USENIX Association

Proceedings of HotOS IX: The 9th Workshop on Hot Topics in Operating Systems

Lihue, Hawaii, USA
May 18–21, 2003

Sensing User Intention and Context for Energy Management

Angela B. Dalton and Carla S. Ellis
Department of Computer Science
Duke University
Durham, NC, 27708
angela,carla@cs.duke.edu

Abstract

Sensors are emerging as a key area of interest in operating systems research, with a main focus on sensor networks. Turning the relationship around, we propose the use of low-power sensors as tools for improving OS-based energy management. Using sensors to detect user intent and physical context we can more directly match system I/O behavior to user needs. FaceOff is a prototype display power management system designed as a test bed and proof-of-concept.

1 Introduction

Managing energy as a resource is key to the future ubiquity of mobile computing systems. Reducing power consumption is also a major challenge in the design of mobile systems that extends beyond advances in battery technology and low-power circuit design. Energy efficient computer systems have broad environmental and economic implications [4, 5, 16]. *This position paper focuses on using sensors to leverage physical context and user intent to reduce a system's energy consumption.* We illustrate this idea with a case study on managing the display.

System level energy management approaches are currently tied almost exclusively to process execution. We believe there is ample opportunity for reducing a system's energy consumption by more directly matching the system's I/O behavior to the user's own behavior. Consider the display, a component that presents unique difficulties for power management and typically represents the largest power consumer after the CPU [9, 15]. The display exists solely for the purpose of user interaction and therefore it is only necessary when someone is looking at it. There are many times when a user may turn his attention away from a computer display, perhaps to answer a phone call or get a cup of coffee. There are also scenarios in which the display is only used briefly or not at all for a particular application. For example, someone using a computing device to play music may only interact with the device to select a song. Each time a song is selected, the user interaction would cause a timeout-based display power management scheme to reinitialize the timer. Similarly, someone may use a Personal Digital Assistant as a travel alarm clock. When the alarm sounds, there may be no need to look at the display, only to press a button in acknowledgement and turn off the alarm. However, the PDA display is turned on by the user pressing a button and would remain on for a timeout period. In these cases, turning off the display immediately or never turning it on rather than waiting for a timeout period would reduce energy consumption.

On the other hand the conventional timeout scheme which is based on lack of user input may be too aggressive for some applications. A user reading an electronic book or examining a web page with complex content might experience the annoying behavior of the display timing out. The same problem exists for a user watching a video or an automated slide show. These are situations where the user interaction is dependent on the display but is not tied to user interface input events.

In the next section we introduce our case study of a sensing system used for power management. We then describe the architecture of the system and the design of our prototype. In Section 4 we present energy measurements of our prototype to justify its potential. This is followed by a section speculating on additional roles of sensors. Section 7 outlines future work while section 6 discusses related work.

2 Architecture

As a case study we evaluate a power management method that uses a web cam mounted to the display of a laptop as a sensor. The camera periodically captures images and a face detection algorithm determines the presence or absence of a user looking at the display. Our research is an initial investigation into using sensors combined with computer vision techniques to enhance display power management. There were several open questions we hoped to address with our evaluation. First, computer vision in general is extremely computationally intensive. Are there optimizations we can use based on the constraints of the specific problem that will make the computing costs small enough to justify? Similarly, can the optimized algorithm run quickly enough to appear seamless to the user? Can this method produce a measurable reduction of energy consumption in the system even after accounting for the added computing energy costs and the energy consumed by the camera? Finally, are there specific applications to which this method is particularly well-suited and what are the situations when it is not appropriate?

We are designing a display power management system called FaceOff as a proof-of-concept and a test bed for taking energy measurements as well as obtaining more subjective user feedback. The FaceOff architecture is simple and leaves significant room for optimizations to maximize energy savings.

The FaceOff design consists, on a high level, of three main components: image capture, face detection, and display power state control (see Figure 1). The program periodically wakes up and calls the image capture component. The image capture mechanism obtains a still image from a camera and sends the image to the face detector for analysis. The face detector

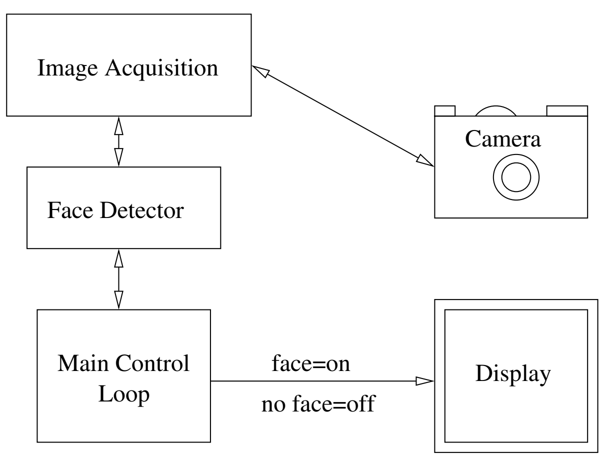

Figure 1: A diagram showing the components of the FaceOff Display Power Management System.

processes the image using an optimized algorithm and returns the Boolean value of true if a face is detected and false if no face is detected. The display power is controlled using the ACPI (www.acpi.info) interface to change the video device power state. When no face is detected the program sets the device power state of the video to a sleep state.

