Implementing a low cost meteorological data network

Maximilian Arbeiter[1] Albrecht Gebhardt[2] Gunter Spöck [3]
Glenda Garcia-Santos[4]

[1]maximilian.arbeiter@aau.at

[2]albrecht.gebhardt@aau.at

[3]gunter.spoeck@aau.at

Department of Statistics,
University Klagenfurt

[4]glenda.garciasantos@aau.at

Department of Geography,
University Klagenfurt

SimStat 2019

Overview I

- Motivation and demands
- Description of hardware
- Software overview
- Data presentation
- Interpolated data

Motivation I

Particle dispersion calculations

Simulations for particle dispersion models needs some input data, among other e.g.

- a digital elevation model (DEM),
- information about land use,
- data for strength and directions of wind
- and maybe other meteorological data

Availability of data

For a given area of interest:

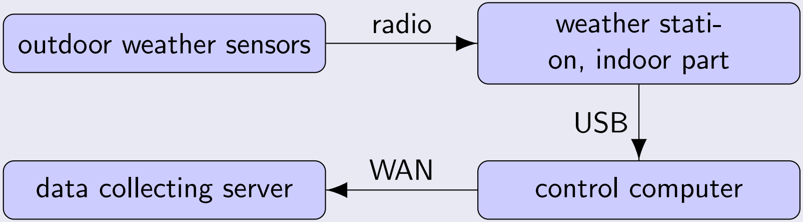
- more or less simple to find: DEM data, other geoinfomation.
- hard to get: detailed meteorological data, specifically live data for arbitrary locations

Motivation II

Resort out of lack of meteorological data situation:

Build your own weather sensor network!

Schematic overview



Demands

A working setup should fulfill the following demands:

- cheap equipment (to allow many stations)
- easy way of connecting sensors, stations and data collecting server,
- small footprint of devices,
- easy way of servicing.

Hardware I

Weather station

Demands:

1. network connection between outside sensors and indoor part, reducing the need of wires for installation.
2. battery powered outdoor part, for ease of installation.
3. possibility of constant data access from station computer e.g. via USB.

There are many consumer grade cheap weather stations satisfying point 1 and 2 but only a few have USB connectivity (point 3).

Hardware II

WH1080 weather station

We choose WH1080/WH3080[a] type stations, their main properties are (see e.g. Conrad Electronic SE [2014]):

- measures temperature, humidity, wind speed and direction (WH3080 only: also luminosity and UV index).
- outdoor part is driven by solar powered rechargeable batteries, reducing maintenance effort.
- 868 MHz radio connection outdoor part $\rightarrow$ indoor part.
- indoor station has USB connection and can also draw power via USB (This gives the possibility to restart it when leaving out its batteries).
- Also contains a DCF77 radio clock.

[a] sold under different brands, e.g. Conrad (`https://www.conrad.at`), originally produced by Fine Offset Electronics (`https://www.foshk.com`)

Hardware III

Control computer

Demands:

1. cheap, small, power saving.
2. WLAN and/or Ethernet network options, USB Port(s) available

Most Raspberry Pi type single board computers will work. (We choose Raspberry Pi 3b at that time, but Orange Pi, Banana Pi etc. would also be ok).
Operating System: Raspian (or Armbian), both Debian based, will work.
If at the site no LAN or WLAN access can be used: WAN modem (USB) needed.

Hardware IV

Server for data collection and presentation

Demands:

1. Webserver with PHP
2. SQL Database

Any LAMP (Linux/Apache/Mysql/PHP) server will work, again even a Raspberry Pi (as we choose) will be sufficient.

Software I

Software

Important note: The USB protocol for data access for WH1080 stations is already reverse engineered and free software for reading is available.
It can be divided into

- a client part running on the Raspberry Pi control computers (frewe-rpi) and
- a server part running on the LAMP server (frewe-server).

Base of our code is “Freetz Weather”, originally written by Alexey Ozerov to run on AVM Fritzbox ADSL routers connected to WH1080 stations (and itself based on earlier work for WH1080 stations), License: GPL (see Alexey Ozerov [2011])

Software II

frewe-rpi

Contains

- the USB reading part of “Freetz Weather”: frewe-client.
- some helper scripts to monitor and restart the frewe-client by power cycling the USB controller of the Raspberry Pi via hub-ctrl[a].

[a] `https://github.com/codazoda/hub-ctrl.c/wiki/Raspberry-Pi`

Software III

frewe-server

Contains

- a PHP script for uploads by the clients (secured via a pre-shared secret and HTTPS), based on work already available in “Freetz Weather”. Uploads are done at (configurable) 10 minutes intervals.
- a new implementation for presenting the collected data on the web (using plot.ly and d3.js libraries for interactive weather charts).

Needs: Apache webserver with PHP and MySql.

