|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ESA STUDY CONTRACT REPORT**  **Deliverable 6 under WP4:**  **Roadmap report on implementation of the toolbox** | | | | |
| ESA Contract No:  4000117034/16/NL/NDe | SUBJECT: **SURGE**: Simulating the cooling effect of urban greenery based on solar radiation modelling and a new generation of ESA sensors | | | CONTRACTOR:  Pavol Jozef Šafárik University in Košice, Institute of Geography |
| \* ESA CR( )No: |  | No. of Volumes:. 1  This is Volume No: 1.0 | | CONTRACTOR’S REFERENCE: |
| ABSTRACT:  This report presents a description of the roadmap for a specialized software toolbox called r.lst implemented in open-source GRASS GIS. This report summarizes the findings and methods developed within this project. The roadmap suggest a 2-year period for the software development and implementation divided into 3 work packages with several subtasks. These subtasks include software design and coding, software testing and verification using validation datasets and implementation in GRASS GIS. We suggest further development based on the Tangible Landscape concept that should improve the communication and data interaction between user and model leading to increased applicability of the model in urban planning and city management. | | | | |
| The work described in this report was done under ESA Contract. Responsibility for the contents resides in the author or organisation that prepared it. | | | | |
| Names of authors: Jaroslav Hofierka | | | | |
| \*\* NAME OF ESA STUDY MANAGER:  DIV:  DIRECTORATE: | | | \*\* ESA BUDGET HEADING: | |

# 1 Introduction

## Contractual

This document has been issued by Institute of Geography, P.J. Šafárik University in Košice for European Space Agency under contract Nr. 4000117034/16/NL/NDe titled “Simulating the cooling effect of urban greenery based on solar radiation modelling and a new generation of ESA sensors (acronym SURGE)”.

## Purpose of the Document

This document presents a roadmap for a software toolbox summarizing the findings and developed methods within this project. A product roadmap is a useful tool for software development and implementation because it acts as a strategic plan that describes the likely development of the software over the next period of time. It supports the product’s goals, vision, and priorities, and helps to keep their developer or stakeholders aligned. The specific goal of this report is to propose further development and implementation steps for the software toolbox which will assess cooling effects of urban greenery supported by various data sets including Sentinel 2 imagery.

