EC 440 Practical Training and Educational Tour Report
Dept. of E & C
National Institute of Technology Karnataka, Surathkal
Academic year 2013-14
VII Semester B.Tech.

BIRD CALL IDENTIFICATION AND DISTANCE MEASUREMENT USING ANDROID

Submitted By: Sachin Vernekar 10EC110
Project Guide: Dr. H.S Jamadagni
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

Place:Indian Institute of Science, Bangalore
Period of Internship: 20th May 2013 – 20th June 2013

Indian Institute of Science, Bangalore

The report covers the work done on building android application for bird call identification and distance measurement using camera which are a part of a bird software called "Pakshi". The project was carried out at the Department of Electronic Systems Engineering(formerly CEDT), IISc Bangalore, where I worked as a summer research fellow.

The Indian institute of science is one of the oldest and the most sought after colleges in India for M Tech programmes and was founded by Jamsetji Nusserwanji Tata in 1909. The institute is associated with great names like Swamy Vivekananda, the Maharaja of Mysore, Shri Krishnaraja Wodeyar IV and many of India´s most distinguished scientists G.N. Ramachandran, Harish Chandra, S. Ramaseshan, A. Ramachandran, C.N.R. Rao and R. Narasimha and others.

Talking about the Department of Electronic Systems Engineering(DESE), it was started in the year 1974 as Centre for Electronic Design and Technology(CEDT) with support from the DoE, UGC and the Swiss Development Cooperation. In January 2012 it was renamed as DESE. Within a few years of its establishment, the M.Tech programme in Electronics Design was started, and this continues to be one of the most sought after programmes in the country. The research done here deal with building prototypes that take into account a multitude of practical issues, including industrial design and product engineering, ergonomics, system-level packaging and thermal design.

MOTIVATION and AN INTRODUCTION TO THE PROJECT

Since I have been a keen student of Digital Signal Processing and Programming, I couldn't find a better project than building DSP applications on an android platform. The Institute where I did my internship is a well-known institute and also the summer research programme was funded by the institute. Hence I chose this institute to work on my summer project.

Bird Call Identification:
Tools used: Matlab, ADT-Bundle for building android application, android phone.
Various species of birds have unique bird calls. These bird calls are distinct based on inflection, length, and context, meaning the same bird may have more than one call. A device that would analyze the signal and identify the bird based on the bird call could be of tremendous help to an ornithologist. This project proposed the development of this application using signal processing on an android platform which is better than and embedded platform in many ways for building a prototype. The first task was to find or create a database of high-quality bird calls to use for identification. Using this database, I compared various features of the bird calls of a certain species and ascertained the features which distinguish that species from other species. Using these features, a recorded bird call was identifiable as a species of bird.

In order to create a bird call identification device, it was needed to correctly utilize several signal processing techniques. Some of these techniques included filters, discrete Fourier transforms, cross-correlation, wavelets, cepstral analysis, and audio spectrograms. Filters were necessary to improve the quality of the bird songs and remove any unwanted noise. Bird songs cover a wide frequency range and discrete Fourier transforms was used to analyze the different frequencies in each call. Cross-correlation was used to compare recorded bird calls with the bird call database, both in time and frequency. Next, I used discrete wavelet transforms. An advantage that discrete wavelet transforms (DWT) have over Fourier transforms is temporal resolution; a DWT captures both frequency and time information. Additionally, audio spectrograms were created using both Fourier and discrete wavelet transforms to examine each bird's song. Lastly mel-frequency cepstral coefficients were used. To find these mel-frequency cepstral coefficients, the Fourier transform is taken and then the power of the spectrum is mapped onto the mel scale (Lee et al., 2006). At that point, the discrete cosine transform (DCT) of the mel logarithms of the power spectrum is taken, and the resulting amplitudes are the mel-frequency cepstral coefficients. Each technique manipulated the bird call in a different way to help identify which species the call originated from.

