Building a **strong** foundation for geospatial innovation

Martijn Visser (@visr), Deltares

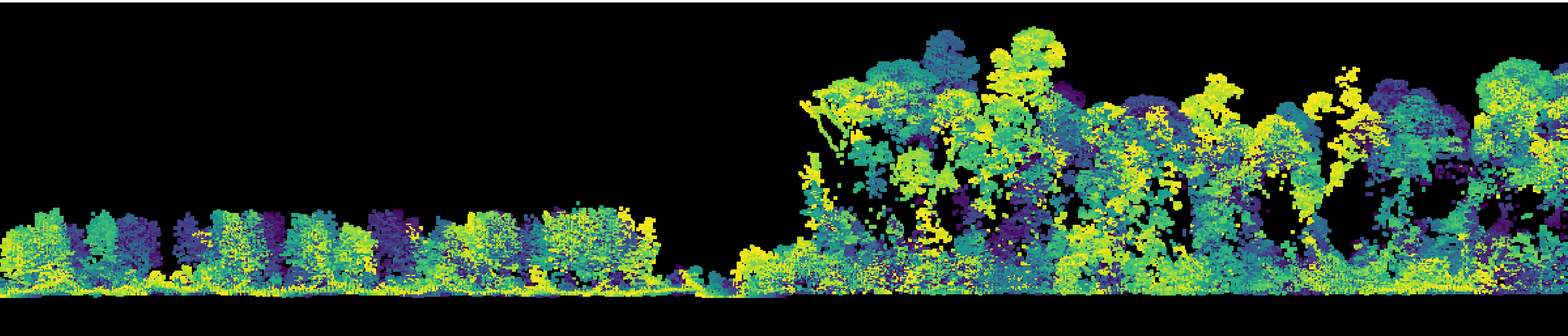
JuliaCon 2018

Deltares

Enabling Delta Life

What got me started on Julia

```julia
function profile!(canvas::Matrix, pointcloud, f::Function;
                  cm=20, zmin=940)
    ni, nj = size(canvas)
    for p in pointcloud
        i = ni - ((p.z - zmin) ÷ cm)
        j = (p.x + (nj ÷ 2) * cm) ÷ cm
        if checkbounds(Bool, canvas, i, j)
            canvas[i, j] = f(p)
        end
    end
end
```



Typical geospatial workflow is highly transorganizational

JuliaArrays

JuliaData

JuliaGeometry

JuliaGeo

JuliaImages

OSGeo

GeoStats.jl

GeometryTypes.jl

GMT.jl

GDAL.jl

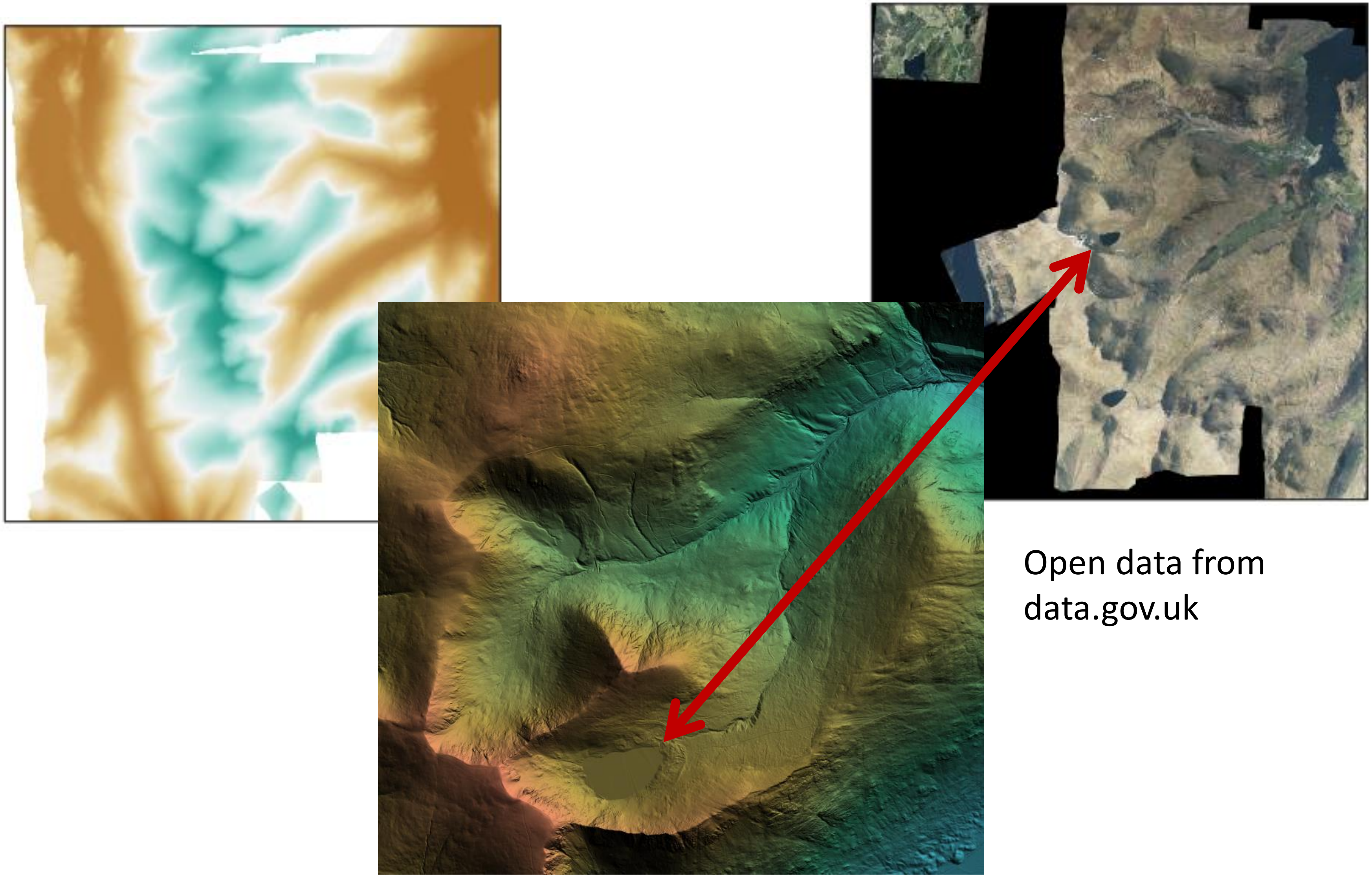
Images.jl

ArchGDAL.jl

GeoInterface.jl

AxisArrays.jl

Workflow demo



Open data from
data.gov.uk

To do / explore

- Better documentation / more examples
- Higher level interfaces
- Integrate rasters and AxisArrays.jl
- Integrate vectors and GeometryTypes.jl
- Finish converting to v0.7

https://github.com/yohanboniface/thank-you-map

In particular thanks to:
Yeesian Ng (@yeesian) – For building much of what is shown
Fabian Gans (@meggart) – Involvement in JuliaGeo since the beginning
Isaiah Norton (@ihnorton) – For Clang.jl, used to generate c wrappers
Elliot Saba (@staticfloat) – For BinaryBuilder.jl, used to provide binaries

And all other contributors! May there be many more ☺