Processing Geodetic Geometry with RADIANCE

Peter Apian-Bennewitz

pab Consulting for Optics & Engineering
info@pab.eu

28th *Radiance* workshop, Lausanne

Motivation for 3D Model of a mid-size Landscape

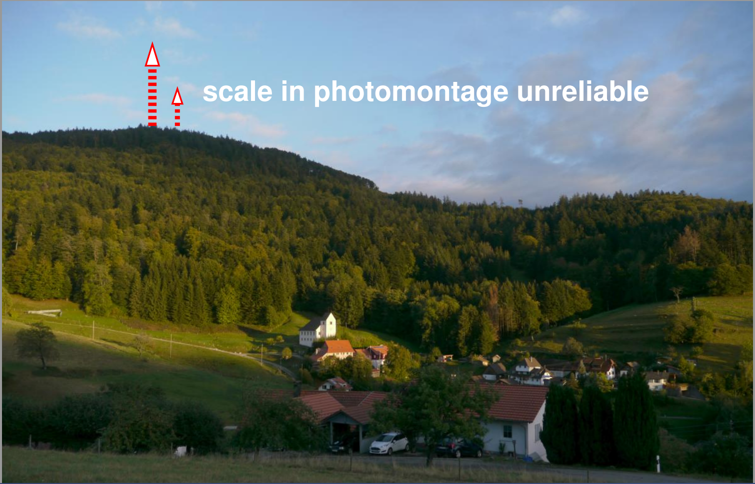
Planned Rotors on Mountain Ridge-line

Motivation for 3D Model of a mid-size Landscape

Planned Rotors on Mountain Ridge-line

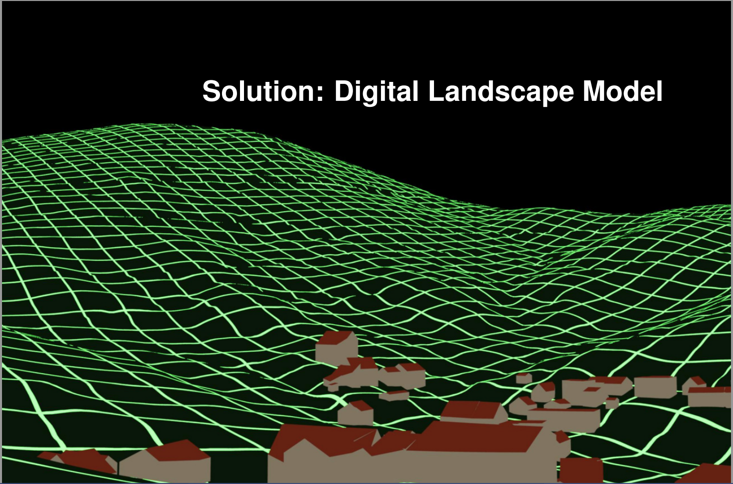
Motivation for 3D Model of a mid-size Landscape

Planned Rotors on Mountain Ridge-line



Motivation for 3D Model of a mid-size Landscape

Planned Rotors on Mountain Ridge-line



Helpful Knowledge Assets

- Basic Geodetic concepts & coordinates
- Geodetic/Geospatial fileformats (as you go along)
- either Geodetic GUI tools,
  or ”UNIX” style: Geodetic commandline-tools + **make**
- how to organise the data processing

The choice of coordinate system for the model

- today, surveying data is given in *Universal Transverse Mercator* (**UTM**) , unit=[m]
  Lausanne: Y=5151575.7402 X= 548543.1772 , zone UTM32

  `https://latlong.info/switzerland/vaud/lausanne#utm`

  Y coordinate: roughly distance from equator
  X coordinate: East/West from reference meridian + offset (”False Easting”)

