

Geo-informatie op variabele schaal voor mobiel internet-GIS

Het omgaan met verschillende kaartschalen (detailniveaus) of het afleiden van een gegeneraliseerde representatie uit een detail-representatie, is een onderwerp dat bijna even oud is als de productie van de eerste geografische kaart. Er is in de praktijk veel behoefte aan consistente series kaartweergaven met verschillende detailniveaus en er is dan ook al veel onderzoek gedaan naar dit onderwerp. In dit artikel beschrijven we een generalisatie-aanpak die het mogelijk maakt dezelfde geo-informatie op verschillende schalen/detailniveaus weer te geven en te gebruiken, zonder dat het nodig is die informatie ook meerdere keren (op verschillende schalen) op te slaan.

De overgang naar digitale geo-informatie was een belangrijke stimulans om te trachten het generalisatieproces te formaliseren en vervolgens te automatiseren. Hoewel er goede vooruitgang is geboekt, kan nog steeds niet worden geclaimd dat het probleem is opgelost. Echter, de behoefte aan automatische generalisatie neemt alleen maar toe. Dit komt ook door het toenemend belang van internet-GIS: het wordt mogelijk om gegevens van verschillende bronnen, en mogelijk met verschillende bronschalen, met elkaar te combineren. Dit is alleen zinvol indien vergelijkbare schalen worden gecombineerd. Om snel te kunnen visualiseren, is het bovendien gewenst om geen onnodige details over te sturen want dat maakt de internet-GIS-toepassing onnodig traag. Voor mobiele internet-GIS-toepassingen geldt bovendien nog eens dat de schermjes vrij beperkt zijn en dus de behoefte aan generalisatie groot is.

In dit artikel wordt een structuur beschreven die het mogelijk maakt de geo-informatie alleen op het meest gedetailleerde niveau op te slaan, maar op elke gewenste schaal/detailniveau te bevragen. De structuur werkt als een soort (luxe) 'index': een gebruiker geeft het gewenste gebied en de schaal op en de structuur zorgt dat de informatie snel kan worden gevonden en opgehaald. Afhankelijk van de toepassing zouden meerdere van dergelijke indexen op dezelfde brondata gedefinieerd kunnen worden zodat er toepassingsspecifiek wordt gegeneraliseerd. De naam van de structuur (of 'index') is de topologische GAP-structuur ofwel tGAP-structuur (GAP = generalized area partition, te vertalen als gegeneraliseerde vlakken partitie). Het doel van de tGAP-structuur is om ruimtelijke data éénmaal op te slaan, zonder redundantie van geometrie, om hier vervolgens op verzoek 'on the fly' verschillende representaties van te kunnen afleiden op elk gewenst detailniveau. Dit artikel bespreekt de volgende twee aspecten van het gebruik van de tGAP-structuur: hoe het progressief (eerst grof, daarna verfijningen) oversturen van vectordata van webserver naar client mogelijk kan worden gemaakt én hoe geleidelijke zoom kan worden gerealiseerd, bij voorkeur in de vorm van een animatie in kleine stappen ('smooth zooming'). Hoewel



Ir. B.M. Meijers, Koninklijke Dirkszwager te Maasvluis, co-auteurs drs. M.E. de Vries en prof.dr.ir. P.J.M. van Oosterom, sectie GIS-technologie, OTB, TU Delft.

de tGAP-structuur oorspronkelijk ontworpen is voor vlakpartities, zijn er geen principiële redenen waarom dezelfde structuur ook niet kan worden toegepast voor lijn- en puntobjecten.

Variabele schaal en progressieve data overdracht

Voor een interactief GIS waarmee ruimtelijke gegevens kunnen worden bewerkt en bevestigd, zijn zoom-operaties (in- en uitzoomen) en het verschuiven van het getoonde gebied (pannen) elementaire operaties. Deze operaties stellen echter wel eisen aan de opgeslagen ruimtelijke data, als de snelheid een belangrijke factor is. Het is van belang dat er structuren binnen de database aanwezig zijn om snel te kunnen selecteren welke informatie precies te presenteren aan de eindgebruiker. Veelal worden hier indexering-mechanismen voor toegepast, waarbij hiërarchische structuren ('grafen' en 'bomen') worden gebruikt om de opslag volgens een bepaald principe te realiseren.

