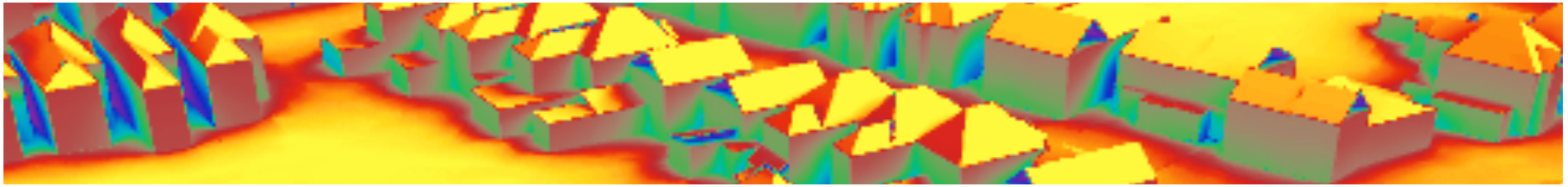
GIS and Irradiation Mapping

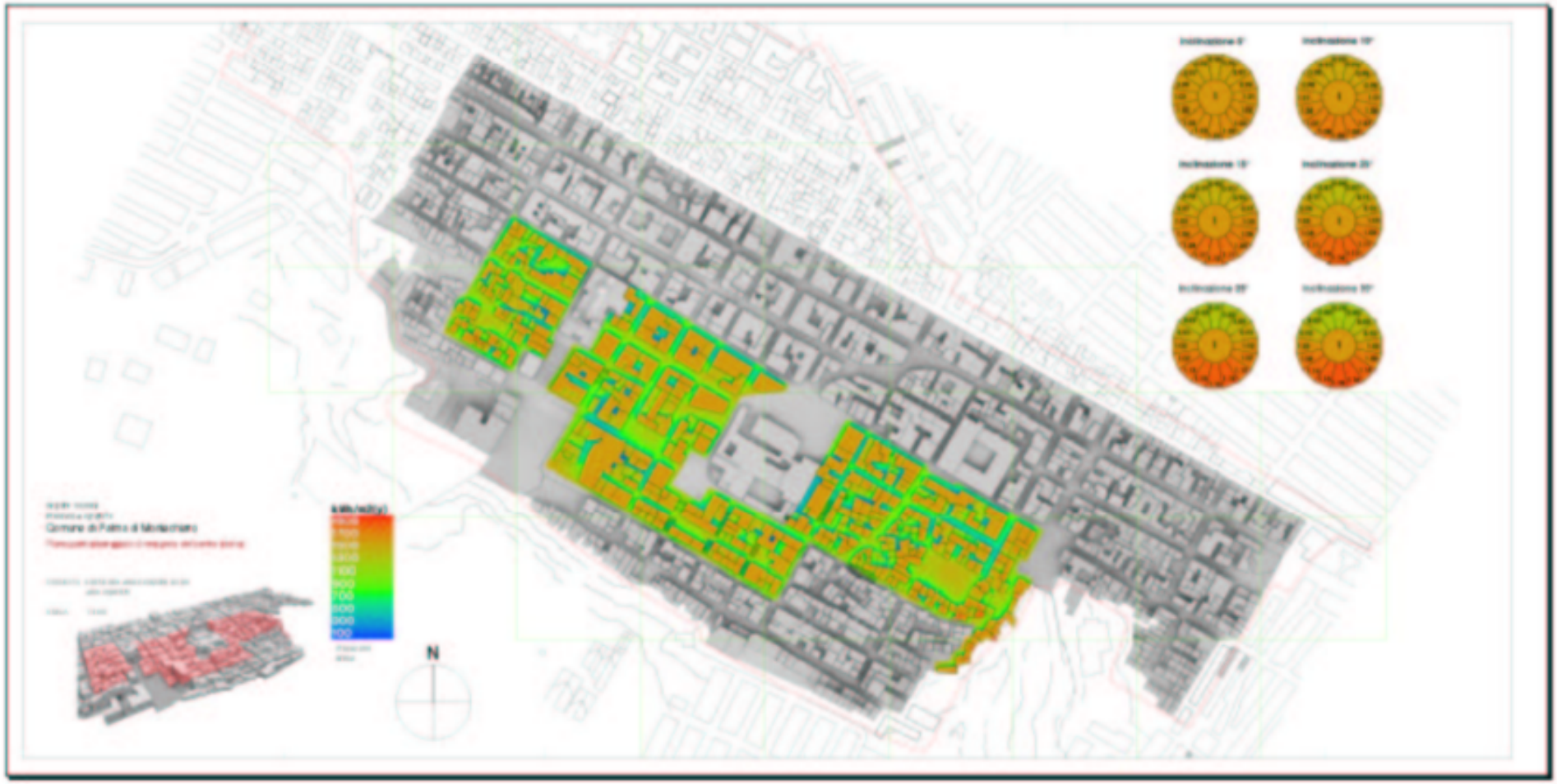


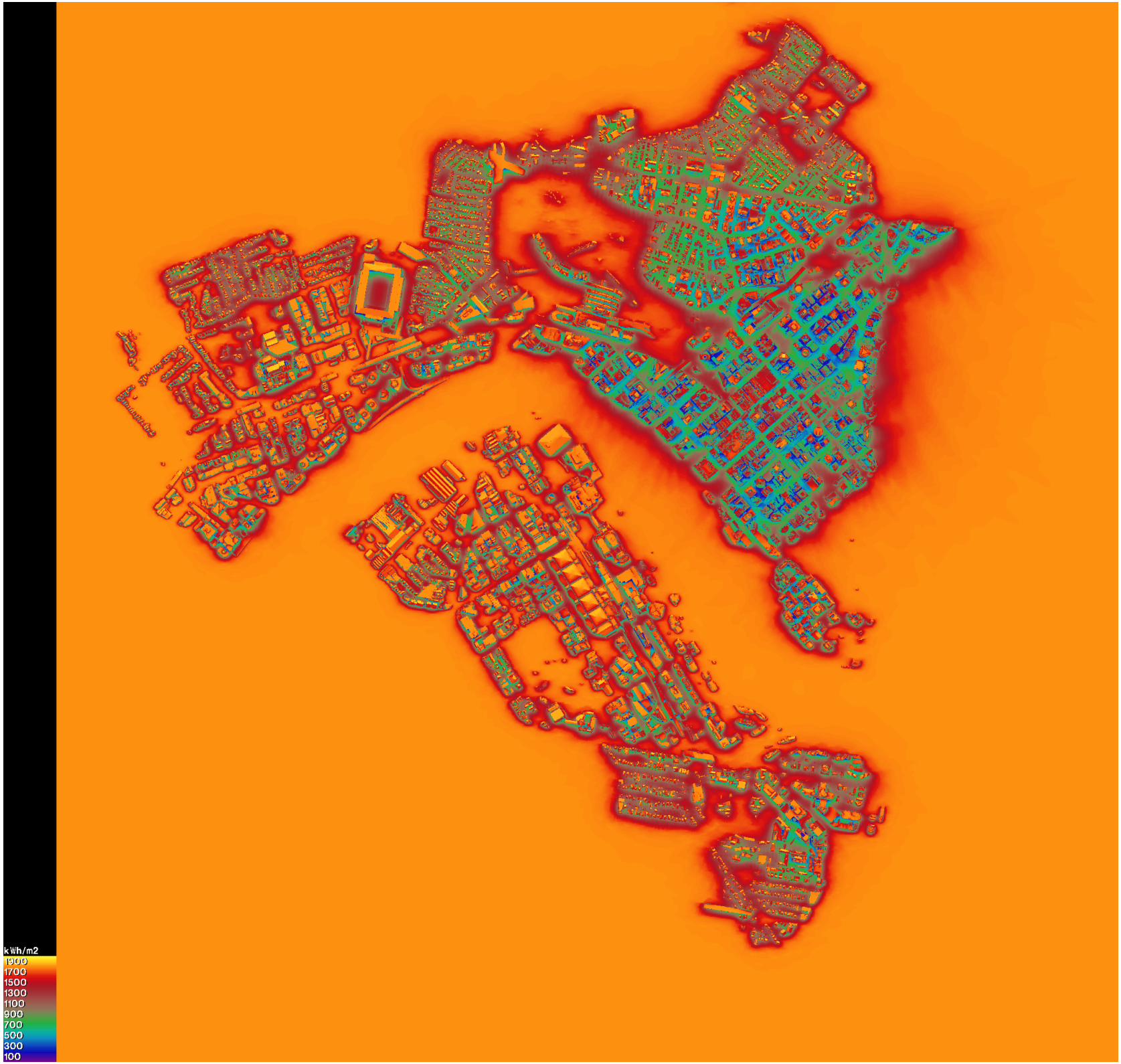
Francesco Anselmo

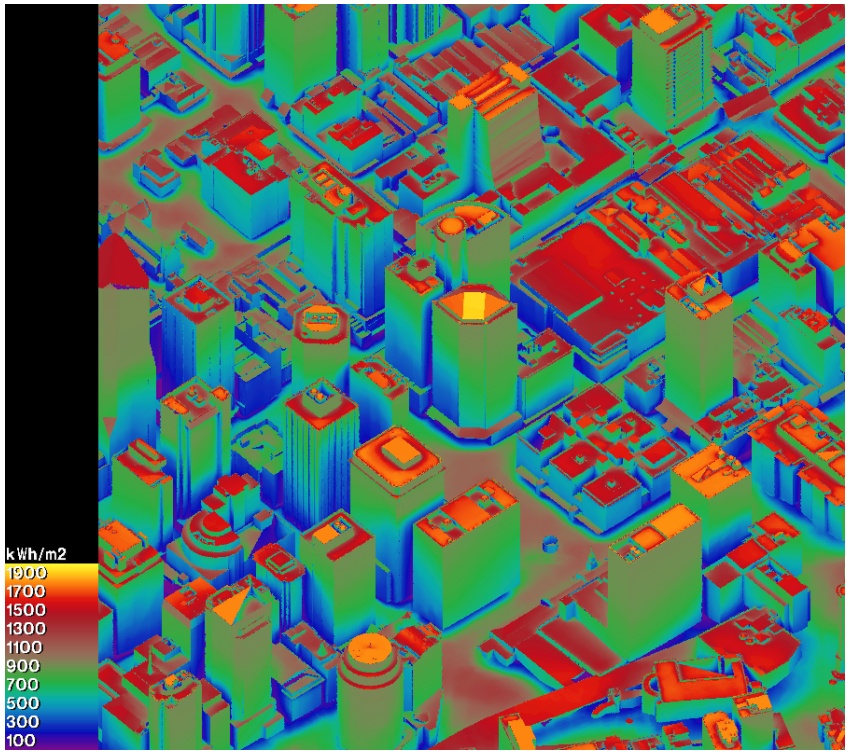
francesco.anselmo@arup.com

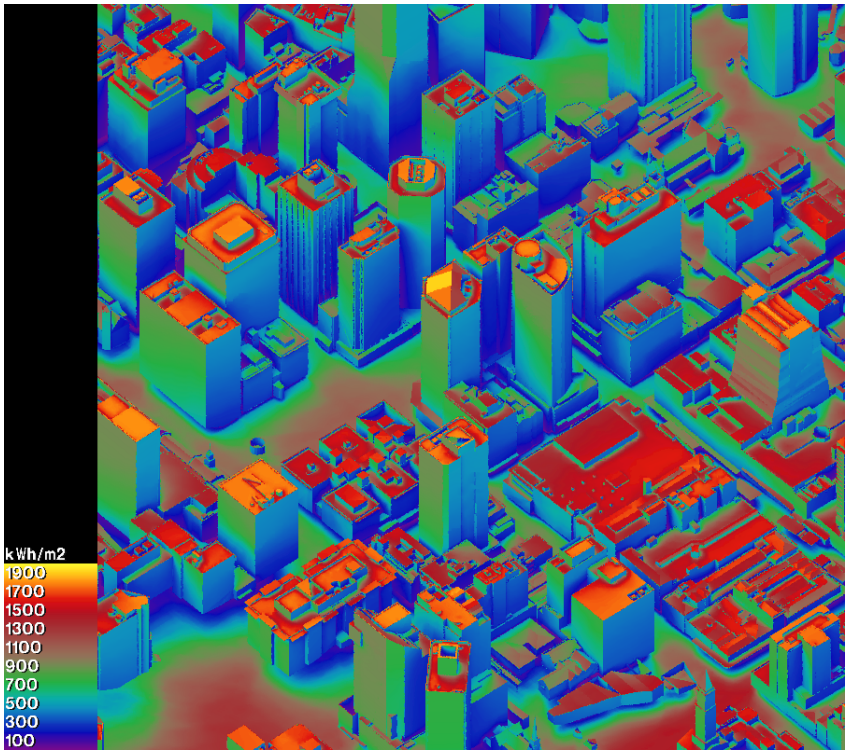
BBC

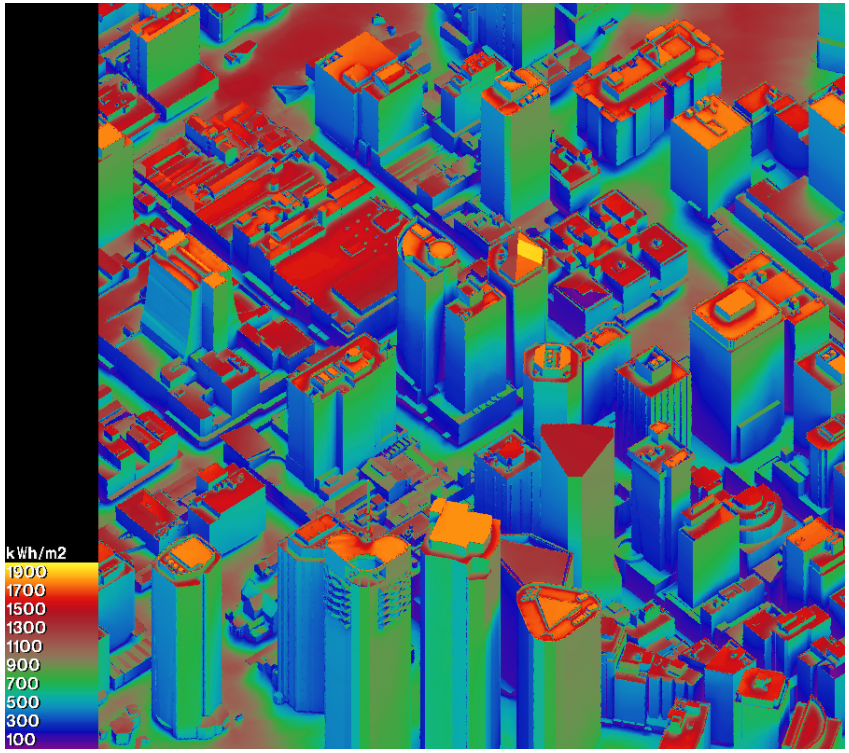
ARUP

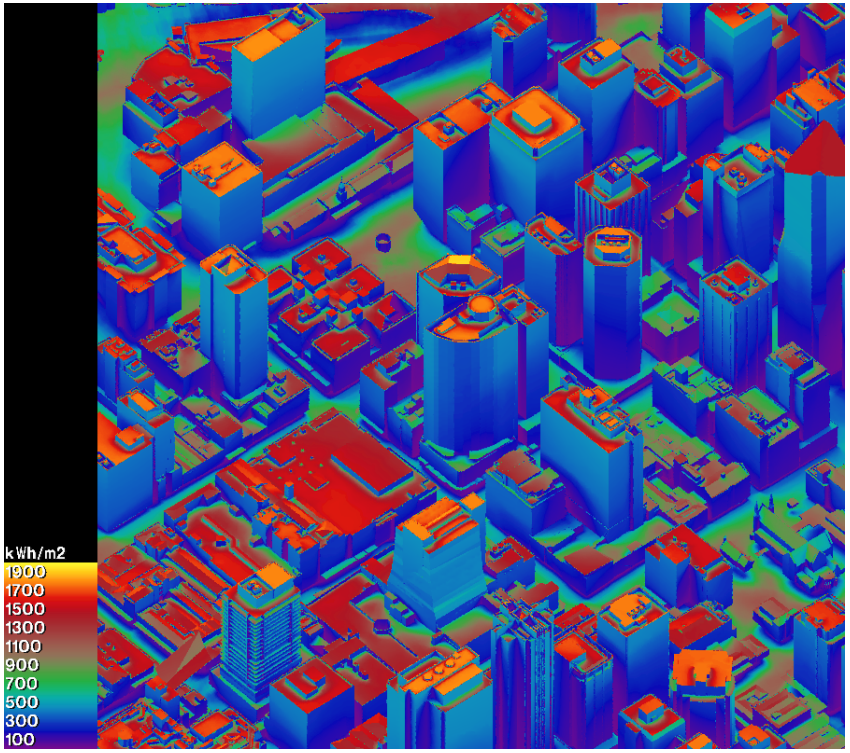

Inclinazione 0°
Inclinazione 10°
Inclinazione 15°
Inclinazione 25°
Inclinazione 35°
Inclinazione 50°
kWh/m²a
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
200
100
N

