Efficient Algorithms for Flood Risk Analysis

March 29, 2017

Lars Arge
Professor and Center Director

AARHUS UNIVERSITY

MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

DABAI

Societal Challenge

Flood risk important societal challenge

- Cost of 2011 Copenhagen flood over 1 billion dollars

Copenhagen 2011

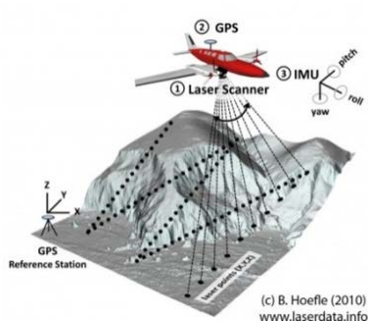
Aarhus 2012

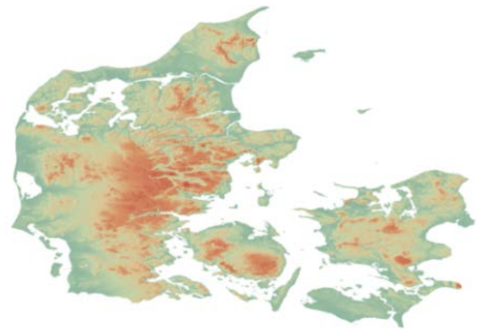
- Danish population (Userneeds)
  - 50% worry about their homes being flooded
  - 90% say high flood risk affect decision to buy house

