

TECHNIEKEN VOOR DE VISUELE ANALYSE VAN MASSIEVE DATASETS TRAJECTDATA

DOOR DE TRAJECTEN DE FLOWS ZIEN

Aantal woorden: 5410 woorden + 7 figuren

Rien Boydens

Studentennummer: 01700429

Promotor: Prof. dr. Nico Van de Weghe

Bachelorproef voorgelegd voor het behalen van de graad bachelor in de Geografie & Geomatica

Academiejaar: 2020 - 2021

INHOUDSTAFEL

| | |
|---|----|
| 1 Inleiding..... | 1 |
| 2 Visualisatiemethoden..... | 2 |
| 2.1 Space-Time Cube..... | 2 |
| 2.2 Animatie..... | 3 |
| 2.3 Stacking Graph..... | 4 |
| 2.4 Density Map..... | 4 |
| 2.5 Heat map..... | 5 |
| 2.6 Meshing en Ruimtelijke abstracties..... | 5 |
| 2.7 OD-matrix en Flowmap..... | 7 |
| 3 Klassieke informatie extractie..... | 8 |
| 3.1 Selectie en filteren..... | 8 |
| 3.1.1 Filteren op attributen..... | 9 |
| 3.1.2 Filteren op locatie..... | 9 |
| 3.1.3 Filteren op tijd..... | 9 |
| 3.2 Clusteren..... | 9 |
| 3.2.1 Hiërarchisch clusteren..... | 9 |
| 3.2.2 Densiteit gebaseerd clusteren..... | 10 |
| 3.2.3 Afstand functies..... | 11 |
| 3.3 Aggregatie..... | 11 |
| 3.3.1 Aggregatie op locaties..... | 12 |
| 3.3.2 Aggregatie op flows..... | 13 |
| 3.4 Aggregaten gefilterd per cluster..... | 13 |
| 4 Te groot voor RAM..... | 14 |
| 4.1 Subset als classifier..... | 14 |
| 4.2 Sequentiële/parallele berekening..... | 14 |
| 5 Besluit..... | 15 |
| 6 Referentielijst..... | 17 |

FIGUREN

| | |
|--|----|
| Figuur 1 : Statische visualisatie van STC operaties: (a) time cutting; (b) time flattening; (c) discrete time flattening; (d) colored time flattening; (e) time juxtaposing; (f) space cutting; (g) space flattening; (h) repeated drilling; and (i) 3D rendering. Bron: Bach et al. (2017) in He et al. (2019)..... | 3 |
| Figuur 2 : Density map van de schepen voor de haven van Rotterdam. De kleur toont het type schip waarbij blauw passagiersvaart is, rood goederenvervoer en groen tankers. Bron: Scheepens et al. (2011)..... | 4 |
| Figuur 3 : Flows tussen meshcellen op twee verschillende niveaus van abstractie. Bron: N. Andrienko & G. Andrienko (2011)..... | 6 |
| Figuur 4 : Ontwerp evolutie van de Circos diagrammen van Zeng et al. (a) origineel ontwerp binnen de genetica; (b) zwevende ring voor het kruispunt zelf; (c) bundelen van bi-directionele linten; (d) statistische weergeven van de inkomende en uitgaande objecten. Bron: Zeng et al. (2013)..... | 6 |
| Figuur 5 : Voorbeelden om kruisende flows leesbaar te maken. (a) Veel kruisende migratie flows van US Census data; (b) edge routing; (c) edge bundling. Bron: He et al. (2019)..... | 8 |
| Figuur 6 : Aggregaat van vliegtuigtrajecten rond een luchthaven. De kleur van het segment heeft de gemiddelde versnelling van de vliegtuigen weer. Bron: Buschmann et al. (2015)..... | 12 |
| Figuur 7 : Workflow van M ³ . De datapunten worden één per één verwerkt waardoor ze niet allen te samen in het computergeheugen moeten geladen worden. Bron: Graser, Widhalm et al. (2020). 15 | 15 |

SAMENVATTING

Trajectdata wordt steeds groter. Inzichten krijgen in trajectdata kan ook zeer nuttig zijn. Onderzoekers hebben een hele toolset aan technieken om patronen te vinden en inzichten te verwerven in trajectdata. Ten eerste zijn er veel verschillende manieren om trajecten voor te stellen. Voorbeelden hiervan zijn space-time cubes, animaties, density maps, flowmaps, etc. Ten tweede zijn er de technieken om de dataset op een zinnige manier te reduceren. Dit kan door de trajecten te filteren. Een andere manier is om gelijkaardige trajecten te clusteren. Of een laatste manier is aggregaten berekenen van de trajecten. Om met de immens grote datasets die groter zijn dan het werkgeheugen van een computer, stellen hedendaagse wetenschappers voor om datapunt per datapunt te werken. Dit door eerst een classifier te maken van een deel van de data en de rest ermee te cluster. Of door de datapunten sequentieel aan een dynamisch aggregaat toe te voegen.

TREFWOORDEN: trajectdata; visuele analyse; massieve data

1 INLEIDING

Hedendaags worden er steeds meer en meer data verzameld over verplaatsingen van mensen, auto's, schepen, vliegtuigen, etc. Dit gebeurt meestal in de vorm van GPS of andere sensoren zoals AIS. (G. Andrienko et al., 2009) Een inzicht krijgen in deze data kan tot veel voordelen leiden. Ten eerste is er het begrijpen van bewegende objecten in de realiteit. (G. Andrienko et al., 2007) Dit kan zijn het begrijpen van stromen van auto's die het stadscentrum binnenrijden. (Rinzivillo et al., 2008) Of dit kan begrijpen zijn waar de risico op een botsing tussen twee schepen het grootst is. (Scheepens et al., 2011) Ten tweede kan dit nuttig zijn voor de predictie van toekomstige flows, zoals verkeerssituaties. Om ruimtelijke patronen te begrijpen, is het zeer sterk om in te spelen op de visuele perceptie van de mens. (G. Andrienko et al., 2007)

Deze data is vaak in de vorm van een serie van punten met xy-coördinates met een tijdswaarde t en een identificatie voor het object dat tijdens t op x , y was. Dit kan ook nog andere velden bevatten zoals de richting en snelheid van de verplaatsing. (G. Andrienko et al., 2007) G. Andrienko et al. (2007) beschrijven een methode voor het extraheren van trips uit deze ruwe xyt-data op basis van tijd tussen de punten en stop bestaande uit veel opeenvolgende punten op één locatie in een langere tijdspanne. Een traject is dan zo'n trip voorgesteld door een verzameling van punten, lijnsegmenten en sub-trajecten. (Ding et al., 2018) Graser (2018) beschrijft en vergelijkt verschillende manieren om trajecten in een computersysteem op te slaan waaronder de OGC standaarden en als een PostGIS LinestringM met de tijdswaarde als M.

