Calhoun

Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

Faculty and Researcher Publications   Faculty and Researcher Publications

2012-04

The Android Smartphone as an Inexpensive Sentry Ground Sensor

Schwamm, Riqui

Monterey, California. Naval Postgraduate School

The Android Smartphone as an Inexpensive Sentry Ground Sensor

Riqui Schwamm, Neil C. Rowe

Cebrowski Institute, U.S. Naval Postgraduate School, 1411 Cunningham Rd., Monterey, CA 93943

ABSTRACT

A key challenge of sentry and monitoring duties is detection of approaching people in areas of little human traffic. We are exploring smartphones as easily available, easily portable, and less expensive alternatives to traditional military sensors for this task, where the sensors are already integrated into the package. We developed an application program for the Android smartphone that uses its sensors to detect people passing nearby; it takes their pictures for subsequent transmission to a central monitoring station. We experimented with the microphone, light sensor, vibration sensor, proximity sensor, orientation sensor, and magnetic sensor of the Android. We got best results with the microphone (looking for footsteps) and light sensor (looking for abrupt changes in light), and sometimes good results with the vibration sensor. We ran a variety of tests with subjects walking at various distances from the phone under different environmental conditions to measure limits on acceptable detection. We got best results by combining average loudness over a 200 millisecond period with a brightness threshold adjusted to the background brightness, and we set our phones to trigger pictures no more than twice a second. Subjects needed to be within ten feet of the phone for reliable triggering, and some surfaces gave poorer results. We primarily tested using the Motorola Atrix 4G (Android 2.3.4) and HTC Evo 4G (Android 2.3.3) and found only a few differences in performance running the same program, which we attribute to differences in the hardware. We also tested two older Android phones that had problems with crashing when running our program. Our results provide good guidance for when and where to use this approach to inexpensive sensing.

This paper appeared in the Proc. SPIE Conf. on Unattended Ground, Sea, and Air Sensor Technologies and Applications XIV, Baltimore, MD, April 2012.

Keywords: Smartphone, Android, sensors, sentry, audio, footsteps, brightness, testing

1.   INTRODUCTION

A key military challenge for ground sensors is detection of approaching people in isolated areas for detection of curfew and border violations, criminal activity, and improvised explosive device (IED) emplacement[1]. This is a separate problem from detection of suspicious activity in crowds which requires different techniques. Isolated person detection would seem easy for fixed-position cameras since differences between images at successive times are due to moving objects and are easy to find, and moving people look quite different from moving vegetation and other objects[2,3,4]. But there are several problems with continuous surveillance using cameras:

·   Cameras suffer from occlusion problems from vegetation, walls, vehicles, and other people. They do not work well in forests or near walls.

·   Cameras usually do not work under poor lighting conditions such as at night or in bad weather. While infrared cameras can be used, they are more expensive, provide a less well defined picture, and can be defeated by particular kinds of clothing.

·   Cameras are expensive compared to most nonimaging sensors. They require a focal plane with a sufficient number of pixels to be useful, and that costs money.

·   Cameras are relatively large sensors since they require a lens which needs at least a few inches of diameter to be useful.

·   Cameras collect large amounts of data. That makes their data difficult to transmit via wireless connections as is desirable. The data can be processed to reduce its size, but this generally requires sophisticated software since the best methods vary with subjects and lighting conditions, and sensor nodes may not have the capabilities to run such software.

We are exploring nonimaging sensors for this task of automated sentry monitoring. Nonimaging sensors can cover wide areas better than cameras can with today's networking technology[5]. The idea is that once people are detected, security personnel can be alerted, or one of a few cameras can be turned on and aimed at the location so that many fewer cameras are needed.

1.1 Previous work

Previous work of ours explored specialized nonimaging sensors for this task. We experimented first with Crossbow sensors[6]. Initial work focused on magnetic and infrared sensors. Both were imprecise and temperamental. These infrared sensors generally did recognize people within 15 feet, but it was not certain. The magnetic sensors were a little better at detecting ferromagnetic material, but this is of limited use for most surveillance activities since most people are not carrying appreciable amounts of ferromagnetic material. Subsequent work[7] assigned probability distributions to the reports of a set of infrared, magnetic, and acoustic sensors, and derived cumulative probabilities of a person at a grid cell at a particular time. This evidence fusion improved the tracking accuracy for people, something important for noticing suspicious behaviors involving acceleration. Previous work[8,9] suggests a number of clues to suspicious behavior that can be obtained from tracking overall body location alone. Our own experiments[5] showed that nonzero accelerations were by far the best clue in detecting suspicious behaviors related to IED emplacement, and they are important in many kinds of criminal activity like theft in public places, so tracking needs to be good enough to detect accelerations.

