



Opdrachtgever: RIKZ

Het samenstellen van contouren op basis van raaigegevens

**- Achtergronden en
Bedieningshandleiding -**

Modelit
Rotterdamse Rijkweg 126
3042 AS Rotterdam
Telefoon +31 10 4623621



info@modelit.nl
www.modelit.nl

Opdrachtgever: RIKZ

Het samenstellen van contouren op basis van raaigegevens

**- Achtergronden en
Bedieningshandleiding -**

Datum 30 september 2004
Herzien 29 Juni 2005

Modelit
KvK Rotterdam 24290229



Documentatiepagina

Opdrachtgever RIKZ

Titel Het samenstellen van contouren op basis van raaigegevens,
Achtergronden en Bedieningshandleiding

Datum 30 September 2003
Herzien 29 Juni 2005

Projectteam opdrachtgever E. Biegel

Projectteam Modelit N.J. van der Zijpp

Projectomschrijving Dit document beschrijft hoe op automatische wijze een
contour kan worden samengesteld die alle raai punten insluit,
maar ook de kustlijn volgt. Hiervoor is na naspeuringen naar
een bestaand algoritme een nieuw algoritme ontwikkeld.

Trefwoorden convex hull, polygon

Inhoud

1 Inleiding.....	1
1.1 Achtergrond.....	1
1.2 Principe van het algoritme.....	1
1.3 Beknopte beschrijving van de bediening.....	2
2 Het sorteren van de raaien op volgorde van kustlijn.....	3
2.1 Waarom sorteren.....	3
2.2 Selectie van unieke raailocaties.....	3
2.3 Het selecteren van de linker- en rechterbuurraaien.....	3
2.4 Automatisch sorteren.....	5
2.5 Handmatig sorteren.....	5
3 Het samenstellen van de contour op basis van gesorteerde raaien.....	7
3.1 Het ideale geval: niet overlappende reeksen.....	7
3.2 Incrementeel samenvoegen.....	7
3.3 Een praktijk voorbeeld.....	9
3.4 Mogelijke problemen bij overlappende reeksen en ontbrekende data.....	10
3.5 Indien fouten optreden tijdens de berekening van de contour.....	11
4 Theoretische verantwoording: Het samenvoegen van twee contouren	13
4.1 probleem beschrijving.....	13
4.2 In eerste instantie toegepaste methode: trace contour.....	13
4.3 Theoretische oplossing: Traveling Salesman Probleem.....	13
4.4 Praktisch implementatie.....	14
4.5 TSP algoritme.....	14
Bijlage: beslistabel voor het traceren van een contour.....	15

1 Inleiding

1.1 Achtergrond

De verzameling van die dieptegegevens voor de Hollandse kust is in veel gevallen een lijn georiënteerd. Langs vaste lijnen, aangeduid als meetraaien, worden jaarlijks data verzameld. Via een interpolatie methode, Digipol genaamd, wordt vervolgens het bodemprofiel bepaald.

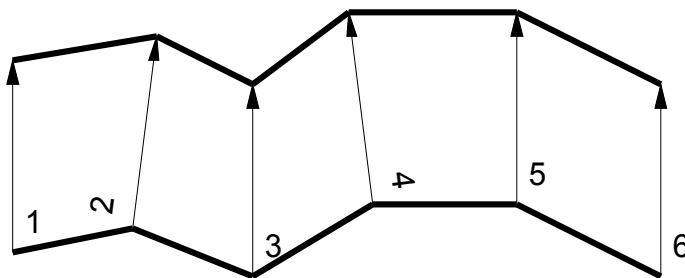
Teneinde de kustlijn te volgen, en niet te interpoleren over land dient een contour te worden opgesteld die het gebied aangeeft waarbinnen geen lid mag worden.

Deze contour is in principe handmatig te tekenen door van links naar rechts de kustlijn te volgen, en te kijken naar de ligging van de datapunten op de raaien. Dit is echter een bewerkelijk proces, daarom is in opdracht van RIKZ een algoritme ontwikkeld om deze taak te automatiseren.

Dit document beschrijft de achtergronden van dit algoritme. In dit hoofdstuk volgt nog een beknopte uitleg over de werking van het algoritme en de bediening ervan. Voor een grondiger begrip van de methode dienen hoofdstuk 2 en 3 geraadpleegd te worden. Technische informatie over de gebruikte algoritmes is voor de geïnteresseerde lezer terug te vinden in hoofdstuk 4.

1.2 Principe van het algoritme

Het idee achter het algoritme is eenvoudig: verbind de begin en de eindpunten van de raaien met elkaar door (zie Figuur 1).



Figuur 1: Door de begin en eindpunten van de raaien met elkaar door te verbinden ontstaat een contour die alle punten insluit en de kustlijn volgt.

1.3 Beknopte beschrijving van de bediening

Het algoritme wordt gestart vanuit het Digipol scherm met de knop “bepaal kleinste insluiting”. In het Digipol scherm worden de kaartbladen geselecteerd die men wil interpoleren. *De datapunten die binnen deze kaartbladen vallen vormen de basis voor het bepalen van de contour.*

Max. afstand tot buur raai (m)	1300
Max. verschil in raaihoek met buur raai (grad)	90

Het algoritme sorteert vervolgens de raai volgorde en presenteert deze (zie Figuur 2). Daarbij wordt een denkbeeldige kustlijn gevolgd die dwars op de raai richtingen ligt. Het **maximale hoekverschil** en de **maximale afstand tussen opeenvolgende raaien** kan worden ingesteld. Het hoekverschil kan worden ingesteld op een maximum van 90 graden.



In incidentele gevallen kan het voorkomen dat het hoekverschil van twee opeenvolgende raaien groter is dan 90 graden. In dit geval kan de buur-relatie niet automatisch worden bepaald en moet desnoods **handmatig** worden ingegrepen. Hiervoor zijn sorteer knoppen in de applicatie opgenomen.