The choice of coordinate system for the model

- today, surveying data is given in *Universal Transverse Mercator* (**UTM**) , unit=[m]
  Lausanne: Y=5151575.7402 X= 548543.1772 , zone UTM32

  https://latlong.info/switzerland/vaud/lausanne#utm

  Y coordinate: roughly distance from equator
  X coordinate: East/West from reference meridian + offset ("False Easting")

- ways to handle the large offset in coordinates :

The choice of coordinate system for the model

- today, surveying data is given in *Universal Transverse Mercator* (**UTM**) , unit=[m]
  Lausanne: Y=5151575.7402 X= 548543.1772 , zone UTM32

  `https://latlong.info/switzerland/vaud/lausanne#utm`

  Y coordinate: roughly distance from equator
  X coordinate: East/West from reference meridian + offset ("False Easting")

- ways to handle the large offset in coordinates :
  1. simply define a "project" reference point and subtract that offset

The choice of coordinate system for the model

- today, surveying data is given in *Universal Transverse Mercator* (**UTM**) , unit=[m]
  Lausanne: Y=5151575.7402 X= 548543.1772 , zone UTM32

  `https://latlong.info/switzerland/vaud/lausanne#utm`

  Y coordinate: roughly distance from equator
  X coordinate: East/West from reference meridian + offset ("False Easting")

- ways to handle the large offset in coordinates :
  1. simply define a "project" reference point and subtract that offset
  2. convert to a proper local Mercator system: higher local distance precision than option 1

The choice of coordinate system for the model

- today, surveying data is given in *Universal Transverse Mercator* (**UTM**) , unit=[m]
  Lausanne: Y=5151575.7402 X= 548543.1772 , zone UTM32

  `https://latlong.info/switzerland/vaud/lausanne#utm`

  Y coordinate: roughly distance from equator
  X coordinate: East/West from reference meridian + offset (”False Easting”)

- ways to handle the large offset in coordinates :
  1. simply define a ”project” reference point and subtract that offset
  2. convert to a proper local Mercator system: higher local distance precision than option 1
  3. simply work in UTM coordinates (straight forward for import/export)

The choice of coordinate system for the model

- today, surveying data is given in *Universal Transverse Mercator* (**UTM**) , unit=[m]
  Lausanne: Y=5151575.7402 X= 548543.1772 , zone UTM32

  `https://latlong.info/switzerland/vaud/lausanne#utm`

  Y coordinate: roughly distance from equator
  X coordinate: East/West from reference meridian + offset ("False Easting")

- ways to handle the large offset in coordinates :
  1. simply define a "project" reference point and subtract that offset
  2. convert to a proper local Mercator system: higher local distance precision than option 1
  3. simply work in UTM coordinates (straight forward for import/export)

- convert data given in other coordinates to UTM (EPSG:25832)
  e.g. "Web Mercator" coordinates, EPSG:3857 $\rightsquigarrow$ EPSG:25832

- convert height-data `GeoTIFF` format to a **mesh**
  - survey data shipped in ”tiles”: merge, sort & cut
  - area 10km x 12km: 1.920.000.000 height datapoints (25cm resolution)
  - requires adaptive down-sampling:
    core region with higher resolution and surroundings with lower
  - convert GeoTIFF to XYZ file ⇝ **gensurf** ⇝ .obj file ⇝ **obj2mesh** ⇝ mesh-file

- convert ortho-photos `GeoTIFF` (and other data) to **colorpict**
  - prepare tiled original data as above
  - convert GeoTIFF to ”normal graphic format” and to RADIANCE pic format
  - extract corner coordinates from GeoTIFF and built transform for **colorpict**