The set of 16 Crossbow sensors cost $10,000 in 2007 to cover about 50 square meters in our experiments. Though subsequently the price decreased, it is still expensive. So we subsequently we experimented with 8 sets of Phidgets sensors (www.phidgets.com) costing a total of about $1,000 in 2010 without wireless networking and covering an area of about 100 square meters[10]. These had broad-range infrared sensors for motion detection, photocell-type infrared sensors for short-range detection, short-range sonar sensors, light-intensity sensors, magnetic sensors, vibration sensors, and pressure strips (to detect people walking on them). We supplemented these with microphones and our own footstep-detection software. We conducted a variety of experiments including both suspicious and nonsuspicious behavior in our sensor field. In these experiments, the infrared motion detector, sonar, and microphones performed the best at both detecting people and noticing suspicious behavior. The light sensor also worked well but need to be significantly blocked by optical filters when used outdoors because it saturated easily under bright conditions.

To reduce the cost of sensing still further, we can look to today's commercial "smartphones" providing telephone and Internet capabilities. These cost around $100 each with a contract and many military personnel are carrying them anyway for no added expense. Smartphones have many sensors built-in, including light, vibration, and magnetic sensors as well as microphones and cameras, and they provide built-in networking capabilities. Thus they would seem to provide more capabilities for a similar price to the Phidgets sensors. Seeing as how the infrared and light sensors were especially useful in the Phidgets experiments, as well the microphones, those two kinds of sensors would seem the highest priority to exploit on a commercial wireless device. Unfortunately, there is no sonar as was helpful with the Phidgets, but the other two kinds of sensors useful previously are present.

Previous experiments by us tested the Iphone as a potential sensor platform in this manner[11]. First tests focused on the vibration/orientation sensor because it sounded as if it could be helpful in detecting footsteps. But results were disappointing. Pedestrians needed to be very close to the phone and needed to make strong footsteps for them to be detected, and many pedestrian transits could not be detected above the background noise. We tried to enhance sensitivity by attaching the Iphone to a stake inserted into the ground, to better pick up the lower frequencies that travel better through the ground since these are better clues to footsteps, but this did not help much. We concluded that the vibration sensor in the Iphone is only useful for detecting gross motions like changing the orientation of the phone itself. On the other hand, the microphone in the Iphone was better at detecting footsteps. We experimented with several ways of filtering the acoustic signal for footsteps and found a good approach focused on 10-500 hertz frequencies at intervals of 0.4 to 1.0 seconds apart which was helpful for the work to be described. However, a key issue for us was the unfriendly environment for development of non-Apple Iphone software.

The Android platform is more developer-friendly and seemed a better choice for long-term development. Examples of use of the Android for similar sensing applications include medical monitoring systems for at-risk patients[12,13], systems for detection of traffic accidents[14], and systems for early warning of earthquakes[15]. All of these are similar to our sentry task in involving sensing of rare but easy-to-detect safety-related events that just need a sensor in the right place at the right time, with relaying of a report of that event to a central collection point.

2. EXPERIMENTAL SETUP

The task we addressed was automated sentry duty where a set of smartphones are set up around a military position to monitor for approaching people. Pictures will be automatically taken of anything that looks like a person and transmitted to a central collection site. The Android's light sensor is on its front and its better camera is on its back; its front one provides only 1.3 megapixels and its driver differ significantly between manufacturers unlike the other camera. So it was found best to orient the Android vertically facing forward with the camera lens pointing backward towards a mirror, with the mirror angled to reflect an approaching person to the lens. This was because the light sensor, as a diffuse sensor, is more sensitive to blocking of its field of view than the camera is.

