Visualizing Wireless Reception

Eli Ribble
School of Electrical and
Computer Engineering
University of Utah
Salt Lake City, Utah 84112
Email: ribble@cs.utah.edu

***Abstract*—In order to create an intuitive sense of how electromagnetic waves are propagated in the spectrum covered by 802.11 wireless communication, a camera was built to take pictures of local signal strength. The camera consists of a 802.11 spectrum analyzer connected to a directional antenna mounted on a motorized tripod. The tripod is controlled by a computer, which also manipulates and stores the data of the spectrum analyzer. A simple camera is also mounted on the unit so that visible light can be stored along with the signal strength data. Several pictures were then taken in various environments to determine the diffuse and specular properties of common building materials at that frequency.**

I. Introduction

Wireless communication is increasingly a major part of modern life. Many important technologies rely on the reception of electromagnetic (EM) waves including cell phones, wireless internet, GPS, and radio communications. While physicist have given us mathematical expressions that can be used to describe the propogation of EM, raw equations fail to grant an intuitive sense of how a given wave at a given frequency will behave in a real-world environment. In order to gain that intution, it is essential that we visualize how strong an EM signal is. From that intuition, it is possible to create simulations that will allow us to predict the optimal placement of a transciever.

As an example, consider your eyes. Human beings have a great deal of experience with the visual portion of the EM spectrum because they have a physical device capable of turning visual light signals into pictures. From this, we have been able to create startlingly complex and accurate simulations of photon propogations. Indeed, the entire field of computer graphics is the study of how to use computers to simulate the behavior of and interaction between the human eye and visible EM waves.

In this paper we will describe a system by which we created a device which can visualize EM signal strength in the 802.11 wireless spectrum (2.4 - 2.48Ghz). We will lay out what some previous work, hardware was used, software programming challenges, results and ideas for future work

II. Previous Work

Very little research work has been done in this field aside from a few simple hobbyist projects. The work that has been done is entirely devoid of any formal writeup or explanation of results. A few scattered images and anecdotes are available, however.

Fig. 1. Adrian Smith's wireless camera

A. Adrian Smith

Adrian Smith posted a 'wireless camera' project on the Internet that quickly gained popularity in computer hardware hacking circles. The vast majority of websites that mention his work show the image in 1, as well as a short movie that he produced. His wireless camera is made from a coffee can, which provides directional reception, and some stepper motors which allow the antenna to be aimed. The camera takes discrete sample points of the wireless signal strength as it moves, composing a picture based on the direction the antenna was pointing when the sample was taken.

Adrian's work does show a brief picture of his resulting image in the movie he made, which is approximately 30x30 pixels. His website mentions that he wrote the software to control the camera in Visual Basic, and used the wireless access point searching feature of his 802.11 networking card to detect signal strength. Nothing more is said about his work.

B. Other Hobbyists

Several other robotics/wireless connectivity enthusiasts have built other, similar cameras. Again, no writeup of their work exists - generally the project is documented in a short video clip that is uploaded to some hobbyist website. Often the camera will be constructed from some simple robotics components (stepper motors, LEGOS, etc) and a directional antenna (almost universally a metal can).

Fig. 2. Example of a hobbyist wireless camera

2 shows a typical wireless camera built by a hobbyist. Notice the use of a Pringles can for directional reception, legos for robotics and a consumer class router for signal strength detection.

C. Aether Architecture

The only professional entity to create a wireless camera is Aether Architecture. A picture of their wireless camera is shown in 3. The construction of the camera generally follows that of the hobbyist works - a series of cans to provide directional reception and some sort of motorized tripod which can aim the antenna via software. You'll notice that while Aether Architecture does use higher-quality hardware for the motorized portion, they still use empty cans of wasabi for the directional antenna.

Because Aether Architecture created the camera as an art piece for conferences, they spend a great deal more time on visualizing the wireless reception data than any other project. Aether separates out each channel of the 802.11 spectrum into different colors, creating 11 different images or a single composite image of many colors. This allows the camera to handle multiple access points. Also, the camera creates the highest resolution images of any other similar project at approximately 300x170 pixels.

III. Approach

Because every previous project had followed a similar design with some success, we decided it would be wise to base our project on that success. Our design is quite similar to previous designs discussed here, though our results are far more thorough.

