

RIJKSINSTITUUT VOOR VISSERIJONDERZOEK

Haringkade 1 - Postbus 68 - 1970 AB IJmuiden - Tel.: +31 2550 64646

Afdeling: Technisch Onderzoek

Rapport: TO 89-11
Het FISHVOL-systeem: Bepaling van
visgewicht met behulp van beeldverwerking in
samenhang met gestructureerd licht .

Auteur(s): Berent Daan, Frank Storbeck

Project: .156
Projectleider: Drs. Frank Storbeck
Datum van verschijnen: December 1989

Inhoud:

1	Inleiding.....	2
2	Probleemstelling.....	2
3	Implementatie.....	3
4	Geometrie.....	4
5	Instelling van de hardware.....	5
6	Extractie van de laserlijn uit het videobeeld.....	6
7	Resultaten.....	8
8	Verdere ontwikkelingen.....	10
9	Conclusie.....	11
	Referenties.....	11
	Tabel I. - Meetgegevens van 28 gestripte schollen op basis van 10 metingen.....	12
	Appendix I - afleiding formule's geometrie.....	13
	Appendix II - Programmatuur.....	14

1 INLEIDING.

Als onderdeel van het EEG-project "*Integrated quality assurance of chilled food fish at sea*", nummer UP 1.67, heeft het RIVO onderzoek gedaan naar een alternatieve methode om het gewicht van platvis te bepalen aan boord van zeegaande kotterschepen. Hoofdaannemer van dit onderzoeksproject is het Technological Laboratory Ministry of Fisheries in Lingby (Denemarken). Naast het RIVO hebben ook Torry Research in Aberdeen (Schotland) en het Instituut voor Visproducten van TNO in IJmuiden een aandeel in dit onderzoeksproject. De door de EEG aan het RIVO gevraagde bijdrage laat zich als volgt omschrijven:

- alleen schol sorteren.
- alleen het gewicht bepalen.
- dient te resulteren in een commercieel haalbaar systeem.

Omdat weegsystemen niet erg betrouwbaar lijken te functioneren is gezocht naar een methode op basis van beeldverwerking in combinatie met gestructureerd licht. Eerst wordt volgens deze methode het volume van de vis bepaald en vervolgens wordt dit vertaald in een bijbehorende gewichtshoeveelheid. Tijdens de ontwikkelingsfase werd nauw samengewerkt met de firma CH&SS in Haarlem. Dit bedrijf stelde een transportband en apparatuur beschikbaar en verleende assistentie bij het ontwikkelen van de programmatuur. In ruil voor dit alles heeft het RIVO een presentatie gegeven van een prototype op een door CH&SS georganiseerd seminar op 12 september 1989¹. Deze overeenkomst maakte een vlotte start en snelle ontwikkeling mogelijk.

2 PROBLEEMSTELLING.

Op dit moment wordt aan boord van kotterschepen de vis op het oog op lengte gesorteerd. Deze methode is echter niet nauwkeurig genoeg en daarom wordt de vis op de wal nog eens gesorteerd op gewicht. Naast de extra arbeid die nodig is om de vis te hersorteren, treedt ook het kwaliteitsverlies op doordat de vis van het ijs moet worden gehaald.

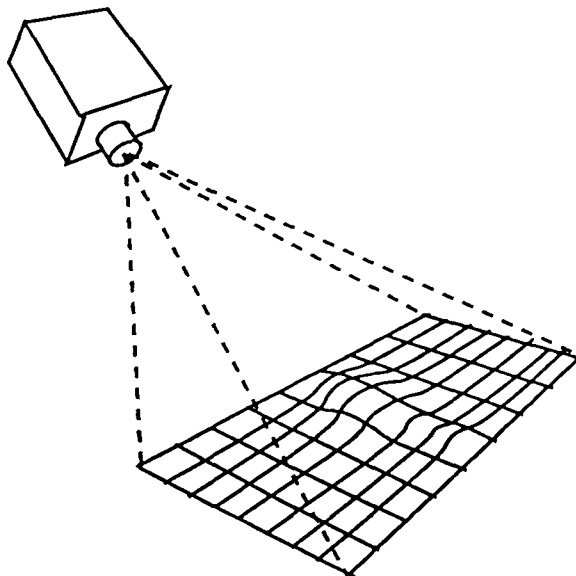


Fig. 1 - Schematische voorstelling principe structured light.

De geautomatiseerde vissortering maakt deel uit van een groter EEG-project om de kwaliteit van aangelande vis te verbeteren. Nevendoelstelling is de bemanning minder in-

¹ Om redenen van een langetermijn financiering van de RIVO-activiteiten heeft de directie besloten het ontwikkelde systeem te octrooieren. Alle rechten zijn derhalve voorbehouden.

tensief bezig te laten zijn met de verwerking van de vangst. Daardoor krijgen zij langere rusttijden. Verwacht mag worden dat daardoor minder ongevallen plaats zullen vinden ten gevolge van oververmoeidheid. Bijkomend voordeel van een geautomatiseerde vissortering is ook dat het mogelijk wordt om voortdurend bij te houden hoeveel er totaal aan vis gevangen is, iets wat tot nog toe door de schipper wordt geschat.

3 IMPLEMENTATIE.

Aan de bestaande methoden voor de gewichtsbepaling kleven nogal wat nadelen, of ze zijn niet geschikt voor toepassing op een bewegend schip, of ze zijn te kostbaar, te arbeidsintensief, etc. Er moest dus een nieuwe methode worden ontwikkeld die al deze nadelen niet had. Gekozen werd voor optische gewichtsbepaling met behulp van gestructureerd licht. Beeldverwerking biedt een aantal voordelen boven een mechanisch weegstelsel aan boord:

- kan compact uitgevoerd worden.
- geen bewegende delen en daardoor een grotere bedrijfszekerheid in een omgeving van zout water.
- alle componenten zijn gemakkelijk verkrijgbaar en vragen geen extra ontwikkelingskosten.