## 1.3 Motivation

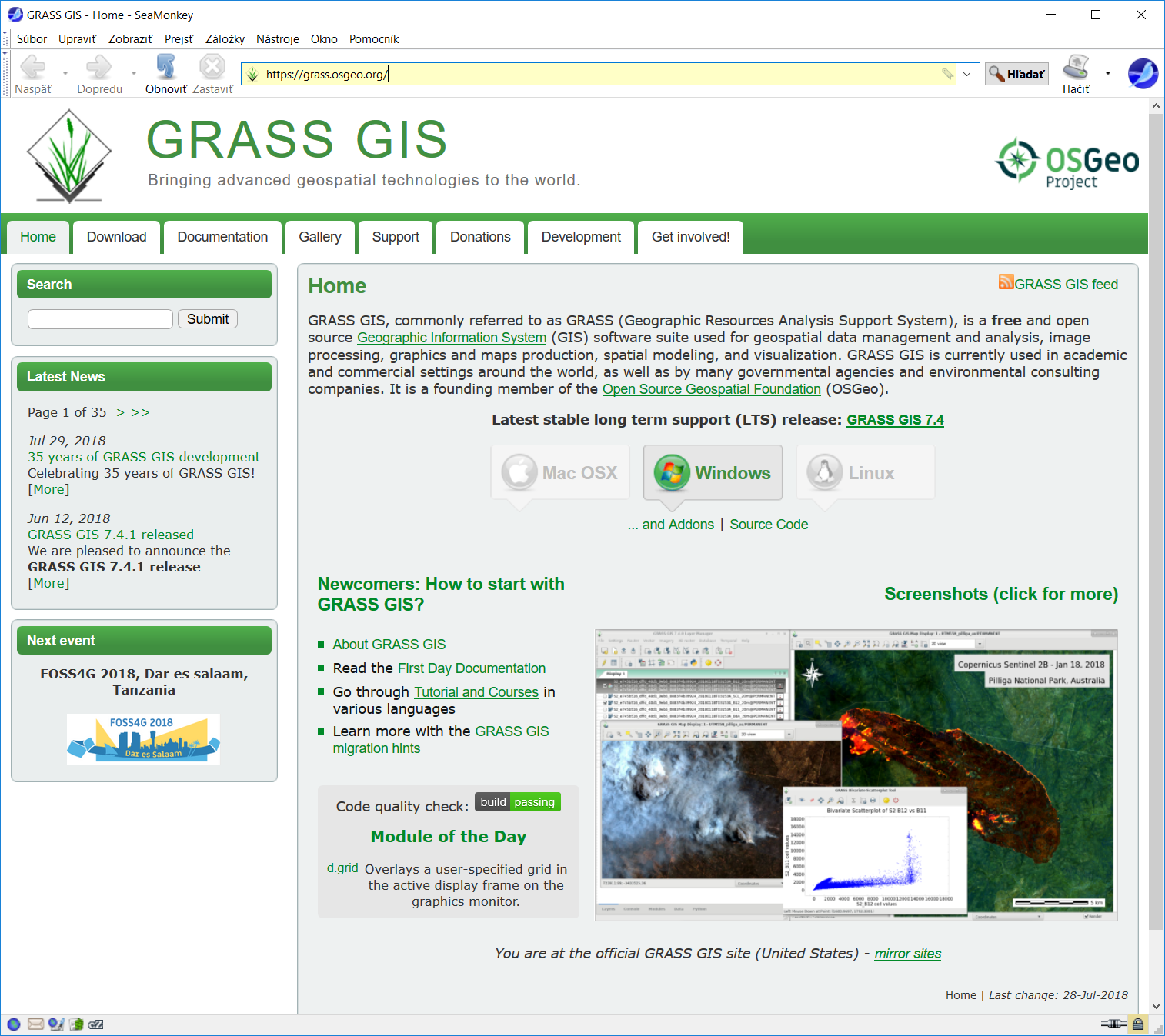
In this report, we propose a roadmap for a specialized software simulating land surface temperature based on publicly available data such as the products of the ESA’s Sentinel 2missions, various municipal, national data or even Open Street Map data. Using this software, we can assess the effects of urban greenery (especially trees) on land surface temperature thus also directly influencing the air temperature in urban areas leading to a creation of urban heat islands (UHIs). To make it even more accessible for public, we decided to propose an open-source software module - toolbox for GRASS GIS (grass.osgeo.org) called r.lst (raster-based module for land surface temperature geospatial modelling). The mitigation of UHIs requires understanding of the factors affecting the land surface temperature during hot sunny days as well as proper tools to simulate various scenarios for urban planning and city management. This software toolbox consists of various software modules implemented in the open-source GRASS GIS. The proposed toolbox will be available for users as a single GRASS GIS module with one, integrated user interface, software and data documentation. The advantage of this approach is the open-source software and free data concept beneficial for the citizens of the cities and countries as well as stimulating further development under free/open-source software licence. Therefore, the implementation phase includes the code submission to GRASS GIS distribution that assures a wider software validation and later sustainable and continuous development and improvement. Our previous active involvement and experience in the development of other GRASS GIS modules such as r.sun and v.sun, and others supports this decision (Hofierka, 1997), (Šúri and Hofierka, 2004), (Hofierka and Zlocha, 2012).

## 1.4 GRASS GIS

# GRASS GIS (https://grass.osgeo.org/) is a general purpose, cross-platform, open-source geographic information system with raster, vector, 3D raster and image processing capabilities (Neteler and Mitasova 2008). It includes more than 400 modules for managing and analysing geographical data and many more user contributed modules available in the add-on repository. GRASS GIS modules can be run using a command-line interface (CLI) or a native graphical user interface (GUI) called wxGUI which offers a seamless combination of GUI and CLI native to the operating system. Modules are organized based on the data type they handle and they follow GRASS GIS-specific naming conventions such as r.\* for raster-based operations (example: r.sun) or v.\* for vector-based operations (v.sun), etc.. Each module has a set of defined options and flags which are used to specify inputs, outputs, or different module settings. Most core modules using GRASS GIS C library are written in C for performance and portability reasons. Other modules and user scripts are written in Python using the GRASS GIS Python Scripting Library which provides a Python interface to GRASS GIS modules.

