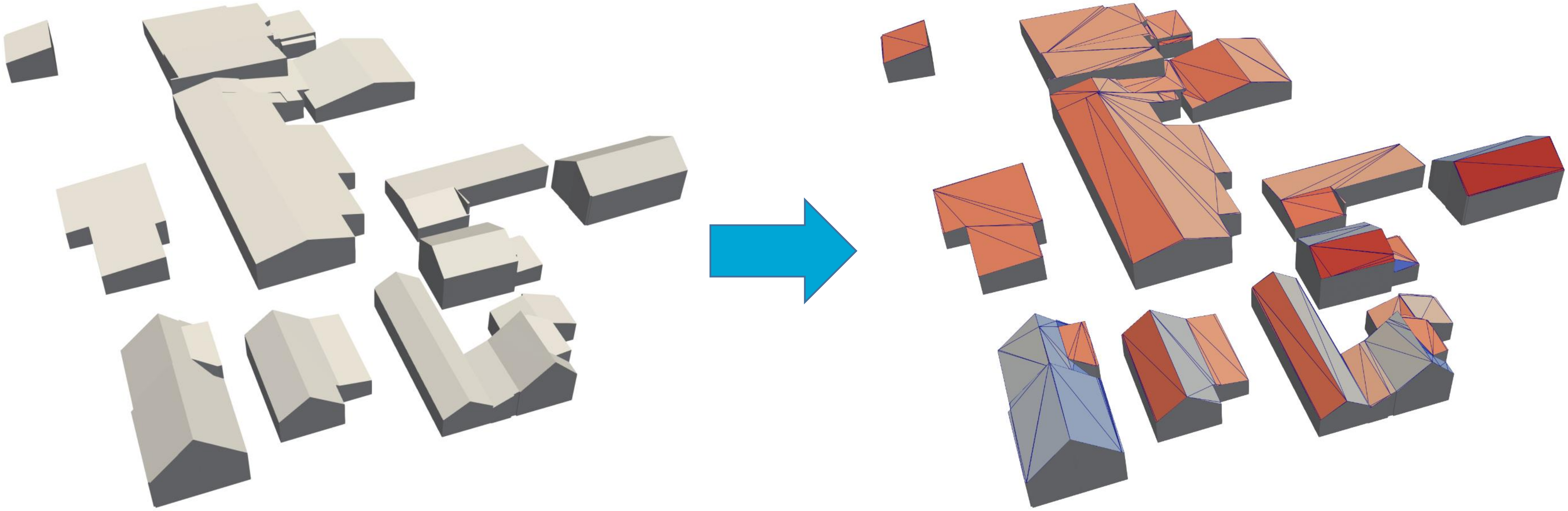
Efficient Solar Potential Estimation of 3D Buildings: 3D BAG as use case



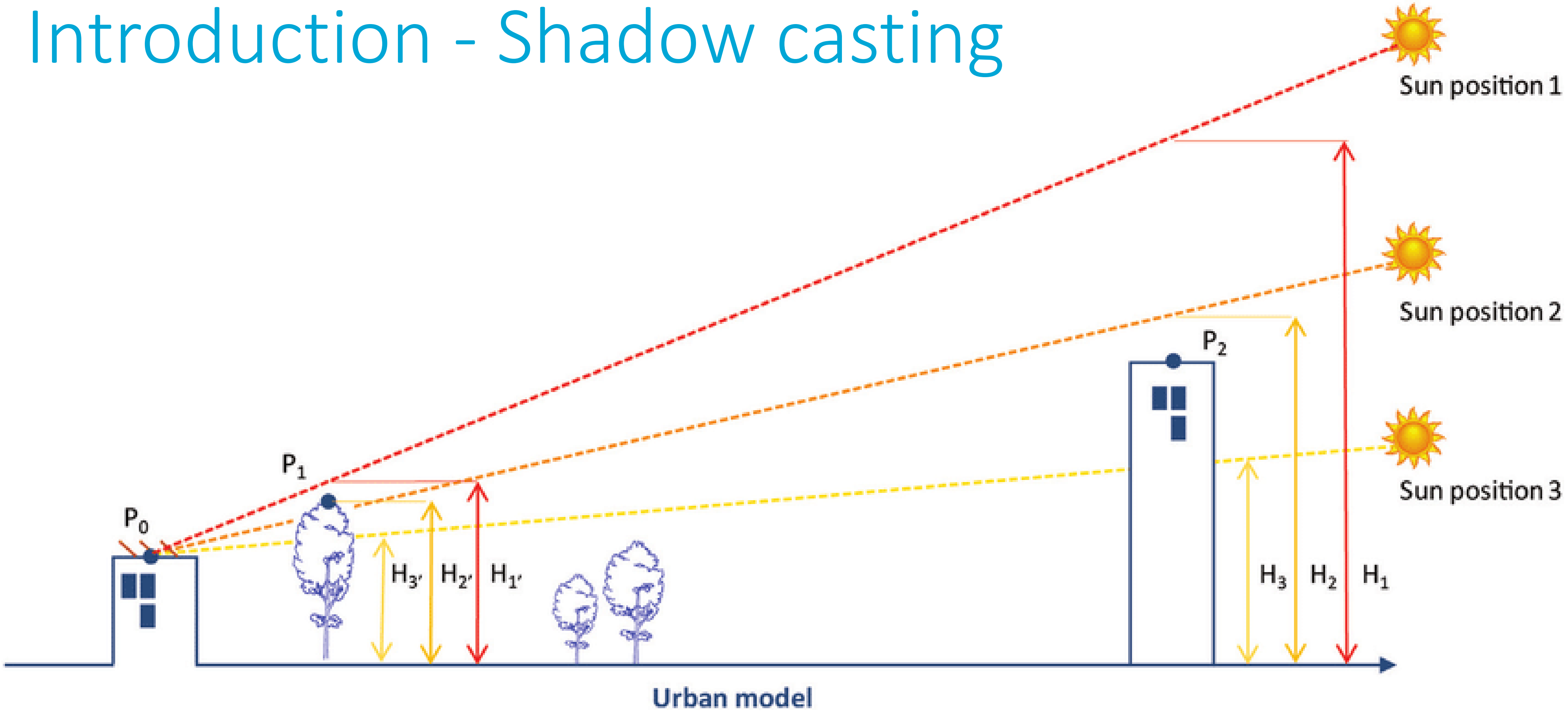
Robin Hurkmans | 11-11-2022

TU Delft

Contents

Introduction

Introduction - Shadow casting

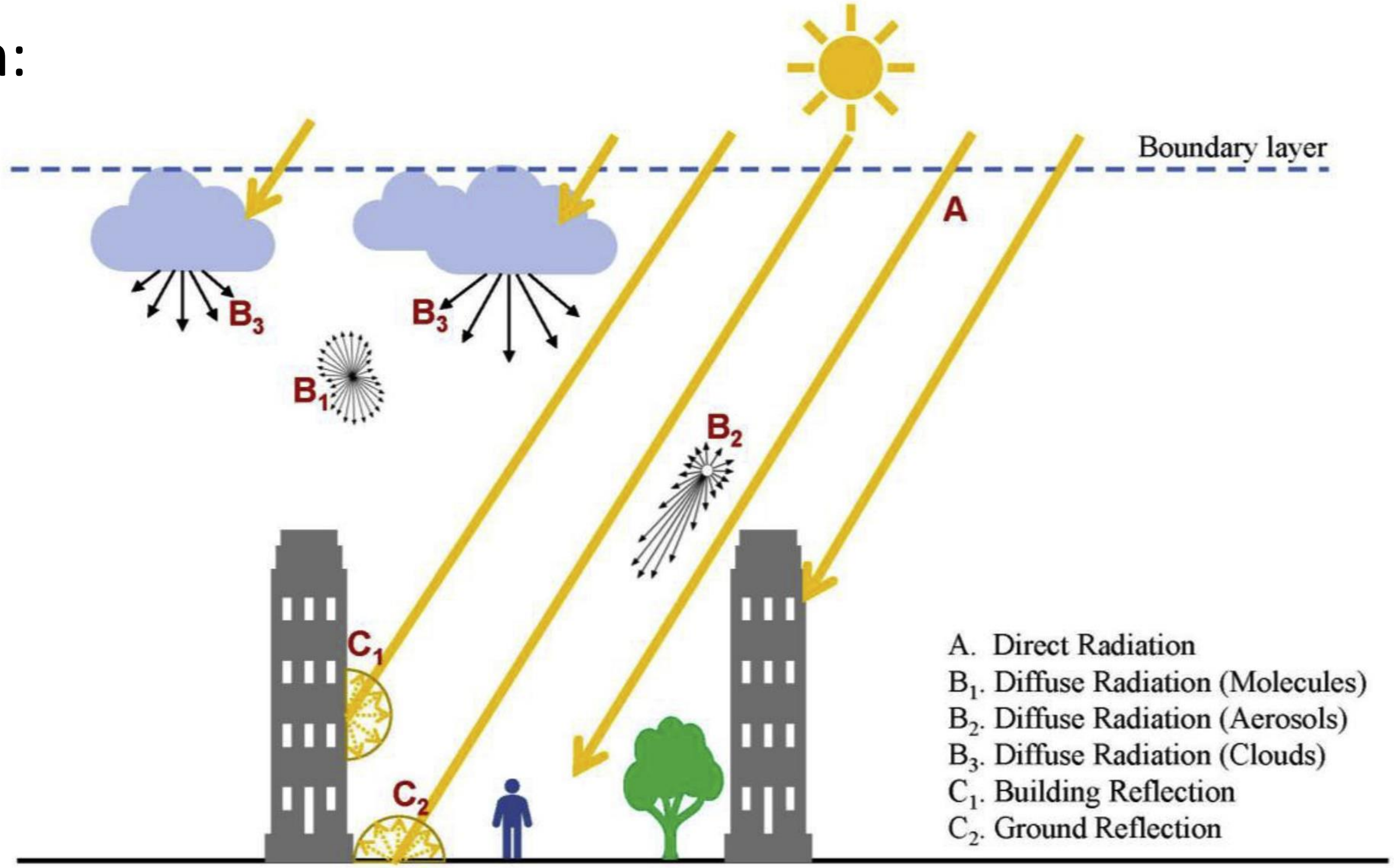


TU Delft

Introduction – Solar radiation

Types of solar radiation:

- Direct (beam)
- Diffuse
- Reflected
- Global



Introduction – Solar radiation models

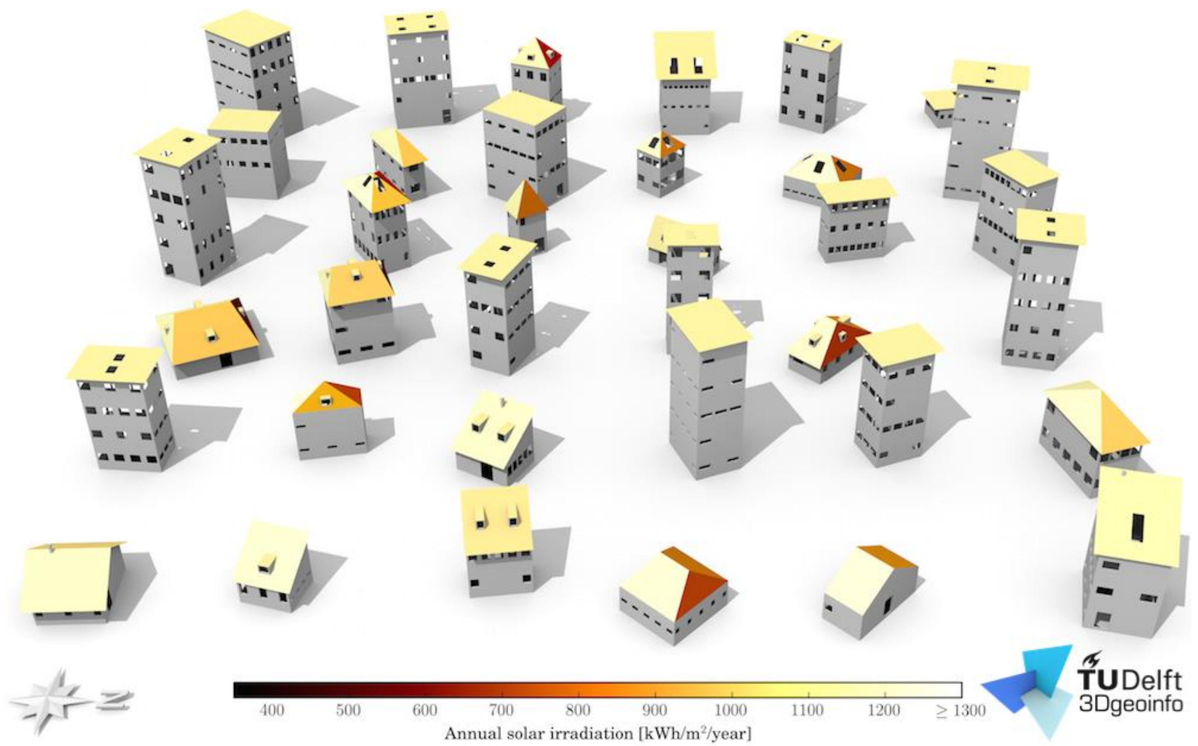
- Raster data
  - Grid of pixels in 2D
  - DEM
  - Viewshed
  - No accurate representation of 3D environment

ArcGIS

TU Delft

Introduction – Solar radiation models

- Vector data
  - Geometric primitives (vertices, lines, polygons)
  - Efficient storage in memory
  - Shadow casting is slow



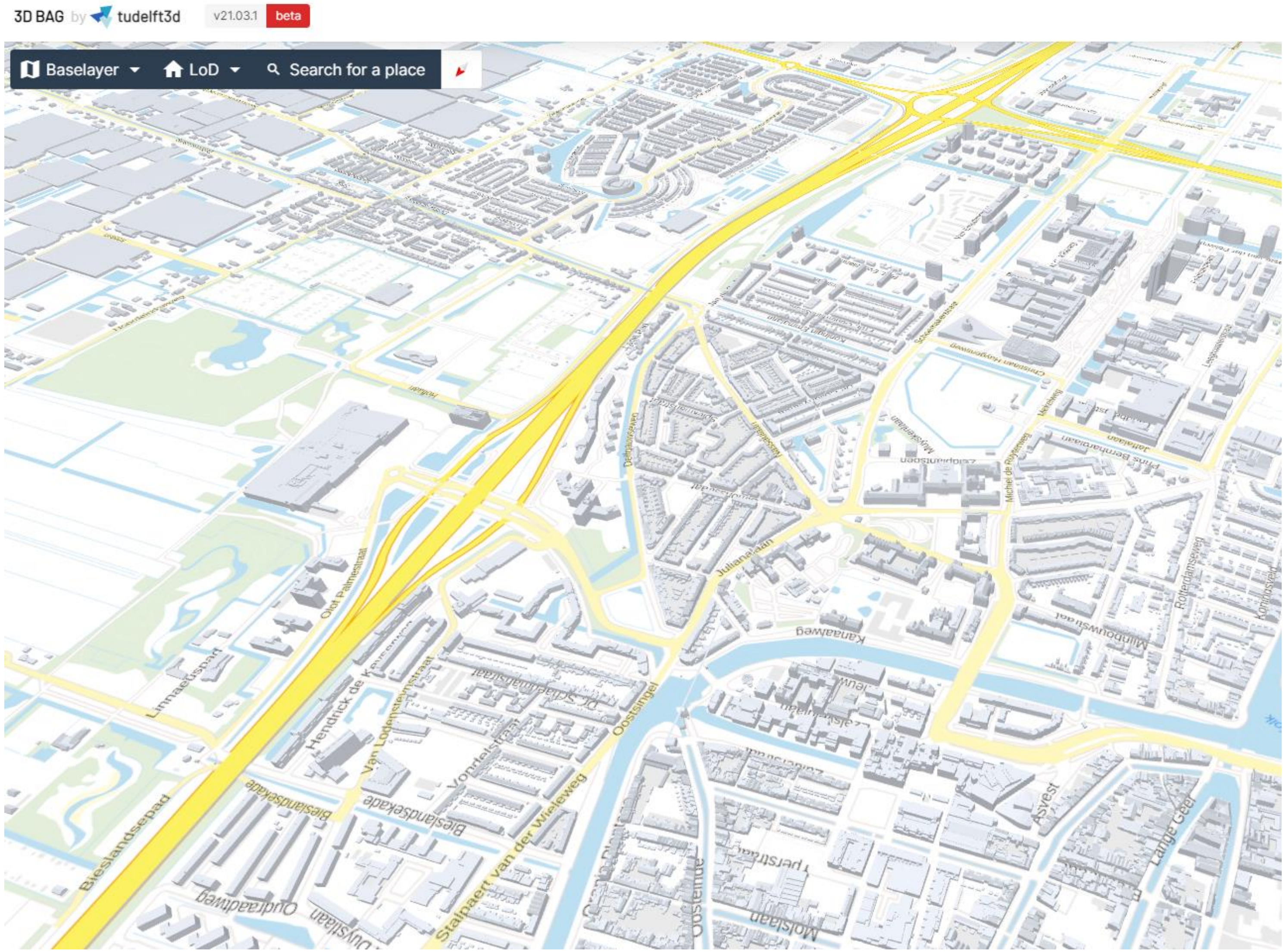
Solar3DCity

Introduction – 3D BAG

Large scale 3D city model of The Netherlands

BAG:

“Basisregistratie Adressen en Gebouwen”



Main research question

- How can the solar potential of vector buildings in large 3D city models, such as the 3D BAG data set, be computed efficiently?

