Een verdeel-en-heers aanpak om voor grote datasets een vario-schaal structuur te creëren

Suba, R.; Meijers, Martijn; van Oosterom, P.J.M.

Publication date
2017

Document Version
Final published version

Published in
Geo-Info

Een verdeel-en-heers aanp een vario-schaal structuur

Zo'n vijf jaar geleden is in Geo-Info het concept van vario-schaal geo-informatie beschreven (Van Oosterom en Meijers, 2012). In dit eerdere artikel werd de eerste echt geleidelijke vario-schaal structuur gepresenteerd: een delta schaal geeft een delta in de kaart (en hoe kleiner de delta schaal hoe kleiner de delta kaart). De afgelopen vijf jaar is er veel R&D verricht om het concept van vario-schaal geo-informatie te realiseren: ontwikkelen van prototypen en testen met echte data. In het kader van het Open Technologieprogramma (OTP van STW, Stichting Technische Wetenschappen) project 11185 'Vario-scale geo-information' is er de afgelopen jaren veel vooruitgang geboekt. De belangrijkste resultaten zullen in een serie beknopte artikelen worden behandeld. Dit is het derde artikel in de serie.

Door Radan Šuba, Martijn Meijers en Peter van Oosterom

Grote datasets met geografische gegevens in vector formaat met een verdeel-en-heers strategie verwerken, is niet eenvoudig. Dit geldt zeker voor het probleem van kaartgeneralisatie, waar (relaties tussen) nabije objecten in ogenschouw moeten worden genomen. Om een vario-schaal structuur af te leiden, waarbij de originele set aan kaartobjecten te groot is om in één keer in het hoofdgeheugen van een computer te laden, is het nodig de input in kleinere blokken op te delen. Daarvoor hebben we een verdeel-en-heers aanpak ontwikkeld, zodat we – onafhankelijk van de grootte van de input – de hele dataset toch om kunnen zetten naar een vario-schaal structuur (waarna deze gebruikt kan worden, bijvoorbeeld in een client-server architectuur, zie het tweede artikel in deze serie, Huang, 2017 – Geo-Info nummer 2, 2017).

Om enorme datasets aan te kunnen, is het dus nodig dat de input wordt opgesplitst in kleinere blokken met data. Deze blokken noemen we 'velden'. Door de kaartobjecten op te delen met behulp van deze velden, wordt de hoeveelheid gegevens die voor elk veld verwerkt moet worden, beperkt. Bijkomstigheid is dat de velden onafhankelijk van elkaar en tevens ook tegelijkertijd kunnen worden behandeld. Dit sluit goed aan op de huidige techniek van multikernprocessoren, waardoor – in vergelijking zonder verdeel-en-heers aanpak – significante tijdswinst te behalen valt. Daartegenover staat dat de kaartobjecten die aan/over de rand van de velden liggen speciale aandacht vereisen.

Aan het ontwerp van de verdeel-en-heers aanpak hebben we de volgende eisen gesteld:

- De aanpak moet onafhankelijk werken van het type kaart dat gegeneraliseerd moet worden.
- De velden (partitionering) moeten zonder tussenkomst van een gebruiker automatisch gegenereerd kunnen worden.
- Elk kaartobject moet slechts eenmalig gegeneraliseerd worden.
- De aanpak moet parallelle verwerking mogelijk maken.
- De gegevens in de resulterende vario-schaal structuur moeten er gelijk of nagenoeg gelijk uitzien qua vulling, als wanneer deze zonder verdeel-en-heers aanpak in één keer worden verkregen.

We zullen nu eerst de aanpak in detail uitleggen, waarna we een aantal gedane experimenten met resultaten zullen toelichten.

De verdeel-en-heers aanpak

Het hele proces om een grote vario-schaal datastructuur gevuld te krijgen, bestaat uit een aantal stappen:

1. het maken van een partitioneringsgrid met velden op meerdere niveaus (Fieldtree) en de input hierover verdelen,
2. de velden een voor een generaliseren,
3. de afronding.

1. Fieldtree maken en objecten verdelen

Om de planaire partitie met nodes, edges en faces in kleinere blokken te verdelen, maken we gebruik van de zogenaamde Fieldtree datastructuur (Frank en Barrera, 1990). De Fieldtree is ontwikkeld voor GIS-toepassingen. Figuur 1 toont dat de velden op meerdere niveaus zijn georganiseerd met steeds verschoven oorsprong voor een niveau. Alle velden op één niveau vormen samen een grid. De structuur wordt compleet vastgelegd door een ingegeven parameter die aangeeft hoe groot de velden op het laagste niveau zijn en de geografische omvang (extent) van de dataset (zie Figuur 1(a)).

