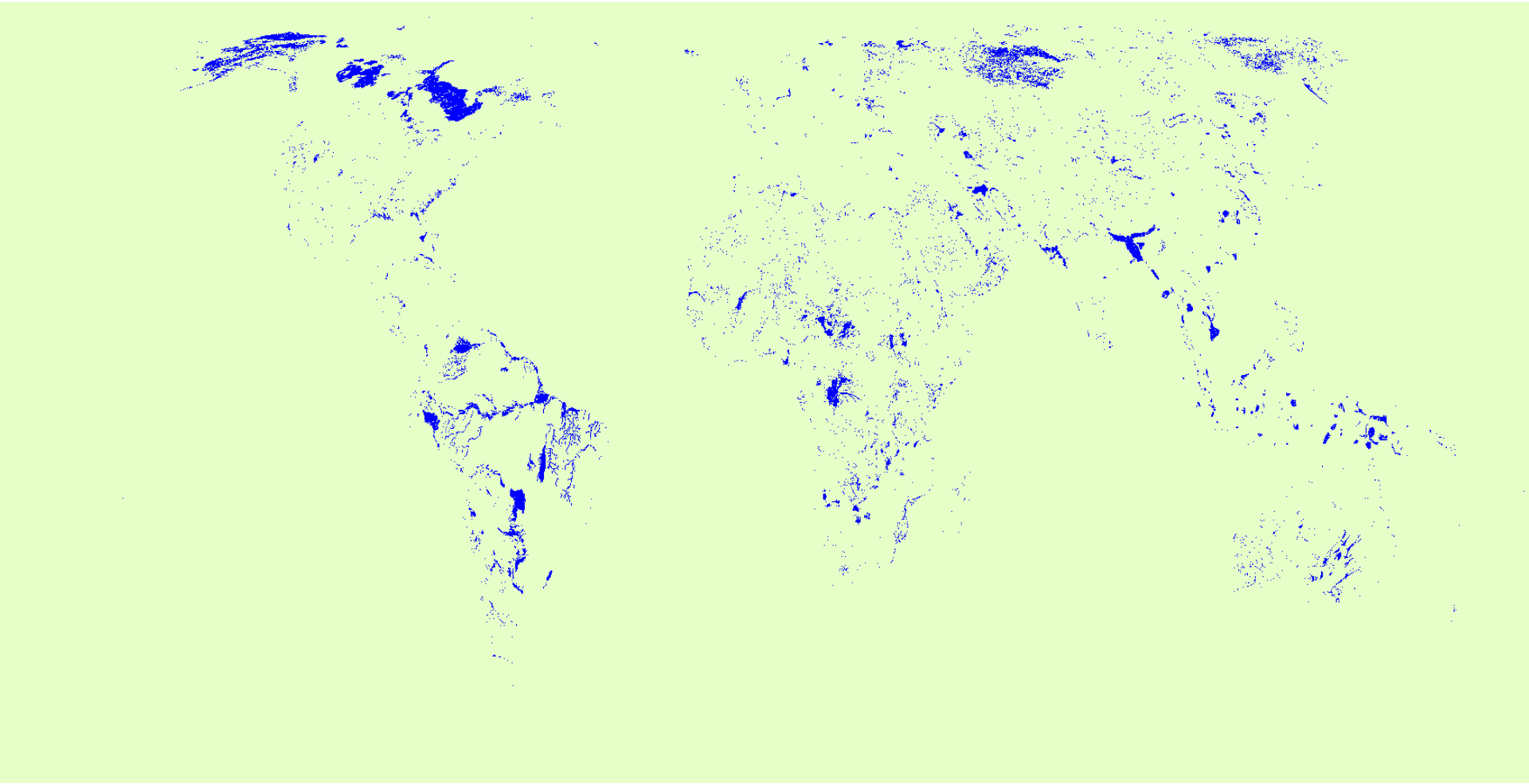