Door een patroon van licht op een voorwerp te projecteren en dit beeld onder een hoek met de lichtval optisch te registreren kan men informatie verkrijgen over de hoogte van het voorwerp langs die lijnen (zie fig. 1.). Met deze informatie kan het volume en daarmee het gewicht van het voorwerp worden bepaald, vooropgesteld natuurlijk dat het soortelijk gewicht van het voorwerp bekend is.

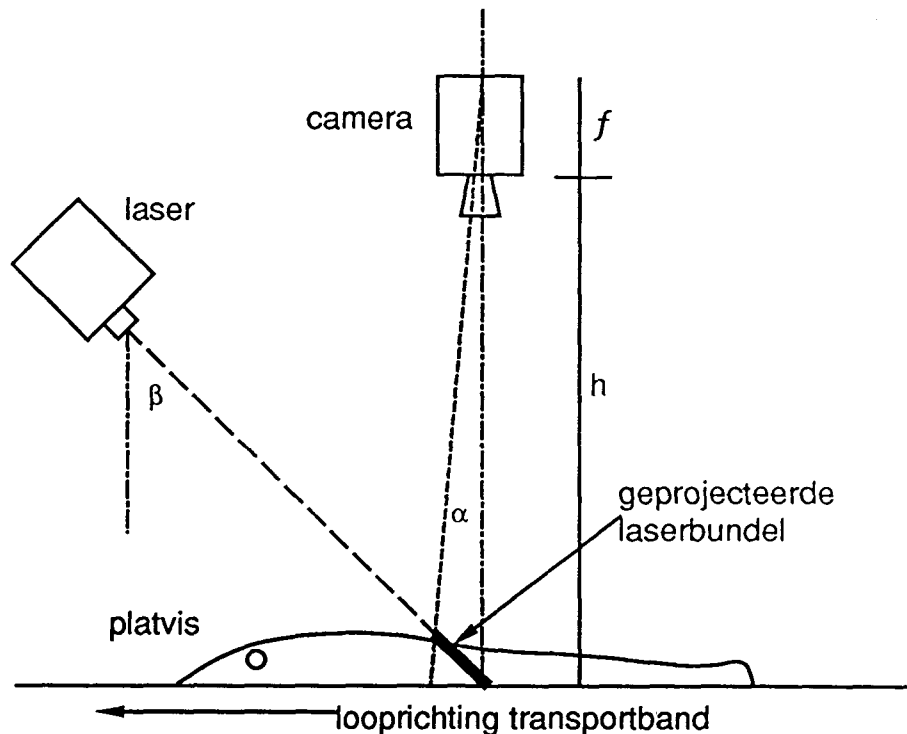


fig. 2 - Schematische doorsnede van de meetopstelling in de richting van de lopende band.

Gekozen is voor een opstelling met een laser en een videocamera, waarbij de laserbundel in een vlak verstrooid wordt door middel van een cilindrische lens. Gebruik van een laserbundel heeft een aantal voordelen:

- goed toepasbaar op zee, ook onder slechte weersomstandigheden.

- de bundel is overal even breed en daardoor treedt geen focuseringsprobleem op.
- het licht heeft een specifieke golflengte waardoor men met een *narrow-band pass filter* eventuele bijkomende reflecties van omgevingslicht weg kan filteren.
- naast de gewenste volume bepalingen kan het systeem uitgebreid worden met programmatuur voor soorthekening en lengtebepalingen

De laser wordt onder een hoek van vijfenvestig graden met de band aangebracht, zodanig dat de laserbundel een lijn dwars over de band projecteert. Loodrecht boven deze lijn bevindt zich de videocamera, waarvan de beeldlijnen weer loodrecht op de laserlijn staan en dus in de loopricting van de band (zie fig. 2). Voor de verwerking van het videobeeld is een drietal VME-kaarten uit de Datacube MAXVIDEO-serie aangeschaft, en daarbij een SUN-workstation als ontwikkelomgeving.

De eerste MAXVIDEO-kaart is de MAXSCAN. Deze was eerst alleen bedoeld voor de beeldaquisitie en -digitalisering, maar bleek later ook heel geschikt voor het bepalen van de plaats van de laserlijn. De tweede is de ROISTORE, geschikt voor het opslaan en bufferen van videodata (beeldinformatie). Tenslotte werd een MAXGRAPH aangeschaft voor de weergave van videobeelden en grafieken wat eigenlijk alleen nuttig is tijdens de ontwikkeling van de programmatuur.