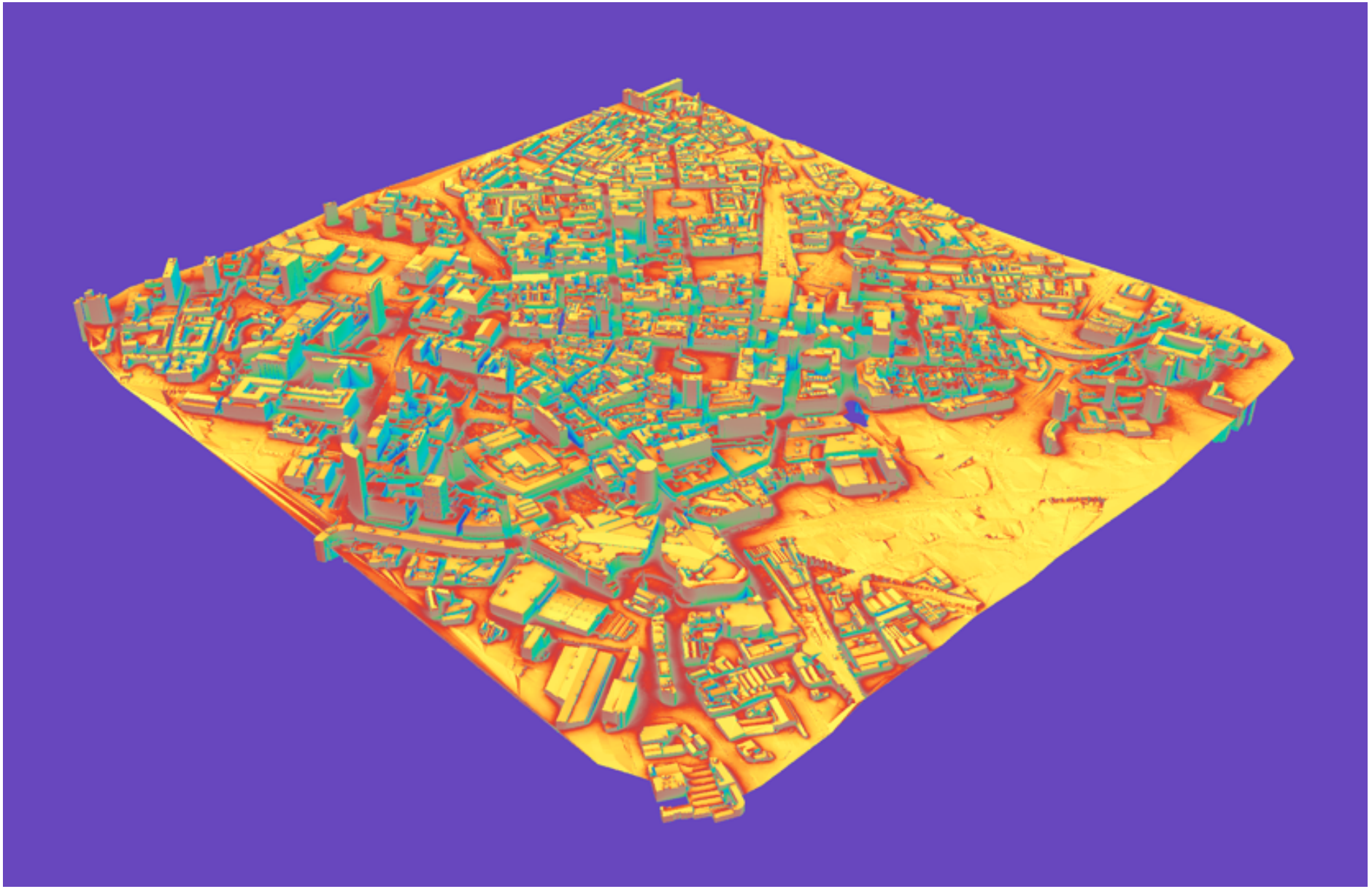

kWh/m2
1900
1700
1500
1300
1100
900
700
500
300
100

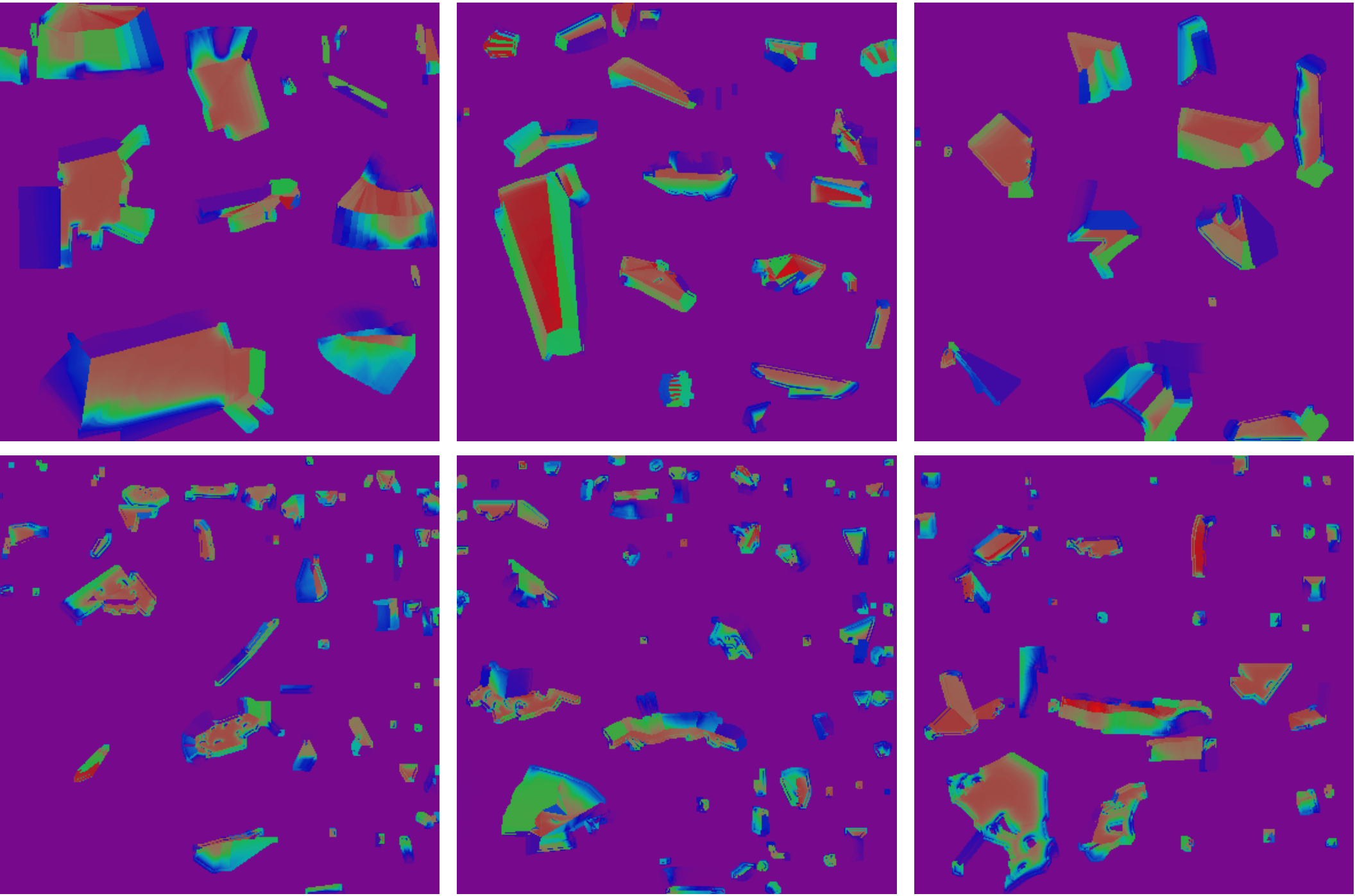

kWh/m2
1900
1700
1500
1300
1100
900
700
500
300
100

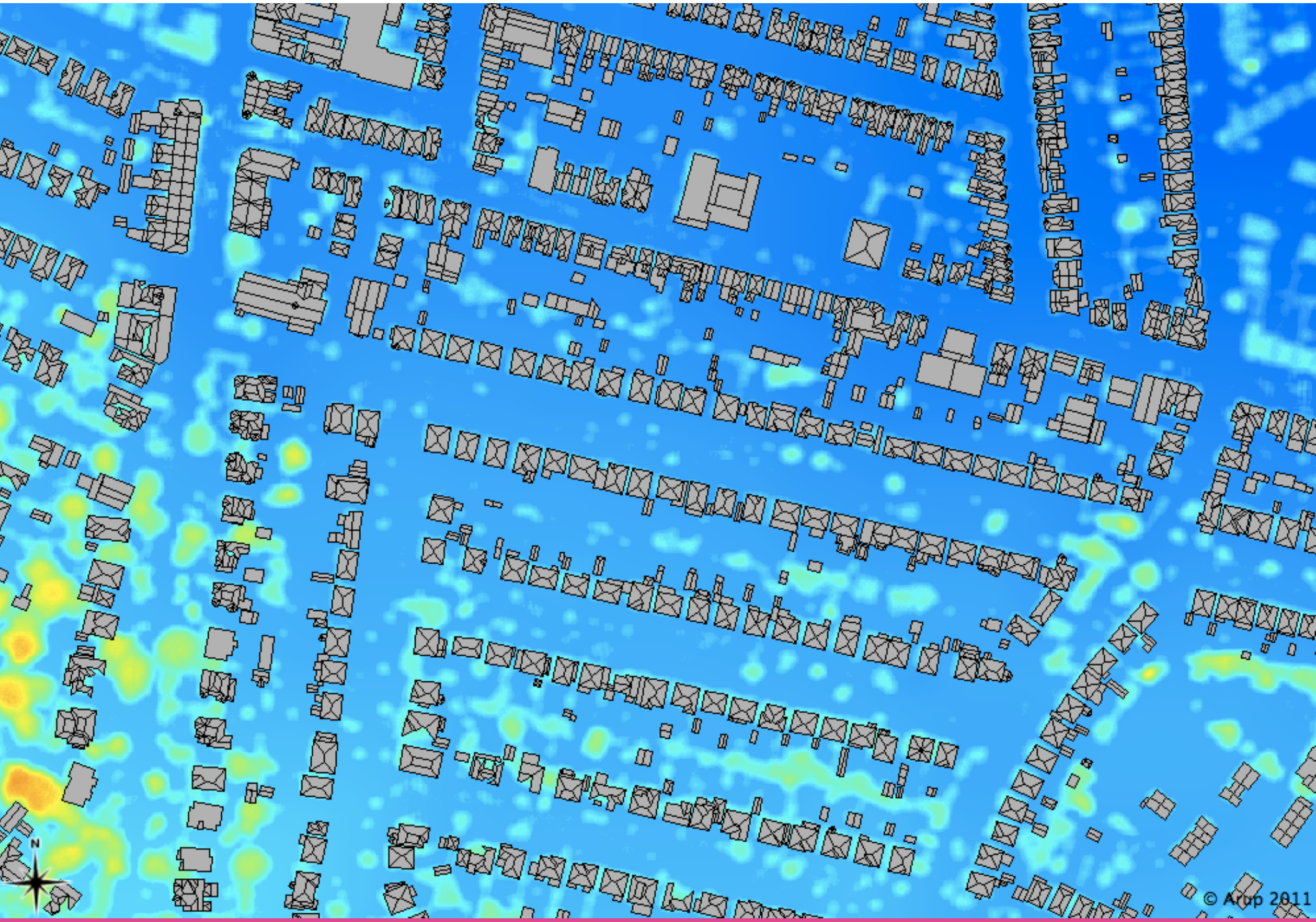

kWh/m2
1900
1700
1500
1300
1100
900
700
500
300
100


kWh/m2
1900
1700
1500
1300
1100
900
700
500
300
100


kWh/m2
1900
1700
1500
1300
1100
900
700
500
300
100

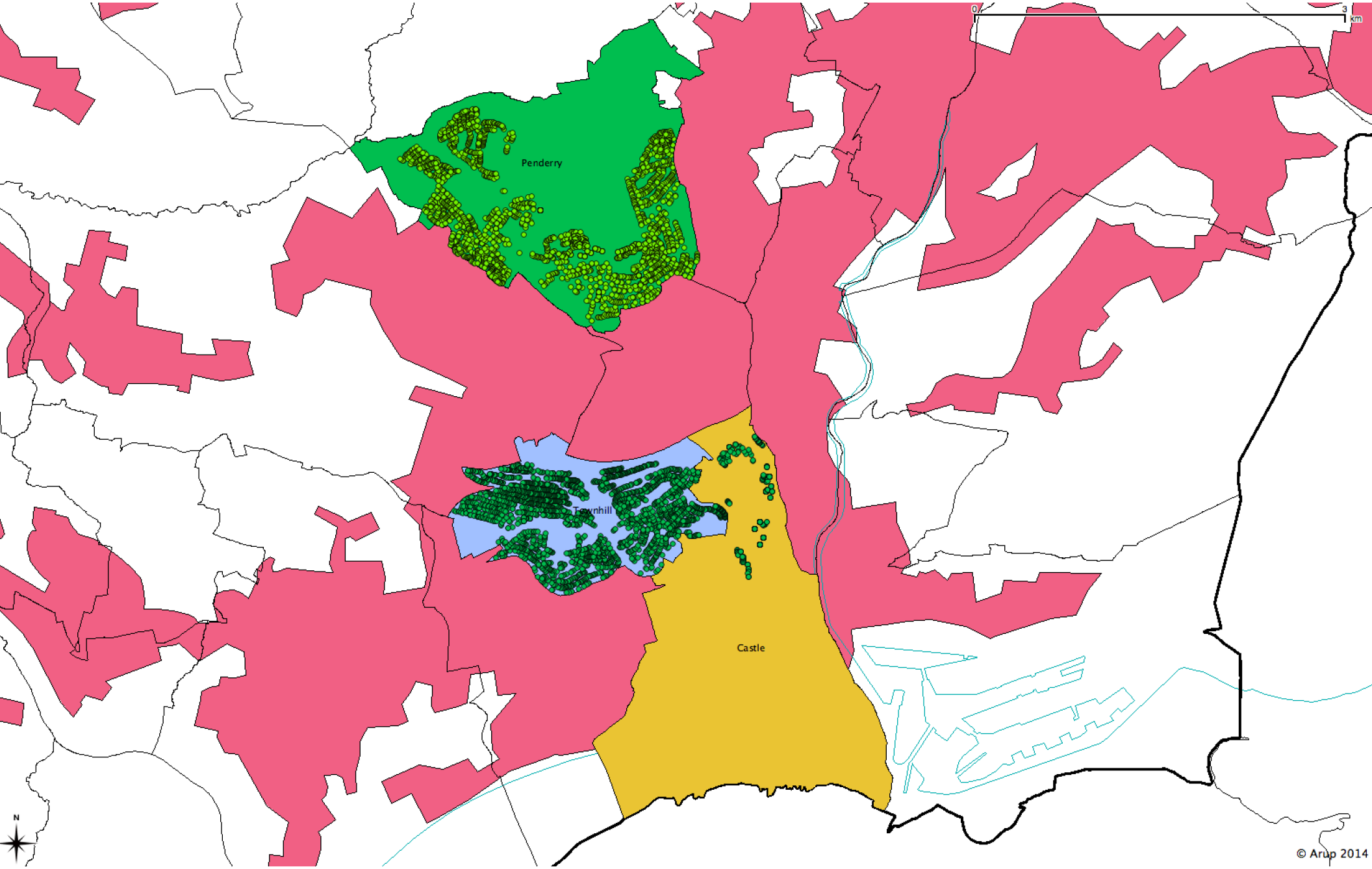






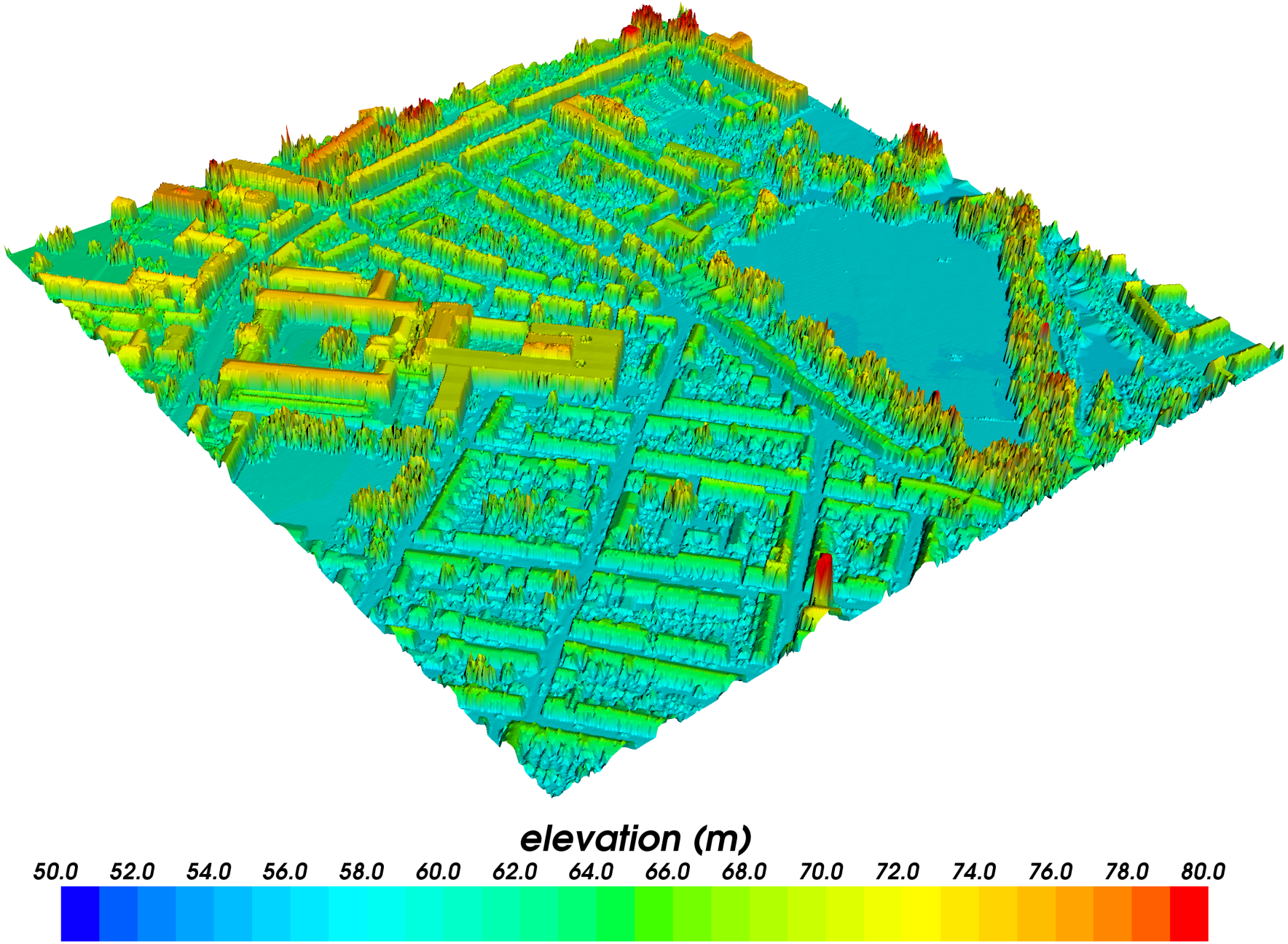
Geographic Information Systems / 3D GIS / Digital Terrain Models

ARUP



vector data / points, lines, polygons, shapes

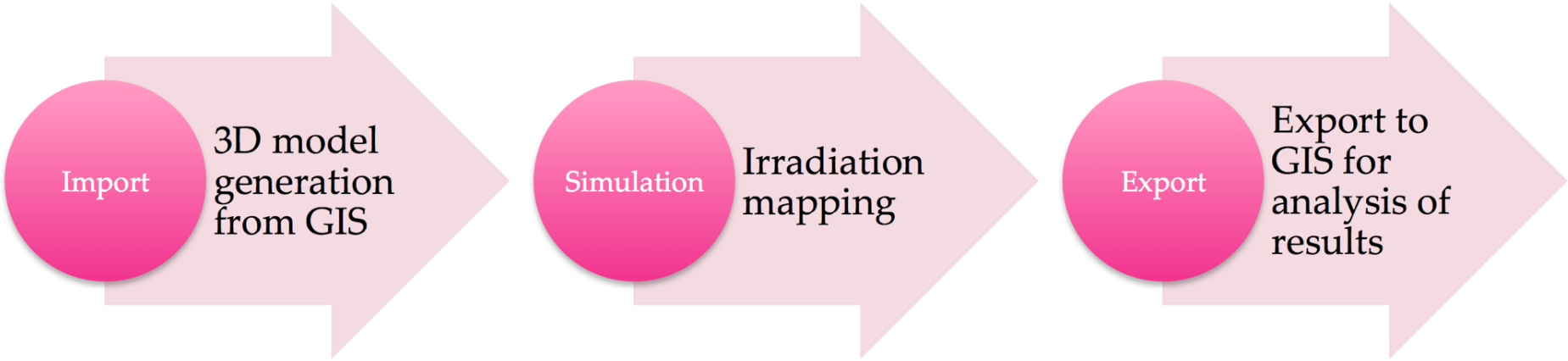
ARUP



raster data / LiDAR data

ARUP

Arup Irradiation Mapping Tool (iMAP)



- Integration of Geographic Information Systems (GIS) and the Radiance enables automation and highly efficient analysis of individual buildings as well as large areas with tens of thousands of addresses

ESRI Shapefile Technical Description

An ESRI White Paper—July 1998

Why Shapefiles?

A shapefile stores nontopological geometry and attribute information for the spatial features in a data set. The geometry for a feature is stored as a shape comprising a set of vector coordinates.

Because shapefiles do not have the processing overhead of a topological data structure, they have advantages over other data sources such as faster drawing speed and edit ability. Shapefiles handle single features that overlap or that are noncontiguous. They also typically require less disk space and are easier to read and write.

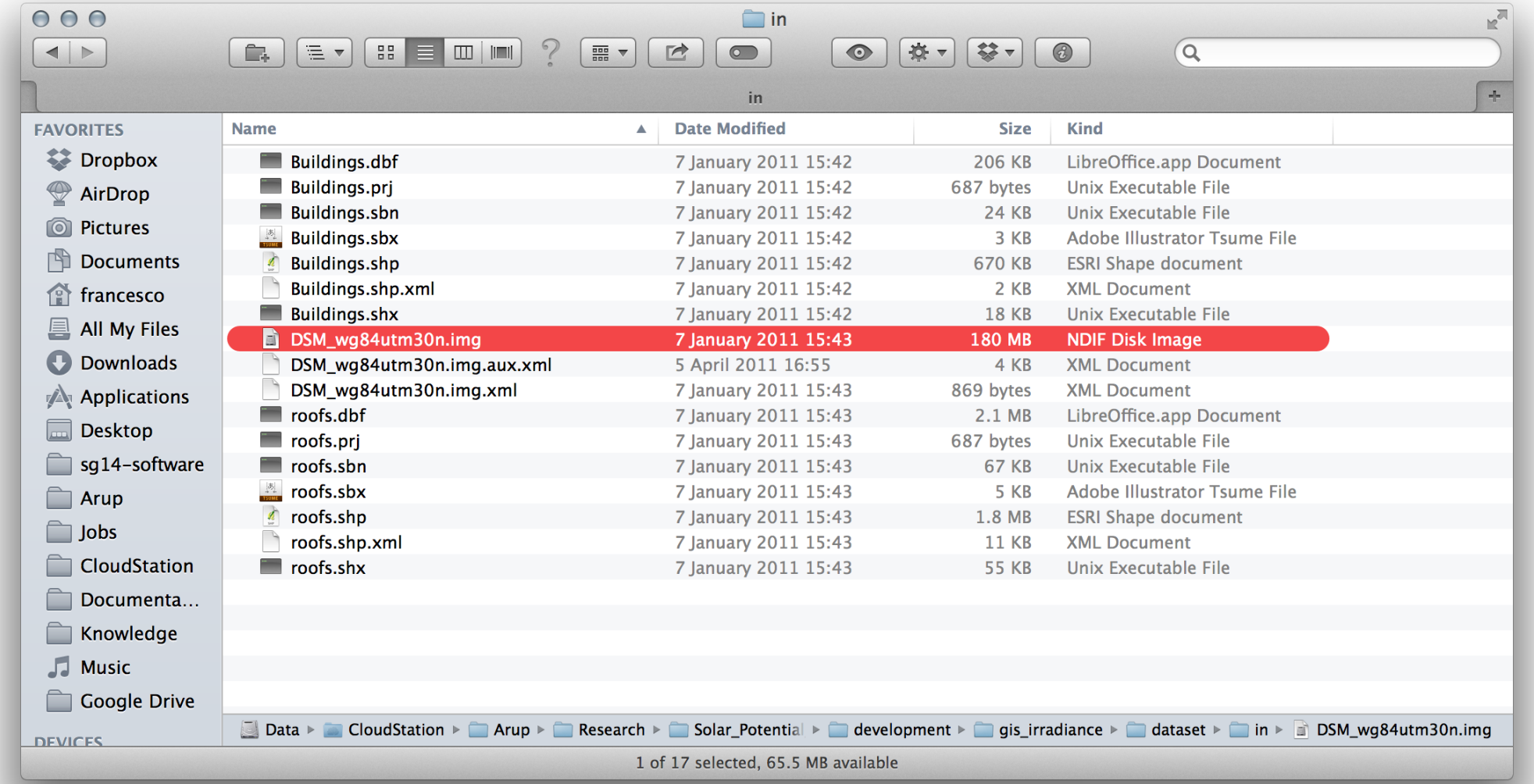
Shapefiles can support point, line, and area features. Area features are represented as closed loop, double-digitized polygons. Attributes are held in a dBASE® format file. Each attribute record has a one-to-one relationship with the associated shape record.

Table 1
Description of the Main File Header

Position	Field	Value	Type	Byte Order
Byte 0	File Code	9994	Integer	Big
Byte 4	Unused	0	Integer	Big
Byte 8	Unused	0	Integer	Big
Byte 12	Unused	0	Integer	Big
Byte 16	Unused	0	Integer	Big
Byte 20	Unused	0	Integer	Big
Byte 24	File Length	File Length	Integer	Big
Byte 28	Version	1000	Integer	Little
Byte 32	Shape Type	Shape Type	Integer	Little
Byte 36	Bounding Box	Xmin	Double	Little
Byte 44	Bounding Box	Ymin	Double	Little
Byte 52	Bounding Box	Xmax	Double	Little
Byte 60	Bounding Box	Ymax	Double	Little
Byte 68*	Bounding Box	Zmin	Double	Little
Byte 76*	Bounding Box	Zmax	Double	Little
Byte 84*	Bounding Box	Mmin	Double	Little
Byte 92*	Bounding Box	Mmax	Double	Little

* Unused, with value 0.0, if not Measured or Z type

Value	Shape Type
0	Null Shape
1	Point
3	PolyLine
5	Polygon
8	MultiPoint
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM
28	MultiPointM
31	MultiPatch


in
in
FAVORITES