Wanneer de lay-out van de Fieldtree vastligt, kunnen de objecten worden verdeeld over de velden. Elk object uit de topologische structuur van de planaire partitie (node, edge of face) wordt toegewezen aan het kleinst mogelijke veld waar het qua omvang in past. In het geval een object groter is dan een veld op het laagste niveau (of deels over de rand ligt), wordt dit object in een veld op een hoger gelegen niveau geplaatst en komt dit object pas later aan de beurt in het generalisatie proces. Door de verschoven grids lukt dit altijd (omdat de grid lijnen steeds verschuiven).

Zodra alle objecten toegewezen zijn aan een veld, kan elk laagstgelegen veld worden gegeneraliseerd. Het generalisatieproces kan onafhankelijk per veld en tegelijk met andere velden worden uitgevoerd. Merk op dat de vlakken (faces) alleen worden gegeneraliseerd als ze compleet binnen een veld liggen (en hier dus ook alle gerelateerde nodes en edges bij aanwezig zijn). Als een vlakobject over de rand heen ligt, komt dit vlak

ak om voor grote datasets te creëren

bij een veld op een hoger niveau aan de beurt. Wanneer alle negen kind-velden verwerkt zijn, kan het één niveau hoger gelegen ouder-veld worden verwerkt. Het proces gaat zo door totdat alle velden compleet gegeneraliseerd zijn.

Doordat de velden op een hoger niveau worden verschoven, kan worden gegarandeerd dat objecten die eerder niet konden worden gegeneraliseerd (omdat ze of te groot waren of op de rand lagen) nu wel aan de buurt kunnen komen. De aanpak probeert om elk veld met een factor 4 te generaliseren. Stel dat de input uit n vlakobjecten (faces) bestaat, dan moet het resultaat van het generalisatieproces van een veld n/4 objecten bevatten. Bij een uniforme dataverdeling wordt zo bereikt dat de hoger gelegen velden net zoveel vlakken te verwerken krijgen als de inputvelden.

2. *Velden een voor een generaliseren*

Nadat de objecten verdeeld zijn over de velden, kan een veld gegeneraliseerd worden. Dit deel van het proces kost de meeste verwerkingstijd. Ons automatische generalisatieproces is gebaseerd op het vinden van het minst belangrijke kaartobject, wat vervolgens gegeneraliseerd wordt. Dit globale criterium verandert door de verdeel-en-heers aanpak: in plaats van globaal is het nu een 'lokaal veld' criterium. Dit is zinvol, omdat bijvoorbeeld de kaartobjecten in het noorden geen directe relatie hebben met de objecten in het zuiden van de dataset. Generalisatie-acties die we in ons proces geïmplementeerd hebben, zijn onder andere dat vlakken worden samengevoegd, hun grenzen worden versimpeld en vlakken omgezet worden naar een lijnrepresentatie en worden verdeeld over hun naaste buren.

Tijdens het verwerken van een veld kunnen twee verschillende categorieën van objecten worden onderscheiden. Ten eerste; objecten die niet meer geldig zijn voor de kaartschaal die voor het huidige veld werd bereikt door het generalisatieproces. Deze objecten zijn gereed (voor hun kaartschaal range) en worden opgeslagen voor de uiteindelijke vario-schaal structuur. Ten tweede; objecten die nog verder versimpeld moeten worden. Deze objecten worden in een veld van de Fieldtree op een hoger gelegen niveau geplaatst – wederom door ze in het kleinste, nog te verwerken veld te plaatsen waar ze compleet in passen. Tezamen met de objecten die over de rand lagen op het lager gelegen niveau aan velden, vormen deze objecten zo samen weer een (deels versimpelde) planaire partitie.

3. *Afronding*

Nadat alle velden verwerkt zijn, vindt de laatste operatie van het proces plaats: het combineren van de informatie over alle definitieve objecten uit de verschillende velden. De individuele tabellen per veld worden samengevoegd om één set aan database tabellen te maken (een node, edge en face tabel), die samen de vario-schaal structuur vormen. Ook wordt in deze stap ruimtelijke indexering toegepast, zodat de resulterende structuur snel te bevragen is.