GRASS GIS software can be downloaded freely from its website (Fig. 1). The download web page offers easy to install binary packages for GNU/Linux, Mac OS X, and Microsoft Windows as well as the source code. The GRASS GIS website also provides additional documentation including manual pages, tutorials, information about the externally developed modules (add-ons) and various publications. Support for developers and users is provided by several mailing lists. The following tutorial provides a quick introduction to GRASS GIS: http://grass.osgeo.org/grass70/manuals/helptext.html.

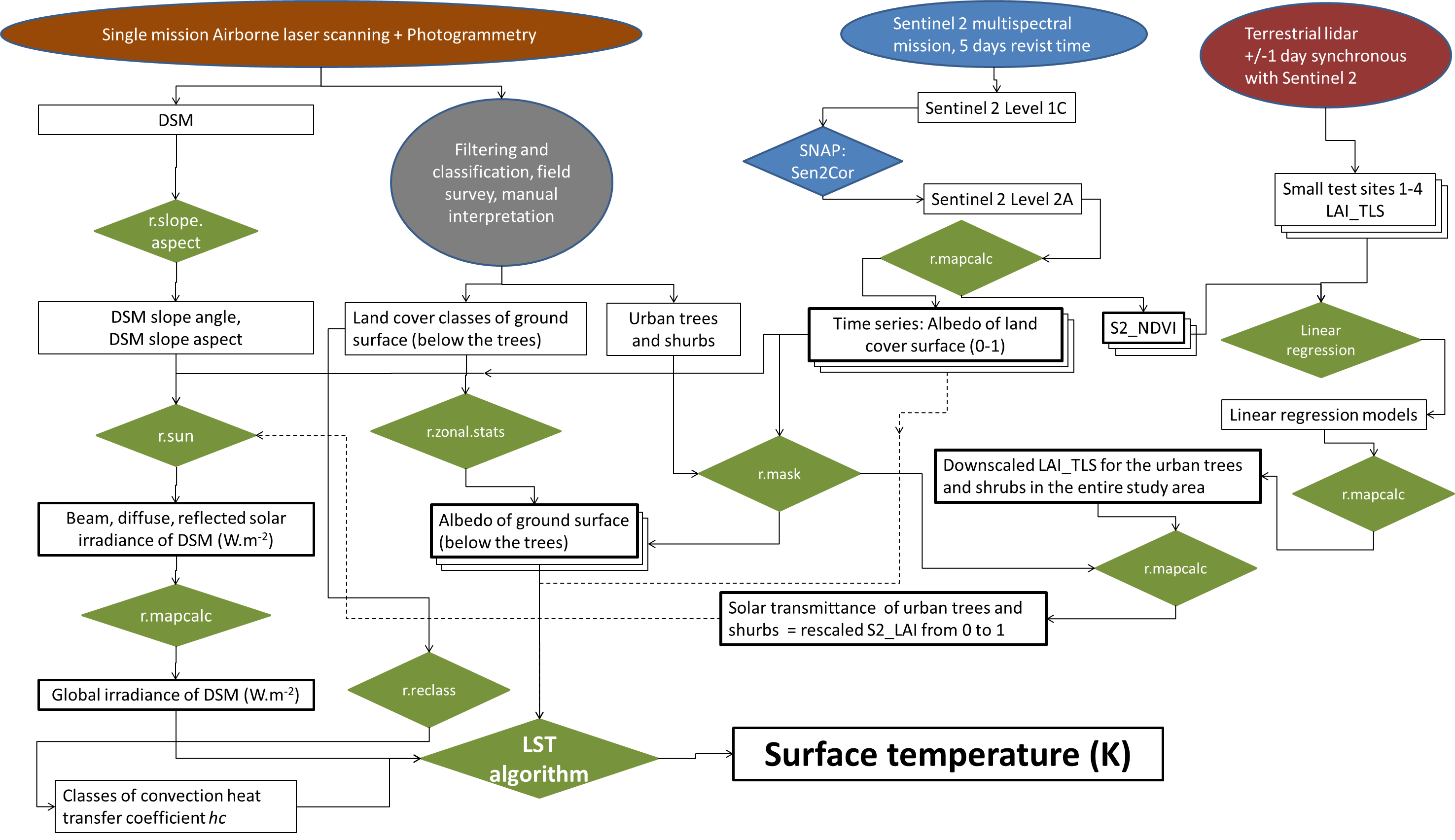
In this project we have suggested to create a specific GRASS GIS module that will help to simulate the land surface temperature based on publicly available data such as Open Street Map, national geographic databases or even municipal data available online (example: http://webgis.presov.sk/). The suggested module called r.lst will integrate a few other GRASS GIS modules (Fig. 2) together with the software code developed within this project (the Stefan-Boltzman model described in the report SURGE\_D5\_ALGOSTRUC) in a coherent way, so the user can get results based on interaction via one user interface.