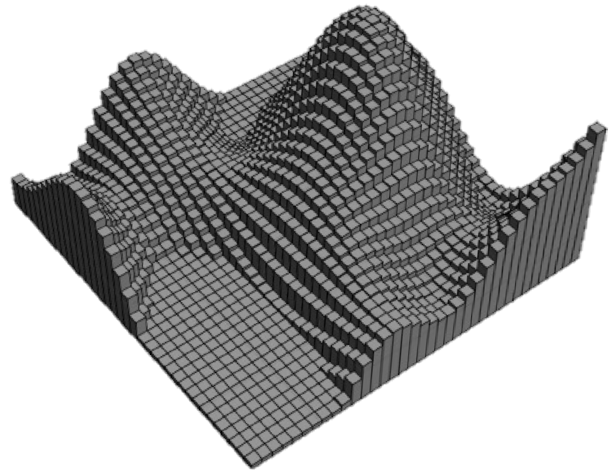
Public Big Data

Precise and detailed national terrain model

- At least 4 measurements per m^2







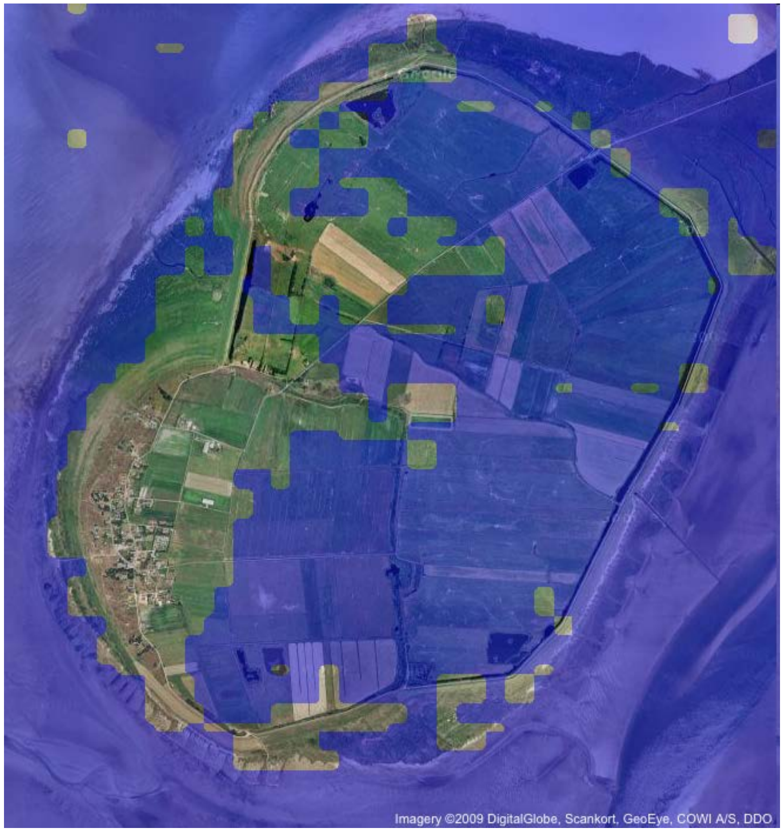
Freely available in base data program

Important basis for flood risk analysis

- Terrain essential for water flow
- Thousands of points in family home lot

Detailed Terrain Data Essential

Sea-level rise (2 meter effect on Mandø)