Resultaten

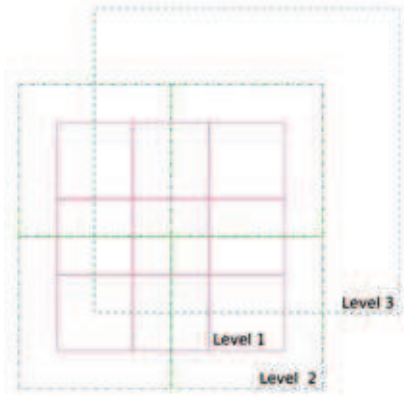
We hebben onze aanpak getest met drie datasets, die behoorlijk verschillen in inhoud. De datasets die we gebruikt hebben zijn:

- CORINE-landcover bestand voor het Verenigd Koninkrijk en Ierland (met 100.000 vlakobjecten) en rond Estland (met zo'n 130.000 vlakobjecten).
- Het Nationaal Wegen Bestand, de vlakken tussen de wegen werden gevormd door lijnen van het wegennetwerk te gebruiken. Ongeveer 200.000 vlakobjecten.
- Een kadastrale dataset met percelen van provincie Gelderland met ongeveer 880.000 vlakobjecten.

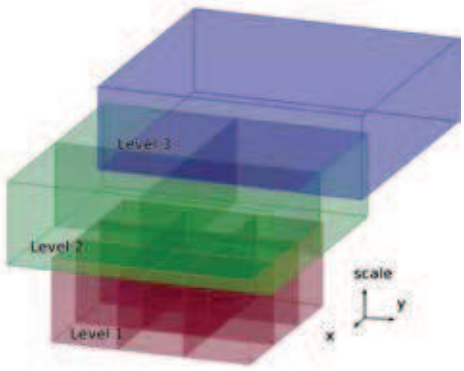
Om inzicht te verkrijgen in wat een redelijke veldgrootte is voor de kleinste velden in de Fieldtree, hebben we het hele proces (Fieldtree maken en objecten verdelen, veld voor veld generaliseren, afronden) met verschillende veldgroottes herhaald. Hierbij varieerden we de veldgroottes gebaseerd op de gemiddelde grootte (lengte van zijn langste zijde) van een vlakobject in de dataset.

```
NRLEVELS = 8      # Gewenst aantal niveaus in de Fieldtree
FINEGRID = 12500  # Veldgrootte (m) meest gedetailleerde niveau
THEORIGX = 0      # Oorsprong (in x- and y-richting)
THEORIGY = 0

for (i=0; i<NRLEVELS; i++) do
   SIZE[i] = FINEGRID * 2^i

ORIGX[NRLEVELS-1] = THEORIGX
ORIGY[NRLEVELS-1] = THEORIGY
for (i = NRLEVELS-2; i>=0; i--) do
   ORIGX[i] = ORIGX[i+1] - SIZE[i+1]/4
   ORIGY[i] = ORIGY[i+1] - SIZE[i+1]/4
```

(a) Pseudo code om de lay out van een Fieldtree vast te leggen (Van Oosterom en Vijlbrief, 1996). Het aantal benodigde niveaus kan worden afgeleid vanuit de extent van de dataset en de parameter van veldgrootte op het meest gedetailleerde niveau.



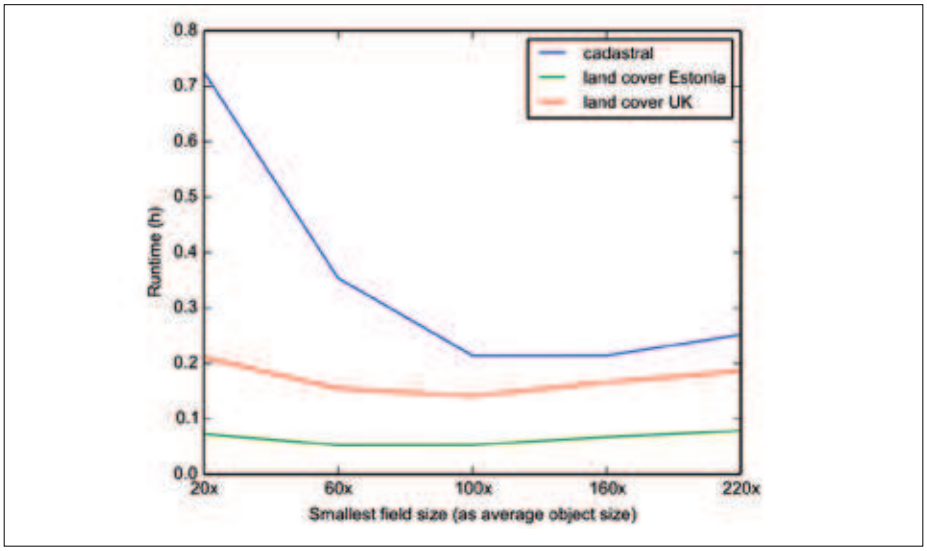
(b) 3 niveaus van de Fieldtree, laagste niveau in roze, middelste in groen en hoogste niveau in blauw.