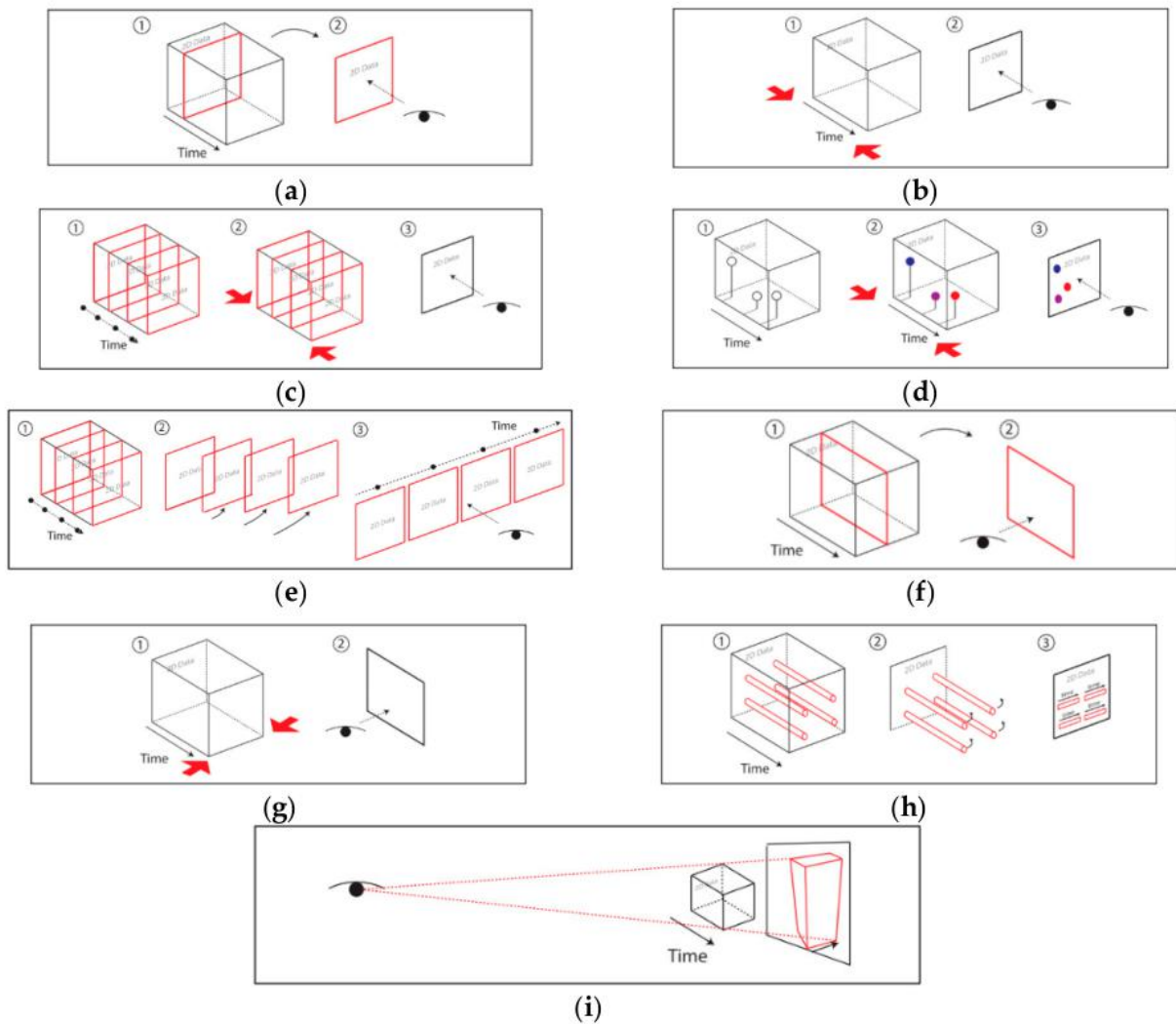
Deze paper bekijkt hoe verschillende auteurs massieve datasets trajectdata visueel analyseren. Het bekijkt hoe dat je in een net van vele overlappende trajecten visueel patronen en inzichten kan vervatten. Ook spendeert het nog wat extra aandacht aan hoe men in hedendaags onderzoek omgaat met steeds grotere wordende datasets van trajectdata. Hoofdstuk 2 van deze paper bespreekt verschillende visualisatiemethoden waarmee men trajectdata kan voorstellen. Hoofdstuk 3 kijkt naar technieken voor het reduceren van de dataset om door de trajecten ook de flows te kunnen zien. Hoofdstuk 4 onderzoekt dan nog hoe dat auteurs omgaan met de limitaties waarbij het werkgeheugen van de computer of RAM kleiner is dan de grote van de dataset.

2 VISUALISATIEMETHODEN

Voor de visuele analyse van trajectdata moet de data visueel worden voorgesteld. De data analist heeft verscheidene methoden tot beschikking voor de visualisatie van trajectdata. Dit hoofdstuk behandelt verschillende van deze visualisatiemethoden waaronder space-time cubes, animaties, stacking graphs, density maps, heat maps, meshing, OD-matrices en flow maps. De meeste software pakketten voor de analyse van tijdruimtelijke data implementeren één of meerdere van deze visualisatiemethoden, sommige staan zelfs toe voor een combinatie van verschillende visualisaties. Het linken van verschillende visualisaties op een interactieve manier in softwarepakketten wordt *brushing* genoemd. (G. Andrienko et al., 2008) Het is de rol van de analist om een geschikte visualisatie of combinatie van visualisaties toe te passen om de gewenste inzichten te bekomen.

2.1 Space-Time Cube

Een *Space-Time Cube* (STC) of Tijdsruimtekubus is een representatie van van tijdruimtelijke processen waarbij de ruimte wordt geprojecteerd op het xy-vlak en tijd op de z-dimensie wordt voorgesteld. (He et al., 2019) Een STC bouwt op Hägerstrand tijdgeografisch kader voorgelegd in zijn werk "*What About People in Regional Science?*". (Kraak & Huisman, 2009) Andrienko et al. (2017) geven statische en dynamische operaties voor het veranderen van het viewpoint van de STC. Statische visualisaties tonen tijdruimtelijke data op een niet-veranderende manier voor. Voorbeelden hiervan zijn *time cutting*, *time flattening*, *time juxtaposing*, *space cutting*, *space flattening*, *repeated drilling* en *3D rendering*. (Figuur 1) (Bach et al., 2017) Dynamische visualisaties veranderen de operatie over tijd in de vorm van een animatie. Een klassieke 3D Tijdsruimtekubus wordt makkelijk gemaakt en gemanipuleerd en kan veel inzichten geven. (Ostermann, 2009) Het is echter niet geschikt voor massieve datasets aangezien deze kan leiden tot *cluttering* en overlap van de trajecten. (N. Andrienko & G. Andrienko, 2011)



Figuur 1: Statische visualisatie van STC operaties: (a) time cutting; (b) time flattening; (c) discrete time flattening; (d) colored time flattening; (e) time juxtaposing; (f) space cutting; (g) space flattening; (h) repeated drilling; and (i) 3D rendering. Bron: Bach et al. (2017) in He et al. (2019)

2.2 Animatie

Zoals voorheen aangehaald kan een animatie gezien worden als een dynamische visualisatie van een STC. De meest gebruikte animatie is een verandering in *time cutting* waarbij verschillende sequentiële timeframes na elkaar worden getoond. Animatie speelt op het vermogen van mensen om veranderingen waar te nemen over tijd. Mensen hebben echter een slecht geheugen als het aankomt op overzichten en dit is dus niet zo goed voor data exploratie. (He et al., 2019) Studies tonen aan dat animaties niet noodzakelijk beter zijn dan statische displays en deze dus beter gebruikt worden voor een eerste algemeen overzicht. (G. Andrienko et al., 2008) In vele software pakketten voor de analyse van trajectdata wordt

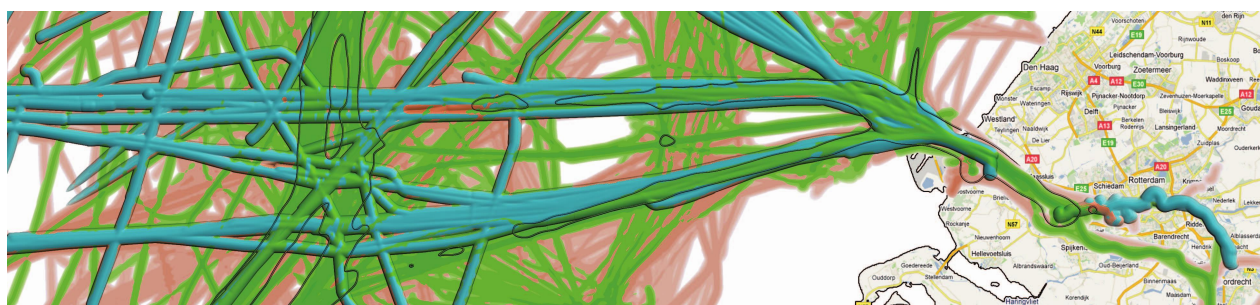
animatie toch interactief gemaakt met behulp van een tijdschuiver. (G. Andrienko et al., 2007; G. Andrienko & N. Andrienko, 2013; N. Andrienko & G. Andrienko, 2005, 2007; Z. Wang et al., 2019)

2.3 Stacking Graph

Het plotten van alle trajecten op een plat vlak kan vaak leiden tot visuele overlap van de trajecten. N. Andrienko & G. Andrienko (2013) zetten de verschillende trajecten in de z-as op elkaar in een stacking graph. Segmenten van deze trajectbanden kleuren ze dan op basis van een attribuutwaarde. Vele onderzoekers hebben ook geprobeerd om de leesbaarheid en verstaanbaarheid van stacking graphs te verbeteren. Voorbeelden hiervan zijn terug te vinden in He et al. (2019). Een specifiek onderdeel van wat traject stacking onderscheid van andere stacking graphs is dat de trajecten langs de z-as geplaatst worden volgens temporele ordening. Hierbij worden de trajecten ook ver genoeg van elkaar geplaatst zodat er geen overlap is tussen verschillende trajecten.