3 Initial Prototype

We are building the FaceOff prototype on an IBM T21 Thinkpad running Red Hat Linux. The camera is a color Logitech QuickCam 3000 web cam that connects via USB to the laptop with an average measured power consumption of 1.5W. The display power states are defined in the ACPI specification and supported by both the laptop hardware and the operating system. On this laptop, the display consumes approximately 8.5W.

Our FaceOff prototype consists of a loop that captures an image from the camera once a second. The image is saved to a file that is processed by the face detection module. An obvious optimization to the prototype is to eliminate use of the disk for storing the image acquired from the camera. However, at this point we are logging information to the disk for later analysis. Currently, the face detection module consists of a skin color detector that looks for a large central area of skin color in the image. Skin color detection was selected as a fast and fairly simple method for the initial prototype, however more accurate fast face detection methods exist and are part of our longer term plans for the FaceOff system. We are integrating ACPI based power state control into the prototype.

4 Evaluation

In order to evaluate the potential for our display power management method to achieve energy savings, we examine usage scenarios that should benefit from this approach. We measure the energy characteristics of the system assuming a best case face detector and compare that to the energy characteristics of the default timeout-based power management scheme on the same system with a typical timeout of five minutes. Current measurements were taken from a multimeter on the laptop's power supply with the laptop's battery removed to eliminate changing effects. Voltage measurements were taken for one case to verify that the voltage remains constant throughout the experiment.

In this section we present three scenarios: a large network transfer, a long, computationally intense process, and playing an mp3 song. The scenarios were selected because they offer a comparison between FaceOff and the default display power management scheme in which the timeout intuitively does not capture the user's behavior well. It is likely that a user might turn away from the computer after initiating a large file transfer, beginning a large compilation, or starting to play a song. The FaceOff system immediately turns off the display, however it continues to perform the image polling and analysis, turning on the display when the user returns. The default system will not turn the display off until the timeout period expires, but it does not have the disadvantage of the FaceOff system overhead.

The first set of experiments measured the energy consumption of the laptop during a large network transfer. The transfers were performed using a wireless network adapter on an internal network with no other concurrent traffic. The measurements were taken assuming the best case in which the user would initiate the transfer and look away, returning as soon as the transfer completed. This application represents the case where FaceOff overhead is not expected to affect performance. We measured the total energy consumed during the tests as well as the time the network transfer took to complete. The FaceOff technique used an average of 29.5% less energy than the default, showing a significant improvement. Table 1 shows a comparison of the energy and time characteristics. Figure 2 shows traces of the power over time for one run each of the network transfer with and without FaceOff.

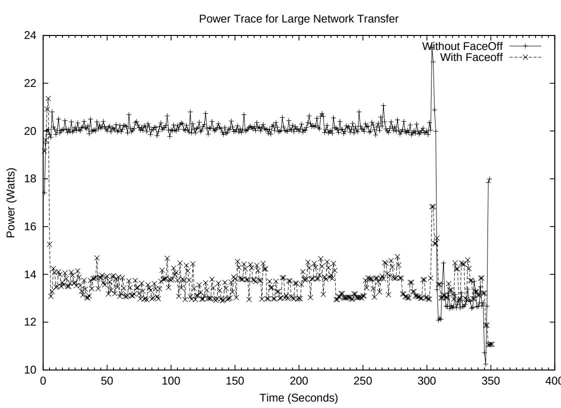

Figure 2: Power traces for large network transfer.

The second experiment measured the energy consumption of a laptop performing a computationally intensive task. In this case, the task was to compile the Linux kernel. No other programs were running on the machine except in the case of the FaceOff measurement the FaceOff prototype was running. This captures the competition for resources imposed by FaceOff. Again, we assumed the best case of the user initiating the compilation, leaving and returning immediately upon completion. The FaceOff technique resulted in an average power savings of nearly 12%. The results of the experiment reflect the fact that the default timeout-based power management system turned off the display close to halfway through the compilation, reducing overall energy consumption. Figure 3 shows sample traces of the power over time for the compilation process with and without FaceOff. The increase in completion time using each method for both experiments was insignificant.

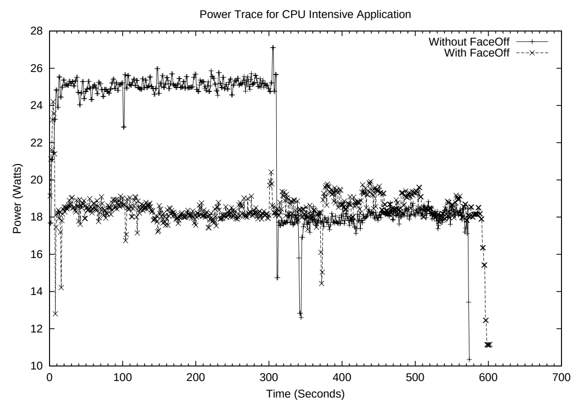

Figure 3: Power traces for Linux kernel compilation.