4 GEOMETRIE.

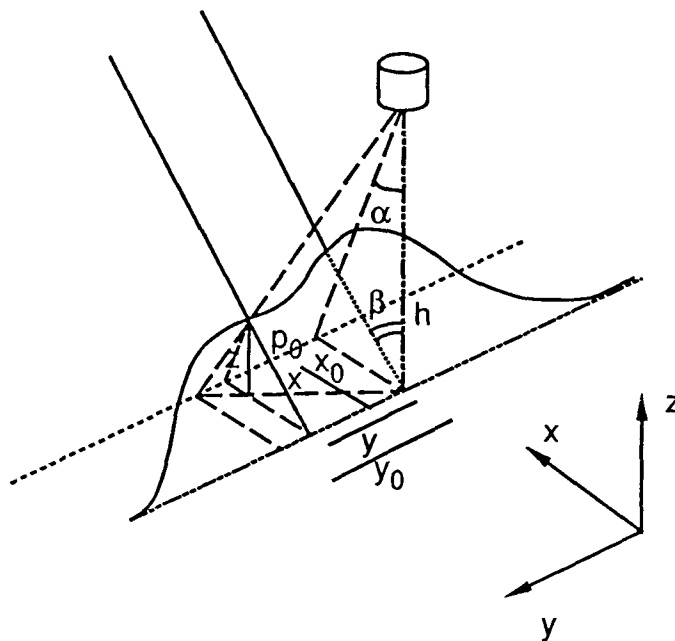


fig. 3 - Geometrie van de meetopstelling

Fig. 3 geeft een beeld van de meetopstelling. De x-as van het assenstelsel staat in de loopricting van de band, de y-as staat dwars op de loopricting in het vlak van de lopende band en de z-as staat loodrecht op de lopende band. De cameraleens bevindt zich op hoogte h. De laserbundel valt op het voorwerp onder een hoek β met de normaal. De gemeten afstand in de x-richting is de afstand x_0 en deze wordt vanuit de camera gezien onder een hoek α . p_0 is de lengte van het verlengde van de laserstraal in het inwendige van de vis. Er geldt nu:

$$p_0 = \frac{h x_0}{\cos(\beta) (h \operatorname{tg}(\beta) + x_0)}$$

Het volume wordt nu bepaald door de volgende integraal:

$$V = h \int_{t_0}^{t_e} \int_{y_1}^{y_r} \frac{x_0}{\cos(\beta) (h \operatorname{tg}(\beta) + x_0)} dy \cos(\beta) dt$$

waarin t_0 en t_e de tijdstippen zijn waarop de vis de laserbundel bereikt respectievelijk weer verlaat en y_1 en y_r de y -coördinaten van het begin en einde van de laser-lijn op de lopende band zijn.

In de experimentele opstelling is $\beta = 45^\circ$ dus $\operatorname{tg}(\beta) = 1$. Dit resulteert in de integraal :

$$V = h \int_{t_0}^{t_e} \int_{y_1}^{y_r} \frac{x_0}{(h + x_0)} dy dt$$

Omdat deze integraal benaderd wordt door voor discrete waarden langs de y -as de bijbehorende waarden voor x te bepalen dient nog een correctie voor de vertekening in de richting langs de y -as in beschouwing genomen te worden (zie fig. 3):

$$y = y_0 \left(1 - \frac{x_0}{h \operatorname{tg}(\beta) + x_0} \frac{h}{\sqrt{x_0^2 + h^2}} \right)$$

waarin y de positie is van projectie op de lopende band van het snijpunt van de laserstraal en het oppervlak van de vis. Nogmaals, $\beta = 45^\circ$ en de term $\frac{h}{\sqrt{x_0^2 + h^2}}$ is ongeveer 1,

dus:

$$y = y_0 \left(\frac{h}{h + x_0} \right)$$

Er zijn 512 beeldlijnen langs de y -as beschikbaar. Als s de snelheid van de lopende band is en Δt het tijdsinterval tussen twee beelden is, dan wordt bovenstaande integraal dus numeriek benaderd door:

$$V = s \Delta t h \sum_{n=0}^N \sum_{i=1}^{511} \frac{x_{i-1}}{h + x_{i-1}} \left(i \left(\frac{h}{h + x_i} \right) - (i - 1) \left(\frac{h}{h + x_{i-1}} \right) \right)$$

waarin N het aantal opgenomen lijnen is en x_i de gemeten afstand op lijn i .

Bij de toepassing van de zogenaamde dubbele runlength encoding is het echter niet nodig om voor alle beeldlijnen deze berekening uit te voeren, omdat de (discrete) waarden van x_i vaak voor enige lijnen achtereenvolgens constant blijft. Het principe van runlength encoding wordt later in dit stuk uiteen gezet.

5 INSTELLING VAN DE HARDWARE.

Aangezien de SUN nog niet beschikbaar was, is er gewerkt met een VME-bus krat met daarin de drie MAXVIDEO kaarten en een ELTEC EUROCOM 5 CPU-board met daarop een

MOTOROLA 68020 μ processor, dit alles draaiend onder OS-9, een UNIX-achtig besturingssysteem, toegespitst op real-time toepassingen.

De drie MAXVIDEO kaarten zijn naast elkaar in de VME-kraat geschoven en aan de voorkant met elkaar verbonden met data en timing kabels. Via de MAXBUS kabels vinden alle video-data transporten plaats, geheel gesynchroniseerd door de MAXBUS P3 en ROI timing bus.

Op de P3-timing bus staan drie kloksignalen (10, 20 en 40 Mhz) en twee video-synchronisatie signalen, een vertical-sync (H-sync) en een horizontal-sync (V-sync). Deze signalen worden door de MAXGRAPH gegenereerd volgens de standaard RS-170 / CCIR norm. Op deze P3 timing bus zijn alle drie de MAXVIDEO kaarten aangesloten.

De MAXSCAN en de ROISTORE zijn daarnaast ook nog verbonden door een ROI timing bus. Op deze bus staan programmeerbare H-sync en V-sync signalen. De ROI timing bus wordt gebruikt voor het verzenden van *Regions Of Interest* van de MAXSCAN naar de ROISTORE. In deze opstelling is de MAXSCAN ROI timing master. De MAXSCAN genereert dus de synchronisatie signalen op de ROI timing bus.

De MAXBUS data-poorten zijn als beschreven in fig. 4 door kabels met elkaar verbonden.