2.4 Density Map

Bij grotere datasets trajectdata is het overzichtelijker om niet alle trajecten individueel voor te stellen. Scheepens et al. (2011) gebruiken een density map om voor de haven van Rotterdam het risico op een accident tussen twee schepen op kaart te brengen. Een density map is een kaart die toont hoe dicht trajecten in een bepaalde omgeving op elkaar liggen. Scheepens et al. (2011) stelden drie verschillende soorten schepen, passagier, cargo en tanker, voor door aan elk soort een ander kleur te geven en de saturatie van het kleur gaf de densiteit weer van dit soort schip op een bepaalde plaats. De visuele analyse kan ook geholpen worden door correcte schaduwing van de density map waardoor densere regio's voller lijken dan minder dense regio's.



Figuur 2: Density map van de schepen voor de haven van Rotterdam. De kleur toont het type schip waarbij blauw passagiersvaart is, rood goederenvervoer en groen tankers. Bron: Scheepens et al. (2011)

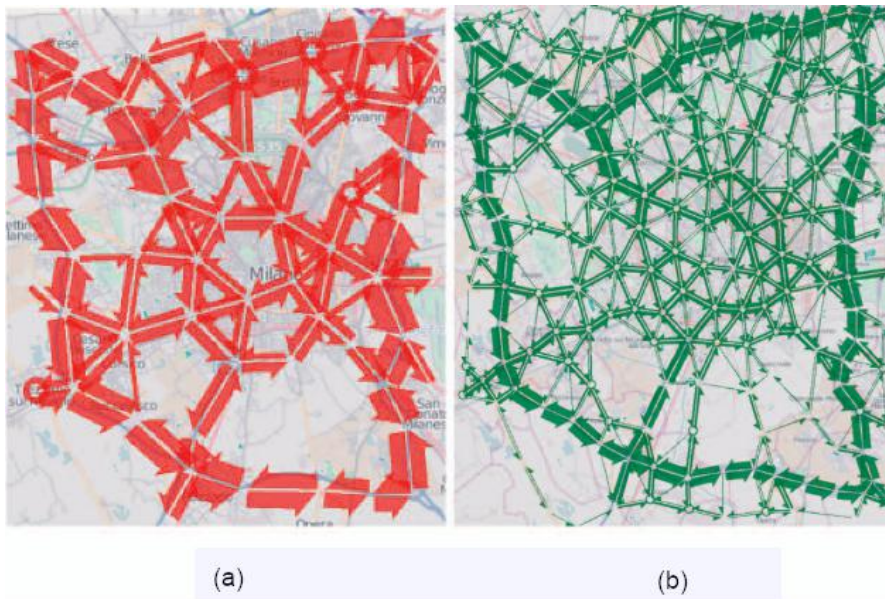
Een density map hoeft niet enkel op een plat XY-vlak te zijn. Du et al. (2016) tonen de trajecten van studenten in een STC voor het modelleren van een epidemie. Density maps in een STC tonen dan plaatsen en tijdstippen waar de epidemie het makkelijkst kan verspreiden. Demšar & Virrantaus (2010) breiden de 2D densiteit uit naar 3D tijdsruimte densiteit. Hierbij berekenen ze eerst de densiteit per traject als de invloed van dit traject op zijn omgeving, en dan voegen ze al deze densiteiten te samen.

2.5 Heat map

Naast een *density map* kan ook een *heat map* gebruikt worden om grotere datasets overzichtelijker te maken. Een *heat map* aggregereert puntdata in een continue vorm door de densiteit van punten op een locatie door kleuren of transparantie voor te stellen. (He et al., 2019) Een *heat map* is dus een soort *density map* voor puntdata waarbij kleuren worden gebruikt in plaats van saturatie. (Buschmann et al., 2015) *Heat maps* zijn niet zo goed in het voorstellen van volledige trajecten aangezien ze werken met puntdata. Ze kunnen wel gecombineerd worden met animaties om zo de densiteit over tijd voor te stellen en zo meer inzichten te krijgen over bijvoorbeeld verkeersdrukke. (He et al., 2019). Graser (2021) vraagt wel om aandachtig te zijn bij het gebruiken van *heat maps* aangezien deze makkelijk outliers in de dataset kunnen verbergen.

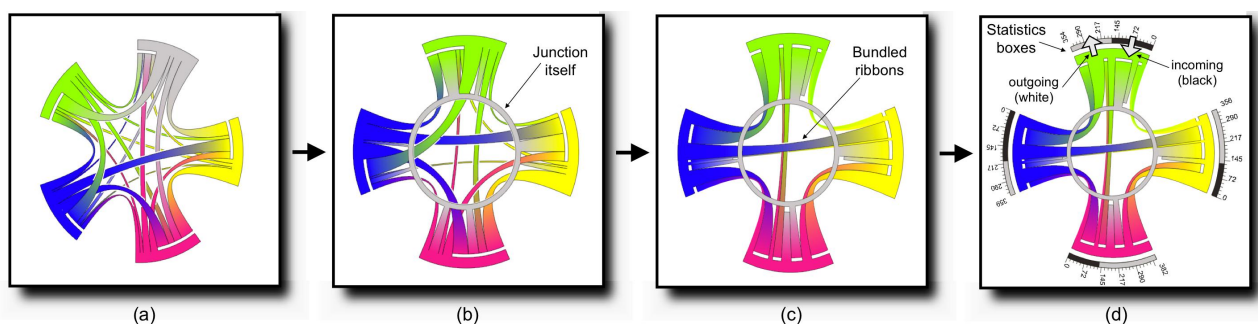
2.6 Meshing en Ruimtelijke abstracties

Meshing houdt in dat de ruimte wordt onderverdeeld in mesh cellen. Als de trajectdata in één zo'n cel homogeen is wordt het direct voorgesteld door één attribuut, anders wordt deze opgesplitst in meerdere cellen. (He et al., 2019) N. Andrienko & G. Andrienko (2007) geven een voorbeeld van meshing waarbij de het aantal ooevaars bij hun migratie noordwaarts per grid cel wordt voorgesteld door de cirkels met verschillende grootte. Een ander voorbeeld dat de auteurs geven is een grids waar het aantal punten wordt voorgesteld door hoe opgevuld een grid cel is. De voorstelling moet ook niet enkel aantal zijn, maar kan ook meerdere statistische gegevens per grid cel zijn. N. Andrienko & G. Andrienko (2007) geven ook een voorbeeld met driehoeken waarbij de hoogte van de driehoek de gemiddelde waarde geeft en de breedte de variantie. G. Andrienko & N. Andrienko (2008) vullen de mesh cellen met balken die de kompasrichting van de dominante flows weergeeft en waarvan de lengte van de balken de het aantal datapunten van die flow toont. Deze balkvoorstelling wordt ook wel een chorisogram genoemd.



Figuur 3: Flows tussen meshcellen op twee verschillende niveaus van abstractie. Bron: N. Andrienko & G. Andrienko (2011)

Als de meshing wordt gegeneraliseerd wordt er gesproken van ruimtelijke abstractie. Ruimtelijke abstractie houdt het simplificeren van ruimtelijke data op verschillende schaalniveaus waardoor er cellen en links ontstaan. (He et al., 2019) N. Andrienko & G. Andrienko (2011) abstraheren GPS data van Milan op verschillende schaalniveaus. Dit doen de auteurs door eerst karakteristieke punten te vinden en de andere punten hier rond te groeperen. Dan bepalen ze de centroiden en Voronoi tessellatie van deze punten en aggregeren ze de flows tussen de Voronoi polygonen en binnenin de Voronoi polygonen. De resulterende kaart heeft een ruimtelijke abstractie van het autoverkeer in Milan.