Aantal te gebruiken buurraaien per increment	2
--	---

Een laatste instelparameter is **het aantal burens dat steeds gegroepeerd** wordt. Het algoritme bepaald steeds voor een aantal burens een convexe contour, en schuift vervolgens de focus een raai naar rechts, om weer een convexe contour van een groep van raaien te bepalen. De aldus verkregen contouren worden vervolgens aan elkaar geplakt. Normaliter zal dit proces goed werken wanneer steeds twee burens worden samen genomen. Wanneer de raaien echter grillig zijn kan de omhullende ook hoekig worden. Door meer raaien samen te nemen ontstaat een gladdere, maar ook ruimere contour.

S	ID	Locatie	Metreering	IDLi	IDRe
*	34	FS70004.200	-999999999	--	33
*	33	FS60004.010	-999999999	34	32
*	32	FS60004.000	-999999999	33	31
*	31	FS50003.800	-999999999	32	30
*	30	FS50003.600	-999999999	31	29
*	29	FS50003.400	-999999999	30	28
*	28	FS50003.200	-999999999	29	27
*	27	FS40003.040	-999999999	28	26
*	26	FS40003.030	-999999999	27	25
*	25	FS40003.020	-999999999	26	24
*	24	FS40003.010	-999999999	25	23
*	23	FS40003.000	-999999999	24	22
*	22	FS30002.800	-999999999	23	21
*	21	FS30002.600	-999999999	22	--

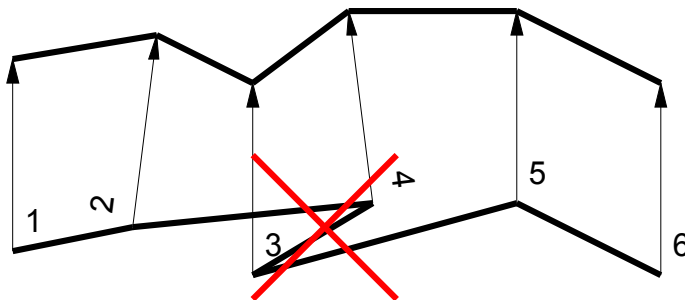
Max. afstand tot buur raai (m) 5000
Max. verschil in raaihoek met buur raai (grad) 90
Aantal te gebruiken buurraaien per increment 2
Te gebruiken aanduiding voor aan te maken contour Increment_2
Creer contour

Figuur 2: Bedienings scherm van de functie “Bepaal kleinste insluiting”

2 Het sorteren van de raaien op volgorde van kustlijn

2.1 Waarom sorteren

Een voorwaarde voor de correcte werking van het algoritme is allereerst dat de raaien in de goede volgorde worden gesorteerd. Als dit niet het geval is wordt een contour gevormd die noch alle data insluit, noch de kustlijn volgt (zie Figuur 3). De sortering van de raaien verloopt in principe automatisch. Er kan echter handmatig worden ingegrepen.



Figuur 3: *Als de raaien niet in de volgorde reeksen worden aangeleverd, wordt een verkeerde contour samengesteld*

2.2 Selectie van unieke raailocaties

Voordat de volgorde van de raaien kan worden bepaald, dient rekening gehouden te worden met de bijzonderheid dat raaien hetzelfde nulpunt en dezelfde raaihoek kunnen hebben. In dit geval worden de datapunten van de betreffende raaien samengevoegd. De huidige procedure rekent vervolgens verder met unieke raailocatie-raaihoek combinaties, zonder datapunten te negeren.

2.3 Het selecteren van de linker- en rechterbuurraaien

De eerste stap in het sorteren is bepalen wie de linker- en rechterburen zijn. Dit gebeurt aan de hand van twee parameters:

- De maximale afstand tussen de beginpunten van twee raaien die als buurraai van elkaar mogen worden beschouwd (kort: max. afstand)
- Het maximale verschil in raaihoek tussen twee raaien die als buurraai van elkaar mogen worden beschouwd (kort: max. hoek)

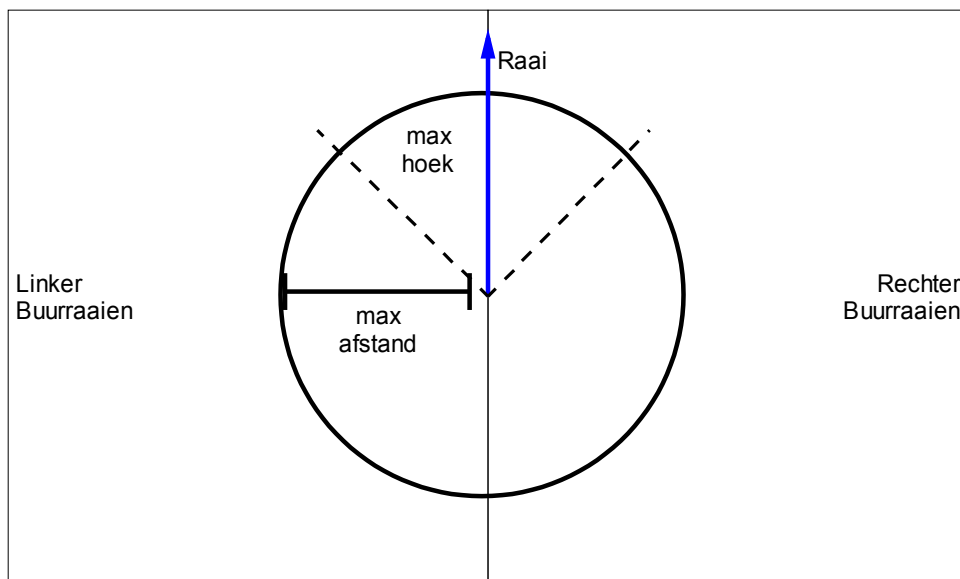
Max. afstand tot buur raai (m)	1300
Max. verschil in raaihoek met buur raai (grad)	90

Figuur 4: Aan de hand van twee parameters worden voor iedere raai buurraaien geselecteerd. Deze worden vervolgens ingedeeld in linker- en rechterburen en op nabijheid gesorteerd.