Dropbox
AirDrop
Pictures
Documents
francesco
All My Files
Downloads
Applications
Desktop
sg14-software
Arup
Jobs
CloudStation
Documenta...
Knowledge
Music
Google Drive
DEVICES
Name
Date Modified
Size
Kind
Buildings.dbf 7 January 2011 15:42 206 KB LibreOffice.app Document
Buildings.prj 7 January 2011 15:42 687 bytes Unix Executable File
Buildings.sbn 7 January 2011 15:42 24 KB Unix Executable File
Buildings.sbx 7 January 2011 15:42 3 KB Adobe Illustrator Tsume File
Buildings.shp 7 January 2011 15:42 670 KB ESRI Shape document
Buildings.shp.xml 7 January 2011 15:42 2 KB XML Document
Buildings.shx 7 January 2011 15:42 18 KB Unix Executable File
DSM_wg84utm30n.img 7 January 2011 15:43 180 MB NDIF Disk Image
DSM_wg84utm30n.img.aux.xml 5 April 2011 16:55 4 KB XML Document
DSM_wg84utm30n.img.xml 7 January 2011 15:43 869 bytes XML Document
roofs.dbf 7 January 2011 15:43 2.1 MB LibreOffice.app Document
roofs.prj 7 January 2011 15:43 687 bytes Unix Executable File
roofs.sbn 7 January 2011 15:43 67 KB Unix Executable File
roofs.sbx 7 January 2011 15:43 5 KB Adobe Illustrator Tsume File
roofs.shp 7 January 2011 15:43 1.8 MB ESRI Shape document
roofs.shp.xml 7 January 2011 15:43 11 KB XML Document
roofs.shx 7 January 2011 15:43 55 KB Unix Executable File
Data ▸ CloudStation ▸ Arup ▸ Research ▸ Solar_Potential ▸ development ▸ gis_irradiance ▸ dataset ▸ in ▸ DSM_wg84utm30n.img
1 of 17 selected, 65.5 MB available

- Commercial
  - ArcGIS
- Open Source
  - QGIS
  - Grass
  - GDAL-OSR
  - pyshp
  - shapely
  - scipy/numpy/matplotlib
  - libLAS
  - mapnik

pyshp

Python Shapefile Library

Search projects

Project Home | Downloads | Wiki | Issues | Source

Summary | People

Project Information

Project feeds

Code license
MIT License

Labels
python, shapefiles, shapefile, gis, dbf, shp, mapping, geospatial, esri

Members
geospati...@gmail.com, jlawh...@geospatialpython.com, import.s...@gmail.com, jlawh...@gmail.com
1 committer

Featured

Downloads
sbn_format.pdf
sbx_format.pdf
Show all »

Links

Blogs
Geospatial Python

External links
Book: Learning Geospatial Analysis with Python"
NVisionSolutions.com
GeospatialPython Sample Code Project Site
SpatialPython on Twitter
Other Geospatial Python Books
Official ESRI Shapefile Specification
XBase (dbf) Specification

Groups
Geospatial Python Discussion Group

Overview

NOTE: This project has been migrated to Github. This google code site will be available indefinitely. As of June 16, 2014, the source and some wiki pages have been migrated over.

This library reads and writes ESRI Shapefiles in *pure* Python. You can read and write shp, shx, and dbf files with all types of geometry. Everything in the public ESRI shapefile specification is implemented. This library is compatible with Python versions 2.4 to 3.x.

