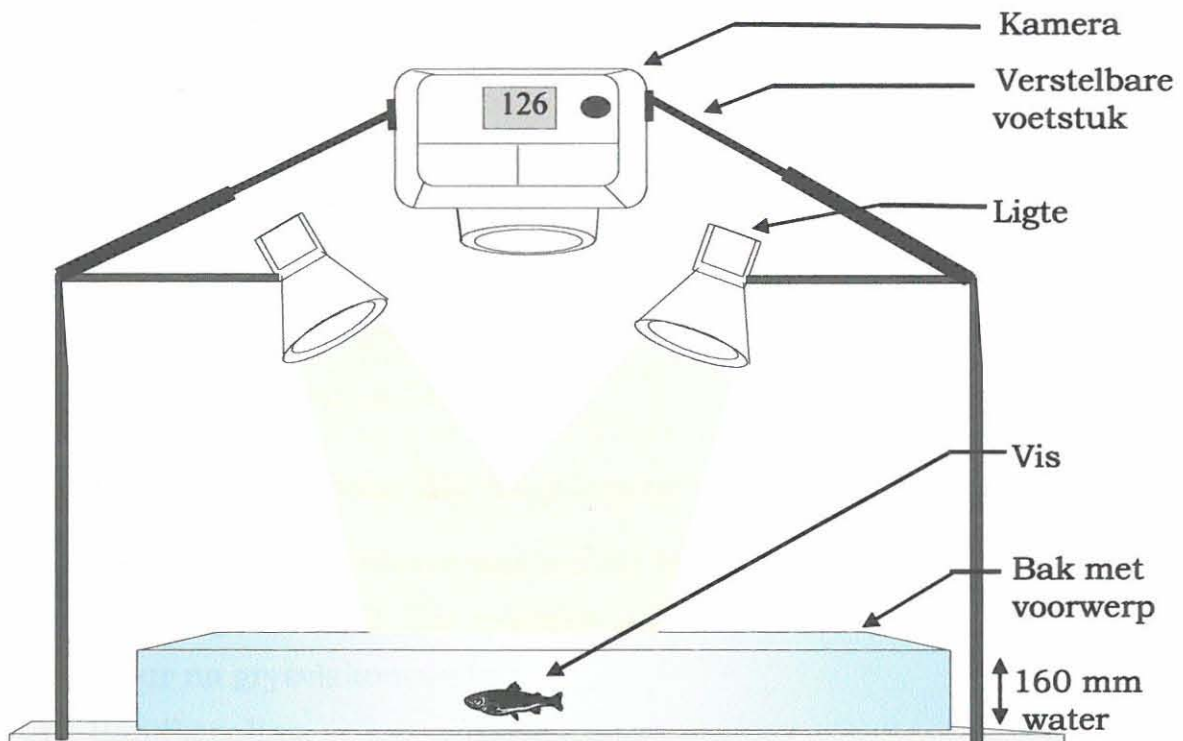


Figuur 4.2 Voorstelling van integrasie van substelsels

4.1.2.1 Kamera-opstelling

Die volgende belangrike punte is in gedagte gehou tydens die eksperimentele opstelling. Dit word grafies in Figuur 4.3 voorgestel.

- Die agtergrond moet 'n enkelkleur wees, sodat 'n duidelike onderskeid tussen die voorwerp van belang en die agtergrond getref kan word. Tydens die eksperimentele opstelling is 'n wit agtergrond gebruik.
- Die stelsel moet van goeie beligting voorsien wees, wat so opgestel is dat dit geen invloed op die kamera se lens sal hê nie, asook geen skaduwees sal veroorsaak nie. Enige eksterne lig moet beheer of gereguleer word om skaduwees tot die minimum te beperk.
- Die afstand tussen die kamera en agtergrond moet konstant gehou vir kalibreringsdoeleindes. 'n Aantal foto's word op 'n sekere hoogte geneem en die stelsel word daarvolgens gekalibreer. Vir eksperimentele doeleindes is vier voorwerplengtes, 50 mm, 137 mm, 175 mm en 435 mm gebruik waarvan die lengtes afsonderlik deur Troutedge bepaal is. Die resultate hiervan word in Hoofstuk 5 grafies voorgestel.



Figuur 4.3 Opstelling van die eksperimentele toestel

’n Alternatiewe metode kan gebruik word om die kalibreringsproses te vervang. ’n Voorbeeld hiervan is lasermeting, waar die afstand outomaties deur ’n laser bepaal word en die lengte in verdere berekeninge toegepas word. As gevolg van onvoldoende fondse, kon die lasermetode nie geïmplimenter word nie.

4.1.2.2 Fotografering van beelde

Die beelde is nie outomaties gefotografeer nie. Elke foto is met die hand geneem sodra die vis onder die lens verbybeweeg het. Die visse kan ook verfilm word met behulp van ’n digitale videokamera, waarna die film geredigeer kan word om die korrekte beeld te verkry.

’n Alternatiewe metode is om sensors op strategiese plekke in die bad te monteer. Dit kan ook in ’n volgende studie ondersoek (nagevors) word. Hierdie sensors sal dan die digitale kamera of digitale videokamera aktiveer, wanneer die beweging van die vis waargeneem word.

4.1.2.3 Oordrag van die gefotografeerde beelde na ’n rekenaar

Die digitale foto’s wat geneem is, is op 1.44 MB-diskette gestoor in bisbeeld (.bmp)-formaat en oorgedra na ’n rekenaar waar die sagtewareprogram, Troutedge, gebruik is vir die verdere verwerking van die beelde. As ’n alternatief kan die kamera direk aan die rekenaar gekoppel word deur ’n “USB”-poort.

4.1.3 Funksie van die sagteware

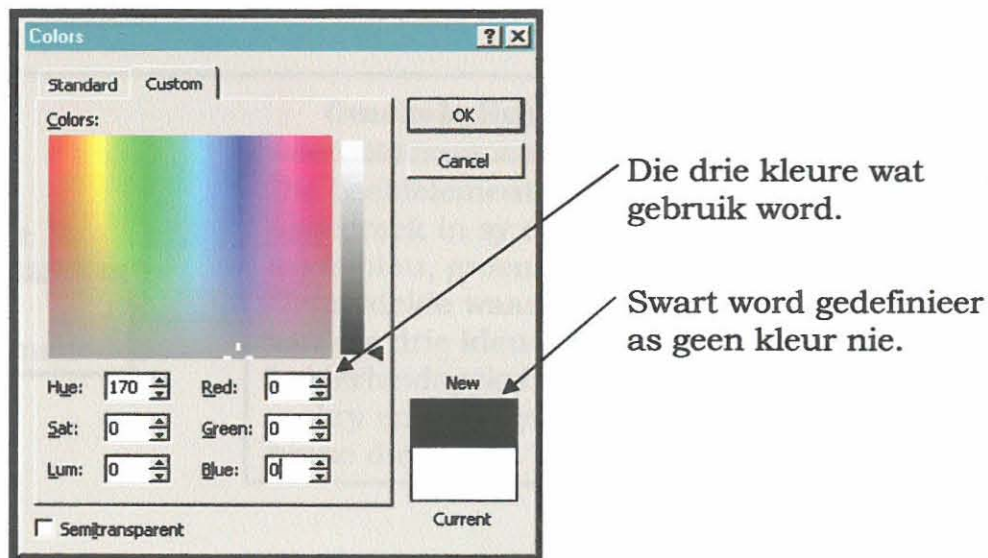
Die doel van die sagteware was om die lengte van ’n voorwerp te bepaal vanaf ’n digitale beeld. Die volgende vyf prosedures is gevolg, naamlik:

- a) Kleur na grysvlakomskakeling.
- b) Randbepaling.
- c) Die bepaling van strategiese koördinate om verdere bewerkinge rakende lengtebepaling te vergemaklik.
- d) ’n Metode vir die omskakeling van lengte na massa is ook ontwikkel.

4.1.3.1 Kleur na grysvlakomskakeling

Die omskakeling van kleur na grysvlakke, vergemaklik die proses om die nodige inligting vanaf grysvlakhistogramme te verkry (sien Hoofstuk 3), aangesien berekeninge met kleurbeelde `n groot hoeveelheid geheue in beslag neem, in teenstelling met grysvlakbeelde. Berekeninge word vinniger gedoen deur grysvlakbeelde te gebruik.

Indien die werking van `n rekenaarmonitor ondersoek word, sal daar gevind word dat slegs drie kleure teenwoordig is. Hierdie kleure is rooi, groen en blou. Elke kleur word ingedeel in `n 'spektrum' wat strek van 0 tot 255. Ten einde die verduideliking te vergemaklik, is die volgende simulaties uit "Microsoft Word" verkry. Indien al drie die kleure `n helderheidswaarde van 0 het, beteken dit daar geen kleur gedefinieer is nie. In so `n geval sal die kleur wat verkry word swart wees, soos sigbaar in Figuur 4.4.



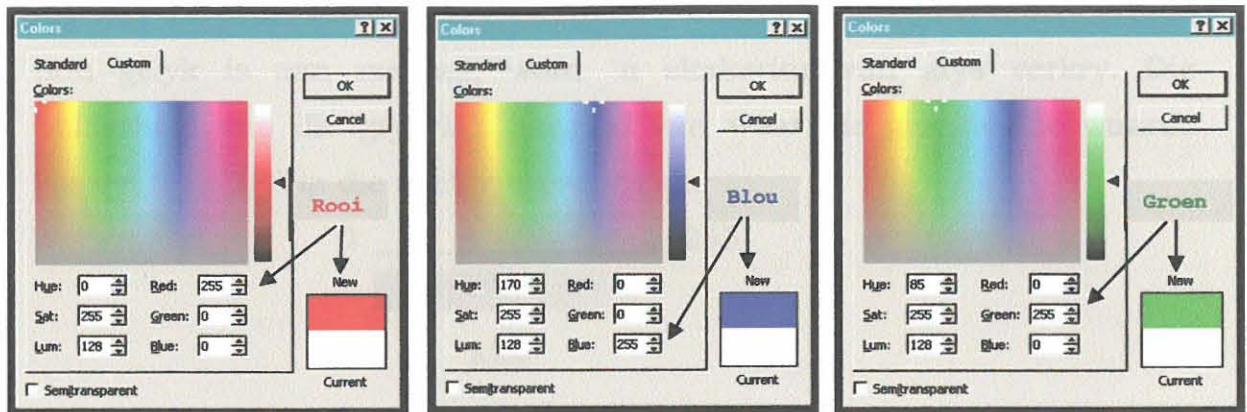
Figuur 4.4 Drie kleure wat gebruik word vir beeldverwerking

As elke kleur afsonderlik geneem word, sal 256 helderheidsintervalle vir elke betrokke kleur verkry word.

Rooi

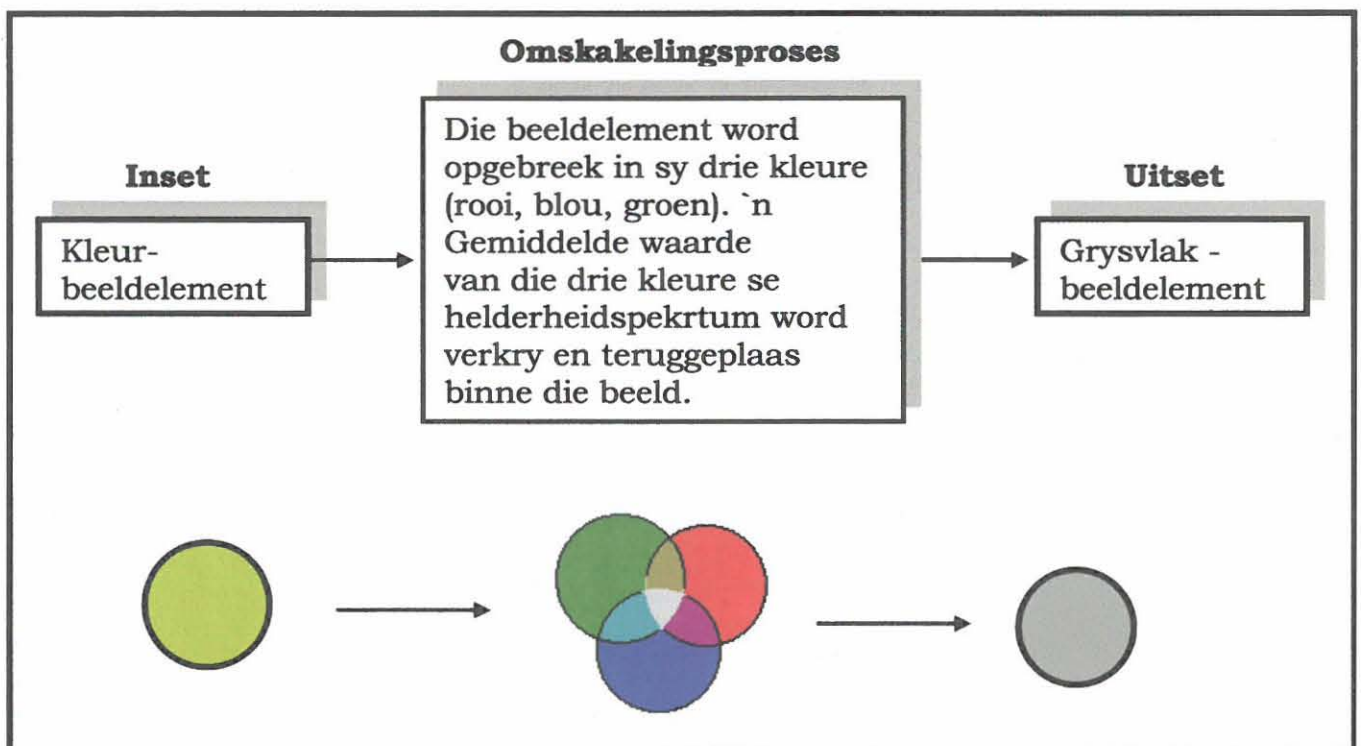
Blou

Groen



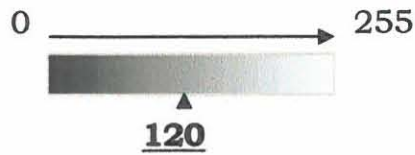
Figuur 4.5 Drie kleure wat gebruik word vir die berekening van grysvlakbeelde

Voordat 'n kleurbeeld omskakeling na 'n grysvlakbeeld gedoen kan word, moet die beeldelemente van die kleurbeeld se helderheid bekend wees. Dit word gedoen deur 'n funksie soortgelyk aan konvolusie (sien afdeling 2.3.1). Dit kan soos volg verduidelik word:



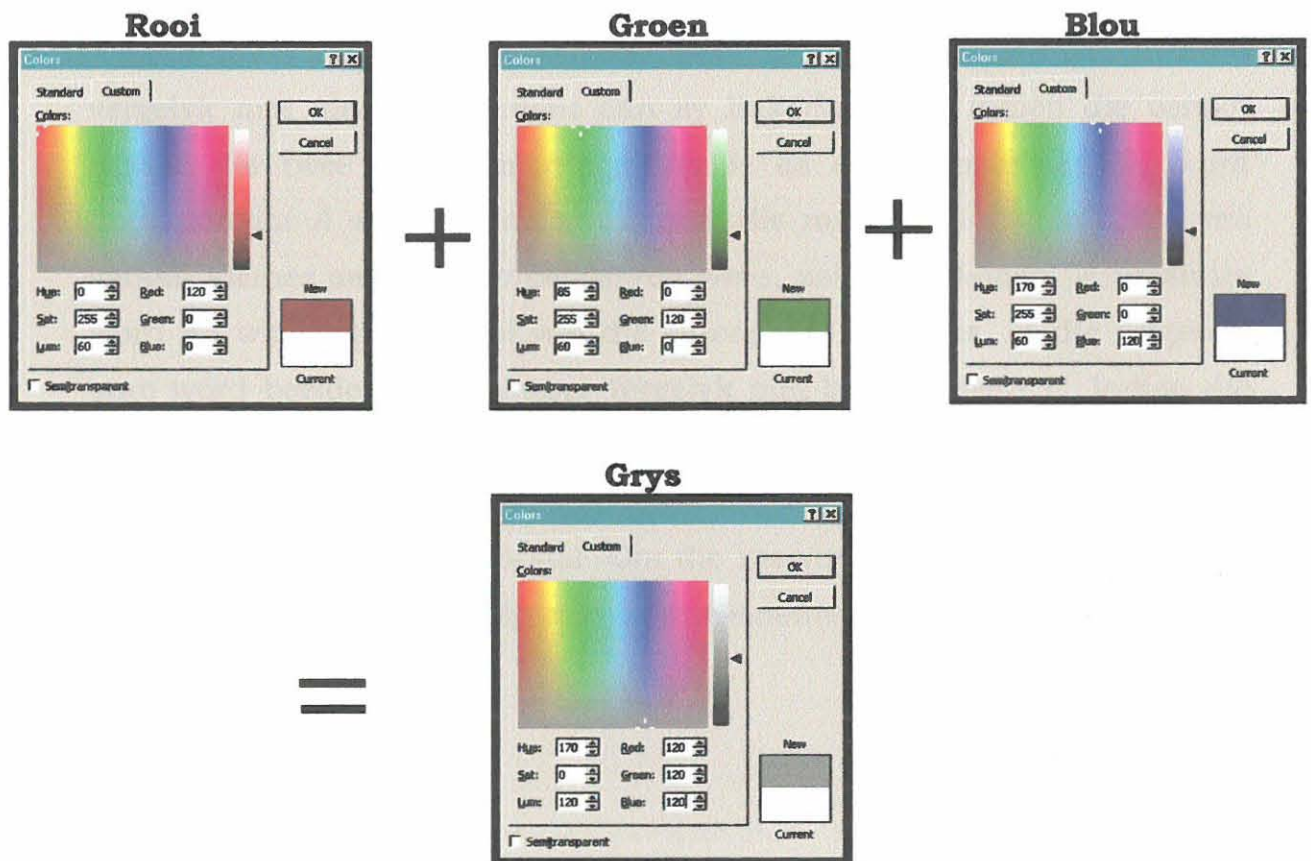
Figuur 4.6 'n Funksie soortgelyk aan konvolusie wat 'n kleurbeeldelement verander na 'n skakering van grys

As die helderheidsvlakke bekend is, word daar 'n gemiddelde waarde vir elk van die drie kleure verkry. Aangesien hierdie drie kleure se helderheid nou gelyk is aan mekaar, word 'n skakering van grys verkry. Die helderheid van die grysvlak hang af van waar die gemiddelde waarde tussen 0 en 255 in die spektrum voorkom.



Figuur 4.7 Helderheidspektrum wat strek van 0 tot 255

Indien die gemiddelde waarde van die drie kleure gelyk aan 120 is, dan sal die helderheid van die grysvlak soos volg voorgestel kan word:

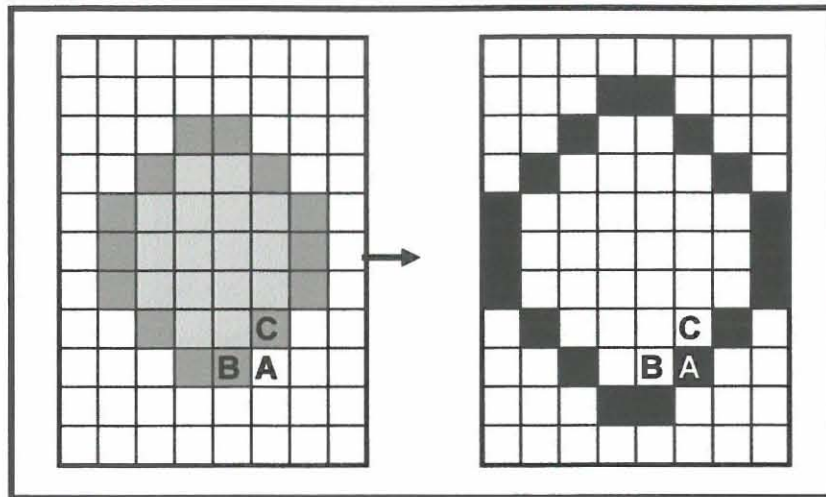


Figuur 4.8 Bepaling van 'n grysskakering met die samevoeging van die drie kleure (rooi, groen en blou)

4.1.3.2 Randbepaling

Soos in afdeling 2.2.2.1 verduidelik is, word 'n plaaslike beeldfunksie gebruik vir randbepaling. Die uitsetwaarde by 'n spesifieke koördinaat is afhanklik van die koördinate in die plaaslike omgewing van die spesifieke insetbeeldelement. Om die rand van 'n voorwerp te verkry, moet daar onderskeid getref word tussen die voorwerp van belang en die agtergrond. 'n Drumpelwaardespektrum strek van 0 tot 255. Hierdie drumpelwaarde kan dien as 'n aanduiding van die helderheid van die randbeeld wat verkry gaan word [13]. Die verkose drumpelwaarde kan ook gesien word as 'n afsnypunt waarmee die oorblywende beeldelemente vergelyk word (sien afdeling 2.5). Die randbepalingsmetode in Figuur 4.9, waar die beeldelemente met mekaar vergelyk word, kan soos volg beskryf word:

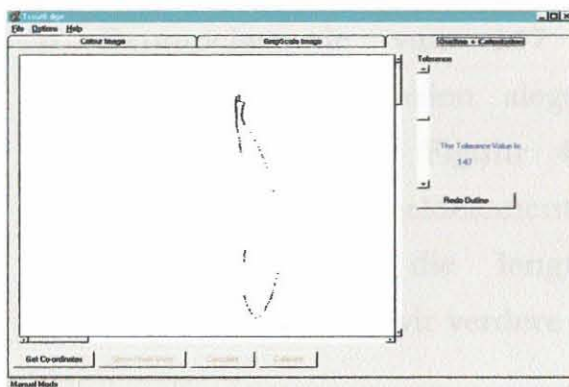
Gestel 'n drumpelwaarde van 80 word gekies. Beeldelement *A* word vergelyk met die beeldelement aan sy linkerkant, *B*. Indien die verskil tussen die twee beeldelemente groter is as die drumpelwaarde, word beeldelement *A* verander na swart, wat die rand gaan voorstel. Sou die verskil kleiner as die drumpelwaarde wees, sal beeldelement *A* verander word na wit, wat dan deel van die agtergrond sal wees. In die volgende stap word beeldelement *A* weer vergelyk met beeldelement *C*. Indien die verskil tussen *A* en *C* groter as die drumpelwaarde is, sal beeldelement *A* na swart verander word, wat weer die rand sal voorstel, andersins na wit, wat die agtergrond sal voorstel. Om die res van die randbeeld te verkry, word elk van die opeenvolgende beeldelemente op soortgelyke wyse met mekaar vergelyk.



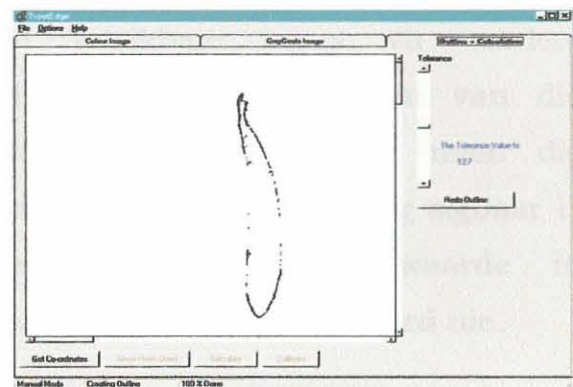
$$A = 255; B = 125; C = 125$$

Figuur 4.9 Randbeeld word verkry deur drie beeldelemente met mekaar vergelyk

'n Voorbeeld van een beeld met drie verskillende drumpelwaardes word in Figuur 4.10 (a-c) getoon.

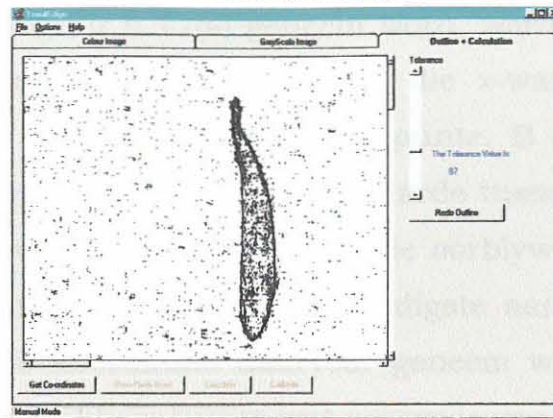


(a) Drumpelwaarde = 147



(b) Drumpelwaarde = 127

Figuur 4.10 Bepaling van 'n randbeeld met die toepassing van verskillende drumpelwaardes



(c) Drumpelwaarde = 87

Figuur 4.10 (vervolg) Bepaling van 'n randbeeld met die toepassing van verskillende drumpelwaardes

Hoe nader die drumpelwaarde aan 225 is, hoe ligter of nader aan wit sal die beeld wees. Indien die drumpelwaarde na 0 neig, word 'n donker beeld verkry met baie onbruikbare beeldelemente (ruis).

'n Drumpelwaarde van 147 sal bruikbaar wees vir verdere beeldverwerking, aangesien slegs die belangrikste buitelyne van die voorwerp sigbaar is. Figuur 4.10(b) kom foutloos voor, maar die teenwoordigheid van beeldelemente wat nie met die blote oog sigbaar is nie, beïnvloed wel die lengtebepaling. Die drumpelwaarde in Figuur 4.10(c) kan nie vir verdere beeldverwerking gebruik word nie.

4.1.3.3 Koördinaatbepaling

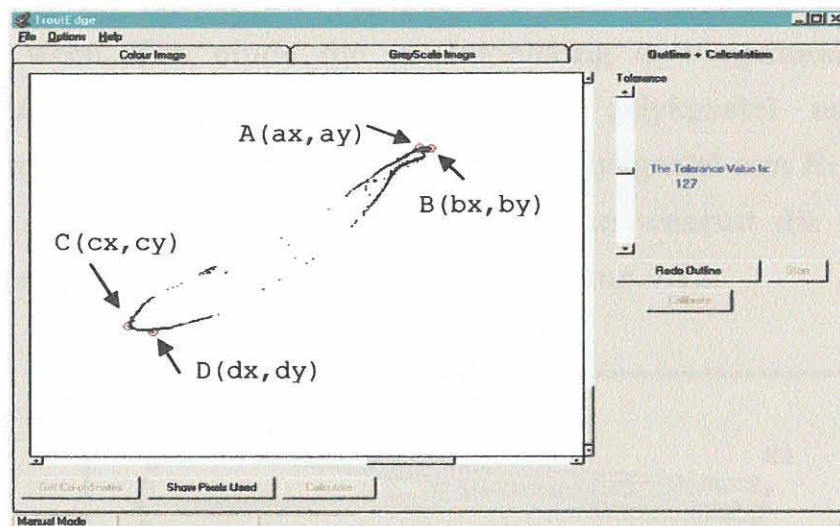
Nadat 'n randbeeld verkry is, word al die koördinate van die randbeeld in 'n databasis binne die sagtewareprogram ingelees. Koördinaatbepaling is ingesluit in die sagtewareprogram, om die tempo waarteen die lengte (resultaat) bereken word, te versnel. Sekere koördinate word oorgedra na 'n ander databasis om sodoende minder geheue in beslag te neem tydens die berekening van die forel se lengte.

Die punt van oorsprong (X_0, Y_0) word links bo in die hoek gevind. Met dit in gedagte, word die vier koördinate soos volg bepaal:

As die beeld in Figuur 4.11(a) geneem word, word die kleinste y -waarde van al die koördinate geneem (ongeveer die x -waarde). Die punt staan bekend as punt A. Die volgende twee punte, B en C hang af van die voorwerp se oriëntasie. Die kleinste y -waarde tussen die twee punte B en C (ongeveer die x -waarde) is punt B en die oorblywende punt sal C wees. Die bepaling van punt D is die eenvoudigste aangesien die grootste y -waarde uit al die koördinate daarvoor geneem word. Figure 4.11(a) en 4.11(b) toon die bruikbare punte wat verkry is van een voorwerp waarvan die oriëntasie verskil.

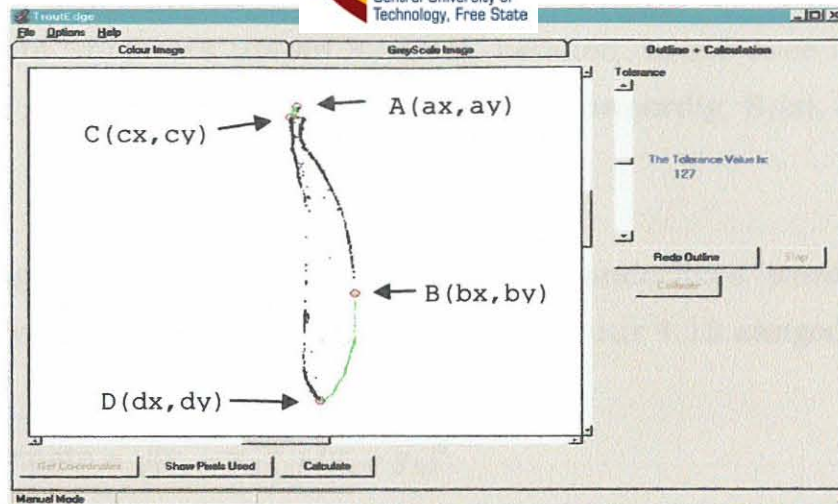
Nadat die vier verskillende punte $[(A_x, A_y), (B_x, B_y), (C_x, C_y)$ en $(D_x, D_y)]$ verkry is, word al die koördinate tussen punt A en punt C in 'n tweede databasis gestoor en die koördinate tussen punt B en punt D in 'n derde databasis.

Die koördinate wat vir lengtebepaling gebruik gaan word tussen A en C en B en D, is in twee afsonderlike databasisse en word met 'n groen kleur aangedui in Figuur 4.11(b).



(a)

Figuur 4.11 Vier koördinate wat gebruik word om die lengtebepaling te vergemaklik



(b)

Figuur 4.11 (vervolg) Vier koördinate wat gebruik word om die lengtebepaling te vergemaklik

4.1.3.4 Lengtebepaling

Verskeie faktore kan die bepaling van die lengte van 'n voorwerp beïnvloed. Hierdie faktore is geïdentifiseer en die sagteware is aangepas om daarvoor voorsiening te maak. Vir die bepaling van die lengte moet die rand eerste verkry word en tweedens moet die nodige koördinaatpunte bekend wees. Ten einde die verduideliking te vergemaklik, word die beeldelemente tussen punte A en C gelykgestel aan S_{1-7} . Die beeldelemente tussen punte B en D word gelykgestel aan E_{1-4} . S_{1-7} en E_{1-4} dui die hoeveelheid koördinate of punte aan waaruit die lengte bepaal gaan word. Die voorstelling is sigbaar in Figuur 4.12.



Figuur 4.12 Voorstelling van moontlike koördinate tussen die vier hoofpunte waarvandaan die lengtes bepaal word

Die lengte word eers tussen S_1 en E_1 bereken. In die twee beeldelemente (S_1 en E_1) is daar nou vier veranderlikes teenwoordig $S_1(x)$, $S_1(y)$ en $E_1(x)$, $E_1(y)$.

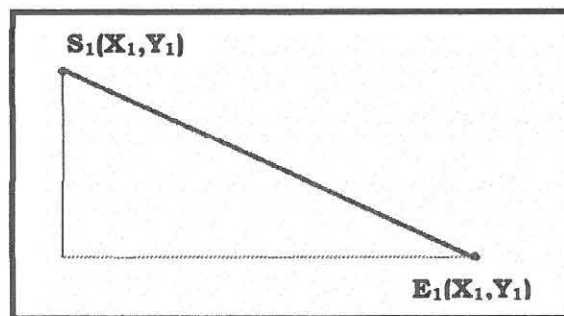
Die lengte tussen die spesifieke vier veranderlikes word bepaal met behulp van Pythagoras (vrgl. 13), soos in Figuur 4.13 aangedui word.

$$\text{Lengte} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (13)$$

waar x = x-koördinaat

y = y-koördinaat

Die volgende lengtes tussen S_1 en die res van die E-beeldelemente word dan met dieselfde formule bereken (Figuur 4.11).