Datastructuren die zulk interactief gebruik van data op meerdere schalen ondersteunen, zijn nog steeds zeldzaam. Een 'oplossing' is om data op meerdere schaalniveaus op te slaan in MRDBs (multiple resolution databases). Om consistentie te kunnen leveren tijdens het gebruik van de data, proberen deze datastructuren een expliciete relatie tussen de objecten op verschillende schaalniveaus op te slaan. Nadeel hierbij is dat deze structuren redundante data opslaan (dezelf-

de coördinaten, afkomstig van dezelfde brondata, op meerdere niveaus) en dat ze alleen een beperkt aantal schaalniveaus kunnen leveren. Deze structuren kunnen echter wel een bijdrage leveren voor interactief gebruik en het snel tonen van kaartmateriaal op een gewenste schaal.

Een ander nadeel van de MRDBs is dat ze niet geschikt zijn voor het progressief oversturen van data omdat voor elke schaal een bepaalde (onafhankelijke) grafische representatie moet worden overgestuurd. Dit is minder efficiënt dan het hergebruik van al eerder overgestuurd materiaal waarbij alleen wijzigingen ten opzichte van de vorige getoonde data hoeven te worden overgestuurd. Mooie voorbeelden van progressief datatransport zijn te vinden bij rasterplaatjes die eerst grof overgestuurd kunnen worden en snel kunnen worden getoond en even later met meer detail aangevuld kunnen worden, als de gebruiker wat langer wacht. Soortgelijke effecten zijn moeilijker te bereiken met vectordata, alhoewel er recentelijk wel een aantal pogingen zijn gedaan [Bertolotto en Egenhofer, 2001; Buttenfield, 2002; Jones et al, 2000; Zhou et al, 2004].

Achtergrond bij de topologische GAP-structuur

We zullen geen poging doen om de datastructuren tot in alle details te beschrijven; daarvoor wordt verwezen naar het kader 'Opbouw tGAP-structuur' op p. 532 en naar [Van Oosterom, 2005; Meijers, 2006]. Hier worden wel de belangrijkste componenten van de tGAP-structuur kort beschreven: de GAP-face tree, het GAP-edge forest, de BLG-tree en een 3D R-tree. De schaal (of belangrijkheid) van de geo-objecten kan worden gezien als een derde dimensie naast de twee dimensies van de geo-informatie zelf zodat efficiënte selectie met een 3D R-tree kan gebeuren.

Aan de basis van de tGAP-structuur ligt de keuze voor een topologisch datamodel. De geo-objecten worden gerepresenteerd door nodes (punten), edges (lijnen) en faces (vlakken). Door tijdens het generalisatieproces iedere keer twee vlakken (of beter gezegd twee faces) samen te voegen en het nieuwe samengevoegde vlak de plaats in te laten nemen van de twee oude vlakken, wordt een binaire boom van

faces opgebouwd: de GAP-face tree. Deze binaire boom kan vervolgens bij het bevragen en visualiseren van de geo-informatie worden gebruikt om te selecteren welke vlakken op welk detailniveau dienen te worden getoond.