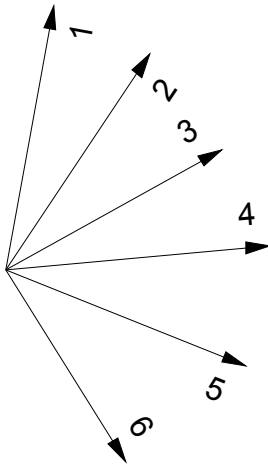
Op basis van deze twee instelparameters kunnen voor iedere raai alle kandidaat buurraaien worden bepaald. De kandidaat buurraaien kunnen worden onderverdeeld in linker- en rechterbuurraaien. Vervolgens moet de meest dichtstbijzijnde linker en rechterbuur worden bepaald. Deze bepaling vindt plaats door achtereenvolgens te sorteren op:

- afstand van oorsprong naar oorsprong
- verschil in raaihoek

Dit tweede sorteercriterium is nodig om ook met zogenaamde sterraaien (zie Figuur 6) om te kunnen gaan.



Figuur 5: Bepaling van de kandidaat-buurraaien, onder te verdelen in linkerburen en rechterburen



Figuur 6: *Voorbeeld van een sterraai*

2.4 Automatisch sorteren

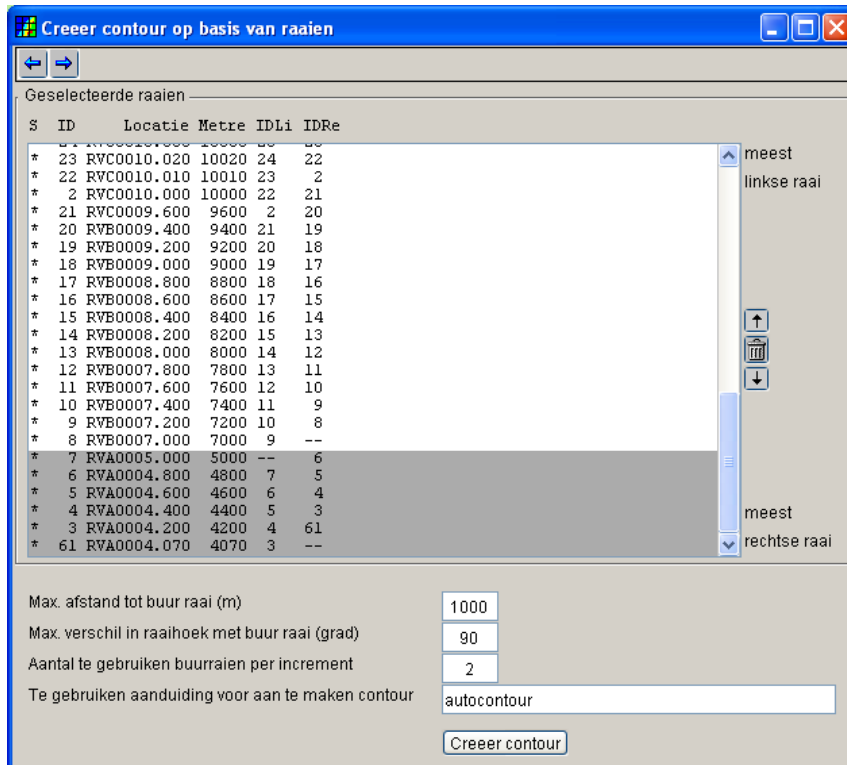
Nadat voor iedere raai de dichtstbijzijnde linker- en rechterbuur is bepaald is het sorteren slechts een kwestie van beginnen met de meest rechtse raai (een raai zonder rechterburen) en daarbij de linkerbuur zoeken, om vervolgens door te gaan met deze raai, en zo voort.

De procedure wordt als succesvol beschouwd indien er precies 1 gesorteerde keten van buurraaien wordt gegenereerd. Wanneer er meerdere ketens resulteren, betekent dit dat de raaien niet bij elkaar aansluiten. Dit gebeurt bijvoorbeeld als ze van verschillende kustdelen of eilanden afkomstig zijn. De gebruiker dient in dit geval uitsluitel te geven omtrent de onderlinge volgorde van deze sets (zie Handmatig sorteren).

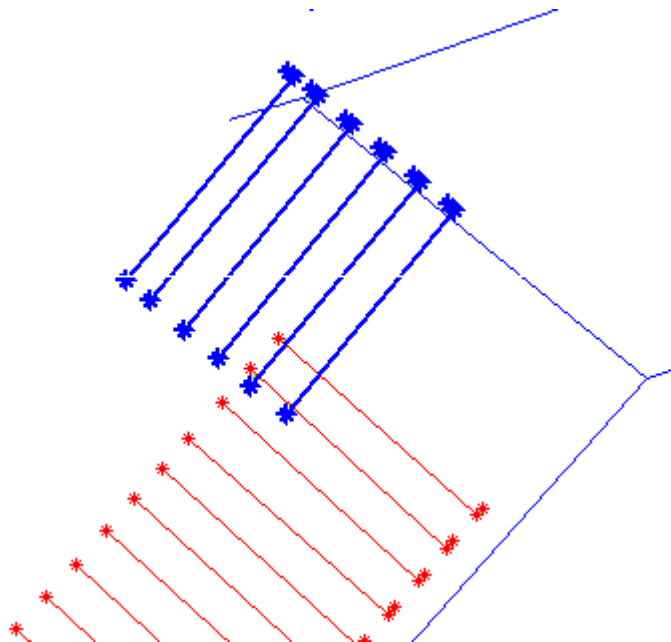
2.5 Handmatig sorteren

In uitzonderlijke gevallen kan het nodig zijn de automatisch aangebrachte volgorde te wijzigen, bijvoorbeeld als raaien ten onrechte aan een buurraai zijn gekoppeld, of indien een buurraai niet is gevonden (zie Figuur 8).