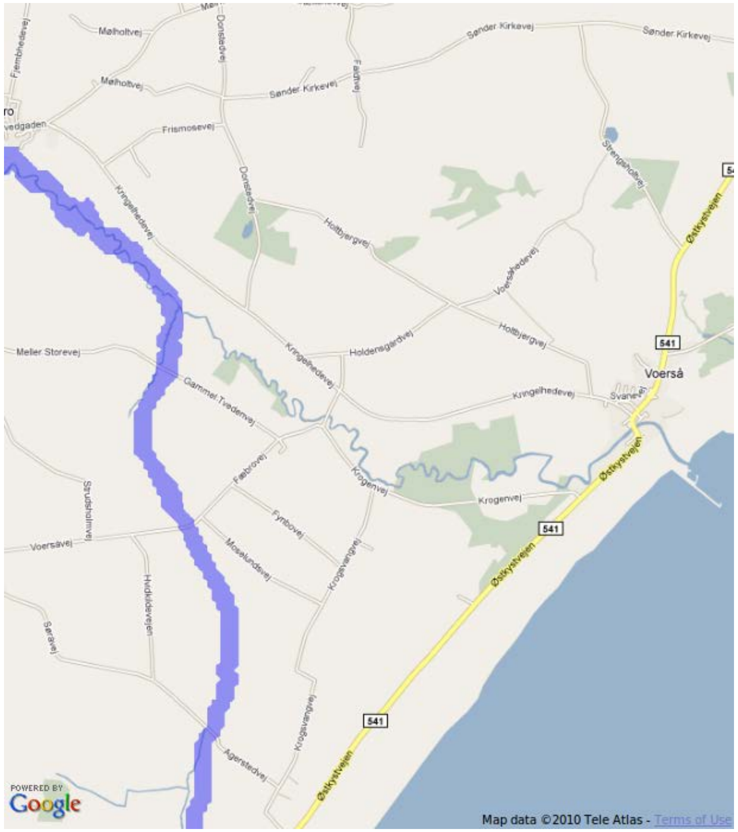


90 meter terrain model

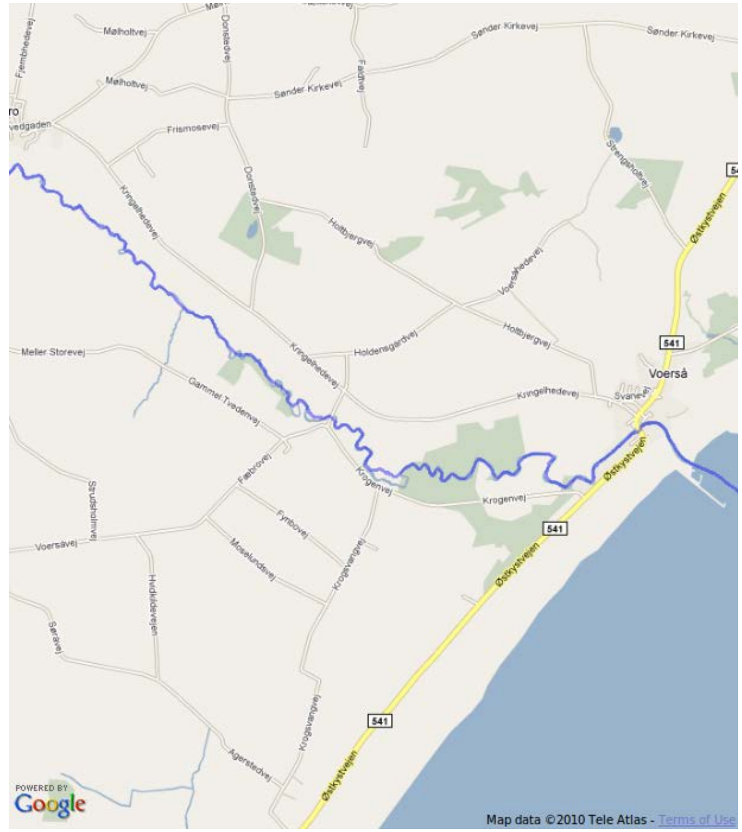
2 meter terrain model

Detailed Terrain Data Essential

Drainage network (flow accumulation)



90 meter terrain model

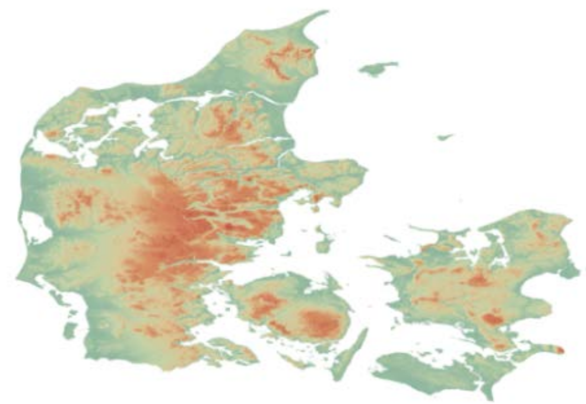


2 meter terrain model

Difficult to Handle BIG Detailed Data

Detailed terrain data BIG

- Denmark (42.000 km^2) ~168 billion measurements
- Aarhus ~2 billion measurements



Most analysis software cannot handle such BIG datasets!

Typical workarounds:

- Tiling (difficult and cumbersome)
- Data simplification (unreliable results)
- Small areas (area identification, border conditions)

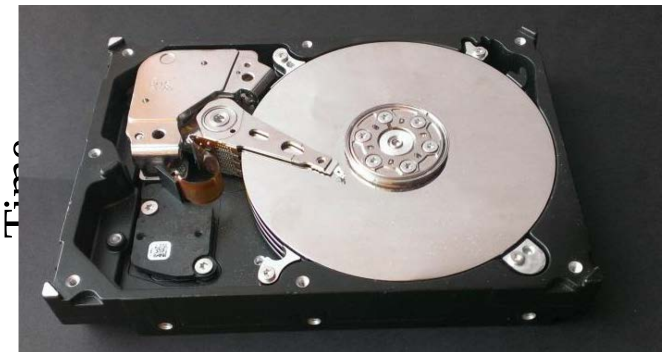
$\Rightarrow$ Hard to use detailed national terrain model even on regional scale

I/O-Efficient Algorithms

Disk often bottleneck when handling massive data

- Disk access is 10^6 times slower than main memory access

"The difference in speed between modern CPU and disk technologies is analogous to the difference in speed in sharpening a pencil using a sharpener on one's desk or by taking an airplane to the other side of the world and using a sharpener on someone else's desk." (D. Comer)