Software IV

Our changes to original “Freetz Weather”

can be summarized to

- improving outlier detection,
- rewrite to use HTTPS for uploads,
- new and interactive data presentation.

Available at
`https://gitlab.aau.at/agebhard/frewe-rpi`
and
`https://gitlab.aau.at/agebhard/frewe-server`

Software V

Connections from and to the stations

As mentioned before the stations have network connectivity via

- LAN,
- WLAN or
- WAN (2G/3G)

This is sufficient for data uploads, not for remote administration! An easy way to make the stations reachable via SSH is to use a VPN (We use OpenVPN[a]).

[a] `https://openvpn.net/community/`

Data

Data quality and problems

WH1080 stations are consumer-grade hardware, so you should be prepared for

- false readings
- USB read timeouts
- lost radio connection to outdoor part

Counter measures

- detect erroneous states and reset the station automatically via power cycling the USB bus,
- detect and clean out implausible spikes,
- apply median filters.

Presentation of data I

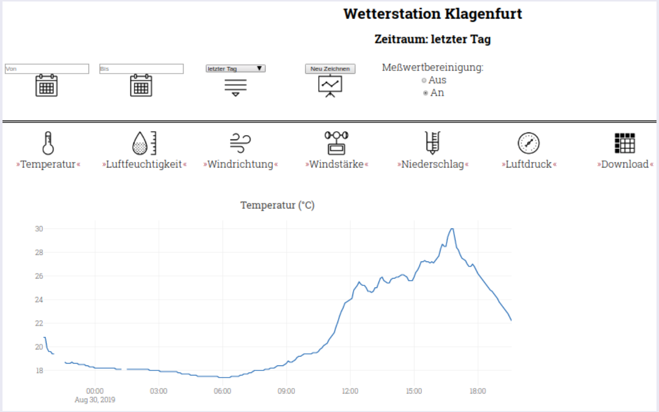
Web interface

The frewe-server implementation contains plots for

- temperature
- dew point temperature
- humidity
- wind speed
- wind direction
- rain
- rain sum
- air pressure
- luminosity (WH3080 only)
- UV index (WH3080 only)

Presentation of data II

Screenshots of frewe-server: navigation and temperature



Wetterstation Klagenfurt

Zeitraum: letzter Tag

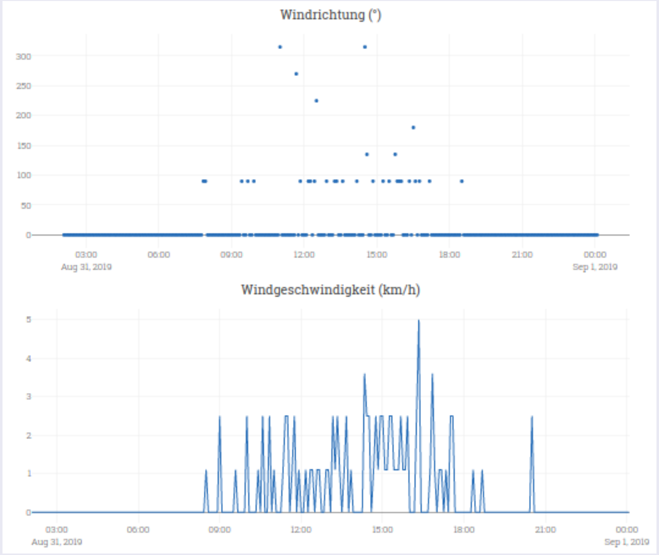
Von | Bis | letzter Tag | Neu Zeichnen

Meßwertbereinigung:
○ Aus
[✓ ● An]

»Temperatur« »Luftfeuchtigkeit« »Windrichtung« »Windstärke« »Niederschlag« »Luftdruck« »Download«