Distance Measure using Camera:

Tools used: ADT-Bundle, android phone Samsung Galaxy S2, Matlab .
Since the "Pakshi" software was developed as a mobile phone application, I had to find a way to measure the distance at which the bird was spotted using the android phone itself, then I found that android camera can be used for that. Here we find the distance 'd'of bird from camera by knowing the height 'h' above the ground from where the bird is focused and the angle 'a' between the perpendicular line from ground to camera and camera to bird, and use the formula

$$d = h * \tan(a)$$

Measuring distance is not as simple as it looks to be considering just the above formula. The problem here is the measurement of angle 'a', which is generally taken from orientation sensor found in most of the modern day android devices. But this angle happens to not very precise. Its precision is only 0.1 degrees, but to get a better range and precision of distance measured we want the precision to be as good as 0.01 degrees. I could achieve this precision by a sensor fusion of orientation sensor and gyroscope which measures angular speed of rotation.

WORK AND EXPERIMENTAL DETAILS

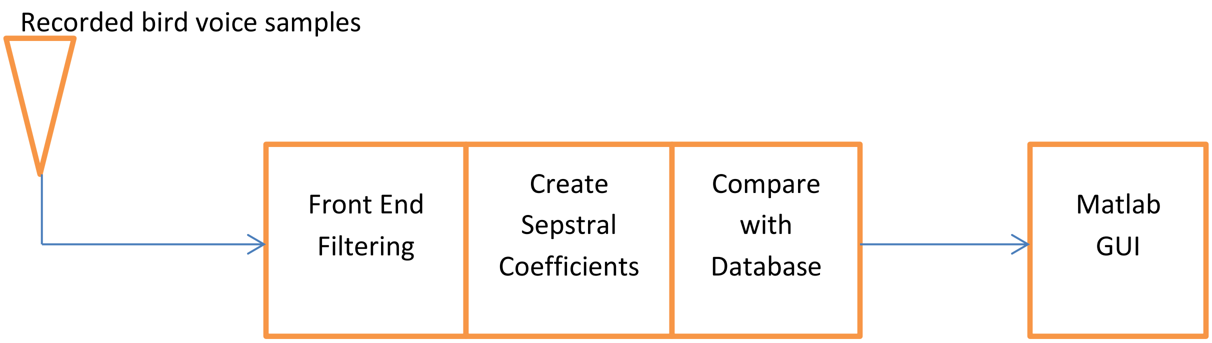
Bird Call Recognition:

The goal of this project was to be able to identify a bird species based on its call. The objectives that were necessary in this endeavor are as follows:

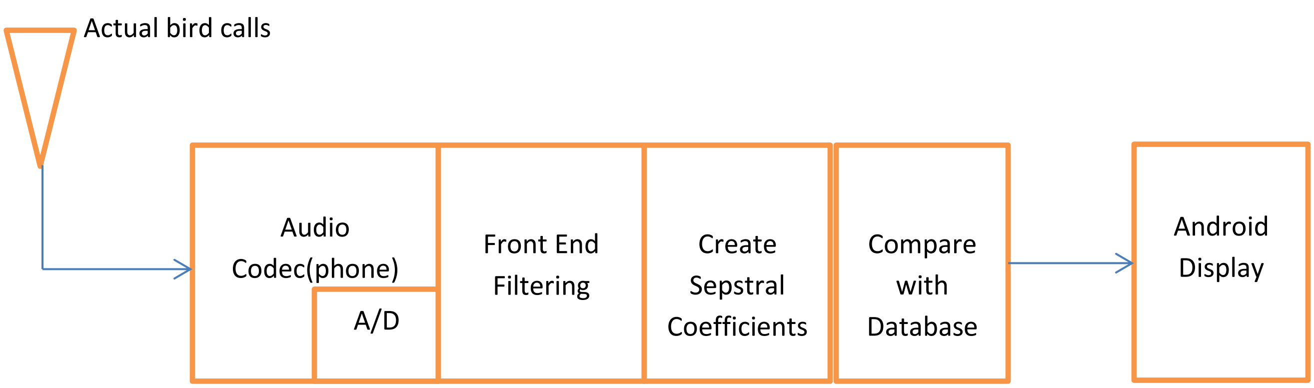
- Resolve a way to identify one bird species despite a variation of calls from that species
- Analyze and identify bird species using MATLAB
- Analyze and identify bird species using the Java
- Implement the Java program on an android platform