Figuur 4: Ontwerp evolutie van de Circos diagrammen van Zeng et al. (a) origineel ontwerp binnen de genetica; (b) zwevende ring voor het kruispunt zelf; (c) bundelen van bi-directionele linten; (d) statistische weergegeven van de inkomende en uitgaande objecten. Bron: Zeng et al. (2013)

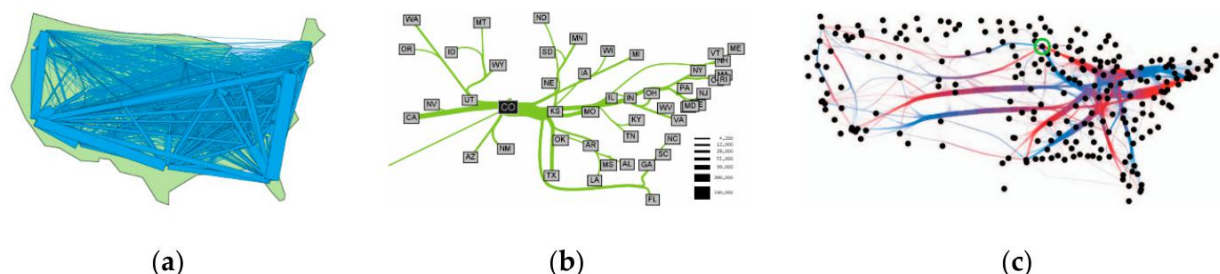
Zeng et al. (2013) stellen de flows binnenin een polygoon voor met behulp van Circos diagrammen. Circos diagrammen zijn circulaire diagrammen die eerst gebruikt werden om de relatie tussen genomen te visualiseren. Zeng et al. gebruikt de Circos diagrammen om het verkeer op een kruispunt voor te stellen. De auteurs delen de cirkel op in vijf delen overeenkomend met de vier straten van het kruispunt en het kruispunt zelf, en trekken dan bogen van het ene deel naar het andere evenredig met de omvang van het verkeer dat die route neemt. Bij verfijnde versies wordt het kruispunt zelf als een centrale vliegende ring op het diagram geplaatst en worden de bogen met elkaar gebundeld om visuele clutter te voorkomen.

2.7 OD-matrix en Flowmap

Indien zoals voorheen karakteristieke punten of regio's worden geïdentificeerd, kunnen de flows tussen deze regio's als Origin-Destination matrix of OD-matrix worden voorgesteld. In een OD-matrix stellen de rijen en kolommen respectievelijk de oorsprongen en destinaties en de waarden in de cellen stellen het aantal verplaatsingen tussen de oorsprong en destinatie. (N. Andrienko & G. Andrienko, 2013; He et al., 2019; S. Wang et al., 2018) Het grote probleem met OD-matrices is dat ze geen ruimtelijke context hebben. Guo et al. (2006) lossen dit probleem op door voor iedere oorsprong/destinatie een kaart te plaatsen op de respectievelijke locatie met de grootte naar/van elke destinatie/oorsprong als kleur van die destinatie/oorsprong. (In N. Andrienko & G. Andrienko, 2013) Wood et al. (2010) deden iets gelijkaardig maar in plaats van een kaart met de oorsprongen en destinaties, gebruiken de auteurs een 2D matrix van oorsprongen ingericht volgens ruimtelijke positie met als cellen telkens een gelijkaardige 2D matrix ingekleurd volgens de flow naar de destinaties van die oorsprong. (In N. Andrienko & G. Andrienko, 2013; He et al., 2019)

Een andere manier om ruimtelijke context aan een OD-matrix te geven is door het maken van een flow map. Een *flow map* is een kaart waarop *flows* worden voorgesteld als rechte of gebogen lijnen of pijlen dat de oorsprongen en destinaties met elkaar verbindt. De grootte van de flow wordt meestal voorgesteld door de dikte van de lijn of de kleur van de lijn. Een *flow map* is de meest gebruikte vorm van flow visualisatie. (N. Andrienko & G. Andrienko, 2013; S. Wang et al., 2018) Waldo Tobler maakte discrete of continue *flow maps*. Discrete *flow maps* toont de flows tussen bepaalde locaties door middel van lijnen of pijlen waarvan de dikte proportioneel is aan het aantal verplaatsingen. (G. Andrienko et al., 2007; 2008; G. Andrienko & N. Andrienko, 2008; N. Andrienko & G. Andrienko, 2007) Deze locaties kunnen vooraf gedefinieerd zijn. G. Andrienko et al. (2011) analyseren de verplaatsingen van Flickr gebruikers tussen de vijf grootste steden in Zwitserland, Zurich, Geneva, Basel, Lausanne en Bern. Locaties kunnen ook bepaald worden op basis van de data. N. Andrienko & G. Andrienko (2011) meten de flows tussen karakteristieke locaties. Deze locaties zijn start- en eindpunten van trajecten, significante bochten en significante stops, lange pauzes in de beweging. Continue *flow maps* geven de trajecten weer als vector velden of stroomlijnen en zijn dus niet

afhankelijke van het aantal karakteristieke locaties. Continue *flow maps* zijn echter wel zeer computationeel zwaar. (G. Andrienko et al., 2008; N. Andrienko & G. Andrienko, 2007)



Figuur 5: Voorbeelden om kruisende flows leesbaar te maken. (a) Veel kruisende migratie flows van US Census data; (b) edge routing; (c) edge bundling. Bron: He et al. (2019)

Bij flows tussen grotere aantallen locaties kunnen de vele kruisende flows onleesbaar worden. Selassie et al. (2011) gebruiken *edge bundling* om minder kruisingen tussen flows te hebben. Bij edge bundling groepeerde gelijkaardige flows via een gemeenschappelijk pad. Zo zijn de grote flows makkelijker te volgen op een kaart. (Hurter et al., 2014) S. Wang et al. (2018) en N. Andrienko & G. Andrienko (2013) zeggen dat edge bundling kan zorgen voor een minder nauwkeurigheid van geografische informatie en dat edge bundling een abstracte ruimte is dat de indruk kan geven een geografische ruimte te zijn. Zo kan het grote wegen tonen dat er in werkelijkheid niet zijn. Doantam et al. (2005) gebruiken een *edge routing* techniek om de overlappende flows via een gewenst, niet overlappend pad te laten lopen.

3 KLASSIEKE INFORMATIE EXTRACTIE

Bij grotere datasets is het vaak niet overzichtelijk om alle trajecten weer te geven in de visualisatie. Vaak wordt de dataset gereduceerd om overzicht te krijgen in de dataset. De meestgebruikte technieken voor het reduceren van een dataset vallen in drie categorieën, selectie en filteren, clusteren en aggregeren.

3.1 Selectie en filteren

Een eerste manier om de grootte van de geanalyseerde dataset te verkleinen is reductie *senso stricto*. Een selectie van trajecten of datapunten kan gemaakt worden om voor te stellen of een filter kan opgesteld worden op basis van bepaalde criteria. (G. Andrienko et al., 2011) Deze criteria worden bepaald door de analist op basis van het doel van de studie. Filteren gebeurt dan ook meestal interactief. Een interactieve filter wordt vaak gebruikt bij Space-Time Cubes. (He et al., 2019) Er zijn drie grote groepen aan filters voor trajecten. Zoals Buschmann et al. (2014) ze ook gebruiken zijn dit filteren op attributen, filteren op locatie en filteren op tijd.

3.1.1 Filteren op attributen

Filteren op attributen houdt in dat enkel maar trajecten of trajectsegmenten worden getoond op basis van een bepaald attribuut van de trajecten. Deze attributen kunnen directe attributen zijn zoals soort voertuig, maar dit kunnen ook afgeleide attributen zijn zoals snelheid, richting, duratie van de trip (G. Andrienko et al., 2007) of de densiteit van de flow bij een flow map (G. Andrienko et al., 2008). Filteren van trajectsegmenten volgens attributen kan de analist helpen om de focus te leggen op de tijdsruimtelijke distributie van bepaalde attribuutwaarden. (N. Andrienko & G. Andrienko, 2013)

3.1.2 Filteren op locatie

Filteren op locatie houdt in dat enkel de trajecten worden getoond die zich op een bepaalde locatie bevinden. Dit kan inzicht geven op de trajecten die al binnen een bepaald gebied vallen. Maar een filter kan ook vallen op enkel de start- of eindlocatie. Graser, Dragaschnig, et al. (2020) filteren de trajecten die een startlocatie hebben in en rond Gotenburg, Zweden.