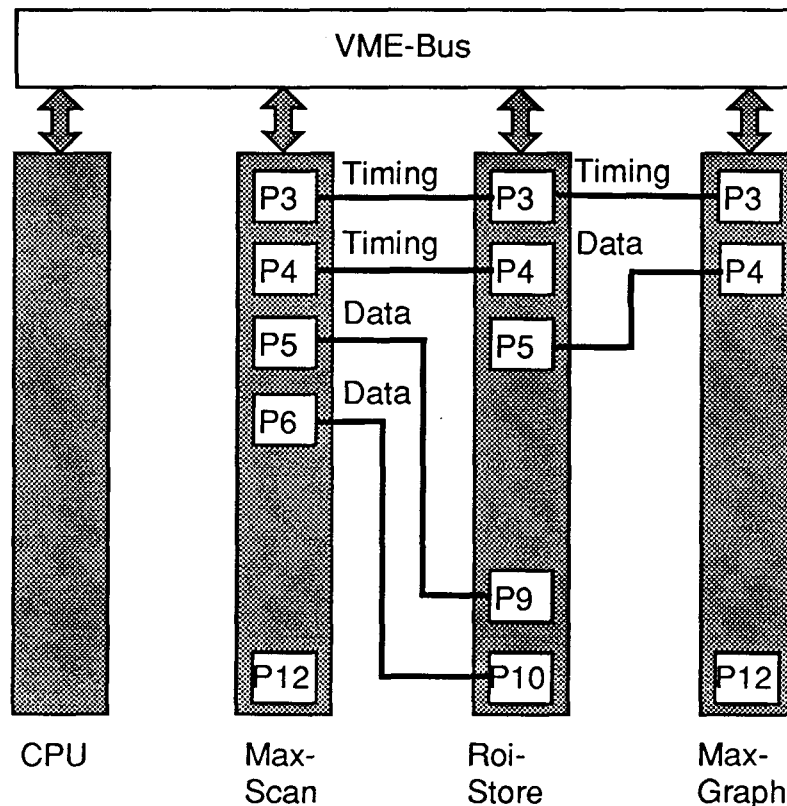


Fig. 4 - Interconnectie MAXVIDEO Boards.

Alle videodata worden in interlaced mode overgezonden, dat wil zeggen dat elk beeldje uit één frame bestaat en elk frame weer uit twee fields, een even en een oneven field. Deze fields bevatten respectievelijk de oneven en de even lijnen van het beeld, en worden na elkaar verzonden, telkens in paren. Voor de technische details over de werking van de verschillende borden wordt verwezen naar de MAXVIDEO handleidingen.

6 EXTRACTIE VAN DE LASERLIJN UIT HET VIDEOBEELD.

In eerste instantie werd er geen gebruik gemaakt van de mogelijkheid tot runlength encoding van de MAXSCAN. De MAXSCAN werd louter gebruikt voor het binnenhalen van het videobeeld en het doorzenden van dit beeld naar de ROISTORE. Elke keer als het programma dat op de CPU draaide daar opdracht toe gaf, werd een beeld in de ROISTORE

ingelezen. In de ROISTORE werd vervolgens het beeldgeheugen afgetast op zoek naar pixelwaarden boven een bepaalde clipwaarde, om zo de plaats van de laserlijn te vinden. Door sommatie werd de oppervlakte van de doorsnede bepaald en tegelijkertijd werd de gevonden lijn getekend in de overlay plane van de MAXGRAPH. Ondanks verscheidene optimalisatierondes bleek deze methode niet snel genoeg om de camerasnelheid van 25 beelden per seconde bij te kunnen houden.

De MAXSCAN biedt zoals genoemd de mogelijkheid tot *runlength encoding*. Dit is een coderingsmethode waarbij per beeldlijn niet de pixelwaarden worden overgezonden, maar de plaatsen op de beeldlijn waar de intensiteitswaarden van de pixels in een andere categorie vallen dan die van het voorgaande pixel. Bovendien wordt aangegeven in welke categorie die waarden dan liggen. Er kan gewerkt worden met vier verschillende categorieën die men kan definiëren in een *Look Up Table (LUT)* in de MAXSCAN. Aan elke pixelpositie worden twee bits toegevoegd waarmee de categorie waarin de nieuwe pixelwaarden vallen wordt aangegeven.

In de onderhavige toepassing wordt er met twee categorieën gewerkt, alle waarden onder een bepaalde clipwaarde behoren tot categorie 0 en alle waarden erboven tot categorie 1. Bovendien zijn we alleen geïnteresseerd in de eerste plaats van een verspringsing van categorie 0 naar categorie 1, dat is namelijk de plaats op de beeldlijn waar de laserlijn zich bevindt.

Deze codes worden in de ROISTORE in word-mode ingelezen, dat wil zeggen 16 bits parallel. Vervolgens zet het programma op de CPU deze codes weer gecodeerd in een array, namelijk alleen die nummers van beeldlijnen waar de plaats van de laserlijn verschilt van de plaats op de vorige beeldlijnen en de plaats van de laserlijn op die beeldlijn. Een soort tweede runlength encoding dus.

Nu krijgt de ROISTORE opdracht om het volgende beeld binnen te halen. Op deze wijze is de hoeveelheid te verwerken gegevens drastisch gereduceerd en kan de verwerking van de gegevens simultaan met het verzenden van data tussen MAXSCAN en ROISTORE geschieden.

Aan de hand van het array wordt de oppervlakte van de doorsnede bepaald, en eventueel geplot op de MAXGRAPH. In fig. 5 is weergegeven hoe een en ander zich in de tijd afspeelt.