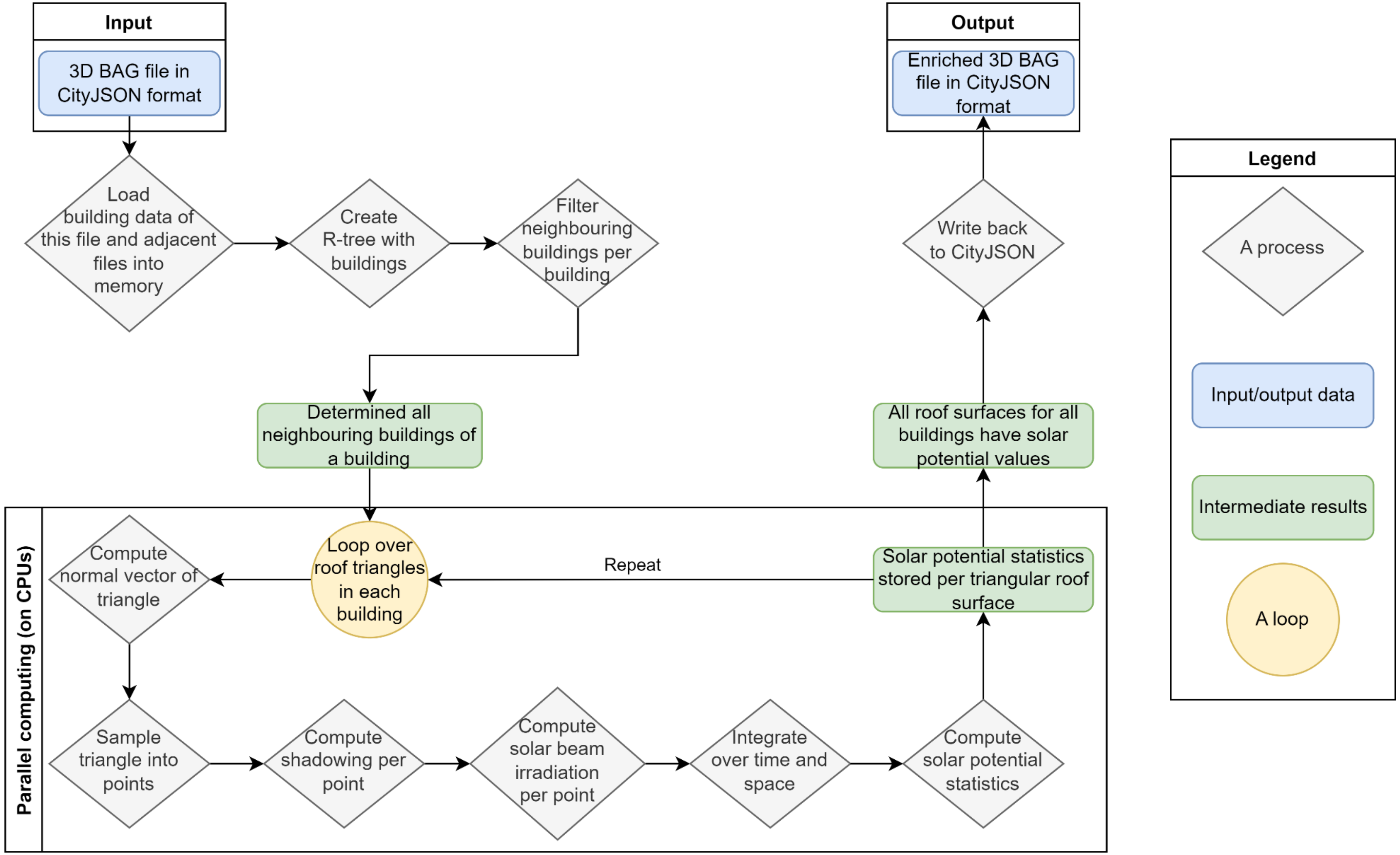
Sub-questions

1. How can spatial indexing be used to speed up shadow casting computations on the 3D BAG vector data set?
2. What simplifications in the solar irradiation model can be applied?
3. How can the solar irradiation model be implemented in Python by using open source libraries and open data?
4. How should the computer memory be managed while processing the buildings stored in tiles in the 3D BAG data set?

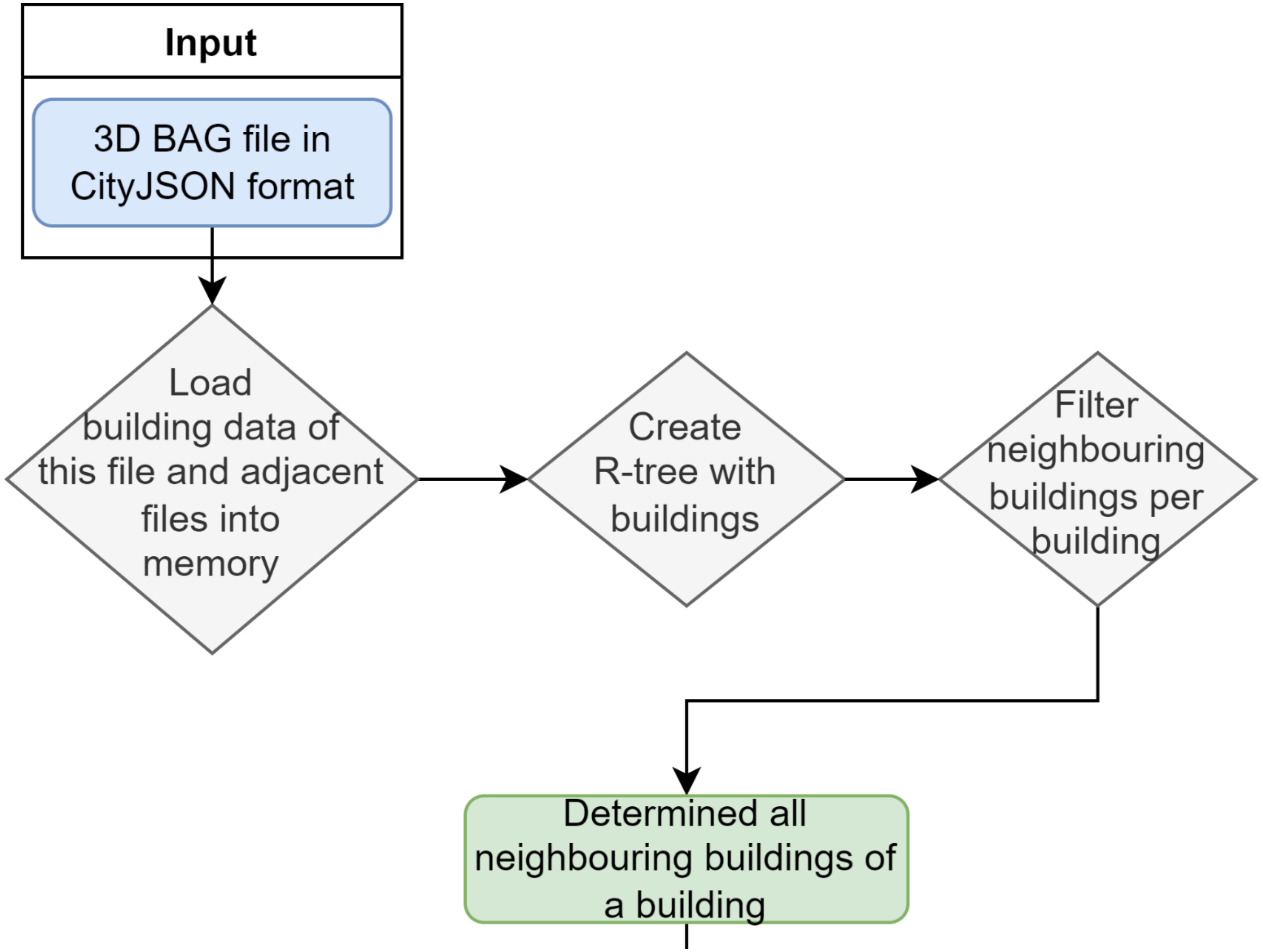
Scope

1. Only direct solar radiation is computed
2. Only on roofs, not on walls
3. For shadow casting, only buildings in 3D BAG are taken into consideration

Methodology



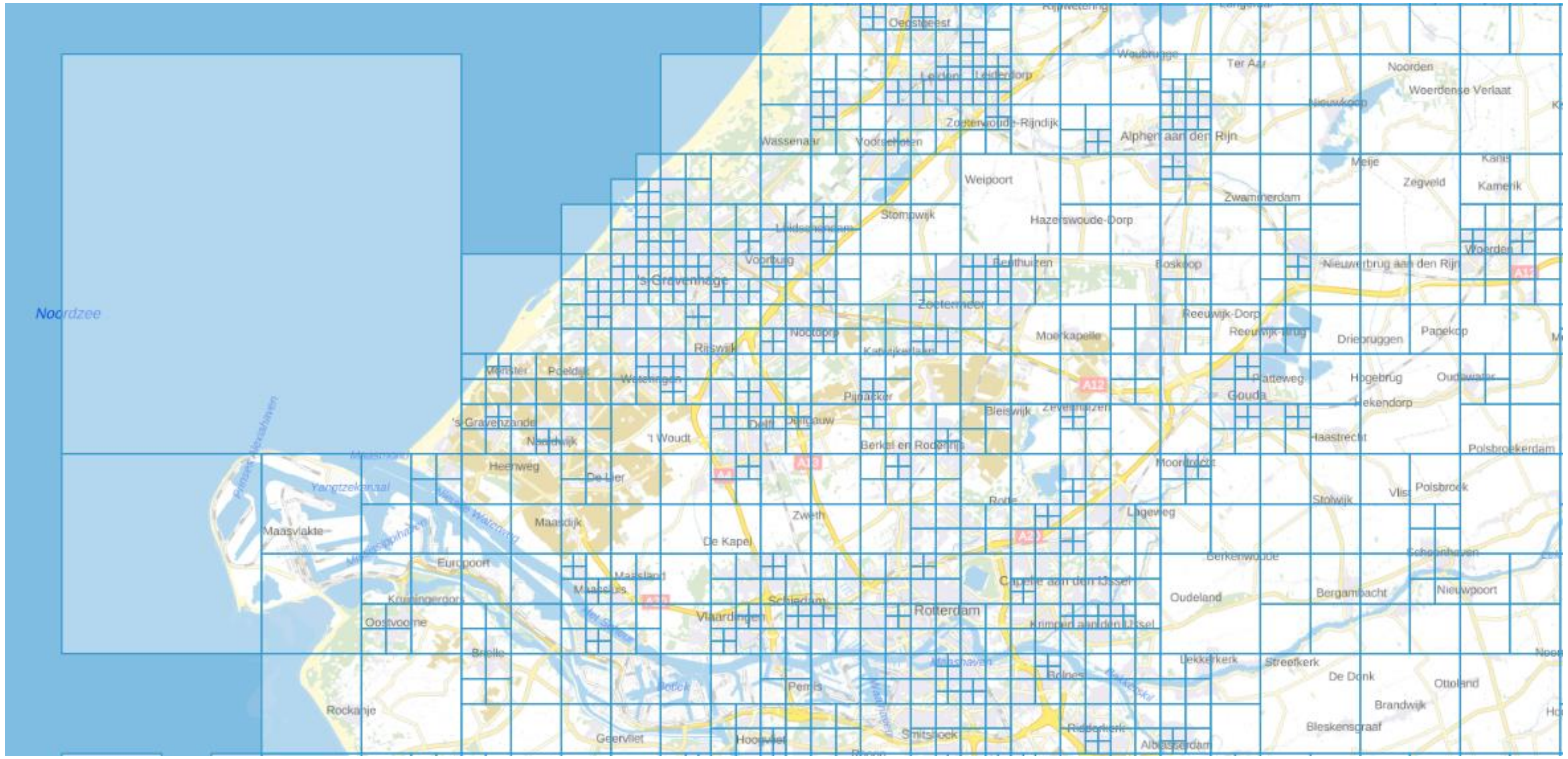
Methodology – Loading input data



Methodology – Loading input data

3D BAG as input data:

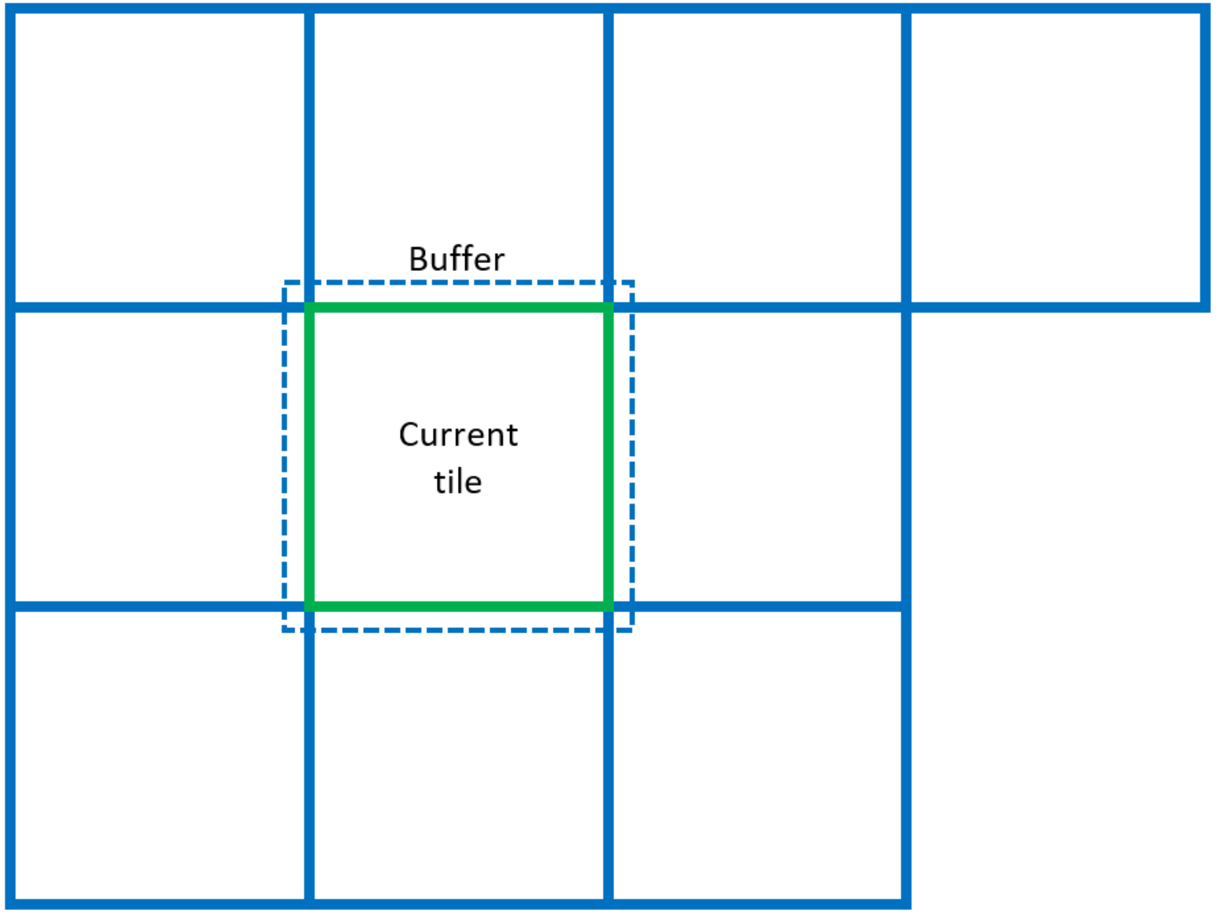
- Subdivided in tiles
- Quad tree



CityJSON

Methodology – Loading input data

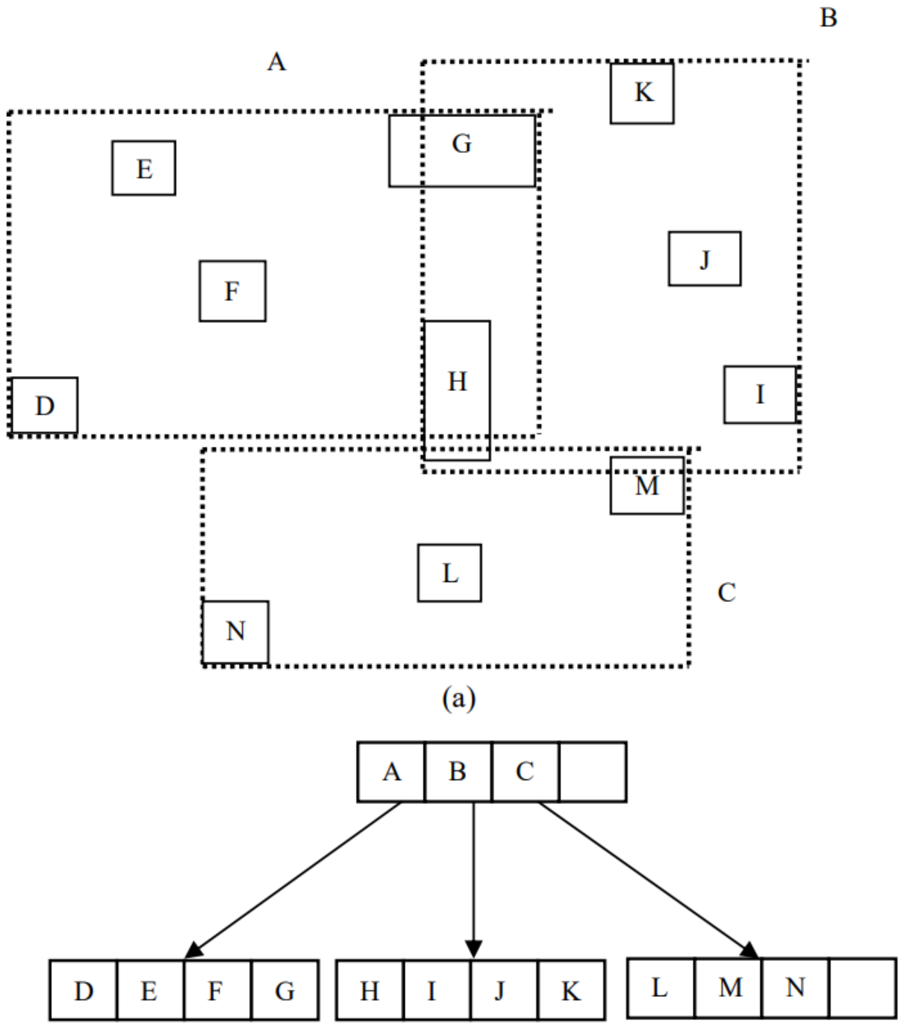
- Tiles are processed one-by-one
- Adjacent tiles needed for potential shadow casting
- Buffer of 100m applied



Methodology – Loading input data

R-tree:

- Efficient storage and retrieval of spatial data – buildings
- Hierarchical tree structure
- Bounding boxes



(a)
(b)

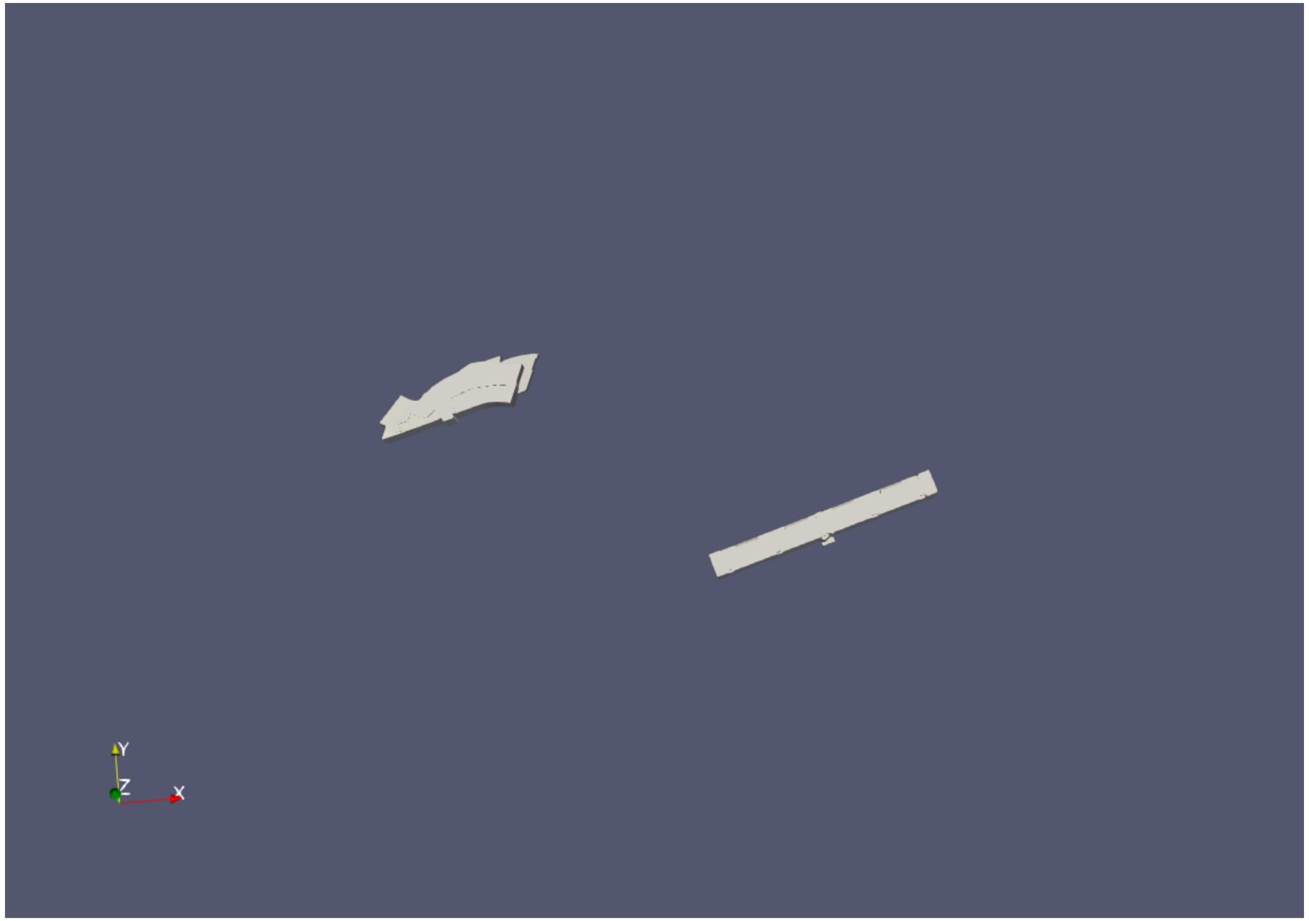
TU Delft

Methodology – Loading input data

- For each building, its neighbouring buildings are filtered on:
  - Distance
  - Height
  - Orientation

