SWEETcam:
Solar Wi-Fi Energy Efficient Tracking Cameras

Baskin Engineering UC SANTA CRUZ

Leland Miller (ledmille@ucsc.edu), Kevin Abas (kabas@soe.ucsc.edu)
Advisor: Katia Obraczka (katia@soe.ucsc.edu)

Inter-Networking Research Group

MOTIVATION

The SWEETcam project is intended to improve upon traditional camera networks in several ways:

- Lower the cost and increase the availability of camera networks. This is especially important in enabling wider range deployments for research.
- Reduce the required infrastructure for camera networks. Allow for the possibility of running a camera network "off the grid."
- Allow for the analysis of video data at recording time, and therefore enable the selective recording of interesting events.

We hope that advancements in these areas will enable video recording networks to have improved utility in applications such as environmental monitoring, security, automated systems monitoring, and computer vision research.

SMART CAMERA WORKFLOW

Our camera network works differently than a normal camera network. Instead of continuously recording video or recording video whenever motion is detected, our cameras have the ability to use computer vision algorithms to determine the importance of detected movement while it is happening.

Figure 1 An example of tagged video files produced by the computer vision algorithms. Note the ability of the algorithm to track the distinct objects even as they cross paths.

1. At its resting state the camera is mostly powered down. There is no communication possible to the camera. Solar panels charge a battery whenever possible and a low-power motion detector is active.
2. When the motion detector is triggered it causes the camera to boot up and begin analyzing the video feed. The camera also immediately begins sending any previously recorded video data and status messages to the server system.
3. The camera begins receiving messages from the server after the server receives the first message sent from the camera.
4. If video activity is deemed important the video is tagged and recording continues, otherwise recording stops.
5. Unless the server requests that the camera stay on to receive more messages, the camera shuts down and the low-powered motion detector waits for more activity.

- Though this workflow does not allow for instant control of the camera from the server or an uninterrupted stream of video, it does conserve enough energy so that we are able to use Wi-Fi on limited battery and solar power.
- This mode of operation also removes much of the human labor involved in constantly monitoring video streams and analyzing recorded video. We hope that this will make applications that monitor large areas with small amounts of activity, such as wildlife monitoring, more accessible.

THE CAMERAS

Figure 2 Raspberry Pi B+ microcontroller

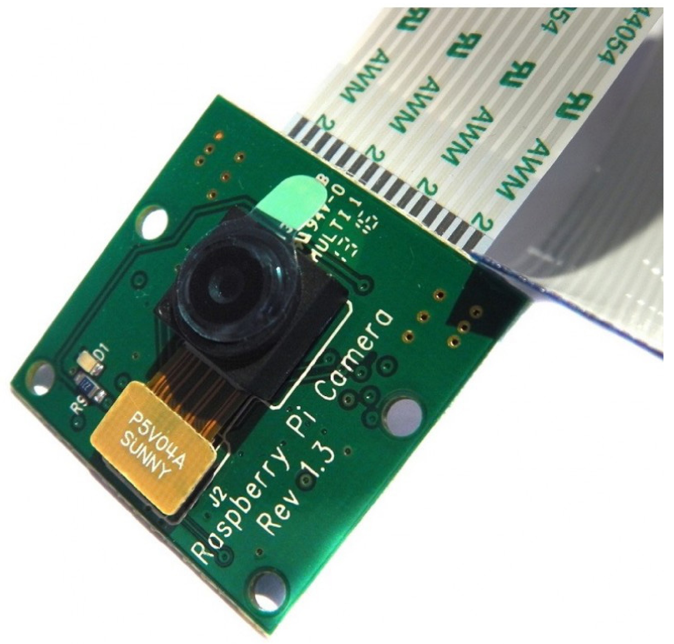

Figure 3 The Raspberry Pi camera module

- A camera node consists of a Raspberry Pi, a camera module, and a Wi-Fi module. There is also a very low power processing unit connected to a motion detector chip that controls the power to the main Raspberry Pi. All of this is powered by a battery that is charged by solar panels.
- The Raspberry Pi offers a compact and power-efficient form while still providing powerful enough GPU for our algorithms.
- We are currently switching to the newly released Raspberry Pi Model B+. This new model offers significant power savings over the previous model while still offering the same performance.
- Camera nodes are completely self-contained. By using Wi-Fi and solar energy they are able to avoid requiring any wires or connection requirements besides a mounting point, a Wi-Fi connection, and occasional sunlight.
- Initially these modules will connect over the campus-wide Wi-Fi network. However, we plan to eventually use ad hoc wireless networks for true "off the grid" deployment.
- Cameras are made mostly out of commercialy available parts. This should help keep down the costs and allow for the easy creation of nodes by other research teams.
- The camera nodes are running a customized version of the Linux kernel that has been optimized for power-efficiency while still meeting the requirements of our computer vision algorithms.