Get Started Instantly

1. Download shapefile.py
2. Start Python
3. `import shapefile`
4. Try one of the examples below

OR

Just run: `easy_install pyshp`

OR

```
pip install pyshp
```

If you are looking for information on .sbn and .sbx file formats some documentation is available here.

Latest News

8/8/2013 - Please upgrade to PyShp 1.1.9 which fixes issues with polylines, polygons, as well as some z-value corner cases. This update resolves (~~Issue 54~~), (~~Issue 55~~), and (~~Issue 56~~).

6/23/2013 - Released PyShp 1.1.7! This release fixes several bugs including (~~Issue 40~~), (~~Issue 37~~), (~~Issue 25~~), and (~~Issue 22~~). Other improvements include:

- Added Python geo_interface convention to export shapefiles as GeoJSON.
- Used is_string() method to detect file names passed as unicode strings.
- Added Reader.iterShapes() method to iterate through geometry records for parsing large files efficiently.
- Added Reader.iterRecords() method to iterate through dbf records efficiently in large files.
- Modified shape() method to use iterShapes() if shx file is not available.
- Fixed bug which prevents writing the number 0 to dbf fields.
- Updated *shape() method to calculate and seek the start of the next record. The shapefile spec does not require the content of a geometry record to be as long as the content length defined in the header. The result is you can delete features without modifying the record header allowing for empty space in records.*

Reading Points in Shapes

```
>>> import shapefile
>>> sf = shapefile.Reader("shapefiles/blockgroups")
>>> shapes = sf.shapes()
>>> # Read the bounding box from the 4th shape
>>> shapes[3].bbox
[-122.485792, 37.786931000000003, -122.446285, 37.811019000000002]
>>>#  Read the 8th point in the 4th shape
>>> shapes[3].points[7]
[-122.471063, 37.787402999999998]
```

Reading Database Attributes

```
>>> # Read the field descriptors for the database file
>>> sf.fields
[("DeletionFlag", "C", 1, 0), ["AREA", "N", 18, 5],
... ["BKG_KEY", "C", 12, 0], ["POP1990", "N", 9, 0], ["POP90_SQMI", "N", 10, 1],
... ["HOUSEHOLDS", "N", 9, 0],
... ["MALES", "N", 9, 0], ["FEMALES", "N", 9, 0]]
>>> # Read the 2nd and 3rd field values of the 4th database record
>>> sf.records[3][1:3]
['060750601001', 4715]
```

Writing Shapefiles

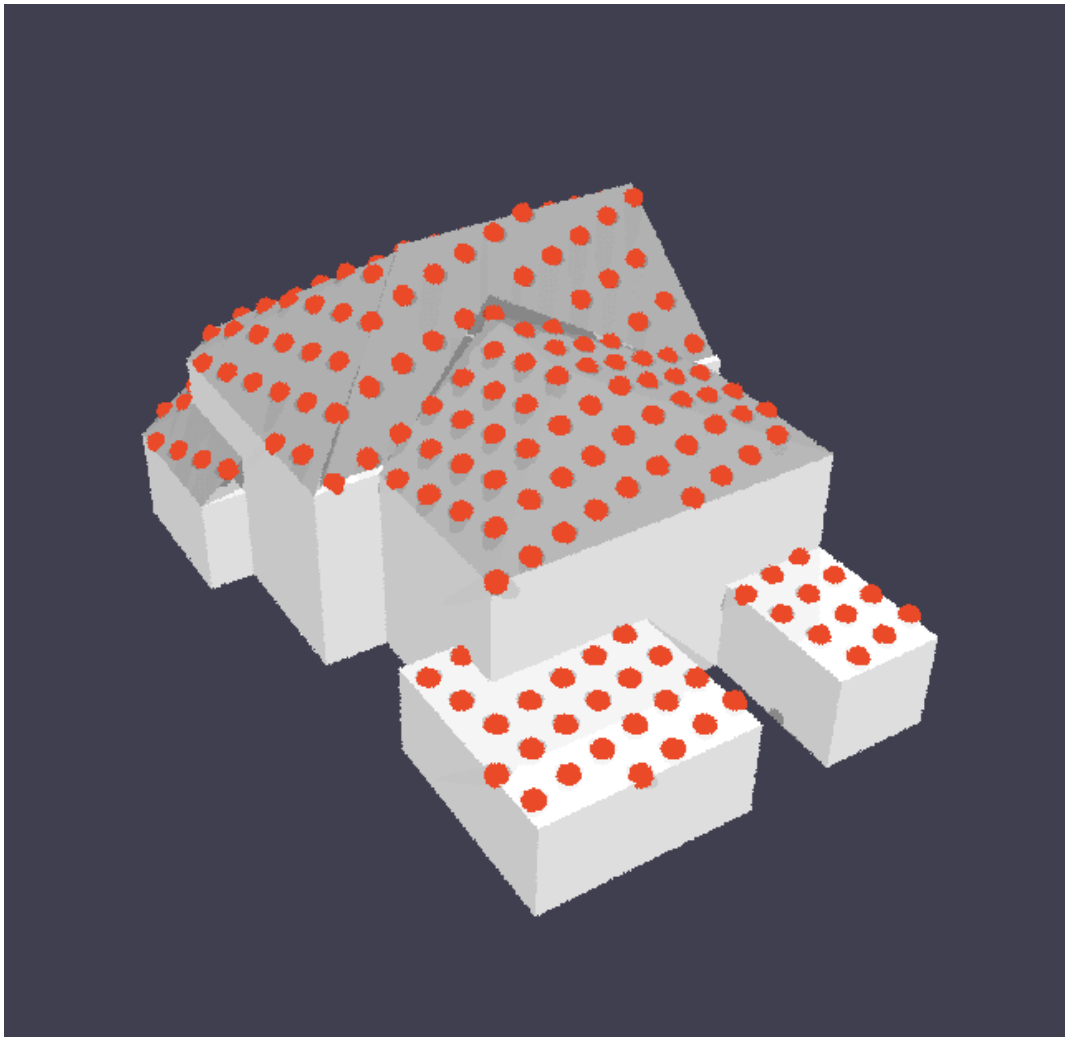
```
>>> import shapefile
>>> # Make a point shapefile
>>> w = shapefile.Writer(shapefile.POINT)
>>> w.point(90.3, 30)
>>> w.point(92, 40)
>>> w.point(-122.4, 30)
>>> w.point(-90, 35.1)
>>> w.field('FIRST_FLD')
>>> w.field('SECOND_FLD','C','40')
>>> w.record('First','Point')
>>> w.record('Second','Point')
>>> w.record('Third','Point')
>>> w.record('Fourth','Point')
>>> w.save('shapefiles/test/point')
>>> # Create a polygon shapefile
>>> w = shapefile.Writer(shapefile.POLYGON)
>>> w.poly(parts=[[[1,5],[5,5],[5,1],[3,3],[1,1]]])
>>> w.field('FIRST_FLD','C','40')
>>> w.field('SECOND_FLD','C','40')
>>> w.record('First','Polygon')
>>> w.save('shapefiles/test/polygon')
```

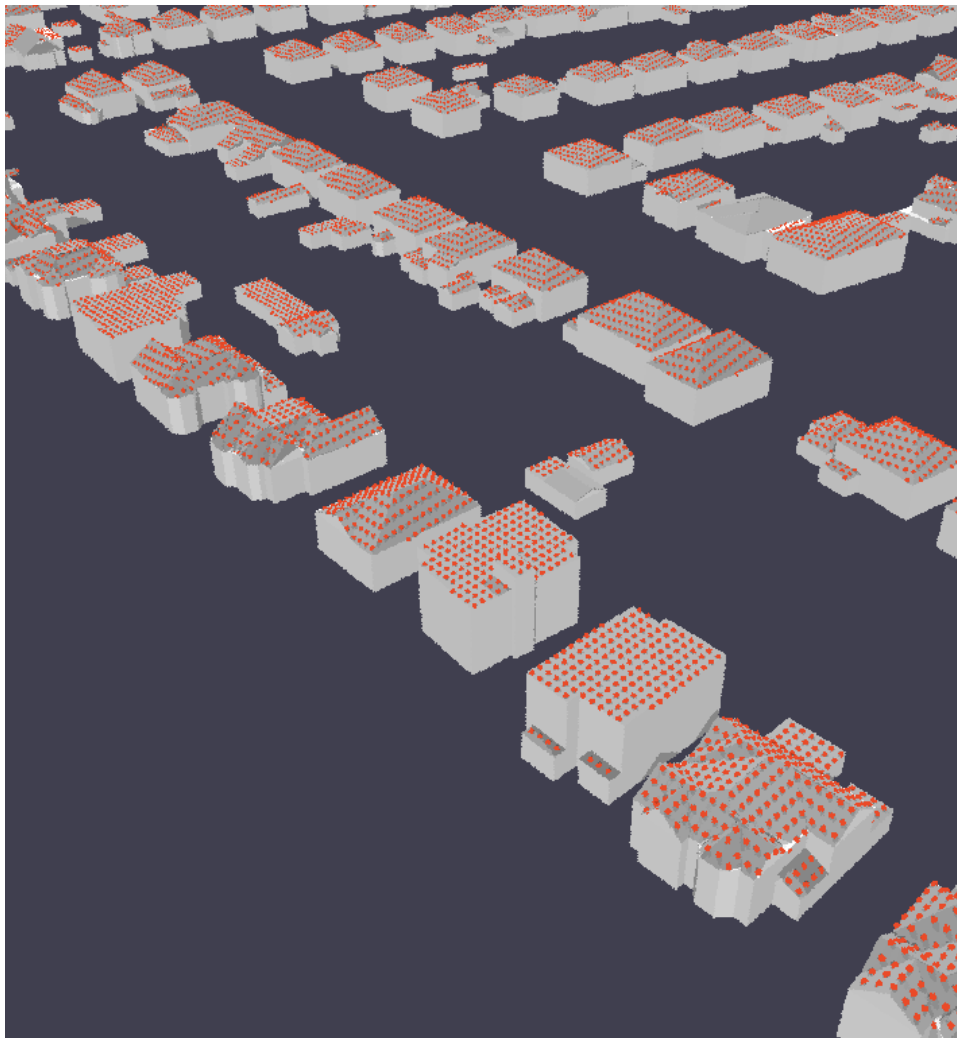
```
PROJCS[“WGS_1984_UTM_Zone_30N",
    GEOGCS[“GCS_WGS_1984",
        DATUM[“D_WGS_1984",
        SPHEROID[“WGS_1984",6378137.0,298.257223563]],
        PRIMEM[“Greenwich”,0.0],UNIT["Degree",0.0174532925199433]],

    PROJECTION[“Transverse_Mercator"],
    PARAMETER[“False_Easting",500000.0],
    PARAMETER[“False_Northing",0.0],
    PARAMETER[“Central_Meridian",-3.0],
    PARAMETER[“Scale_Factor",0.9996],
    PARAMETER[“Latitude_Of_Origin",0.0],
    UNIT["Meter",1.0]]
```

```
from sys import argv, exit, stdout
from os.path import exists, isdir, join, basename, splitext
import shapefile
from os import system, mkdir, listdir
from commands import getoutput
from subprocess import Popen
from optparse import OptionParser
from shapely.geometry import Polygon
from math import sqrt, atan, asin, acos, degrees, radians, pi
from scipy import matrix
from numpy import dot, cross
import vtk
import pyproj
from osgeo import ogr
import mapnik
from PIL import Image
```

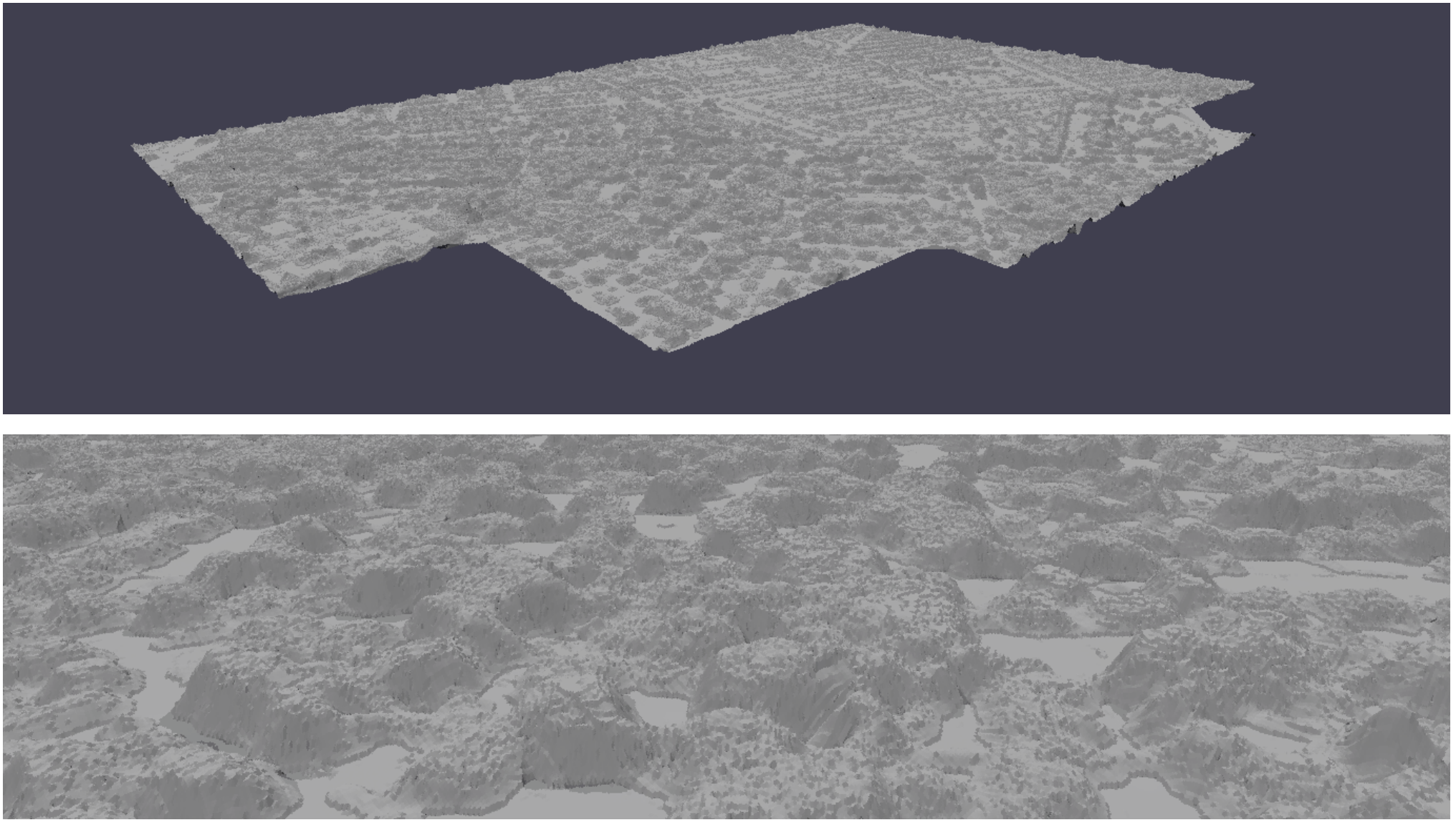
```
sf = get_options() #, material
shapefilename, shapefileext = get_name_ext(sf)
#print shapefilename, shapefileext
if exists(sf):
                    # get barycentre from bounding box
                    rectangle = getbbox(shapefilename)
                    print '%s bounding box:' % sf,  rectangle
                    orig_X = (rectangle[2]+rectangle[0])/2
                    orig_Y = (rectangle[3]+rectangle[1])/2
                    driver = ogr.GetDriverByName('ESRI Shapefile')

                    # get source projection
                    ds = driver.Open(sf)
                    layer = ds.GetLayer()
                    sr = layer.GetSpatialRef()
                    print 'source projection:', sr.ExportToProj4()
                    print 'UTM Zone:', sr.GetUTMZone()

                    srcProj = pyproj.Proj(sr.ExportToProj4())
                    dstProj = pyproj.Proj(proj='longlat', ellps='WGS84', datum='WGS84')

                    print 'dest projection:', dstProj.srs

                    longitude, latitude = pyproj.transform(srcProj, dstProj, orig_X, orig_Y)
                    dest_X, dest_Y = pyproj.transform(dstProj, srcProj, out_lon, out_lat)
                    dest_long, dest_lat = pyproj.transform(srcProj, dstProj, out_x, out_y)

                    print "UTM zone %s coordinate x, y (%0.4f, %0.4f) = (%0.4f, %0.4f) lat, long " % (sr.GetUTMZone(), orig_X, orig_Y, latitude, longitude)
                    print "lat, long coordinate x, y (%0.4f, %0.4f) = (%0.4f, %0.4f) lat, long " % (out_lat, out_lon, dest_X, dest_Y)
                    print "UTM zone %s coordinate x, y (%0.4f, %0.4f) = (%0.4f, %0.4f) lat, long " % (sr.GetUTMZone(), out_x, out_y, dest_lat, dest_long4)
```