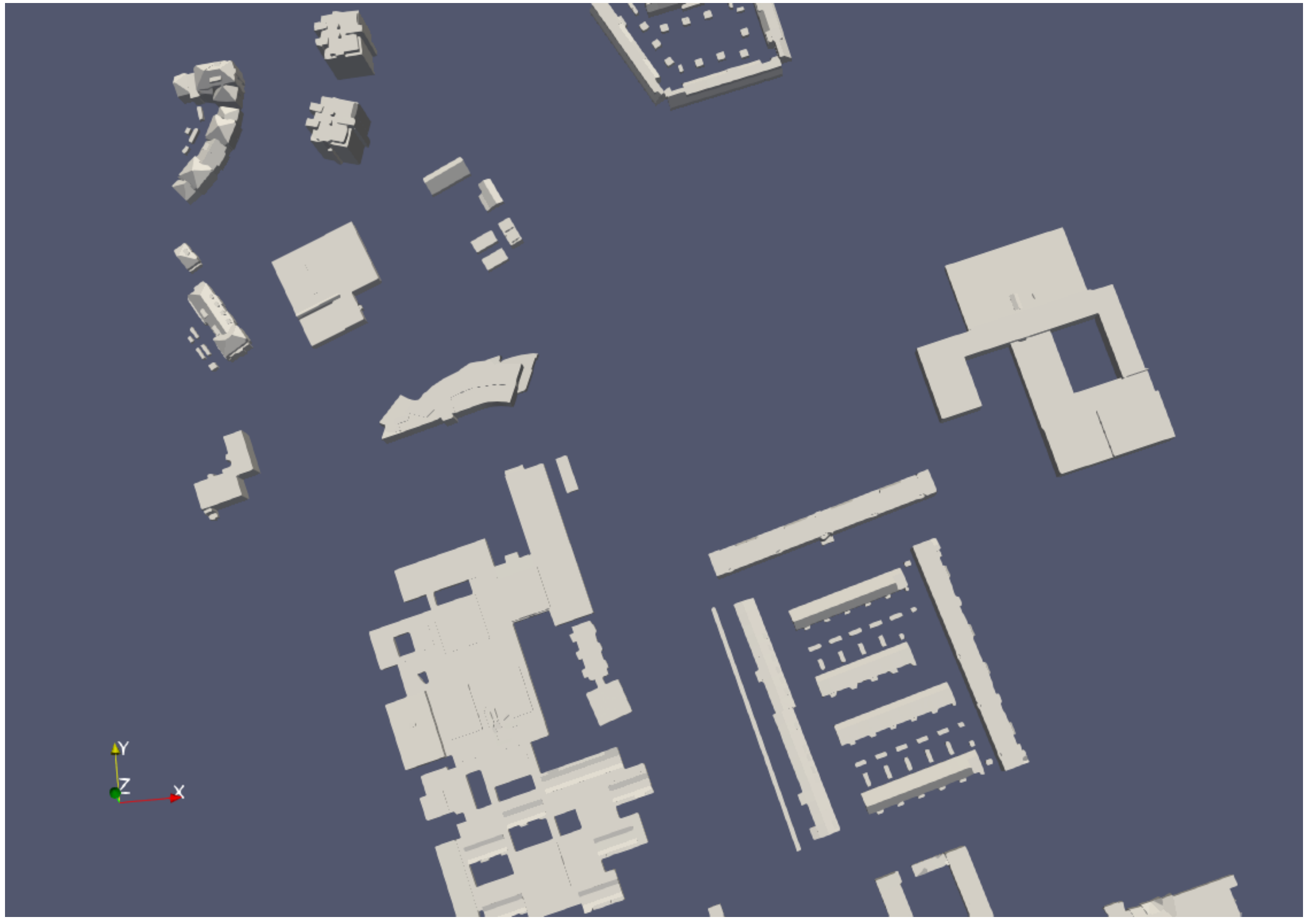
Methodology – Loading input data

Example for two buildings



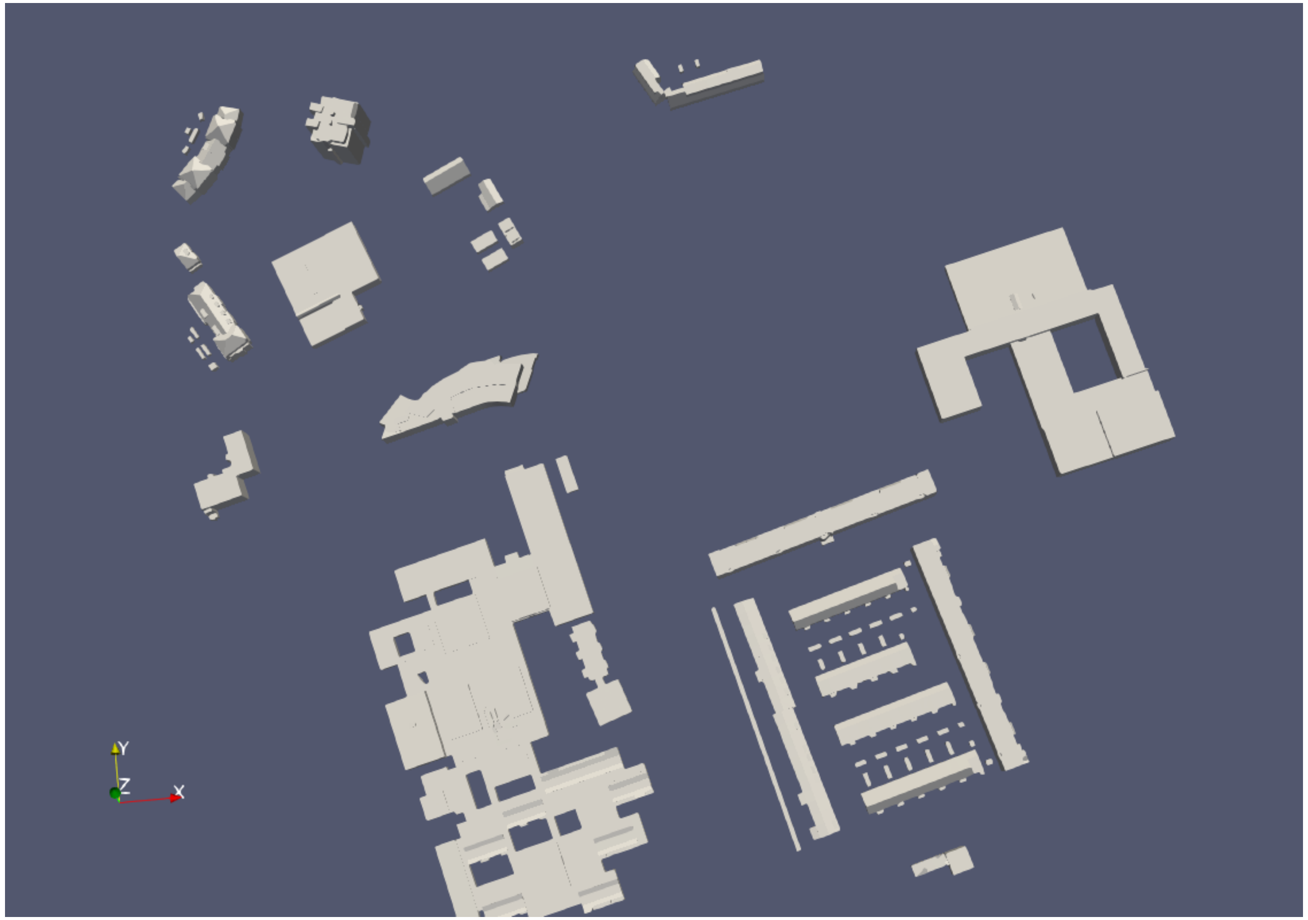
Methodology – Loading input data

All neighbouring buildings visible



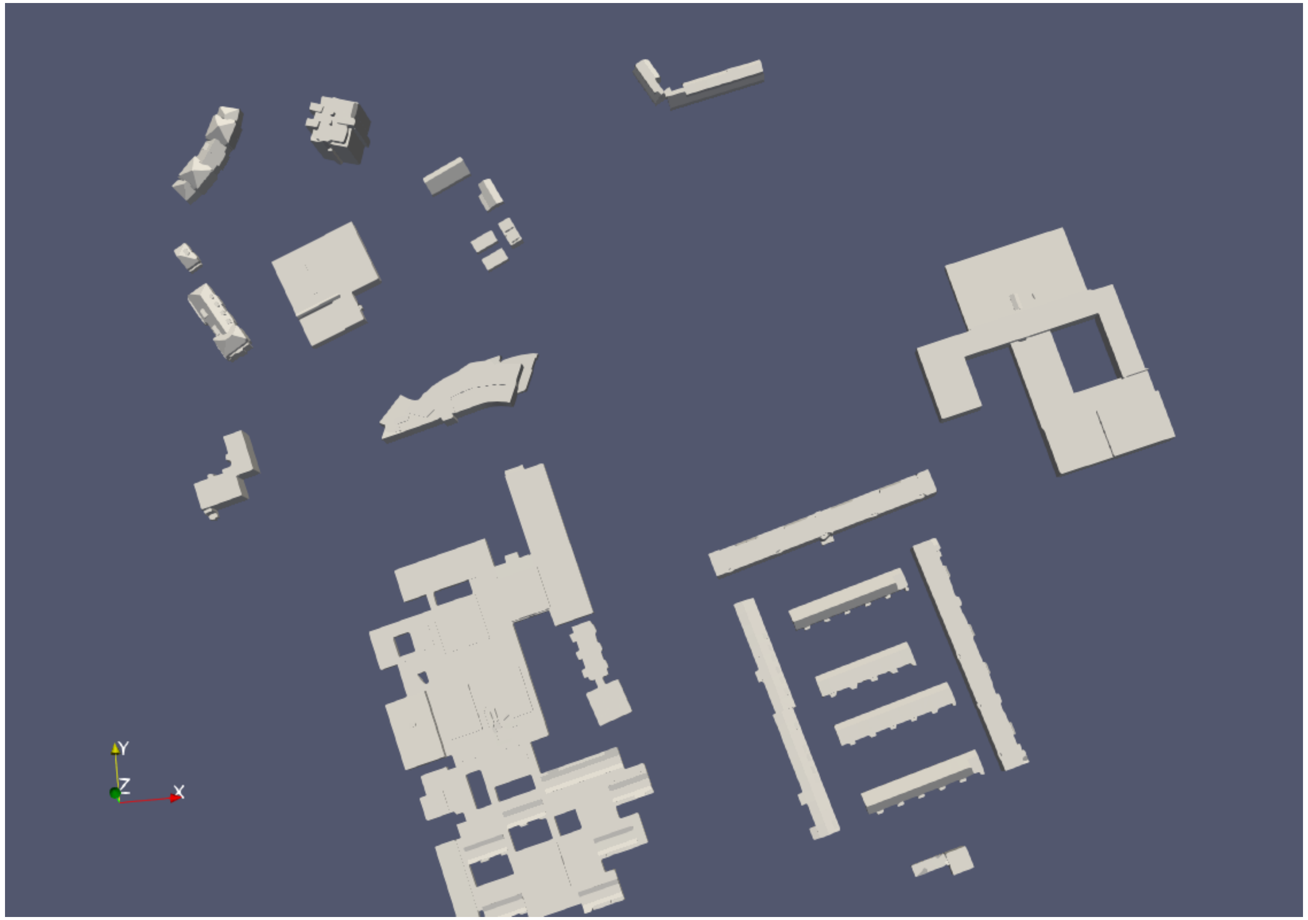
Methodology – Loading input data

Filtered on distance to neighbouring buildings



Methodology – Loading input data

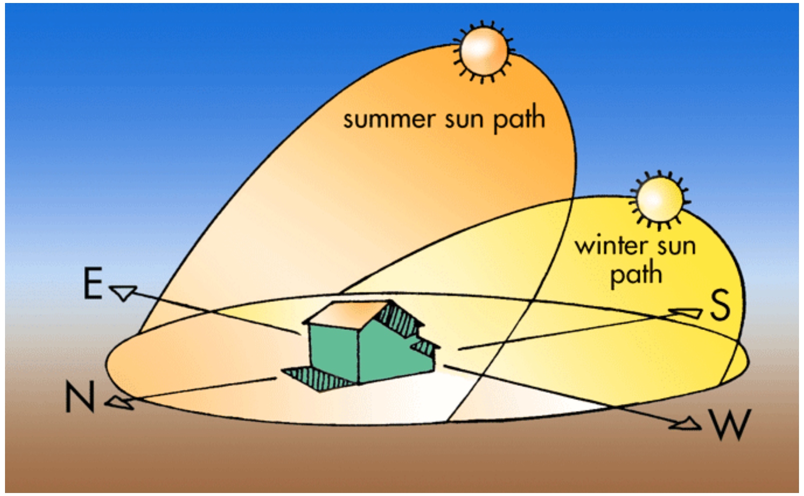
Filtered on height of neighbouring buildings

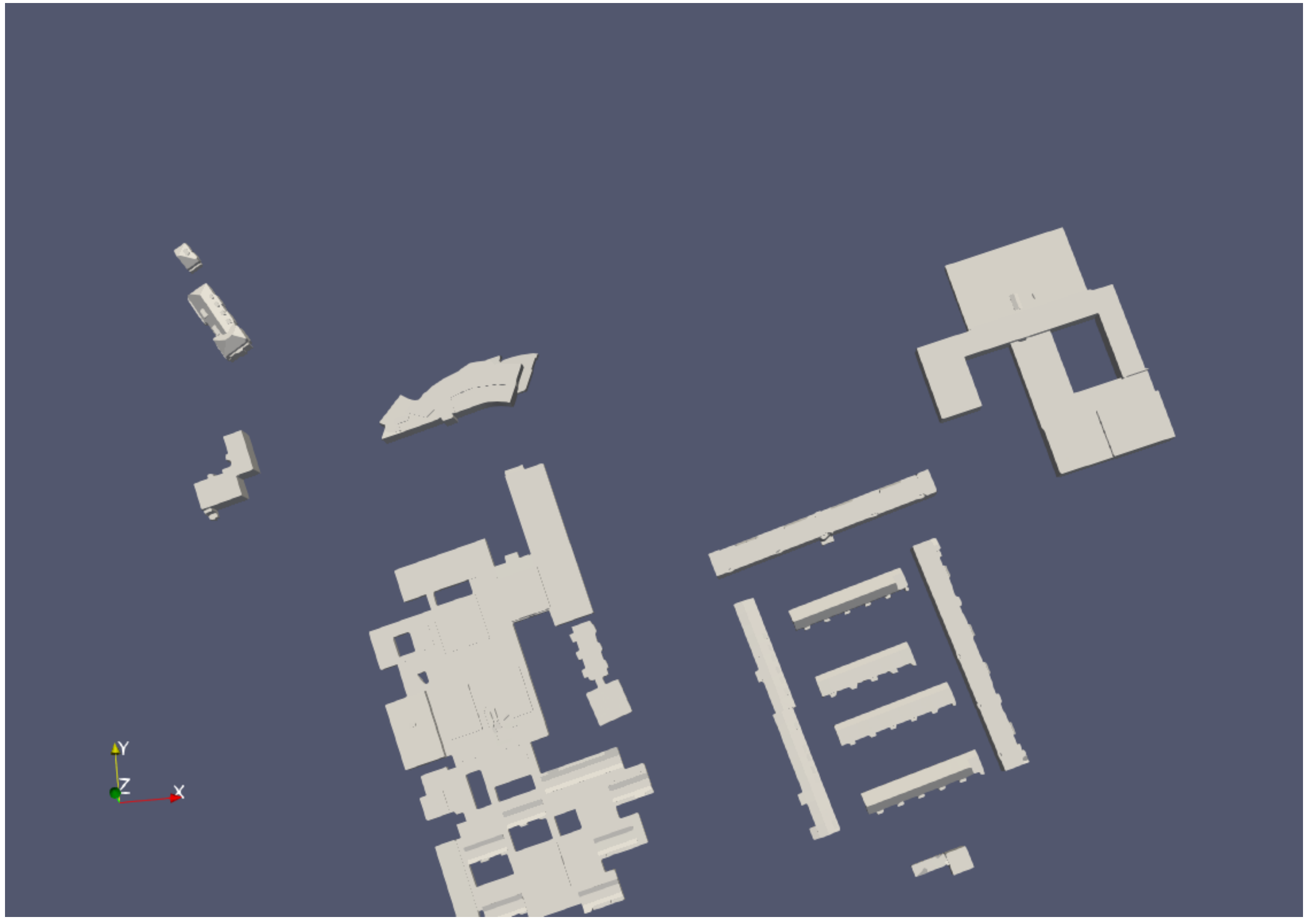


TU Delft

Methodology – Loading input data

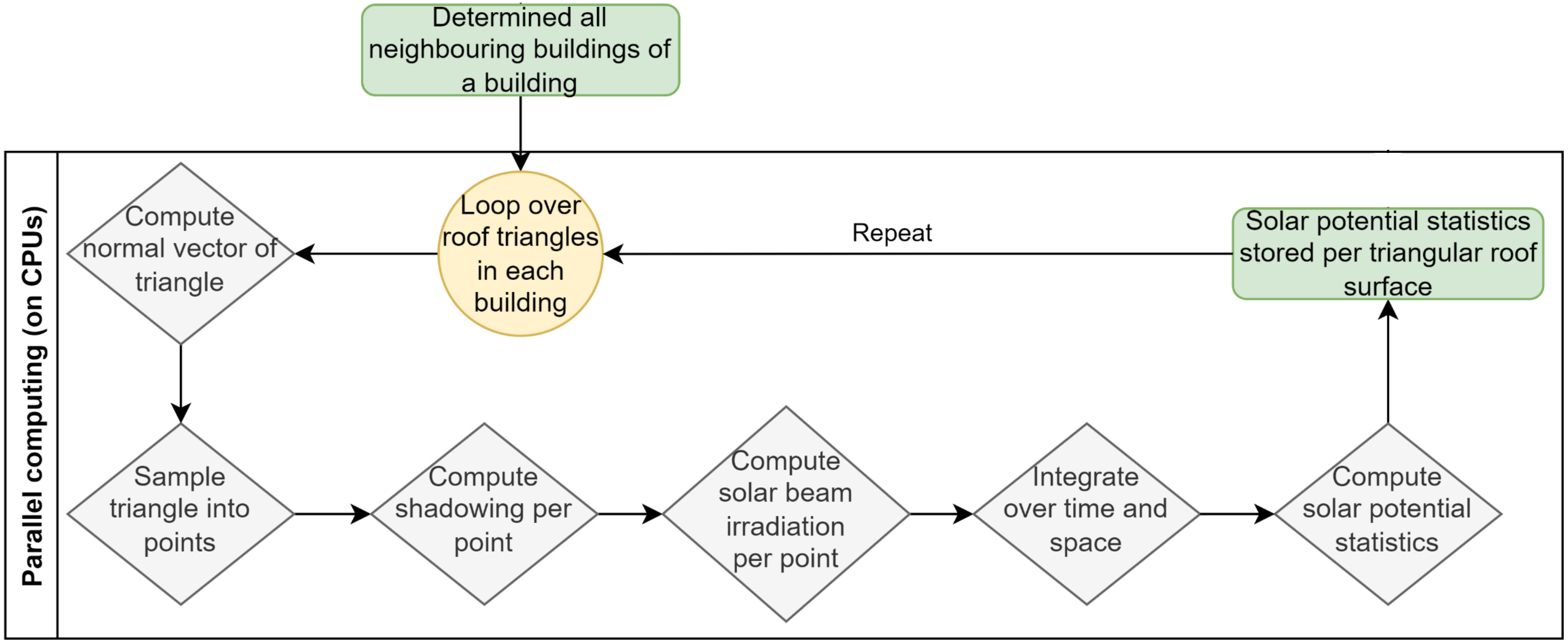
Filtered on orientation of neighbouring buildings based on the sun's position





TU Delft

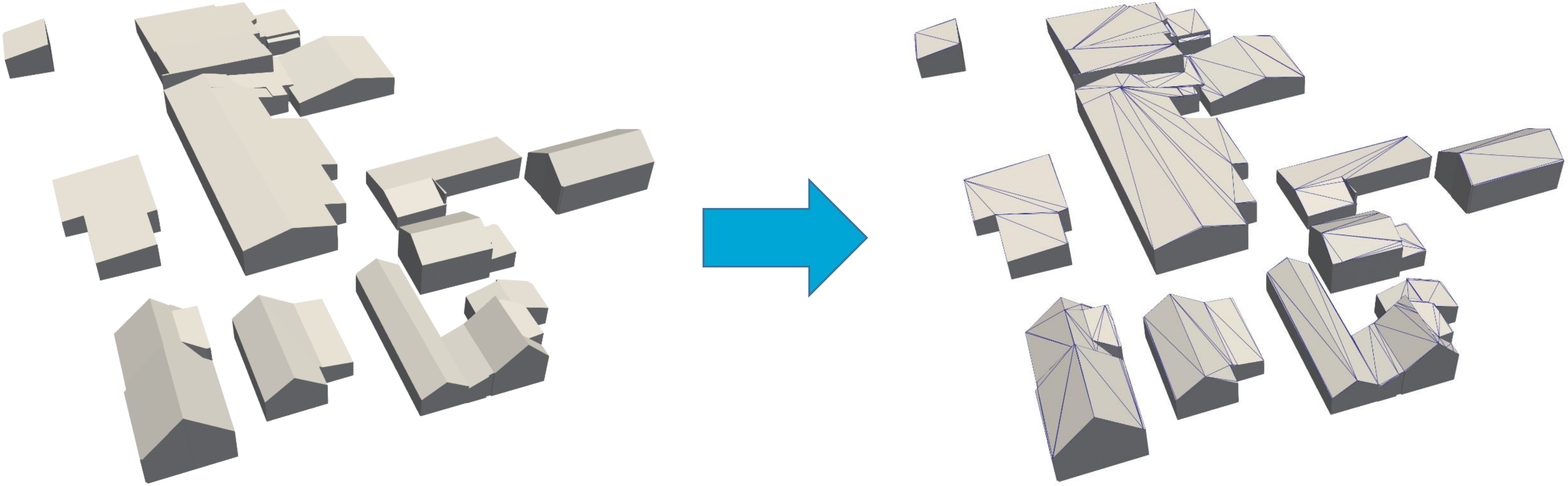
Methodology – Processing building geometries



TU Delft

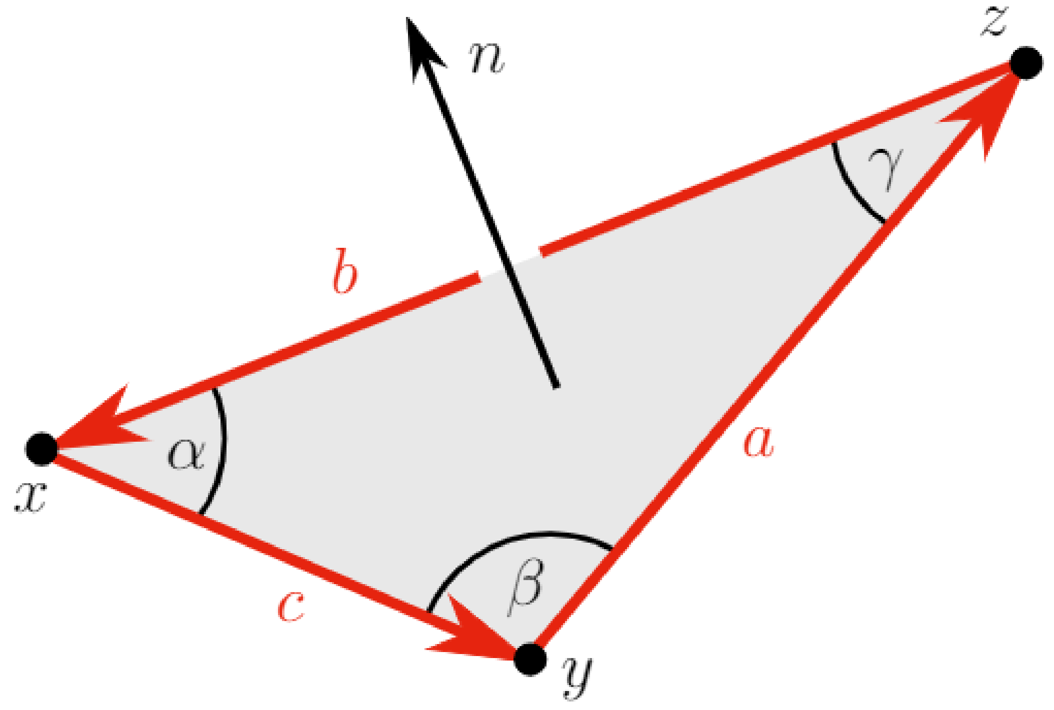
Methodology – Processing building geometries

The buildings are made up of triangular surfaces



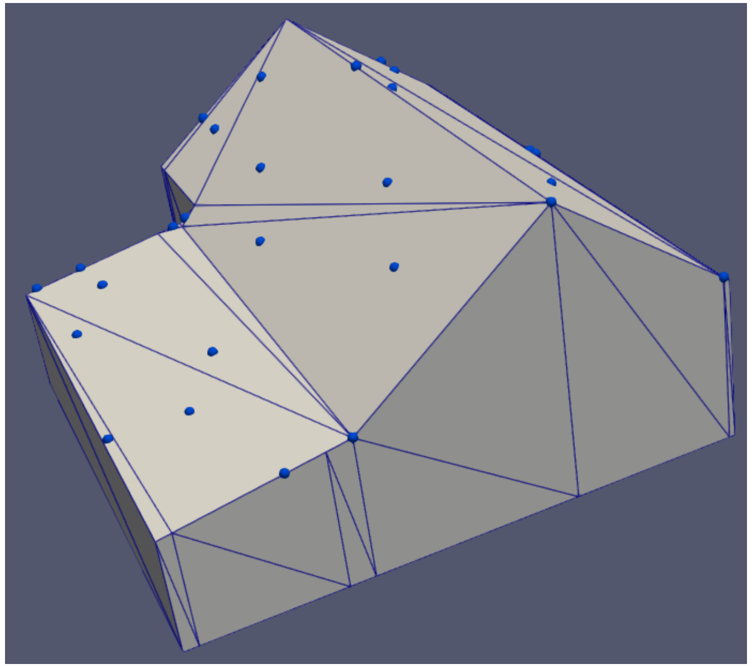
Methodology – Processing building geometries