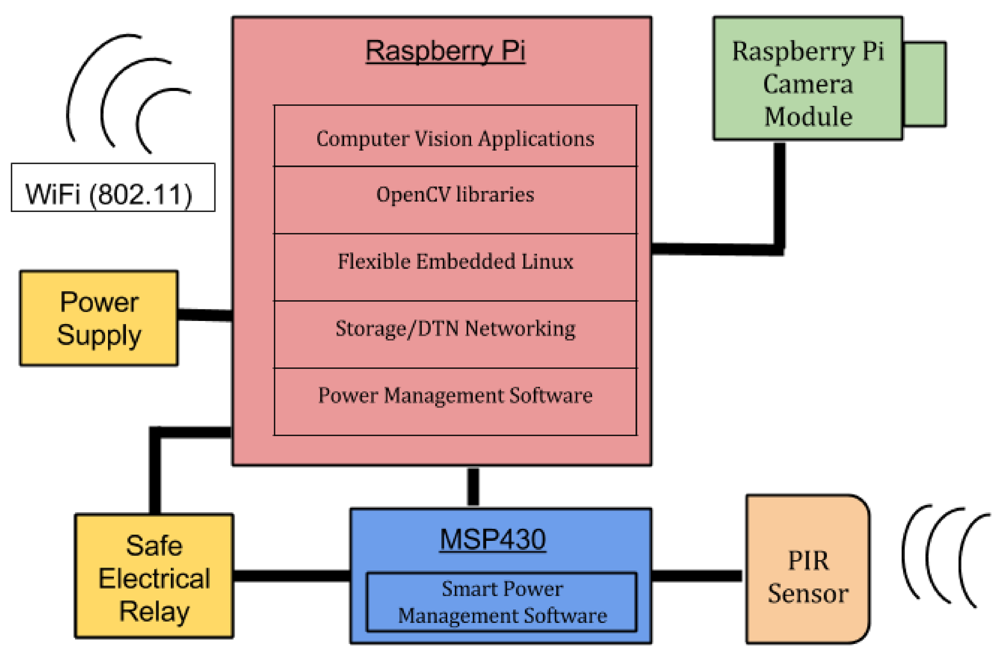

Figure 4 Hardware design of the SWEETcam module

Figure 5 Passive infrared sensor (for motion detection)

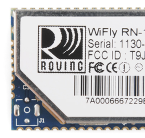

Figure 6 802.11 (Wi-Fi) module

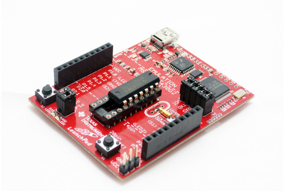

Figure 7 MSP430 microprocessor

THE NETWORK

SWEETnet, the network supporting the SWEETcam project, consists of three major parts. There is a server responsible for receiving video from the cameras, a server responsible for all other communication with the cameras, and a server responsible for packaging this data in a way that it is useful to applications.
All network protocols were chosen to be easy to understand, reliable, and extensible. Along with these traits, power consumption was an important factor, and design decisions were made balancing all of these factors.