Met behulp van een interface kan de raai volgorde gewijzigd worden. Selecteer 1 of meerdere raaien en beweeg de hele groep omhoog of omlaag. Het is ook mogelijk om bepaalde raaien te negeren bij het bepalen van de contour.



Figuur 7: Nadat de raaien automatisch zijn gesorteerd krijgt de gebruiker nog eens de mogelijkheid handmatig de volgorde te wijzigen. Dit kan nodig zijn indien de buurraaien niet automatisch kunnen worden aangewezen, bijvoorbeeld als de hoek tussen twee groepen van raaien groter dan de ingestelde waarde is.



Figuur 8: In dit praktijkvoorbeeld wijkt de raaihoek van de blauw gearceerde raaien meer dan 90 graden af van de raaihoek van de rood gearceerde raaien. Hierdoor is het onmogelijk om de buurrelatie automatisch vast te stellen.

3 Het samenstellen van de contour op basis van gesorteerde raaien

3.1 Het ideale geval: niet overlappende reeksen

In het ideale geval liggen alle raaien parallel naast elkaar, is voor iedere raai de buurraai correct bepaald, en liggen de datapunten van iedere nieuwe raai duidelijk buiten de contour die om de vorige raaien getrokken kan worden.

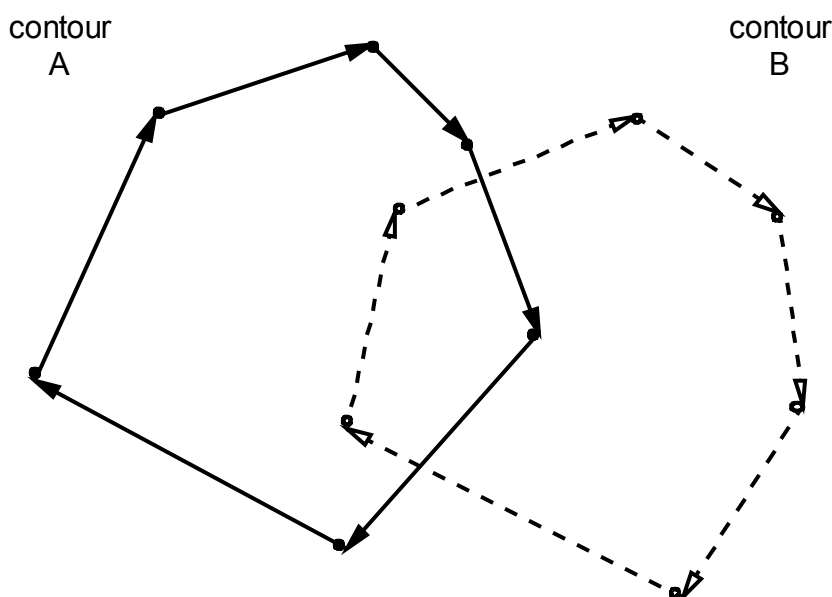
In dit geval laten de kleinste omhullende op een eenduidige manier vormen. Er zijn verschillende manieren denkbaar om de contour af te leiden. Binnen Maria wordt de methode van het incrementeel toevoegen gebruikt. Deze methode wordt hieronder geïllustreerd.

3.2 Incrementeel samenvoegen

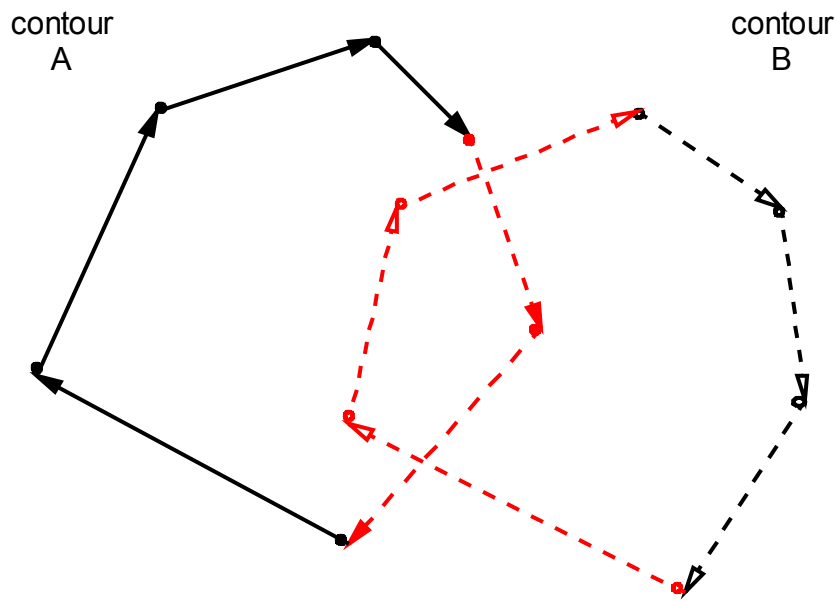
Nadat de volgorde van de raaien is bepaald, kan een contour worden afgeleid. Dit gebeurt via een incrementeel proces. Tijdens dit proces wordt begonnen met de twee meest linkse raaien. Om deze twee raaien wordt de kleinste convexe omhullende bepaald (de zogenaamde elastiekjes methode). Deze methode is gelijk aan de standaard functie voor omhullende bepaling die ook wordt aangeroepen vanuit de functie "Bepaal omhullende" die elders wordt toegelicht.