(c) 3D impressie van de velden. Merk op dat de 3e dimensie schaal is.

Figuur 1 (a, b en c) - Fieldtree.

We hebben groottes gebruikt van 20 tot 220 keer de gemiddelde vlakgrootte. De grafiek van Figuur 2 laat zien dat we, als vuistregel, de kleinste velden ongeveer 100 keer de gemiddelde vlakgrootte moeten geven (dus zo'n 10.000 objecten per veld). Dan is de benodigde tijd voor het hele proces optimaal. Verder kunnen we concluderen dat als de veldgrootte te klein gekozen wordt, dit leidt tot vrij veel overhead. Er wordt dan onnodig veel tijd besteed aan het kopiëren van data van lager gelegen velden naar hoger gelegen velden, terwijl het generalisatieproces niet veel kan versimpelen (veel objecten die over de veldrand liggen op lagere niveaus).



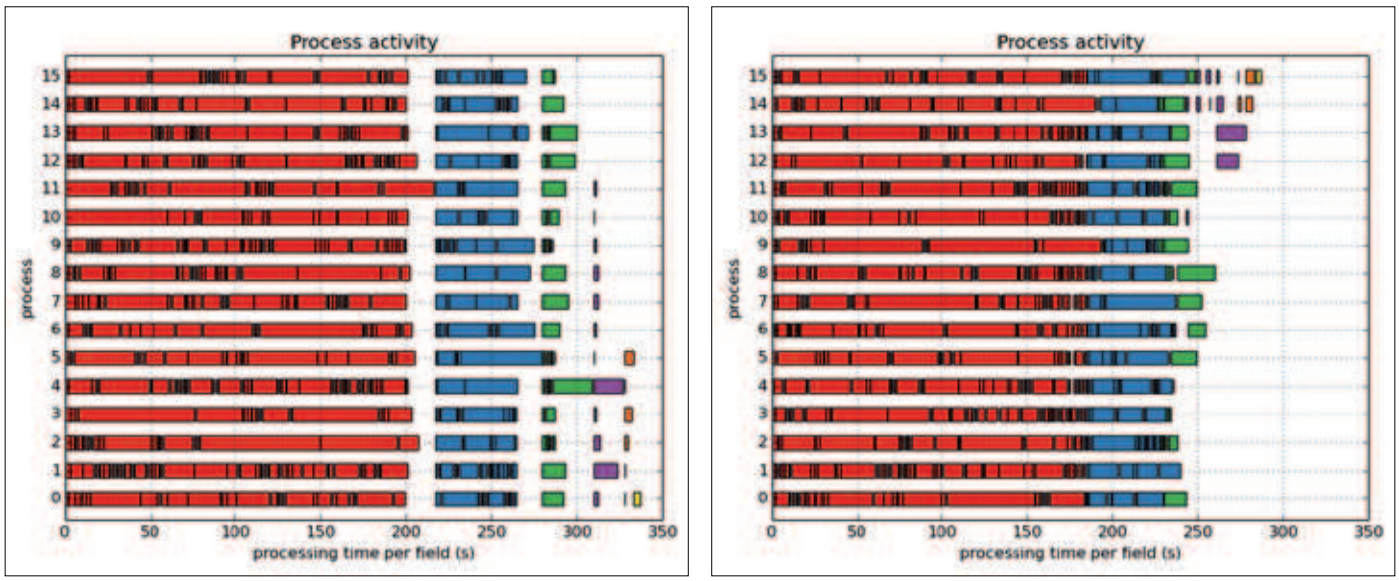
Figuur 2 – Benodigde rekentijd versus veldgrootte van het laagste niveau.