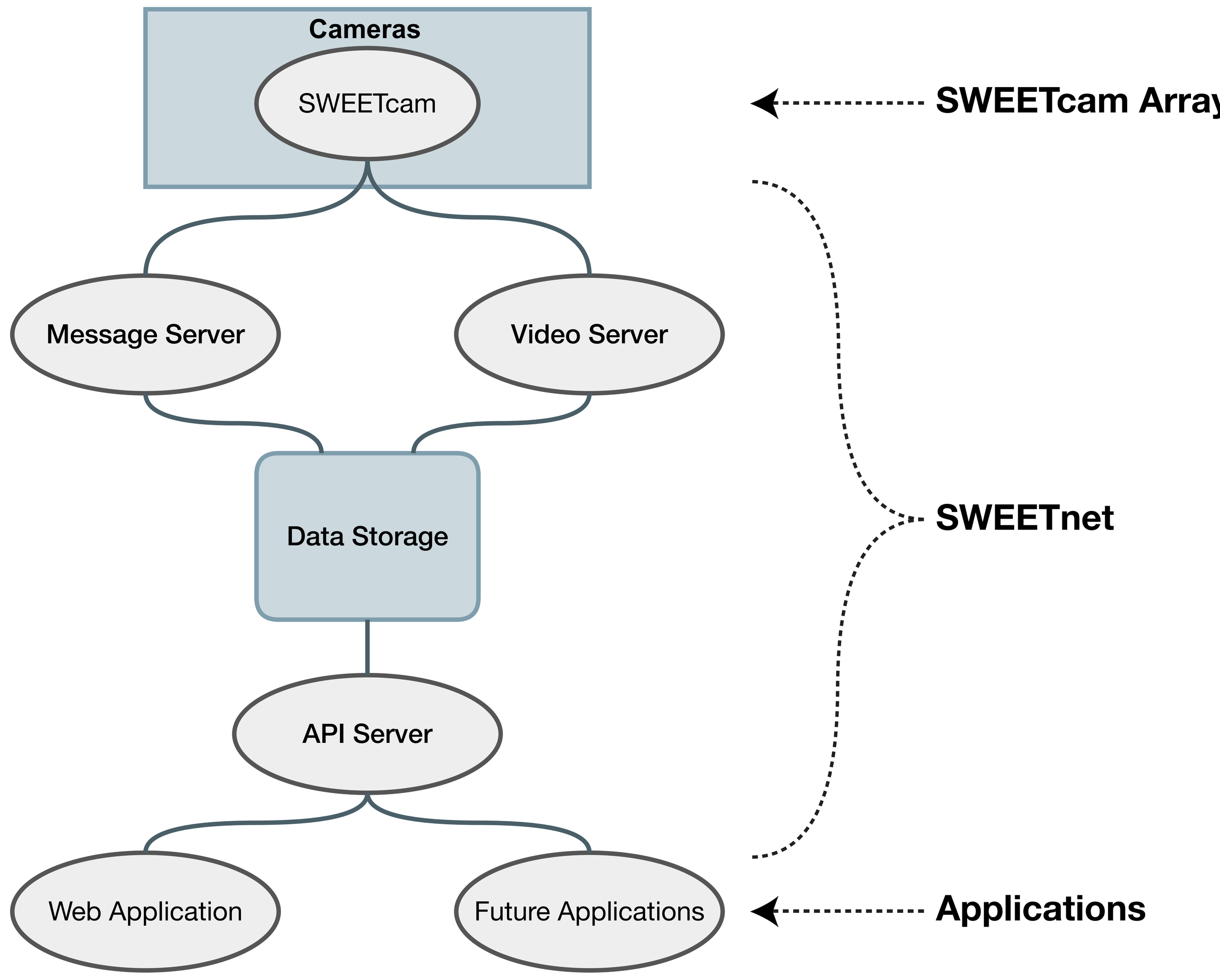

Figure 8 Graph showing the structure of the whole project, and the role of SWEETnet

- Our network software is written for the Node.js server-side JavaScript platform and uses MongoDB for data storage.
- The modular design of the network software means that we can easily test and refactor major portions of the code. It also allows us to make experimental modification to some functionality without worrying about breaking the rest of the network.
- The network is designed so that all communication is initiated by the camera. This means that any messages that need to be sent to a camera need to wait for that camera to come alive before they can be sent.
- The network is also designed so that a new camera is automatically recognized and registered in the system for ease of deployment.
- The API server provides a consistent interface for building end-user applications. This API is designed to support as broad a range of applications as possible in addition to the web app we are developing.

ACKNOWLEDGEMENTS

We would like acknowledge and thank Kerry Veenstra for all of the mentoring and advice he provided us with on this project. We would also like to thank Katia Obraczka for all of the support and guidance she offered throughout our research.

IMAGES

Figure 2 - http://www.amazon.com/Raspberry-Pi-Model-Plus-512MB/dp/B00LPESRUK
Figure 3 - http://www.pi-supply.com/wp-content/uploads/2013/07/raspberry-pi-camera-board-close-800x800.jpg
Figure 5 - http://www.parallax.com/product/555-28027
Figure 6 - http://www.tenettech.com/product/1931/wifly-rn-171-80211bg-serial-module-roving-networks
Figure 7 - http://fabacademy.org/archives/2012/students/pello.david/classes/input_devices/launchpad.jpg