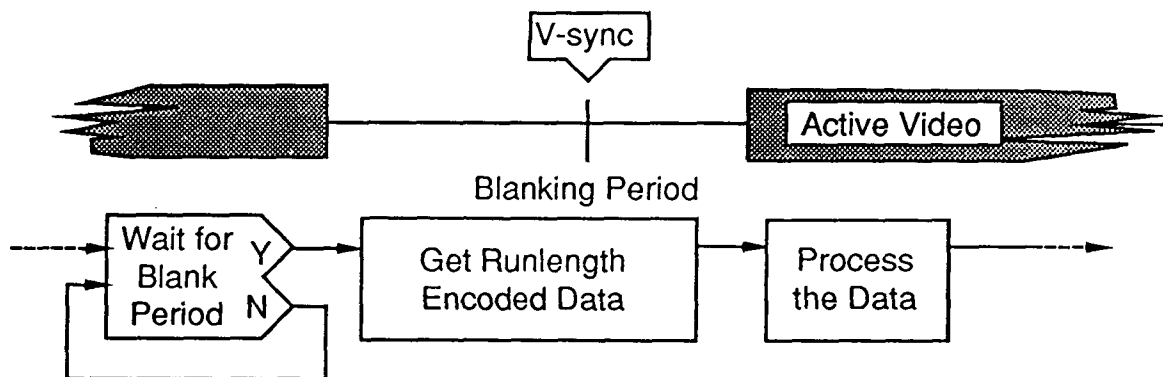


Fig. 5 - Verloop van het proces in de tijd.

Op deze wijze is het mogelijk om de snelheid van 25 beeldjes per seconde bij te houden met de EUROCOM 5 CPU. Daarvoor moest dan wel de real-time plotfunctie worden uitgeschakeld, want deze vertraagde het geheel dusdanig dat af en toe een plaatje werd overgeslagen. Om toch een beeld omtrent de betrouwbaarheid en werking van het systeem te verkrijgen werd de plotfunctie daarom pas uitgevoerd nadat de vis de laserlijn volledig gepasseerd is (fig. 6).

Een ander probleem was dat bij sterk stijgende of dalende vlakken de laserlijn heen en weer leek te springen tussen twee waarden. Dit verschijnsel was het gevolg van de interlacing van het videobeeld. De camera zendt eerst de oneven lijnen en daarna de even lij-

nen van elk beeldje over, waardoor de oneven lijnen eigenlijk horen bij een vroeger tijdstip dan de even lijnen.

Door de even en oneven lijnen apart in te lezen werd een vrij continu verloop van de laserlijn zichtbaar, bovendien kon de nauwkeurigheid worden vergroot omdat op deze wijze met een snelheid van 50 beeldjes per seconde wordt gemeten.

7 RESULTATEN.

Afhankelijk van de positie van het object wordt aan het begin en het einde van het object een doorsnede te veel of te weinig gemeten. Dit houdt dan al een foutgrote in die gelijk is aan de som van de eerste en laatste doorsneden. Bij platvis hebben we te maken met een geleidelijk toenemende doorsnede, waardoor deze fout zeer beperkt blijft.

Verder treedt een probleem op wanneer een object gedeeltelijk boven de band zweeft, daar de ruimte onder het object gewoon meetelt voor het volume (zie fig. 6). Weer geldt voor platvis dat ook hier de fout beperkt blijft omdat de vis zo plat op de band ligt dat de ruimte eronder relatief klein is. Ligt de vis echter op z'n rug (of 'bolle' zijde), dan vindt men meteen een structurele afwijking in de metingen terug, zoals in fig. 7 is te zien.

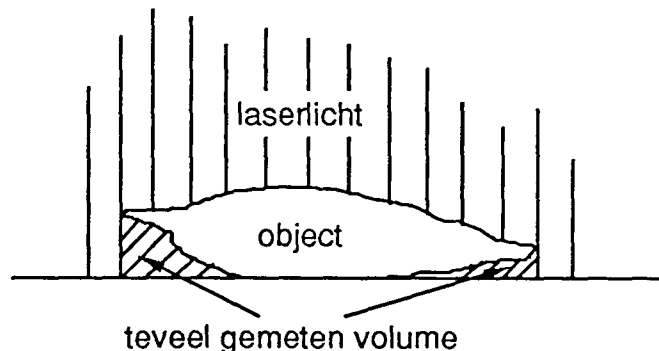


Fig. 6 - Te veel gemeten volume door ruimte onder het object.

De metingen in figuur 7 zijn gebaseerd op gestripte schol. 28 Gestripte schollen werden gewogen met een nauwkeurigheid van 1 gram en vervolgens werd hun volume bepaald met een nauwkeurigheid van ongeveer 5 ml. [1]. De regressielijn geeft aan dat het soortelijk gewicht van gestripte schol inderdaad een constante is (1.06 g/ml).

Fig. 8 geeft het verband tussen het volgens de in dit rapport beschreven methode gemeten volume en het werkelijke volume van de vis. De 28 schollen werden in een willekeurige volgorde ieder 10 maal door het systeem gemeten. De resultaten van deze proef zijn weergegeven in Tabel I. Uit deze tabel en figuur 8 blijkt zonneklaar dat de methode goed voldoet. Op de verticale as van in de figuur zijn de metingen van het FISHVOL-systeem terug te vinden met de bijbehorende standaard afwijkingen. De correlatiecoëfficiënt van de regressielijn bedraagt bijna één. De standaardafwijking bij enkele exemplaren van de 28 schollen is opvallend groot. De oorzaak hiervan ligt in het feit dat gebruik is gemaakt van schol die diepgevroren werd aangevoerd en ook na ontthooing nog rigor mortis verschijnselen vertoonden, d.w.z. omhoog staande koppen en vinnen.

