# Rooftop Photovoltaic

## An Algorithmic Solution for Obtaining Total Potential Power Generation by Processing Solar Irradiance Data
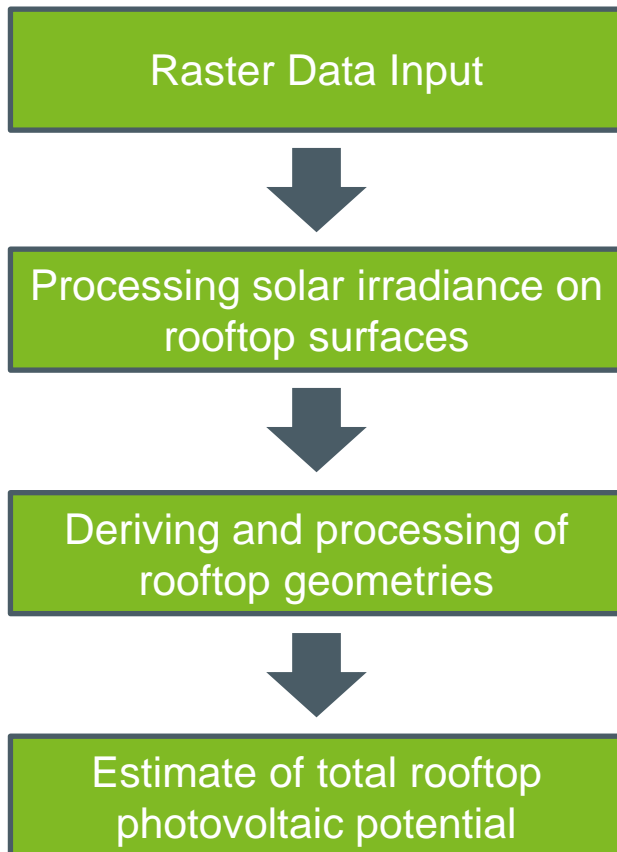
M. Sc. Hannes Koch, Prof. Dr.-Ing. Stefan Lechner, M. Sc. Sebastian Erdmann
hannes.koch@me.thm.de
THM – University of Applied Sciences
Gießen, Germany

## *Overview*

- Goal of the Approach and Novel Aspects

- Data Sources

- Methodology

- Programmatic Approach

- Results

# *Goal of the Approach and Novel Aspects*

| Raster Data Input |

↓

| Processing solar irradiance on rooftop surfaces |

↓

| Deriving and processing of rooftop geometries |

↓

| Estimate of total rooftop photovoltaic potential |

1. Any amount of high resolution solar irradiance raster data can be processed fast and efficiently.

2. Geometrical traits of rooftop surfaces can be evaluated solely based on raster data.

3. The methods eligible to evaluate surfaces are interchangeable.

4. The algorithm's efficiency is barely dependant on the method applied to investigate the raster cells.
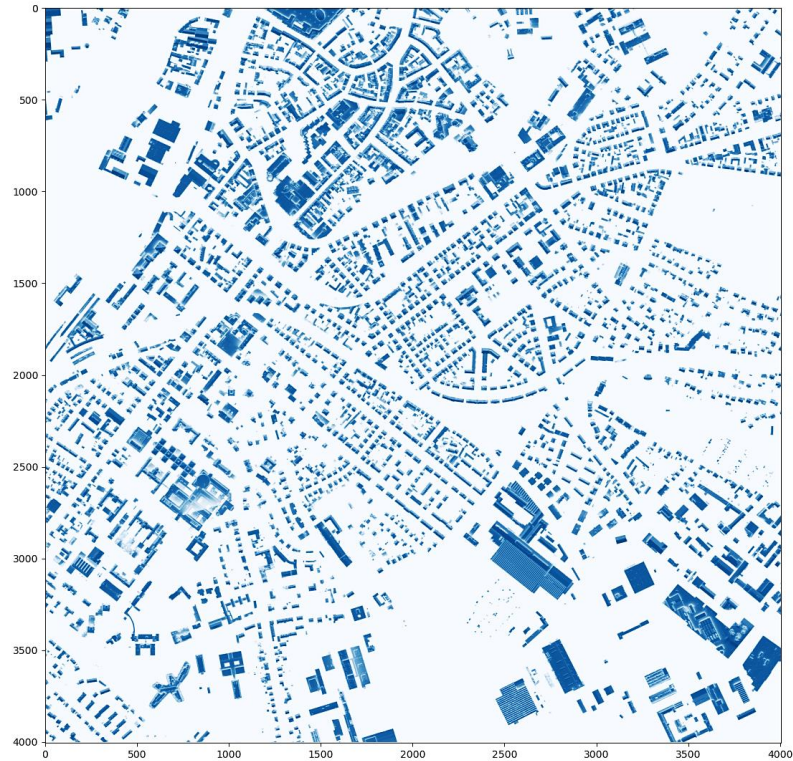
# *Data Sets (1)*

