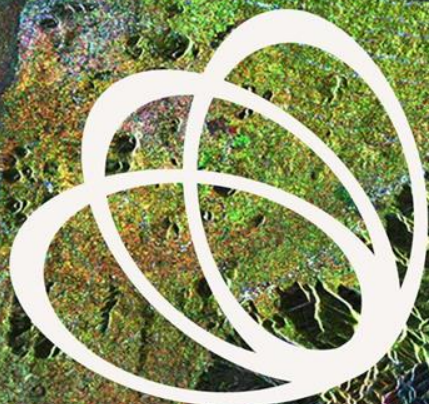


Forests 2020

CONFERENCE

Heiko Balzter

National Scale Mapping
and International
Reporting



The importance of national scale mapping

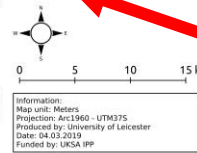
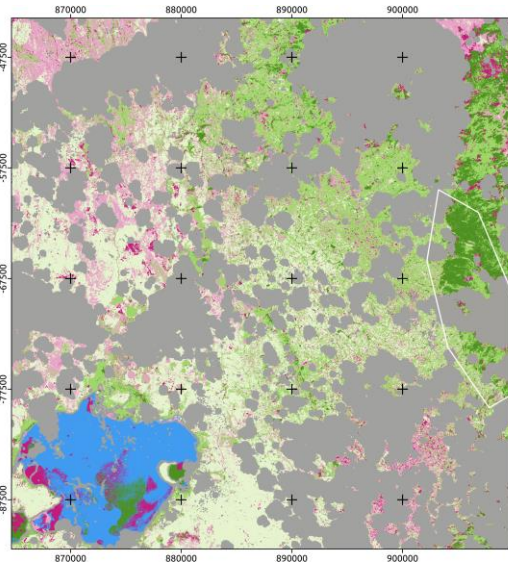
- Forest management
 - Forest inventory updating
 - Growing timber stock estimation → harvest planning
 - Monitoring protected areas
 - Protecting forest land from illegal logging and land conversion
 - Monitoring and safeguarding vital ecosystem services
- UN Framework Convention on Climate Change (UNFCCC)
 - Reporting greenhouse gas emissions and sequestration to UNFCCC
 - Greenhouse gas emissions accounting for the Kyoto Protocol & Paris Climate Agreement
 - Demonstrating REDD+ readiness
 - Attracting investment in carbon stocks via green climate finance



Challenges

- IPCC greenhouse gas accounting best practice guidance is based on **land use**, while satellite data gives us **land cover**.
- Need 'emission factors' for specific types of land use change and forest management.
- Need to establish Forest Reference Emission Levels (FREL) on past deforestation and degradation rates for at least a decade.
- Ongoing capacity building is needed to maintain ability to use the latest satellite imagery.
- Exponentially rising amounts of data require investment in data storage or usage of cloud computing or large computer centres.
- Phenology of dryland forests makes it difficult to monitor them on an ongoing basis.
- Cloud cover restricts the use of optical satellite data and creates the need for using radar.
- Near-real-time monitoring is needed to be able to act on deforestation information.
- Compatibility of satellite data with national forest inventories needs to be built.

University of Leicester Contributions to Forests 2020



PYEO

```
def create_new_image_from_polygon(polygon, out_path, x_res,
                                  projection, format="GTiff", dat
    """Returns an empty image of the extent of input polyg
    # TODO: Implement nodata
    bounds_x_min, bounds_x_max, bounds_y_min, bounds_y_max
    final_width_pixels = int((bounds_x_max - bounds_x_min)
    final_height_pixels = int((bounds_y_max - bounds_y_min)
    driver = gdal.GetDriverByName(format)
    out_raster = driver.Create(
        out_path, xsize=final_width_pixels, ysize=final_he
        bands=bands, eType=datatype
    )
    out_raster.SetGeoTransform([
        bounds_x_min, x_res, 0,
        bounds_y_max, 0, y_res * -1
    ])
    out_raster.SetProjection(projection)
    return out_raster

def resample_image_in_place(image_path, new_res):
    """Resamples an image in-place using gdalwarp to new
    # I don't like using a second object here, but hey
    with TemporaryDirectory() as td:
        args = gdal.WarpOptions(
            xRes=new_res,
            yRes=new_res
        )
        temp_image = os.path.join(td, temp_image.tif")
        gdal.Warp(temp_image, image_pa, options=args)
        shutil.move(temp_image, image_path)
```