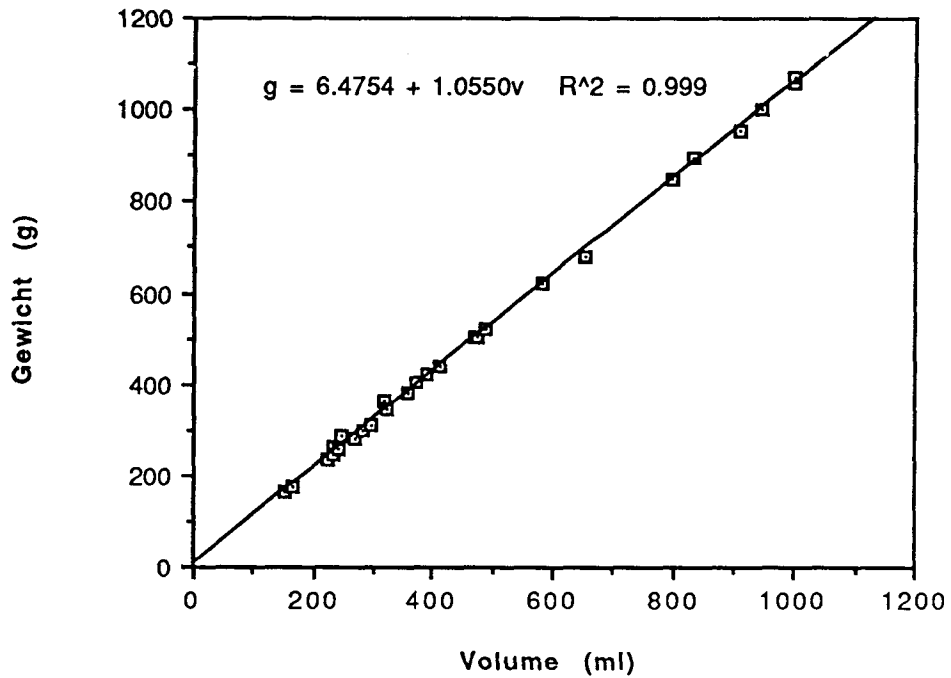


Fig. 7 - Gemeten gewicht als functie van werkelijke volume.

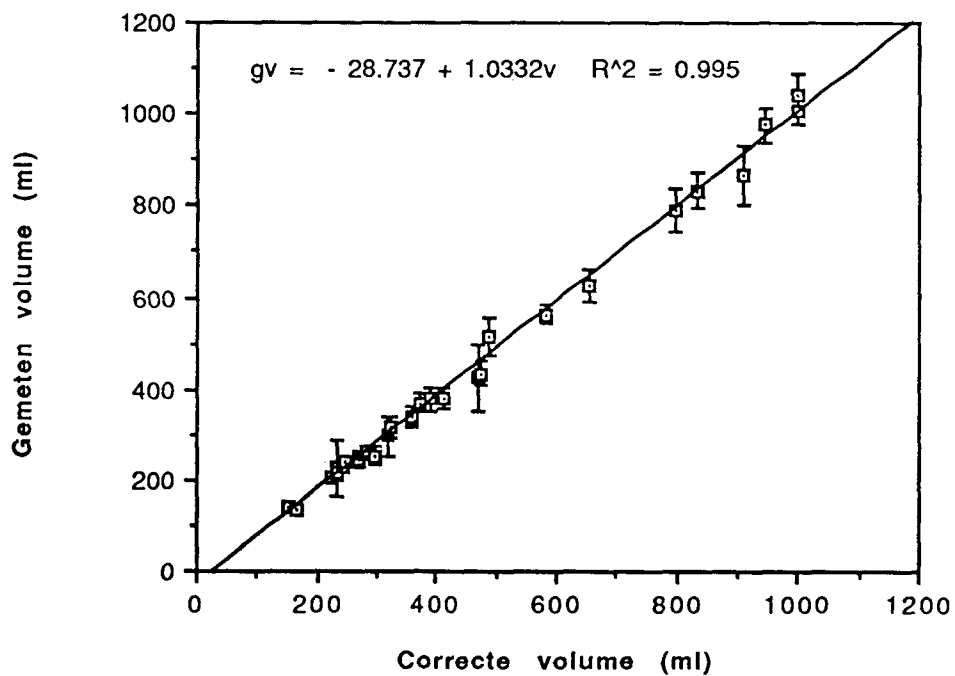


Fig. 8 - Gemeten volume van 28 gestripte schollen als functie van werkelijk volume met de standaardafwijkingen over een tiental metingen.

De zich soepel gedragende dode vissen vertoonden een veel consistentere beeld bij de metingen. Verwacht mag worden dat de laatste situatie aan boord van schepen gebruikelijker is.

Aan de hand van de regressie lijn werd een gecorrigeerde waarde voor het volume berekend. Op deze wijze is een empirische benadering verkregen van het volume zoals deze door het systeem afgeleverd zou moeten worden. Deze resultaten werden vervolgens omgezet in een schatting van het gewicht aan de hand van de volume gewichts relatie. Deze zijn in tabel I weergegeven, samen de optredende fouten. Hier blijkt een grote relatieve fout steeds gepaard te gaan met een grote standaardafwijking in de 10 gewichtsbepalingen.

Bij de metingen aan rondvis kwam ook een duidelijk lineair verband naar voren tussen het gemeten volume en het gewicht van de vis. Hieruit valt af te leiden dat het teveel gemeten volume lineair toeneemt met het werkelijke volume van de vis. Het probleem is echter dat op deze wijze voor elke vissoort een specifieke gewichtbepaling moet worden gehanteerd, afhankelijk van de mate van 'rondheid' van de vissoort.

Om het gewicht van verschillende vissoorten door elkaar te kunnen berekenen zal het dus nodig zijn eerst te bepalen met wat voor vis men te maken heeft, liefst met de gegevens die met de volumebepaling verzameld worden, zodat geen extra metingen behoeven te worden uitgevoerd.

Uit de resultaten moge blijken dat het systeem voor platvis goed werkt. Voor wat betreft de eis van een "commerciële haalbaar systeem" valt het volgende op te merken. Een systeem dat bestaat uit:

- een camera,
- een laserbron,
- een VME-krat met daarin een MAXSCAN, ROISTORE en dedicated CPU-kaart,
- de nodige software,

zal ongeveer kf 40 gaan kosten. Bij deze schatting is al rekening gehouden met de nodige winstmarge voor de leverancier. Dit is gezien de huidige investering van zo'n Mf 8 voor een nieuwe kottor zeker een acceptabele investering.