Automatic generation of 3D model and sampling points on roofs from GIS shape-files

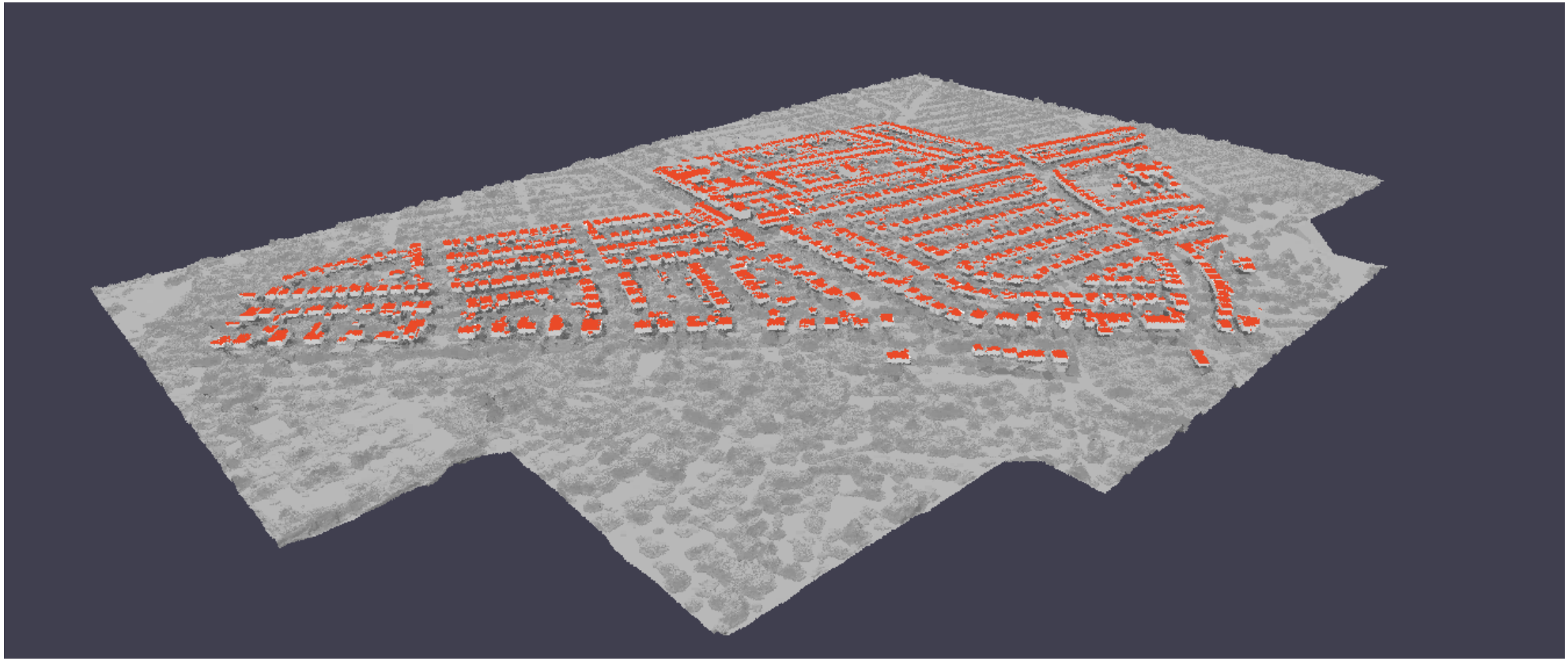




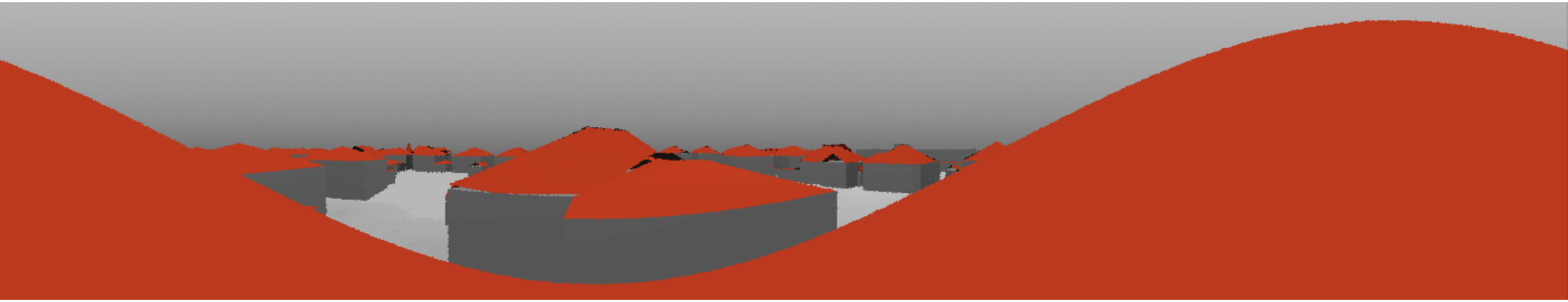
Terrain and obstructions model from LIDAR

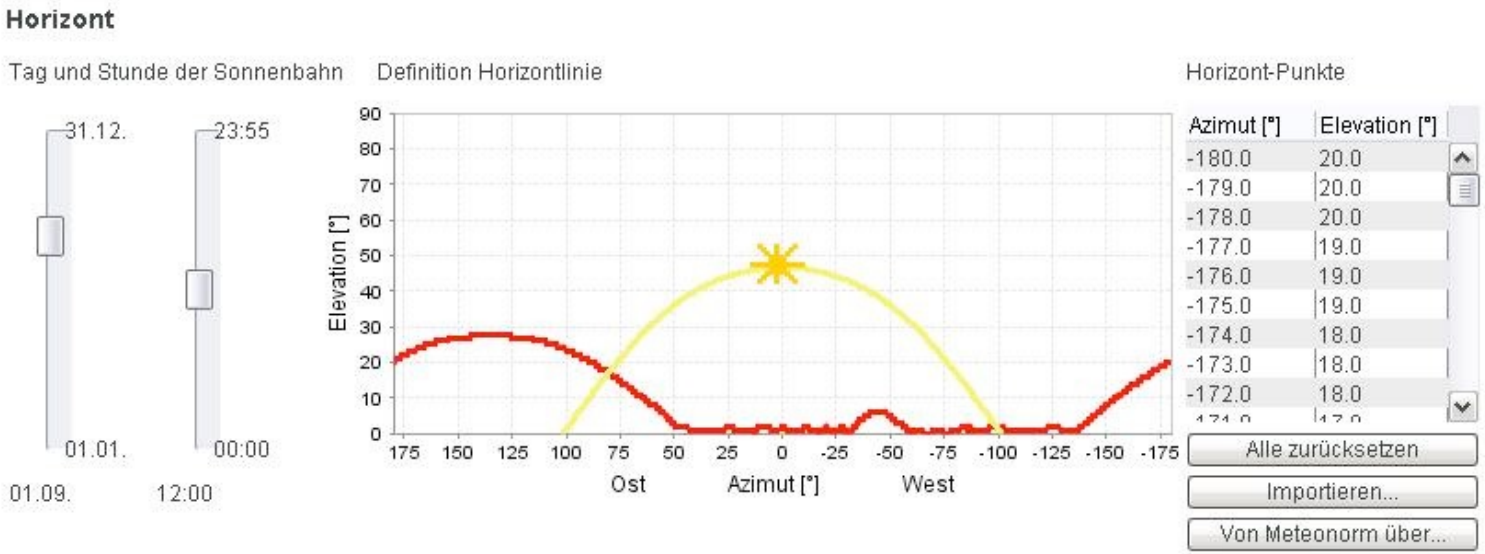


Combined geometry and terrain model for more accurate analysis

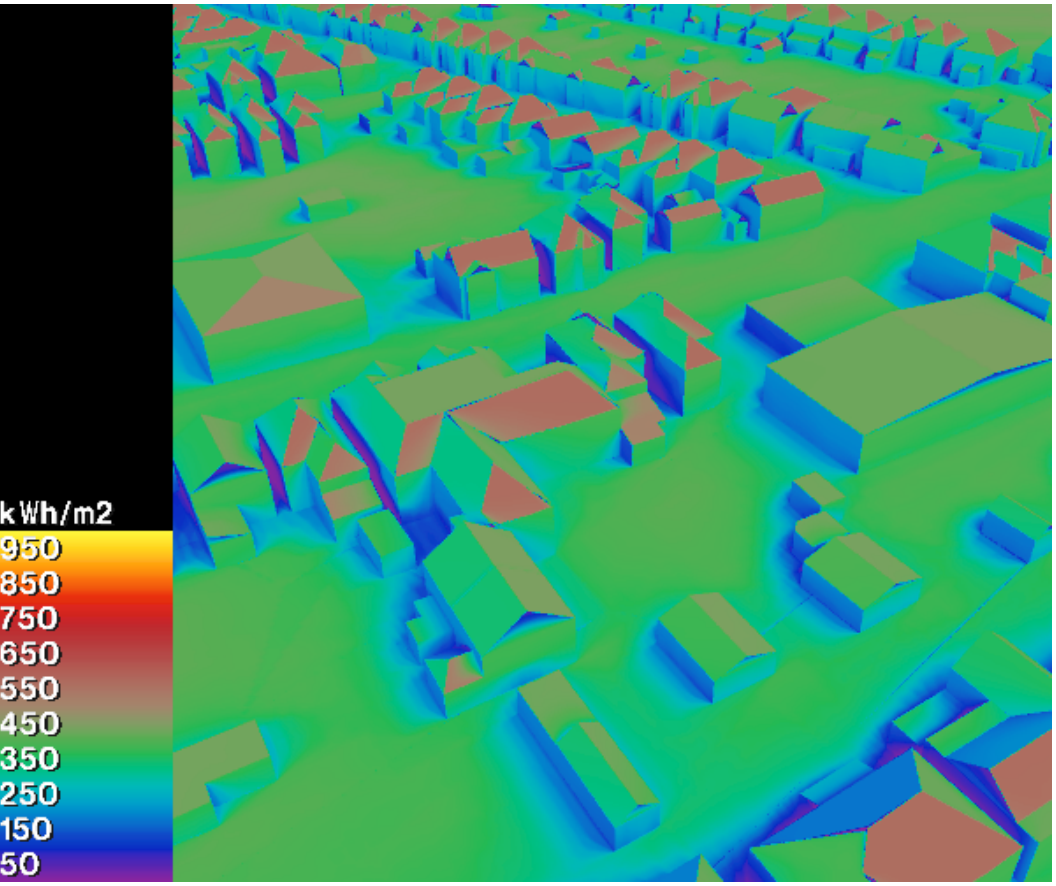


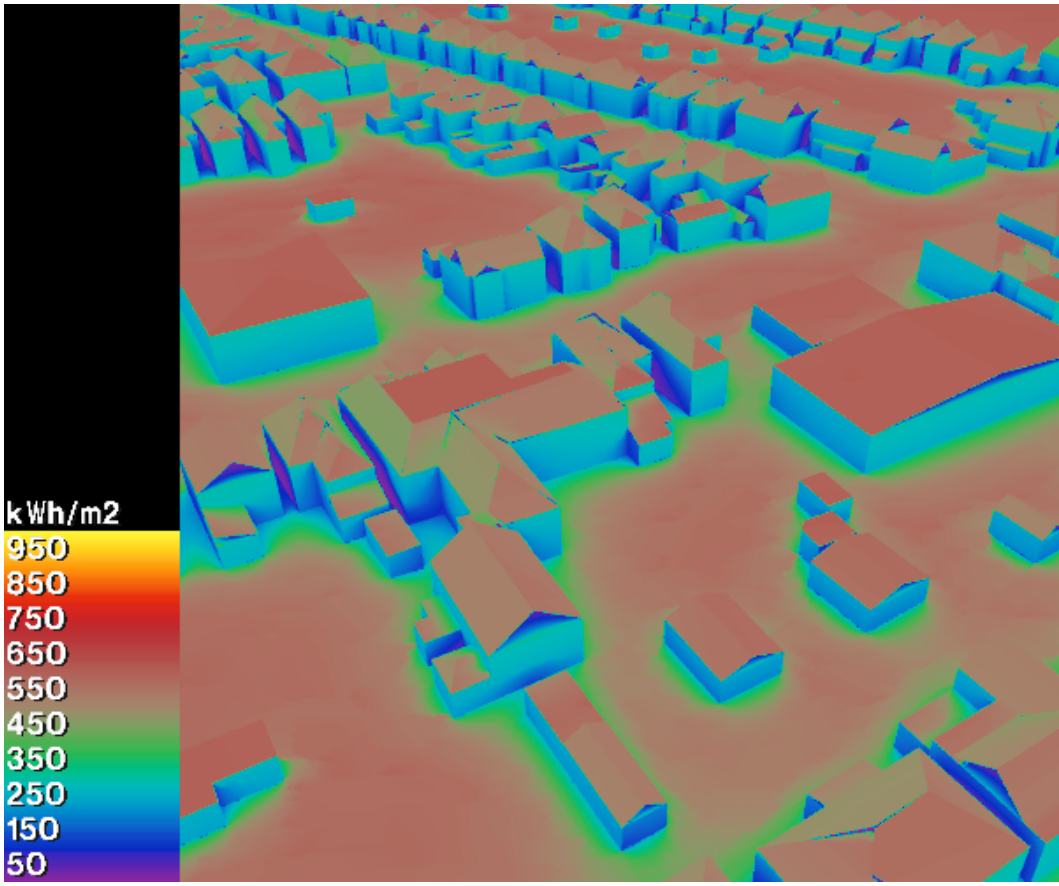
Local and background obstructions / horizon profiles



