CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title:	BiVo: An Open-Source Foundation for Remote Monitoring of Bird Vocalizations
Sponsor Information: NAU NORTHERN ARIZONA UNIVERSITY School of Informatics, Computing, and Cyber Systems	**Dr. Paul Flikkema**, Professor School of Informatics, Computing, and Cyber Systems Paul.Flikkema@nau.edu Northern Arizona University

Project Overview

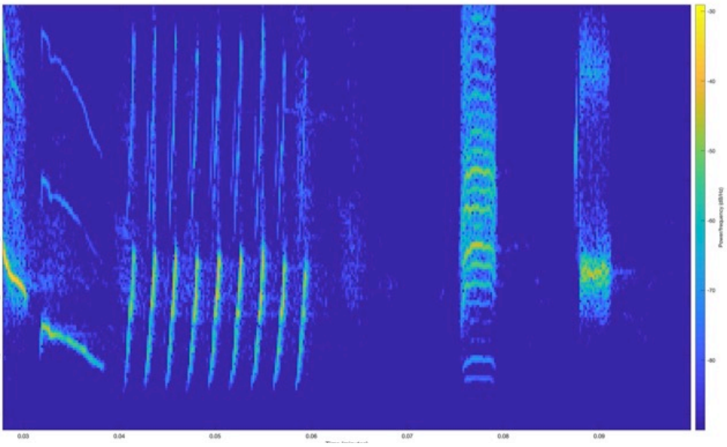

Figure 1. Spectrogram, or frequency vs. time plot, of about 4 seconds of vocalization from a Northern mockingbird. Data from D. Kroodsma, *The Singing Life of Birds*, Houghton Mifflin, 2005.

Understanding the behavior of birds and the ways that they communicate is an area of active research in wildlife ecology. While we know that birds communicate through calls and songs, we do not yet know the languages that they use, and the ways in which they use them. Moreover, detection and classification of bird vocalizations (Figure 1) enable us to better understand where and when bird species and individuals forage for food and nest, and how these change over time.

People have been listening to birds and analyzing their vocalizations for decades. For example, the Audubon Society gives guidance on how to assemble costly kits of microphones and recorders for in-person monitoring. Scientists also use these techniques. However, many studies use unattended sensing. Proven, robust devices are available (e.g., The Song Meter Mini from Wildlife Acoustics), but they are expensive and use proprietary hardware and software. More recently, researchers have developed low-cost, open-source alternatives based on low-cost microcontrollers and MEMS microphones; perhaps the most popular of these is the audiomoth. However, the audiomoth documentation does not easily support evolution to new capabilities that we are interested in: scalability of processing (the ability to add more intelligence to the processing algorithms), wireless networking for multi-sensor localization and forwarding to servers for more detailed analysis, and integration of solar energy harvesting. What is needed is an open-source hardware/software foundation for a low-cost, evolvable, networked remote sensor.

The BiVo Project

Fortunately, driven by commercial interest in the Internet of Things, microcontroller starter and development kits are becoming available for prototyping tiny sensors with audio recording and processing capabilities. This project will use the Silicon Labs Thunderboard EFM32GG12, which was selected for the following reasons: (1) it uses a powerful, industry-standard 32-bit floating-point MCU, (2) it integrates two reasonably-sensitive MEMS microphones, (3) the integrated development environment (IDE) is well-designed and available for Windows and OS X, and (4) the MCU, board, and IDE provide strong debugging and performance analysis features.

The key product deliverable is embedded software (see functionality sketch below) that will run on the Thunderboard, enabling packaging with a battery to enable field testing and deployment. As mentioned above, it will capture sound, perform some initial processing, and make interesting data available to the second component, a basic desktop application.

A more detailed sketch of desired functionality includes:

The basics: minimum viable product:

- Core processing application that runs on the Thunderboard to do the following key functions:
    - Capture the audio data streams from two microphones
    - Split the data into 4-second blocks
    - Compute the spectrogram of each block
    - Classify each block as interesting or not interesting (we will discuss how to make this decision)
    - Store interesting blocks in FLASH for later recovery (format TBD since MCU will not have a file system)
- Basic desktop application (OS negotiable) to connect to and control the BiVo embedded application and retrieve data; a simple command line interface is acceptable

Beyond the basics: features for a usable product

- Augmented desktop GUI interface stuff
- Connect to BiVo, review status, configure more things in the GUI
- Control recording behavior from GUI in more detail
- View all data files collected, be able to download/delete
- Enable export in format to enable manipulation, analysis, and visualization in Matlab

Stretch goals: moving towards real world deployment

- Programmable monitoring intervals throughout the day
- Forward especially interesting blocks over wireless to servers (requires interfacing with additional wireless MCU)

Knowledge, skills, and expertise required for this project:

- C language programming for microcontrollers
- Some knowledge of or interest in microcontrollers and embedded systems
- Will need to learn about processing of audio data streams from microphones
- Programming in a suitable language for development of desktop control UI
- Optional: Graphing and analysis of audio data streams

Equipment Requirements:

- Silicon Labs Thunderboard EFM32GG12
- Sponsor will make available audio files of bird representative bird sounds to dev/test with
- All other software are IDEs are freely available; no further specialty items are required

Equipment not free or immediately available will be purchased and/or facilitated by the sponsor

Software and other Deliverables:

- The software applications as described above, deployed and tested successfully with real data. Must include a complete and clear User Manual for configuring and operating the software, as well as *continuing its development*.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development and/or extension of the product.
- Complete professionally-documented codebase, delivered both as a repository in GitHub, BitBucket,or some other version control repository; and as a physical archive on a USB drive