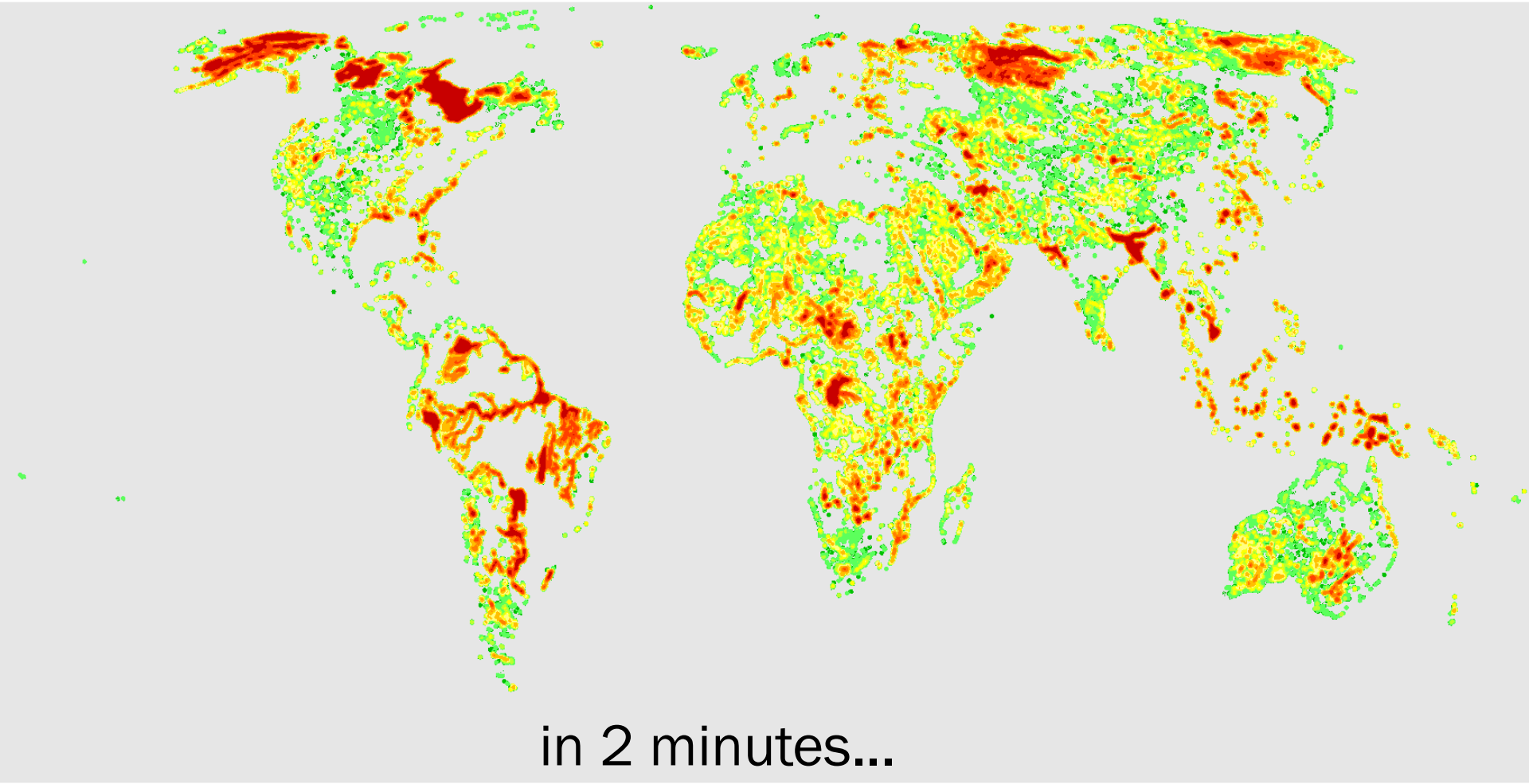
FROM