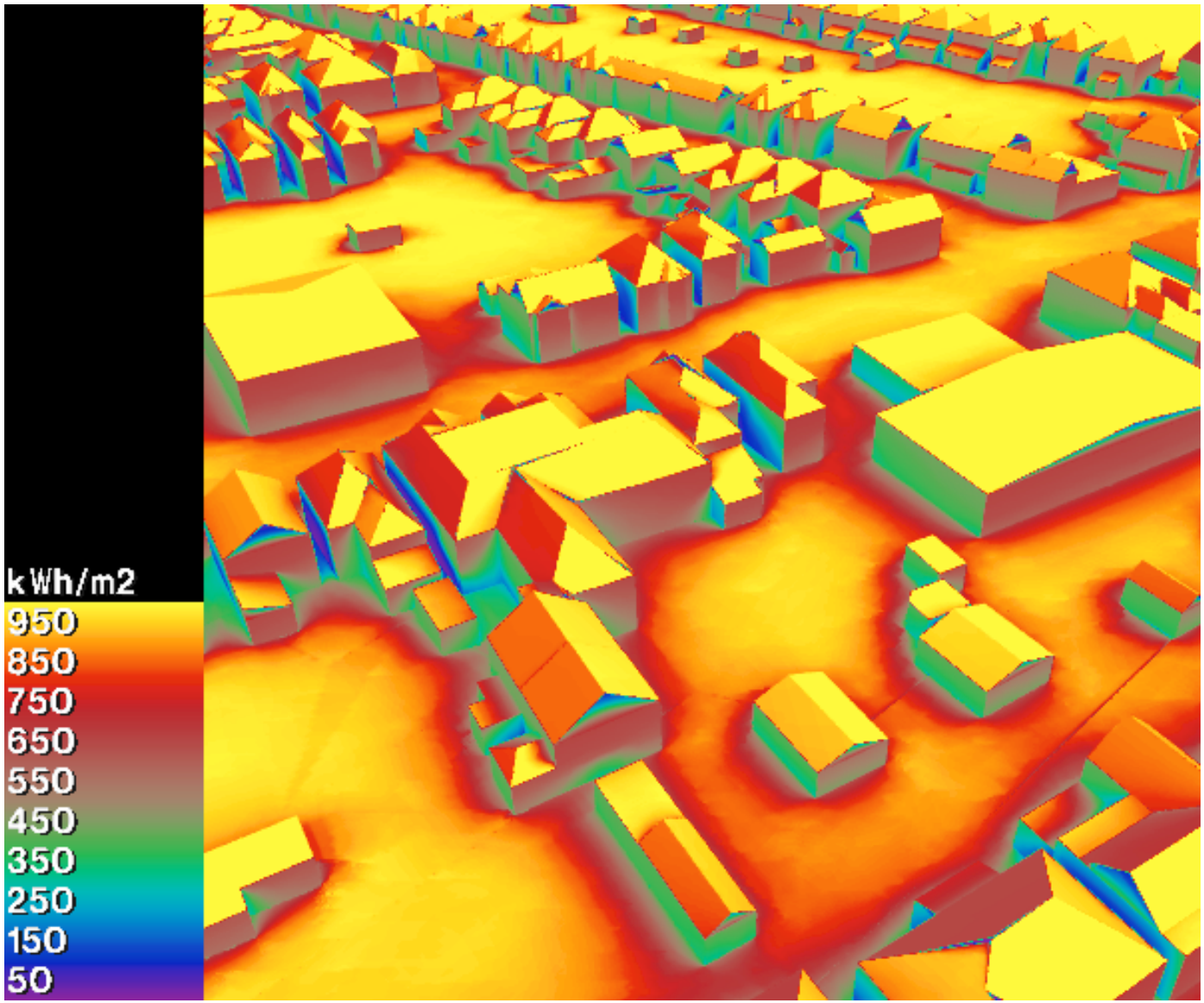


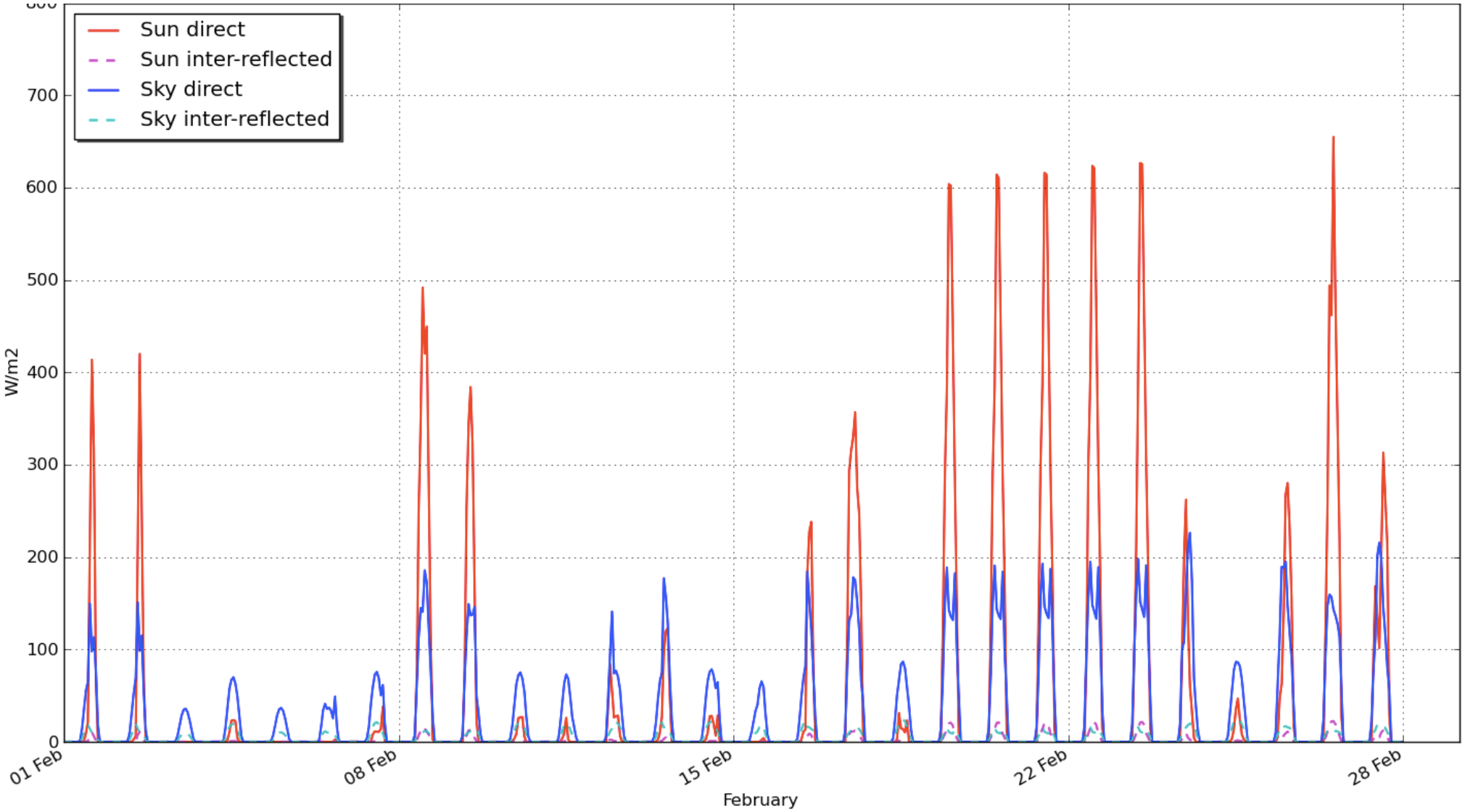
Irradiation components: direct and diffuse