Georeferenced Vector Formats to RADIANCE

de-facto standard: Esri's SHAPE , proprietary, but well documented

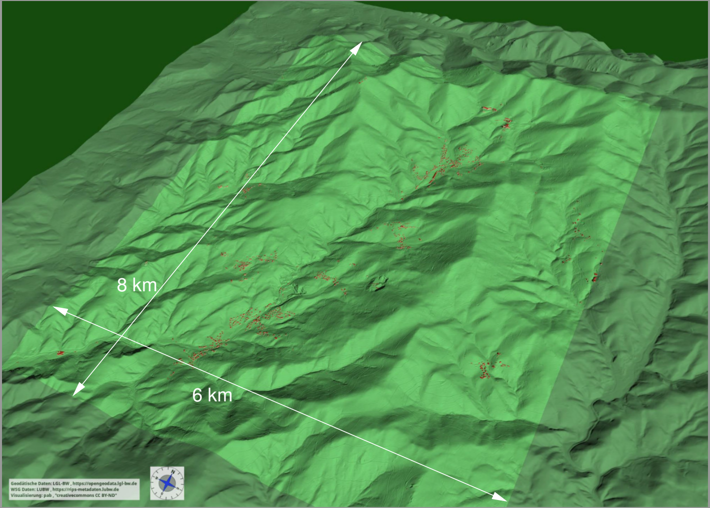
- convert closed polygons to **colorpict**
  - use geodetic tools to rasterise closed polygons, then treat as GeoTIFF
  - problem: limits on how much information is colour-coded in an image
- convert polygons to geometry
  - problem: most survey boundaries are given in 2D UTM coordinates, yet the generated geometry should be placed on the 3D mesh
  - solution: pab **gen_drape**, using **rtrace** to get the Z-coordinate
- placing objects on the 3D mesh
  - manually maintain file with 2D UTM coordinates
  - generate instances with pab **geninsta**, using **rtrace** to get the Z-coordinate
  - optionally ”clusters” of instances

- abstracted buildings from surveying department in CityGML format
- CityGML format read by **blender** $\rightsquigarrow$ .obj file
- **obj2rad** .obj $\rightsquigarrow$ .rad file

RADIANCE scene-version and rendering management

- intention: visualise combinations of selected data-sets on the mesh
- strategy: one-octree-per-data-set-combination
- method: **gmake** and **m4** macro-processor working on `scene.rad.m4` et al
- currently 1283 lines in 7 makefiles, 17 octree variants, 42 view points, 134 pics
- advantage: changes in geometry ”cascade”-through automatically
- keep all large scratch files on fast extra storage, (e.g. 300MB/s SSD SATA/M2 drives) yet the loading times of octrees, mesh and patterns, becomes prominent.