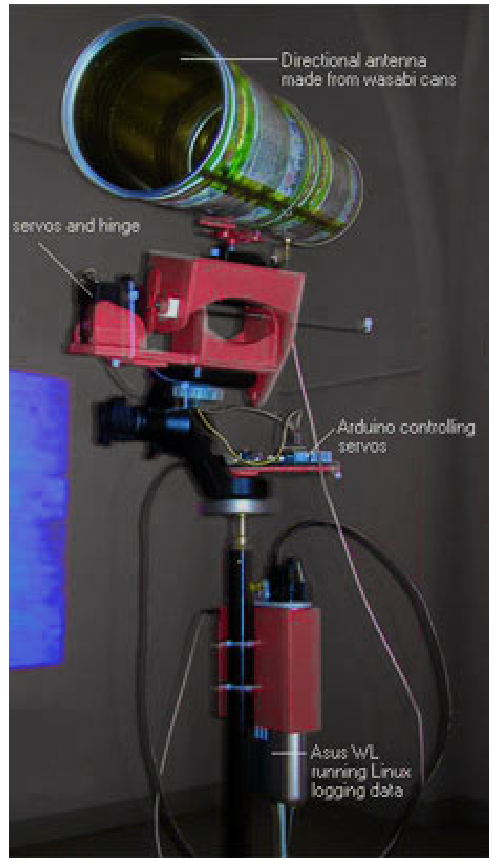

Fig. 3. Aether Architecture's wireless camera

A. Hardware

The first stage oft he project was to decide upon hardware. As this project was funded by a research grant, we thought it wise to use hardware that would be easy to program and robust, and require a minimum of design time on our part. This means that all of the parts to our camera are off-the-shelf products with a known programming interface.

We chose the Eagletron PowerPod for the mechanized portion of the camera. The PowerPod can handle a load of 5lbs and offers 160 of pan and 110 degrees of tilt, with a 53 degree/second rotation speed. The PowerPod interfaces with a computer via USB, and includes static development libraries that allow for simple commands such as an offset movement and an absolute position movement.

For the antenna we opted for a Super Cantenna from Wireless Garden Inc. Every project we had seen thus far had used cans as a directional antenna, and we thought it would be egotistical of us to break the trend. Wireless Garden Inc. claims a gain of 11dB, though actual tests show that it can be as high as 15.

To analyze the incoming wireless data signal, we chose a WiSpy 2.4x, which offers a host of customizable options. The WiSpy 2.4x is actually not a wireless card, but rather a spectrum analyzer, and therefore faster and more thorough than a standard 802.11 wireless card or router. The WiSpy

Fig. 4. The wireless camera we produced

didn't come with drivers, but through communications with MetaGeek, the manufacturer, and looking through the code of an open-source application that uses the WiSpy, we were able to create a driver for it. The WiSpy also communicates via USB.

Finally, we wanted to corroborate wireless data with visual data, so that we would know exactly what the wireless camera was looking at when the picture was taken. We used a Logitech Communicator STX webcam which was taped on to the front of the cantenna. The webcam interfaces via USB, and while we didn't receive drivers directly from Logitech, Windows includes drivers to capture images from any Windows compatible webcam, and we were able to use those.

A picture of the completed camera is shown in 4 for the mechanized portion of the camera.

B. Software

One of the greatest difficulties of this project was in determining how to go about integrating the disparate software libraries we had for the hardware into a single program. The PowerPod library was a static C library. The WiSpy was a C# class which is derived from a standard USB Human Interface Device. The Logitech Camera used a C# library that was based on Microsoft's DirectX.

Overall, since we knew we would need to create a user interface and we would have to use Windows due to the PowerPod static library, we decided to code up the main application in C#. We did this by creating a C++ wrapper around the PowerPod library, then a C# dll call to that wrapper. Since the PowerPod only needed a single command - go to absolute position - this was fairly easy. From there we created C# classes to control the WiSpy and the Logitech camera.

The WiSpy proved to be quite difficult. We were able to contact the manufacturer, who graciously sent us the specification on the USB communications with the WiSpy. Unfortunately, that communication was very atypical, and required a great deal of experimentation to get correct. The WiSpy acts as a Human Interface Device (HID) when plugged into a Windows computer. Generally, HIDs are things such as mice, keyboards, or joysticks. The WiSpy is then configured using a HID Feature Report, which sets its initial configuration. Once the

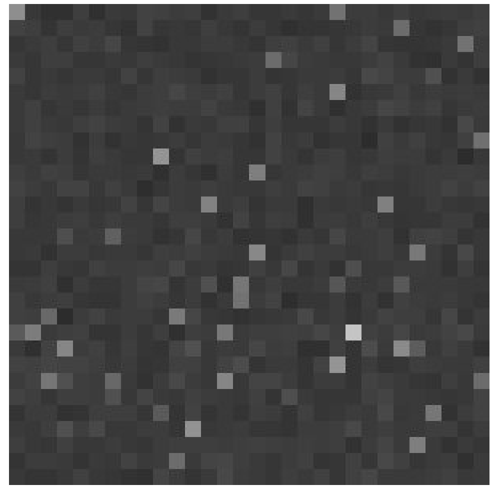

Fig. 5. One of the images taken with our wireless camera

configuration is set, it constantly returns output data reports that must be asynchronisly read back in. This means that a separate thread must be dedicated to handling the WiSpy interface, and that thread's data synchronized with the user interface.