The team began by resolving a way to identify one bird species despite a variation of calls from that species. To do this, the literature was researched to find a way to isolate that species from all other species based on certain characteristics of the bird call. The team elected to utilize the mel-frequency cepstral coefficients. The specific details of this implementation will be discussed later.
Next, this processing stream was coded in MATLAB. Many of the necessary functions were available online. The team's contribution to this code was correlating the resulting MFCCs from one bird call with another and plotting the results.
After that, the group implemented the same processing stream using the Java programming language. This code was written mostly from scratch although some of the functions were adapted from online open source code available. The platform selected for algorithm implementation is an android phone specifically Samsung Galaxy S2.
After the Java code was complete, a significant amount of time was spent debugging it. Although the code compiled, there were fundamental errors in the processing that needed to be resolved which was done with on-board debugging option available in ADT-Bundle software.

System Block Diagram:



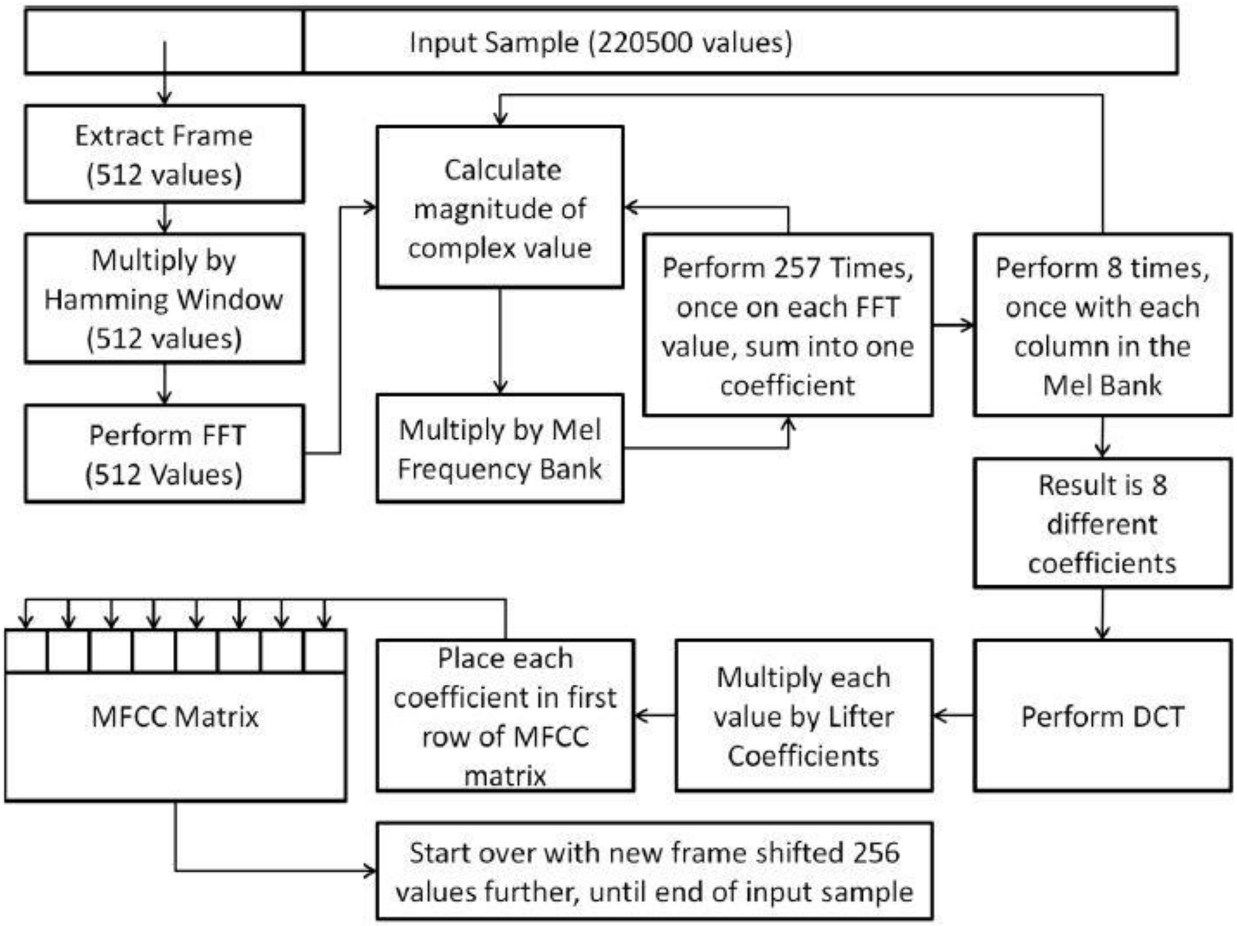
Matlab Implementation



Android Implementation

The analog input the system receives from the microphone is converted through an audio codec available in the android device and then is put through a lowpass filter. The lowpass filter has a passband up to 10 kHz and a stopband that begins at 11 kHz. Some birds' songs do go above this range, but it is rare for them to as the average bird song frequency is 4 kHz (Brand, 1938). Testing the bird song signals of interest confirmed that a 10 kHz passband was sufficient for the purpose of the team's samples. There was no highpass filter because a highpass filter would remove the low frequency envelope information, which is part of the information represented by the MFCCs.

Next, the MFCCs were found to identify the species of bird. To find the MFCCs, the Fourier transform is first taken of a portion of the signal. Then, the power of the spectrum obtained is mapped onto the mel scale and take the logs of the powers at each of the mel frequencies. Finally, the discrete cosine transform of the list of mel log powers is taken and the amplitudes of the resulting spectrum are the MFCCs.
By comparing these MFCCs to MFCCs calculated from training data collected for each bird species, the algorithm can identify the bird species. For this project, these trained MFCC collections were generated using at least three bird calls from four different bird species and use the same MFCC method described above. These MFCC replicas formed the four specie database.
From there, the system outputs the closest three matches to the console, complete with how well the database MFCCs correlated with the MFCCs of the incoming signal. If the correlation is below 0.50, the particular database signal is not considered a match. This prevents a signal that does not correlate well with any signal in the database from being incorrectly identified.