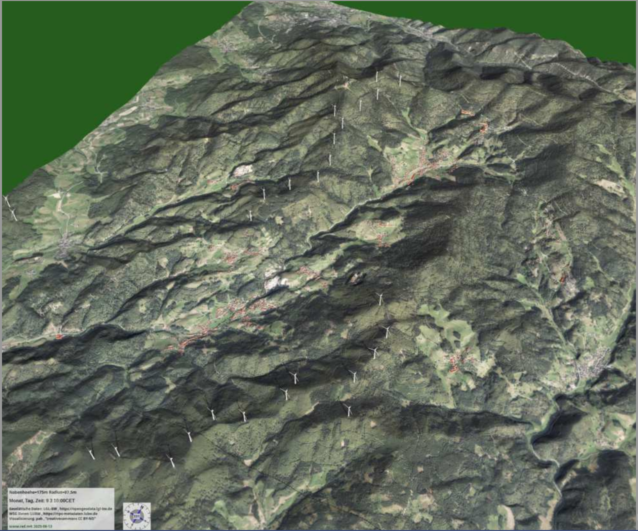
Examples renderings



surface model + buildings

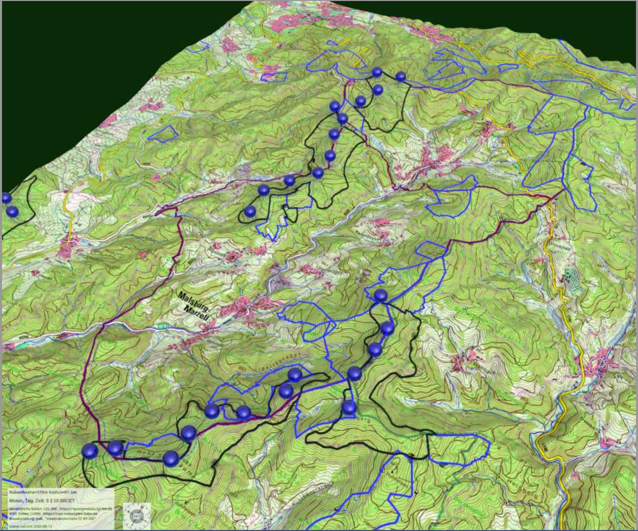
Examples renderings

ortho photo



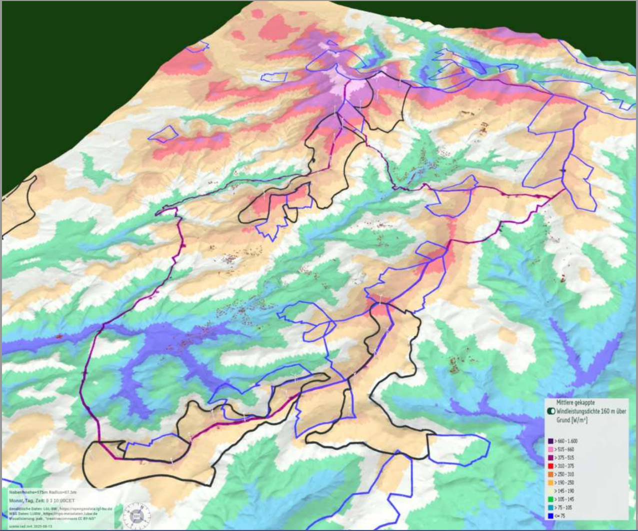
Examples renderings

topographic map plus boundaries



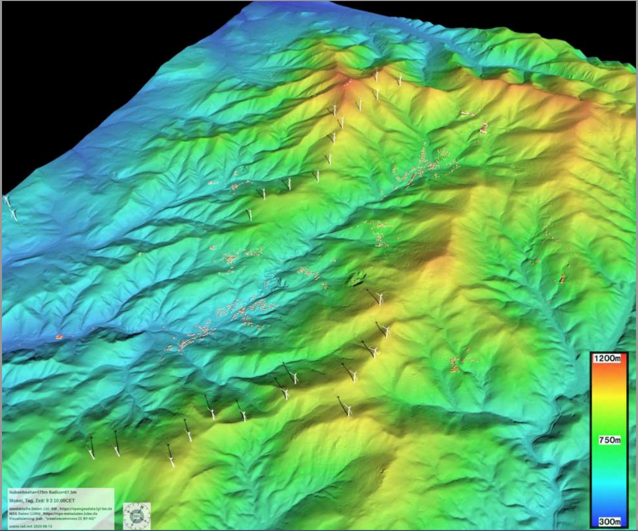
Examples renderings

CFD energy density

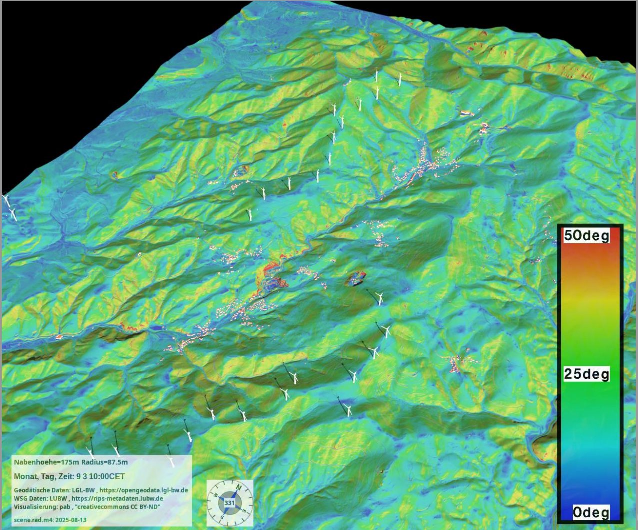


Examples renderings

function file: height

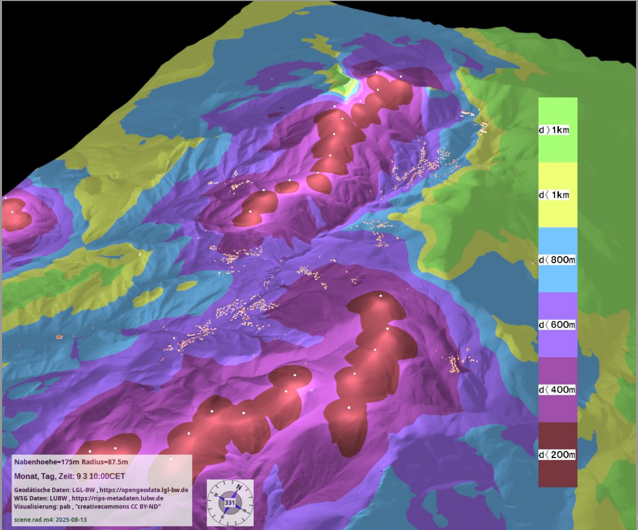


function file: surface angle

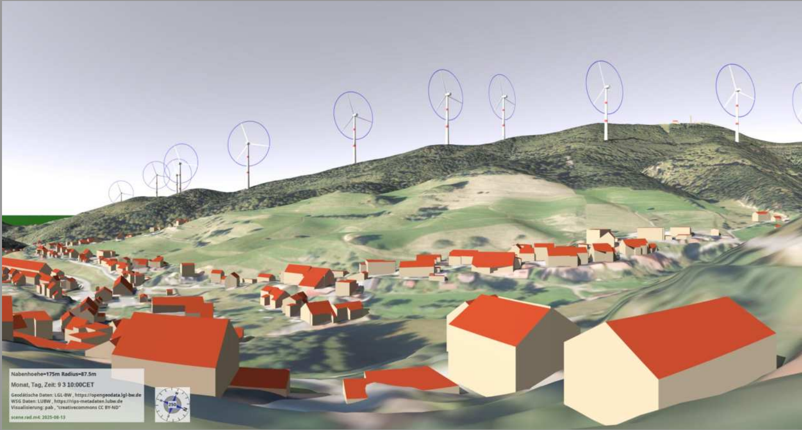


Examples renderings

function file: effective distance



resident's perspectives



Rotor Daily Cumulative Shading Analysis

aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball

Rotor Daily Cumulative Shading Analysis

aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball
- step sun in N-minute intervals through sun-rise to sun-set

Rotor Daily Cumulative Shading Analysis

aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball
- step sun in N-minute intervals through sun-rise to sun-set
- for each sun position: render irradiance picture
  identify areas where shadow is cast by balls

Rotor Daily Cumulative Shading Analysis

aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball
- step sun in N-minute intervals through sun-rise to sun-set
- for each sun position: render irradiance picture
  identify areas where shadow is cast by balls
- for each day: sum up shaded areas with an N minutes weight

Rotor Daily Cumulative Shading Analysis

aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball
- step sun in N-minute intervals through sun-rise to sun-set
- for each sun position: render irradiance picture
  identify areas where shadow is cast by balls
- for each day: sum up shaded areas with an N minutes weight
- generate **falsecolor** image and combine with topographic background

Rotor Daily Cumulative Shading Analysis

aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball
- step sun in N-minute intervals through sun-rise to sun-set
- for each sun position: render irradiance picture
  identify areas where shadow is cast by balls
- for each day: sum up shaded areas with an N minutes weight
- generate **falsecolor** image and combine with topographic background
- repeat for a set of days, January ... June