TO

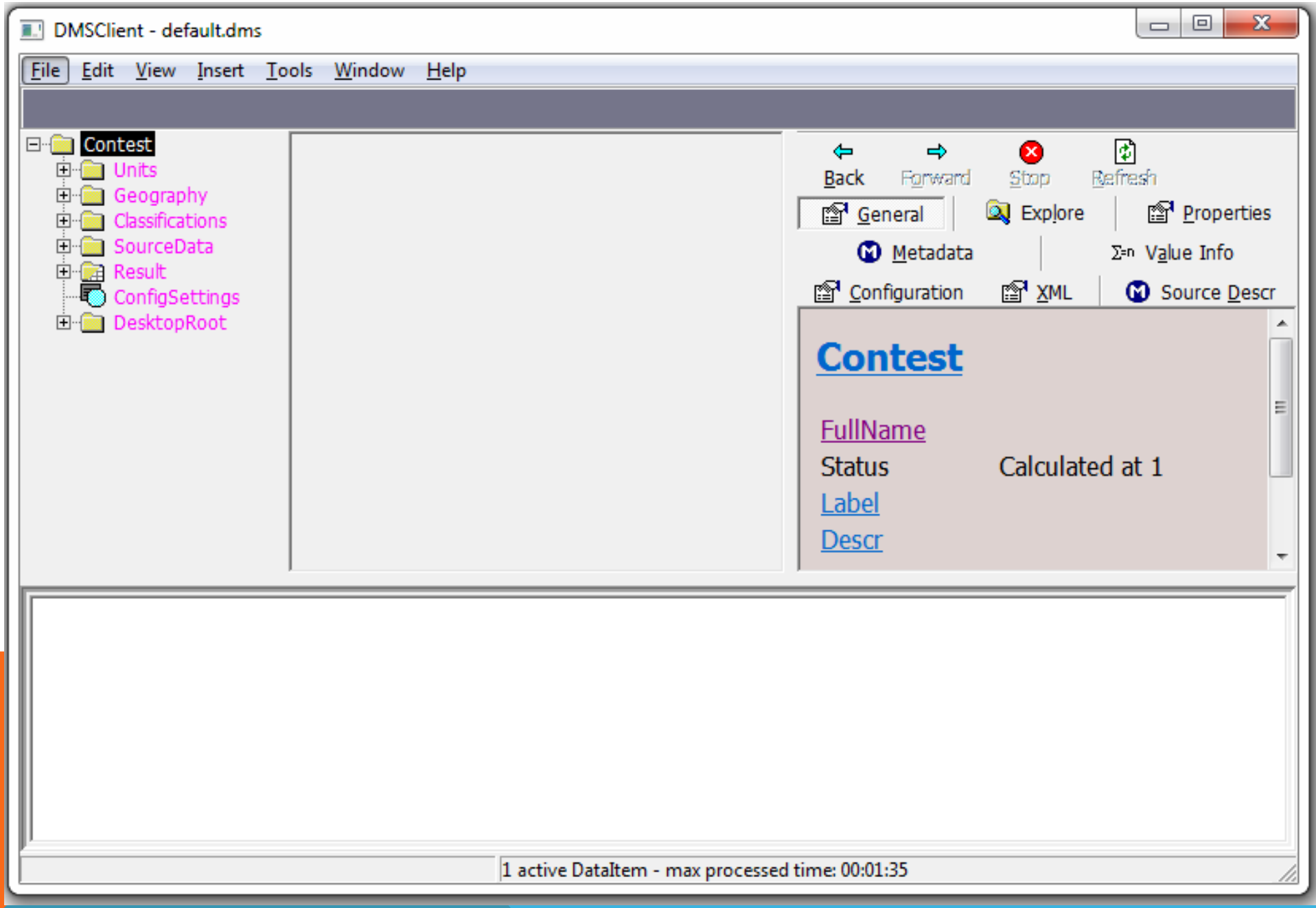


in 2 minutes...