- Large disk access time amortized by transferring large data blocks

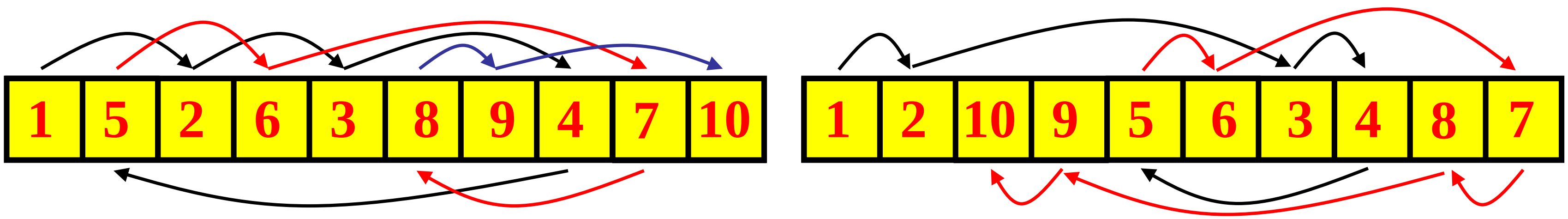
→ Important to store/access data to take advantage of blocks

I/O-efficient algorithms:

- Move as few disk blocks as possible to solve given problem

I/O-Efficient Algorithms Matter

- Example: Traversing linked list (List ranking)
  - Array size N = 10 elements
  - Disk block size B = 2 elements
  - Main memory size M = 4 elements (2 blocks)



Algorithm 1: N=10 I/Os   Algorithm 2: N/B=5 I/Os

- Difference between N and N/B large since block size is large
  - Example: $N = 256 \times 10^6$, $B = 8000$, $1ms$ disk access time