3.1.3 Filteren op tijd

Filteren op tijd is een vaak gebruikte techniek om trajecten of trajectsegmenten te filteren. Hierbij wordt dan enkel de trajectsegment getoond waarvan het tijdstip binnen een gegeven tijdsinterval ligt. Meerdere tijdsintervallen kunnen ook gegeven worden, bijvoorbeeld cyclische tijdsintervallen zoals bepaalde uren per dag. (Buschmann et al., 2014) Een interactieve tijdsfilter wordt meestal gegeven in de vorm van een tijdsschuiver. (G. Andrienko et al., 2007; Zeng et al., 2013) Deze tijdsschuiver en tijdsfilter is ook een zeer goede methode voor het creëren van animaties. (G. Andrienko et al., 2007)

3.2 Clusteren

Clusteren is een andere goede manier om de grootte van de dataset te reduceren. Clusteren laat de analist toe om de dataset als groepen van trajecten te zien in plaats van individuele trajecten. (Rinzivillo et al., 2008) Het clusteren houdt in dat er groepen van trajecten worden gevormd waardoor de overeenkomst van trajecten binnen een groep maximaal wordt terwijl de overeenkomst tussen groepen minimaal. (Miller & Han, 2009) Er bestaan verschillende cluster algoritmen die kunnen gebruikt worden voor het clusteren van trajectdata.

3.2.1 Hiërarchisch clusteren

Een eerste cluster algoritme is de k-means of k-medoids. K-means probeert k aantal clusters te maken waardoor de gemiddelde afstand van de punten naar het midden van de cluster zo klein mogelijk is. Dit

gebeurt door k willekeurige middens te definiëren en alle punten in de cluster van de dichtstbijzijnde midden te steken. Daarna worden de nieuwe middens van de clusters berekend en herhaald dit proces zich tot de clusters niet meer veranderen. (Rinzivillo et al., 2008) K-means is niet zo geschikt voor het clusteren van trajecten. Een traject is een complex datatype en daar kan geen gemiddelde van berekend worden. Ook eist k-means dat het aantal clusters vooraf bekend is, wat meestal niet het geval is. (G. Andrienko et al., 2009)

3.2.2 Densiteit gebaseerd clusteren

Betere clusteralgoritmen voor trajecten zijn de densiteit gebaseerde clusteralgoritmen zoals DBSCAN (Ester et al., 1996) en haar evolutie OPTICS (Ankerst et al., 1999). DBSCAN neemt als input een maximum afstand ϵ en een minimum aantal punten om een dichte regio te vormen. Het zoekt dan per punt alle punten die binnen een afstand ϵ vallen, de ϵ -buren, en als dit er meer zijn dan het minimum aantal punten dan wordt dit punt gezien als centraal punt en wordt voor al haar ϵ -buren hetzelfde gedaan. Als een punt dicht genoeg bij een centraal punt ligt maar niet genoeg ϵ -buren heeft, is dit een grens punt en hoort die nog steeds tot de cluster. Alle andere punten zijn ruis. (Schubert et al., 2017)

OPTICS maakt geen clusters zelf maar maakt een Bereikbaarheidsplot waarop de clusters in de data intuïtief kunnen afgeleid worden. (Nanni & Pedreschi, 2006; Rinzivillo et al., 2008) Er zijn twee begrippen belangrijk in OPTICS, kernobject en bereikbaarheidsafstand. Een kernobject is een datapunt dat binnenin een afstand ϵ zeker een minimum aan n punten heeft. De kernafstand van het kernobject is de verste afstand van de n dichtste punten. Het kernobject is niet zozeer voor de clustering zelf, maar is meer om de cluster van de ruis te onderscheiden. Dan wordt een willekeurig punt gekozen en zoekt het telkens het dichtste punt dat nog niet bezocht is. Hiervan wordt de bereikbaarheidsafstand bepaald als de afstand tussen de twee punten tenzij ze te dicht bij elkaar liggen. Dan is het de kernafstand. Deze bereikbaarheidsafstanden worden dan in een balkgrafiek gezet en de clusters zijn te herkennen als valleien tussen pieken van ruis. (Nanni & Pedreschi, 2006; Rinzivillo et al., 2008)

Voordelen van deze densiteit gebaseerde methoden is dat er niet vooraf moet geweten worden hoeveel clusters er zijn. Er moet enkel een minimum cluster grootte zijn en een maximum afstand tussen datapunten in een cluster. Deze maximum afstand is zelfs niet nodig voor OPTICS. DBSCAN en OPTICS hebben ook een zeer goede tolerantie tegen ruis. (G. Andrienko et al., 2009) Graser & Widhalm (2018) vinden wel dat de complexiteit van DBSCAN, $O(n^2)$ of $O(n \log n)$ met een ruimtelijke index, niet goed schaalbaar is bij massieve datasets.

3.2.3 Afstand functies

De besproken clustermethoden hebben een afstandsfunctie nodig. Een afstandsfunctie heeft de similariteit, of eerder dissimilariteit weer van twee datapunten. Een voorbeeld van een afstandsfunctie voor puntdata is de Euclidische afstand. Trajectdata is een complex datatype en heeft dus andere afstandsfuncties nodig. Een universele afstandsfunctie dat alle eigenschappen van trajecten in rekening brengt zal zeer complex zijn om te begrijpen en zeer veel computationele kracht nodig hebben om uit te voeren. (Rinzivillo et al., 2008) Niet alle eigenschappen van trajecten zijn dan ook nodig voor een bepaalde analysetaak. Rinzivillo et al. (2008) geven een aantal afstandsfuncties voor trajecten.

Een eerste paar afstand functies dat de auteur geven zijn "*common source*" en "*common destination*". Deze twee geven respectievelijk de geografische of Euclidische afstand tussen de beginpunten of eindpunten van de trajecten. De functie "*k points*" neemt het begin- en eindpunt plus nog k andere punten van elk traject en berekent de gemiddelde afstand tussen corresponderende punten. "*time step*" werkt gelijkaardig aan *k points* alleen neemt die een vast tijdsinterval in plaats van een vast aantal punten. De functies *k points* en *time step* werken enkel goed met "perfecte data".

Een meer complexere afstandsfunctie is "*common route*" gegeven door G. Andrienko et al. (2007). *Common route* is ontworpen om incomplete trajecten te tolereren. Het scant de twee trajecten naar dichtstbijzijnde punten en voegt hun afstand toe aan het resultaat. Als een punt wordt omvergeslagen komt er een straf afstand bij. Een uitbreiding op *common route* is "*common route & dynamics*". Deze neemt ook de relatieve tijd tussen twee opeenvolgende punten mee in de berekening.

Andere afstandsfuncties bestaan ook zoals "*time warping distance*", "*Least Common Sub-Sequence measure*" en "*direction-oriented distances*". (G. Andrienko et al., 2009) Natuurlijke kunnen nog veel andere afstandsfuncties gedefinieerd worden die veel andere eigenschappen van het bewegend object in achtting brengen zoals vervoersmiddel, demografie van het bewegende personen, etc. (Rinzivillo et al., 2008) Ook worden afstandsfuncties vaak gecombineerd in verschillende stappen van het clusteren. (N. Andrienko & G. Andrienko, 2013)