Rotor Daily Cumulative Shading Analysis

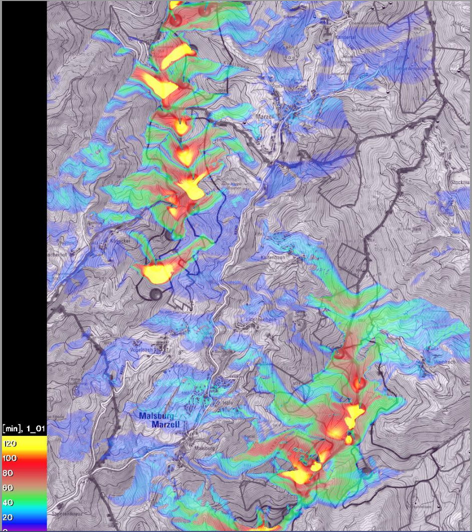
aim: Calculate shadowing by rotor area (standard procedure for ”flicker”)

our implementation with RADIANCE :

- worst case = rotor points to the sun at all times $\rightsquigarrow$ model rotor volume as ”glass” ball
- step sun in N-minute intervals through sun-rise to sun-set
- for each sun position: render irradiance picture
  identify areas where shadow is cast by balls
- for each day: sum up shaded areas with an N minutes weight
- generate **falsecolor** image and combine with topographic background
- repeat for a set of days, January ... June
- advantages: Topography is taken into account,
  ”static” shadowing (e.g. by tower) is not counted.

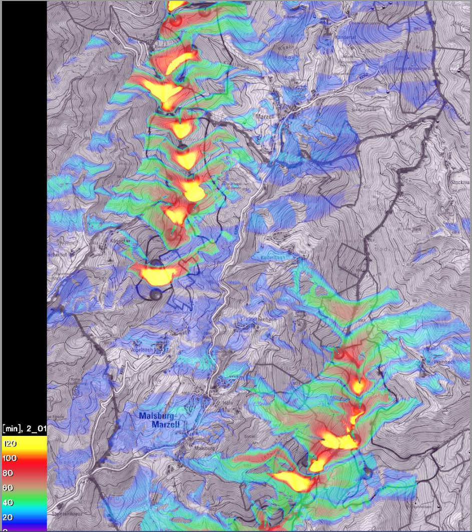
Rotor Daily Cumulative Shading Analysis, Examples

January 1st



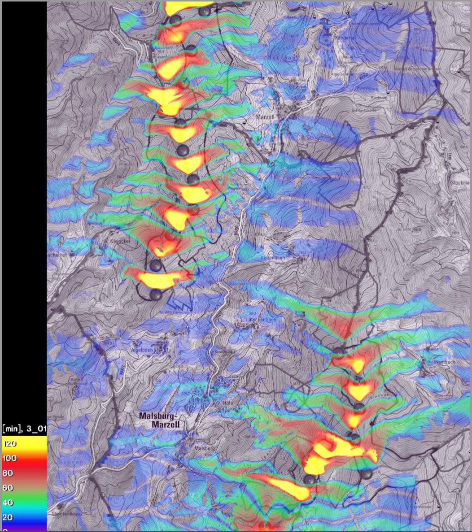
Rotor Daily Cumulative Shading Analysis, Examples

February 1st



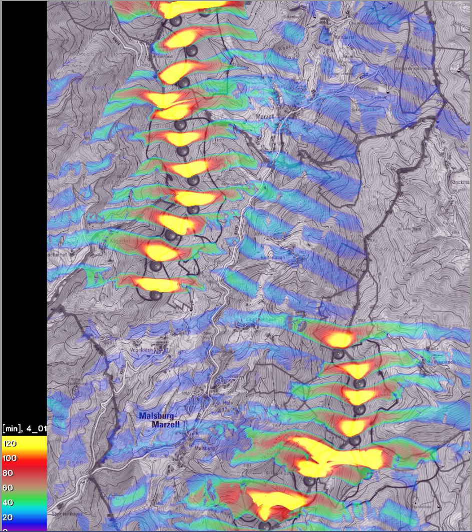
Rotor Daily Cumulative Shading Analysis, Examples