Figuur 4.13 Pythagoras se stelling word gebruik om die lengte tussen twee beeldelemente te bepaal

Nadat al die moontlike afstande tussen E en S bepaal is, word die langste lengte geselekteer. Hierdie lengte sal dan die lengte van die voorwerp wees.

4.1.4 Die wyse waarop die sagteware werk

Die sagteware is vir Windows 95- en 98-platforms geskryf in die grafiese gebruikerskoppelvlak, (GUI)-formaat³. “Microsoft Visual Basic 6” is

³ “Graphical user interface”: Dit is die formaat waarmee die gebruiker berekeninge grafies op die skerm kan uitvoer en resultate kan waarneem.

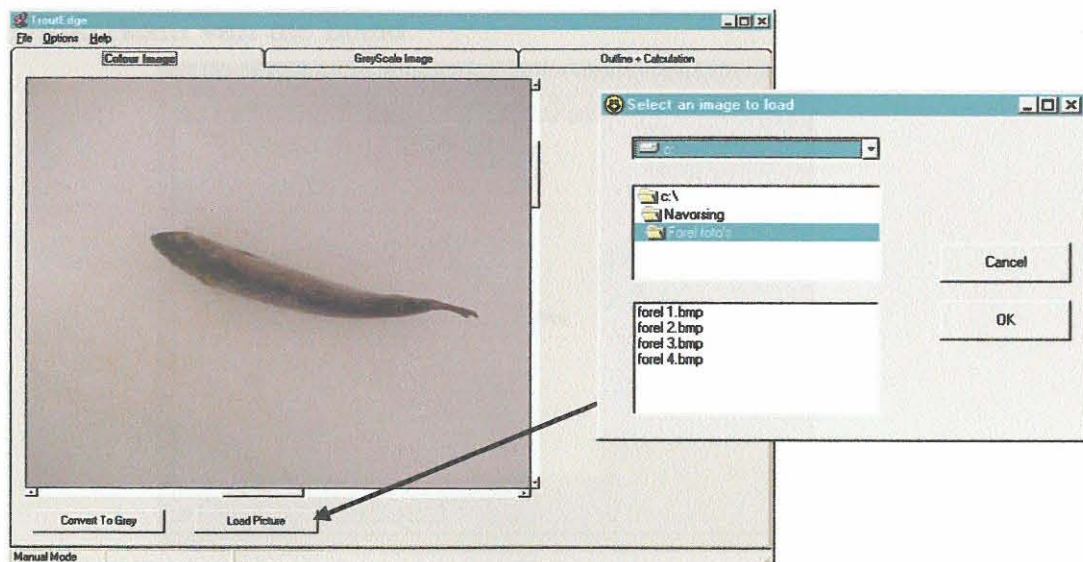
gebruik as programmeringstaal, ontwerp om die ontwikkeling van grafiese “Windows”-sagteware te vereenvoudig. Daar is hoofsaaklik drie ontwerpstappe in “Visual Basic” [17, p.3]:

- Die plasing van `n beheeritem op die beheerraampie.
- Die voorsiening van die nodige metodes aan die beheeritem.
- Die skryf van die “Basic”-programkode.

Voorbeelde van beheeritems is die drukknoppe, opsieknoppe, teksraampies en talle ander. Die metodes en eienskappe waarvan melding gemaak word, spesifiseer voorkomsverwante karakteristieke van die onderskeie beheeritems [3, p. 22]. “Visual Basic” is gebruik om die bewerkinge wat uigevoer word op beeld voor te stel.

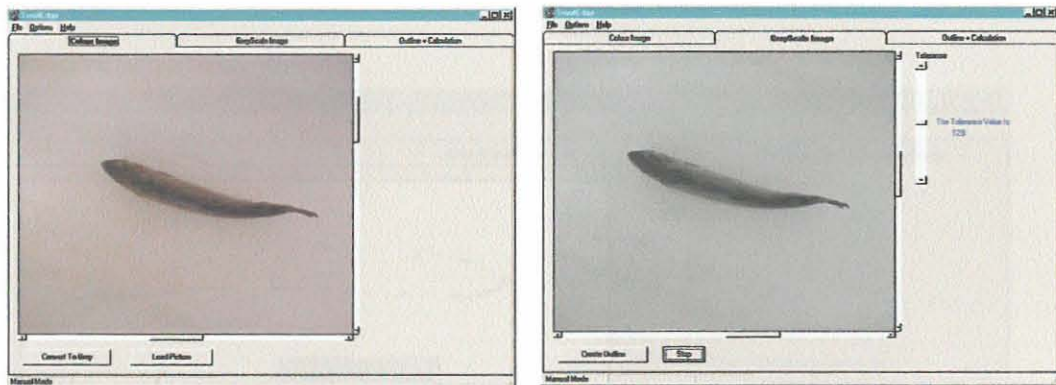
Troutedge is ontwikkel vir die meting van die lengte van gedigitaliseerde forelfoto’s, maar kan ook vir ander toepassings gebruik word. Ter verduideliking van die sagteware, sal gebruik gemaak word van die verskillende vensters in die program.

In die eerste venster is daar drie stensils: Kleurbeeld(“Colour Image”), Grys vlakbeeld(“GrayScale Image”) en Raamwerk + Berekening (“Outline + Calculation”). Elk van hierdie drie stensils bestaan uit die nodige knoppies vir `n volgende bewerking. Op die eerste stensil moet die beeld gelaai word deur middle van die knoppie ‘Laai Beeld’ (“Load Picture”):



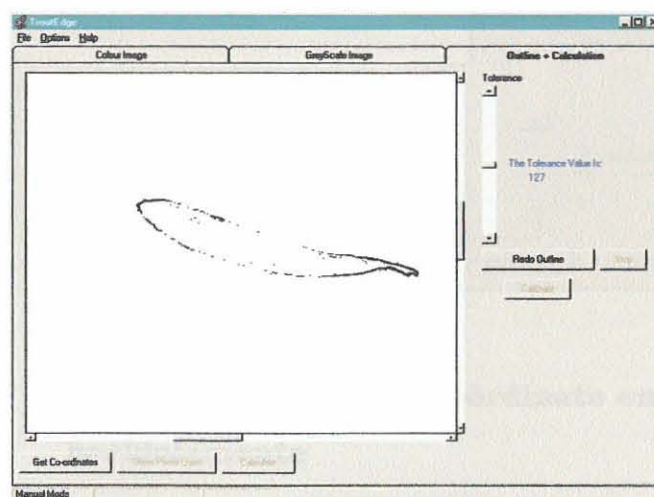
Figuur 4.14 Metode hoe `n beeld in die program gelaai word

Nadat 'n kleurbeeld ingelees is, moet dit omgeskakel word na 'n grysvlakbeeld. Dit word gedoen deur te druk op die knoppie, 'Omskakeling na Grys' ("Convert to Gray"). Hierdie beeldverwerkings-tegniek skakel outomaties oor van die eerste stensil na die tweede waar die verandering sigbaar raak. Die verskil is sigbaar in Figuur 4.15. Die kode nodig vir die omskakeling word aangegee in Bylae A.



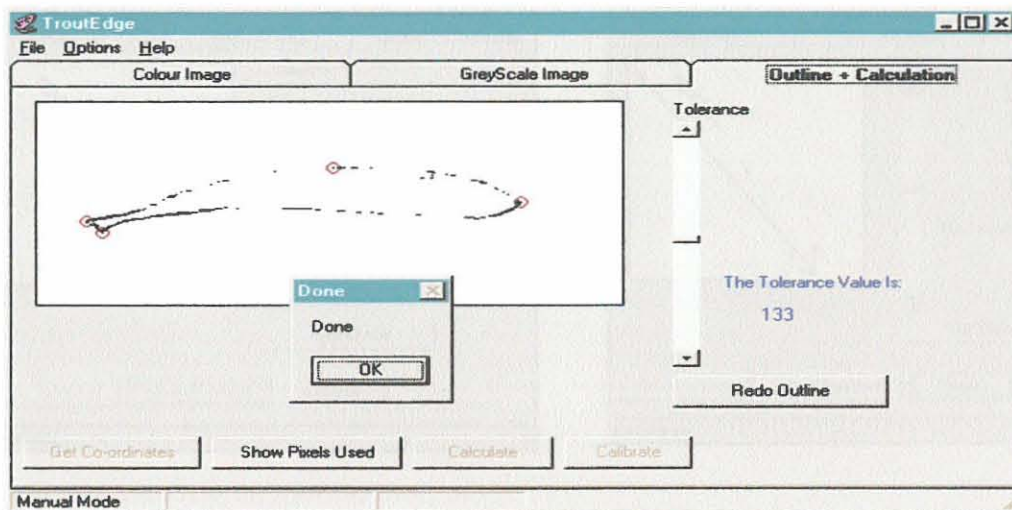
Figuur 4.15 Kleur- na 'n grysvlakbeeldomskakeling

Deur op die knoppie 'Skep Raam' ("Create Outline") te druk, word die grysvlakbeeld omgeskakel na 'n randbeeld. Hier is die drumpelwaarde 'n belangrike aspek, aangesien dit die breedte en suiwerheid van die rand beïnvloed. Die knoppie 'Oordoen van Raamwrek' ("Redo Outline") op die derde stensil kan gebruik word om die randbeeld herhaaldelik te bepaal, indien daar enige ongewenste elemente teenwoordig sou wees. Die drumpelwaarde word verstel deur middel van die vertikale skuier aan die linkerkant van die beeld.

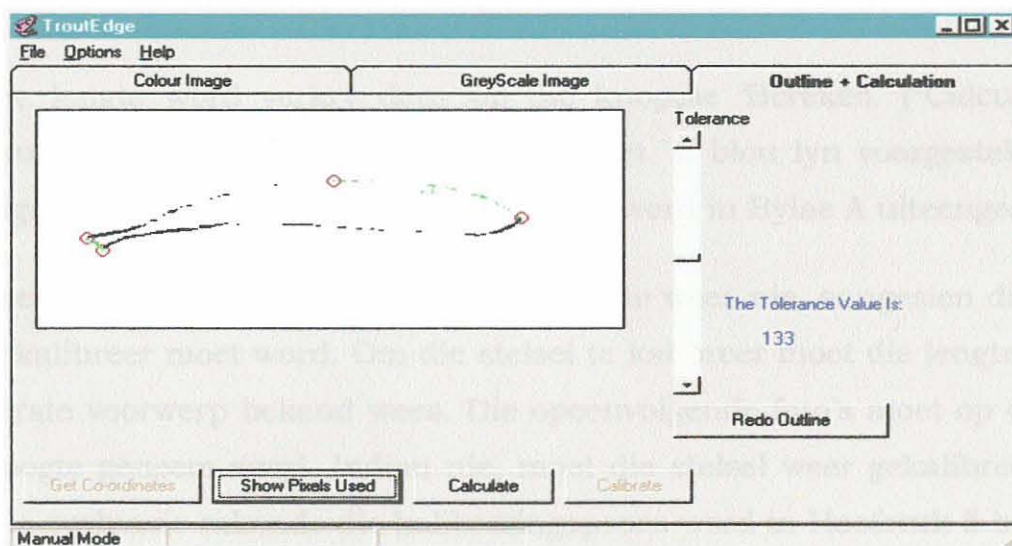


Figuur 4.16 Bepaling van die randbeeld

Sodra die hele beeld geskandeer en die randbeeld verkry is, word die knoppie 'Verkry Koördinate' ("Get Co-ordinates") geaktiveer. Sodra daarop gedruk word, sal die proses begin om die vier hoofpunte te verkry. Daarna sal daar 'n "Done"-venster verskyn en die volgende knoppie, 'Vertoon beeldelemente gebruik' ("Show pixels used"), sal geaktiveer word. Die beeldelemente wat gebruik word tussen hierdie vier punte word in groen aangedui.



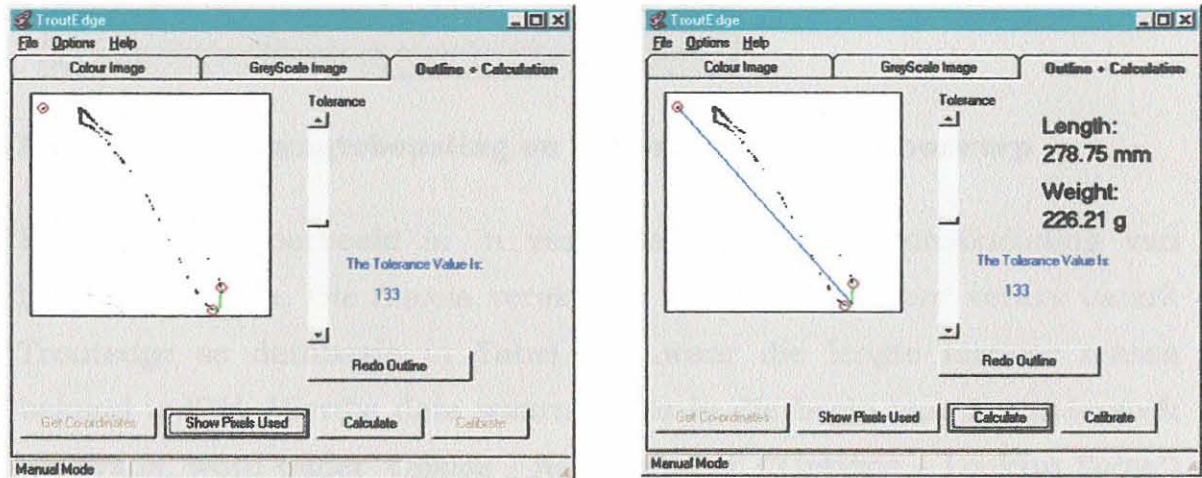
(a)



(b)

Figuur 4.17 Bepaling van die vier koördinate en die nodige beeldelemente

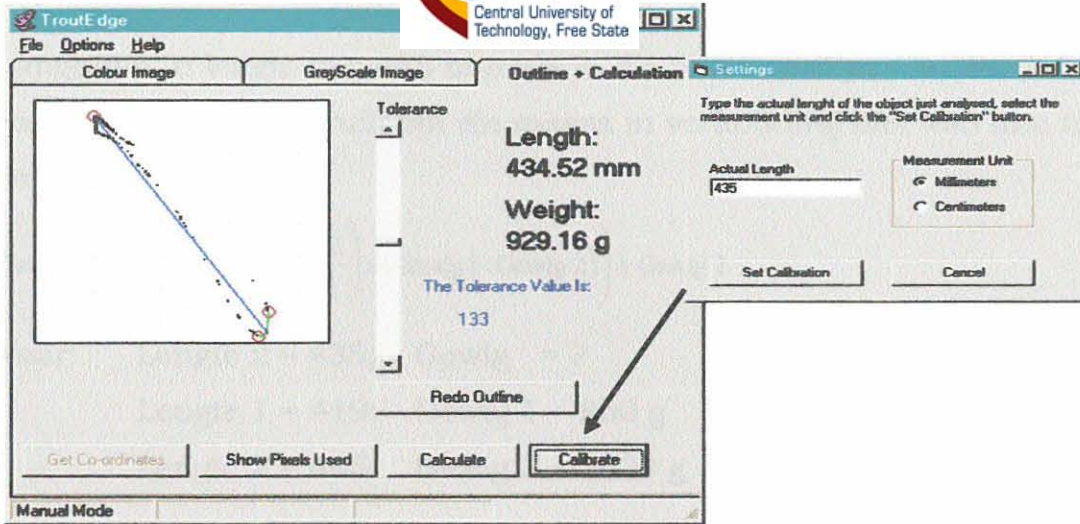
Die beeldelemente (groen), tussen die vier coördinate (rooi punt), word aangedui om foutsporing te vergemaklik, aangesien die moontlikheid bestaan dat daar 'n ongewenste beeldelement enige plek op die beeld teenwoordig kan wees wat nie deur die blote oog sigbaar is nie. Die teenwoordigheid van so 'n beeldelement kan dus maklik opgespoor word indien enige van die vier koördinate nie op die randbeeld sigbaar is nie. 'n Voorbeeld hiervan is sigbaar in Figuur 4.18.