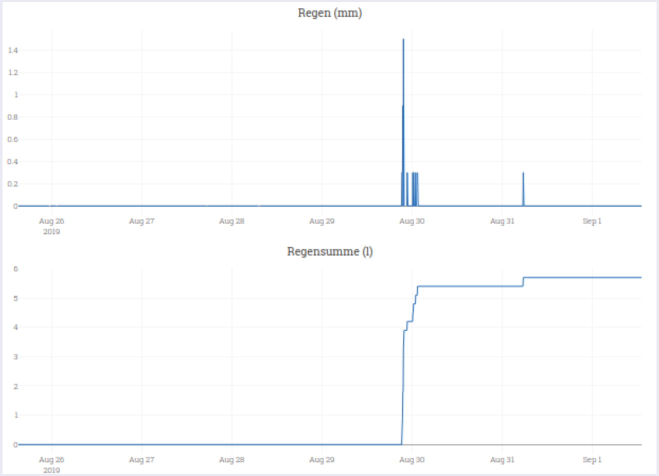
Presentation of data III

Screenshots of frewe-server: wind data



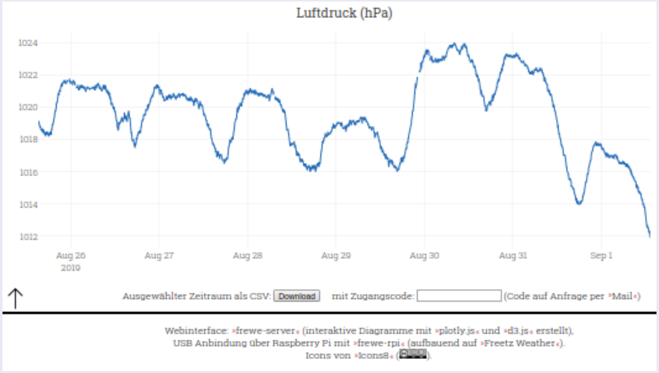
Presentation of data IV

Screenshots of frewe-server: rain data



Presentation of data V

Screenshots of frewe-server: air pressure and CSV download



Ausgewählter Zeitraum als CSV: Download mit Zugangscode: (Code auf Anfrage per »Mail«)

Webinterface: »frewe-server« (interaktive Diagramme mit »plotly.js« und »d3.js« erstellt),
USB Anbindung über Raspberry Pi mit »frewe-rpi« (aufbauend auf »Freetz Weather«).
Icons von »Icons8« ( ).

Presentation of data VI

Interaction

is based on plot.ly[a], a graphing library in JavaScript. Possible interactions are

- zoom in/out,
- scroll left/right after zoom in,
- pick values from curve.

Internal data handling is done via d3.js[b], a JavaScript library for data driven documents.

[a] `https://plot.ly/javascript/`
[b] `https://d3js.org/`

Station networks I

Implementations

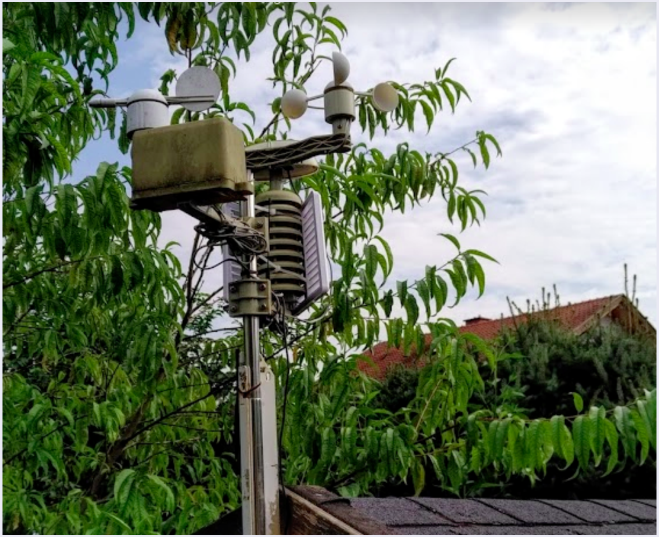
- In cooperation with the citizens' initiative "Rettet das Görtschitztal"[a] 13 stations in central carinthia have been installed.
- Another network with stations in Klagenfurt is also growing within the scope of the project crowdmeteodata[b].

Estimated costs per station:
WH1080: $\sim$ 100 EUR, Raspberry Pi + SD card + Power supply: $\sim$ 50 EUR, eventually extra 20 EUR for an USB UMTS stick.

[a] https://www.rettetdasgoertschitztal.at
[b] https://crowdmeteodata.aau.at/meteo0/

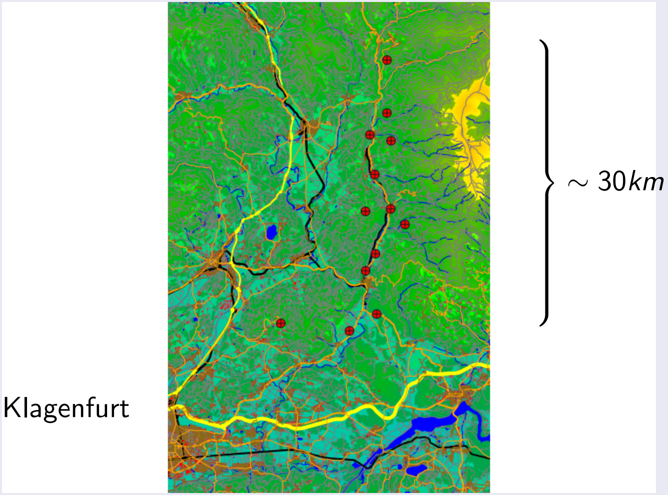
Station networks II

WH1080 installation



Station networks III

Görtschitztal network



Interpolation I

Interpolation on a 3-dim. grid

Wind speed data for dispersion simulation is needed on a three dimensional grid G, therefore interpolate station data $u(z_i), i = 1, \dots, n$. This is done in several steps.