*Figure 1. Official web page of GRASS GIS.*

# 2 Roadmap

The proposed software toolbox roadmap is based on our developed methodology for land surface temperature modelling. The methodology, data and results are described in the Report on definition of algorithmic structure of the toolbox (Deliverable 5, SURGE\_D5\_ALGOSTRUC). The key components of the proposed toolbox are depicted in Fig. 2 showing a schematic workflow of the modelling.



*Figure 2. Schematic workflow for the land surface temperature modelling. Ellipses denote the data acquisition methods, rectangles denote the data layers or products, pinacoids denote processes (calculations). Green colour marks the use of GRASS GIS modules for specific calculations.*

The software toolbox, r.lst, will be in the form of Python script that will call the required GRASS GIS modules in a coherent way using specified input data prepared outside of the toolbox. GRASS GIS includes the GRASS Python Scripting Library that makes development of new Python modules much easier (Zambelli et al., 2013). GRASS GIS 7.2.0+ comes with a "Simple Python Editor" which enables users to author Python scripts directly in GRASS GIS GUI. User can also run the script easily in GRASS GIS environment with all the dependencies loaded. This Python editor comes with several examples, templates, and links to documentation. The r.lst module will be developed for final submission and implementation in GRASS GIS. First as an "add-on" module in a repository outside of the standard GRASS GIS distribution but fully available for all GRASS GIS users via its web site. Later on as a standard module embedded in one of the future GRASS GIS release versions. The toolbox development and implementation plan is depicted in Fig. 3.



*Figure 3. The r.lst roadmap chart.*

*WP1 - Design and coding*

Module structure and design - This subtask analyses the results of the feasibility study and allocates a manpower capacity to perform all the tasks. Analyses a current programing environment of GRASS GIS and recommends final design issues (e.g., the use of Python vs. shell script, modification of existing r.sun/v.sun modules, solves the problems with I/O operations, etc.).

Coding - Actual coding of the module with 3 software versions delivered: alpha version that is a working version with almost a full functionality that, however, probably still has lots of bugs that needs to be worked out. Beta version is near completion. All of the features are working and the software can be open up to external testers for testing. Code bugs that will be discovered and had not shown up in the alpha testing will be fixed. Final version is the last stage of the code development with software supposed to be complete and finished.

*WP2 - Testing and verification*

Test data preparation - Based on the Report on definition of algorithmic structure of the toolbox (Deliverable 5), the necessary data required for comprehensive software testing will be prepared. This will include a wide range of data values and situations.

Code testing - The tests will be performed using sample data with a wide range of input parameters to test the stability and sensitivity of outputs. The obvious software bugs will be documented, suspicious software behaviour described to perform further analysis by programmers.

Output verification - data simulated in the software will be verified using real world data, including field measurements. We expect an extensive use of thermal sensors within the study area, calibration of the model and verification of results.