    $\Rightarrow$ N I/Os take 256×10^3 sec = 4266 min = 71 hr

    $\Rightarrow$ N/B I/Os take $256/8$ sec = 32 sec

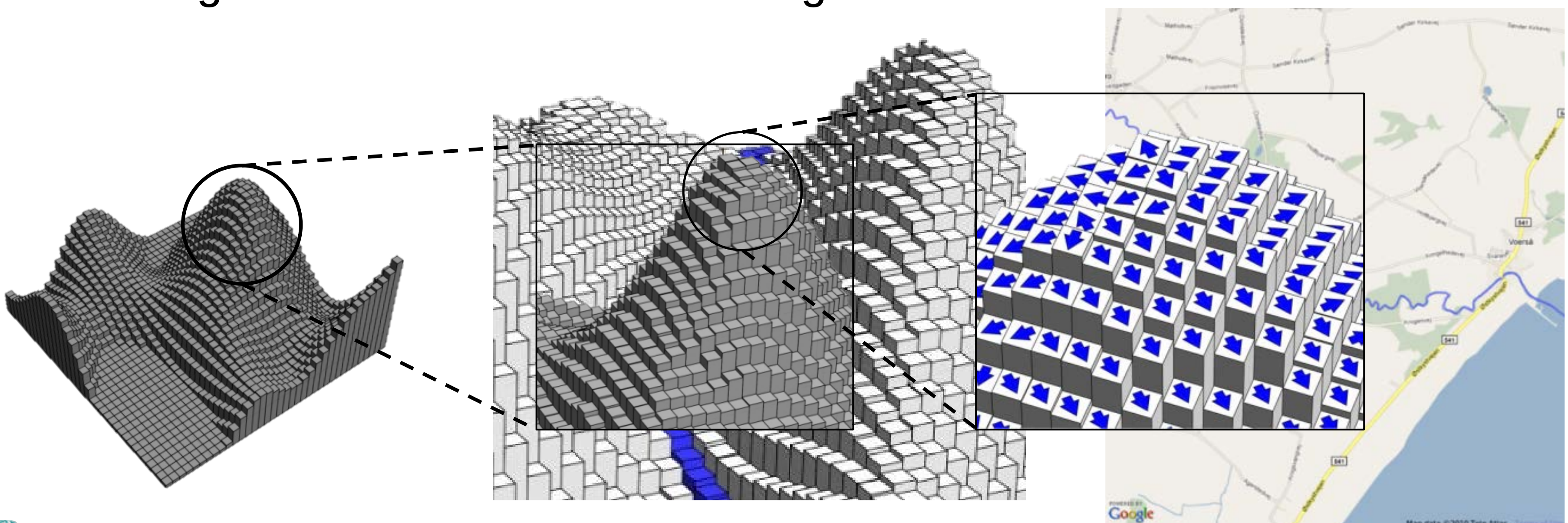
Surface Flow Modeling

Flow accumulation on grid terrain model:

- Initially one unit of water in each grid cell
- Water (initial and received) distributed from each cell to lowest lower neighbor cell (if existing)

⇒ Flow accumulation of cell is total amount of water through it

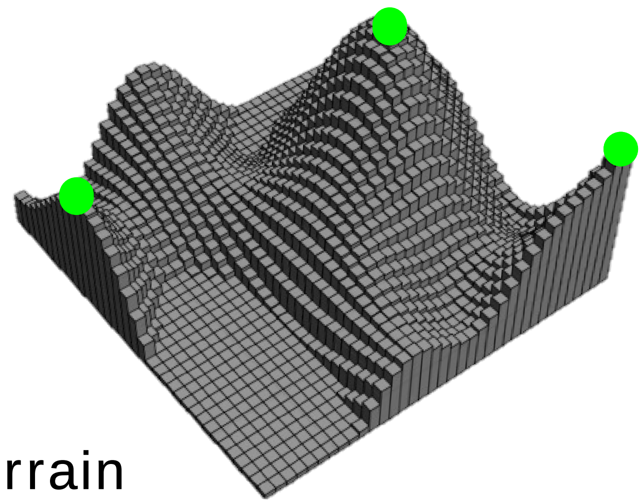
⇒ Drainage network = cells with high flow accumulation



Flow Accumulation Algorithm

Natural algorithm

- Sort cells by height
- Visit cells in height order and for each cell
  - Read height, height of neighbors and water amount
  - Add water to lowest neighbor



Problem: Disk access for each cell

- Almost same height cells scattered over terrain

$\Rightarrow$ Algorithm cannot handle Denmark model

Efficient Flow Accumulation Algorithm

Our I/O-efficient algorithm avoids scattered access by

- Data duplication
- "Lazy write"
  (using I/O-efficient priority-queue)

⇒ ~N/B I/O algorithm

⇒ Easily handle Denmark on normal desktop

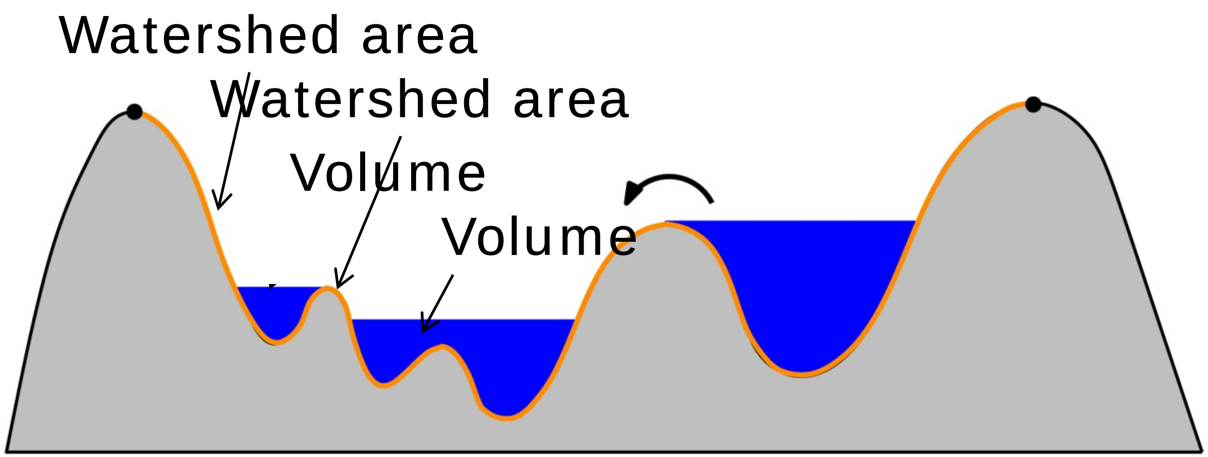
Several other I/O-efficient flood risk analysis algorithms have been developed