WITH THE GEODMS

- Modelling framework developed by Object Vision, see: http://www.objectvision.nl/geodms
- Used for Land Use Scanner and EUClueScanner
- Focus on structured modelling, performance and large data sets
- Open Source Software (GNU-GPL)

GEODMS GUI


DMSClient - default.dms
File
Edit
View
Insert
Tools
Window
Help
Contest
Units
Geography
Classifications
SourceData
Result
ConfigSettings
DesktopRoot
Back
Forward
Stop
Refresh
General
Explore
Properties
Metadata
Value Info
Configuration
XML
Source Descr
Contest
FullName
Status
Calculated at 1
Label
Descr
1 active DataItem - max processed time: 00:01:35

c:\GeoDms\prj\SpaceToGo\cfg\Default.dms

GEODMS SCRIPT

```
container Contest
{
    container Units
    {
        unit<uint8> WelNietUint8: cdf = "Classifications/WelNietUint8_K2/Classes";
        unit<int16> Potentiaal:   cdf = "Classifications/potentiaal_K10/Classes";
    }

    container Geography: Using = "Units"
    {
        unit<fpoint> WorldBaseUnit;
        container DistMatrices
        {
            unit<uint32>  Dist2Range;
            unit<spoint>  pot50km: expr = "range(spoint, point(-int16(50), -int16(50)), point(int16(51) , int16(51)))"
            {
                attribute<Dist2Range> distMatr:  expr = "dist2(point(int16(0), int16(0), .), Dist2Range)";
                attribute<Potentiaal> AbsWeight: expr = "iif(distMatr <= uint32(2500), Int16(1.0) , Int16(0.0))";
            }
        }
    }

    #include <Classifications.dms>

    container SourceData
    {
        unit<wpoint> aig
        :   Descr           = "Arc/Info binary grid"
        ,   StorageName     = "%sourcedatadir%/SpaceToGo/Contest/wetlands1"
        ,   StorageType     = "gdal.grid"
        ,   DialogData      = "Geography/WorldBaseUnit"
        ,   StorageReadOnly = "True"
        {
            unit<wpoint>                  World1kmGrid:           Expr  = "TiledUnit(point(uint16(1024), uint16(1024),aig))";
            attribute<Units/WelNietUint8> ReadData (World1kmGrid): isHidden = "True";
            attribute<bool>               WetLands (World1kmGrid): Expr = "ReadData == 1[Units/WelNietUint8]";
        }
    }
    container Result
    {
        attribute<float32> Potential50km (SourceData/aig/World1kmGrid): Expr =
            "potential("
                "  float32(SourceData/aig/WetLands)"
                ", float32(Geography/DistMatrices/pot50km/AbsWeight)"
            ")"
        ,   StorageName = "%LocalDataProjDir%/Potential50km.tif";
    }
}
```

VIEW AND EXPORT RESULTS

- Source data and intermediate results can be visualised
  - SourceData/aig/WetLands
  - Kernel: Geography/DistrMatrices/pot50km/AbsWeight
- Results
  - Result/Potential50km_float, data is calculated, stored in tiff file and visualised in a mapview.
  - Path for resulting file is indicated in Detail Page General > Data Target.
  - After calculating, the pop-up menu option: export primary data > bitmap/tiff can also export a classified tiff file with a palette.
- GeoDMS Run.exe can batch generate results, cmd:
  - "C:\Program Files (x86)\ObjectVision\GeoDms6048\GeoDmsRun.exe" <*chosen projectfolder*>\cfg\default.dms /Result/Potential50km_float

COMPARISON

- Processing Times
  - ArcGIS: 26 hours, 24 minutes and 35 seconds (Alfred's Machine)
  - GeoDMS:

	1 year old laptop: - Windows 7-64 - Normal hardisk (no ssd) - 8 GB internal memory	Fast server: - Windows 7-64 - ssd harddisk - 24GB internal memory
• Reading source data • Calculate results • Export to tiff • Visualise in thematic Map	2 minutes and 48 seconds (566 times faster)	2 minutes and 0 seconds (792 times faster)

- Other Advantages of the GeoDMS:
  - The Results are stored in a CalcCache. They are directly availabe for future use.
  - Kernel can be adapted, e.g. to introduce a distance decay factor.

