Building your own Sky Imager with cloud coverage estimation

I. Introduction

In this tutorial, you will learn how to create your own custom made Sky Imager. You will program a Raspberry Pi to instruct a camera to take pictures of the sky at regular intervals. You will install a web interface to access the captured images and you will apply signal processing techniques to compute the cloud coverage.

II. Hardware requirements

In order to build your own sky imager, the following hardware is required:

- A digital camera, controllable using the Gphoto2 library. Supported cameras are listed here: `http://gphoto.org/proj/libgphoto2/support.php`. Note that the camera needs to be continuously powered, and that the USB port will be used to control the camera. Depending on your model, you might need to use a power adapter, which uses the battery ports to supply electricity from a power socket.
- (optional) Depending on the camera you use, you can consider using a fish-eye lens in order to increase the viewing angle of your sky imager and capture the entire sky hemisphere in a single image.
- A Raspberry Pi single board computer with its SD card.
- A thumb-drive or hard drive with sufficient capacity to store the captured pictures.
- A sealed casing with a transparent dome in order to protect the components from the outside environment. Although this tutorial does not focus on this part, this shouldn't be neglected. Use your imagination! Placing the system behind a roof window can also be an option, especially if you do not use a fish-eye lens.
- A wired internet connection is needed to read the captured images using the web interface.
- A screen, a keyboard and a mouse are also needed during the programming stage, but are not required during operation afterwards.

III. Assembling the hardware

The camera needs to be both powered and connected by USB to the Raspberry Pi, which itself needs to be powered and connected to the internet. In development stage, a keyboard, a mouse and a screen need to be connected to the Raspberry Pi. They can be unplugged afterwards.

If the Sky Imager is deployed outside, a weather-proof casing for protecting those components is needed. A transparent enclosure needs to be present for the lens. If you use a fish-eye lens, your enclosure shall have a dome shape and the lens should be slightly inside it to avoid occlusions.

See Fig. 1 for examples. If the Sky Imager is used behind a window, a tripod for the camera can be sufficient.

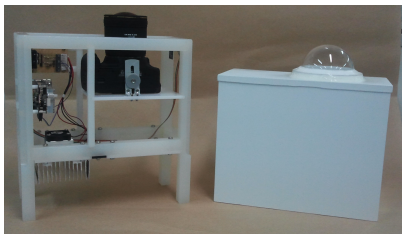


Figure 1: Examples of weather-proof casings

IV. Programming the Raspberry Pi

I. General setup

A general tutorial on how to setup a Raspberry Pi can be found here: `https://www.howtoforge.com/tutorial/howto-install-raspbian-on-raspberry-pi/`. Get your Raspberry Pi running using the Raspbian OS.

The rest of this tutorial will solely use terminal commands. You have two options: you either continue using the GUI, or you can use SSH to remote control from another PC (see here: `https://www.raspberrypi.org/documentation/remote-access/ssh/`).

You need to download the source code for the project from the github repository. Install git using: "`sudo apt-get install git`". While being in the home folder ("`cd ~`"), download the content using: `git clone git://github.com/FSavoy/DIY-sky-imager.git` Make those files executable using: "`chmod 777 DIY-sky-imager/*`"

We also need to prepare the thumbdrive or hard drive we will use to store the pictures. Follow this tutorial to automatically mount it to the location of your choice: `http://www.modmypi.com/blog/how-to-mount-an-external-hard-drive-on-the-raspberry-pi-raspian`

II. Controlling the camera

We control the camera using the *gphoto2* library. Install it using "`sudo apt-get install gphoto2`"

The script to capture and download an image is named `capture_image.py`. We need to instruct the Raspberry Pi to execute this script at regular intervals, say 2 minutes. We use *cron* for this purpose. Type "`sudo crontab -e`" in a terminal, choose an editor and add the following line at the bottom of the file:

```
*/2 7-19 * * * python /home/pi/DIY-sky-imager/capture_image.py
```

This command instructs the operating system to execute our script every 2 minutes from 7am to 7pm every day.

This script needs the camera to be available, which is not the case when the Raspberry Pi is started in the desktop mode. Use `sudo raspi-config` (Set *Enable Boot to Desktop/Scratch* to *Command line*) to instruct the board to boot in command line mode. Then restart it (`sudo reboot`).

III. Computing the cloud coverage

We use python, along with OpenCV library to perform the cloud coverage computation. Install OpenCV using the following tutorial: `http://www.pyimagesearch.com/2015/07/27/installing-`

`opencv-3-0-for-both-python-2-7-and-python-3-on-your-raspberry-pi-2/`. We use *python3* in this project. The scripts to create a binary cloud mask are launched by `capture_image.py`. They create a second binary version of the image where white indicates the presence of clouds.

You need to modify two variables inside the file, which can be opened using nano: `nano ~/DIY-sky-imager/capture_image.py`. Modify *basedir* to point to the location of the mounted storage device. If you use a fish-eye lens and observe a black circular margin around the image, set *fisheye_radius* to its radius. Otherwise set it to `None`.

IV. Installing the web interface

In this project we are using *browsepy* (`https://github.com/ergoithz/browsepy`) as a web interface to access the captured images. Install it using "`sudo pip install browsepy`". You need to be outside the virtual environment (by running "`source ~/.profile`" if needed).

To run the server, type the following in a command line (replace basedir with the absolute path to the storage device):
`sudo python -m browsepy 0.0.0.0 80 --directory` *basedir*
Again, we will use *cron* to run the server at every reboot. Type "`sudo crontab -e`" in a terminal and add the following line at the bottom of the file:
`@reboot python -m browsepy 0.0.0.0 80 --directory` *basedir*

You can now type the IP address of the Raspberry Pi (run "`ifconfig eth0 | grep 'inet addr:' | cut -d: -f2 | awk '{ print $1}'`" to retrieve it) in the browser of a machine connected to the same network to access the images.

V. Conclusions

The sky imager is now ready. You can now unplug the keyboard, mouse and screen and start capturing images of the sky. Have a look at the scripts computing the cloud masks, they use Otsu's method to detect clouds.

References

[1] S. Dev, Y. H. Lee, S. Winkler, Systematic Study of Color Spaces and Components for the segmentation of sky/cloud images, *Proc. IEEE International Conference on Image Processing (ICIP)*, 2014