De aanpak biedt twee mogelijke manieren voor het verwerken van velden ('schedulen'). Bij de eerste manier (Figuur 3(a)) worden eerst alle velden op één niveau compleet verwerkt. Pas als voor dit niveau alle velden klaar zijn, wordt gestart met velden van het volgende niveau. Bij de tweede manier (Figuur 3(b)) wordt gebruik gemaakt van de ouder-kind relatie die velden in de Fieldtree hebben en wordt een ouder gepland om verwerkt te worden, als de negen kind-velden compleet gegeneraliseerd zijn. We hadden verwacht dat de tweede strategie behoorlijk wat sneller zou kunnen zijn, maar hier was slechts sprake van een beperkte reductie van benodigde rekentijd (van 340 naar 290 seconden, wat een reductie betekent van 15%).

Naast deze statistieken, wilden we ook inzichtelijk krijgen of het resultaat van het generalisatieproces kwalitatief beïnvloed wordt door de verdeel-en-heers aanpak. Hiervoor hebben we dezelfde dataset omgezet naar vario-schaal structuur, zowel met onze verdeel-en-heers aanpak versimpeld, als ook zonder de verdeel-en-heers aanpak toe te passen (als één grote taak). Figuur 4 toont dat er in dit geval geen opmerkelijk grote verschillen zijn waar te nemen tussen de twee resultaten.

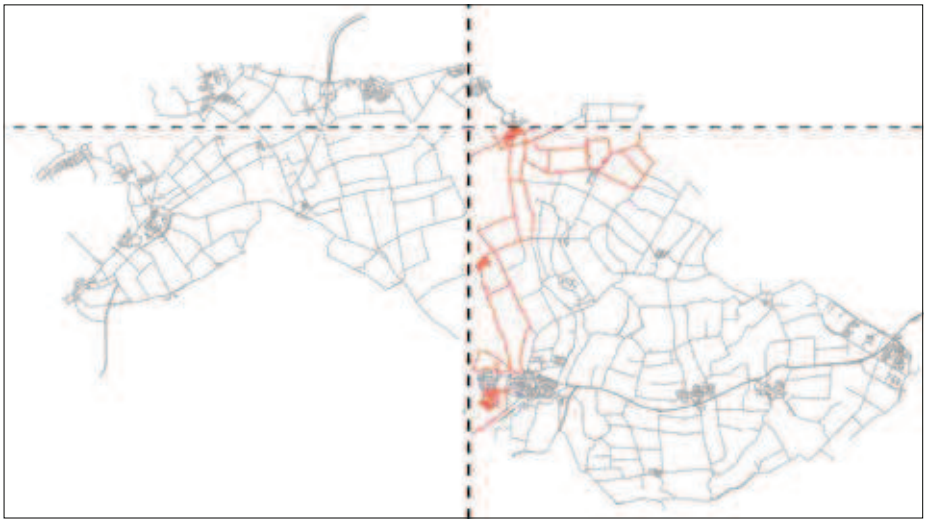
Bij het toepassen van de verdeel-en-heers aanpak op het generalisatieproces dat we voor wegen hadden ontwikkeld (zie eerste artikel in deze serie, Šuba, et al. 2017 – Geo-Info nummer 1, 2017) bleek dat de verdeling van kaartobjecten over de velden een probleem vormt voor de analyses waar buurobjecten moeten worden meegenomen. Om dit op te lossen, stellen we binnen een veld een 'alleen lezen'-buffer in. De objecten die te dichtbij de rand van het veld liggen, mogen deze ronde zelf niet meedoen in het generalisatieproces (ze worden dus op dit niveau zelf nog niet versimpeld), maar kunnen wel gebruikt worden bij analyses voor de generalisatie van andere objecten. Te dichtbij de rand wordt in dit geval bepaald met behulp van een topologische maat (Figuur 5 toont dat alle objecten in het rood die binnen twee stappen vanaf de rand van het veld afliggen nog niet worden versimpeld). Een andere optie de we hebben overwogen (maar niet geïmplementeerd) is het toevoegen van een buffer aan elk veld, zodat voor deze buffer vervolgens 'alleen lezen' objecten beschikbaar zijn. Het voordeel is dat er per veld meer gegeneraliseerd kan worden, nadeel is dat er meer data per veld (van niveau naar niveau) gekopieerd moet worden.