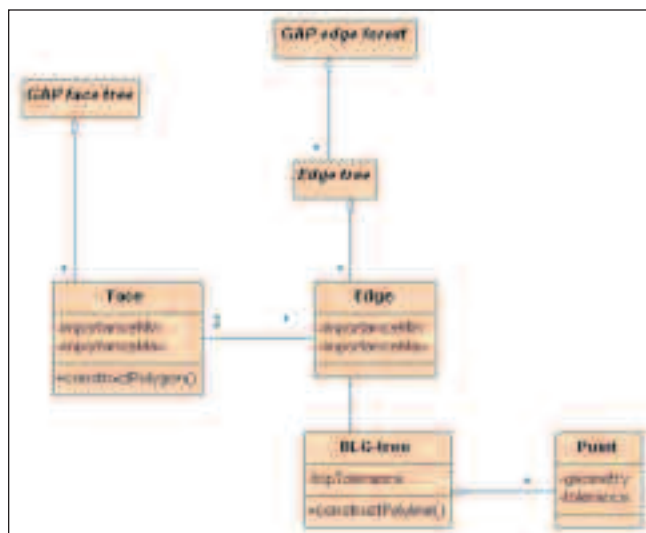
De edges in de datastructuur worden op hun beurt gerepresenteerd door BLG-trees (binaire lijn generalisatie, op basis van het Douglas-Peucker algoritme voor lijnsimplificatie) die het mogelijk maken om meer of minder detail van de lijnen te tonen. Tijdens het samenvoegen van de vlakken kan de gemeenschappelijke lijn van twee vlakobjecten worden weggelaten op het lagere detailniveau. Wanneer twee of meer lijnstukken worden gecombineerd tot een nieuw lijnstuk door een samenvoegingslag van twee buurvlakken, worden de twee BLG-bomen aan elkaar gekoppeld zodat ook voor het nieuwe lijnstuk, door de combinatie van BLG-bomen, informatie over lijnsimplificatie beschikbaar is.

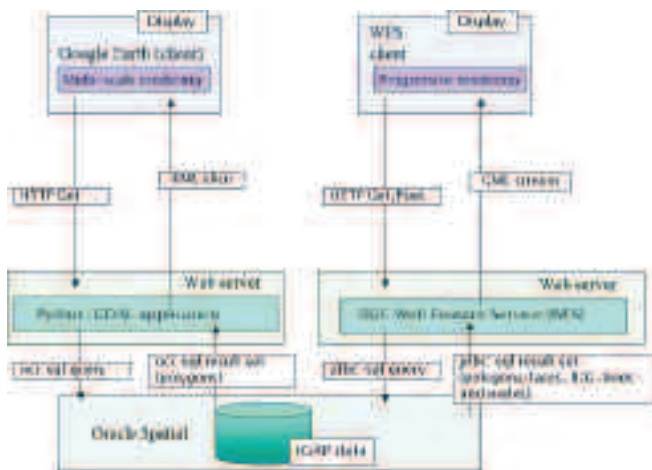
Het opbouwen van de vlakken-boom kan worden gestuurd door het toekennen van 'belangrijke waarden' aan alle faces. Op basis van deze waarden wordt de keuze gemaakt welk vlak als eerste moet worden samengevoegd met de meest geschikte buur. De 'compatibiliteitsmatrix' tussen de verschillende typen van vlakobjecten wordt hiervoor mede gebruikt. Het uiteindelijke resultaat is dan applicatieafhankelijk want de belangrijkheidswaarden en de compatibiliteitsmatrix verschillen per toepassing. Afhankelijk van het doel van de kaart wordt er gegeneraliseerd. Meerdere tGAP-structuren kunnen boven op dezelfde dataset worden gebouwd. Echter, de datastructuur is tot op heden nog statisch: wijzigingen in de brondata leiden nog niet automatisch tot wijzigingen in de opgebouwde tGAP-structuur. Het dynamisch maken van deze structuur is lopend onderzoek in het kader van het Bsik RGI-233 project 'Usable well-scaled mobile maps for consumers', evenals het experimenteren met het opnemen van lijn- en puntobjecten (www.gdmc.nl/uwsm2).

Mogelijke gebruik van de datastructuur

De tGAP-structuur is in een afstudeeronderzoek geïmplementeerd in een geo-DBMS (Oracle Spatial) omgeving [Meijers, 2006]. Het UML-klassendiagram (fig.1) toont hoe het conceptuele model van de tGAP-structuur er uitziet: boomstructuren voor de GAP-faces en de GAP-edges, en per edge

Fig.1. Conceptueel model tGAP-structuur.





een BLG-tree bestaande uit de punten met hun geometrie en tolerantiewaarde. Bij de implementatie zijn alleen de niet-cursieve klassen omgezet naar expliciete tabellen in de database. Verder is onderzocht hoe de tGAP-structuur effectief ingezet kan worden in een webserver/client-omgeving.