Figuur 4.18 Die resultate wat deur onsigbare beeldelemente beïnvloed is

Die lengte word verkry deur op die knoppie 'Bereken' ("Calculate") te druk. Die lengte wat bepaal is, word met 'n blou lyn voorgestel soos in Figuur 4.19. Die kode vir lengtebepaling word in Bylae A uiteengesit.

Die eerste lengte sal nie 'n korrekte lengte wees nie, aangesien die stelsel gekalibreer moet word. Om die stelsel te kalibreer moet die lengte van die eerste voorwerp bekend wees. Die opeenvolgende foto's moet op dieselfde hoogte geneem word. Indien nie, moet die stelsel weer gekalibreer word. Die evaluasie rakende die kalibreringsposes word in Hoofstuk 5 bespreek. As daar op die knoppie 'Kalibreer' ("Calibrate") gedruk word, bestaan die opsie om die lengte in millimeter of sentimeter in die spasie daarvoor gelaat, te tik. Deur gebruik te maak van die knoppie 'Stel Kalibrering' ("Set Calibration"), word die verhouding bereken en die lengte dienooreenkomstig verander.



Figuur 4.19 Lengtebepaling en kalibrering van 'n voorwerp

Die volgende voorbeeld is 'n verduideliking vir die omskakeling van lengte na massa. Die massa verskaf in Figuur 4.20, word verkry vanuit Troutedge se databasis in Tabel 4.1, waar die lengte teenoor massa bekend is [26]. Hierdie data waarna daar in die begin van die ondersoek verwys is, word onder 'Opsies - Naslaantabel' ("Options - Lookup Table") geplaas. Dit lyk soos volg:

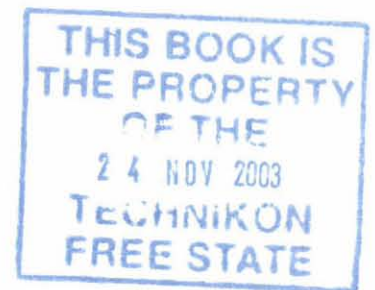
Lookup Table

Length must be entered in Millimeters.
Weight must be entered in Grams.
Click on cell to edit and press Enter when done.

Length	Weight
368	566.99
381	623.69
394	680.39
406	765.44
419	850.49
432	907.18
445	1020.58
457	1105.63
470	1190.68
483	1304.08
495	1417.48
500	1500.00

Add Line Delete Line

Save Cancel



Figuur 4.20 Tabel van lengte teenoor massa

Aangesien 'n lengte van 435 mm nie in die tabel verskyn nie, word die volgende formule gebruik om die massa in verhouding met 435 mm te bereken.

$$\text{Gewig} = \left(\left(\frac{\text{Lengte } x - \text{Lengte } 2}{\text{Lengte } 1 - \text{Lengte } 2} \right) \times (\text{Gewig } 1 - \text{Gewig } 2) \right) + \text{Gewig } 2 \quad (14)$$

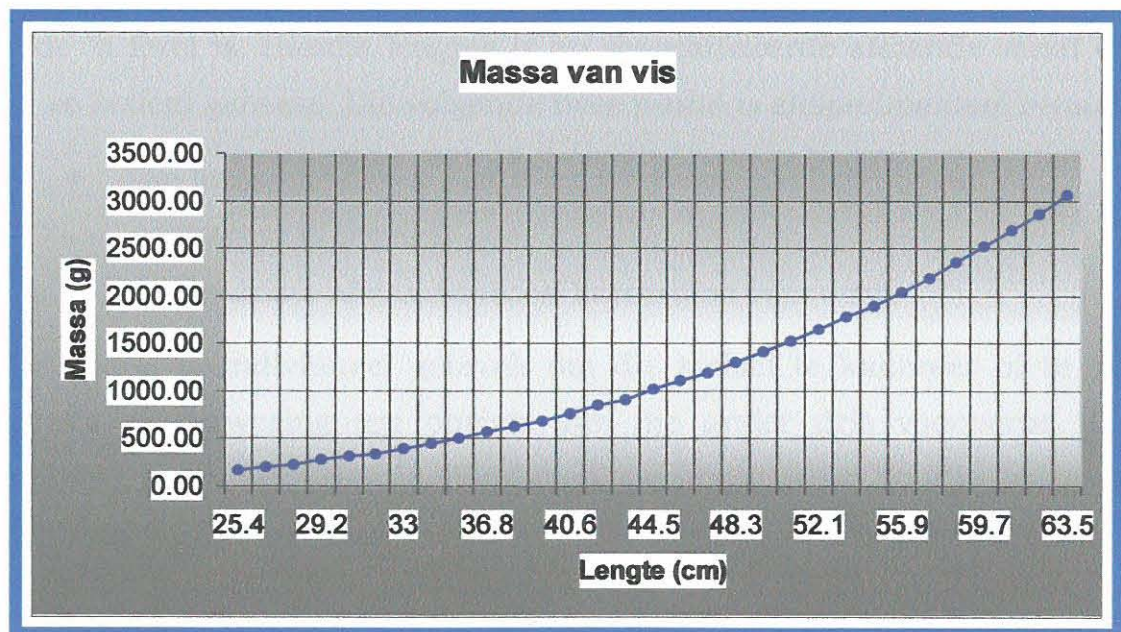
waar: Lengte x = 435; Gewig = ?

Lengte 1 = 419; Gewig 1 = 850 g

Lengte 2 = 445 Gewig 2 = 1020 g

4.1.4.1 Lengte-massaverhouding

Nadat verskeie onsuksesvolle ondersoeke geloods is om die verband tussen die lengte en die massa van 'n forel kry, is 'n webtuiste van 'n groep forelnavorsers in New York ontdek. Die massa soos aangedui in Tabel 4.1 is 'n gemiddelde waarde van die betrokke visspesie wat in die staat van New York nagevors is [26]. As die verband tussen die lengte en die gewig grafies voorgestel word met die data in Tabel 4.1, vertoon die grafiek soos volg.



Figuur 4.21 Grafiese voorstelling van die verband tussen die lengte en die massa van forelle [26]

HOOFSTUK 5

STELSELEVALUASIE EN RESULTATE

5.1 Stelselevaluasie

Die navorsing toon dat tweedimensionele beeldverwerkingstegnieke so ontwikkel kan word om die forelboer te help met belangrike bestuur en bemarkingsbesluite. Die forelboer kan groei data gebruik om nuttige inligting soos gewigeenvormigheid en die voorkoms van siektes te identifiseer. Die navorsingstelsel en tegnieke is op lewendige visse beproef en die resultate toon dat dit wel moontlik is om die massa suksesvol te bepaal. Die stelsel is bekostigbaar en kan tot groot voordeel wees om toepaslike of soorgelyke senarios op te los.

5.2 Resultate

Een van die eksperimente wat gedoen is, behels die fotografering van vier voorwerp lengtes (50 mm, 137mm, 175mm en 435mm). Die eerste drie voorwerpe (prop, pen en liniaal) is voorstellings van vis lengtes terwyl die vierde `n forel is. Hierdie lengtes is op gespesifiseerde afstande vanaf die bad se bodem geneem. Die volgende twee punte is eksperimenteel bepaal:

- Die mees effektiewe voorwerp lengte vir kalibrering van die sagteware is bepaal.
- Die geskikte afstand tussen kamera en voorwerp.

Elke lengte is individueel gebruik om die stelsel te kalibreer of te wel gebruik as verwysing ten opsigte van die ander drie voorwerpe. Die resultate verkry gedurende hierdie eksperiment word in die volgende tabelle en grafieke voorgestel.

Tabel 5.1 Kalibrering van die stelsels met die 50 mm voorwerp

Stelsel evaluasie A				
Afstand tussen kamera en bodem	Berekende lengte deur Troutedge			
mm	50 mm	137 mm	175 mm	435 mm
300	49.53	129	162	~
500	50.05	139.1	170.95	445
700	49.98	134	177.38	449
900	49.38	135.66	168	450.3
1100	50.16	141	171	446
1300	49.16	137	173	448

Tabel 5.2 Kalibrering van die stelsels met die 137 mm voorwerp

Stelsel evaluasie B				
Afstand tussen kamera en bodem	Berekende lengte deur Troutedge			
mm	50 mm	137 mm	175 mm	435 mm
300	52.07	135.71	170.56	~
500	49.28	136.96	165.76	439.72
700	51	137	175	447
900	50.4	136.8	170.4	446
1100	49.98	136.71	166.11	442.47
1300	49.59	136.8	172.71	446.31

Tabel 5.3 Kalibrering van die stelsels met die 175 mm voorwerp

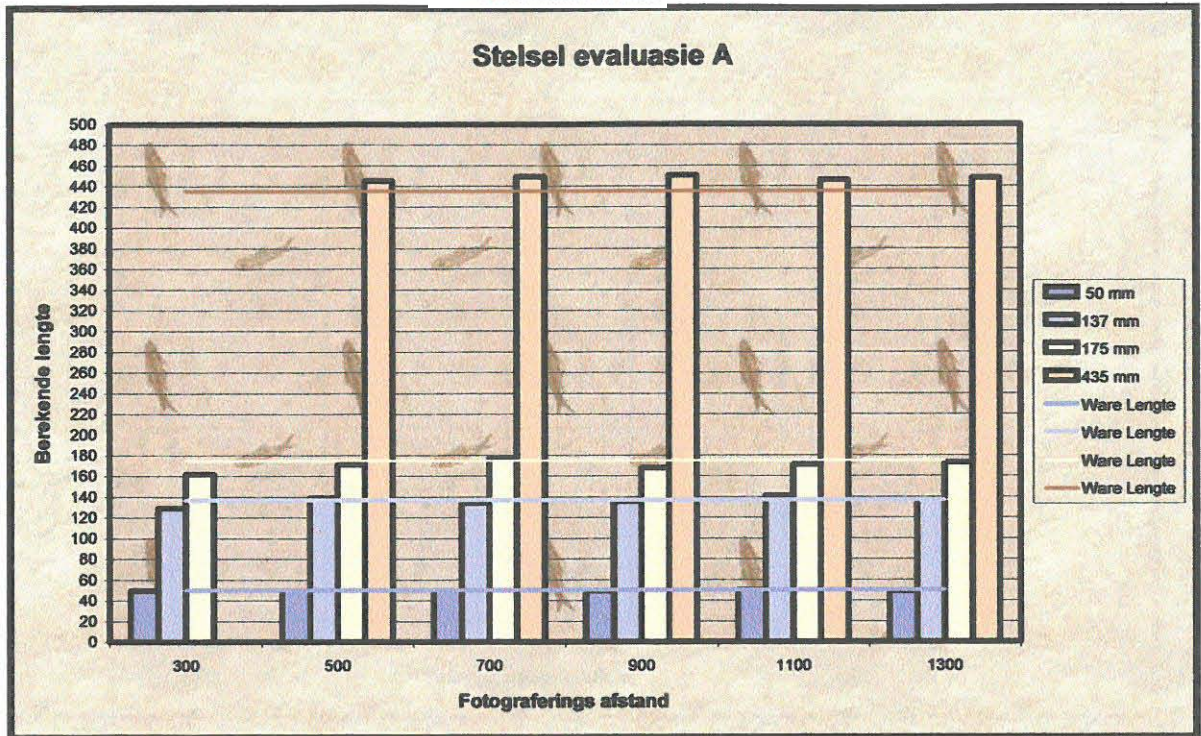
Stelsel evaluasie C				
Afstand tussen kamera en bodem	Berekende lengte deur Troutedge			
mm	50 mm	137 mm	175 mm	435 mm
300	53.35	139.02	174.72	~
500	52.36	145.52	176.12	446
700	51.51	138.5	175.43	450
900	52.5	142.5	175	448
1100	52.7	145.7	175.4	448
1300	55.36	138.7	174.37	447

Tabel 5.4 Kalibrering van die stelsels met die 435 mm voorwerp

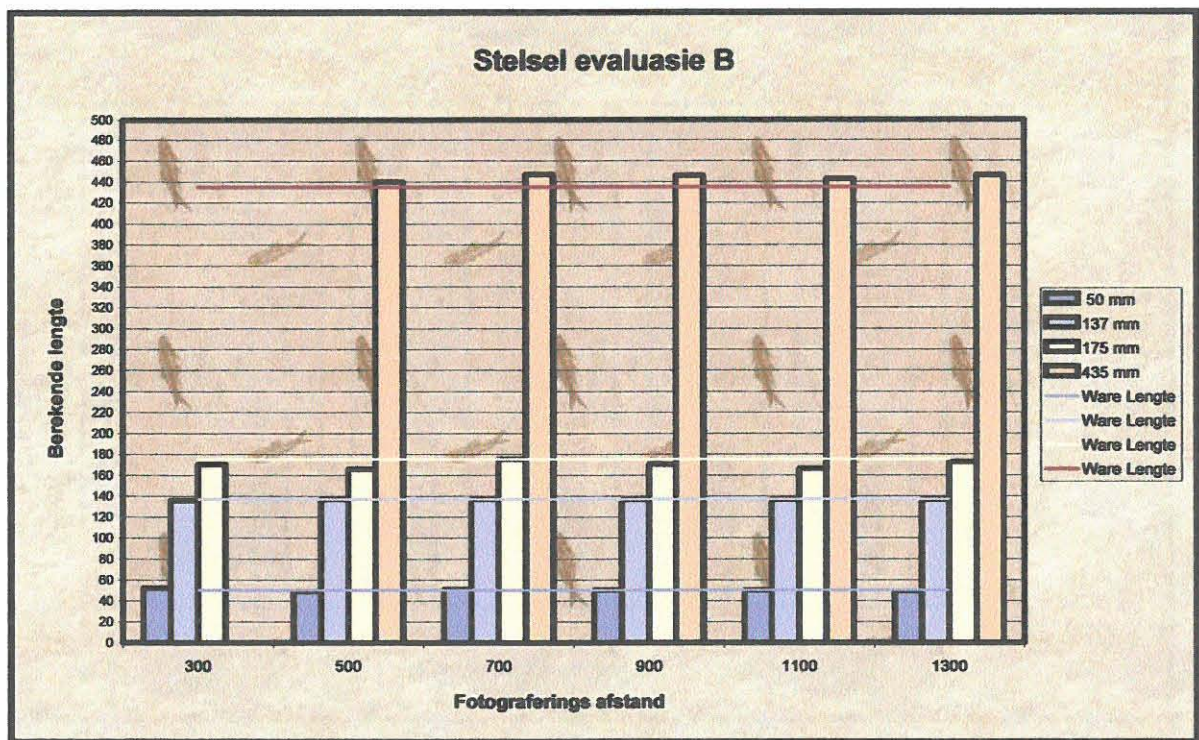
Stelsel evaluasie D				
Afstand tussen kamera en bodem	Berekende lengte deur Troutedge			
mm	50 mm	137 mm	175 mm	435 mm
300	~	~	~	~
500	48.51	134.82	173.17	432.18
700	45.39	130	171.09	433.43
900	47.88	131.2	171.88	433.2
1100	49.3	134.6	173.7	434.6
1300	48.43	134.7	178.67	435.87

In tabelle 5.2 tot 5.4 word die simbool “~” gebruik in die posisie waar die nodige waardes nie verkry kon word nie. In stelsel evaluasie D is daar geen waardes by 300 mm beskikbaar nie, aangesien 435 mm te groot is om op ’n afstand van 300 mm gefotografeer te word. Dië lengte kon dus nie vir kalibrering doeleindes gebruik word nie.

Dieselfde inligting soos uiteengesit in die tabelle kan ook voorgestel word in die volgende grafieke. Die horisontale lyne op elke grafiek stel die ware lengte van die visse voor terwyl die balkies die berekende lengte aandui.