March 1st



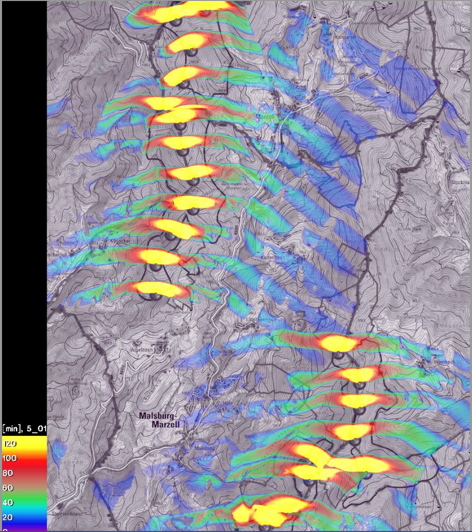
Rotor Daily Cumulative Shading Analysis, Examples

April 1st



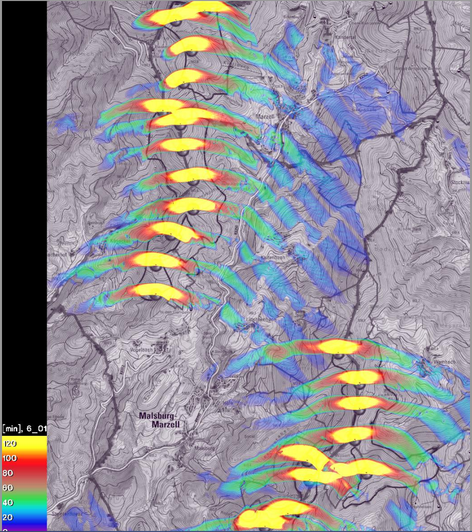
Rotor Daily Cumulative Shading Analysis, Examples

May 1st

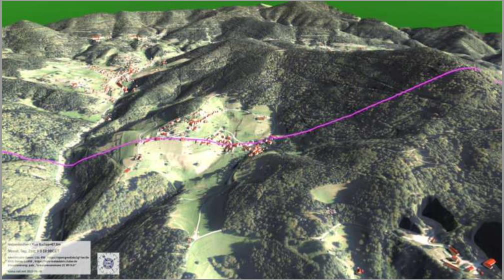


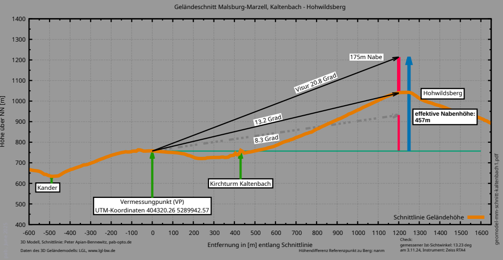
Rotor Daily Cumulative Shading Analysis, Examples

June 1st



advantage of 2D cuts: model Altitude angles can be verified on-site with precision $< 1'$





Lessons Learned & Suggestions

- **rtrace** is extremely helpful in scene-building and analysis
- more decimals in view files for position (printf format ”%.Xg”)
- **gensurf** flip added (to make **glow**-mesh work), thanks !
- suggestions: automatic Level-Of-Detail handling when loading octrees based on distance ?

... thanks

- Thanks to Greg for promptly fixing 3 bugs in mesh and polygon intersection code
- And thanks to you for your interest and attention, Happy rendering !

`$RCSfile: pab-geospatial-model.tex,v $ $Revision: 1.17 $ $Date: 2025/08/24 20:32:47 $`
contact info@pab.eu prior to commercial use.

compiled using LaTeX *beamer* class