Tijdens het bouwproces van de tGAP-structuur worden de databasetabellen gevuld, uitgaande van de meest gedetailleerde basisgegevens (een topologisch opgeslagen vlakkenpartitie). Dit proces resulteert dus niet in een beperkt aantal 'schalen', zoals bij MRDB's, maar in een topologische variabele-schaal structuur.

Fig. 2. Gebruik van de tGAP-structuur; links met Google Earth client voor het genereren van een specifieke schaal, rechts met WFS client voor progressieve transfer.

Nadat de tGAP-structuur is opgebouwd, kan deze datastructuur op verschillende manieren worden gebruikt:

- het afleiden van een complete representatie op een bepaalde gewenste schaal;
- het genereren van een delta-representatie met aanvullende details om van een gegeven schaal naar een hoger detailniveau te komen voor grotere schaal.

De eerste manier om de tGAP-structuur te gebruiken, is om met de opgeslagen verwijzingen efficiënt een specifieke schaal 'on the fly' van de brondata af te leiden. Hiervoor is het noodzakelijk om de gewenste schaal uit te drukken in een 'belangrijke waarde' waarmee de datastructuur de gewenste generalisatie kan aanleveren. In het afstudeeronderzoek is deze opzet uitgewerkt, waarbij Google Earth is gebruikt voor de visualisatie (fig. 2).

Fig. 3. De tGAP-structuur in actie: meer detail bij inzoomen (Google Earth client).

Vanuit Google Earth wordt een bounding box verstuurd naar de variabele-schaal geo-informatieserver. Met deze bounding box kan worden bepaald



welke belangrijkheidswaarde de te tonen vlakken tenminste moeten hebben. Het uitrekenen welke belangrijkheidswaarde moet worden gebruikt, gebeurt in een Python-applicatie die als module binnen de webserver draait. Vanuit de opgeslagen verwijzingen kan de geometrie voor de vlakobjecten worden gereconstrueerd. De Python-applicatie maakt gebruik van de Geospatial Data Access Library (GDAL) en verzorgt het ophalen van alle geometrie uit de database aan de hand van de topologische verwijzingen. De gereconstrueerde geometrie wordt vervolgens aan Google Earth geretourneerd in het KML-formaat, waarna de vlakobjecten op het gewenste detailniveau in Google Earth worden getoond. Elke keer als er wordt in- of uitgezoomd, wordt de berekening herhaald en wordt opnieuw een complete vlakkenpartitie opgehaald op een ander detailniveau (fig. 3).

Deze manier biedt dus nog geen mogelijkheid voor hergebruik van reeds overgestuurde data: voor elk zoomniveau wordt een volledige nieuwe set gegevens opgehaald en gevisualiseerd (een KML-'slice'). Dit is echter niet optimaal, met name niet voor mobiele toepassingen waarbij bandbreedte nog altijd een beperkende factor is.

Geleidelijk zoomen en pannen in de mobiele client

De tweede manier van gebruik van de tGAP-structuur maakt betere gebruikersoriëntatie mogelijk door progressieve data-overdracht en geleidelijk zoomen. Ondanks de groeiende populariteit van geo-informatie (kaarten) in allerlei mobiele toepassingen, zoals navigatiesystemen of het geven van ruimtelijke (context-) informatie voor een bepaald object, zijn er momenteel serieuze problemen bij het gebruik van deze mobiele kaarten. Eén van de belangrijkste problemen wordt veroorzaakt door de beperkte afmetingen van de schermjes van de mobiele apparaten die slechts kleine kaartfragmenten kunnen tonen. Een gebruiker moet zoomen en pannen om voldoende ruimtelijk begrip te krijgen, dat wil zeggen: een gevoel van de grootte, de richtingen en de afstanden tussen de relevante objecten en hun context. Met de huidige technologie in de meeste mobiele GISsen (Location Based Services) is het mogelijk te zoomen

en te pannen. Echter, na een zoom- of een pan-actie wordt in bijna alle gevallen de representatie volledig opnieuw getekend. De gebruiker raakt vaak de 'mentale' verbinding tussen de twee representaties kwijt. Huidige ervaringen tonen aan dat gebruikers hierdoor gedesorienteerd raken en dat ze geen goede mentale kaart kunnen opbouwen van de omgeving waar ze zich bevinden. Eén van de belangrijkste oplossingen voor het beschreven probleem van gebruikersdesoriëntatie is vario-schaal-kaarten. Er zijn ten minste twee mogelijke manieren om vario-schaal kaarten te gebruiken:

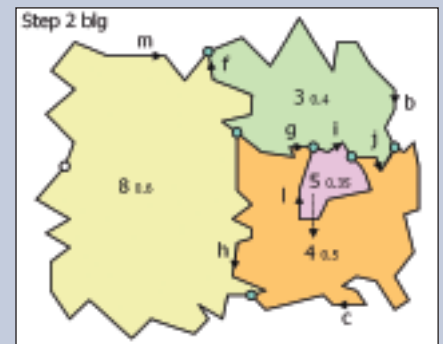
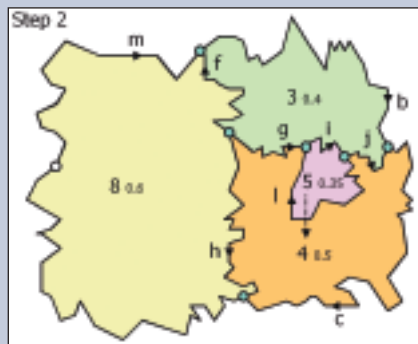
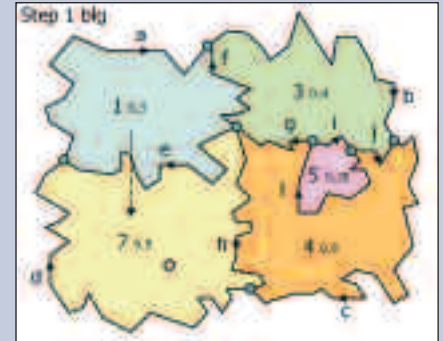
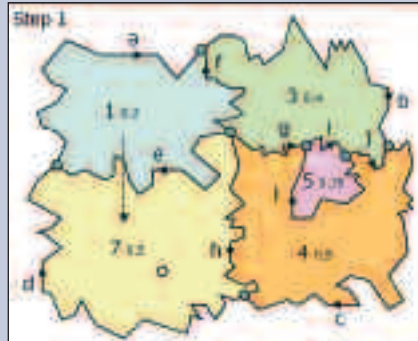
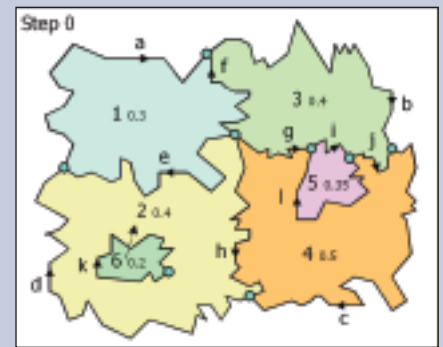
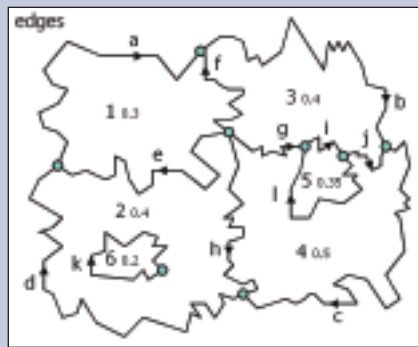
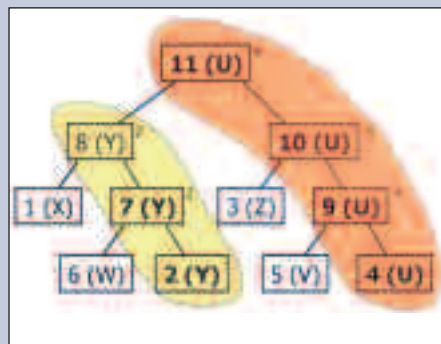
- de eerste benadering is een kaart waar het object van belang (plaats of route) op hoger detailniveau (grootschalig) wordt getoond en de omringende objecten worden getoond met minder detail (kleinschalig) zodat dit resulteert in een niet-uniforme schaal binnen één beeld ('fish-eye' lens idee);
- de tweede benadering is gebaseerd op het geleidelijk zoomen en pannen: op het startmoment heeft de kaart een bepaalde schaal maar in een animatiestijl van visualisatie wordt de kaart continu aangepast richting de doel-schaal (zoomen) of doelgebied (pannen).