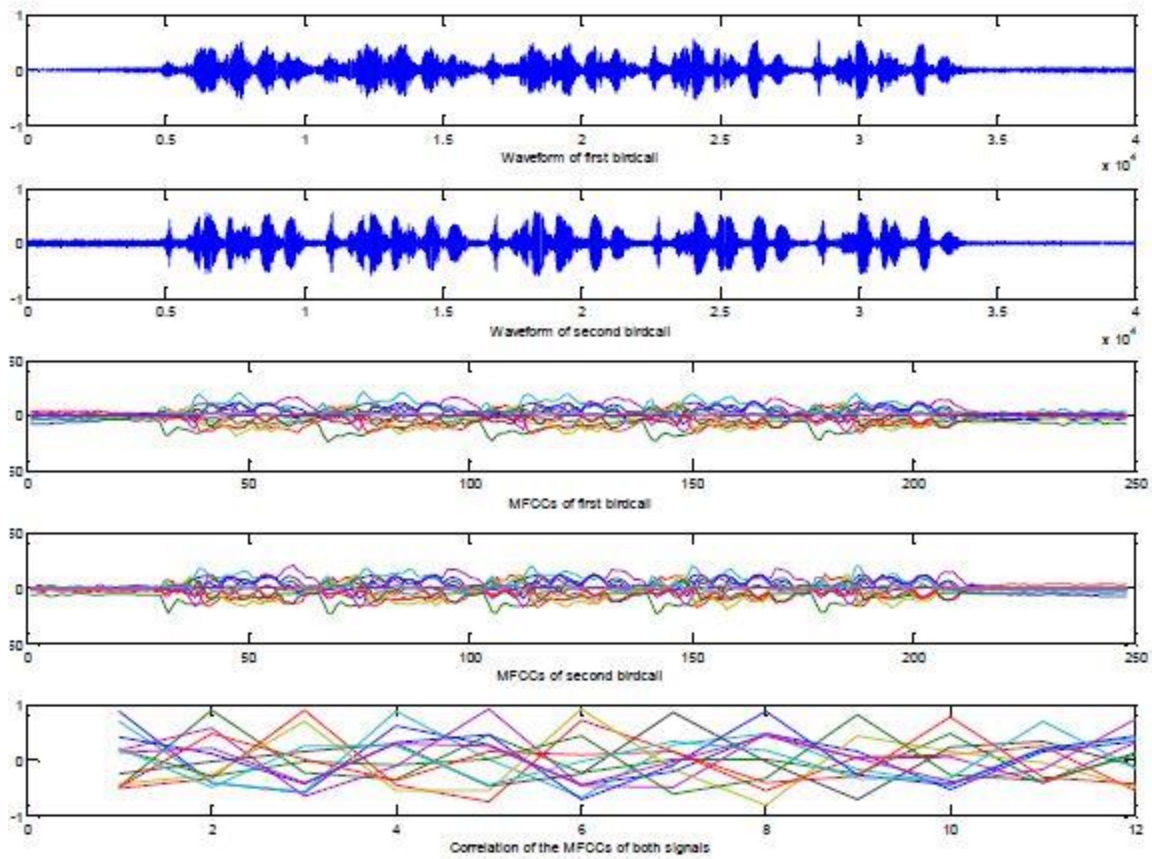


Algorithm Flow Chart

The main result of the project is to get the algorithm working in Matlab first and the same then can be easily implemented for android.

RESULT

It was necessary to test the MFCC algorithm to be sure that it would function in the intended manner. To do this, a variety of bird songs were sampled and compared with the algorithm. To start out simply, two signals compared were from a single Crow. As seen in the figure below, the MFCCs of each bird call had many similarities. The correlation match-up for these bird calls was around 85%.



Comparison of Two calls from the same Crow

It was necessary to test this algorithm with signals that are vastly different from bird songs. Therefore, the algorithm was also tested with two human whistle trills. The correlation of these signals had an average correlation of MFCCs around 88%.

After identifying that the algorithm was reasonably reliable at identifying bird songs, the MATLAB algorithm was automated for further testing. One song was taken from each bird song type of each bird species and added to the database in order to decrease the processing time of the identification algorithm. The automated algorithm took in only one bird song as well as the number of MFCCs needed and the sampling frequency. The algorithm then compared the unknown bird song with all of the songs in the database.

The algorithm was then tested with bird songs that were not in the database to see how accurate the algorithm was with signals unknown to the system.

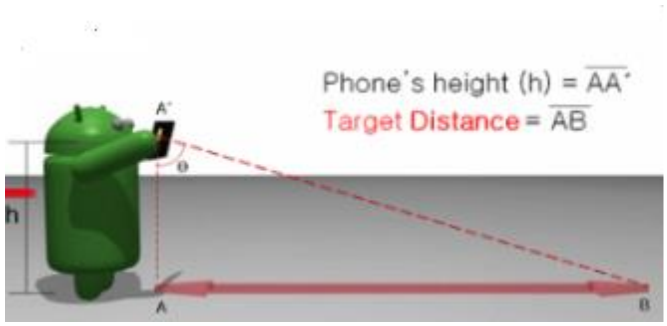
The available songs from the Macaulay Library that were not part of the database were tested to find the accuracy of the algorithm for a small sample set. For this sample set, the algorithm was 93.75% accurate in identifying bird songs. However, although the algorithm correctly identified the birds most of the time, some bird calls would not correlate well with the same species. For example, the House Wren's "song" is for the most part just a series of tweets. When creating the algorithm, it was not a consideration that the term “bird song” may mean something very different from one species to the next.