- Compute the normal vector of a triangle
- Needed to determine the direction the triangle points towards

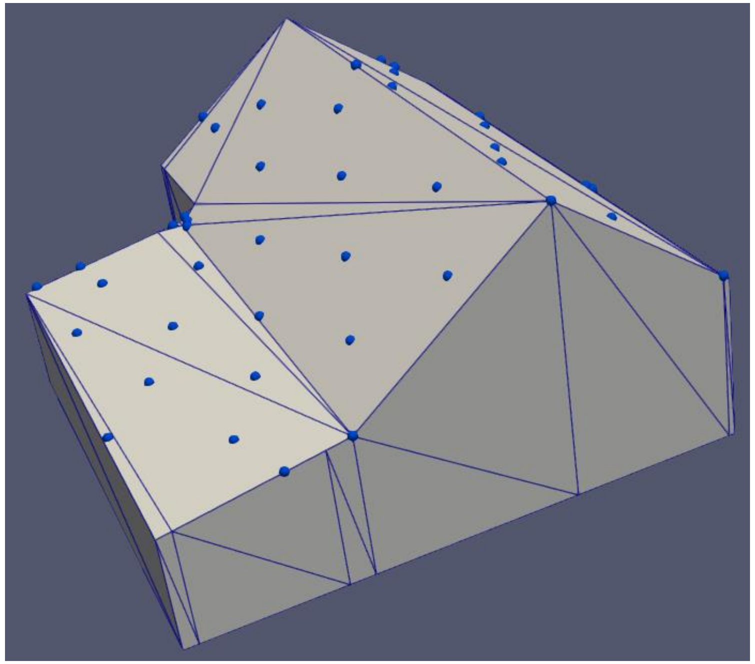


Methodology – Processing building geometries

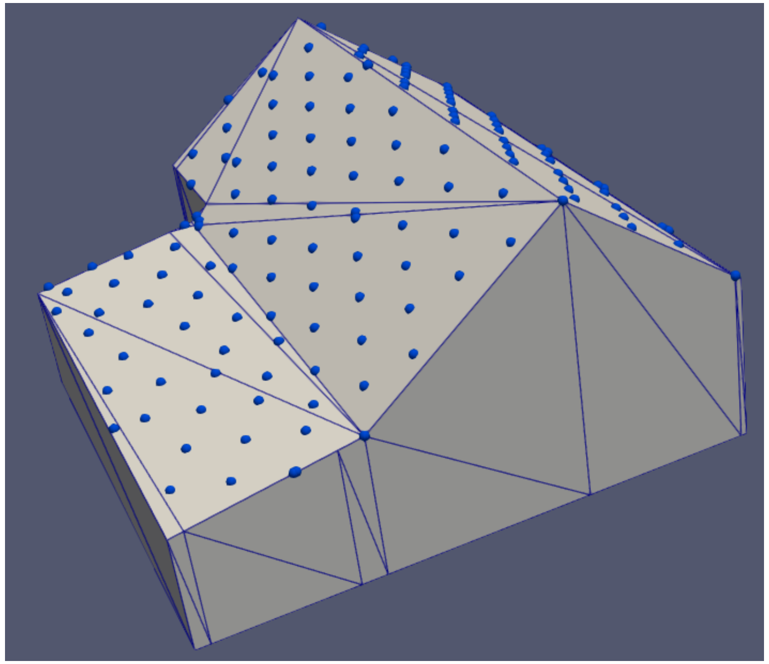
- Sample triangle into a grid of points



3m distance



2m distance

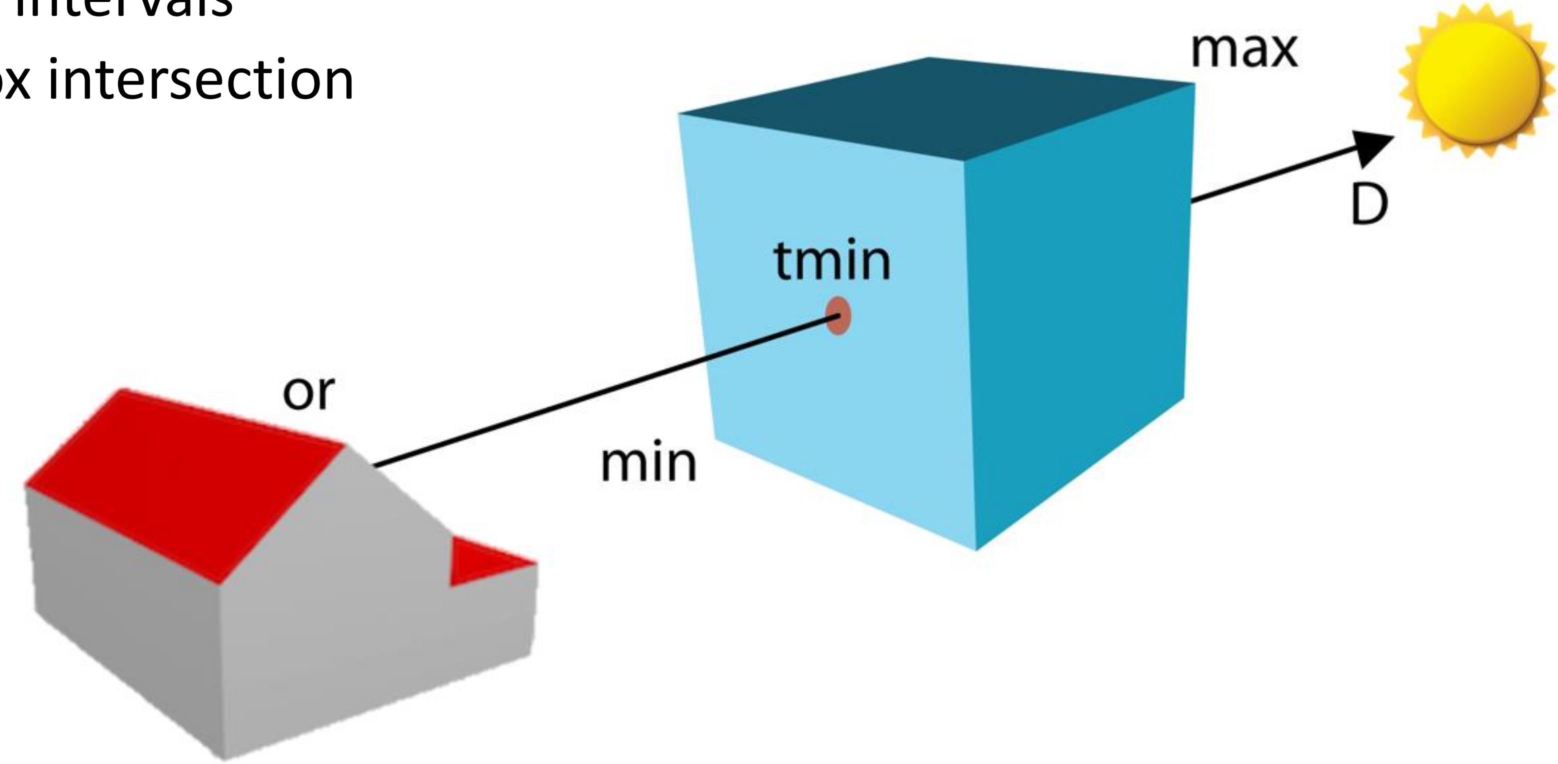


1m distance

TU Delft

Methodology – Processing building geometries

- Compute shadowing per point
  - Hourly intervals
  - Ray-box intersection

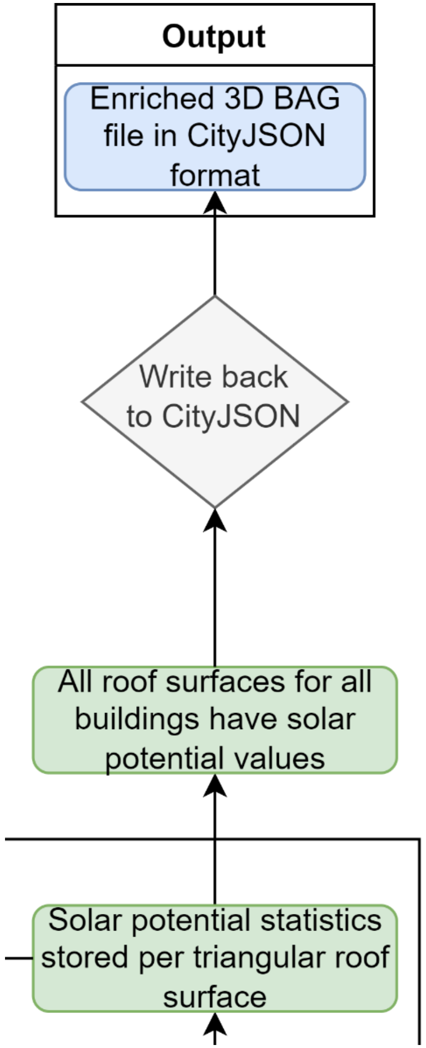


TU Delft

Methodology – Processing building geometries

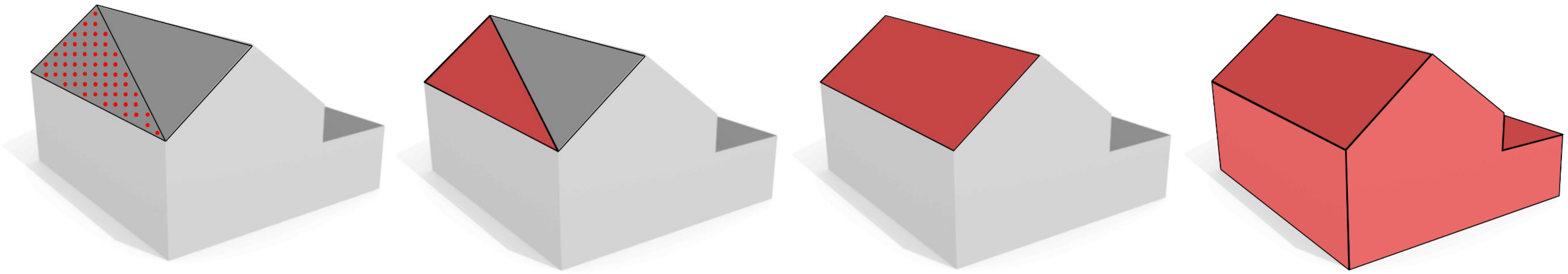
- Solar beam radiation is computed by using the solarpy library in Python
- Computed per point
  - Hourly intervals
  - 1 day per month
  - 12 months a year
- Aggregate solar radiation over time and space
  - Hour → day → month → year
  - Point → triangle

Methodology – Storing and writing solar potential values



Methodology – Storing and writing solar potential values

- Solar potential can be stored in various ways



Per sampled grid point

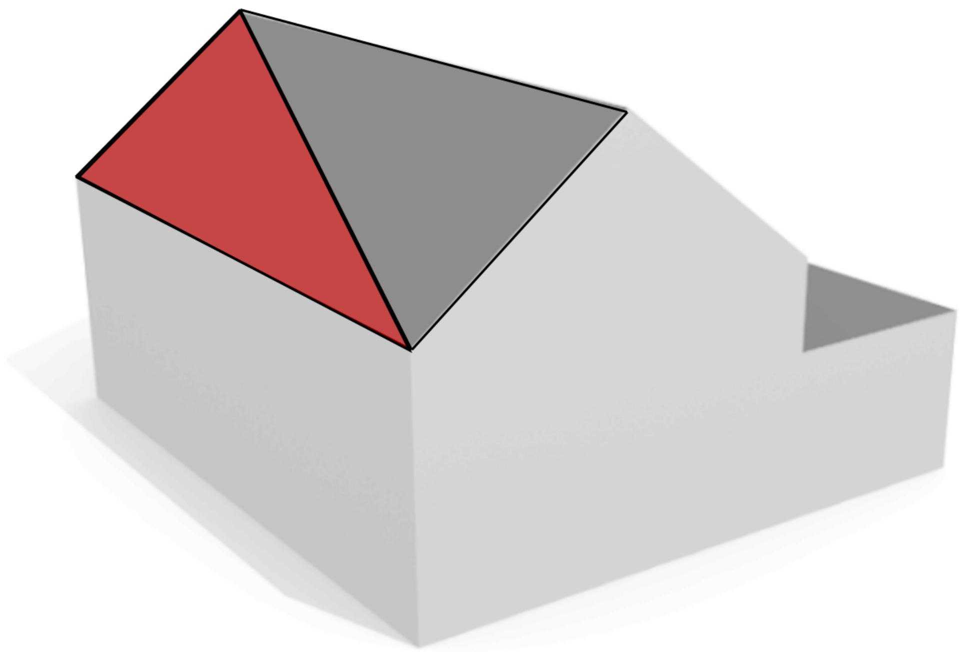
Per triangular roof surface

Per polygonal roof surface

Per building as a whole

Methodology – Storing and writing solar potential values

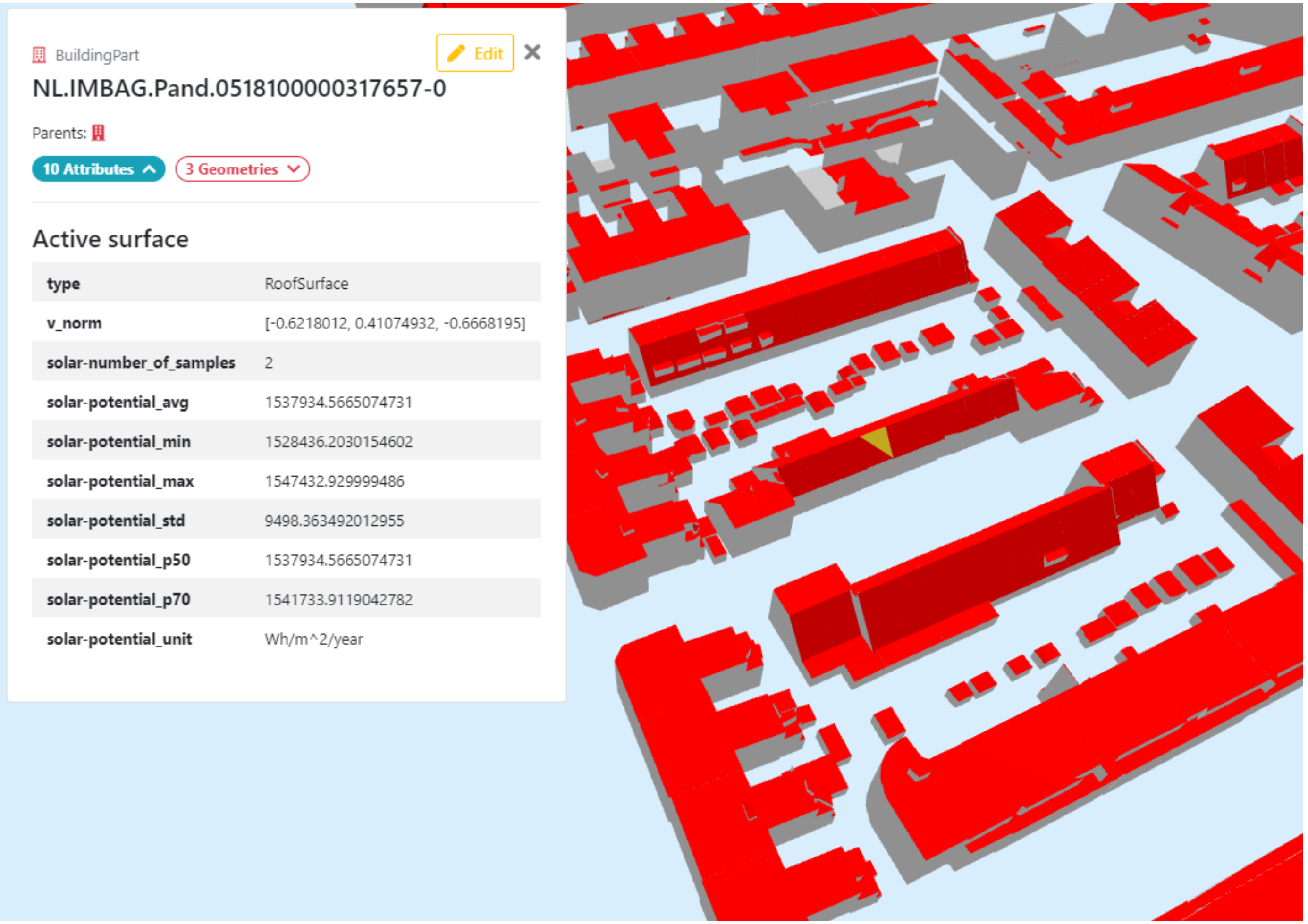
- Chosen to store solar potential per triangular roof surface



