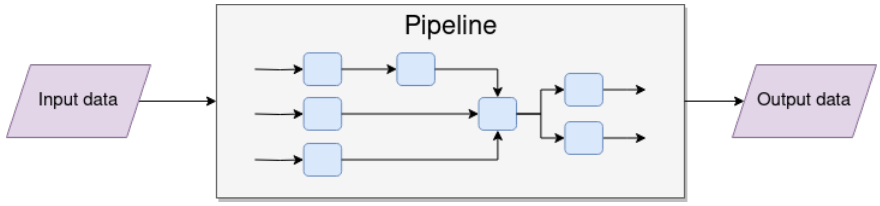
On the optimization of imaging pipelines

Schoonhoven, R.A.

Samenvatting

Veel van onze industriële en wetenschappelijke taken zijn tegenwoordig afhankelijk van krachtige computersystemen die enorme hoeveelheden gegevens verwerken. Deze taken worden uitgevoerd door computationele pijplijnen die vaak uit verschillende algoritmen bestaan, en die berekeningen op data uitvoeren en aan elkaar doorgeven. Conceptueel gezien bestaat het ontwerpproces van een computationele pijplijn uit verschillende fasen.

- **De ideeën fase**: Hier ontwerpen we de pijplijn, beslissen we welke algoritmen we gaan gebruiken en hoe ze op elkaar aansluiten. Figuur S1 toont een conceptueel voorbeeld van hoe we de procedure zouden kunnen ontwerpen.
- **De softwarefase**: Hier beslissen we hoe elk algoritme zijn taak moet uitvoeren. We proberen de code zo te schrijven om de algoritmes zo efficiënt en nauwkeurig mogelijk te maken.
- **De hardwarefase**: Hier wordt bepaald welke computerhardware elk algoritme tot zijn beschikking heeft en hoe de berekening efficiënt uitgevoerd kan worden op die hardware.

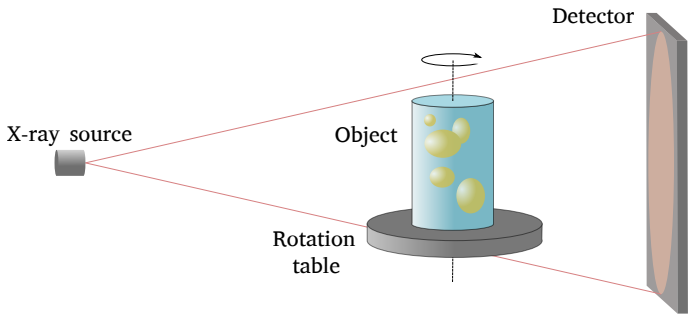


Figuur S1: Voorbeeld van een computationele pijplijn waar data door verschillende algoritmen wordt verwerkt en doorgegeven wordt totdat het uiteindelijke resultaat is berekend.

Het praktisch draaien van zo'n computationele pijplijn kan een uitdaging zijn. We moeten vaak de manier waarop de algoritmen werken aanpassen, ervoor zorgen dat ze geen tijd en energie verspillen, de hardware verbeteren en soms moeten we de hele pijplijn opnieuw ontwerpen om het beste resultaat te krijgen. Bovendien moeten we, gezien de huidige milieu overwegingen, ook proberen om onze pijplijn milieuvriendelijker te maken. Om dit te doen richten we ons in dit proefschrift op alle drie de fasen om onze computationele pijplijnen te optimaliseren.

Een van de toepassingen waar zulke uitdagende computationele pijplijnen voorkomen is bij de röntgen-computertomografie. Computertomografie (CT) is een techniek om de binnenkant van objecten te visualiseren door middel van bestraling (Figuur S2). Dit is een krachtige technologie omdat we de binnenkant

van objecten kunnen bestuderen zonder ze te beschadigen of open te maken. Één van de computationele uitdagingen komt voort uit het feit dat CT meestal in 3D wordt uitgevoerd, wat betekent dat het om grote hoeveelheden gegevens gaat en dat elke computationele stap ook een grote hoeveelheid gegevens moet werken. Bovendien bevatten CT-pijplijnen meestal verschillende verwerkingsstappen die we willen optimaliseren.



Figuur S2: Een voorbeeld CT opstelling die men in vaak in laboratoria aantreft.

Verschillende instellingen van een CT-scanner, zoals de energie van de bundel en vanuit welke positie de röntgenopnames worden opgevangen, moeten correct worden gekozen om een scherp CT-beeld met een hoog contrast te krijgen. Dit creëert een uitdaging vanuit een optimalisatie oogpunt. Vaak is het gewenste resultaat pas aan het einde van de berekening zichtbaar, terwijl het veroorzaakt kan worden door een instelling aan het begin van de CT procedure. De uitlijning van de machine kan bijvoorbeeld niet goed zijn, of we hebben onze algoritmen met de verkeerde parameters gedraaid. Het probleem wordt nog verergerd doordat men vaak extra bewerkingsstappen wil uitvoeren aan het einde van de reconstructies.

In hoofdstuk 2 proberen we met AI een zogenaamde segmentatiestap toe te voegen aan een CT procedure, zodat de berekening live kan worden uitgevoerd. In hoofdstuk 3 proberen we het optimalisatieprobleem aan te pakken om algoritmische parameters over de gehele pijplijn tegelijk te optimaliseren. In hoofdstuk 4 kijken we nog eens naar de AI die we gebruikten in hoofdstuk 2, en proberen we deze aanzienlijk te versnellen. We doen dit door een nieuwe methode te ontwikkelen die het neurale netwerk kan afslanken tot een veel kleinere versie van zichzelf, maar met een vergelijkbare nauwkeurigheid. In hoofdstuk 5 kijken we naar de hardwarefase. In het bijzonder kijken we naar de grafischekaart (Graphics Processing Unit of GPU genoemd), en bekijken we welke algoritmen het beste zijn voor het structureren van deze GPU berekeningen. In hoofdstuk 6 bekijken we GPU berekeningen vanuit een milieu- en energieverbruik perspectief, en introduceren we een model om deze energie-efficiënter te maken. Uiteindelijk bekijken we hoeveel energie onze methode bespaart voor een groot Europees netwerk van radiotelescopen genaamd de Low-frequency Array (LOFAR).