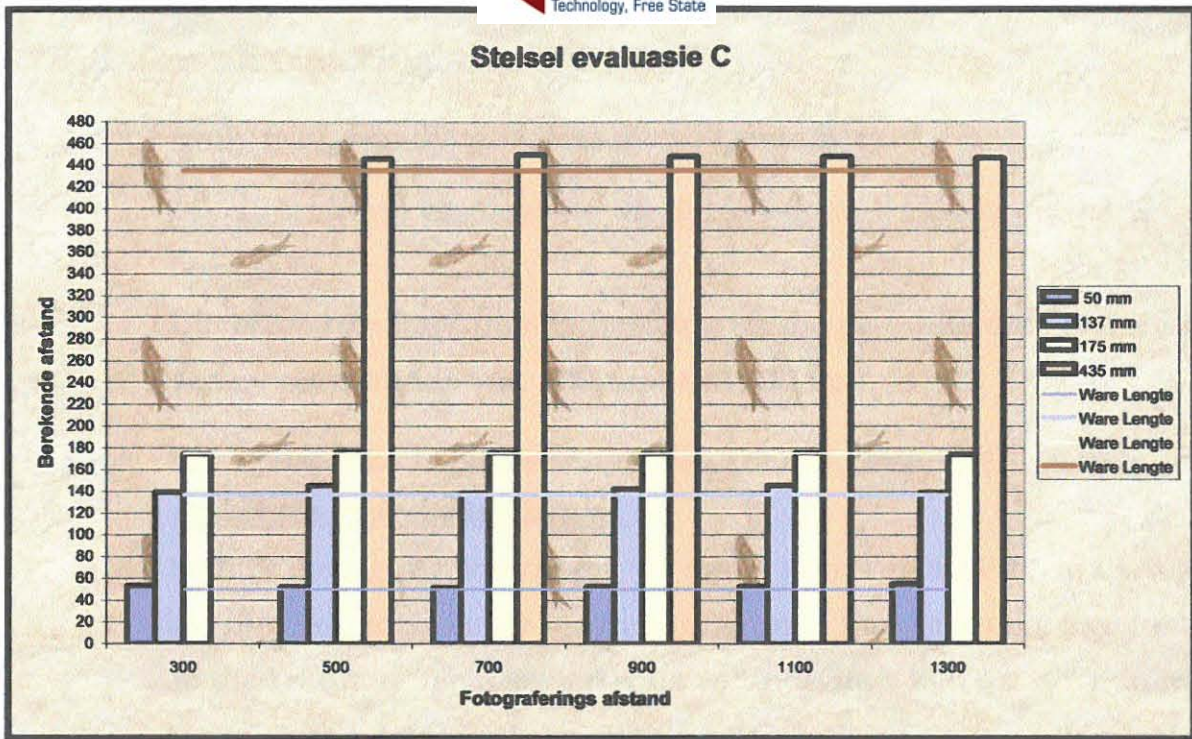


(a)

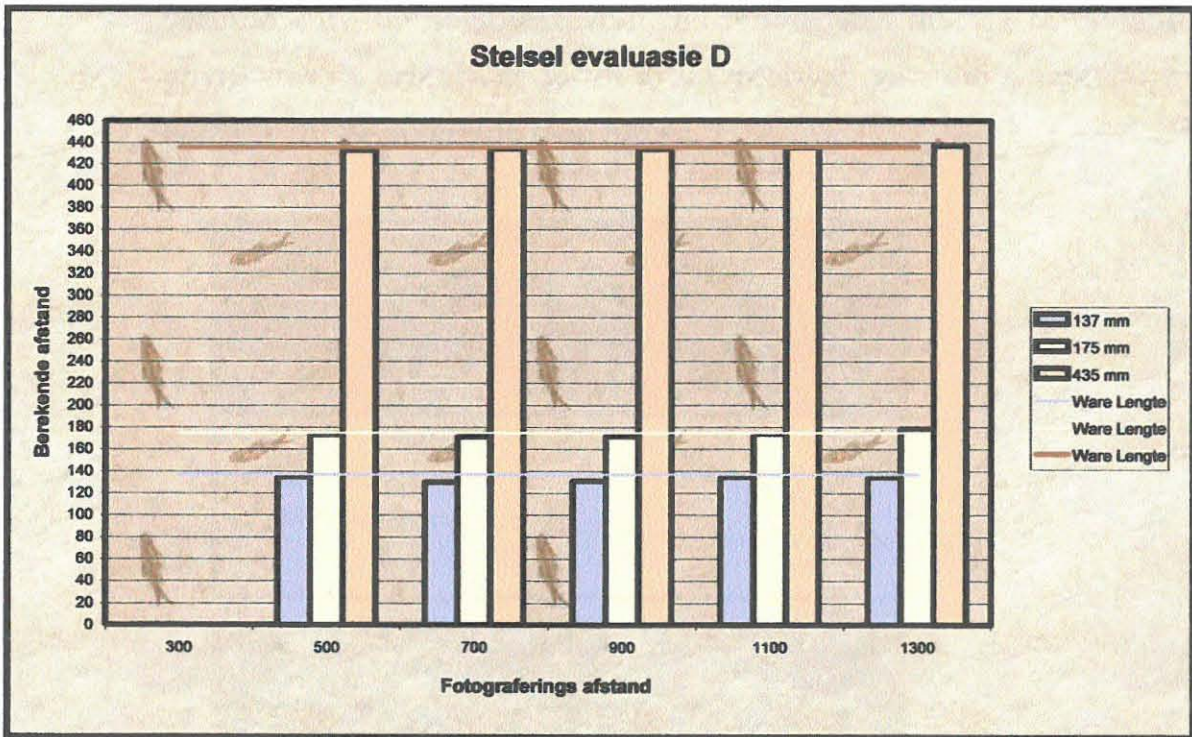


(b)

Figuur 5.1 Die grafieke stel die berekende lengte teenoor die fotograferingsafstand voor



(c)



(d)

Figuur 5.1(vervolg) Die grafieke stel die berekende lengte teenoor die fotograferingsafstand voor

5.3 Gevolgtrekking

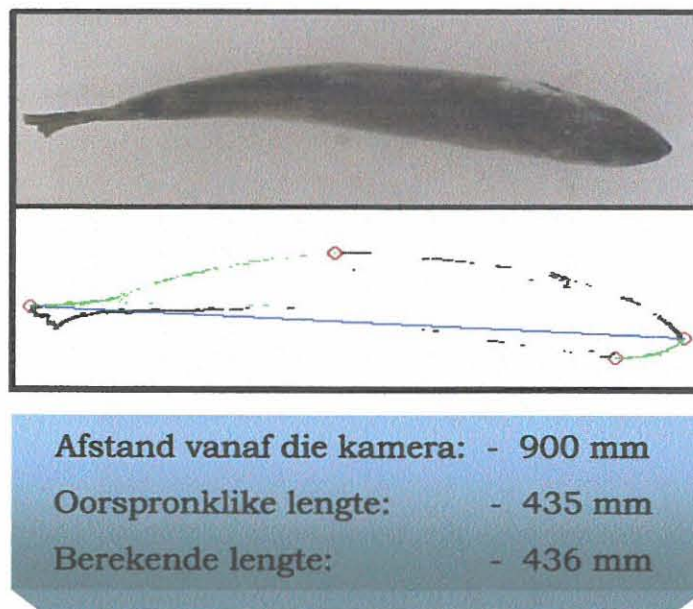
Tydens evaluering van die projek is die volgende bevind:

- **Die akuraatste afstand vir die fotografering van die visse is bepaal.**

Die beste resultate (met betrekking tot die ge-evalueerde stelsel) is verkry op afstande van 700 mm to 1300 mm.

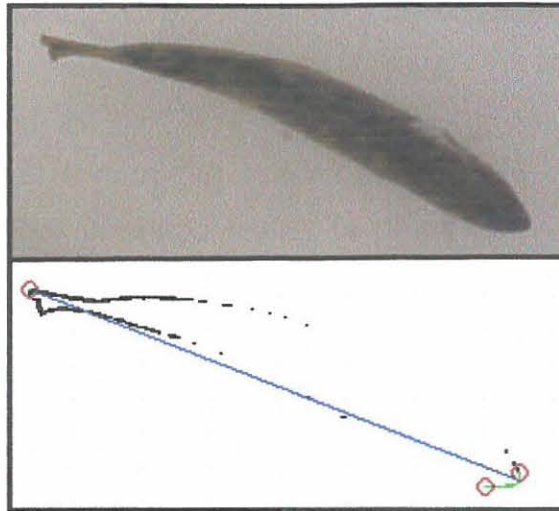
- **Die bewegings oriëntasie van die vis het geen effek op die resultate wat verkry is nie.**

Solank die rigting waarin die vis swem geen vertikale verandering het nie, het horisontale veranderinge geen noemenswaardige invloed op die resultate nie. Die vertikale veranderinge kan tot die minimum beperk word deur die hoeveelheid water in die bad so te reguleer dat daar net genoeg water vir die vis is om horisontaal te beweeg. Die vermoë van die sagteware om die lengte van die vis te bepaal ten spyte van sy oriëntasie is tot groot voordeel van die gebruiker en is sigbaar in die volgende drie figure.



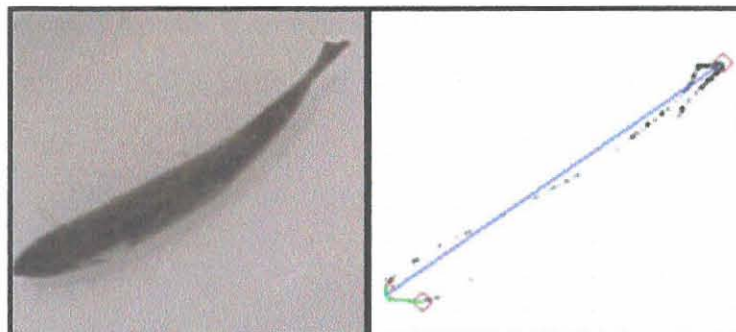
(a)

Figuur 5.2 Lengteberekening by verskillende horisontale oriëntasies



Afstand vanaf die kamera: - 1000 mm
Oorspronklike lengte: - 435 mm
Berekende lengte: - 434 mm

(b)



Afstand vanaf die kamera: - 1000 mm
Oorspronklike lengte: - 435 mm
Berekende lengte: - 434 mm

(c)

Figuur 5.2 (vervolg) Lengteberekening by verskillende horisontale oriëntasies

- **Die effek van die geboë posisie van die vis op lengtebepaling.**

Volgens navorsing resultate in Tabel 4.1 van “Your fish.....With a ruler”[26] is die belangrikste lengte teenoor massa intervalle aangedui. Indien die verskil tussen die eerste twee lengtes van dië tabel vergelyk word, is daar ± 1 cm verskil in lengte. As gevlog hiervan is daar `n redelik variasie wat toegelaat kan word indien nodig. Die sagteware (Trout edge) is ook so ontwikkel dat indien `n lengte tussen twee intervalle sou voorkom, word die verhouding van lengte teenoor massa bereken as aanvaar word dat die intervalle tussen enige twee gegewe punte lineieër is.

Op die grafieke word `n verskil in die berekende lengte en die ware lengte waargeneem. Verskeie faktore kan `n invloed hê op die verskille wat eksperimenteel verkry is byvoorbeeld:

- Die kamera en die vis was nie noodwendig loodreg ten opsigte van mekaar tydens die fotografering van die visse nie.
- Die wateroppervlak was nie ten alle tye spieëlglad nie wat `n steuring(ruis) op die foto veroorsaak het.
- Foto's wat nie 100% in fokus is nie kan beelverwerking belemmer.

HOOFSTUK 6

SAMEVATTING

Dit is moontlik om die benaderde massa van 'n forel te bepaal as die lengte bekend is. Om die benaderde massa van 'n Reënboogforel, Bruin forel, Stroomforel of 'n Meerforel te bepaal, moet die lengte vanaf die stert tot by die mond gemeet word.

Die studie het ten doel gehad die daarstelling van 'n stelsel wat dit moontlik sal maak om die massa van 'n forel te verkry deur middel van die lengte met behulp van digitale beeldverwerking. Deur die gebruik van verskeie tweedimensionele metodes en tegnieke, kan die lengte van 'n voorwerp vanaf 'n digitale beeld bepaal word. Hierdie metodes sluit in: kleur-na-grysvlak omskakeling; randbepaling deur middel van 'n drumpelwaarde; koördinaatbepaling en lengtebepaling.

In Hoofstuk 2 is inleidende inligting verskaf oor beelde, eienskappe van beelde en beeldfunksies, wiskundige berekeninge soos: konvolusie, Fourier-analise en tegnieke vir beeldmanipulasie, asook kodes wat op beelde toegepas kan word. Daar word kort na die menslike waarneming verwys, aangesien dit 'n belangrike invloed op stelselontleding het en daar is ook verwys na verskillende tipes ruis hoofsaaklik vir CCD-kameras en asook die werking van CCD-kameras.

In Hoofstuk 3 is die bepaling van grysvlak histogramme en die bepaling van drumpel waardes ondersoek. Verskeie metodes van randbepaling is ontleed en die effek van verskillende drumpelwaardes daarop is bestudeer. Segmenteringstegnieke is gebruik om die voorwerp van belang te vind, of te wel die voorgrond van die agtergrond te onderskei.

In Hoofstuk 4 word die toerusting wat gebruik is behandel asook hoe al die substelsels wat gebruik is om resultate te verkry met mekaar

geïntegreer word. Die werking en beskrywing van die sagteware word ook verduidelik.

In Hoofstuk 5 is die werking van die stelse geëvalueer en word die resultate wat verkry is tydens die stelsel evaluasie bespreek. Die moontlike oorsake wat 'n invloed op die resultate kon hê is uitgelig.

Navorsings moontlikhede

Soos in die navorsingsstudie bespreek is, kan 'n meer outomatiese sisteem ontwikkel word. Lasermeting is een van die tegnieke wat ingesluit kan word sodat die afstand outomaties deur 'n laser bepaal kan word. Die afstand kan dan in die sagteware ingelees word waarna die lengte van die visse bereken kan word om die massa te verskaf.

In samewerking met die laser kan sensors op strategiese plekke binne die bad gemonteer word. Hierdie sensors kan dan die digitale kamera of digitale videokamera aktiveer wanneer die beweging van 'n vis waargeneem word.

Die groei sowel as die toename in gewig van 'n aantal forelle oor 'n sekere tydperk kan geïmplimenter word om outomaties self voerders bo die forel teëldamme aan te bring. Dit kan die teëling van forelle vergemaklik.