For our experiments, we used the Android Development Environment (developer.android.com/sdk/index.html) with the Windows Vista Business SP2 32-bit operating system, the Eclipse Hellios Service Release 2 development environment (www.eclipse.org), and the Level 7 Android API (developer.android.com/reference/packages.html). Two Android smartphones were used for this project to test the hardware independence of the sensor capabilities. One was the Motorola Atrix 4G (Android 2.3.4) on an AT&T contract with a Nvidia Tegra 2 Dual Core 1Ghz CPU, 1 GB of memory, 16 GB of internal storage, a KXTF9 3-axis Accelerometer sensor, AK8975 3-axis Magnetic field sensor, AK8575 Orientation sensor, ISL29030 Proximity sensor, ISL29030 LIGHT sensor, a Gravity Sensor, a Linear Acceleration Sensor, and a Rotation Vector Sensor. The other Android was a HTC Evo 4G (Android 2.3.3) Sprint Unlocked with a Qualcomm QSD 8650 1Ghz CPU, 1 GB of memory, 8 GB of internal storage, a BMA150 3-axis Accelerometer, AK8973 3-axis Magnetic field sensor, AK8973 Orientation sensor, CM3602 Proximity sensor, CM3602 Light sensor, Gravity Sensor, Linear Acceleration Sensor, and a Rotation Vector Sensor. Our programs were written in Java using ideas from several authors[16,17].

The first thing that needed to be tested was the usefulness of each sensor on the smartphones. Each sensor was tested to determine which one would be the most useful for detecting footsteps. Code from an open-source Android project (https://bitbucket.org/nonninz/android-sensor-logger/overview) was used to dump raw sensor data into text files. The application provided a way to record all sensor data simultaneously in real time. The original project code only supported the accelerometer, orientation and magnetic field sensor. The code was later modified to include support for light, proximity and audio recording. All five sensor data (saved as .txt file) and audio recording (saved as .wav files) was recording real-time during the test runs. Each text and audio file was time stamped during the experiment.

A time-lapse camera application[18] was used separately to test how well the built-in camera can capture the environment. The idea was to create an application that uses a sensor to trigger the camera. The results were very promising. The camera was able take multiple snapshots at a relatively fast pace of one per half second. Due to the time it takes for the smartphone to process the image, running it any faster seems to cause the program to become unstable and crash.

Several test runs were done on several different surfaces (concrete, dirt, grass). The smartphone was placed directly on the ground or on a stand during the test runs. The subject would start from one end (10 feet away from the smartphone) and walks pass it until the subject reaches the other end (10 feet away on the other side). The subject would pass the smartphone at a very close proximity (one to two feet).

Figure 1: Experimental setup.

The sensors that yielded the best results were light, vibration (accelerometer) and audio (microphone). The built-in microphone was able to clearly record footsteps. The footsteps can clearly be identified in audio editing software such as Audacity (audacity.sourceforge.net).

Figure 2: Example footsteps recorded by the audio processing program.

The light sensor was not very useful indoors, but very useful outdoors. When the subject walks passed the smartphone the values fluctuate enough to indicate a trigger. The only problem is the subject has to cast a shadow onto the sensor for it to react. It can be very tricky to set the right threshold for the sensor to trigger. The camera was very useful to capture the environment and subject. It can take two pictures per second. It cannot handle continuous snapshots. One snapshot per half a second seems to be the limit. We could get the effect of the light sensor by taking a picture and averaging its brightness, but there is latency in both the camera and required processing, and this would be too slow to provide the necessary rapid alert capability we needed.

Other sensors such as the proximity, magnetic and orientation sensor did not return any useful results. Despite being an infrared sensor, the proximity sensor can only detect objects that are closer then 3cm. The magnetic sensor only detects objects that have a magnetic field, not ferromagnetic materials as more usefully with the Crossbow sensors, and the target must be holding an electronic device close to the smartphone or it will not trigger. It may be useful in some situations, but for this project it was not used in the final application. The orientation sensor worked very well to detect tilting and angle of the smartphone. However, like the magnetic sensor it is not very useful for this project.

From the above test results it seems best to use the sensor as a trigger to take a snapshot with the camera. This image could then be transmitted to a collection point, and humans could make a judgment based on appearance of whether the moving object was suspicious (or an animal, a fallen object, etc.).

3. RESULTS

Figures 3 and 4 show some sample pictures taken by the smartphone.