Global annual irradiation



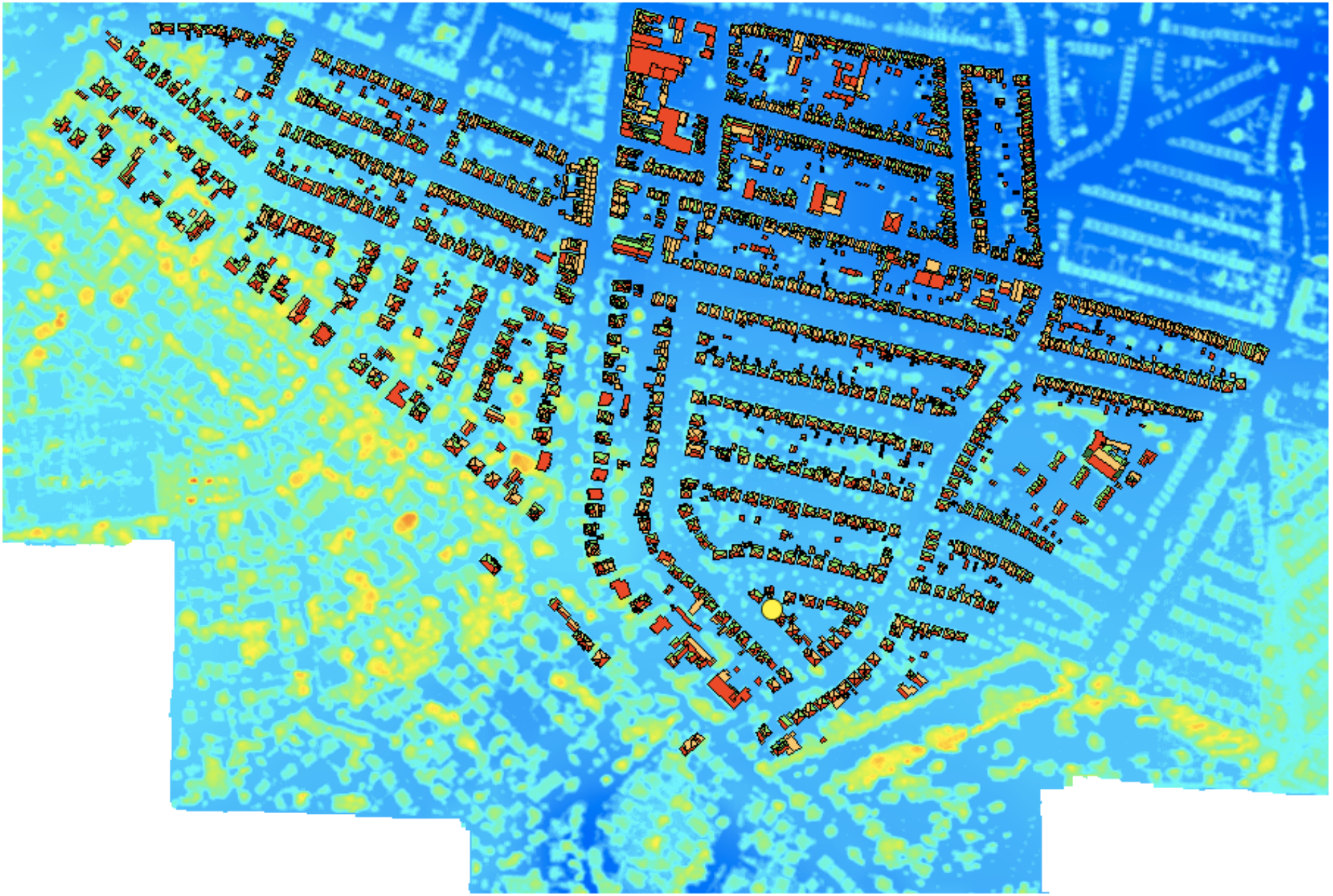


Instantaneous irradiance profiles

ARUP

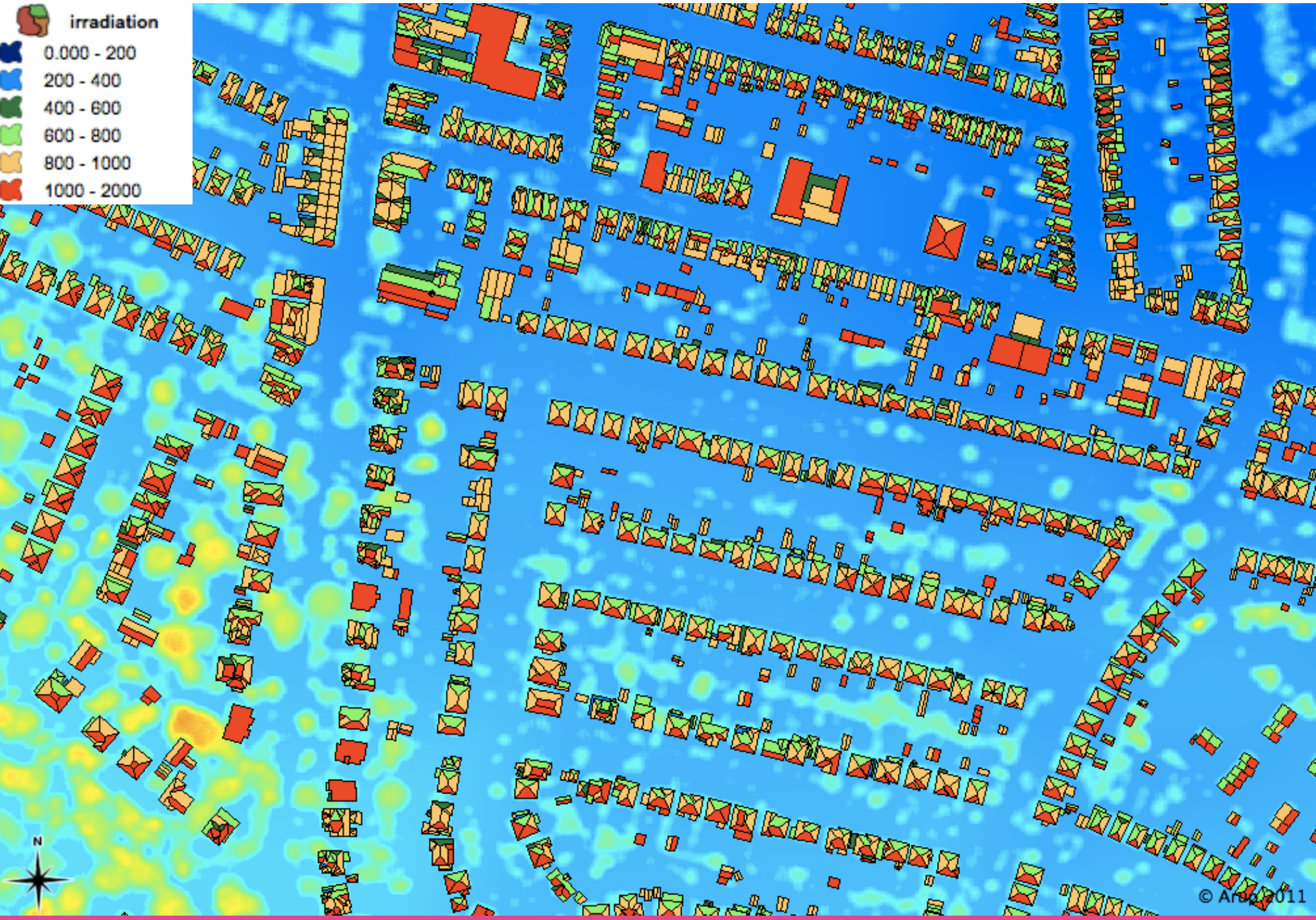
UK roofs

ARUP



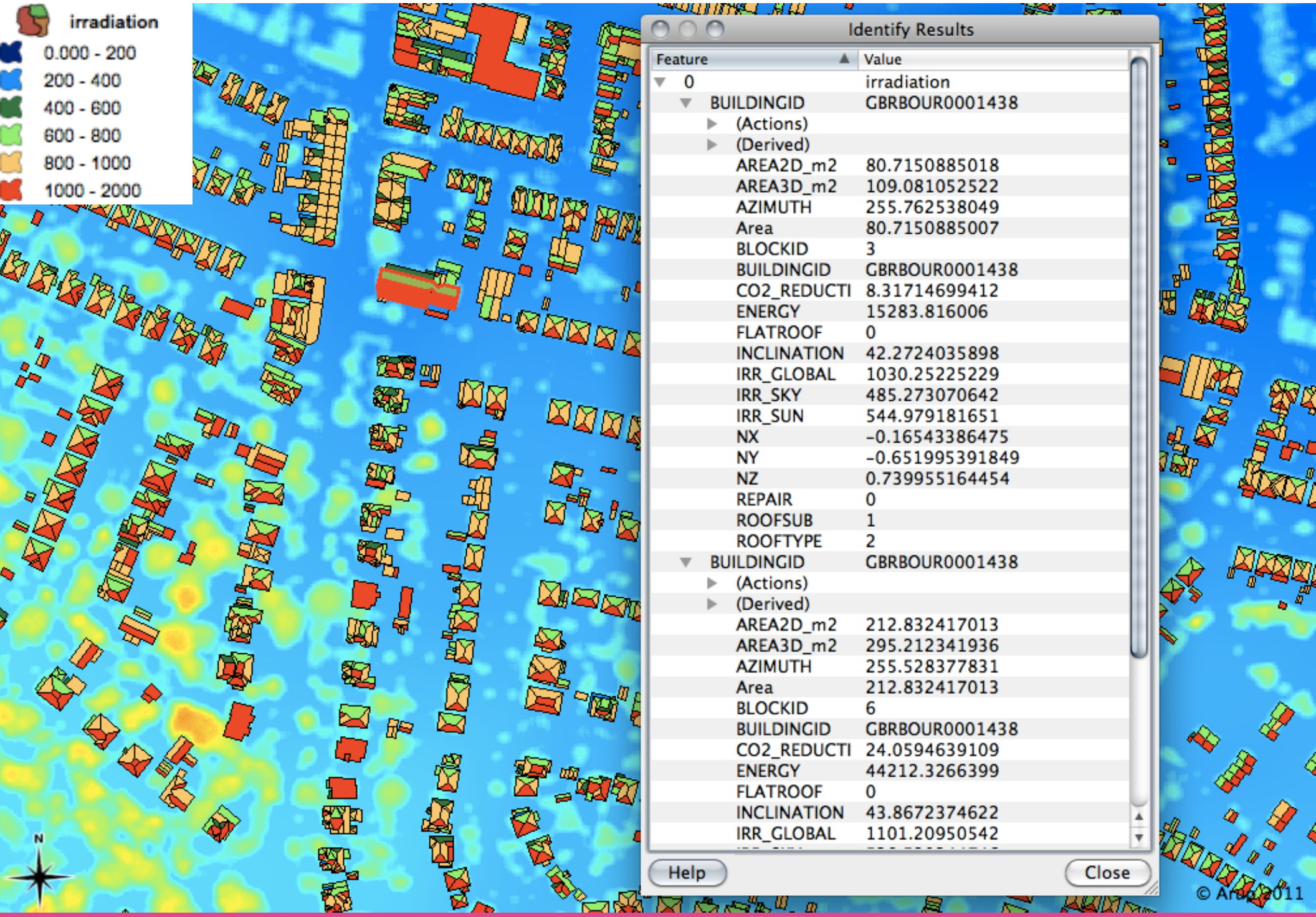
Irradiance data into GIS

ARUP



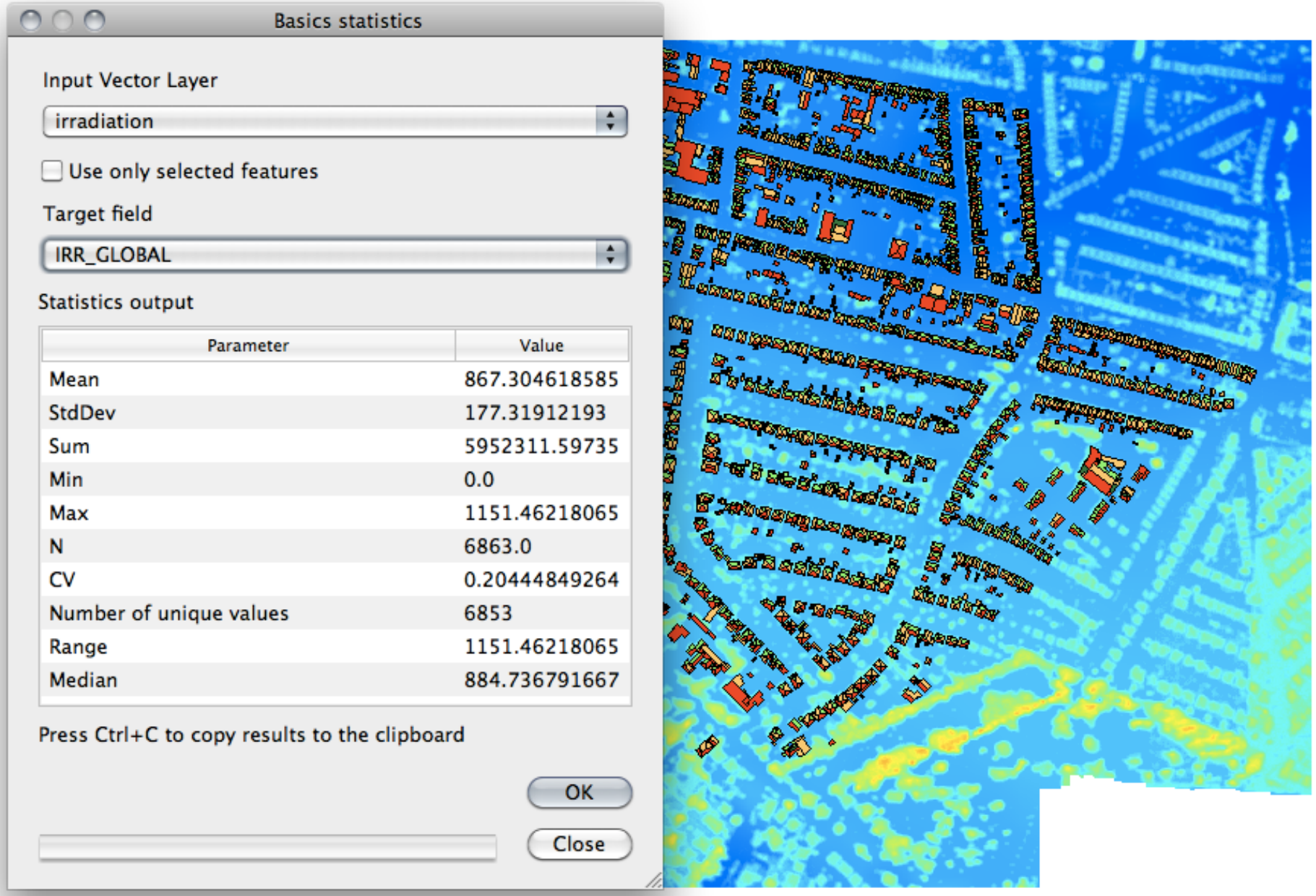
Irradiance data into GIS

ARUP



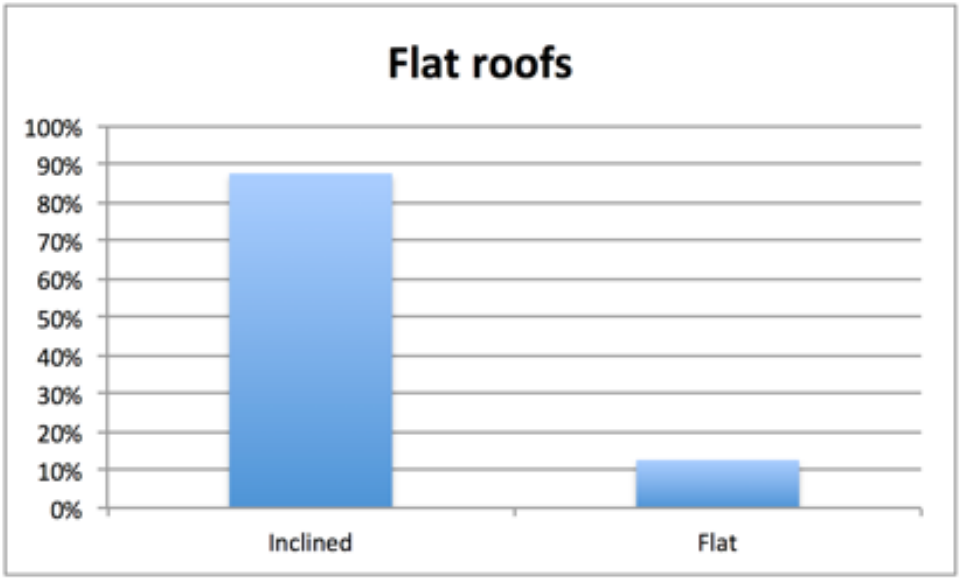
Interrogation of GIS model

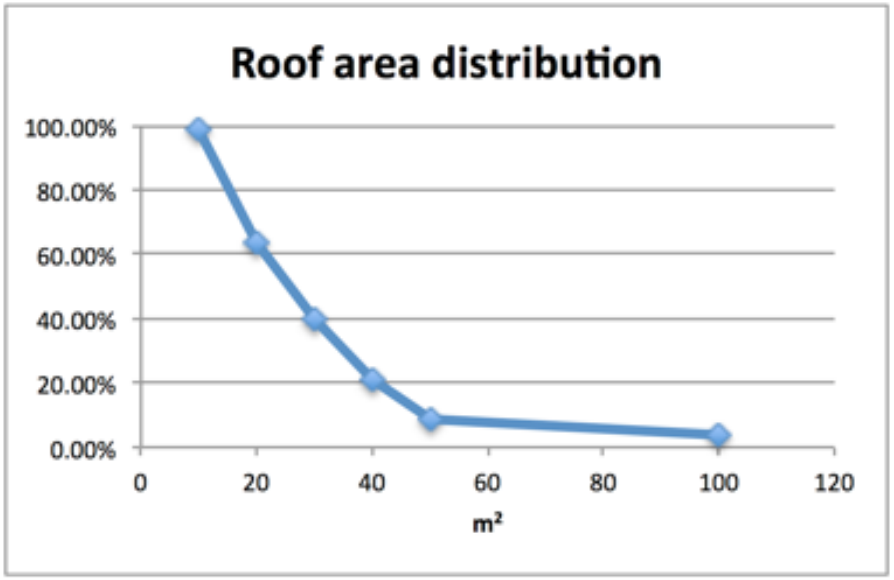
ARUP

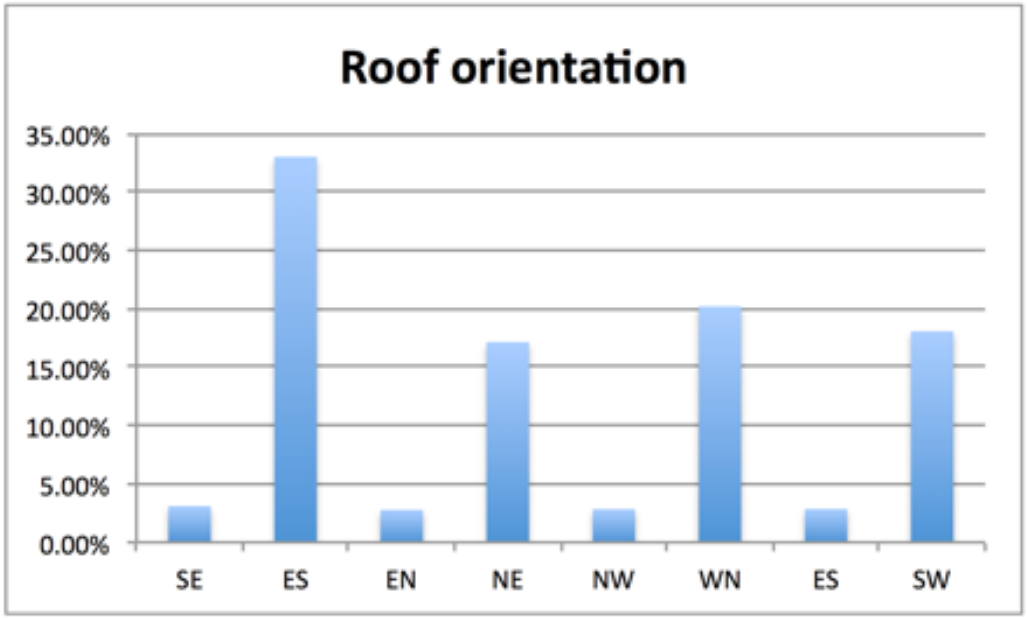


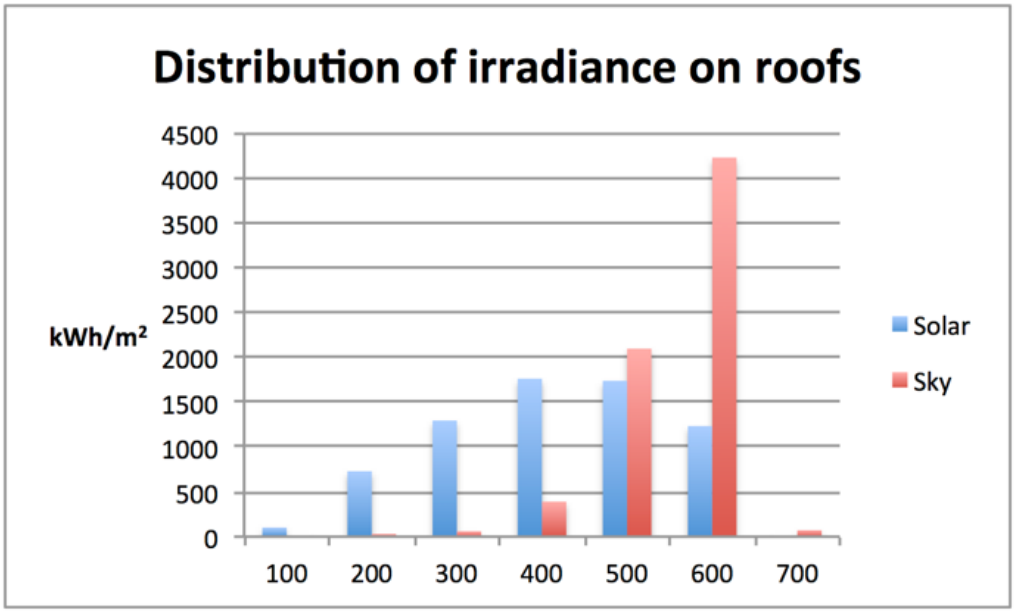
Statistics from GIS model

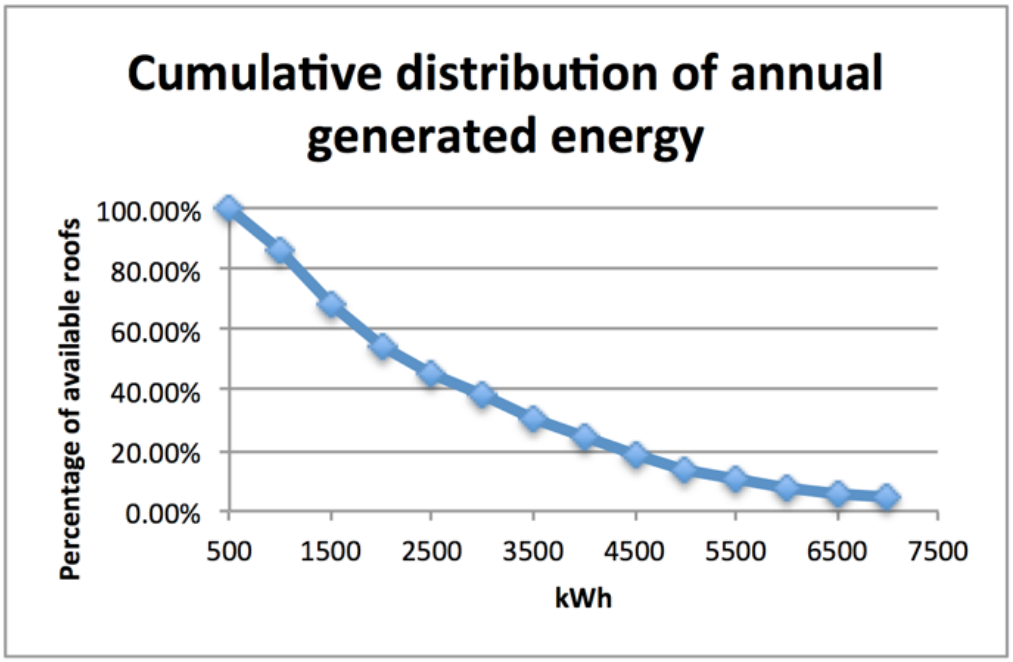
ARUP


Flat roofs
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Inclined
Flat


Roof area distribution
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
0
20
40
60
80
100
120
m²


Roof orientation
35.00%
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
SE
ES
EN
NE
NW
WN
ES
SW


Distribution of irradiance on roofs
kWh/m²
4500
4000
3500
3000
2500
2000
1500
1000
500
0
100
200
300
400
500
600
700
Solar
Sky