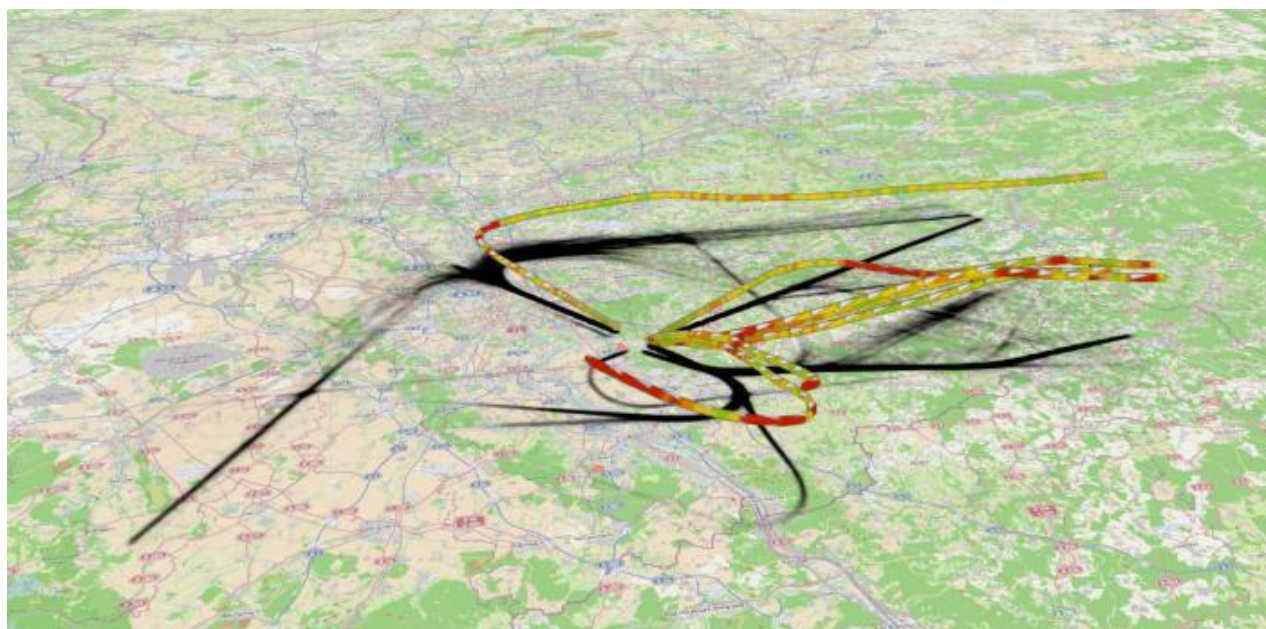
3.3 Aggregatie

Aggregatie is het proces waarbij verschillende datapunten worden samengevoegd en voorgesteld als een geheel. (N. Andrienko & G. Andrienko, 2011) Aggregeren laat toe om de dataset als een versimpelde weergave te hebben, een soort data abstractie en cartografische generalisatie. G. Andrienko & N. Andrienko (2008) en N. Andrienko & G. Andrienko (2013) onderscheiden twee grote groepen van aggregatie taken voor trajectdata. Ten eerste is er de aggregatie van bewegende objecten op verschillende

locaties en hun temporele variatie. Ten tweede is er de aggregatie van de flows, van bewegende objecten tussen locaties.

3.3.1 Aggregatie op locaties

Aggregeren op locatie houdt in dat statistieken zoals aantal, tijd gespendeerd of andere maten worden berekend per locatie over een continu denseiteitsoppervlak of een discreet grid of andere opdeling. (N. Andrienko & Andrienko, 2013) Scheepens et al. (2011) aggregeren de positie van schepen voor de haven van Rotterdam samen met het type schip om een risicokaart te maken in de vorm van een denseiteitskaart. (Figuur 2) De kaart toont de plaatsen waar veel schepen passeren en waar er dus een groot risico is op een botsing. Ook zijn de grote zeelanen, en dus grote flows, er duidelijk op te herkennen. Buschmann et al. (2015) gebruiken een denseiteitskaart in 3D om data van vliegtuigen rond een luchthaven te aggregeren. De denseiteitskaart is ingekleurd volgens de gemiddelde versnelling op die locatie.



Figuur 6: Aggregaat van vliegtuigtrajecten rond een luchthaven. De kleur van het segment heeft de gemiddelde versnelling van de vliegtuigen weer. Bron: Buschmann et al. (2015)

Aggregaten op locatie kunnen ook voorgesteld worden met meer abstracte symbolen. (N. Andrienko & G. Andrienko, 2007) stellen verschillende aspecten van de aggregaten voor met symbolen zoals cirkels, vierkanten op driehoeken in een regelmatig grid. N. Andrienko & G. Andrienko (2011) maken een grid van Voronoi polygonen op basis van karakteristieke punten en kleurt dan de grid cellen volgens het aantal schepen en de grootte van de punten volgens de duur van een bezoek in een grid cel. G. Andrienko & N. Andrienko (2008) plaatsen op een regelmatig grid een grafiek die de drukte van die grid cel weergeeft per

uur van de week. Krueger et al. (2016) geven een vorm van raster calculus op aggregaten met gelijke grid cellen als een soort van multi-aggregaat.

De aggregatie op locatie kan ook de dominante richting per grid cel weergeven. G. Andrienko & N. Andrienko (2008) geven per grid cel de dominante richting van het verkeer weer als balkjes waarbij de grootte van de balkjes de grootte van het verkeer in die richting voorstelt. Graser, Widhalm et al. (2020) geven per grid cel een paar prototypes per richting en gemiddelde snelheid. Deze prototypes stellen op grotere schaal ook de dominante flows voor.

3.3.2 Aggregatie op flows

Aggregeren op flows is aggregeren op een paar van locaties waarbij een statistiek genomen wordt van de flows tussen die locaties. Dit kan zijn het aantal bewegende objecten, het aantal identieke objecten, de gemiddelde snelheid, duur van de verplaatsing, etc. (N. Andrienko & G. Andrienko, 2013) Er zijn twee grote groepen om de aggregatie op flows te bepalen: aggregatie tussen oorsprong en bestemming en aggregatie op de route of langs alle tussenliggende locaties.

Bij de aggregatie tussen oorsprong en bestemming wordt de aggregatie gemaakt voor de trajecten die hun oorsprong hebben in het ene karakteristieke punt en hun bestemming in het andere. Een zeer bekende presentatie hiervan is de OD-matrix. (N. Andrienko & G. Andrienko, 2013) Zoals eerder aangehaald in 2.7 heeft een OD-matrix een gebrek aan ruimtelijke context. Dit kan worden opgelost door gebruik te maken van een flow map in plaats van een OD-matrix, maar als de oorsprong en bestemming niet naast elkaar ligt kunnen er vele kruisingen ontstaan tussen de aggregaten.

Bij de aggregatie op de route wordt de aggregatie gemaakt tussen karakteristieke punten met alle trajecten die erdoor lopen. Deze punten of regio's zijn dan ook regio's die naast elkaar liggen. Ook moet de temporele resolutie van de data fijn genoeg zijn om hier een mooi resultaat uit te krijgen. (N. Andrienko & G. Andrienko, 2013) Deze aggregaten kunnen zeer mooi voorgesteld worden op een flow map. Ook kunnen er verschillende dingen uit geleerd worden door te spelen met de grootte en aantal van karakteristieke punten en onderverdelingen. (N. Andrienko & G. Andrienko, 2011)

3.4 Aggregaten gefilterd per cluster

Nu dat deze drie technieken besproken zijn, is het belangrijk om te weten dat deze niet apart moeten gebruikt worden. Rinzivillo et al. (2008) gebruiken interactieve filtering om dynamisch berekende aggregaten te maken. G. Andrienko et al. (2009) filteren hun inputdata voor het clusteren om irrelevante cluster weg te werken. G. Andrienko & N. Andrienko (2008) maken aggregaten per cluster voor de grote

auto flows die centrum Milaan binnenrijden. Het is dus duidelijk te zien dat deze drie technieken zelden apart worden gebruikt en altijd in combinatie met elkaar. Het zijn drie tools in één ruimere toolbox.