The indoor surfaces tested were linoleum, carpeting, hardwood, and smooth concrete. The outdoor surfaces tested were concrete, dirt, gravel, and grass. The smartphone was setup on the ground with a harness to prop the phone up. The subject will walk past the smartphone from left to right and then back again. A total of ten passes were made in front of the smartphone. Two different distances were used during the test run (4-feet & 10-feet).

4 feet was the minimal distance necessary for taking a picture. Any closer and the camera will only be able to capture the subject's lower half of the body. The point of the experiment was to see if the phone can capture the subjects face. The smartphone could have been tilted back further to allow for closer ranges, but this would prevent the camera from fully capturing the surrounding environment.

After a few experiments it became apparent how unreliable the vibration sensor could be. The vibration sensor seemed to work relatively well in close proximity, but the results were very inconsistent. For instance, on one test run it would trigger 50% of the time, but on another test run it would not trigger at all. On one occasion the vibration sensor in the HTC Evo 4G would trigger constantly while the Atrix 4G vibration sensor would not trigger at all on the same transits. Because of these results we focused on using just the audio and light sensor for this project.

Adjustment of the thresholds for triggering the camera was critical to performance as our results below attest. We discovered right away that it is essential to turn off the clicking sound produced by the camera when taking a picture because this produced sufficient low frequencies to trigger itself continually. Another important issue is latency of the camera function. It took an average of a half a second for the camera to take a picture once the software ordered it to do so. This means that a fast-moving subject could be out of camera range by the time the camera triggered. This could be a problem with detecting vehicles, but was not a problem with detecting pedestrians since these will generally follow very straight paths remaining within a few feet of the nearest-approach distance for several seconds. However, it did generally mean that we often got the side or rear of subjects rather than their faces.

Our measures of performance were recall and precision. Recall for this application was the percentage of transits of the smartphone that resulted in a picture of the transiting subject. Precision was the percentage of pictures taken by the smartphone that showed the transiting subject.

3.1 Camera with audio trigger

The audio trigger is very simple. When the application starts it creates a baseline value. The application takes 8000 samples per second for the first 3 seconds (24000 samples total) and set it as the baseline. A threshold value is then added to the baseline. If the real time value of the sensor is higher than the threshold it will trigger the camera to take a snapshot. The threshold value for audio was 25 for an average value of 0-9.

Creating a baseline value in the start of the application helps minimize false positives. Subtle noises like leaves rustling in the wind do not trigger the sensor. However, any loud noise can trigger the sensor. This is not much of a problem indoors, but it did occur outdoors with car and airplane noise (our campus is beneath an airport approach).

At first, the various surfaces (carpet, dirt, gravel, grass) did not return any good results. All surfaces returned 40-50% for both the recall and precision. It quickly became apparent that the threshold value was set too high. After the threshold value was set to half of the initial value it returned better results.

At four feet away, the best results were achieved on the concrete surface (outdoor). Many test runs resulted in a 100% recall rate. Precision, on the other hand, was still at 40-50%. The lower threshold value caused the camera to trigger more often, which resulted in a higher rate of false positives. Carpeted surfaces (indoor) returned the worst results. It appeared that the carpeted surface would absorb most of the sound and the camera would not trigger. Even if the threshold was set relatively low it would only trigger once or twice during the test run.

At ten feet away, the concrete surface (outdoor) still returned good results from 10-feet away. There was one instance where the recall and precision were very low (30%/43%). But for the most part the recall was 70-90% and the precision was 40-50%.

Table 1: Results with just an audio trigger.

Distance	Trigger	Capture	Recall	Precision
4 feet	29	12	100%	41%
4 feet	32	10	100%	31%

4 feet	20	10	100%	50%
10 feet	29	12	90%	53%
10 feet	32	10	30%	43%
10 feet	20	10	70%	16%
10 feet	9	7	70%	37%

3.2 Camera with light-intensity trigger

A threshold value is hard coded into the application. If the real time value of the sensor is lower than the threshold it will trigger the camera to take a snapshot. 10000 was used outdoors and 10 indoors (for average sensor values of 16000-19000 and 30-500 respectively).