KODELYS VAN TROUTEDGE

Hoofvorm

Public Message As String
Public StopGray As Boolean
Public StopOutline As Boolean

Dim result As String

Option Explicit

'Roetine om die lengte van die vis te bepaal

Private Sub cmdCalculate_Click()

Dim LargestLength As Integer
Dim LResult As Integer

Dim X1 As Integer
Dim X2 As Integer
Dim Y1 As Integer
Dim Y2 As Integer

Dim LX1 As Integer
Dim LX2 As Integer
Dim LY1 As Integer
Dim LY2 As Integer

Dim J As Integer

LargestLength = 0

For I = 0 To Arr1Size

DoEvents

'Gebruik al die onderste koördinate

X1 = Arr1(I, 0)

Y1 = Arr1(I, 1)

For J = 0 To Arr2Size

'Gebruik al die boonste koördinate

X2 = Arr2(J, 0)

Y2 = Arr2(J, 1)

'Gebruik Pythagoras om die afstand tussen twee 'koördinate te kry
LResult = Sqr(((X2 - X1) ^ 2) + (Y2 - Y1) ^ 2)

If LResult > LargestLength Then

LargestLength = LResult

LX1 = X1

LX2 = X2

LY1 = Y1

LY2 = Y2

End If

Next J

Next I

'Teken 'n blou lyn om aan te dui tussen watter punte die langste lengte is

picOutline.Line (LX1, LY1)-(LX2, LY2), QBColor(9)

Length = LargestLength

lblLength.Visible = True

'Vertoon die lengte in die regte eenheid

If Unit = "cm" Then

lblLengthVal.Caption = ((Length * Ratio) / 10) & " " & Unit

Else

lblLengthVal.Caption = (Length * Ratio) & " " & Unit

End If

lblLengthVal.Visible = True

Call CalcWeight

cmdCalibrate.Enabled = True

End Sub

Private Sub cmdCalibrate_Click()

'Vertoon die instellingvenster wanneer die kalibreerknoppe gedruk word

frmSettings.Show

End Sub

```
Private Sub cmdCTG_Click()
```

```
cmdCTOutline.Enabled = False  
cmdRedoOutline.Enabled = False  
cmdGetPoints.Enabled = False  
cmdShow.Enabled = False  
cmdCalculate.Enabled = False  
lblLength.Visible = False  
lblLengthVal.Visible = False
```

```
If mnuManual.Checked = False Then  
result = MsgBox("Processing will be automatic. Proceed?", vbYesNo +  
vbApplicationModal + vbExclamation, "")  
Else  
ConvertToGray  
Exit Sub  
End If
```

```
Select Case result
```

```
Case vbYes  
ConvertToGray
```

```
Case vbNo  
DoEvents
```

```
End Select
```

```
End Sub
```

```
-----  
Private Sub cmdCTOutline_Click()
```

```
cmdGetPoints.Enabled = False  
cmdRedoOutline.Enabled = False  
cmdShow.Enabled = False  
cmdCalculate.Enabled = False
```

```
cmdCalibrate.Enabled = False
```

```
lblLength.Visible = False  
lblLengthVal.Visible = False  
lblWeight.Visible = False  
lblWeightVal.Visible = False
```

```
'Hou die kleur van die beeldelement minus die kleur van die  
beeldelement langsaan
```

```
Dim Col As Long  
Dim Total As Long  
Dim X As Integer
```

Dim Y As Integer

picOutline.Width = picGray.Width
picOutline.Height = picGray.Height

'Vertoon die tabuleerder met die randbeeld

SSTab1.Tab = 2

SSTab1.Enabled = False

DoEvents

Message = "Creating Outline"

StatusBar1.Panels(2).Text = Message

Timer1.Enabled = False

Timer1.Enabled = True

StopOutline = False

'Bepaal die aantal beeldelemente in die beeld (word gebruik om die vordering van die omskakelingsproses aan te dui)

Total = picGray.ScaleWidth * picGray.ScaleHeight

picOutline.Cls

'skandeer deur die x-beeldelemente

For X = 1 To picGray.ScaleWidth

'skandeer deur die y-beeldelemente

For Y = 1 To picGray.ScaleHeight

'Hou die kleur van die beeldelement minus die kleur van die beeldelement bokant

Col = Abs(picGray.Point(X, Y) - picGray.Point(X, Y - 1))

'Kies of die kleur van hoë kontras is.

If Col > vsbTolerance.Value ^ 3 Then Col = 0 Else Col = vbWhite

'Karteer beeldelement

If Col = 0 Then picOutline.PSet (X, Y), Col

'Hou die kleur van die beeldelement minus die kleur van die beeldelement links

Col = Abs(picGray.Point(X, Y) - picGray.Point(X - 1, Y))

'Kies of die kleur van hoë kontras is

If Col > vsbTolerance.Value ^ 3 Then Col = 0 Else Col = vbWhite

'Karteer beeldelement

If Col = 0 Then picOutline.PSet (X, Y), Col

Next Y 'loop through the y-pixels

picOutline.Refresh
DoEvents

If StopOutline = True Then GoTo Stopp

StatusBar1.Panels(3).Text = Round(((X * (Y / 10)) / Total) * 1000) & " %
Done"

'Skandeer deur die x-beeldelemente
Next X

SSTab1.Enabled = True
Timer1.Enabled = False
Timer2.Enabled = True

cmdGetPoints.Enabled = True
StatusBar1.Panels(2).Text = Message
cmdRedoOutline.Enabled = True
GoTo SubEnd

Stopp:
Message = "Aborted"
StatusBar1.Panels(2).Text = Message
SSTab1.Enabled = True
Timer1.Enabled = False
Timer2.Enabled = True

SubEnd:
End Sub

Private Sub cmdGetPoints_Click()

Dim Y As Integer
Dim X As Integer

Dim X1 As Integer
Dim X2 As Integer

Dim Y1 As Integer
Dim Y2 As Integer
Dim Temp1X As Integer
Dim Temp1Y As Integer
Dim Temp2X As Integer
Dim Temp2Y As Integer

Dim RSFilter As Recordset

Dim XX1 As Integer
Dim YY1 As Integer


```
Dim Counter As Integer  
Dim ArrSize As Integer
```

```
cmdShow.Enabled = False  
cmdCalculate.Enabled = False  
cmdRedoOutline.Enabled = False  
lblLength.Visible = False  
lblLengthVal.Visible = False
```

```
SSTab1.Enabled = False
```

```
'Maak die databasis oop  
db = App.Path & "\Coordinates.mdb"  
Set Maindb = OpenDatabase(db)
```

```
'Verwyder alle rekords uit die "Coord" tabel sodat  
'die nuwe koördinate in die tabel gestoor kan word  
Maindb.Execute "Delete * from Coord"
```

```
'Maak die databasis toe  
Maindb.Close
```

```
DoEvents
```

```
'Open die databasis weer  
db = App.Path & "\Coordinates.mdb"  
Set Maindb = OpenDatabase(db)
```

```
'Stel 'n rekordstel op (sal leeg wees want die tabel  
'is skoon), om die nuwe waardes te huisves  
rs = "Select * from Coord"  
Set rsCoord = Maindb.OpenRecordset(rs)
```

```
'Stel die uiterste perke na uitermatige waardes toe  
X1 = 2000  
X2 = 0  
Y1 = 2000  
Y2 = 0
```

```
'Werk deur die hele hoogte van die prent  
For Y = 1 To picOutline.ScaleHeight
```

```
'Werk deur die hele wydte van die prent  
For X = 1 To picOutline.ScaleWidth
```

```
If picOutline.Point(X, Y) = QBColor(0) Then
```

```
'Stoor die koördinate van die swart beeldelemente in die  
'rekordstel  
With rsCoord  
.AddNew
```

```
!X = X  
!Y = Y  
.Update  
End With
```

'Die volgende kode bepaal die waardes van die uiterste linker, regter, boonste en onderste beeldelemente

```
If X < X1 Then  
X1 = X  
Temp2X = X  
Temp2Y = Y  
End If
```

```
If X > X2 Then  
X2 = X  
Temp1X = X  
Temp1Y = Y  
End If
```

```
If Y < Y1 Then  
Y1 = Y  
AX = X  
AY = Y  
End If
```

```
If Y > Y2 Then  
Y2 = Y  
DX = X  
DY = Y  
End If
```

```
End If
```

```
Next X
```

```
Next Y
```

'Maak die databasis toe
Maindb.Close

```
If AX > DX Then  
BX = Temp1X  
BY = Temp1Y  
CX = Temp2X  
CY = Temp2Y  
Else  
BX = Temp2X  
BY = Temp2Y  
CX = Temp1X
```

CY = Temp1Y
End If

'Trek rooi sirkeltjies om die vier uiterste koördinate

picOutline.Circle (AX, AY), 4, QBColor(4)
picOutline.Circle (BX, BY), 4, QBColor(4)
picOutline.Circle (CX, CY), 4, QBColor(4)
picOutline.Circle (DX, DY), 4, QBColor(4)

picOutline.Refresh

DoEvents

'Stel databasis pad op

db = App.Path & "\Coordinates.mdb"

'Maak databasis oop

Set Maindb = OpenDatabase(db)

'-----Onderste Waardes-----

'Stel navraag ("query") op vir onderste waardes

rs = "Select * from Coord where X between "
rs = rs & CStr(CX)
rs = rs & " and "
rs = rs & CStr(DX) & " AND "
rs = rs & "Y between "
rs = rs & CStr(CY)

rs = rs & " and "
rs = rs & CStr(DY)

'Vra databasis om die waardes te kry

Set RSFilter = Maindb.OpenRecordset(rs)

'Bepaal die aantal rekords in die rekordstel

With RSFilter
.MoveLast
.MoveFirst

'Stel 'n veranderlike gelyk aan die hoeveelheid rekords

Counter = .RecordCount
End With

Arr1Size = Counter - 1
ReDim Arr1(Counter - 1, 1)

With RSFilter
.MoveFirst

For I = 1 To .RecordCount

'Skryf die x-koördinaat na 'n veranderlike

XX1 = !X

'Skryf die x-koördinaat na die X-kolom van die "array"

Arr1(I - 1, 0) = XX1

'Skryf die y-koördinaat na 'n veranderlike

YY1 = !Y

'Skryf die y-koördinaat na die Y-kolom van die "array"

Arr1(I - 1, 1) = YY1

'Gaan na die volgende rekord toe

.MoveNext

Next I

End With

RSFilter.Close

'-----Boonste Waardes-----

'Stel navraag "query" op vir boonste waardes

rs = ""

rs = "Select * from Coord where X between "

rs = rs & CStr(AX)

rs = rs & " and "

rs = rs & CStr(BX) & " AND "

rs = rs & "Y between "

rs = rs & CStr(AY)

rs = rs & " and "

rs = rs & CStr(BY)

'Vra databasis om die waardes te kry

Set RSFilter = Maindb.OpenRecordset(rs)

With RSFilter

'Gaan na die laaste rekord (om te tel hoeveel rekords

'in die rekordstel is - anders is "RecordCount" = 0)

.MoveLast

'Gaan terug na die eerste rekord toe

.MoveFirst

'Stel 'n veranderlike gelyk aan die hoeveelheid rekords

Counter = .RecordCount

End With

Arr2Size = Counter - 1

ReDim Arr2(Counter - 1, 1)

With RSFilter

.MoveFirst

```
For I = 1 To .RecordCount
'Skryf die x-koördinaat na 'n veranderlike
XX1 = !X
'Skryf die x-koördinaat na die X-kolom van die opstelling
Arr2(I - 1, 0) = XX1
'Skryf die y-koördinaat na 'n veranderlike
YY1 = !Y
'Skryf die y-koördinaat na die Y-kolom van die opstelling
Arr2(I - 1, 1) = YY1
'Skuif aan na die volgende rekord in die rekordstel
.MoveNext
Next I

End With

'Maak die databasis toe, sodat ander applikasies dit ook kan
'gebruik indien nodig (slegs vir goeieprogrammering)
Maindb.Close

'Vertoon 'n boodskap om te bevestig dat al die nodige
'rekords verkry is
result = MsgBox("Done", vbOKOnly, "Done")

SSTab1.Enabled = True

cmdShow.Enabled = True

cmdRedoOutline.Enabled = True
cmdGetPoints.Enabled = False

End Sub

-----
Private Sub cmdLoad_Click()

'Form vir die "Load Picture" aksie
frmLoad.Show

End Sub

-----
Private Sub cmdRedoOutline_Click()

'Roetine wanneer die "Redo Outline"-knoppie gedruk word

lblLength.Visible = False
lblLengthVal.Visible = False

'Roep die oorspronklike roetine
Call cmdCTOutline_Click
```

End Sub

Private Sub cmdShow_Click()

**'Roetine om te wys watter koördinate gebruik word
'vir die lengtebepaling**

Dim XX1 As Integer
Dim YY1 As Integer

cmdCalculate.Enabled = False
lblLength.Visible = False
lblLengthVal.Visible = False
lblWeight.Visible = False
lblWeightVal.Visible = False

'Onderste koördinate

For I = 0 To Arr1Size
XX1 = Arr1(I, 0)
YY1 = Arr1(I, 1)

'Maak die punte groen

picOutline.PSet (XX1, YY1), QBColor(10)
Next I

'Boonste koördinate

For I = 0 To Arr2Size
XX1 = Arr2(I, 0)
YY1 = Arr2(I, 1)

'Maak die punte groen

picOutline.PSet (XX1, YY1), QBColor(10)
Next I

cmdCalculate.Enabled = True

End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)

**'Roetine wanneer die ontsnapsleutel op die sleutelbord gedruk
'word. Om die bepaling van die randbeeld of die "grayscale" beeld te
'stop**

If KeyAscii = vbKeyEscape Then
StopGray = True
StopOutline = True
End If

End Sub

Private Sub Form_Load()

'Hierdie kode word uitgevoer wanneer die program laai

mnuManual.Checked = True
StatusBar1.Panels(1).Text = "Manual Mode"

cmdCTOutline.Enabled = False
cmdGetPoints.Enabled = False
cmdShow.Enabled = False
cmdCalculate.Enabled = False
cmdRedoOutline.Enabled = False

SSTab1.Tab = 0

lblTestTolerance.Caption = vsbTestTolerance.Value

vsbTolerance.Value = vsbTestTolerance.Value
lblTolerance.Caption = vsbTolerance.Value

'Maak die .ini-lêer oop

Open App.Path & "\Settings.ini" For Input As #1

'Lees die gestoorde konstantes na veranderlikes

Line Input #1, LastDrive
Line Input #1, LastDir
Line Input #1, RatioString

Line Input #1, Unit
Line Input #1, ToleranceString

'Maak die .ini-lêer toe

Close #1

Ratio = Round(CSng(RatioString), 2)
Tolerance = CInt(ToleranceString)
vsbTestTolerance = Tolerance
vsbTolerance = Tolerance

db = App.Path & "\Coordinates.mdb"
Set Maindb = OpenDatabase(db)
rs = "Select * from Lookup"
Set RSlookup = Maindb.OpenRecordset(rs)

LargestdbLength = 0
SmallestdbLength = 2000

With RSLookup

.MoveLast

.MoveFirst

For I = 1 To .RecordCount

If !Length < SmallestdbLength Then

SmallestdbLength = !Length

End If

If !Length > LargestdbLength Then

LargestdbLength = !Length

End If

.MoveNext

Next I

dbResolution = Round(((LargestdbLength - SmallestdbLength) /
.RecordCount), 0)

End With

End Sub

Private Sub Form_Terminate()

'Maak die .ini-lêer oop

Open App.Path & "\Settings.ini" For Output As #1

RatioString = CStr(Ratio)

ToleranceString = CStr(Tolerance)

'Stoor die veranderlikes na konstantes

Print #1, LastDrive

Print #1, LastDir

Print #1, RatioString

Print #1, Unit

Print #1, ToleranceString

'Maak die .ini-lêer toe

Close #1

End Sub

Private Sub hsbColour_Change()



**'Roetine wanneer die waarde van die horisontale skuiwer
'van die kleurbeeld verander**

picColour.Left = -hsbColour.Value

End Sub

Private Sub hsbColour_Scroll()

**'Roetine wanneer die horisontale skuiwer van die
'kleurbeeld geskuif word**

picColour.Left = -hsbColour.Value

End Sub

Private Sub hsbGray_Change()

**'Roetine wanneer die waarde van die horisontale skuiwer
'van die grysvlakbeeld verander**

picGray.Left = -hsbGray.Value

End Sub

Private Sub hsbGray_Scroll()

**'Roetine wanneer die horisontale skuiwer van die
'grysvlakbeeld geskuif word**

picGray.Left = -hsbGray.Value

End Sub

Private Sub hsbOutline_Change()

**'Roetine wanneer die waarde van die horisontale skuiwer
'van die randbeeld verander**

picOutline.Left = -hsbOutline.Value

End Sub

Private Sub hsbOutline_Scroll()

**'Roetine wanneer die horisontale skuiwer van die randbeeld
'geskuif word**

picOutline.Left = -hsbOutline.Value

End Sub

Private Sub mnuAbout_Click()

**'Wys die "About"-vorm wanneer die "Help/About"-opsie op die
'keuselys gekies word**
frmAbout.Show

End Sub

Private Sub mnuExit_Click()

**'Termineer die program wanneer die "File/Exit" opsie op die
'keuselys gekies word**
End

End Sub

Private Sub mnuLookup_Click()

**'Vertoon die tabelvenster wanneer die Opsoektabel op die keuselys
'gekies word**
frmLookup.Show

End Sub

Private Sub mnuManual_Click()

If mnuManual.Checked = True Then
mnuManual.Checked = False
cmdCTG.Caption = "Calculate"
StatusBar1.Panels(1).Text = "Automatic Mode"
Else
mnuManual.Checked = True
cmdCTG.Caption = "Convert To GrayScale"
StatusBar1.Panels(1).Text = "Manual Mode"
End If
End Sub

Private Sub picColour_Change()