Vervolgens wordt hetzelfde gedaan voor de 2^e en 3^e raai van links. Om deze manier ontstaan twee contouren die tenminste 1 punt gemeenschappelijk hebben. Deze twee contouren worden vervolgens samengevoegd onder de volgende voorwaarde.

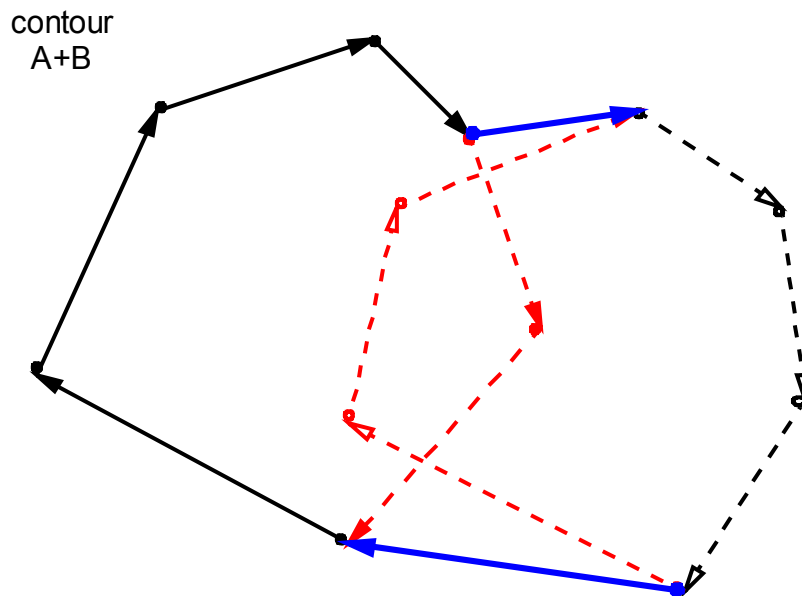
De samengevoegde contour bevat alle contourpunten van de twee samenstellende contouren, voorzover deze niet worden ingesloten door de ander contour.



Figuur 9: Samen te voegen contouren A en B



Figuur 10: Verwijder de contourpunten die door elkaar worden ingesloten. En de lijnstukken die elkaar kruisen. Er ontstaan tenminste twee lijnstukken



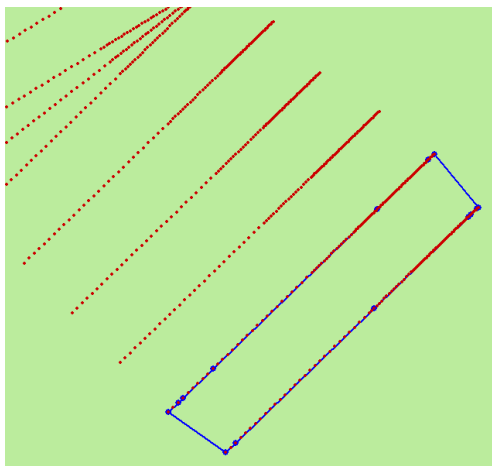
Figuur 11: Voeg de lijnstukken samen tot een gesloten contour. Doe dit op de manier waarbij ieder lijnstuk precies 1 maal terugkomt in de samengevoegde contour en de totale lengte van de ingevoegde lijnstukken (in blauw aangegeven) minimaal is.

3.3 Een praktijk voorbeeld

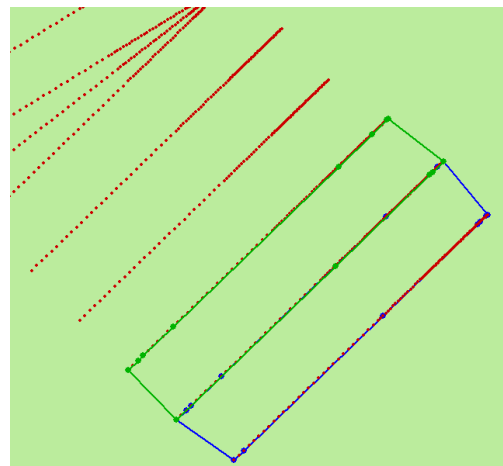
De onderstaande tabel laat zien hoe de methode in de praktijk uitwerkt:

Tabel 1: *Iteratie schema samenstellen contour*

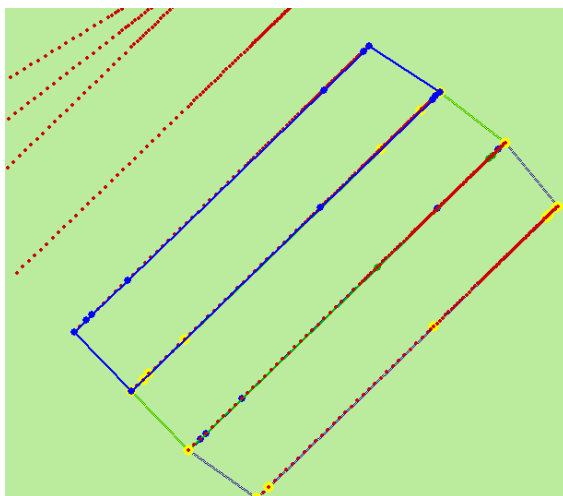
Iteratie	Start iteratie met: stapsgewijs convexe omhullende van raaiverzameling	Voeg toe: convexe omhullende van raaiverzameling	Resultaat: stapsgewijs convexe omhullende van raaiverzameling	Zie Figuur
1	{}	{1,2}	{1,2}	Figuur 12
2	{1,2}	{2,3}	{1,2,3}	Figuur 13
3	{1,2,3}	{3,4}	{1,2,3,4}	Figuur 14
4	...	...	...	Figuur 15