The sensor will only trigger if the subject casts a showdown onto the sensor. If the light source is not strong enough it will not trigger the sensor. This can be a problem in environments with soft lighting and/or multiple light sources.
Due to the extremely limited nature of the light sensor no extensive testing was done for this sensor. Since the camera would only trigger with light, it was unnecessary to test various surfaces. The light sensor would only trigger if the subject casts a shadow onto the smartphone's light sensor. This would mean that the sensor would not trigger on the 10-feet mark, simply because the subject would be too far away to cast a shadow. However, it did return very good results compared to the vibration sensor. As long as there was a good light source the light sensor would trigger at a relatively high rate. The test two test runs below were done on a very sunny day where the subject's shadow would cover the entire smartphone each time the subject walked by. The sensor did not trigger with soft lighting indoors.

Table 2: Results with just a light-intensity trigger.

Trigger	Capture	Recall	Precision
20	17	100%	85%
10	10	100%	71%

3.3 Camera with audio and light trigger

This is a combination of the two previous applications. A baseline for the audio trigger is used and a simple threshold value comparison is used for the light sensor. A snapshot is taken when one of the sensors is triggered. If both sensors are trigged at the same time only one snapshot is taken (to avoid duplication). Two surfaces were tested, an outdoor sidewalk of concrete, and an indoor one of linoleum.

The combination of the two sensors returned decent results for the recall. The added light trigger caused the camera to take more pictures which resulted in a much lower precision. Unfortunately, the combination of the two sensors did not improve the overall test results.

Table 3: Results with both audio and light-intensity triggers.

Surface	Distance	Trigger	Capture	Recall	Precision
concrete outdoor	4 feet	45	12	100%	27%
concrete outdoor	4 feet	18	7	70%	39%
concrete outdoor	4 feet	34	10	100%	29%
linoleum	4 feet	16	3	30%	19%
linoleum	4 feet	17	9	90%	53%
linoleum	4 feet	14	10	100%	71%
linoleum	4 feet	14	7	70%	50%
linoleum	4 feet	15	8	80%	53%

Many new surfaces and environments were tested for this last portion. The audio sensor seemed to be the most reliable sensor at this point. Different environments were tested to see how well the audio sensor would hold up.

Table 4: More results with both audio and light-intensity triggers.

Surface	Distance	Trigger	Capture	Recall	Precision
linoleum	9 feet	9	6	60%	67%
linoleum	9 feet	20	5	50%	25%
hardwood	10 feet	35	20	100%	57%
hardwood	10 feet	33	16	100%	48%
hardwood	10 feet	35	18	100%	51%
concrete indoor	4 feet	35	10	100%	29%
concrete indoor	4 feet	24	11	100%	46%
concrete indoor	4 feet	19	9	90%	47%
concrete indoor	10 feet	12	4	40%	33%
concrete indoor	10 feet	22	10	100%	45%
concrete indoor	10 feet	16	11	100%	69%

3.4 Differences between smartphone manufacturers

Our tests primarily tested two platforms, the Motorola Atrix 4G (Android 2.3.4) and HTC Evo 4G (Android 2.3.3). Although they both had the same kinds of sensors (seven plus a microphone and a camera), the appropriate thresholds for light and audio were different and required separate adjustments. Speed of the program was similar even though the Motorola was a dual-core processor, from which we conclude that our program did not tax the processors very much.

On the Atrix 4G, the application installed and ran without a problem on this phone. The test data shown above is from this phone. The HTC Evo 4G was similar, except it had some lag issues when the camera would trigger frequently but it would still capture enough information during the test runs. On some rare occasions the phone would freeze up during the rapid camera trigger.

Other Android phones tested for compatibility and performance for the application were the HTC G1 (Android 1.6), the HTC Droid Eris (rooted - Android 2.3.7: CyanogenMod), and the Samsung Google Nexus S (Android 2.3). On the HTC G1, the application would not run at all; it would properly install but would always crash during the launch process. Even after recompiling the application to API Level 4 (down from API Level 7) it would not run properly. This problem could be related to the slow processor for the phone (ARM 11 528MHz). On the HTC Droid Eris, the application installed properly and would run for a while but would often crash after a camera trigger. This could be due to the low processing power of the phone (Qualcomm MSM7600, 528MHz). There were only a few successful runs where the phone would not crash. On the Samsung Google Nexus S, the application installed and ran without a problem, and provided similar performance to the Atrix 4G without lag issues or crashing of the application. From these tests we conclude that our application does require relatively recent Android hardware to work properly.