Methodology – Storing and writing solar potential values

- Compute and store solar potential statistics per triangle:
  - Number of sampled points
  - Average
  - Maximum
  - Minimum
  - Standard deviation
  - 50^{th} percentile
  - 70^{th} percentile
  - The unit in $Wh/m^2/year$

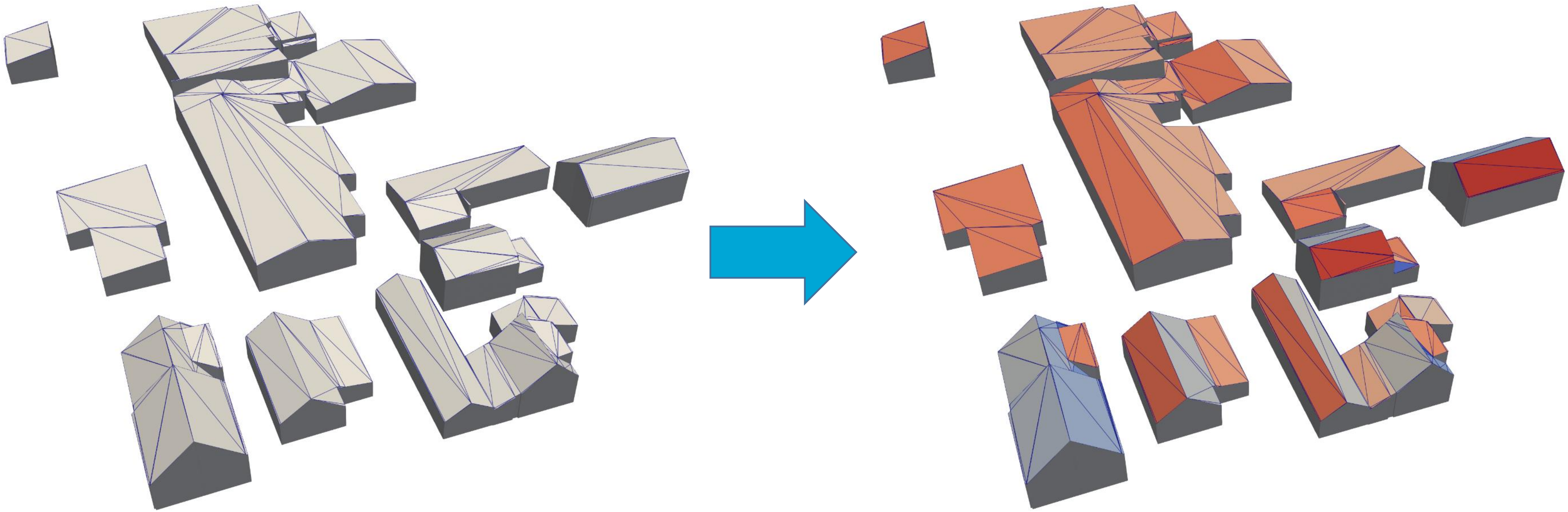
Methodology – Storing and writing solar potential values



Results

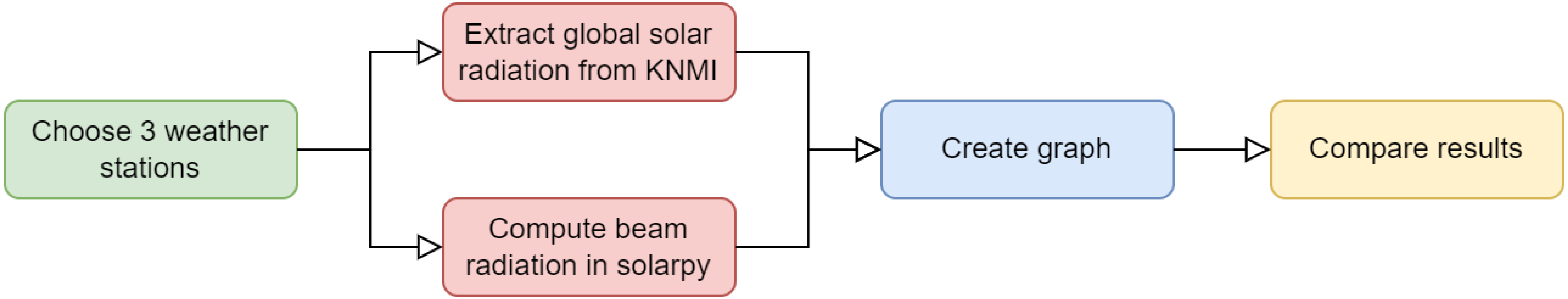
- General
- Quality Assessment
  - Ground truth comparison
  - ArcGIS comparison
- Scalability Assessment

Results - General



Results – Quality Assessment

- Ground truth comparison
  - Workflow

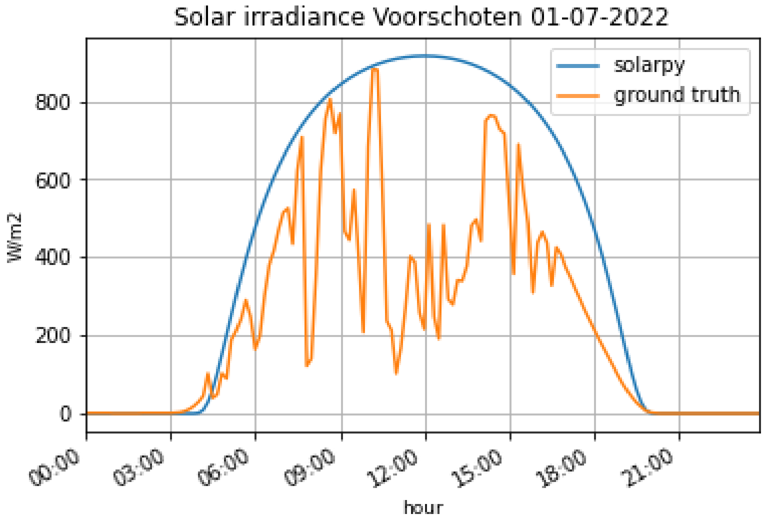


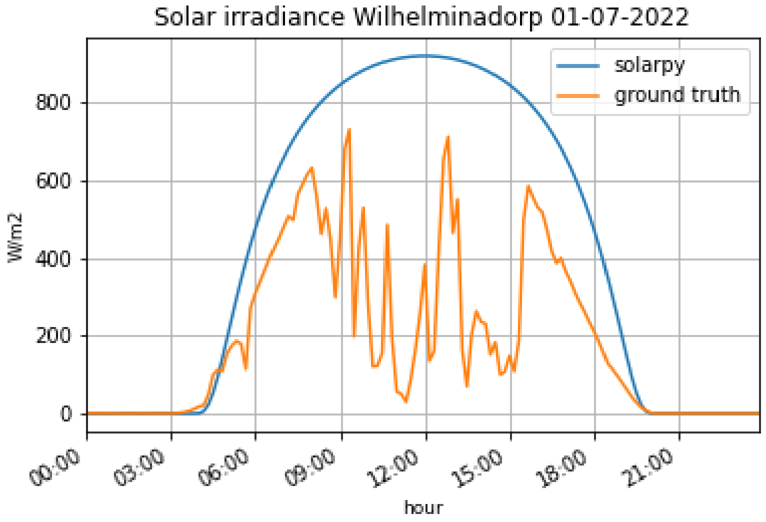
Results – Quality Assessment

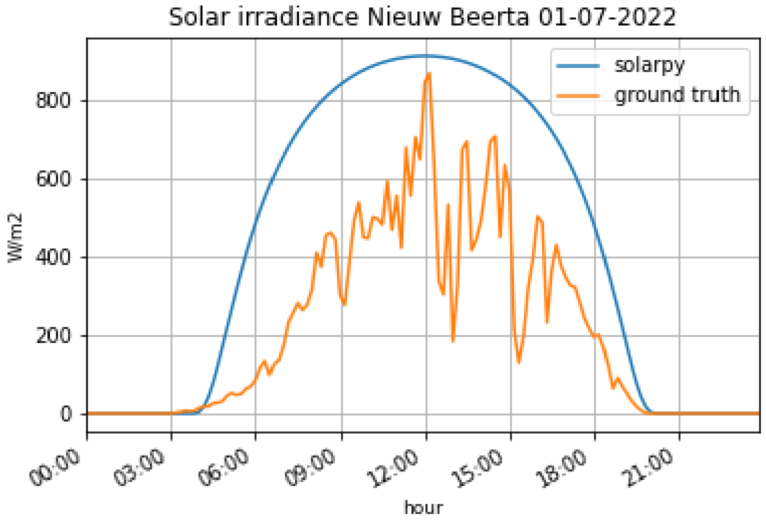
- Ground truth comparison
  - Weather stations

Results – Quality Assessment

- Ground truth comparison results
  - Weather stations versus solarpy

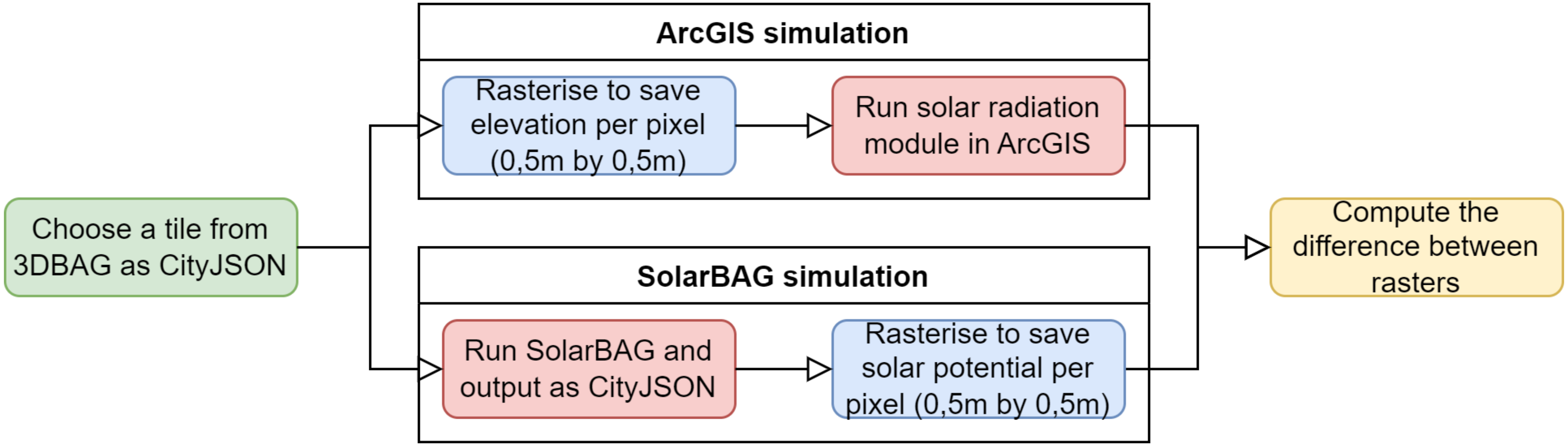






Results – Quality Assessment

- ArcGIS raster comparison

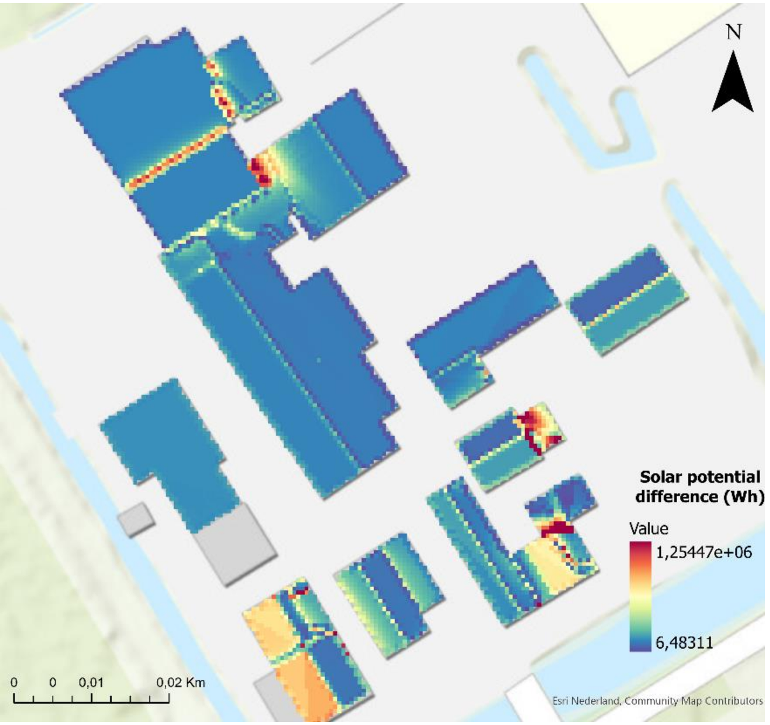


TUDelft

Results – Quality Assessment

Solar radiation raster for ArcGIS method

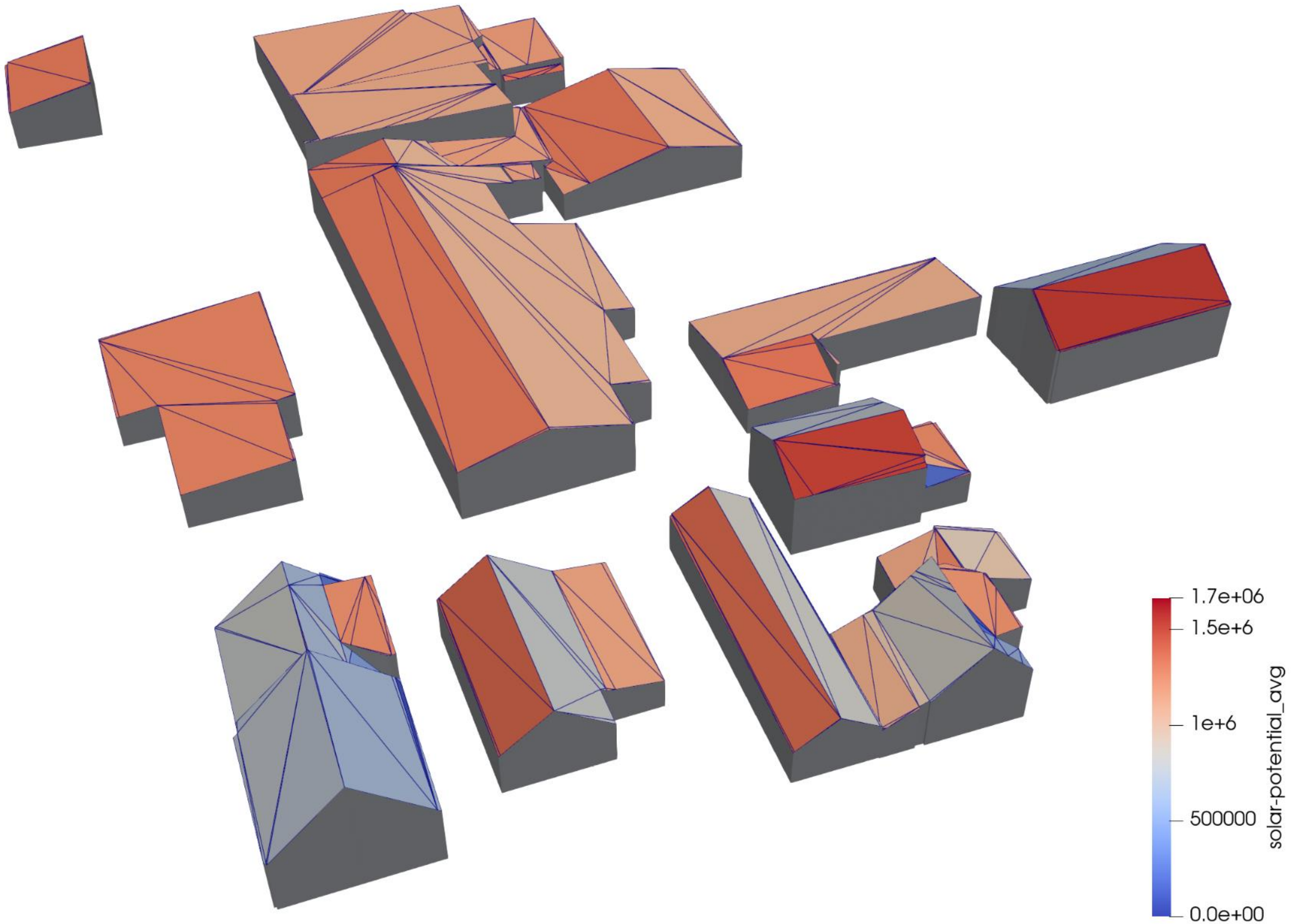
Solar radiation raster for vector method (SolarBAG)