Next we were able to successfully implement the same algorithm on an android platform.

Distance Measure using Camera:
The goal of this project was to be able to measure the distance at which the bird is spotted. The objectives that were necessary in this endeavor are as follows:

- Be able to measure distance using just the android device
- Should be able to measure distances at least up to 40m.

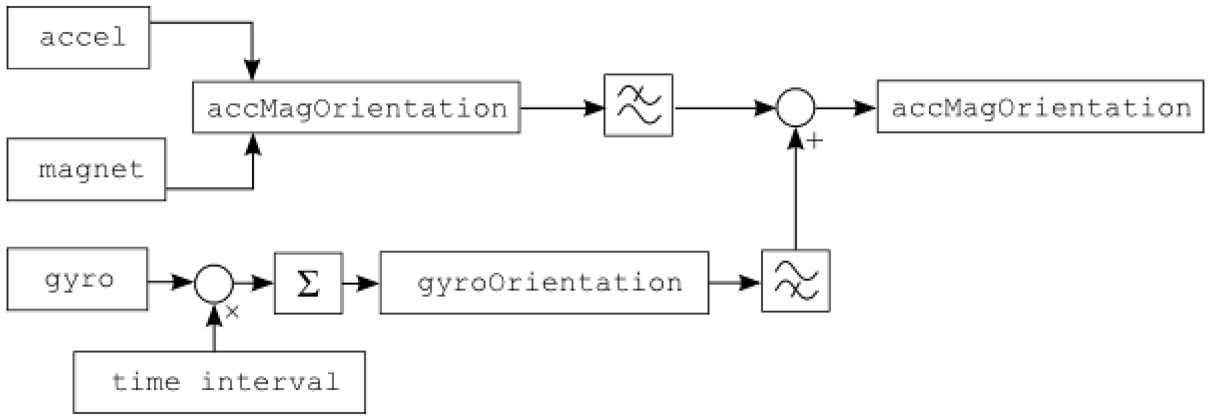
Hence android camera was used to complete the objective using triangulation technique. The implementation is as follows:



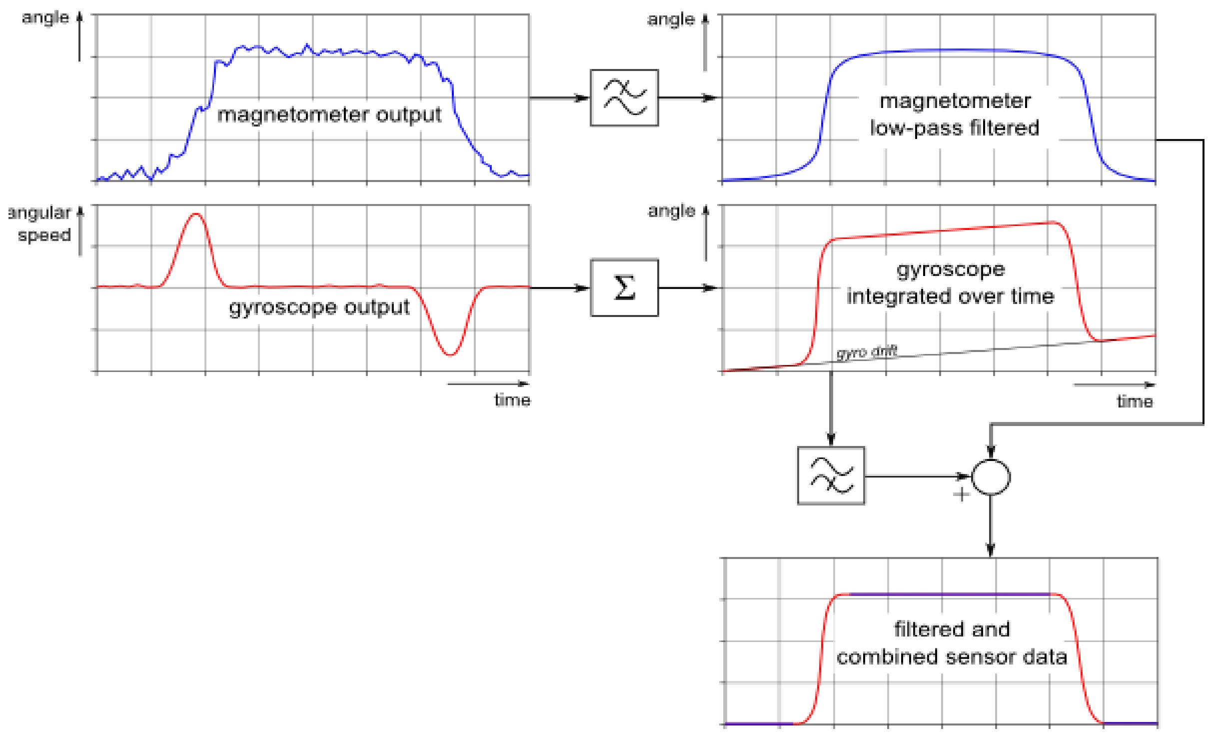
The calculation of distance is done as shown. d = h*tan(a)