De tGAP-structuur kan beide oplossingen ondersteunen. De tweede benadering, in combinatie met progressieve data-overdracht en geleidelijk zoomen, lijkt het meest geschikt voor een mobiele toepassing. Door bij een inzoomactie steeds meer details, in een goede volgorde, over te sturen ('streamen') kan de client deze gebruiken voor het geleidelijk zoomen. De client heeft een gedeeltelijk gevulde kopie van de tGAP-structuur (die dus continu wordt aangevuld met nieuwe objecten en/of details). Zeer frequent wordt de tGAP-structuur uitgelezen en van de gewenste (tussen-)zoom/gebied stap wordt een visualisatie afgeleid. Zodra deze visualisatie gereed is, vervangt die in één keer het oude beeld. Dit resulteert in geleidelijke zoom-weergaven, waarbij de gebruikers het context-gevoel houden en niet gedesorienteerd raken. De ene weergave sluit namelijk goed aan op de volgende: veel kleine stapjes maken dan samen één grotere zoom-actie compleet die wordt weergegeven in de vorm van een animatie. Uitzoomen en pannen werken op soortgelijke manier.

Experimenten zullen moeten aantonen of deze aanpak echt geleidelijk overkomt en of de gebruikers hierdoor minder snel het context-gevoel kwijt raken. Door de abrupte aard van sommige tGAP-structuur acties (bijvoorbeeld het verwijderen/toevoegen van een extra punt op een lijn of het samenvoegen/splitsen van twee vlakken) is het denkbaar dat het kaartbeeld toch schokkerig overkomt (hoewel het op deze beeldschermgrootte/schaal niet erg zichtbaar zou moeten zijn). Hiervoor zijn ook weer oplossingen voorhanden: verzamel een aantal tGAP-structuurverfijningen, toon deze dan echter niet in één keer maar via nog kleinere tussenstapjes. In geval van het toevoegen van een punt: laat dit punt dan beginnen ergens midden op de rechte lijn en schuif het daarna geleidelijk naar de eigen positie. In geval van het toevoegen van een nieuw vlak: laat dat klein beginnen (mogelijk zelfs als punt) en geleidelijk groeien tot het eigen vlak. De hier aangegeven technieken vereisen dus een intelligente client (met eigen schaduw tGAP-structuur die geleidelijk aan steeds meer gevuld zal raken met gegevens van de server).

Opbouw van de tGAP-structuur

De serie plaatjes toont hoe in een aantal stappen de tGAP-structuur wordt opgebouwd door het herhaaldelijk verwijderen van het minst belangrijke vlak (en dit samen te voegen met de meest geschikte buur). Van elke stap zijn twee plaatjes weergegeven: de eerste geeft puur het samenvoegen van vlakken weer, de tweede geeft tevens het effect van de lijngeneralisatie weer. Uiteindelijk blijft dan één, het meest belangrijke, vlak over. De GAP-face geeft weer welke vlak in welk vlak is overgegaan via de ouder-kind relaties. Aangezien in elke stap twee vlakken worden samengevoegd (net zolang totdat het laatste vlak is bereikt), is de resulterende structuur een binaire boom, zoals weergegeven in de onderstaande figuur.



De tGAP-structuur is, in deze opzet met een variabele schaal geo-informatie server en een intelligente client, in staat om ook dergelijke geavanceerde visualisatie-toepassingen te ondersteunen.