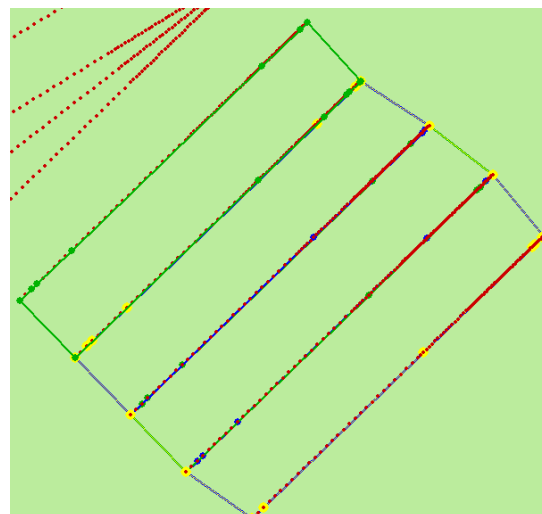
Figuur 12: *Iteratie 1*



Figuur 13: *Iteratie 2*



Figuur 14: *Iteratie 3*



Figuur 15: *Iteratie 4*

3.4 Mogelijke problemen bij overlappende reeksen en ontbrekende data

In de praktijk kunnen zich problemen voordoen met raaien elkaar kruisen of raaien die diep in zee gemeten zijn die afgewisseld worden met raaien die maar over een lengte van enkele honderden meters gevaren zijn. De resulterende contour kan er in dit geval onnatuurlijk en hoekig uit komen te zien.

Wanneer men niet tevreden is met het resultaat kan de parameter “Aantal te gebruiken buurraaien per increment” worden verhoogd.

In dit geval wordt begonnen met de N meest linkse raaien, met N de ingestelde waarde. Om deze N raaien wordt weer de kleinste convexe omhullende bepaald. Vervolgens wordt hetzelfde gedaan voor de 2^e tot 2+N^e raai van links. Om deze manier ontstaan weer twee contouren die tenminste 1 punt gemeenschappelijk hebben. Deze twee contouren worden vervolgens samengevoegd, om daarna door te gaan met de volgende groep van N raaien.

Door N groter dan 2 te kiezen wordt de methode robuuster voor overlappende reeksen en ontbrekende data, maar wordt de kustlijn minder nauwkeurig gevolgd. Figuur 17 geeft hier een voorbeeld van.

Naarmate het aantal raaien per increment hoger wordt gekozen gaat de methode “Bepaal kleinste insluiting” meer lijken om de meer eenvoudige methode “Bepaal omhullende” (ook wel elastiekjes methode genoemd).

Als richtlijn voor het invullen voor de parameter geldt dat de waarde op het minimum van 2 ingesteld moet worden zolang de data dit toelaat. Als de resulterende contour te hoekig wordt, dient een grotere waarde ingesteld te worden. Hoger dan 5 zal in de praktijk echter zelden nodig zijn.

De rekentijd voor het berekenen van de contour neemt af naarmate een hogere waarde wordt ingesteld. Dit komt doordat de contour minder hoekpunten bevat bij een hoge waarde van het aantal raaien per increment.

Aantal te gebruiken buurraaien per increment	2
--	---

Figuur 16: *Invulveld voor het instellen van het aantal te gebruiken raaien per increment. De minimale waarde is 2.*



Figuur 17: Voorbeeld waarbij de contour bereken is op basis van incrementen van 2 raaien (groen) en 5 raaien (rood).

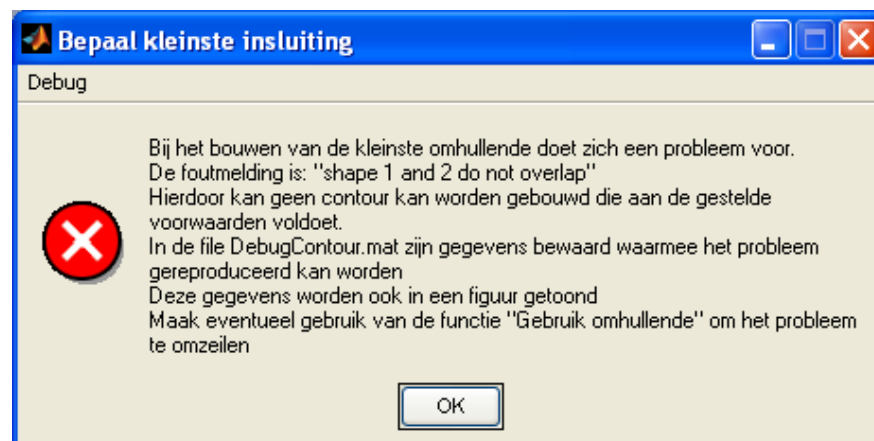
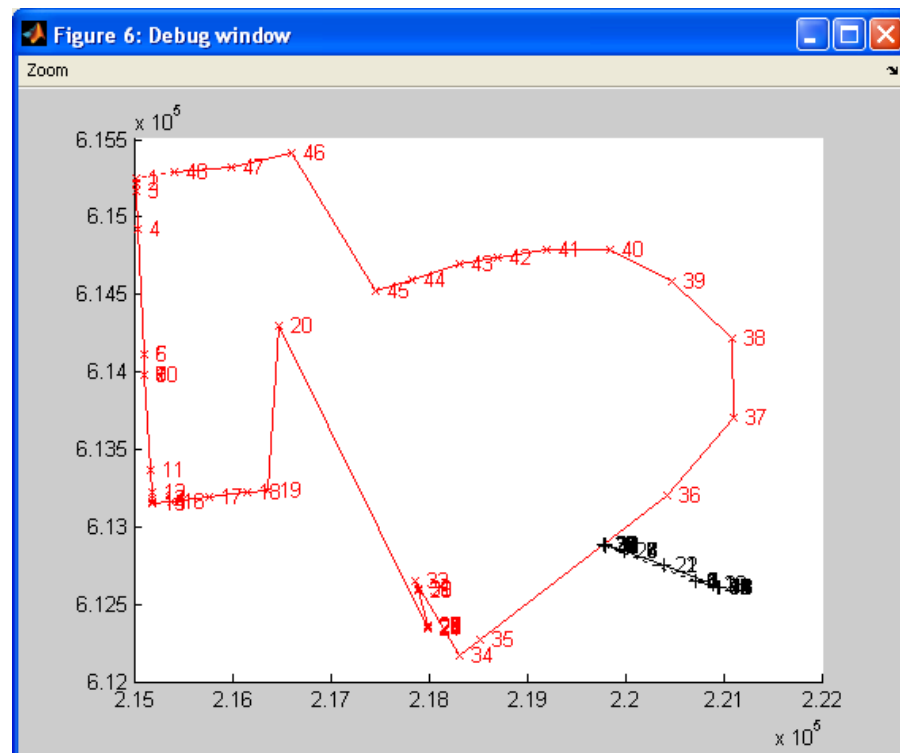
3.5 Indien fouten optreden tijdens de berekening van de contour