*WP3 - Implementation*

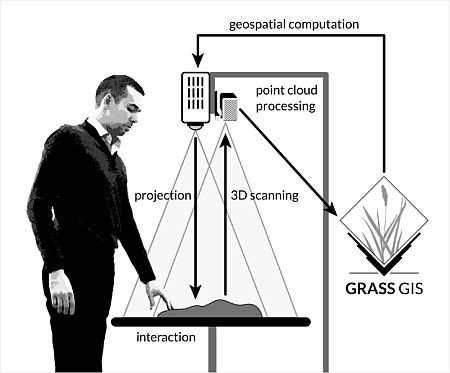
Documentation - The documentation will be done in two parts: a standard manual page for the r.lst module (usually a few pages describing the aim of the module, input, output and operation behaviour of the module and a few references for further reading) and a comprehensive pdf documentation describing scientific background of the simulation supplemented by scientific papers when published. We expect 1-2 scientific papers to be published over the next year or two with main findings of this project.

GRASS GIS module submission - the finalised module will be submitted to the add-on repository of GRASS GIS. The submission will be done in two steps: Release candidate version 1, which is expected to be bug-free. If any problem or suggestion from a wider user community arises during the use of the software, these will be fixed or implemented until Release Candidate version 2. Soon after a relatively short period of time (3-4 months), a final product will be released for public use.

The suggested software toolbox will be realised using the following manpower resources: 2 programmers (WP1), 3 researchers for testing and validation (WP3) and 1 programmer-GRASS developer for GRASS GIS implementation (WP3).

## Future development

Recently, the Tangible Landscape concept was successfully tested also for urban environment (Petrasova et al., 2018). The idea behind this concept is to use a physical model of the landscape to interact with or change it by hands to see real-time changes simulated by models implemented in GRASS GIS. The results of the models are represented by colour draped over the model (Fig. 4). For example, Petrasova et al. (2018) used the output of the r.sun module to simulate solar irradiation distribution over small urban area segment with buildings (Fig. 5). This concept provides new form of geospatial data interaction and better understanding of spatial relations. The complex interactions of various factors within UHI phenomena can be also better understood by city managers or general public using this tool. For example, city planners can analyse different scenarios for city development plans, emergency situations during very hot summer days and use of quick protection measures such as sprinkling the city roads. The technology can be also useful in schools to explain the UHI phenomenon and how to mitigate it.



*Figure 4. Tangible Landscape concept.*



*Figure 5. Solar irradiation computed by r.sun in GRASS GIS for two different building setting (a, b) and visualized using the Tangible Landscape physical model (adapted from Petrasova et al., 2018)*

# Conclusions

The presented report provides a roadmap for a future GRASS GIS module that will simulate land surface temperature using the methodology developed in this project. The r.lst module will use the GRASS Python Scripting Library and a set of other GRASS GIS modules as well as our implementation of the Stefan Boltzmann land surface temperature model described in the deliverable 5 SURGE\_D5\_ALGOSTRUCT. The proposed toolbox will be tested and validated using test data verified using field data taken from a set of thermal sensors on ground or from UAVs. The r.lst module will be submitted to GRASS GIS in several stages. The open-source concept of GRASS GIS will stimulate further development of the module and stimulate a wider use of the module including the data form ESA sensors such as Sentinel 2.

# References

Hofierka, J. (1997). Direct solar radiation modelling within an open GIS environment. In Proceedings of the 1997 Joint European GI Conference, Vienna, Austria, 575–584.

Hofierka, J., Zlocha, M. (2012). A New Solar Radiation Model for 3-D City Models. Transactions in GIS, 16, pp. 681-690.

Neteler, M., Mitasova, H., (2008). Open Source GIS: A GRASS GIS Approach. Third Edition. The International Series in Engineering and Computer Science, Volume 773, Springer, New York

Petrasova, A., Harmon, B., Petras, V., Tabrizian, P., & Mitasova, H. (2018). Tangible Modeling with Open Source GIS. Cham, Springer.

Šúri, M., Hofierka, J. (2004). A new GIS-based solar radiation model and its application to photovoltaic assessments. Transactions in GIS 8, 175–190.

Zambelli, P., Gebbert, S., Ciolli, M. (2013). Pygrass: An Object Oriented Python Application Programming Interface (API) for Geographic Resources Analysis Support System (GRASS) Geographic Information System (GIS). ISPRS International Journal of Geo-Information 2, 201–219.