8 VERDERE ONTWIKKELINGEN.

Uit het voorgaande blijkt dat voor een algemeen toepasbaar systeem de soortherkenning van essentieel belang is. Het belangrijkste kenmerk wat voorhanden is en voor de herkenning kan worden gebruikt is de vorm of contour van de vis. Deze wordt verkregen door de punten waarop de laserlijn boven het nulniveau (de bandhoogte) stijgt en er weer op terug komt aaneen te schakelen tot een gesloten vorm. Er zijn vervolgens een aantal mogelijkheden om aan de hand van die contour te beslissen wat de soort van de vis is. Het systeem geeft bovendien extra informatie omtrent de dikte van de vis. Door een aantal geometrische parameters en hun onderlinge verhoudingen te berekenen kan waarschijnlijk een redelijk aantal soorten worden onderscheiden. Hierbij denken we in de eerste plaats aan de verhouding lengte / breedte. Om deze parameters te kunnen berekenen zal eerst de oriëntatie van de contour moeten worden bepaald.

Een nadeel van een dergelijke herkenningwijze is dat veel informatie over de vorm verloren gaat.

Een ander probleem dat zich voor kan doen is dat de gestripte vis nog een stuip trekking kan vertonen, terwijl hij of zij zich in de laserbundel bevindt. Dit zou men kunnen oplossen door twee evenwijdige laserbundels over de vis te projecteren en vervolgens te eisen dat de resultaten van beide bundels een identiek resultaat moeten opleveren. Met de runlength encoding techniek zoals hier beschreven is dat zeer eenvoudig te realiseren.

Het kan zijn dat de verwerkingssnelheid aanzienlijk opgevoerd moet worden. Dit kan met realiseren door twee vissen naast elkaar, maar wel goed gescheiden, te verwerken. Daarmee wordt een verdubbeling van de capaciteit bereikt.

9 CONCLUSIE.

Voor zover het de volumebepaling van platvis betreft voldoet het systeem. Ook voor rondvis metingen lijkt het systeem bruikbaar. Door de programmatuur te porten naar bijvoorbeeld een MAXPROC kan deze samen met de MAXSCAN de gehele functie van de volumebepaling uitvoeren. De MAXPROC is een VME-kaart met daarop een 68030 μ processor en twee beeldgeheugens en die in staat geacht moet worden voormeld algoritme in real-time uit te voeren. Het enige wat dan nog nodig is is een sorteringssysteem dat zorg draagt voor een juiste verwerking van de vis op basis van de verkregen meetgegevens.

REFERENTIES

- Anonymus, MaxScan user's manual, DataCube Inc., Peabody, Ma. USA, doc. nr. UM0053-2.0.
- Anonymus, RoiStore user's manual, DataCube Inc., Peabody, Ma. USA, doc. nr. UM0054-2.0.
- Anonymus, MaxGraph videographics generation module, DataCube Inc., Peabody, Ma. USA, doc. nr. UM0055-2.1.
- Anonymus, MaxWare Installation and software manual, DataCube Inc., Peabody, Ma. USA, doc. nr. UM0060-2.0.
- Scherle, W.; A simple method for volumetry of organs in quantitative stereology; in Mikroskopie, Bd. 26 (1970), S. 57-60.

TABEL I. - MEETGEGEVENS VAN 28 GESTRIPTTE SCHOLLEN OP BASIS VAN 10 METINGEN

Vis	Gewicht (g)	Volume (ml)	Gemiddeld volume (ml)	Stand. afw. (ml)	Gecorr. volume (ml)	Gemeten gewicht (g)	Absol. fout (g)	Relat. Fout (%)
27	164	152	138	8	163	179	15	8.9*
4	173	164	137	11	162	177	4	2.5
13	236	222	204	13	227	245	9	4.0
14	246	234	216	15	238	258	12	4.7*
26	257	240	226	11	248	268	11	4.2*
15	263	232	226	62	247	267	4	1.7
1	283	267	245	16	266	287	4	1.3
20	286	245	240	9	261	282	-4	-1.3
11	300	281	261	16	281	303	3	1.0
10	310	295	253	20	273	295	-15	-4.9*
2	347	323	314	23	332	357	10	2.8
7	379	360	337	13	355	381	2	0.5
6	382	358	340	22	358	384	2	0.5
18	404	370	371	20	387	415	11	2.7
3	420	390	378	25	394	422	2	0.5
16	439	413	382	23	398	426	-13	-3.0
19	503	470	425	71	439	470	-33	-6.6*
21	503	474	435	28	449	481	-22	-4.5*
22	520	488	513	41	524	559	39	7.5*
5	677	655	628	36	635	676	-1	-0.1
8	622	580	563	20	572	610	-12	-1.9
23	850	795	790	47	791	841	-9	-1.0
24	895	835	833	37	832	884	-11	-1.2
25	955	910	867	66	865	919	-36	-3.8
17	1003	945	976	39	971	1030	27	2.7
9	1062	998	1041	46	1033	1096	34	3.2
28	1069	1000	1004	28	997	1058	-11	-1.0

APPENDIX I - AFLEIDING FORMULE'S GEOMETRIE.

(Zie fig. 3 voor de betekenis van de symbolen.)

$$\frac{z}{h} = \frac{x_0 - z \operatorname{tg}(\beta)}{x_0} \Rightarrow$$

$$z x_0 = h x_0 - z h \operatorname{tg}(\beta) \Rightarrow$$

$$z (x_0 - h \operatorname{tg}(\beta)) = h x_0 \Rightarrow$$

$$z = h \frac{x_0}{h \operatorname{tg}(\beta) + x_0} \Rightarrow$$

$$p_0 = \frac{h x_0}{\cos(\beta) (h \operatorname{tg}(\beta) + x_0)}$$

$$\frac{z}{h} = \frac{y_0 - y}{y_0} \Rightarrow$$

$$z y_0 = (y_0 - y) h \Rightarrow$$

$$y h = y_0 (h - z) \Rightarrow$$

$$y = y_0 \left(1 - \frac{z}{h}\right) = y_0 \left(1 - \frac{x_0}{h \operatorname{tg}(\beta) + x_0}\right) \Rightarrow$$

$$y = y_0 \left(1 - \frac{x_0}{h \operatorname{tg}(\beta) + x_0}\right)$$

APPENDIX II - PROGRAMMATUUR.