The common way to get the altitude of an Android device is to use the SensorManager.getOrientation() method to get the three orientation angles. These two angles are based on the accelerometer and magnetometer. output. In simple terms, the accelerometer provides the gravity vector (the vector pointing towards the center of the earth) and the magnetometer works as a compass. The Information from both sensors suffices to calculate the device's orientation. However both sensor outputs are inaccurate, especially the output from the magnetic field sensor which includes a lot of noise.
The gyroscope in the device is far more accurate and has a very short response time. Its downside is the dreaded gyro drift. The gyro provides the angular rotation speeds for all three axes. To get the actual orientation those speed values need to be integrated over time. This is done by multiplying the angular speeds with the time interval between the last and the current sensor output. This yields a rotation increment. The sum of all rotation increments yields the absolute orientation of the device. During this process small errors are introduced in each iteration. These small errors add up over time resulting in a constant slow rotation of the calculated orientation, the gyro drift. To avoid both, gyro drift and noisy orientation, the gyroscope output is applied only for orientation changes in short time intervals, while the magnetometer/accelerometer data is used as support information over long periods of time.

This is equivalent to low-pass filtering of the accelerometer and magnetic field sensor signals and high-pass filtering of the gyroscope signals. The overall sensor fusion and filtering looks like this:



So what exactly does high-pass and low-pass filtering of the sensor data mean?
The sensors provide their data at (more or less) regular time intervals. Their values can be shown as signals in a graph with the time as the x-axis, similar to an audio signal. The low-pass filtering of the noisy accelerometer/magnetometer signal are orientation angles averaged over time within a constant time window. Later in the implementation, this is accomplished by slowly introducing new values from the accelerometer/magnetometer to the absolute orientation The high-pass filtering of the integrated gyroscope data is done by replacing the filtered high-frequency component from accMagOrientation with the corresponding gyroscope orientation values
In fact, this is already our finished complementary filter. Assuming that the device is turned 90 degree in one direction and after a short time turned back to its initial position, the intermediate signals in the filtering process would look something like this:



Notice the gyro drift in the integrated gyroscope signal. It results from the small irregularities in the original angular speed. Those little deviations add up during the integration and cause an additional undesirable slow rotation of the gyroscope based orientation.

RESULT

With the above technique I was able to get accurate results up to a distance of 80m with 1m accuracy, and angle values are highly accurate with precision of 0.01 degrees may be with very little error depending on the device.