Absolute difference raster between the two methods

TU Delft

Results – Quality Assessment

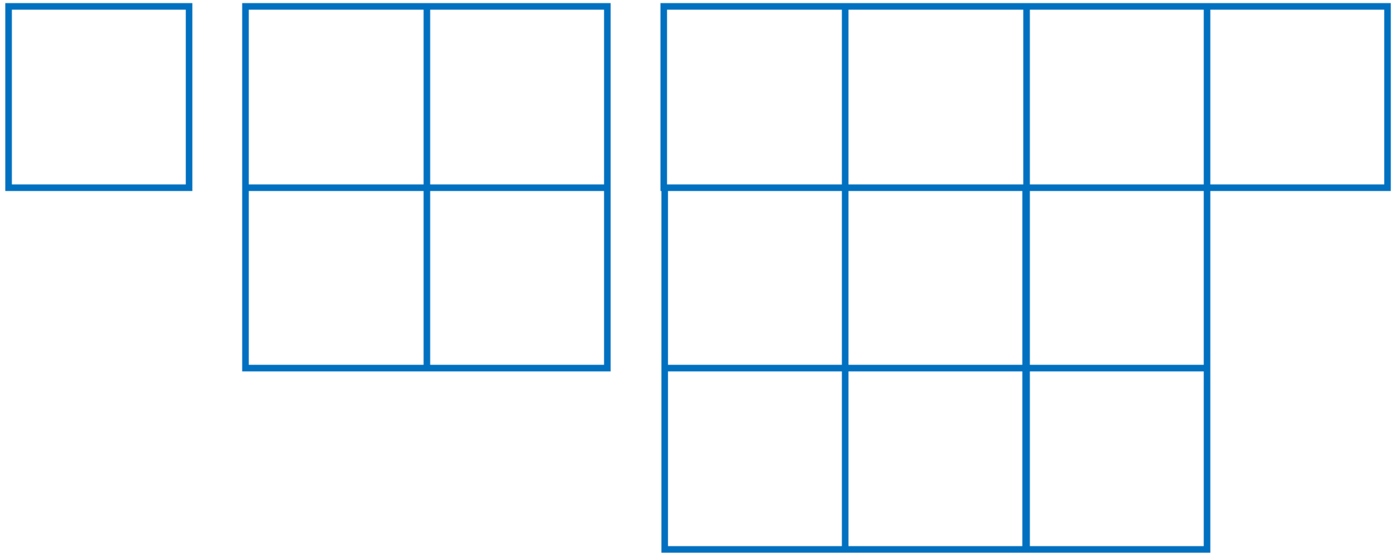


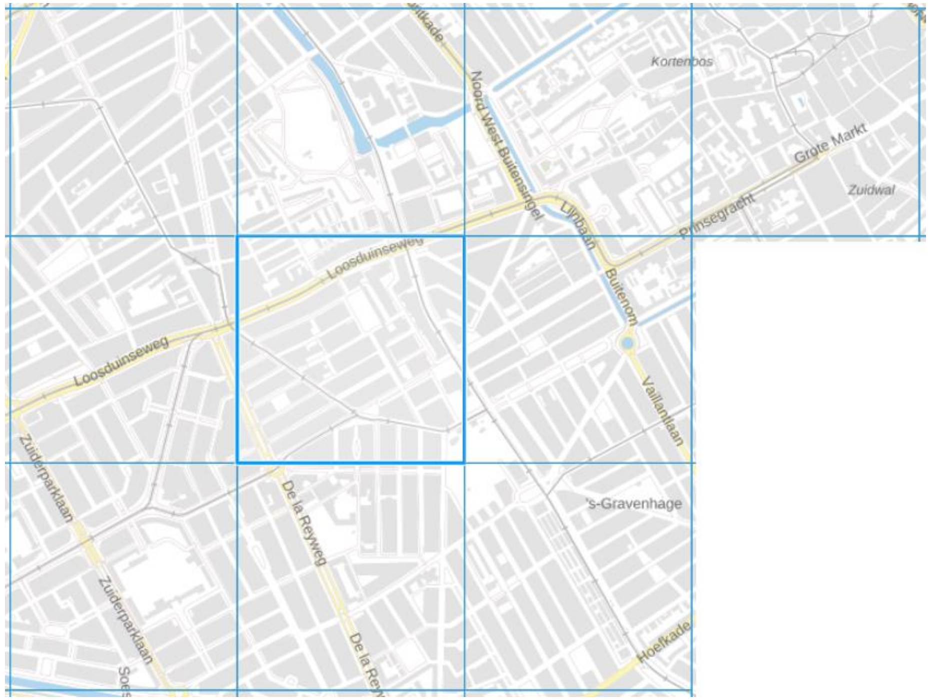
Results – Quality Assessment

- Factors causing the differences between the rasters:
  - Raster/sampling/storing resolution
  - Different algorithms/formulas
  - Impossible to match input parameters exactly
    - Transmittivity factor

Results – Scalability Assessment

1 tile **4 tiles** **10 tiles**

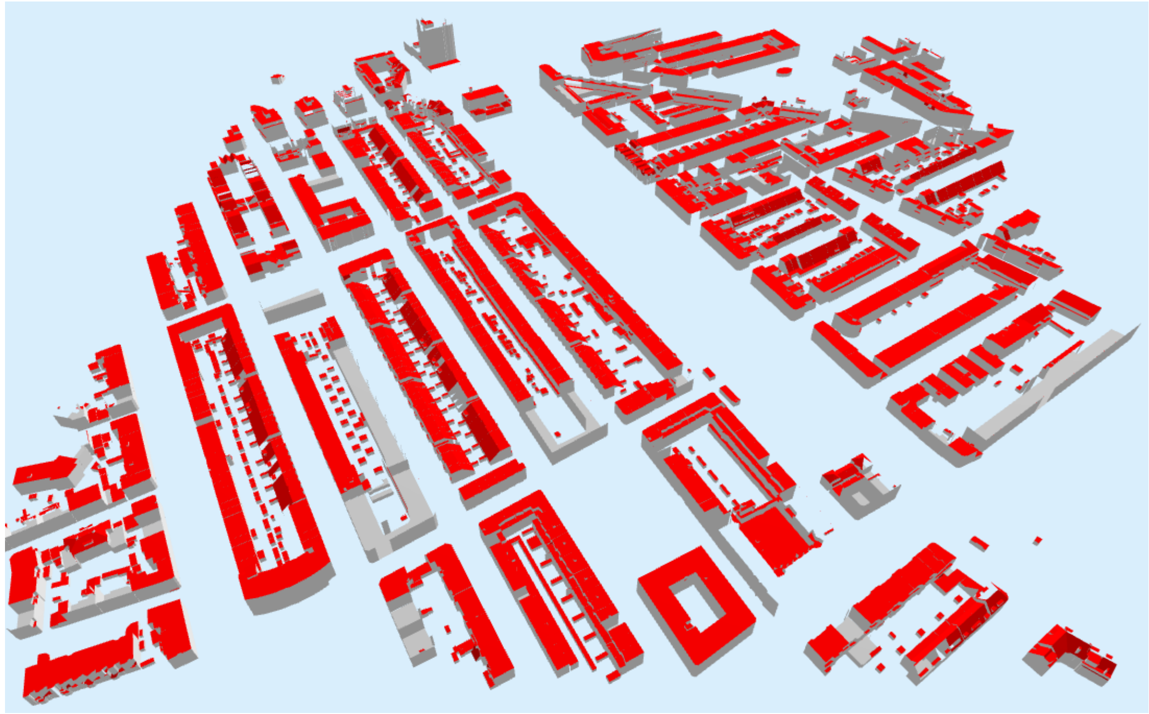




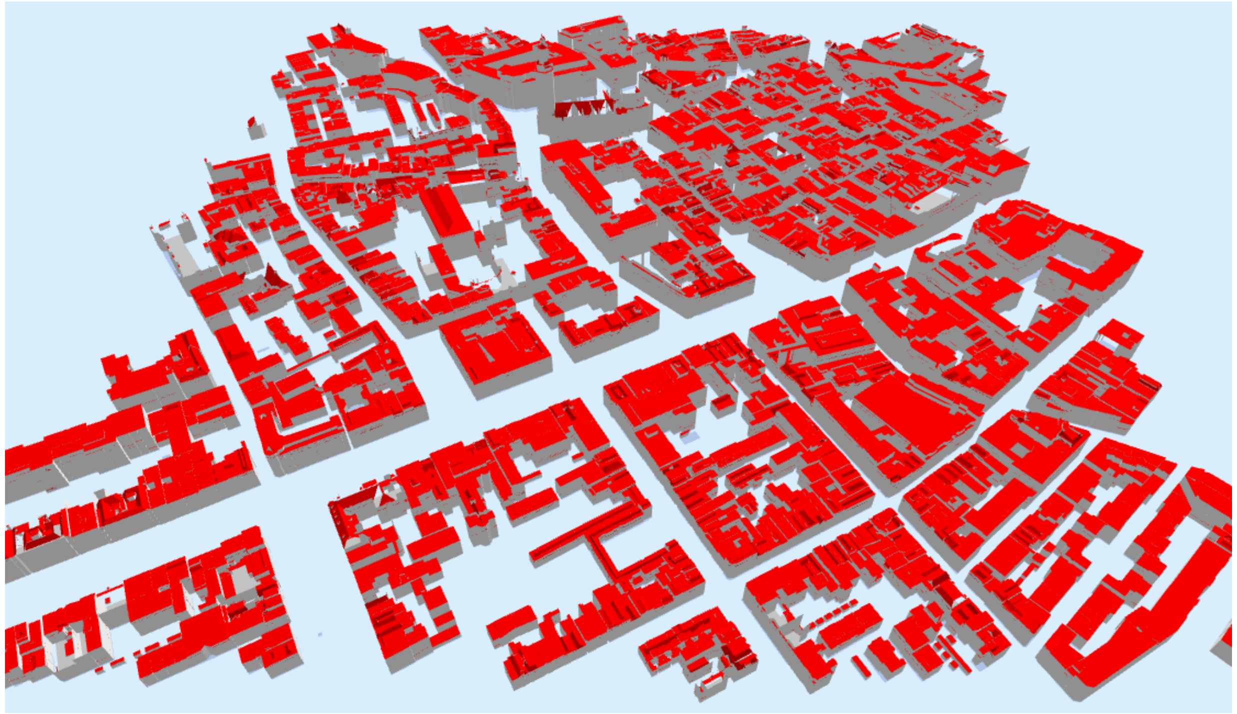
Results – Scalability Assessment

	Tile ID	Tile extent (m)	No. of buildings in tile	No. of buildings in R-tree	No. of neighbour tiles	Time (hh:mm:ss)
Tile 1	5801	652 x 686	1735	2760	3	02:24:26
Tile 2	5802	652 x 638	1620	2646	5	01:11:06
Tile 3	5805	636 x 635	1688	2408	3	00:47:08
Tile 4	5872	644 x 654	1724	2644	5	00:48:23
Tile 5	5873	650 x 648	1400	1976	3	00:45:50
Tile 6	5875	671 x 647	1236	2412	8	01:06:04
Tile 7	5876	658 x 639	955	1951	5	00:52:54
Tile 8	5877	658 x 659	888	1667	4	00:44:12
Tile 9	5878	636 x 679	1352	1981	6	01:02:03
Tile 10	5880	694 x 646	1124	1475	2	01:46:05
Total time						**11:28:11**

Results – Scalability Assessment



Tile 5 | 1400 buildings
3 neighbouring tiles | 00:45:50



Tile 10 | 1124 buildings
2 neighbouring tiles | 01:46:05

Conclusion

Main research question

How can the solar potential of vector buildings in large 3D city models, such as the 3D BAG data set, be computed efficiently?

1. Using R-trees for spatial indexing
2. Filtering neighbouring buildings for shadow casting
3. Using multiprocessing and open source libraries in Python
4. Loading and processing 3D BAG tiles one-by-one

Conclusion

- The implemented solar radiation model (SolarBAG) enriches 3D buildings with solar radiation values
- It is an indication to determine buildings with highest solar potential
- Further inspection is needed to actually install solar panels

Conclusion - Limitations

- Solar radiation values are not always correct
  - Inconsistencies due to resolution
  - Flipped orientation of roof surfaces
- The solar radiation model is not complete
- The model is still quite slow
- The model is not applicable to other data formats

Future work

- Efficient implementations improvements
  - Make use of GPU
  - Apply more and better filters for selecting neighbouring buildings
  - Mesh simplification
- Solar radiation model extensions
  - Incorporate diffuse and reflected radiation
  - Extend to solar radiation computation on walls as well
  - Use a database instead of processing files separately
- Additional analyses
  - Parameter tweaking to find a trade-off between performance and accuracy

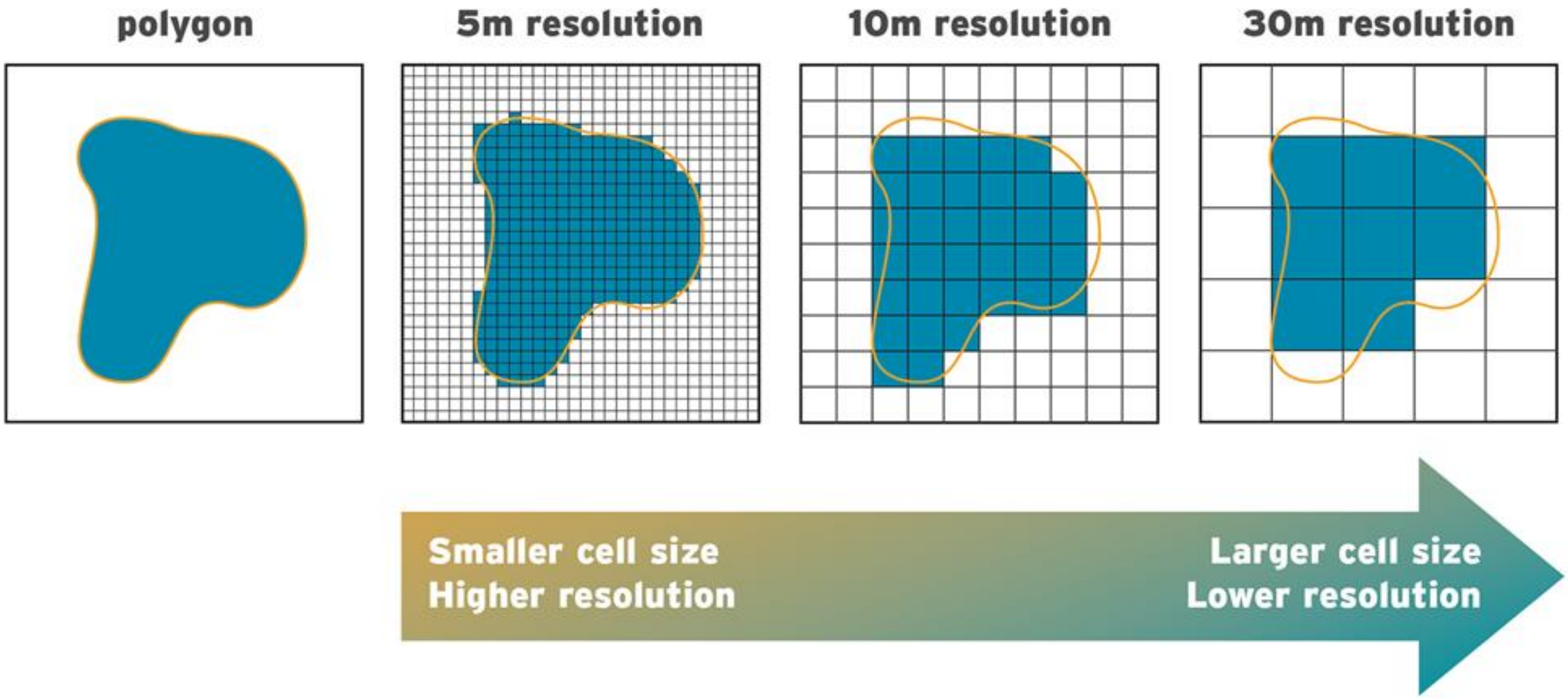
Thank you for your attention!