The algorithm requires:

1. A high resolution data set containing information on yearly solar irradiance, e.g. in $\left[\frac{kWh}{m^2 \cdot a}\right]$
2. A polygon layer containing the building ground areas to be examined

Preprocessing: Clip raster data to building ground areas

# *Data Sets (2)*

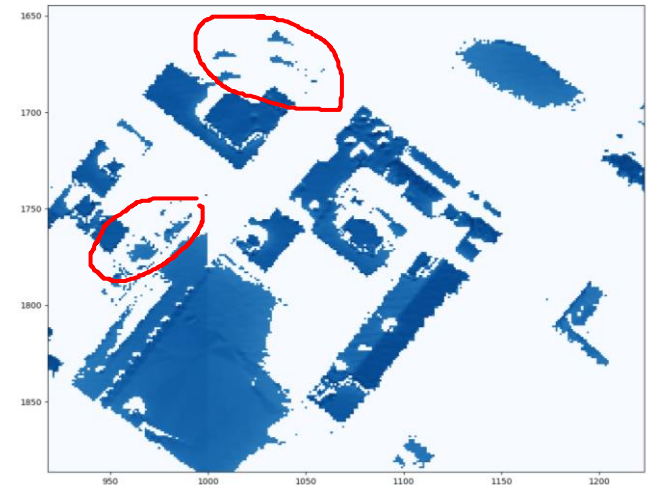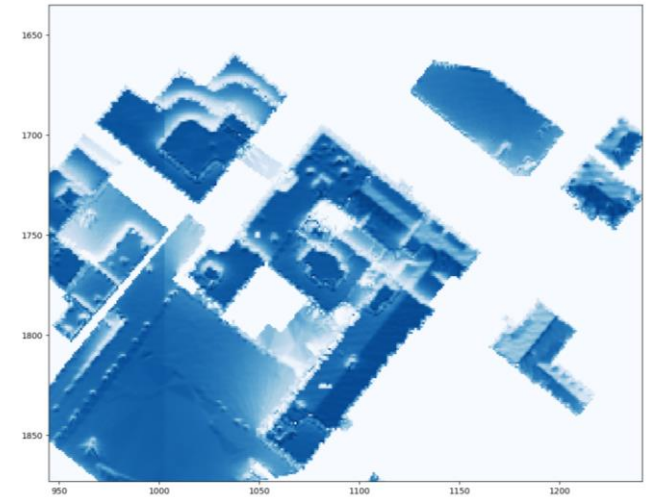Polygon layer and raster layer (grey scaled)     Raster layer clipped to polygon layer



[QGIS 2020, Sol 2019]

## *Methodology (1)*

1. Set a threshhold value for solar irradiance, e.g. 800 $\left[\frac{kWh}{m^2 \cdot a}\right]$

2. Set all raster values below given threshhold to zero

„Fragments" remain, representing areas uncapable of hosting PV modules!
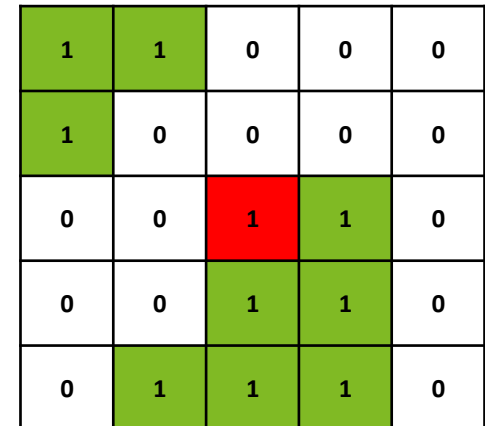
# *Methodology (2)*

- Extract buildable surfaces by investigating each raster cell's neighbourhood

- Proposed method (others are possible) – Step One:

  1. Choose a number of cells in the expanded MOORE neighbourhood that have to be non-zero

  2. Set the investigated cell's value to zero if the number of non-zero cells is below the chosen value

| | | | | |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | 🟥 | | |
| | | | | |
| | | | | |
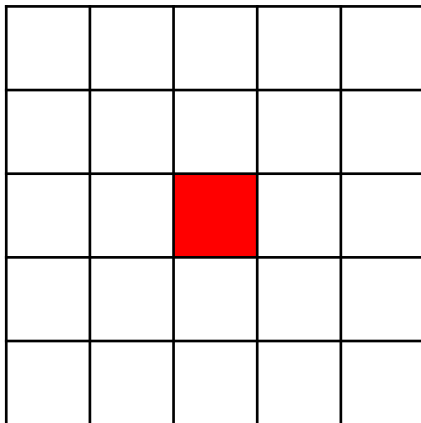
Investigate Expanded MOORE neighbourhood [Guan, 2009]

| | | | | |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | 🟥 | 🟩 | 🟩 |
| | | | 🟩 | 🟩 |
| | | | 🟩 | 🟩 |
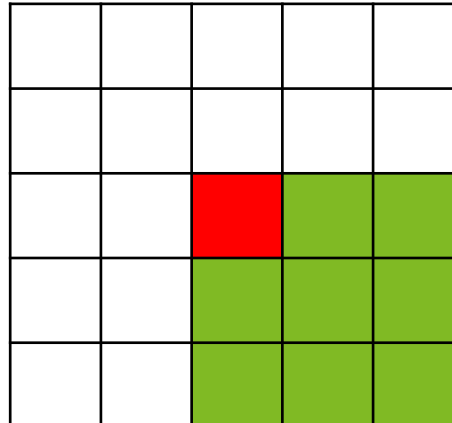
Setting the criterion to allow corner points of rooftops

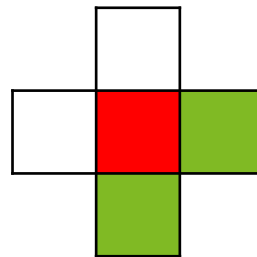| | | | | |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |

Check for criterion using a Boolean Mask of raster data set

# *Methodology (3)*

- Proposed method (others are possible) – Step Two:

  3. Choose a number of adjoining cells that have to be non-zero

  4. Repeat the investigation done before, now regarding the VON-NEUMANN-neighbourhood
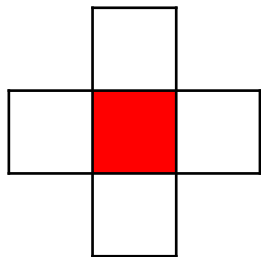


Investigating Expanded VON-NEUMANN neighbourhood [Guan, 2009]

Setting the criterion to at least two adjoining cells

Preventing non adjoining cell structures from being classified as buildable

# *Programmatic Approach*

- Large amount of raster cells requires vectorization, minimizing memory latency

- Operations carried out across two coordinate axes require a third axis to be vectorized [v. d. Walt, 2011]

- Appending neighbours' values to the third axis can be done simultaneously

- Reduces the number of calculation steps to the number of cells in the expanded MOORE neighbourhood

| | | |
|---|---|---|
| (i-1, j-1) | (i-1, j) | (i-1, j+1) |
| (i, j-1) | (i, j) | (i, j+1) |
| (i+1, j-1) | (i+1, j) | (i+1, j+1) |

(i+1, j+1)

(...)

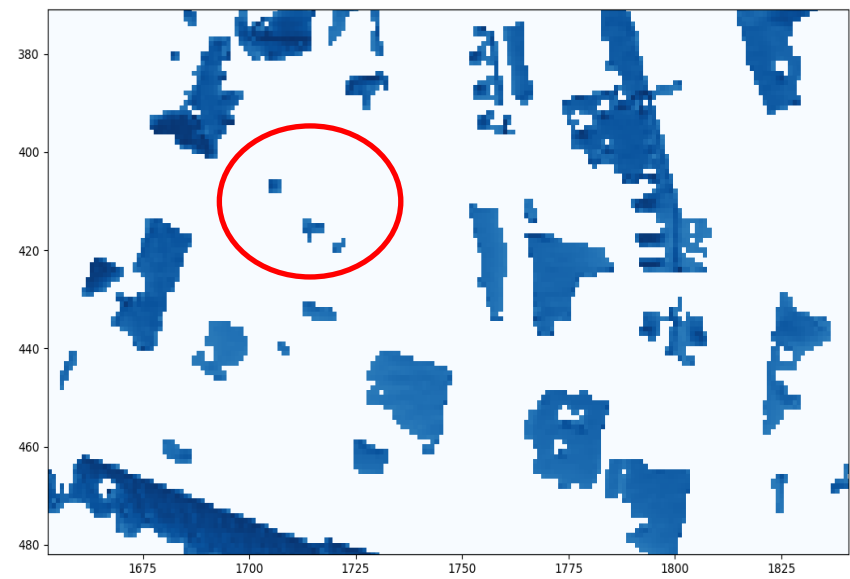(i-1, j)