4. CONCLUSIONS

The audio sensor in the Android smartphones proved to be useful enough for detecting suspicious behavior. The test runs returned promising data in various environments. The smartphone was able to capture the subject reasonably quickly. False positives are still an issue, but this did not affect the overall performance of the smartphone or the recall value. It may be possible to improve precision by running additional tests with different threshold values. The light sensor should be useful in certain outdoor environments. Even though combining audio and light-intensity information lowered precision in our tests, it should be important for detecting subjects that do not make much noise. The application can be made into an .apk package for a wide range of devices. Both smartphones worked well for this experiment and returned good results.

REFERENCES

[1] Hackwood, S., and Potter, P., "Signal and image processing for crime control and crime prevention," in Proc. Intl. Conf. on Image Processing, Kobe, Japan, 3, 513-517 (1999).

[2] Barbara, D., Domeniconi, C., Duric, Z., Filippone, M., Mansfield, R., and Lawson, E., "Detecting suspicious behavior in

surveillance images," Proc. International Conference on Data Mining Workshops, Pisa, Italy, (2008).

[3]   Gibbins, D., Newsam, G., and Brooks, M., "Detecting suspicious background changes in video surveillance of busy scenes," Proc. 3rd IEEE Workshop on Applications of Computer Vision, December, 22-26 (1996).

[4]   Rowe, N., and Chan, A., "Rating whole-body suspiciousness factors in automated surveillance of a public area," Proc. Intl. Conf. on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, NV, (2011).

[5]   Valera, M., and Velastin, S., "Intelligent distributed surveillance systems: a review," *IEE Proceedings – Vision, Image, and Signal Processing*, 152, 192-204 (2005).

[6]   Sundram, J., Sim, P., Rowe, N., and Singh, G., "Assessment of electromagnetic and passive diffuse infrared sensors in detection of suspicious behavior," Proc. International Command and Control Research and Technology Symposium, Bellevue, WA (2008).

[7]   Rowe, N., Reed, A., and Flores, J., "Detecting suspicious motion with nonimaging sensors," Proc. Third IEEE International Workshop on Bio and Intelligent Computing, Perth Australia (2010).

[8]   Panangadan, A., Mataric, M., and Sukhatme, G., "Detecting anomalous human interactions using laser range-finders," in Proc. Intl. Conf. On Intelligent Robots and Systems, 3, 2136-2141 (2004).

[9]   Wiliem, A., Madasu, V., Boles, W., and Yarlagadda, P., "Detecting uncommon trajectories," Proc. Digital Image Computing: Techniques and Applications, Canberra Australia (2008).

[10] Rowe, N., Reed, A., Schwamm, R., Cho, J., Flores, J., and Das, A., "Networks of simple sensors for detecting emplacement of improvised explosive devices," [F. Flammini (Ed.) *Critical Infrastructure Protection],* WIT Press, New York, 241-254 (2012).

[11] Young, P., Rowe, N., Anderson, T., and Singh, G., "A mobile phone-based sensor grid for distributed team operations," Symposium on Sensor and Data Fusion of the Military Sensing Society (MSS), Las Vegas, Nevada US, (2010).

[12] Cardei, M., Marcus, A., Cardei, I., and Tavtilov, T., "Web-based heterogeneous WSN integration using pervasive communication," Proc. IEEE 30th Intl. Performance Computing and Communications Conference, November, 1-6 (2011).

[13] Fang, S.-H., Liang, Y.-C., and Chiu, K.-M., "Developing a mobile phone-based fall detection system on an Android platform," Proc. Computing Communications and Applications Conference, 143-146 (2012).

[14] White, J., Thompson, C., Turner, H., Dougherty, B., and Schmidt, D., "WreckWatch: automatic traffic accident detection and notification with smartphones," *Mobile Networking Applications*, 16, 285-303 (2011).

[15] Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, K., and Krause, A., "The next big one; detecting earthquakes and other rare events from community-based sensors," Proc. 10th Intl. Conf. on Information Processing Sensor Networks, 13-24 (2011).

[16] Komatineni, S., [*Pro Android 3]*, Apress, New York (2011).

[17] Steele, J., [*The Android Developer's Cookbook: Building Applications with the Android SDK]*, Addison-Wesley Professional, New York (2010).

[18] Van Every, S., [*Pro Android Media: Developing Graphics, Music, Video, and Rich Media Apps for Smartphones and Tablets]*, Apress, New York (2010).