EXPERIENCE FROM THE PROJECT

Since both the project I did on my internship needed me to basically implement everything on an android platform, I had to learn Java programming language and understand the features of modern day android mobile phones. Since I hail from E&C background I took initial few days to learn coding for android, it improved my understanding of object oriented programming language and also helped to be an efficient and professional android programmer.

For Bird Call Identification project I had to do a lot of literature survey in the field of digital signal processing techniques to recognize voices, this helped me to add on to my knowledge in DSP. I also learnt many different concepts like mel-frequency cepstrum techniques, audio compression, different practical noise filtering techniques etc.

To build distance measuring application, I had to have the complete knowledge of every bit of sensor available on an android phone and also know how to write code for all of them. This required me to do extensive study especially on gyroscope and magnetometer. I learnt how practically sensor values or noise can be filtered and also a new technique to measure distance.

In the whole I learnt how to approach a given practical problem and also to verify if the problem is actually viable, that is I was also working on an application to measure weight using android phone and its linear accelerometer, but in fact this was conceptually not viable. I could learn how to work and adapt in a practically working environment where we work as a team and not as an individual.

SUGGESTIONS FOR IMPROVEMENT AND FURTHER SCOPE OF RESEARCH

As far as Bird Call Recognition application is concerned, the MATLAB results were excellent for their purpose. It was possible to determine a bird's species with a high degree of confidence given the small sample set. Further testing is necessary to determine how well the algorithm is able to handle very large databases containing more than six species, but the team believes the MATLAB results can be used as a valid proof of concept.

So the future consideration would be, as described above, the database that began in MATLAB needs to be significantly larger. It is possible that the MATLAB results change as more birds are added to the database. The algorithm may struggle identifying two similar species that are not currently present within the database. Furthermore, it is likely that as the bird songs within the database increases, the probability for multiple songs to match the input signal increases. Results that prove the algorithm is still valid for a database with a larger number of bird songs would confirm that the MFCCs are indeed identifying the unique differences between signals and classifying them accordingly.

In terms of android implementation, the graphic user interface could be enhanced to give the users a better experience with the application like by adding pictures of birds and giving information regarding the identified birds.

With reference to Distance Measuring Application using Camera the accuracy of distance could be even bettered using more robust noise filtering techniques, and also code written for it could be further optimized and also additional options could be kept for the user like measuring the height given the distance and GUI front end could be further improved.

CONCLUSION

The result obtained for bird call recognition application on matlab conclude that the mel-frequency cepstrum technique which is generally used for speech recognition is also valid for bird call recognition up to very high extent and could be practically implemented and used. But since mel-frequency cepstrum technique is not that robust in the presence of background noise, a better recording device is needed to be used with better noise filtering algorithms. Instead of taking discrete cosine transform, discrete wavelet transform could be used which would preserve information both in time and frequency domain to get better results considering the fact that the length of the bird call also gives us a feature to distinguish the birds.

As far as distance measuring application using camera is concerned, the results obtained were fairly accurate enough with more than required range of measurement required. But one improvement that could be made is in making the application device independent by having an option to calibrate the sensor values. Since the height of the camera from the ground has to be entered by the user, any new technique could be figured out to make it automatic.

In summary the entire summer internship program was enjoyable and more than satisfactory and helped me improve my knowledge in the area of my interest.

REFERENCE

- Android Sensor Fusion Tutorial,www.thousandthoughts.com.
- Wei-Meng Lee,Beginning Android 4 Web Development.
- http://en.wikipedia.org/wiki/Mel-frequency_cepstrum
- Identiflyer. For the Bird Inc. Retrieved October 24, 2009, from http://www.identiflyer.com/
- http://en.wikipedia.org/wiki/Discrete_Fourier_transform
- http://en.wikipedia.org/wiki/Cross_correlation
- http://ibird.com/Compare.aspx
- Proakis, John & Manolakis, Dimitris. (2005). Digital signal processing. Prentice Hall.
- Nelson, Douglas A. (1989). The Importance of Invariant and Distinctive Features in Species Recognition of Bird Song. The Condor, Vol. 91, No. 1, pp. 120-130