Niet alle situaties die zich in de praktijk kunnen voordoen kunnen vooraf worden voorzien. Het algoritme kan echter wel controleren of aan de minimale voorwaarden voor het bepalen van een contour is voldaan. Als dit niet zo is, dan zal het algoritme

een apart scherm openen en een weergave van de situatie tonen, met daarbij een foutmelding die de aard van het probleem toelicht.

Over het algemeen zijn er drie manieren om het probleem op te lossen:

- Controleer of de raaien in de goede volgorde zijn aangeboden. Als de raaien verkeerd gesorteerd zijn kan geen contour worden bepaald;
- Verhoog de parameter: aantal raaien per increment;
- Gebruik in plaats van de huidige methode de methode "Bepaal omhullende". Deze is weliswaar minder nauwkeurig, maar wel meer robuust.



Figuur 18: Wanneer niet aan de nodige voorwaarden voor het automatisch bepalen van een contour is voldaan volgt een situatieschets en een foutmelding.

4 Theoretische verantwoording: Het samenvoegen van twee contouren

4.1 probleem beschrijving

Het bouwen van een omhullende is een iteratief proces. Steeds wordt de bestaande contour uitgebreid met een nieuwe contour. Deze nieuwe contour is bepaald als de convexe omhullende van een aantal raaien. Door deze werkwijze wordt enerzijds gegarandeerd dat de kustlijn gevolgd wordt, en anderzijds dat alle raaipunten in de contour worden opgenomen.

Iedere contour die wordt toegevoegd is een zogenaamde convexe contour.

De nieuwe contour bestaat volledig uit punten die ook op de randen van de samenstellende contouren liggen. Punten van contour 1 die binnen contour 2 liggen zijn niet vertegenwoordigd in de samenstellende contour, net zomin als punten van contour 2 die binnen contour 1 liggen.

Het is dus betrekkelijk eenvoudig om de verzameling vast te leggen *welke* de punten die de omhullende contour vormen. Des te moeilijker is het om de *volgorde* van deze punten te bepalen.

4.2 In eerste instantie toegepaste methode: trace contour

Oorspronkelijk is voor het bepalen van de omhullende van twee samengevoegde contouren een algoritme ontwikkeld dat gebaseerd was op het “volgen” (tracen) van een contour. Het algoritme werkte in grote lijnen als volgt:

Het algoritme begon met een extern punt op contour 1, en sprong vervolgens na het volgende punt. Wanneer dit ook een extern punt was werd het lijnstuk toegevoegd aan de contour. Wanneer het volgende punt van de oorspronkelijke contour een intern punt was van contour 2, dan werd een snijpunt bepaald en toegevoegd aan de nieuwe contour. Vervolgens werd het proces voortgezet op contour 2. In de bijlage staat in meer detail toegelicht hoe het proces werkt.

Alhoewel aan het algoritme veel tijd besteed, bleek het niet mogelijk om dit algoritme werkend te krijgen voor alle configuraties van raaien die zich voort deden. Vooral raaien en die elkaar snijden en raaien die niet parallel lopen blijven steeds voor nieuwe problemen zorgen.

Uiteindelijk is deze aanpak daarom verlaten.

4.3 Theoretische oplossing: Traveling Salesman Probleem

De theoretische oplossing van het volgorde probleem is dat de volgorde zo gekozen wordt dat:

- alle rand punten worden gebruikt;
- de totale lengte van de omhullende minimaal is.

Op deze wijze wordt contour gevormd die overeenkomt met het intuïtieve idee van een omtrek.

Het bovenstaande probleem staat wiskundig bekend onder de naam Traveling Salesman Probleem (TSP). Dit is een zogenaamd NP-hard probleem. Dit betekent dat bij een grote aantal rand punten een exacte oplossing van het probleem niet meer mogelijk is binnen acceptabele rekentijd.

Een TSP-probleem met minder dan 10 punten kan echter zonder problemen exact worden opgelost. Bij TSP-problemen met minder dan circa 30 punten biedt een eenvoudig te implementeren gerandomiseerde oplossingsmethode over het algemeen nog wel uitkomst. Boven deze afmetingen zijn gespecialiseerde algoritmes nodig, waarvan de Lin-Kernighan code de bekendste is. Het implementeren van dergelijke codes is echter zeer bewerkelijk.