HOW CAN WE BE FAST

Modelstructuur:

- Modelbeschrijvingen zoveel mogelijk convex en niet chaotisch

Datastructuren:

- Memory Array based processing (as with R and IDL)
- Memory Mapped Files utilizes the paging mechanism to have data outside the working set standby without occupying Virtual Address Space.
- In het datamodel en bij rekenoperaties volledig ondersteunen van SubByte datatypes zodat in 1 DWORD kunnen worden opgeslagen: 32 Booleans, 16 Uint2 values (van 2 bits ieder) of 8 Uint4's (4 bits ieder).
- Arrays van Sequences zoals text strings, line strings en polygons zijn geïmplementeerd als dual arrays (één index array en één element value array) ter vermijding van heap memory allocations per object.
- Geometrische data elementen ook met single precision en integer coördinaten ondersteunen (en niet alleen met doubles zoals de meeste GIS software).
- Tiling aka Segmentation (for non rasterdata) to limit the use of the Virtual Address Space (VAS) and to divide calculation tasks.

Rekenmanagement:

- Symbolic rewriting teneinde veelvoorkomende simplificities toe te passen en dubbel rekenwerk te identificeren.
- CalcCache management teneinde tussenresultaten te bewaren totdat ze voor het laatst nodig zijn en niet langer.
- Optimaliseren van de volgorde van rekenwerk (bijvoorbeeld de afweging tussen per land alle periode's uitrekenen of per periode alle landen).
- Wel of niet ontkoppelen van pre- en post processing (data preparatie en indicator berekening als onderdeel of los van een simulatie uitvoeren).

Algorithmes:

- Sampling and Scaling.
- Convolution by FFT
- Gebruikt van ruimtelijke indexen in geometrische algoritmes en visualisatie.

Progammeer Architectuur

- Gebruik C++ templates, stl, boost,
- Bijhouden van run time properties van datasets voor selectie van de snelst toepasselijke algoritmes, zoals counting-sort en vergelijkbare methoden voor modus, join, etc.

Systeem Architectuur

- Gebruik hardware accelerated ibraries: Intel Performance Primitives, GDI

CONVOLUTION

http://en.wikipedia.org/wiki/Convolution

http://en.wikipedia.org/wiki/Convolution_theorem

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

Naive impl. of the '*' requires approx. $n_r \cdot n_c \cdot k \approx 5.84\text{x}10^{12}$ multiplications

with $n_r = 16924$ rows; $n_c = 33844$ cols; kernel size $k = 2 \cdot 50[\text{km}]+1 = 101$

Convolution by FFT approx. $3n \cdot \log_2(n)$ with $n= (n_r + k - 1) \cdot (n_r + k - 1) \approx 5\text{x}10^{10}$ muls.

With tiles of 1000^2 values each: $3n \cdot \log_2(1101^2) \approx 3.5\text{x}10^{10}$ muls. (31% better).

$\Rightarrow$Time factor due to FFT + tiling: $101^2 : 3 \cdot 20 \approx 170 : 1$

$\Rightarrow$Kernel sized tiles : $3n \cdot 2\log_2(2k\text{-}1) \approx 2.6\text{x}10^{10}$ muls (24% more).

WE CAN MAKE IT FASTER

- **Integer variant:** We already implemented an experimental version, which was about 33% faster, but has rounding errors
- **Multi Threading:** Use multiple cores to process tiles in parallel (up to 8x acceleration)
- Recycle the FFT of the kernel for tiles of the same size (up to 33% acceleration)
- Ipp 5.0 > Ipp 7.0 or FFTW
- Win64
- O(n) algorithm available for convolution with rectangular block filters

DO TRY THIS AT HOME

- Download sourcedata from (http://wtrns.fr/YJ1tf3zZzXu2kt) , unzip the source data to the folder: C:\SourceData\SpaceToGo\Contest
- Use TortoiseSVN to Checkout the project folder from: http://svn.objectvision.nl/public/geodms/trunk/prj/SpaceToGo (login with guest, guest), to a local folder. After downloading, this folder should contain:
  - a cfg\default.dms file
  - a cfg\default\classifications.dms file
  - a cfg\default\config.ini file
- Download and run the GeoDms6048-SetupW32.exe from http://svn.objectvision.nl/public/geodms/trunk/distr (login with guest, guest)
- Run the GeoDmsGui and open the cfg/default.dms in the chosen local folder.