Spatial interpolation

- using IDW (inverse distance weighting) method as a very simple model for spatially dependent data:

$$\forall z_0 \in G : u(z_0) = \sum_{i=1}^{n} \lambda_i u(z_i),$$

with weights $\lambda_i = \frac{|z_0 - z_i|^{-1}}{\sum_{j=1}^{n} |z_0 - z_j|^{-1}}$
data at large distance is down weighted,
maximum for λ_i is 1 for $z_0 = z_i$.

Interpolation II

Vertical extrapolation

two variants for extrapolating measured speed u at height z_r into greater heights z_h:

using power law:

$$u(z_h) = u(z_r)\left(\frac{z_h}{z_r}\right)^p$$

where p is dependent on stability and roughness.
(see Sedefian [1979])

Interpolation III

using log wind profile:

$$u(z_h) = u(z_r)\frac{\ln((z_h - d)/z_0)}{\ln((z_r - d)/z_0)}$$

z_0 is a roughness parameter depending on ground structure, taken from tables, and d denotes the zero plane displacement (=height where obstacles slow down wind speed to zero).
(see Tennekes [1973])

Interpolation IV

Map speed vectors to DEM data

- Taking topography into account:
  - Weather stations deliver only two dimensional wind directions with x and y components
  - we choose a z-component for the wind direction by taking a gradient vector from the DEM data for some fixed radius d around the station so that its vertical projection matches the measured 2-dim vector.
    Wind speed vectors in flat region will remain two dimensional, at steeper ascends or descends wind speed vectors follow these gradients of the topographical surface.

Interpolation V

Implementation details

IDW interpolation is implemented

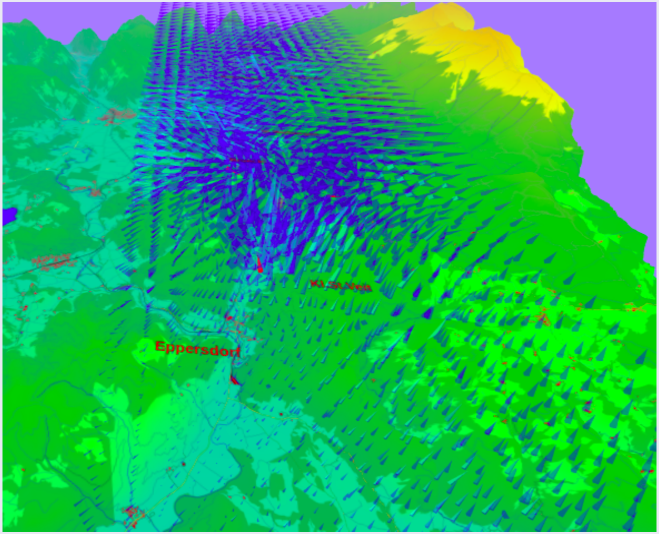
- as part of an R package,
- utilizing parallel computing via CUDA, if available.

The interpolation service for visualization of the 3d-grid of wind speeds is even running on small embedded NVidia hardware (Jetson TX2 with NVidia Pascal GPU[a])

[a] `https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-tx2/`

Interpolation VI

3d-plot of interpolation



blue: interpolated, red: data,
arrow length proportional to wind speed.

Outlook I

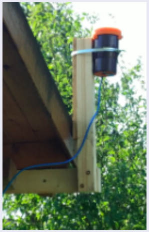
Current use

- Wind speed vectors are used in particle dispersion simulations.
- Comparison of data with official weather stations, we already have two sites located at such stations (Airport Klagenfurt, Landesregierung Kärnten). Not yet finished.

Outlook II

Further developments

- An interactive 3d-plot for interpolated wind fields (almost finished, see screenshot above).
- An allsky cam to determine degree of cloud coverage as another input for the dispersion models. (almost finished)



- Use of rain data for simulation of wet deposition of particles.

References

References

Alexey Ozerov. FREETZ Weather für WH1080/WH3080. https://baublog.ozerov.de/wetterstation/freetz-weather-datenlogger-fuer-wh1080-wh3080/, 2011. [Online; accessed 31-August-2019].

Conrad Electronic SE. Radio weather station with usb and touchscreen. http://www.produktinfo.conrad.com/datenblaetter/650000-674999/672861-an-01-ml-FUNK_WETTERSTATION_MIT_USB___TOUCH_de_en.pdf, 2014. [Online; accessed 31-August-2019].

Leon Sedefian. On the vertical extrapolation of mean wind power density. *Journal of applied meteorology*, 19:p.488–493, 1979.

H. Tennekes. The logarithmic wind profile. *Journal of the Athmospheric Sciences*, 30:p.234–238, 1973.