References

- Slide 3: https://www.consumentenbond.nl/zonnepanelen/zonnepanelen-kopen
- Slide 4: https://www.cupapizarras.com/usa/news/10-types-of-roofs/
- Slide 5: https://www.researchgate.net/publication/323940752_Solar_Energy_Potential_Assessment_on_Rooftops_and_Facades_in_Large_Built_Environments_Based_on_LiDAR_Data_Image_Processing_and_Cloud_Computing_Methodological_Background_Application_and_Validation_in_G
- Slide 6: https://www-sciencedirect-com.tudelft.idm.oclc.org/science/article/pii/S0360132318306437
- Slide 7: https://www.hukseflux.com/
- Slide 8: https://learn.arcgis.com/en/projects/estimate-solar-power-potential/
- Slide 9: https://www.researchgate.net/publication/280933639_Propagation_of_positional_error_in_3D_GIS_estimation_of_the_solar_irradiation_of_building_roofs
- Slide 10: https://3dbag.nl/en/viewer
- Slide 16: https://3dbag.nl/en/download
- Slide 18: https://www-semanticscholar-org.tudelft.idm.oclc.org/paper/Location-Based-R-Tree-and-Grid-Index-for-Nearest-Mon-Than/08343897152db8e1c5b2a543a737759c4f48d67a
- Slide 26: https://3d.bk.tudelft.nl/lod/
- Slide 28: https://www.researchgate.net/publication/228444959_Generalized_Gauss_maps_and_integrals_for_three-component_links_Toward_higher_helicities_for_magnetic_fields_and_fluid_flows_part_II
- Slide 40: https://www.knmi.nl/over-het-knmi/nieuws/knmi-weerstation-in-voorschoten

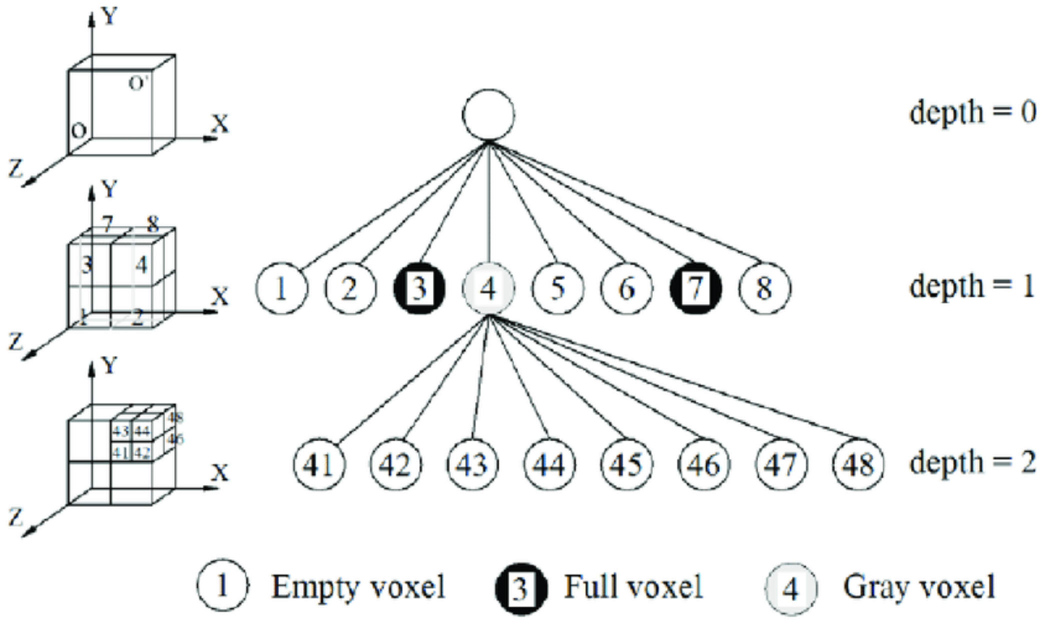
TU Delft

Raster resolution

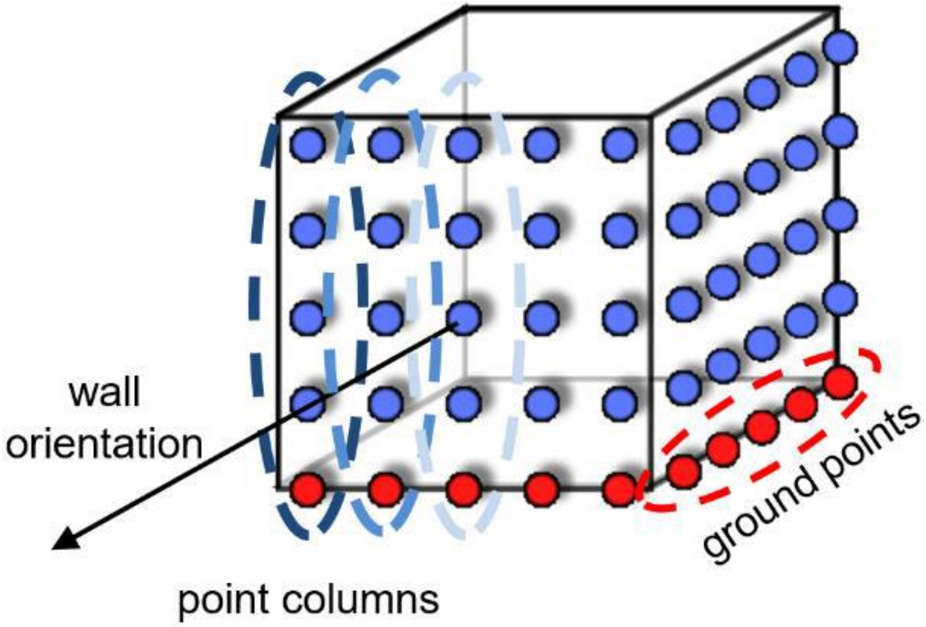


Extra related work

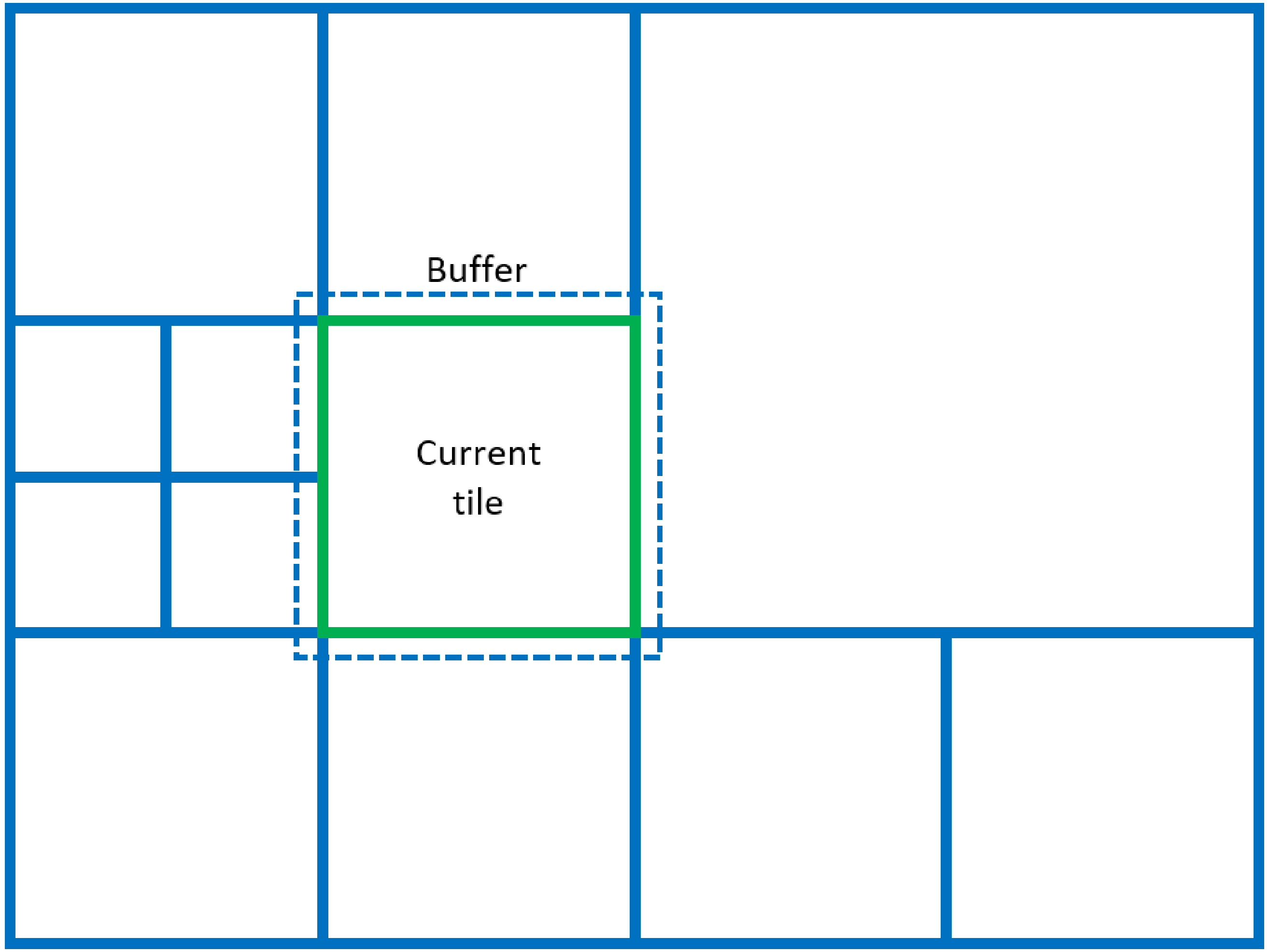
Voxelisation – sparse voxel octree (SVO)



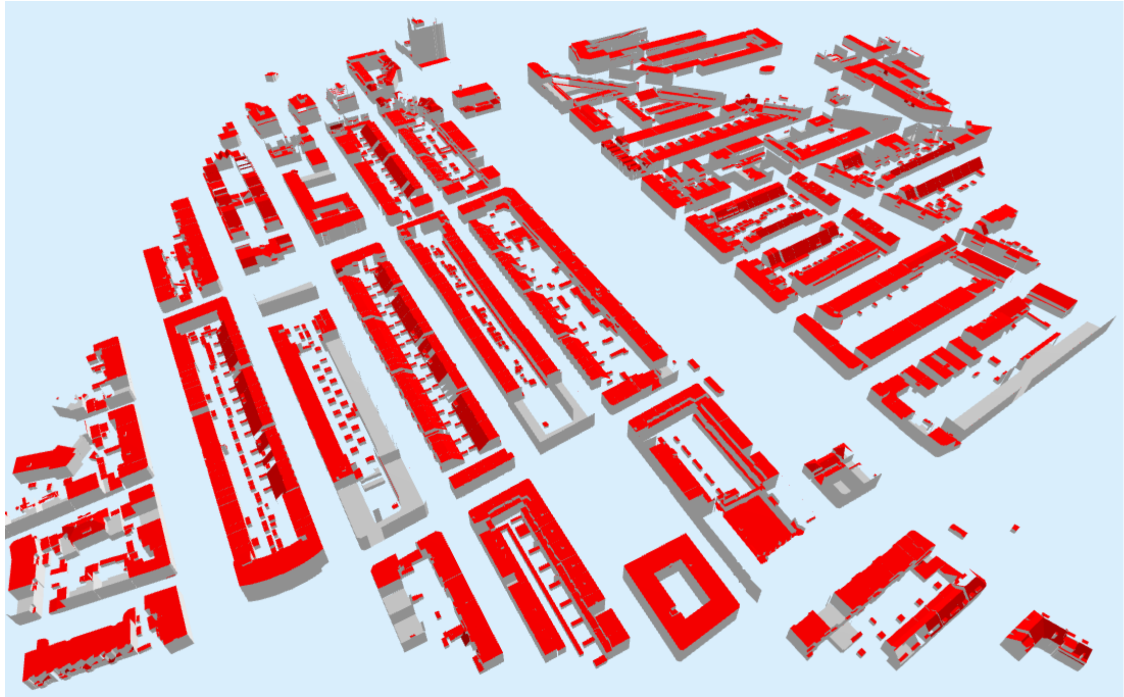
Walls – observer point columns

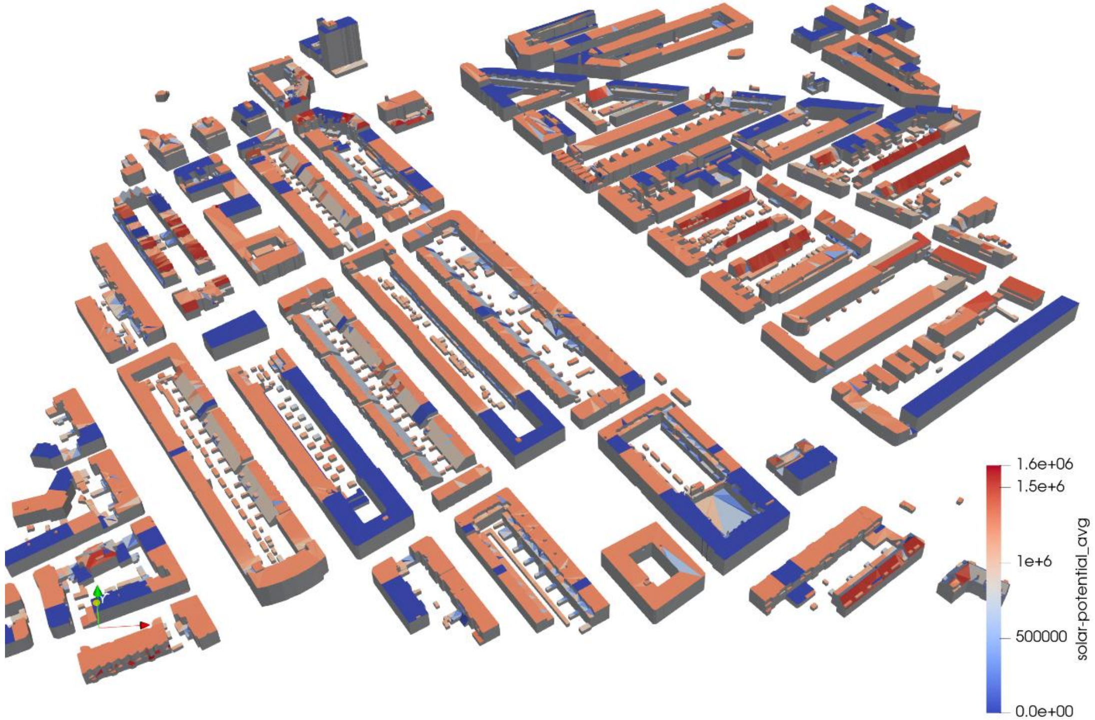


Alternative tile configuration



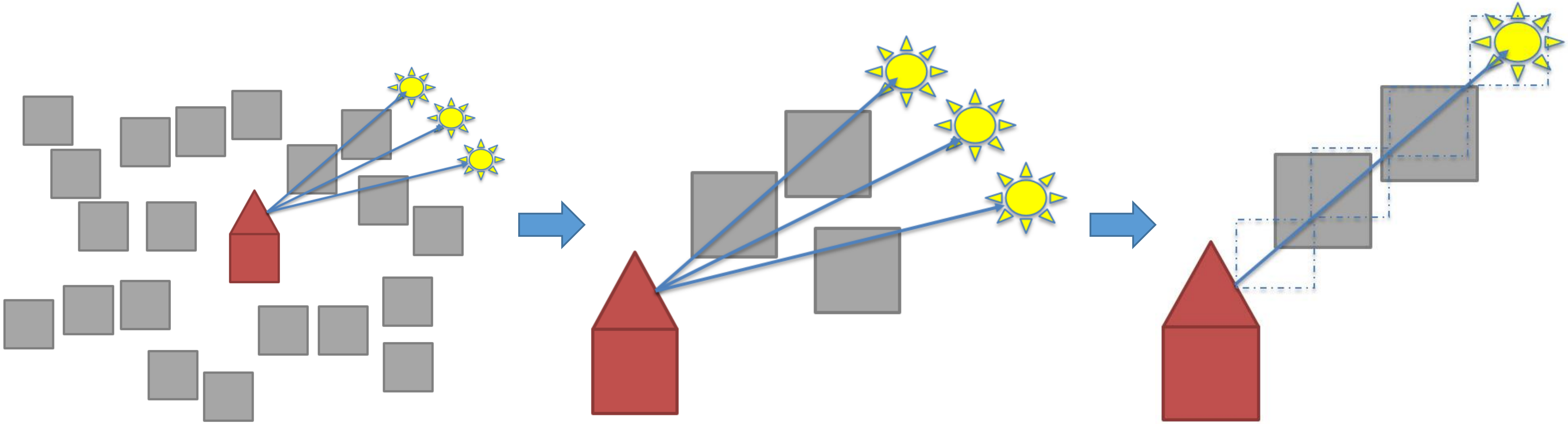
Results - General





TU Delft

Filter neighbouring buildings on the extent of the sun ray



TU Delft