Door te testen met de verschillende datasets hebben we laten zien dat de aanpak in principe werkt op verschillende soorten data. Verder laat het testen met de wegenkaart zien dat het ook mogelijk is om buurobjecten tot de beschikking te hebben (als dit nodig is voor het generalisatie-



Figuur 3 - Verschillende manieren om velden te verwerken. Merk op dat de verschillende kleuren overeenkomen met verschillende niveaus in de Fieldtree (rood komt overeen met laagste niveau).

Figuur 4 - Test data, input en output (met en zonder verdeel-en-heers aanpak).



Figuur 5 - Toepassen van een inwaartse 'alleen lezen' rand. Objecten die op dit niveau voor het veld rechtsonder niet versimpeld mogen worden, worden in het rood getoond.

proces). Met de aanpak kunnen velden in parallel worden verwerkt, terwijl elk kaartobject slechts eenmaal wordt gegeneraliseerd.

Conclusie

We hebben een verdeel-en-heers aanpak gepresenteerd die gebruikt kan worden om voor zeer grote datasets een vario-schaal structuur te vullen met data. We hebben laten zien dat de aanpak toegepast kan worden op verschillende soorten input data, zoals CORINE-landcover of een kaart van een wegennetwerk (Nationaal Wegen Bestand). De aanpak maakt het mogelijk dat het generalisatieproces in parallel uitgevoerd kan worden. Hierbij is de Fieldtree een zeer goed uitgangspunt gebleken: naast het verdelen van de objecten over velden beschikt de Fieldtree ook over meerdere niveaus en worden de velden per niveau 'slim' verschoven. De objecten die niet gegeneraliseerd kunnen worden op een lager gelegen niveau, worden versimpeld op een hoger niveau (waarbij meerdere delen van een kaartobject weer aan elkaar gelegd worden). Deze Fieldtree gebaseerde verdeel-en-heers aanpak past goed bij het generalisatieprobleem, maar is mogelijk ook bruikbaar voor andere ruimtelijke problemen (bijvoorbeeld het produceren van een datastructuur met expliciete topologie). Mogelijk kan de Fieldtree tevens behulpzaam zijn bij het updaten van onze vario-schaal structuur. Door het 'lokale veld' criterium in plaats van het globale criterium en door van de objecten de relatie met de Fieldtree velden te onthouden, kan bepaald worden welk deel van de vario-schaal structuur opnieuw moet worden gegeneraliseerd. Dit is werk voor toekomstig onderzoek.

Bronnen

- Peter van Oosterom, Martijn Meijers, Variabele-schaal geo-informatie, Geo-Info, 9(10), pp. 14-19, 2012.
- Radan Šuba, Martijn Meijers, Peter van Oosterom, Wegennetwerken in vario-schaal structuren, Geo-Info, 14(1), pp. 44-48, 2017.
- Lina Huang, Martijn Meijers, Radan Šuba, Peter van Oosterom, Vario-schaal gegevens in een Geoweb context, Geo-Info, 14(2), pp. 74-78, 2017.
- Peter van Oosterom, Tom Vijlbrief, The Spatial Location Code, Proceedings of the 7th International Symposium on Spatial Data Handling, SDH'96, Delft, August, 1996.
- Frank, A. U., Barrera, R., The Fieldtree: A data structure for Geographic Information Systems. Proceedings of the First Symposium on Design and Implementation of Large Spatial Databases, SSD '90, Springer-Verlag New York, Inc., New York, NY, USA, pp. 29–44, 1990.
- Judith van Putten, Peter van Oosterom, Generaliseren van vlakkenpartities (2); GAP-trees, testresultaten en verbeteringen, Geodesia, 42(11), pp. 499-505, 2000.

Dit artikel is een bewerking van het Engelstalige artikel: Martijn Meijers, Radan Šuba, Peter van Oosterom, Parallel Creation of Vario-Scale Data Structures for Large Datasets, In: ISPRS Archives Volume XL-4/W7, 4th ISPRS International Workshop on Web Mapping and Geoprocessing Services, Sardinia, pp. 1-9, 2015.

Radan Šuba is promovendus GIS technologie bij de TU Delft. Hij is bereikbaar via R.Suba@tudelft.nl.

Martijn Meijers is onderzoeker GIS technologie bij de TU Delft. Hij is bereikbaar via B.M.Meijers@tudelft.nl.

Peter van Oosterom is professor GIS technologie bij de TU Delft. Hij is bereikbaar via P.J.M.vanOosterom@tudelft.nl.