Conclusies

De voordelen van de gekozen tGAP-structuur zijn de volgende:

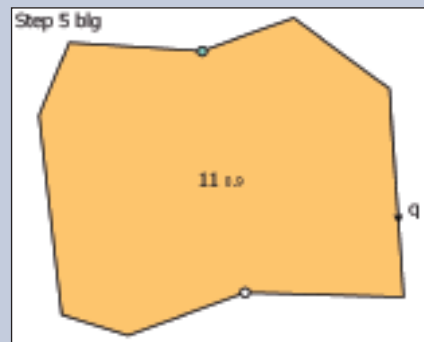
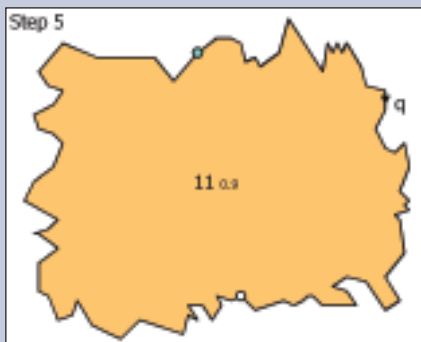
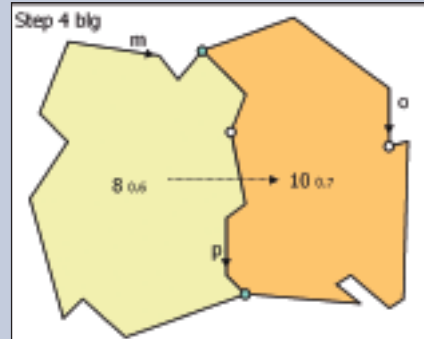
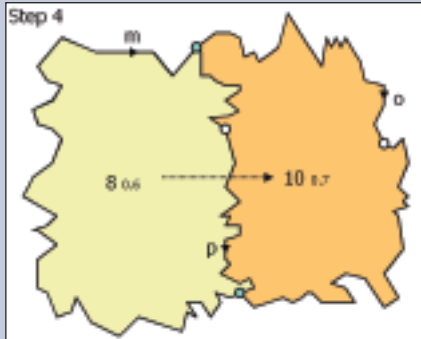
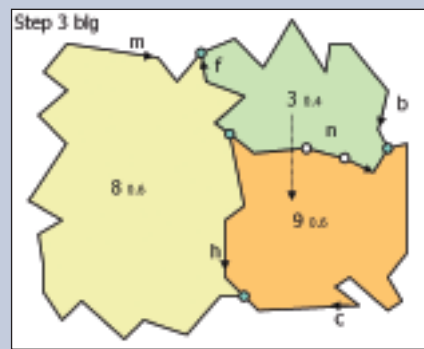
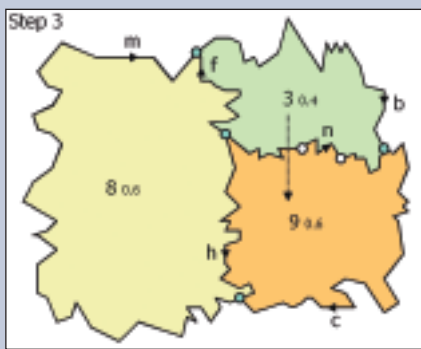
- het is mogelijk om, na opbouwen van de tGAP-structuur, zeer efficiënt (in 'real time') representaties op elke gewenste schaal af te leiden;
- al overgestuurd materiaal kan worden hergebruikt zodat progressieve data-overdracht mogelijk is;
- geleidelijk zoomen en pannen is mogelijk door de progressieve data-overdracht van de informatie en een (geavanceerde) client die dit omzet in een animatiestijl zoom- of pan-weergave;
- meerdere tGAP-structuren kunnen op dezelfde broninformatie worden gedefinieerd, waardoor dezelfde geo-informatie bruikbaar is voor verschillende toepassingen. Voor het bouwen van meerdere tGAP-structuren (indexen) hoeft alleen per toepassing te worden aangegeven hoe zwaar alle geografische klassen ten opzichte van elkaar wegen en hoe de compatibiliteitsmatrix tussen de verschillende klassen eruit ziet.