De volgende gedeelten uit de broncode werken op een SUN 4/330 met MAXSCAN, ROISTORE en MAXGRAPH. De gekozen gedeelten hebben betrekking op de in dit rapport aangegeven methoden. Alles wat te maken heeft met de instelling van de apparatuur en de user interface is weggelaten. Voor alle code hier geldt:

© 1989, RIJKSINSTITUUT VOOR VISSERIJONDERZOEK, IJmuiden.

```
Callibrate()
{
    UnsignedWord *lineBasePtr, runLengthCode;
    int cumLength, count, y;

    cumLength = 0;
    count = 0;
    lineBasePtr = rsBase;
        /* rsBase is the baseadres of the RoiStore */
    rsWFire(rsfd);
    rsWHalt(rsfd);

    for (y = MinY; y < MaxY; y++) {
        runLengthCode = *lineBasePtr;
        if (!(runLengthCode & 0x2000)) {
            cumLength += runLengthCode & 0xffff;
            count++;
        }
        lineBasePtr += LineLength;
    }
    xZero = (cumLength / count) + 1;
    printf("Laser position: %d\n", xZero);
    rsWSVMd(rsfd, RS_VCONT);
    rsWFire(rsfd);
}

ScanFish()
{
    int z, area, cumArea, count, rllEven[1000], rllOdd[1000];

    do {
        z = 0;
        cumArea = 0;
        do {
            rsWHalt(rsfd);
            rsWFire(rsfd);
            GetRunLengthEncoding(rllEven, rllOdd);
            area = ScanSurface(rllEven);
            area += ScanSurface(rllOdd);
        } while(area < minimalArea);
        do {
            cumArea += area;
            PlotRunLengthEncoding(rllEven, z);
            PlotRunLengthEncoding(rllOdd, z + 1);
            rsWHalt(rsfd);
            rsWFire(rsfd);
            GetRunLengthEncoding(rllEven, rllOdd);
            area = ScanSurface(rllEven);
            area += ScanSurface(rllOdd);
            z++;z++;
        } while (area > minimalArea);
        printf("Objectnr. : %d \t Volume: %d cm3\t Length : %4.2f cm\n",
            fishCount, cumArea / Factor, z * 0.44);
    } while (1);
}
```

```

GetRunLengthEncoding(rllEven, rllOdd)
int *rllEven, *rllOdd;
{
    register int *rll, y, x, lastX;
    register UnsignedWord *linePtr;

    rll = rllEven;
    linePtr = rsBase;
    x = xZero;
    lastX = x;
    *(rll++) = MinY;
    *(rll++) = 0;
    for (y = MinY; y < MaxY; y += 2) {
        if (!(*linePtr & 0x2000)) {
            x = *linePtr & 0xfff;
            if (x != lastX) {
                lastX = x;
                *(rll++) = y;
                *(rll++) = xZero - x;
            }
        }
        linePtr += LineLength * 2;
    }
    *(rll++) = MaxY;
    *rll = 0;

    rll = rllOdd;
    linePtr = rsBase + LineLength;
    x = xZero;
    lastX = x;
    *(rll++) = MinY;
    *(rll++) = 0;
    for (y = MinY + 1; y < MaxY; y += 2) {
        if (!(*linePtr & 0x2000)) {
            x = *linePtr & 0xfff;
            if (x != lastX) {
                lastX = x;
                *(rll++) = y;
                *(rll++) = xZero - x;
            }
        }
        linePtr += LineLength * 2;
    }
    *(rll++) = MaxY;
    *rll = 0;
}

int ScanSurface(rll)
register int *rll;
{
    register int thisX, thisY, lastX, lastY;
    int area = 0;
    register float thisFactor;

    thisY = *(rll++);
    thisX = *(rll++);
    thisFactor = CameraHeight/(thisX + CameraHeight);
    thisY = 256 + (thisY - 256) * thisFactor;
    do {
        lastY = thisY;
        lastX = thisX * thisFactor;
        thisY = *(rll++);
        thisX = *(rll++);
        thisFactor = CameraHeight/(thisX + CameraHeight);
        thisY = 256 + (thisY - 256) * thisFactor;
        cumArea += (thisY - lastY) * lastX;
    }
}

```

```

    } while (*rll != MaxY);
    return(cumArea);
}

PlotRunLengthEncoding(rll, z)
register int rll[];
int z;
{
    register int y, hz, vz, iAx, iAy;
    hz = z*HZF;
    vz = 200 + z*VZF;

    iAx = hz + *rll/HYF;
    iAy = vz - *(rll+)/VYF;
    iAy += *(rll++) * XF;
    ghAmove(ghfd, iAx, iAy);
    do {
        if ((*rll - *(rll - 2)) > 5)
            if (((*(rll + 1) - *(rll - 1)) / 5) != 0) {
                rll -= 2;
                *(rll) = *(rll + 2);
            }
        y = *(rll++);
        iAx = hz + y/HYF;
        iAy = vz - y/VYF + *(rll++) * XF;
        ghAline(ghfd, GH_AMDO, GH_CMDO, GH_REPLACE, iAx, iAy);
        ghAmove(ghfd, iAx, iAy);
    } while (y != MaxY);
}

```