4 TE GROOT VOOR RAM

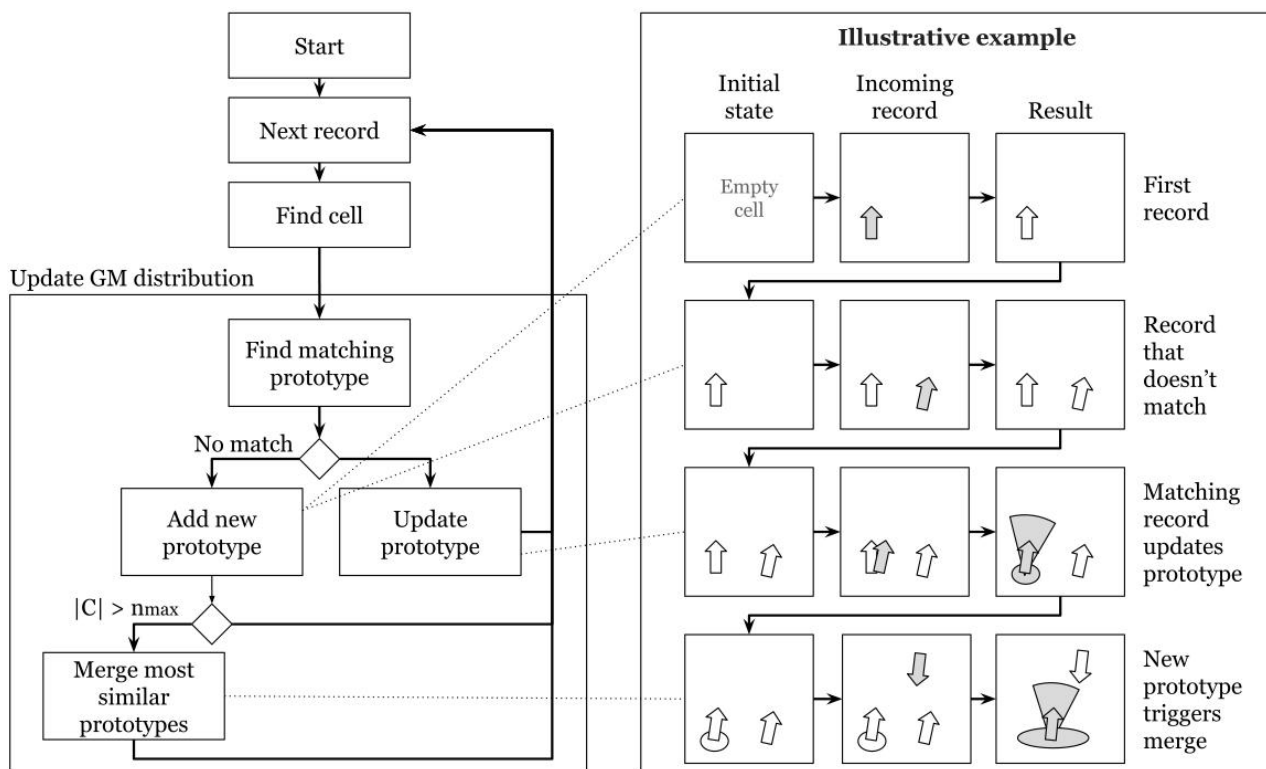
Een groot probleem dat de vooraf besproken methoden hebben, is dat ze hun berekeningen doen op de volledige dataset te samen, waardoor de volledige dataset moet gematerialiseerd zijn in het werkgeheugen of RAM van de computer. Dit is een probleem als de dataset te groot is voor het RAM-geheugen. Zeker hedendaags met het gemak waarmee trajectdata wordt verzameld dankzij technologieën zoals GPS of AIS (Graser, Dragaschnig, et al., 2020), worden de datasets steeds groter en groter. Brunsdon & Comber (2020) zeggen dat “Big Data” altijd maar groter en groter zal worden, en dat de “big data” van dit jaar, een normale grote dataset van volgend jaar zal zijn. Er zullen altijd software en hardware updates zijn om meer grotere data te kunnen werken. Toch is er de laatste tijd wat onderzoek gebeurd om beter met massieve datasets trajectdata te kunnen omgaan.

4.1 Subset als classifier

G. Andrienko et al. (2009) gebruiken een classifier om trajecten één per één in de juiste cluster te steken. Dit doen ze door eerst een deel van de data te clusteren, en daarna een classifier te maken bestaande uit een aantal prototypes en afstandsrempels zo dat al de trajecten in de cluster niet verder liggen dan een afstandsrempel van een prototype. Na visuele inspectie van deze prototypes kunnen ze gebruikt worden om de rest van de trajecten de clusteren. Na de classificatie is de classifier nog steeds nuttig voor verdere analyses op andere datasets.

4.2 Sequentiële/parallele berekening

Graser, Widhalm, et al. (2020) behandelen alle datapunten van de trajecten apart en voegen ze toe aan een aggregaat gebaseerd op prototypes. De prototypes zijn gebaseerd op Gaussian Mixture Models en bevatten statistieken zoals gemiddelde en covariantie van de vector die vaarrichting en snelheid beschrijft. Bij elk datapunt wordt de meest gelijke prototype gevonden en geüpdatet of wordt een nieuw prototype gecreëerd. Zo moet er telkens maar één datapunt en het aggregaat van prototypes in RAM worden gematerialiseerd. Om de processing tijd te reduceren en parallel processing toe te laten wordt de spatio-temporele ruimte opgedeeld in in regelmatige delen met behulp van quadrees (Graser & Widhalm, 2018) en worden deze verdeeld over verschillende processing nodes.



Figuur 7: Workflow van M³. De datapunten worden één per één verwerkt waardoor ze niet allen te samen in het computergeheugen moeten geladen worden. Bron: Graser, Widhalm et al. (2020)

5 BESLUIT

Onderzoekers hebben verschillende technieken ter beschikking om inzichten te verwerven uit massieve datasets trajectdata. Ten eerste zijn er de vele visualisatiemogelijkheden om informatie van trajecten op kaart te brengen. Voorbeelden hiervan zijn interactieve Space-Time Cubes, animaties die spelen op de menselijke perceptie van veranderingen, stacking graphs die trajecten verticaal ordenen, density maps en heat maps die de ruimtelijke spreiding en samenligging van trajecten weergeeft, meshing dat de ruimte opdeelt volgens karakteristieke punten in de trajecten, en OD-matrices en flow maps die flows weergeeft tussen verschillende regio's al dan niet gebaseerd op karakteristieke punten. Deze methoden staan natuurlijk zelden apart maar worden vaak samen gebruikt in softwarepakketten in de vorm van brushing en linked displays.

Ten tweede moeten de datasets op een zinnige manier gereduceerd kunnen worden. Dit kan gebeuren door reduceren *senso stricto*, waarbij maar een selectie van trajecten wordt genomen waar de onderzoeker in geïnteresseerd is of een filter wordt opgesteld op basis van de vereisten van het onderzoek. Deze filter kan gebaseerd zijn op attributen zoals soort verplaatsing of voertuig, of afgeleide attributen zoals snelheid of richting. Ook kan er gefilterd worden op locaties zoals trajecten die volledig binnen een bepaald gebied

vallen of trajecten die hun start of eindlocatie rond een bepaald punt hebben liggen. Een laatste vorm van filteren is filteren volgens tijdspanne. Hiermee kan de onderzoeker focussen op een bepaald moment, of kan met behulp van een tijdslijder animaties gemaakt worden.

Een tweede manier van reduceren is het groeperen of clusteren van trajecten. Voor trajecten is de beste methode om te clusteren meestal de densiteits gebaseerde methoden zoals DBSCAN en OPTICS. Hierbij is dan meestal ook een afstandsfunctie voor nodig die heel complex kan zijn als je alle aspecten van een traject mee in rekening wil brengen. Simpelere afstandsfuncties zoals *common source/destination* of *similar path* worden dan meestal gecombineerd. Een laatste veel gebruikte manier om te reduceren is aggregatie. Aggregatie op locatie geeft meer informatie over flows die door een locatie gaan. Aggregatie op flows geeft meer info over flows tussen de locaties. Filteren, clusteren en aggregaten worden vaak in samenwerking met elkaar gebruikt.

Bij erg grote datasets moet er gekeken worden naar methoden waarbij de datasets niet volledig in het computergeheugen moet geladen worden. Dit kan voor het clusteren door eerst met een subset van de data een classifier te maken en deze dan los te laten op de rest of de dataset. Of dit kan voor aggregatie door datapunt per datapunt te werken en telkens enkel het huidige datapunt en het huidige eindresultaat in het computergeheugen te laden.