Cumulative distribution of annual generated energy
Percentage of available roofs
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
500
1500
2500
3500
4500
5500
6500
7500
kWh


Solar Radiation Mapping
0 - 200
200 - 400
400 - 600
600 - 800
800 - 1000
1000 - 1200

{

"type": "FeatureCollection",

"features": [

{ "type": "Feature", "properties": { "BUILDINGID": "GBRBOUR0001395", "BLOCKID": 1, "ROOFTYPE": 2, "ROOFSUB": 1, "REPAIR": 0, "Area": 7.103009 }, "geometry": { "type": "Polygon", "coordinates": [[[-208526.475416, 6575233.842579], [-208526.965916, 6575231.910008], [-208535.579035, 6575234.108296], [-208535.088539, 6575236.040709], [-208526.475416, 6575233.842579]]] } }

,

{ "type": "Feature", "properties": { "BUILDINGID": "GBRBOUR0001395", "BLOCKID": 2, "ROOFTYPE": 2, "ROOFSUB": 1, "REPAIR": 0, "Area": 2.748727 }, "geometry": { "type": "Polygon", "coordinates": [[[-208519.872014, 6575232.157297], [-208517.506644, 6575233.361556], [-208516.038868, 6575231.736187], [-208519.482283, 6575230.000100], [-208519.872014, 6575232.157297]]] } }

,

{ "type": "Feature", "properties": { "BUILDINGID": "GBRBOUR0001395", "BLOCKID": 2, "ROOFTYPE": 2, "ROOFSUB": 1, "REPAIR": 0, "Area": 5.808792 }, "geometry": { "type": "Polygon", "coordinates": [[[-208519.872014, 6575232.157297], [-208519.482283, 6575230.000100], [-208526.965916, 6575231.910008], [-208526.475416, 6575233.842579], [-208519.872014, 6575232.157297]]] } }

,

{ "type": "Feature", "properties": { "BUILDINGID": "GBRBOUR0001395", "BLOCKID": 2, "ROOFTYPE": 2, "ROOFSUB": 1, "REPAIR": 0, "Area": 3.901491 }, "geometry": { "type": "Polygon", "coordinates": [[[-208514.575594, 6575237.500646], [-208516.038868, 6575231.736187], [-208517.506644, 6575233.361556], [-208516.341564, 6575237.951251], [-208514.575594, 6575237.500646]]] } }

,

{ "type": "Feature", "properties": { "BUILDINGID": "GBRBOUR0001395", "BLOCKID": 3, "ROOFTYPE": 3, "ROOFSUB": 1, "REPAIR": 0, "Area": 24.153967 }, "geometry": { "type": "Polygon", "coordinates": [[[-208531.171831, 6575248.206274], [-208525.177928, 6575242.341525], [-208517.261731, 6575244.656177], [-208514.799085, 6575244.027724], [-208523.197967, 6575239.588700], [-208533.519228, 6575242.222763], [-208531.949915, 6575248.404821], [-208531.171831, 6575248.206274]]] } }

,

{ "type": "Feature", "properties": { "BUILDINGID": "GBRBOUR0001395", "BLOCKID": 3, "ROOFTYPE": 3, "ROOFSUB": 1, "REPAIR": 0, "Area": 15.543203 }, "geometry": { "type": "Polygon", "coordinates": [[[-208523.197967, 6575239.588700], [-208514.799085, 6575244.027724], [-208516.341564, 6575237.951251], [-208517.506644, 6575233.361556], [-208523.197967, 6575239.588700]]] } }

,

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>OGR2Layers</title>
<style>
 #map{width:400px;height:400px;}
</style>
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
var map, selectsControls
function init(){
        var option = {
                projection: new OpenLayers.Projection("EPSG:900913"),
                displayProjection: new OpenLayers.Projection("EPSG:4326")
        };
        map = new OpenLayers.Map('map', option);
        olmapnik = new OpenLayers.Layer.OSM("OpenStreetMap Mapnik", "http://tile.openstreetmap.org/${z}/${x}/${y}.png");
        map.addLayer(olmapnik);
        map.setBaseLayer(olmapnik);
        var ls= new OpenLayers.Control.LayerSwitcher();
        map.addControl(ls);
        ls.maximizeControl();
        map.addControl(new OpenLayers.Control.MousePosition());
        map.addControl(new OpenLayers.Control.Scale());
        map.addControl(new OpenLayers.Control.Permalink());
        map.addControl(new OpenLayers.Control.Attribution());
        map.addControl(new OpenLayers.Control.OverviewMap());
        map.addControl(new OpenLayers.Control.PanZoomBar());
        var roofs_template = {
                strokeColor: "#000000",
                strokeOpacity: 1,
                strokeWidth: 0.26,
                fillColor: "#0122c1",
                fillOpacity: 1
        }
```

```
var roofs_style = new OpenLayers.Style(roofs_template)
	//START QUERY roofs
	function onPopupCloseroofs(evt) {
		selectControl.unselect(selectedFeature);
	}
	function onFeatureSelectroofs(feature){
		selectedFeature = feature;
		tableroofs="<html><meta http-equiv='Content-Type' content='text/html; charset=UTF-8'><body><table><tr><td><b>BUILDINGID:</b></
td><td><i>"+feature.attributes.BUILDINGID+"</i></td></tr><tr><td><b>BLOCKID:</b></td><td><i>"+feature.attributes.BLOCKID+"</i></td></
tr><tr><td><b>ROOFTYPE:</b></td><td><i>"+feature.attributes.ROOFTYPE+"</i></td></tr><tr><td><b>ROOFSUB:</b></
td><td><i>"+feature.attributes.ROOFSUB+"</i></td></tr><tr><td><b>REPAIR:</b></td><td><i>"+feature.attributes.REPAIR+"</i></td></tr><tr><td><b>Area:</
b></td><td><i>"+feature.attributes.Area+"</i></td></tr></table></body></html>";
		popup = new OpenLayers.Popup.FramedCloud("popup",
			feature.geometry.getBounds().getCenterLonLat(),
			new OpenLayers.Size(1000,500),
			tableroofs,
			null,
			true,
			onPopupCloseroofs
		);
		feature.popup = popup;
		map.addPopup(popup);
	}
	function onFeatureUnselectroofs(feature) {
		map.removePopup(feature.popup);
		feature.popup.destroy();
		feature.popup = null;
	}
```

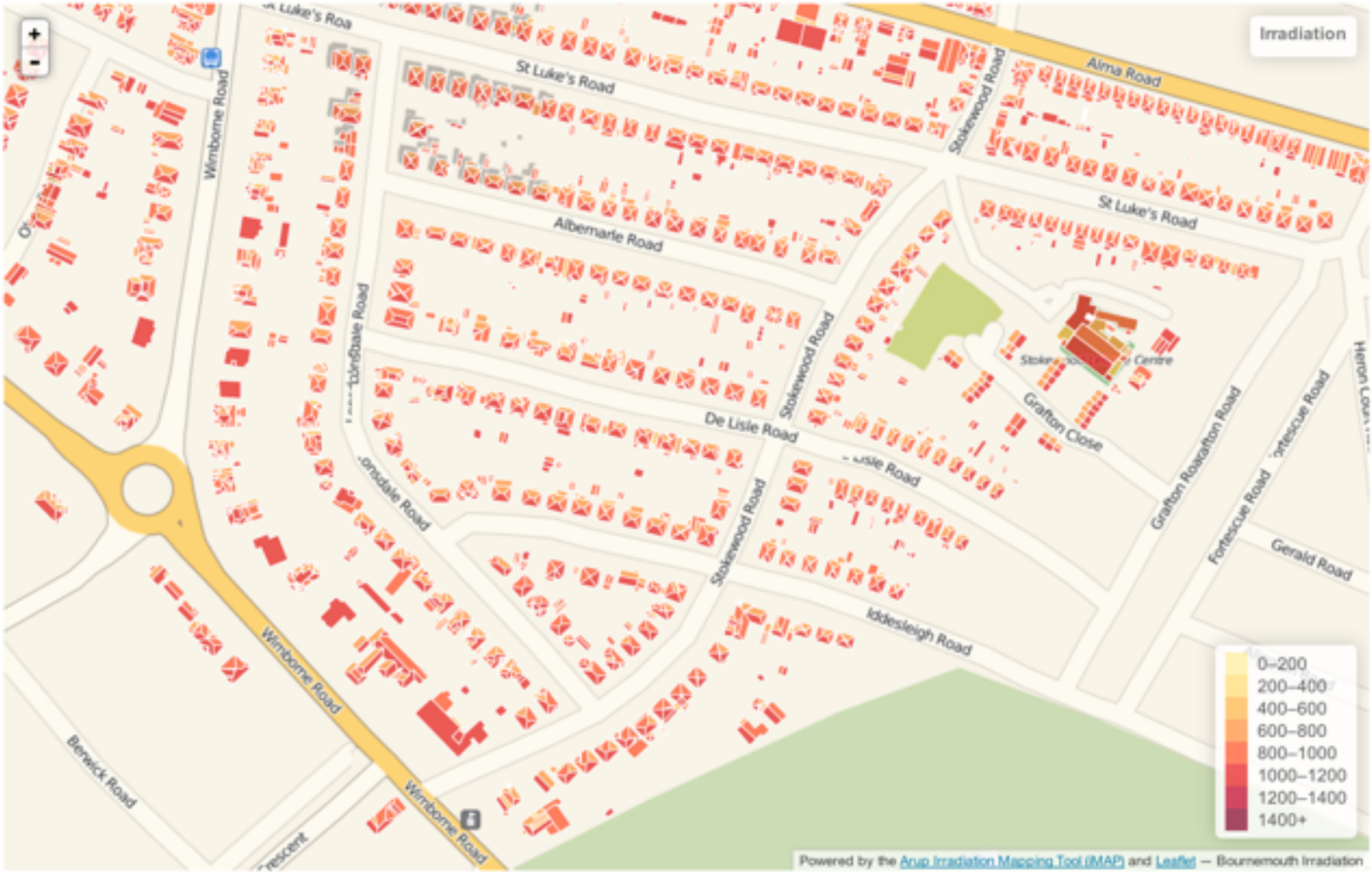
```
var roofs = new OpenLayers.Layer.GML("roofs GeoJSON", "roofs.GeoJSON", {format: OpenLayers.Format.GeoJSON, styleMap: roofs_style});
        map.addLayer(roofs);
        selectControl = new OpenLayers.Control.SelectFeature(
                [roofs, ],
                {
                        clickout: true, toggle: false,
                        multiple: false, hover: false,
                        toggleKey: "ctrlKey", // ctrl key removes from selection
                        multipleKey: "shiftKey" // shift key adds to selection
                }
        );

map.addControl(selectControl);
        selectControl.activate();
        roofs.events.on({
                "featureselected": function(e) {
                        onFeatureSelectroofs(e.feature);
                },
                "featureunselected": function(e) {
                        onFeatureUnselectroofs(e.feature);
                }
        });
        extent = new OpenLayers.Bounds(-1.8882,50.732696,-1.869302,50.741725).transform(new OpenLayers.Projection("EPSG:4326"), new
OpenLayers.Projection("EPSG:900913"));
        map.zoomToExtent(extent);
};
</script>
</head>
<body onload="init()">
<h1>Bournemouth Solar Potential Mapping</h1>
<div id="map"></div>
</body>
</html>
```

Bournemouth Interactive Irradiation Map

lightlab.arup.com/imap/

Bonjour | The Perfect ...nd Tutorials | Affinity | Queens | Pinpoint | Zero-to-App | LatLong | RB | GE | CC | Keep It! | CIE | GE | Bozzocloud | Bozzograo | Water

Arup Irradiation Mapping Test - Bournemouth

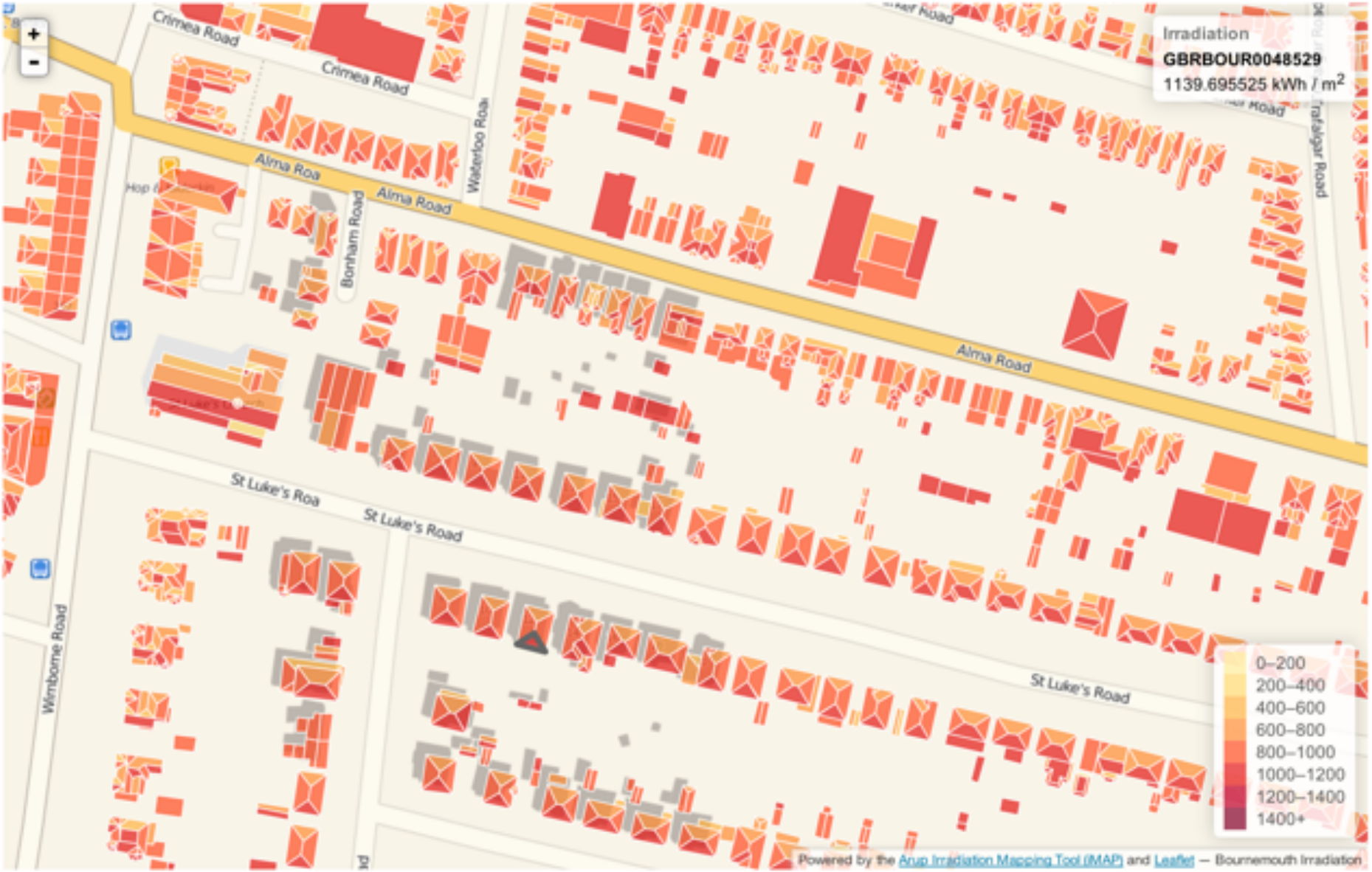


Bournemouth Interactive Irradiation Map

lightlab.arup.com/imap/

Bonjour | The Perfect ...nd Tutorials | Affinity | Queens | Pinpoint | Zero-to-App | LatLong | RB | GE | CC | Keep It! | CIE | GE | Bozzocloud | Bozzograo | Water

Arup Irradiation Mapping Test - Bournemouth



Thank you!

Francesco Anselmo

francesco.anselmo@arup.com

ARUP