In lopend onderzoek trachten we, op basis van de eerste ervaringen, de tGAP-structuur nog verder te verbeteren en uit te breiden:

- het geheel vormt een flinke brij van verwijzingen, die ook moet worden opgeslagen en bijgehouden. Dus als er wat te besparen valt op het aantal verwijzingen zonder dat dit

ten koste van functionaliteit en/of performance gaat dan is dit zeker winst ten opzichte van de huidige implementatie;

- momenteel is de datastructuur statisch: er kan niet op de datastructuur worden gemuteerd, wat inhoudt dat mutaties op het basismateriaal geen weerslag hebben op de tGAP-structuur. Als de structuur dynamisch wordt, kan deze worden bijgehouden bij het muteren van de basisdata, alhoewel dit niet eenvoudig is: hiervoor is meer onderzoek nodig;
- naast vlakobjecten zouden ook lijnen en puntobjecten opgeslagen moeten kunnen worden, zowel op het basismateriaal (meest gedetailleerde weergave) als op het afgeleide niveau (een object zou moeten kunnen veranderen van vlakobject naar lijnobject). Op die manier vormt de tGAP-structuur een flexibele basis voor meerdere typen generalisatie: niet alleen aggregatie van objecten tot grotere eenheden en lijnsimplificatie, maar ook het veranderen van representatie of het verplaatsen van objecten als dat de kaartweergave ten goede komt. ■



Literatuur

- Bertolotto, M. and M.J. Egenhofer (2001), *Progressive Transmission of Vector Map Data over the World Wide Web*. *GeoInformatica*, 5 (4): 345-373.
- Buttenfield, B.P. (2002), *Transmitting Vector Geospatial Data across the Internet*. In: *Proceedings GIScience 2002*, Egenhofer, M.J. and Mark, D.M (eds.). Berlin: Springer Verlag, *Lecture Notes in Computer Science* 2478, p. 51-64.
- Jones, C. B., A.I. Abdelmoty, M.E. Lonergan, P.M. Van Der Poorten, and S. Zhou, (2000), [Multi-Scale Spatial Database Design for Online Generalisation]. In: *Proceedings 9th International Symposium on Spatial Data Handling*, sec. 7b, p. 34-44.
- Meijers, B.M. (2006), *Implementation and testing of variable scale topological data structures - Experiences with the GAP-face tree and GAP-edge forest*. TU Delft, MSc Geomatics thesis, juni 2006.
- Oosterom, P. van (2005), *Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest*, *Cartography and Geographic Information Science*, 32 (4): 331-346.
- Zhou, X., Prasher, S., Sun, S. and Xu, K. (2004), *Multiresolution Spatial Databases: Making Web-based Spatial Applications Faster*. In: *Proceedings Sixth Asia Pacific Web Conference (APWeb'04)*, 14-17 april, 2004, Hangzhou, China, *Lecture Notes in Computer Science* 3007, Editors: Jeffrey Xu Yu, Xuemin Lin, Hongjun Lu, et al, p. 36-47.

Samenvatting

Geo-informatie op variabele schaal voor mobiel internet-GIS

Er is een groeiende behoefte om geografische data op meerdere schalen aan eindgebruikers te kunnen aanbieden. Daarvoor is datareductie nodig. Generalisatie van geografische informatie is een mogelijk middel om deze reductie te realiseren.

Binnen veel huidige structuren werd de geometrie tot nu toe redundant opgeslagen. Daarom is er een nieuw conceptueel model uitgewerkt met boomstructuren voor GAP-faces en voor de GAP-edges. De complete structuur is 'tGAP-structuur' gedoopt, en onder meer getest op een prototype. Het blijkt mogelijk om binnen de viewer interactief elke willekeurige uitsnede van een gebied te bekijken.

Summary

Geo Information at variable scale for mobile internet-GIS

There is an increasing demand to provide end users with geographic data at multiple scales. This requires data reduction. Generalisation of geographic data is a possible means to realise this reduction. The object geometry is often stored redundantly in current data structures. Hence, a new conceptual model has been devised using tree structures for GAP faces and GAP edges. The complete structure was baptised "tGAP-structure" and was tested on a prototype. From within the viewer any area subset can be interactively explored.