'Indien die kleurprent verander, sal die roetine uitgevoer word
cmdCTOutline.Enabled = False
cmdGetPoints.Enabled = False
cmdShow.Enabled = False
cmdCalculate.Enabled = False

```
cmdRedoOutline.Enabled = raise  
cmdCalculate.Enabled = False
```

```
lblLength.Visible = False  
lblLengthVal.Visible = False  
lblWeight.Visible = False  
lblWeightVal.Visible = False
```

```
picGray.Cls  
picOutline.Cls
```

```
ResizeWindow
```

```
End Sub
```

```
-----  
Private Sub Timer1_Timer()
```

```
'Elke 500 ms sal die kode uitgevoer word. Dit word gebruik om die  
'statusboodschap te laat flikker  
Dim Temp As String
```

```
If StatusBar1.Panels(2).Text = Message Then  
StatusBar1.Panels(2).Text = ""  
Else  
StatusBar1.Panels(2).Text = Message  
End If
```

```
End Sub
```

```
-----  
Private Sub Timer2_Timer()
```

```
'Hierdie kode word 2 sekondes na die bepaling van die  
'grysvlakbeeld en die randbeeld uitgevoer om die statusboodschap  
'te verwyder  
StatusBar1.Panels(2).Text = ""  
StatusBar1.Panels(3).Text = ""  
Timer2.Enabled = False
```

```
End Sub
```

```
-----  
Private Sub vsbColour_Change()
```

```
'Roetine wanneer die waarde van die vertikale skuiwer  
'van die kleurprent verander  
picColour.Top = -vsbColour.Value
```

```
End Sub
```

Private Sub vsbColour_Scroll()

**'Roetine wanneer die vertikale skuiwer van die kleurprent geskuif
'word**

picColour.Top = -vsbColour.Value

End Sub

Private Sub vsbGray_Change()

**'Roetine wanneer die waarde van die vertikale skuiwer van die
'grysvlakbeeld verander**

picGray.Top = -vsbGray.Value

End Sub

Private Sub vsbGray_Scroll()

**'Roetine wanneer die vertikale skuiwer van die grysvlakbeeld
'geskuif word**

picGray.Top = -vsbGray.Value

End Sub

Private Sub vsbOutline_Change()

**'Roetine wanneer die waarde van die vertikale skuiwer van die
'randbeeld verander**

picOutline.Top = -vsbOutline.Value

End Sub

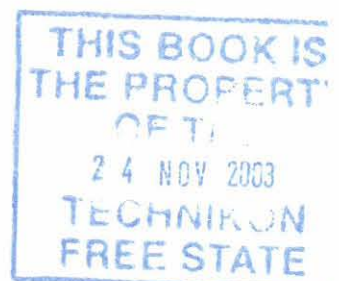
Private Sub vsbOutline_Scroll()

**'Roetine wanneer die vertikale skuiwer van die
'randbeeld geskuif word**

picOutline.Top = -vsbOutline.Value

End Sub

Private Sub vsbTestTolerance_Change()



**'Roetine wanneer die waarde van die vertikale skuiwer
'van die toetstoleransie verander**

lblTestTolerance.Caption = vsbTestTolerance.Value
vsbTolerance.Value = vsbTestTolerance.Value

Tolerance = vsbTestTolerance.Value

End Sub

Private Sub vsbTestTolerance_Scroll()

**'Roetine wanneer die vertikale skuiwer van die
'toetstoleransie geskuif word**

lblTestTolerance.Caption = vsbTestTolerance.Value
vsbTolerance.Value = vsbTestTolerance.Value

End Sub

Private Sub vsbTolerance_Change()

**'Roetine wanneer die waarde van die vertikale skuiwer
'van die toleransie verander**

vsbTestTolerance.Value = vsbTolerance.Value
lblTolerance.Caption = vsbTolerance.Value
Tolerance = vsbTolerance.Value

End Sub

Private Sub vsbTolerance_Scroll()

**'Roetine wanneer die vertikale skuiwer van die
'toleransie geskuif word**

vsbTestTolerance.Value = vsbTolerance.Value
lblTolerance.Caption = vsbTolerance.Value

End Sub

'Subroetine om die omskakeling na grysvlak te hanteer

Private Sub ConvertToGray()

Dim Col As Integer

Dim J As Integer

Dim Total As Double
Dim TotalDone As Long

**'Stel die grootte van die grysvlakbeeld dieselfde as die van die
'kleurprent**

picGray.Width = picColour.Width
picGray.Height = picColour.Height
picGray.Cls
Message = "Converting To GrayScale"

SSTab1.Tab = 1

DoEvents

StatusBar1.Panels(2).Text = Message
Timer1.Enabled = True
SSTab1.Enabled = False

Total = picColour.ScaleHeight * picColour.ScaleWidth

StopGray = False

'Werk deur die hele hoogte van die kleurprent
For I = 0 To picColour.ScaleHeight - 1

'Werk deur die hele wydte van die kleurprent
For J = 0 To picColour.ScaleWidth - 1
Col = GrayFromColor(picColour.Point(J, I))
picGray.PSet (J, I), RGB(Col, Col, Col)
Next J

TotalDone = ((I * (J / 10)) / Total) * 1000
StatusBar1.Panels(3).Text = TotalDone & " % Done"

DoEvents

If StopGray = True Then GoTo Stopp

Next I

Timer1.Enabled = False
Timer2.Enabled = True
cmdCTOutline.Enabled = True
SSTab1.Enabled = True
StatusBar1.Panels(2).Text = Message

GoTo SubEnd

Stopp:
Message = "Aborted"

SSTab1.Tab = 2

```
picOutline.Width = picColour.Width  
picOutlineBox.Width = picColourBox.Width  
hsbOutline.Visible = False  
hsbOutline.Value = 0  
lblTolOutline.Left = 3600  
vsbTolerance.Left = 3600  
lblLength.Left = 4920  
lblLengthVal.Left = 4920  
lblWeight.Left = 4920  
lblWeightVal.Left = 4920  
lblTolValOutline.Left = 4080  
lblTolerance.Left = 4440  
cmdRedoOutline.Left = 3600
```

SSTab1.Tab = 0

Else

If picColour.Width <= (Screen.Width - 4110) / Screen.TwipsPerPixelX
Then

```
picColourBox.Width = picColour.Width * Screen.TwipsPerPixelX
```

```
frmMain.Width = 7005 + (picColourBox.Width - 2895)  
frmMain.Left = (Screen.Width - frmMain.Width) / 2  
SSTab1.Width = 7005 + (picColourBox.Width - 2895)
```

```
hsbColour.Visible = False  
hsbColour.Value = 0
```

'Grysvlaktabuleerder

```
SSTab1.Tab = 1  
picGray.Width = picColour.Width  
picGrayBox.Width = picColourBox.Width  
hsbGray.Visible = False  
hsbGray.Value = 0  
lblTolGray.Left = 3600 + (picColourBox.Width - 2895)  
vsbTestTolerance.Left = lblTolGray.Left  
lblTolValGray.Left = 4080 + (picColourBox.Width - 2895)  
lblTestTolerance.Left = 4440 + (picColourBox.Width - 2895)
```

'Raamwerktabuleerder

```
SSTab1.Tab = 2  
picOutline.Width = picColour.Width  
picOutlineBox.Width = picColourBox.Width  
hsbOutline.Visible = False  
hsbOutline.Value = 0  
lblTolOutline.Left = 3600 + (picColourBox.Width - 2895)  
vsbTolerance.Left = lblTolOutline.Left
```

```
cmdRedoOutline.Left = lblTolOutline.Left  
lblTolValOutline.Left = 4080 + (picColourBox.Width - 2895)  
lblTolerance.Left = 4440 + (picColourBox.Width - 2895)  
lblLength.Left = 4920 + (picColourBox.Width - 2895)  
lblLengthVal.Left = lblLength.Left  
lblWeight.Left = lblLength.Left
```

```
lblWeightVal.Left = lblLength.Left  
SSTab1.Tab = 0
```

```
Else  
DoEvents  
picColourBox.Width = Screen.Width - 4110  
frmMain.Width = Screen.Width  
frmMain.Left = 0
```

```
SSTab1.Width = frmMain.Width - 100
```

```
hsbColour.Visible = True  
hsbColour.Value = 0  
hsbColour.Width = picColourBox.Width  
hsbColour.Max = picColour.Width - (picColourBox.Width /  
Screen.TwipsPerPixelX)  
hsbColour.LargeChange = Round(hsbColour.Max / 5)  
vsbColour.Left = picColourBox.Left + picColourBox.Width
```

'Grysvlaktabuleerder

```
SSTab1.Tab = 1  
picGray.Width = picColour.Width  
picGrayBox.Width = picColourBox.Width  
hsbGray.Visible = True  
hsbGray.Value = 0  
hsbGray.Width = picColourBox.Width  
hsbGray.Max = hsbColour.Max  
hsbGray.LargeChange = hsbColour.LargeChange  
vsbGray.Left = picGrayBox.Left + picGrayBox.Width  
lblTolGray.Left = 3600 + (picColourBox.Width - 2895)  
vsbTestTolerance.Left = lblTolGray.Left  
lblTolValGray.Left = 4080 + (picColourBox.Width - 2895)  
lblTestTolerance.Left = 4440 + (picColourBox.Width - 2895)
```

'Raamwerktabuleerder

```
SSTab1.Tab = 2  
picOutline.Width = picColour.Width  
picOutlineBox.Width = picColourBox.Width  
hsbOutline.Visible = True  
hsbOutline.Value = 0  
hsbOutline.Width = picColourBox.Width  
hsbOutline.Max = hsbColour.Max  
hsbOutline.LargeChange = hsbColour.LargeChange  
vsbOutline.Left = picOutlineBox.Left + picOutlineBox.Width
```



```
lblTolOutline.Left = 3600 + (picColourBox.Width - 2895)
vsbTolerance.Left = lblTolOutline.Left
cmdRedoOutline.Left = lblTolOutline.Left
lblTolValOutline.Left = 4080 + (picColourBox.Width - 2895)
lblTolerance.Left = 4440 + (picColourBox.Width - 2895)
lblLength.Left = 4920 + (picColourBox.Width - 2895)
lblLengthVal.Left = 4920 + (picColourBox.Width - 2895)
lblWeight.Left = 4920 + (picColourBox.Width - 2895)
lblWeightVal.Left = 4920 + (picColourBox.Width - 2895)
```

```
SSTab1.Tab = 0
```

```
End If
```

```
End If
```

```
If picColour.Height <= 3615 / Screen.TwipsPerPixelY Then
```

```
frmMain.Height = 6000
frmMain.Top = (Screen.Height - frmMain.Height) / 2
SSTab1.Height = 5055
```

```
picColourBox.Height = picColour.Height * Screen.TwipsPerPixelY
vsbColour.Visible = False
vsbColour.Value = 0
cmdCTG.Top = 4440
cmdLoad.Top = cmdCTG.Top
```

```
'Grysvlaktabuleerder
```

```
SSTab1.Tab = 1
picGray.Height = picColour.Height
```

```
picGrayBox.Height = picColourBox.Height
vsbGray.Visible = False
vsbGray.Value = 0
cmdCTOutline.Top = cmdCTG.Top
```

```
'Raamwerktabuleerder
```

```
SSTab1.Tab = 2
picOutline.Height = picColour.Height
picOutlineBox.Height = picColourBox.Height
vsbOutline.Visible = False
vsbOutline.Value = 0
cmdGetPoints.Top = cmdCTG.Top
cmdShow.Top = cmdCTG.Top
cmdCalculate.Top = cmdCTG.Top
cmdCalibrate.Top = cmdCTG.Top
```

```
SSTab1.Tab = 0
```

```
Else
```




```
If picColour.Height <= (Screen.Height - 2385) / Screen.TwipsPerPixelY  
Then
```

```
picColourBox.Height = picColour.Height * Screen.TwipsPerPixelY
```

```
frmMain.Height = 6000 + (picColourBox.Height - 3615)
```

```
frmMain.Top = (Screen.Height - frmMain.Height) / 2
```

```
SSTab1.Height = 5055 + (picColourBox.Height - 3615)
```

```
vsbColour.Visible = False
```

```
vsbColour.Value = 0
```

```
cmdCTG.Top = 4440 + (picColourBox.Height - 3615)
```

```
cmdLoad.Top = cmdCTG.Top
```

'Grysvlaktabuleerder

```
SSTab1.Tab = 1
```

```
picGray.Height = picColour.Height
```

```
picGrayBox.Height = picColourBox.Height
```

```
vsbGray.Visible = False
```

```
vsbGray.Value = 0
```

```
cmdCTOutline.Top = cmdCTG.Top
```

'Raamwerktabuleerder

```
SSTab1.Tab = 2
```

```
picOutline.Height = picColour.Height
```

```
picOutlineBox.Height = picColourBox.Height
```

```
vsbOutline.Visible = False
```

```
vsbOutline.Value = 0
```

```
cmdGetPoints.Top = cmdCTG.Top
```

```
cmdShow.Top = cmdCTG.Top
```

```
cmdCalculate.Top = cmdCTG.Top
```

```
cmdCalibrate.Top = cmdCTG.Top
```

```
SSTab1.Tab = 0
```

```
Else
```

```
DoEvents
```

```
picColourBox.Height = Screen.Height - 2385
```

```
frmMain.Height = Screen.Height
```

```
frmMain.Top = 0
```

```
SSTab1.Height = 5055 + (picColourBox.Height - 3615)
```

```
vsbColour.Visible = True
```

```
vsbColour.Value = 0
```

```
vsbColour.Height = picColourBox.Height
```

```
hsbColour.Top = picColourBox.Top + picColourBox.Height
```

```
vsbColour.Max = picColour.Height - (picColourBox.Height /  
Screen.TwipsPerPixelY)
```

`vsbColour.LargeChange = Round(vsbColour.Max / 5)`

`cmdCTG.Top = 4440 + (picColourBox.Height - 3615)`

`cmdLoad.Top = cmdCTG.Top`

'Grysvlaktabuleerder

`SSTab1.Tab = 1`

`picGray.Height = picColour.Height`

`picGrayBox.Height = picColourBox.Height`

`vsbGray.Visible = True`

`vsbGray.Value = 0`

`vsbGray.Height = picColourBox.Height`

`vsbGray.Max = vsbColour.Max`

`vsbGray.LargeChange = vsbColour.LargeChange`

`hsbGray.Top = picGrayBox.Top + picGrayBox.Height`

`cmdCTOutline.Top = cmdCTG.Top`

'Raamwerktabuleerder

`SSTab1.Tab = 2`

`picOutline.Height = picColour.Height`

`picOutlineBox.Height = picColourBox.Height`

`vsbOutline.Visible = True`

`vsbOutline.Value = 0`

`vsbOutline.Height = picColourBox.Height`

`vsbOutline.Max = vsbColour.Max`

`vsbOutline.LargeChange = vsbColour.LargeChange`

`hsbOutline.Top = picOutlineBox.Top + picOutlineBox.Height`

`cmdGetPoints.Top = cmdCTG.Top`

`cmdShow.Top = cmdCTG.Top`

`cmdCalculate.Top = cmdCTG.Top`

`cmdCalibrate.Top = cmdCTG.Top`

`SSTab1.Tab = 0`

`End If`

`End If`

`End Sub`

`Public AX As Integer`

`Public AY As Integer`

`Public BX As Integer`

`Public BY As Integer`

`Public CX As Integer`

`Public CY As Integer`

Public DX As Integer
Public DY As Integer

Public Arr1Size As Integer
Public Arr2Size As Integer

Public Arr1() As Integer
Public Arr2() As Integer

Public LastDrive As String
Public LastDir As String
Public RatioString As String
Public Ratio As Single
Public Unit As String
Public Tolerance As Integer
Public ToleranceString As String

Public Length As Integer
Public Weight As Integer

Public LargestdbLength As Single
Public SmallestdbLength As Single
Public dbResolution As Integer

Public db As String
Public rs As String

Public Maindb As Database
Public rsCoord As Recordset
Public RSLookup As Recordset
Public I As Integer

Dim Weight1 As Single
Dim Weight2 As Single
Dim Length1 As Integer
Dim Length2 As Integer
Dim Diff As Integer