6 REFERENTIELIJST

- Andrienko, G., & Andrienko, N. (2008). Spatio-temporal aggregation for visual analysis of movements. *VAST'08 - IEEE Symposium on Visual Analytics Science and Technology, Proceedings*, 51–58. <https://doi.org/10.1109/VAST.2008.4677356>
- Andrienko, G., & Andrienko, N. (2013). Visual analytics of movement. In *Visual Analytics of Movement*. Fraunhofer IAIS.
- Andrienko, G., Andrienko, N., Bak, P., Keim, D., Kisilevich, S., & Wrobel, S. (2011). A conceptual framework and taxonomy of techniques for analyzing movement. *Journal of Visual Languages and Computing*, 22(3), 213–232. <https://doi.org/10.1016/j.jvlc.2011.02.003>
- Andrienko, G., Andrienko, N., Chen, W., Maciejewski, R., & Zhao, Y. (2017). Visual analytics of mobility and transportation: State of the art and further research directions. *IEEE Transactions on Intelligent Transportation Systems*, 18(8), 2232–2249. <https://doi.org/10.1109/TITS.2017.2683539>
- Andrienko, G., Andrienko, N., Kopanakis, I., Ligtenberg, A., & Wrobel, S. (2008). Visual analytics methods for movement data. In F. Giannotti & D. Pedreschi (Eds.), *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery* (1st ed., pp. 375–410). Springer-Verlag. https://doi.org/10.1007/978-3-540-75177-9_14
- Andrienko, G., Andrienko, N., Rinzivillo, S., Nanni, M., Pedreschi, D., & Giannotti, F. (2009). Interactive Visual Clustering of Large Collections of Trajectories. *2009 IEEE Symposium on Visual Analytics Science and Technology*, 3–10. <https://doi.org/10.1109/VAST.2009.5332584>
- Andrienko, G., Andrienko, N., & Wrobel, S. (2007). Visual analytics tools for analysis of movement data. *ACM SIGKDD Explorations Newsletter*, 9(2), 38–46. <https://doi.org/10.1145/1345448.1345455>
- Andrienko, N., & Andrienko, G. (2005). *Exploratory Analysis of Spatial and Temporal Data: A systematic Approach*. Springer-Verlag.
- Andrienko, N., & Andrienko, G. (2007). Designing visual analytics methods for massive collections of movement data. *Cartographica*, 42(2), 117–138. <https://doi.org/10.3138/carto.42.2.117>
- Andrienko, N., & Andrienko, G. (2011). Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2), 205–219. <https://doi.org/10.1109/TVCG.2010.44>
- Andrienko, N., & Andrienko, G. (2013). Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1), 3–24. <https://doi.org/10.1177/1473871612457601>
- Ankerst, M., Breunig, M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering Points to Identify the Clustering Structure. *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, 49–60.
- Bach, B., Dragicevic, P., Archambault, D., Hurter, C., & Carpendale, S. (2017). A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes. *Computer Graphics Forum*, 36(6), 36–61. <https://doi.org/10.1111/cgf.12804>
- Brunsdon, C., & Comber, A. (2020). Big Issues for Big Data: challenges for critical spatial data analytics. *Journal of Spatial Information Science*, 21, 89–98. <https://doi.org/10.5311/JOSIS.2020.21.625>

- Buschmann, S., Trapp, M., & Dollner, J. (2014). Real-time animated visualization of massive air-traffic trajectories. *2014 International Conference on Cyberworlds*, 174–181. <https://doi.org/10.1109/CW.2014.32>
- Buschmann, S., Trapp, M., & Döllner, J. (2015). *Concepts for Real-time Processing and Visualization of Massive Air-Traffic Data in Digital Landscapes Mobility and Reachability Analytics View project Image and Video Abstraction View project Real-Time Visualization of Massive Movement Data in Digital Lan.*
- Demšar, U., & Verrantaus, K. (2010). Space-time density of trajectories: Exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science* 24, 1527–1542.
- Ding, X., Chen, L., Gao, Y., Jensen, C. S., & Bao, H. (2018). UITraMan: A unified platform for big trajectory data management and analytics. *Proceedings of the VLDB Endowment*, 11(7), 787–799. <https://doi.org/10.14778/3192965.3192970>
- Doantam, P., Ling, X., Yeh, R., & Hanrahan, P. (2005). Flow map layout. *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS 2005)*, 219–224.
- Du, F., Zhu, A. X., & Qi, F. (2016). Interactive visual cluster detection in large geospatial datasets based on dynamic density volume visualization. *Geocarto Int.*, 31, 597–611.
- Ester, M., Kriegel, H. P., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining*, 226–231.
- Graser, A. (2018). Evaluating Spatio-temporal Data Models for Trajectories in PostGIS Databases. *GI_Forum*, 1, 16–33. https://doi.org/10.1553/giscience2018_01_s16
- Graser, A. (2021). An exploratory data analysis protocol for identifying problems in continuous movement data. *Journal of Location Based Services*. <https://doi.org/10.1080/17489725.2021.1900612>
- Graser, A., Dragaschnig, M., Widhalm, P., Koller, H., & Brandle, N. (2020). Exploratory Trajectory Analysis for Massive Historical AIS Datasets. *Proceedings - IEEE International Conference on Mobile Data Management, 2020-June*, 252–257. <https://doi.org/10.1109/MDM48529.2020.00059>
- Graser, A., & Widhalm, P. (2018). Modelling massive AIS streams with quad trees and Gaussian Mixtures. *21st AGILE International Conference on Geographic Information Science*. <https://doi.org/10.31223/osf.io/sz34w>
- Graser, A., Widhalm, P., & Dragaschnig, M. (2020). The M³ massive movement model: a distributed incrementally updatable solution for big movement data exploration. *International Journal of Geographical Information Science*, 1–24. <https://doi.org/10.1080/13658816.2020.1776293>
- Guo, D., Chen, J., MacEachren, A., & Liao, K. (2006). A visualization system for space-time and multivariate patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics*, 12(6), 1461–1474.
- He, J., Chen, H., Chen, Y., Tang, X., & Zou, Y. (2019). Diverse Visualization Techniques and Methods of Moving-Object-Trajectory Data: A Review. *ISPRS International Journal of Geo-Information*, 8(2), 63. <https://doi.org/10.3390/ijgi8020063>
- Hurter, C., Ersoy, O., Fabrikant, S. I., Klein, T. R., & Telea, A. C. (2014). Bundled visualization of dynamic graph and trail data. *IEEE Transactions on Visualization and Computer Graphics*, 20(8), 1141–1157. <https://doi.org/10.1109/TVCG.2013.246>

- Kraak, M.-J., & Huisman, O. (2009). Beyond Exploratory Visualization of Space-Time Paths. In H. J. Miller & J. Han (Eds.), *Geographic Data Mining and Knowledge Discovery, Second Edition* (Second Edi, pp. 431–442). CRC Press. <https://doi.org/10.1201/9781420073980>
- Krueger, R., Sun, G., Beck, F., Liang, R., & Ertl, T. (2016). TravelDiff: Visual comparison analytics for massive movement patterns derived from Twitter. *IEEE Pacific Visualization Symposium, 2016-May*, 176–183. <https://doi.org/10.1109/PACIFICVIS.2016.7465266>
- Miller, H. J., & Han, J. (2009). Geographic data mining and knowledge discovery. In H. J. Miller & J. Han (Eds.), *Geographic Data Mining and Knowledge Discovery, Second Edition* (Second Edi). CRC Press. <https://doi.org/10.1201/9781420073980>
- Nanni, M., & Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3), 267–289. <https://doi.org/10.1007/s10844-006-9953-7>
- Ostermann, F. (2009). *Modeling, analyzing and visualizing human space appropriation*. 161.
- Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., & Andrienko, G. (2008). Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3–4), 225–239. <https://doi.org/10.1057/palgrave.ivs.9500183>
- Scheepens, R., Willems, N., Van De Wetering, H., Andrienko, G., Andrienko, N., & Van Wijk, J. J. (2011). Composite density maps for multivariate trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2518–2527. <https://doi.org/10.1109/TVCG.2011.181>
- Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3), 1–21. <https://doi.org/10.1145/3068335>
- Selassie, D., Heller, B., & Heer, J. (2011). Divided edge bundling for sirectional data. *IEEE Transactions on Visualization and Computer Graphics*, 17, 2351–2363.
- Wang, S., Du, Y., Jia, C., Bian, M., & Fei, T. (2018). Integrating algebraic multigrid method in spatial aggregation of massive trajectory data. *International Journal of Geographical Information Science*, 32(12), 2477–2496. <https://doi.org/10.1080/13658816.2018.1512713>
- Wang, Z., Yuan, Y., Chang, L., Sun, X., & Luo, X. (2019). A Graph-Based Visual Query Method for Massive Human Trajectory Data. *IEEE Access*, 7, 160879–160888. <https://doi.org/10.1109/ACCESS.2019.2948304>
- Wood, J., Dykes, J., & Slingsby, A. (2010). Visualisation of Origins, Destinations and Flows with OD Maps. *The Cartographic Journal*, 47(2), 117–129.
- Zeng, W., Fu, C.-W., Arisona, S. M., & Qu, H. (2013). Visualizing Interchange Patterns in Massive Movement Data. *Computer Graphics Forum*, 32(3), 271–280. <https://doi.org/10.1111/cgf.12114>