(i-1, j-1)
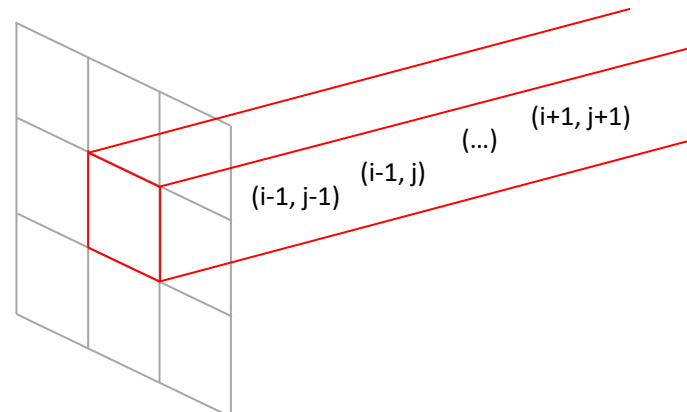
## *Results (1)*

Raster layer before neighbour investigation
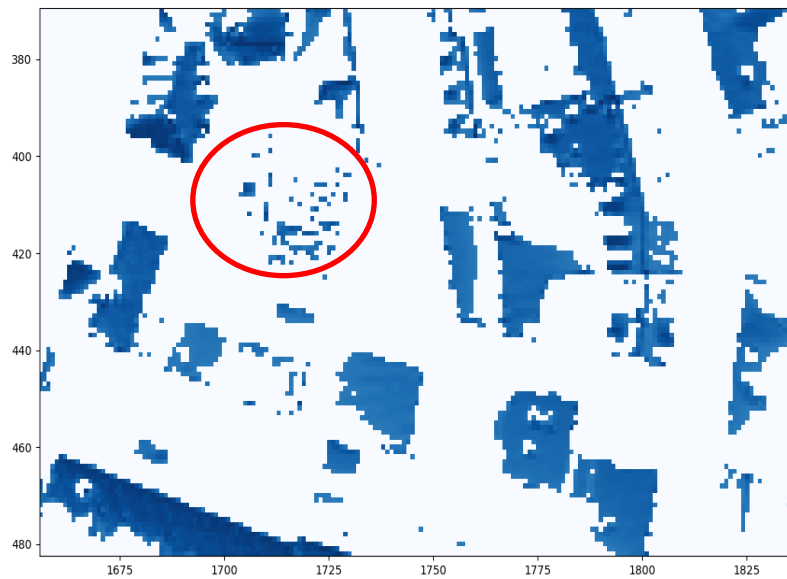
Raster layer after neighbour investigation

# *Results (2)*



[Gmaps, 2020]

- The algorithm processes a 4000x4000 raster file in approx. 0.7 seconds



```
Fri Sep 11 15:52:22 2020      raster/stats_vectorized

        497 function calls (469 primitive calls) in 0.680 seconds
```

- For comparison, a procedural solution with two for-loops takes approx. 5.8 seconds



```
Fri Sep 11 15:52:28 2020      raster/stats_DoubleForLoop

        458 function calls (430 primitive calls) in 5.776 seconds
```

# Thank you for your attention!

Please refer any questions to:
hannes.koch@me.thm.de

M. Sc. Hannes Koch, Prof. Dr.-Ing. Stefan Lechner, M. Sc. Sebastian Erdmann
hannes.koch@me.thm.de
THM – University of Applied Sciences
Gießen, Germany

# *Literature*

[Sol, 2019]   Provided by the Hessian Ministry of Economics, Energy, Transport and Housing, publicly available via "Solarkataster Hessen". https://www.energieland.hessen.de/solar-kataster, [Accessed: 23.07.2019]

[QGIS, 2020]   QGIS Development Team (2020). QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org

[Gmaps, 2020]   Google Maps, *Satellite imagery*, accessed via QGIS application in May 2020.

[v. d. Walt, 2011]   Stéfan van der Walt et. al. (2011): The NumPy Array: A Structure for Efficient Numerical Computation, p. 3, Stellenbosch University South Africa, DOI: 10.1109/MCSE.2011.37

[Guan, 2009]   Qinfeng Guan, "pRPL: an open-source general-purpose parallel Raster Processing programming Library", p. 58, University of California, Santa Barbara, DOI: 10.1145/1517463.1517471