- E.g. event based flood risk

Depression Flood Risk: Flash Flood Map

Models how surface water gathers in depressions as it rains

- Water from watershed of depression gathers in the depression
- Depressions fill, leading to (potentially dramatic) increase in neighbor depression watershed size



Flash Flood Mapping:

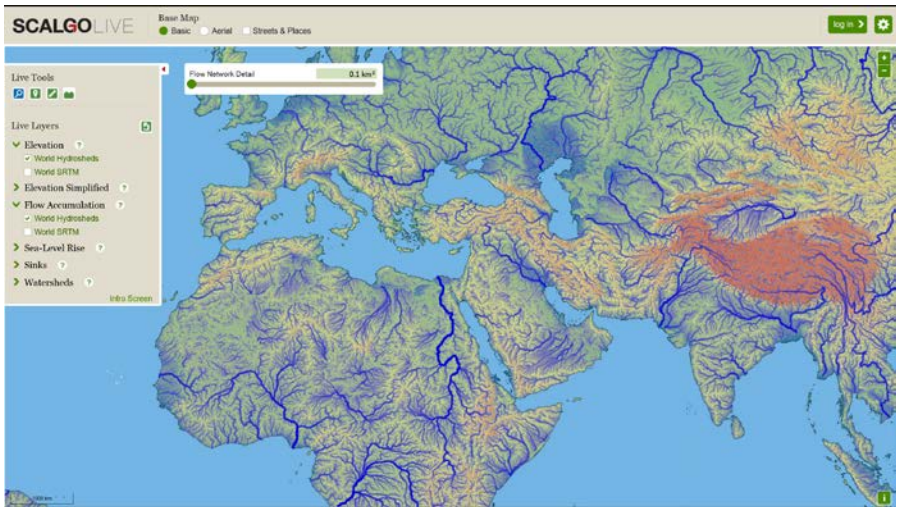
- Amount of rain before any given raster cell is below water

Commercialization

Flood risk algorithm technology commercialized by start-up SCALGO

Developed successful product SCALGO live

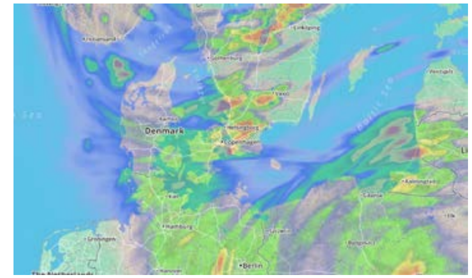
- Cloud based flood risk analysis platform
- Made possible by free data and algorithm technology
- Important interplay between efficient algorithms and visual analytics



DABAI Flood Risk Cases

The DABAI project will push BIG data flood risk analysis further

- Combining algorithms, visual analytics and machine learning technology
- Involving large number of partners



Currently four planned cases

1. ***Semi-automatic identification of hydrological corrections***
   - AU, DTU, KU, SDFE
2. ***Integration of forecast data with flood risk screening***
   - AU, DMI, SCALGO, SDFE, RM
3. Analysis updating after data update
   - SCALGO, SDFE, AU
4. River flood risk using integrated terrain and river data
   - COWI, NIRAS, SCALGO, SDFE, AU, RM

Thanks for your attention

large@cs.au.dk

DABAI