Public Sub CalcWeight()

If (Length * Ratio) < SmallestdbLength Or (Length * Ratio) >
LargestdbLength Then
result = MsgBox("The length calculated is not in the boundaries of the
lookup table" & Chr(13) & Chr(10) & "thus the weight can not be
calculated.", vbOKOnly, "Length overflow")

frmMain.lblWeight.Visible = False

frmMain.lblWeightVal.Visible = False
Exit Sub

End If

'Bepaal die gewig volgens die tabel af

db = App.Path & "\Coordinates.mdb"

Set Maindb = OpenDatabase(db)

rs = "Select * from Lookup where Length = " & CStr(Length)

Set RSLookup = Maindb.OpenRecordset(rs)

On Error GoTo Cont

With RSLookup

frmMain.lblWeightVal.Caption = !Weight

End With

Maindb.Close

frmMain.lblWeight.Visible = True

frmMain.lblWeightVal.Visible = True

Exit Sub

Cont:

rs = "Select * from Lookup where Length between "

rs = rs & CStr(Round((Length * Ratio) - (dbResolution * 2), 0))

rs = rs & " and "

rs = rs & CStr(Round((Length * Ratio) + (dbResolution * 2), 0))

db = App.Path & "\Coordinates.mdb"

Set Maindb = OpenDatabase(db)

Set RSLookup = Maindb.OpenRecordset(rs)

Diff = 1000

With RSLookup

.MoveLast

.MoveFirst

For I = 1 To .RecordCount

If ((Length * Ratio) - !Length) < Diff And ((Length * Ratio) - !Length) > 0

Then

Diff = (Length * Ratio) - !Length

Length1 = !Length

Weight1 = !Weight

End If

.MoveNext

Next I

```
.MoveFirst
Diff = -1000
```

```
For I = 1 To .RecordCount
If ((Length * Ratio) - !Length) > Diff And ((Length * Ratio) - !Length) < 0
Then
Diff = (Length * Ratio) - !Length
Length2 = !Length
Weight2 = !Weight
End If
.MoveNext
Next I
```

```
End With
```

```
Maindb.Close
```

```
frmMain.lblWeightVal.Caption = CStr(Round((((Length * Ratio) -
Length2) / (Length1 - Length2)) * (Weight1 - Weight2)) + Weight2, 2)) &
" g"
```

```
frmMain.lblWeight.Visible = True
frmMain.lblWeightVal.Visible = True
```

```
End Sub
```

```
-----
'laaivorm
Private Sub Command1_Click()
```

```
Unload Me
```

```
End Sub
```

```
-----
Private Sub Command2_Click()
```

```
Dim Temp As String
Dim result As String
```

```
On Error GoTo Err
```

```
Temp = Dir1.Path
Temp = Temp & "\"
Temp = Temp & File1.FileName
```

```
LastDrive = Left(Drive1.Drive, 2)
```

```
LastDir = Right(Dir1.Path, Len(Dir1.Path) - 3)
```

```
frmMain.picColour.Picture = LoadPicture(Temp)
```

Open App.Path & "\\Settings.ini" For Output As #1

RatioString = CStr(Ratio)
ToleranceString = CStr(Tolerance)

'Stoor die instellings in die korrekte volgorde

Print #1, LastDrive
Print #1, LastDir
Print #1, RatioString
Print #1, Unit
Print #1, ToleranceString

'Close the .ini file

Close #1

Unload Me

Exit Sub

Err:

result = MsgBox("Incorrect Choice", vbExclamation + vbOKOnly,
"Error")

End Sub

Private Sub Dir1_Change()

File1.Path = Dir1.Path

End Sub

Private Sub Drive1_Change()

Dir1.Path = Drive1.Drive

End Sub

Private Sub Form_Load()

On Error GoTo ErrHand

Drive1.Drive = LastDrive
Dir1.Path = LastDrive & "\\ " & LastDir

Exit Sub

ErrHand:
Dir1.Path = Drive1.Drive & ""

End Sub

'Opsoekvorm

Dim db As String
Dim rs As String
Dim Maindb As Database
Dim RSLookup As Recordset

Dim I As Integer
Dim Sellentxt As Integer
Dim Selstarttxt As Integer

Private Sub cmdAdd_Click()

grdLookup.Rows = grdLookup.Rows + 1

If grdLookup.Rows > 12 Then
grdLookup.TopRow = grdLookup.Rows - 10
End If

End Sub

Private Sub cmdCancel_Click()

Unload Me

End Sub

Private Sub cmdDelete_Click()

grdLookup.RemoveItem (grdLookup.Row)

ClearGrid

End Sub

```
-----  
Private Sub cmdSave_Click()  
  
db = App.Path & "\Coordinates.mdb"  
  
Set Maindb = OpenDatabase(db)  
  
Maindb.Execute "Delete * from Lookup"  
  
Maindb.Close  
  
DoEvents  
  
db = App.Path & "\Coordinates.mdb"  
  
Set Maindb = OpenDatabase(db)  
  
rs = "Select * from Lookup"  
  
Set RSLookup = Maindb.OpenRecordset(rs)  
  
grdLookup.Row = 1  
  
For I = 1 To grdLookup.Rows - 1  
  
    grdLookup.Row = I  
  
    With RSLookup  
  
        grdLookup.Col = 0  
        If grdLookup.Text = "" Then  
  
            GoTo WithEnd  
        Else  
            grdLookup.Col = 1  
            If grdLookup.Text = "" Then  
                GoTo WithEnd  
            End If  
        End If  
  
        .AddNew  
        grdLookup.Col = 0  
        !Length = CInt(grdLookup.Text)  
        grdLookup.Col = 1  
        !Weight = CSng(grdLookup.Text)  
        .Update  
  
    End With  
  
WithEnd:  
    End With  
  
End For  
  
End Sub
```

Next I

Maindb.Close

Call CalcWeight

Unload Me

End Sub

Private Sub Form_Load()

grdLookup.Row = 0
grdLookup.Col = 0
grdLookup.Text = "Length"

grdLookup.Col = 1
grdLookup.Text = "Weight"

db = App.Path & "\Coordinates.mdb"

Set Maindb = OpenDatabase(db)
rs = "Select * from Lookup"

Set RSLookup = Maindb.OpenRecordset(rs)

With RSLookup

.MoveLast

.MoveFirst

For I = 1 To .RecordCount
grdLookup.Rows = grdLookup.Rows + 1
grdLookup.Row = I
grdLookup.Col = 0
grdLookup.Text = !Length
grdLookup.Col = 1
grdLookup.Text = !Weight
.MoveNext
Next I

End With

Maindb.Close

'MainDb.Execute "Delete * from Coord"

'MainDb.Close

'DoEvents

'db = App.Path & "\Coordinates.mdb"

'Set MainDb = OpenDatabase(db)

'rs = "Select * from Coord"

'Set RSCoord = MainDb.OpenRecordset(rs)

End Sub

Private Sub grdLookup_MouseUp(Button As Integer, Shift As Integer, X
As Single, Y As Single)

Dim TempRow As Integer

If Button = vbRightButton Then

If Y > 240 And Y < 480 Then

grdLookup.Row = grdLookup.TopRow + 0

GoTo EXT

End If

If Y > 480 And Y < 720 Then

grdLookup.Row = grdLookup.TopRow + 1

GoTo EXT

End If

If Y > 720 And Y < 960 Then

grdLookup.Row = grdLookup.TopRow + 2

GoTo EXT

End If

If Y > 960 And Y < 1200 Then

grdLookup.Row = grdLookup.TopRow + 3

GoTo EXT

End If

If Y > 1200 And Y < 1440 Then

grdLookup.Row = grdLookup.TopRow + 4

GoTo EXT

End If

If Y > 1440 And Y < 1680 Then

grdLookup.Row = grdLookup.TopRow + 5

GoTo EXT

End If

If Y > 1680 And Y < 1920 Then
grdLookup.Row = grdLookup.TopRow + 6
GoTo EXT
End If

If Y > 1920 And Y < 2160 Then
grdLookup.Row = grdLookup.TopRow + 7
GoTo EXT
End If

If Y > 2160 And Y < 2400 Then
grdLookup.Row = grdLookup.TopRow + 8
GoTo EXT
End If

If Y > 2400 And Y < 2640 Then
grdLookup.Row = grdLookup.TopRow + 9
GoTo EXT

End If

If Y > 2640 And Y < 2880 Then
grdLookup.Row = grdLookup.TopRow + 10

GoTo EXT
End If

EXT:

TempRow = grdLookup.Row

ClearGrid

grdLookup.Row = TempRow
grdLookup.Col = 0
grdLookup.CellBackColor = vbRed
grdLookup.Col = 1
grdLookup.CellBackColor = vbRed

Exit Sub
Else

If grdLookup.Col = 0 Then

txtLength.Top = 1440 + ((grdLookup.Row - grdLookup.TopRow) * 240)
txtLength.Visible = True
txtLength.Text = grdLookup.Text

```
txtLength.SetFocus  
txtLength.SelStart = 0  
txtLength.SelLength = Len(txtLength.Text)
```

```
Else
```

```
If grdLookup.Col = 1 Then  
txtWeight.Top = 1440 + ((grdLookup.Row - grdLookup.TopRow) * 240)  
txtWeight.Visible = True  
txtWeight.Text = grdLookup.Text  
txtWeight.SetFocus
```

```
txtWeight.SelStart = 0  
txtWeight.SelLength = Len(txtWeight.Text)
```

```
End If  
End If  
End If
```

```
End Sub
```

```
-----  
Private Sub txtLength_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then  
grdLookup.Text = txtLength.Text  
txtLength.Visible = False  
Else  
DoEvents  
End If
```

```
End Sub
```

```
-----  
Private Sub txtLength_LostFocus()
```

```
'grdLookup.Text = txtLength.Text  
txtLength.Visible = False
```

```
End Sub
```

```
-----  
Private Sub txtWeight_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then  
grdLookup.Text = txtWeight.Text  
txtWeight.Visible = False  
Else  
DoEvents  
End If
```


End Sub

Private Sub txtWeight_LostFocus()

'grdLookup.Text = txtWeight.Text
txtWeight.Visible = False

End Sub

Private Sub ClearGrid()

Dim I As Integer

For I = 1 To grdLookup.Rows - 1

grdLookup.Row = I

grdLookup.Col = 0
grdLookup.CellBackColor = vbWhite
grdLookup.Col = 1
grdLookup.CellBackColor = vbWhite
Next I

End Sub

' **Instellingsvorm**

Private Sub cmdCancel_Click()

Unload Me

End Sub

Private Sub cmdSet_Click()

Dim Temp As Single

'**Maak die .ini-leër oop om die instellings te stoor.**
Open App.Path & "\Settings.ini" For Output As #1

If optMillimeters.Value = True Then

Unit = "mm"

Temp = CSng(txtLength.Text)

```
Ratio = (Temp / Length)  
Ratio = Round(Ratio, 2)  
RatioString = CStr(Ratio)
```

```
frmMain.lblLengthVal.Caption = (Length * Ratio) & " " & Unit
```

```
Else
```

```
Unit = "cm"
```

```
Temp = CSng(txtLength.Text)
```

```
Temp = Temp * 10  
Ratio = (Temp / Length)  
Ratio = Round(Ratio, 2)  
RatioString = CStr(Ratio)
```

```
frmMain.lblLengthVal.Caption = ((Length * Ratio) / 10) & " " & Unit  
End If
```

```
'Stoor die instellings in die korrekte volgorde
```

```
Print #1, LastDrive  
Print #1, LastDir  
Print #1, RatioString  
Print #1, Unit  
Print #1, ToleranceString
```

```
'Maak die .ini-leër toe
```

```
Close #1
```

```
Call CalcWeight
```

```
Unload Me
```

```
End Sub
```

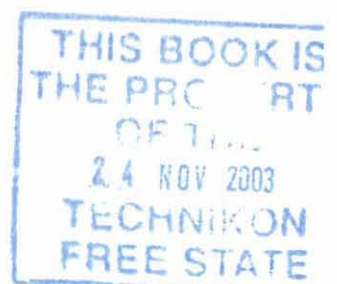
```
-----  
Private Sub Form_Load()
```

```
If Unit = "mm" Then  
optMillimeters.Value = True  
optCentimeters.Value = False  
txtLength.Text = (Length * Ratio)  
Else  
optCentimeters.Value = True  
optMillimeters.Value = False  
txtLength.Text = (Length * Ratio) / 10  
End If
```

```
End Sub
```

LITERATUURLYS

1. **Awcock, G.J. and Thomas, R.** Applied Image Processing. Houndmills. Macmillan Press LTD, 1995.
2. **Castelman, K.R.** Digital image processing. UK, Prentice Hall, 1996.
3. **Cornell, G.** Visual Basic from ground up, 1st ed, VSA Osborn/Mcgraw-Hill, 1997.
4. **Ewing, R.P. and Kaspar T.C.** Software for measuring root length from desktop scanner images. Agronomy Journal, Vol.89, 1997, pp. 932-940.
5. **Farrell, R.E., Walley, F.L., Lukey, A.P. and Germida, J.J.** Manual and digital line-intercept methods of measuring root length: A comparison. Agron, J. no.85, 1993, pp. 1233 – 1237.
6. **Gonzales, R.C. and Woods, R.E.** Digital image processing. New York, Addison-Wesley Publishing, 1992.
7. **Gonzales, R.C. and Wintz.P.** Digital image processing. 2nd ed. England, Addison-Wesley publishing, 1987.
8. **Harris, G.A. and Campbell, G.S.** Automated quantification of roots using a simple image analyzer. Agron, J. no.81, 1989, pp.935 – 938.
9. **Horn, B.K.P.** Robot vision. Cambridge, MIT Press, 1986.
10. **Jain, A.K.** Fundamentals of digital image processing. UK, Prentice-Hall, 1989.



11. **Lebowitz, R.J.** Digital image analysis measurement of root length and diameter. Environ. Exp. Bot., no.28, 1988, pp.267 – 273.
12. **LU, N.** Fractal image. San Diego USA, Academic Press, 1997.
13. **Marr, D.C. and Hildreth, E.** Theory of edge detection. London, Proceeding of the Royal Society of London, 1980.
14. **Marion, A.** An introduction to image processing. London, Chapman and Hall, 1991.
15. **Murphy, S.L. and Smucker, A.J.M.** Evaluation of video image analysis and line-intercept methods for measuring root systems of alfalfa and ryegrass. Agron, J. no.87, 1995, pp.865 – 868.
16. **Pan, W.L. and Bolton, R.P.** Root quantification by edge discrimination using desktop scanners. Agron, J. no.83, 1991, pp.1047 – 1052.
17. **Pappas, C.H. and Murray, W.H.** Visual Basic Programming with the windows API, 1st ed. VSA, Prentice-Hall, 1998.
18. **Pearson, D. E.** Image processing. UK, McGraw-Hill, 1991.
19. **Pratt, W.K.** Digital image processing. Canada, John Wiley & Sons Inc, 1978.
20. **Rosandich, R.G.** Intelligent visual inspection using artificial neural networks. 1ste. London , Chapman & Hall. 1997.
21. **Russ, J.C.** The Image Processing Handbook. 2nd ed. London, CRC Press, 1995.
22. **Scion Image**, 1998, <http://www.scioncorp.com>

23. **Travis, A.J., Murison, S.D. and Chesson, A.** Estimation of plant cell wall thickness and cell size by image skeletization. *J. Agric, Sci.*, no.120, 1993, pp.279 – 287.
24. **Webster, Webster's New Colledgeate Dictionary**, G. & C. Merriam Co., The Riverside Press, 1960.
25. **Young,I.T., Gerbrands,J.J. and Vliet,L.J.V.** Image processing fundamentals. Florida, CRC Press in cooperation with IEEE Press. 1998.
26. **Your fish.....With a ruler**, 2000,
<http://www.dec.state.ny.us/website/dfwmr/seasons/foe4cwgf.html>
27. **Zoon, F.C. and Tienderen, P.H.**. A rapid quantitative measurement of root length and root branching by microcomputer image analysis. Plant Soil, no. 126, 1990, pp.301 – 308.