Another difficulty arose in that the PowerPod does not return any sort of value or flag when requested motion is complete. All motion is done asynchronously, so there is no way of knowing if the PowerPod is still moving or not from a software standpoint. This required us to have variable wait times depending on where the PowerPod was sent that wait the maximum time for the PowerPod to complete the requested motion. This considerably slowed down the speed of the camera.

Overall, our software approach required several different threads handling communication with the hardware and display of the incremental results to the screen. This was further made difficult by the lack of multi-thread supporting C# forms. Any time a value needed to be updated on the screen, the thread in question would have to set a flag and wait for the display thread to catch the flag, update the display, then clear the flag.

IV. Results

The results of the project are promising, but incomplete. To speed up the time it takes the camera to capture an image, we only sample every third pixel. This leaves a highly pixelized image. Also, we have not yet found a place where there is sufficiently low ambient 802.11 signal to form control images. All images taken thus far have a high degree of noise due to the many wireless access points in use in the vacinity.

5 shows one of the images produced by our camera. The image took 2 minutes to produce, and is at a resultion of 300x300 pixels. This image was taken on the University of Utah campus, which is heavily saturated with 802.11 signals. Several items are noteworthy about this image. First, there are several very bright spots on the image. This comes about

because wireless data is not sent as a steady stream, but rather as rapid, short burts. If a burst fires at the precise moment that our camera samples the pixel, a bright white spot will appear. Second, there is no overall structure to the image. The image was taken pointing at a wall composed of drywall and steel girders. At this point, it is difficult to know whether these material behave in a specular or a diffuse fashion. What that means is, we do not know yet if a given material will tend to perfectly reflect an EM wave in the 802.11 frequency, or whether it will spread it out. The purpose of this project is to gain an intution for these things, and a great deal of future work will be necessary to determine common properties of building materials.

Another difficulty of our resultant images is that our antenna receives data in a cone shape, not a ray, with varying reception strengths at the different parts of a cone. This means that a given pixel will tend to 'bleed over' to neighboring pixels, because the signal in a given direction can be picked up by the conical antenna at multiple pixels. Future work on this project will involve a great deal of signal processing to remove this bleed-over effect.

Finally, because we optimized the camera for speed of image capture, and not for accuracy at any given pixel, we did not give each pixel sufficient time to resolve to a firm signal strength. This is an artifact of the WiSpy, which is designed to sweep through all of the channels of the 802.11 spectrum and return a signal strength for over 40 samples points in that spectrum. It therefore takes the WiSpy approximately 30 seconds for a given position of the antenna to resolved signal strengths to a stable value. At that rate, an image that is 300x300 pixels would take 750 hours to complete, which is unacceptable.

V. Future Work

Funding for this project is ongoing, and we will continue work over the Summer of 2008 to refine the camera. The first step will be to modify the software to dynamically change the settling time for the WiSpy spectrum analyzer. This will allow the user to specify how many seconds to wait on a given pixel to allow the WiSpy to resolve the measured signal strenght to a given value. While this will cause the images to take a great deal longer to produce, it is necessary to allow us to get extremely accurate pictures from which to glean material properties in the 2.4Ghz range.

The next major work on the camera will be to add signal processing, as mentioned previously, to remove the pixel bleed over. Experiments will be conducted to determine the exact size and shape of the conical reception volume from the Cantenna. From there, image sharpening techniques will be used to de-blur the resultant image.

In order to produce control images, we will also construct an antenna which constantly broadcasts a single signal at 2.45Ghz. This is to overcome the data bursting problem associated with standard computer networks. Since network communication is far from constant, it's impossible to know at any given moment whether a wireless access point is broadcasting or not. A custom emitter will allow us to work around that, and will be instrumental in determining the size and shape of the conical reception volume, as mentioned above.

Finally, with all of that done, we will increase the picture resolution to utilize the full range of motion of the PowerPod. This will allow us to take full screen images (900x900) of wireless reception. With signal processing and a constant emitter, we hope then to use these high resolution images to determine if a given material is diffuse, specular, transparent, or translucent in the 802.11 range.

VI. Conclusion

We have herein demostrated a method for visualizing 802.11 wireless networking reception using a combination of directional antenna, software-controlled mechanical tripod and spectrum analyzer. While the results were not overwhelmingly clear, from them were are able to plot out a course for future work whereby we may enhance our understanding of material properties at the 2.4-2.48Ghz range. It is our expectation that future research in this area will allow us to build accurate simulations of EM wave propogation in any given setting for which material measurements have been made. From this we expect to gain an intuition that will allow us to enhance wireless communication through informed decisions about transceiver positioning.

Acknowledgment

The authors would like to thank the University of Utah School of Computing for providing research facilities and administration. Also, the Army Research Center and the people of the United States of America for funding to persue this project.