4.4 Praktisch implementatie

Om het rekenwerk zoveel mogelijk te beperken is ervoor gekozen om voor het toepassen van het TSP algoritme het probleem zoveel mogelijk in te perken. Dit gebeurt door de randpunten niet te beschouwen als individuele punten, maar als ketens van punten. De volgende ketens worden onderscheiden:

- Ketens bestaande uit strikt externe punten van contour 1 die met elkaar verbonden zijn via lijnstukken die contour 2 niet snijden.
- Ketens bestaande uit strikt externe punten van contour 2 die met elkaar verbonden zijn via lijnstukken die contour 1 niet snijden.
- Geïsoleerde punten van contour 1 die op de rand van contour 2 liggen
- Geïsoleerde punten van contour 2 die op de rand van contour 1 liggen

Een keten kan dus ook bestaan uit een enkel punt.

Nadat de ketens zijn geïdentificeerd worden dubbele punten indien mogelijk verwijderd. Dit heeft geen invloed op de een oplossing, maar wel op de hoeveelheid rekenwerk.

Stel dat op deze manier 8 ketens worden geïdentificeerd. Dan wordt vervolgens een afstandmatrix berekend tussen deze ketens. Elementen van deze matrix, $A(\text{herkomst}, \text{bestemming})$, definiëren de afstand tussen het eindpunt van de reeks "herkomst" en het beginpunt van de reeks "bestemming".

4.5 TSP algoritme

De laatste stap bestaat uit het oplossen van het TSP probleem. In de meeste gevallen gaat het om een probleem met minder dan 9 punten. Dit kan gemakkelijk via een brute-force methode worden opgelost. Indien het aantal punten groter is dan 9, blijkt de rekentijd zeer snel op te lopen. Daarom wordt in deze gevallen een randomized algoritme gebruikt. Bij alle praktische voorbeelden die zijn doorgerekend bleek deze grens echter niet te worden overschreden.

In alle waarnemingen die tot nu toe zijn gedaan wijzen vindt dit randomized algoritme de exacte oplossing. Dit is echter (uiteraard) nooit gegarandeerd.

Bijlage: beslistabel voor het traceren van een contour

De onderstaande tabel beschrijft heuristiek voor het volgen van een contour. Deze heuristiek is oorspronkelijk in een algoritme geïmplementeerd maar wordt nu niet meer gebruikt. De reden hiervoor is dat het steeds weer opnieuw nodig bleek om de heuristiek met extra beslisregels uit te breiden teneinde specifieke situaties het hoofd te kunnen bieden.

	Actieve contour	Ligging beginpunt t.o.v. andere contour	Ligging eindpunt t.o.v. andere contour	Actie
A	1	out	out	- Lijnstuk toevoegen aan nieuwe contour (→A) Ga verder op contour 1 met het eindpunt
	1	out	on	- Lijnstuk toevoegen aan nieuwe contour (→B) Ga verder op contour 1 met het eindpunt
	1	out	in	Dit lijnstuk snijdt tenminste 1 maal contour 2. Bepaal de het eerste snijpunt. - Bepaal het eindpunt (op contour 2) van het gesneden lijnstuk. - Toevoegen aan nieuwe contour: (beginpunt van contour 1, eindpunt van contour 2) (→D) Ga verder op contour 2 met het eindpunt
B	1	on	out	- Lijnstuk toevoegen aan nieuwe contour (→A) Ga verder op contour 1 met het eindpunt
	1	on	on	Dit lijnstuk valt mogelijk samen met de buitengrens van contour 2, of loopt volledig door het interne gedeelte van contour 2. Een simpele test daarvoor is om te controleren of: (beginpunt + eindpunt)/2 is intern punt van contour 2. Indien het lijnstuk samenvalt met de buitengrens van contour 2: - Lijnstuk toevoegen aan nieuwe contour (→B) Ga verder op contour 1 met het eindpunt Anders: (→E) Ga verder op contour 2 met het beginpunt
	1	on	in	(→E) Ga verder op contour 2 met het beginpunt
C	1	in	out	deze toestand doet zich nooit voor
	1	in	on	deze toestand doet zich nooit voor
	1	in	in	deze toestand doet zich nooit voor

Het samenstellen van contouren op basis van raaigegevens
Achtergronden en Bedieningshandleiding

D	2	out	out	- Lijnstuk toevoegen aan nieuwe contour (→D) Ga verder op contour 2 met het eindpunt
	2	out	on	- Lijnstuk toevoegen aan nieuwe contour (→E) Ga verder op contour 1 met het eindpunt
	2	out	in	Dit lijnstuk snijdt tenminste 1 maal contour 1. Bepaal de het eerste snijpunt. - Bepaal het eindpunt (op contour 1) van het gesneden lijnstuk. - Toevoegen aan nieuwe contour: (beginpunt van contour 2, eindpunt van contour 1) (→A) Ga verder op contour 1 met het eindpunt
E	2	on	out	- Lijnstuk toevoegen aan nieuwe contour (→D) Ga verder op contour 2 met het eindpunt
	2	on	on	Ga verder op contour 1 met het beginpunt
	2	on	in	Omdat het niet zeker is dat contour 1 convex is, zou er een snijpunt kunnen bestaan waardoor deze lijn niet volledig binnen contour 1 valt. Indien dit snijpunt bestaat: - bepaal de eerste snijdende lijn - Toevoegen aan nieuwe contour: (beginpunt van contour 2, eindpunt van snijdende lijn op contour 1) - Ga verder op contour 1 met het eindpunt van de snijdende lijn Anders - ga verder op contour 1 met het beginpunt
F	2	in	out	deze toestand doet zich nooit voor
	2	in	on	deze toestand doet zich nooit voor
	2	in	in	deze toestand doet zich nooit voor