The third experiment, playing an mp3 song, was primarily a validation that FaceOff would cause no noticeable effect on the playback of the song. In addition, such a scenario highlights one in which the default timeout mechanism will never cause the display to turn off. The song we used in the experiment lasted 4 minutes and 11 seconds, and played with no noticeable effect when the FaceOff prototype was running. The average energy used in the default case was 4,714 Joules, versus 3,403 Joules with FaceOff, a 28% energy savings.

While the experiments we have presented provide a basis for our argument of using context awareness and user intent for power management, we believe that technological trends provide further weight in our favor. The web cam we used for the prototype system requires more power than we would anticipate a camera in any production version of the system to need. Low power, tiny CMOS cameras are now available that can be embedded into computer systems and consume as little as 20mW maximum power[7, 11]. Compared with the 1.5W average power consumption we measured for the prototype's web cam, clearly the overhead can be much lower.

Network Transfer	FaceOff	Timeout
Time (s)	351.3	348.6
Energy (J)	4791.2	6795.4
Kernel Compile	FaceOff	Timeout
Time (s)	603.5	575
Energy (J)	11023.7	12506.85

Table 1: Energy and Time Comparisons

5 More Roles for Sensors

Although our initial measurements show that the possibility exists to save energy using our method of power management, the method has significant overhead taking away from the benefits. Most notably, as realized in the initial prototype, the image capture and face detection are continuing costs, whereas an extended idle period incurs no overhead once the timeout expires.

The first observation is that people radiate thermal energy and are detectible with small motion sensors. If no person is present, no face will be detectable and therefore we do not need to capture images or run the face detection computation. Extremely small, low power pyro-electric sensors are available that can detect even slight human motion [2]. Integrating such a sensor into our system would allow the camera to be powered down until movement triggers the sensor. A conservative approach designed to minimize the delay waiting for the display to turn on would immediately turn on the display and capture an image. The face detection method would then take over until no face is detected and no motion is present. A similar optimization would be to use a touch sensor in the laptop wristrest or on the edges of a handheld device. The observation in this case is that if a user's hands are on the keyboard or holding a PDA in a particular orientation, even with no input, the user is most likely looking at the display. We can therefore either reduce the frequency of image capture or eliminate it completely and keep the display on. Face detection could also be completely suspended when there are frequent user interface events (i.e., in a sense, merging with the traditional approach).

While the main focus of discussion in this paper is display power management, there are other opportunities in which sensing context could be used for system level energy management. For example, microphones could be used to determine background noise level and possibly adjust speaker volume. Sensors could be used to determine whether to completely turn off the speakers if, for example, nobody is around to hear them. Sensing 802.11 signal strength could be used to determine whether to offload computation to a server. Remote process execution has been shown to significantly reduce energy consumption of mobile devices [13].

6 Future Work

The FaceOff prototype is a framework for applying user intent sensing to display power management. We would like to expand its capabilities to include motion sensing and enhance the face detection module, possibly adding face or gaze tracking at various levels of detail. Our aim is to evaluate what is needed to improve the accuracy of the power states while minimizing the system overhead. We plan to experiment with the frequency of image sampling to make the system more responsive during the times when motion is detected and decrease or halt the image sampling when no motion is detected. We eventually plan to incorporate a light sensor into the prototype for determining optimal display brightness.

We also plan to use the prototype to evaluate the user's experience, generally a qualitative measure rather than a quantitative one. For this reason we are examining ways of quantifying the user's experience with the prototype, for example by adding a button to indicate annoyance at the display state changes. We would like to be able to gauge the accuracy of the prototype in determining context.

We plan to do a comprehensive user study characterizing usage patterns similar to the study done in [8]. We will use the study to provide estimates of energy savings from our display power management technique under realistic laptop workloads. In addition to capturing usage patterns for laptops, which we can do using our FaceOff prototype, we would like to study other mobile devices that could benefit from our system.

7 Related Work

As power management at a software level has gained attention both in research and industry, several standards have emerged. The first standard was *Advanced Power Management (APM)*, a BIOS-based power management specification. APM provides CPU and device power management. Device power states are transitioned based on timeouts. Problems found as APM matured led to the development of the *Advanced Configuration and Power Interface (ACPI) Specification*. In ACPI, power management decisions are made by the operating system rather than the BIOS [9]. Both APM and ACPI provide an interface for changing the power state of the display through software using *Display*

Power Management Signaling (DPMS). APM and ACPI provide hooks to manage the power state of the display, however currently the only strategy for taking advantage of the lower power states is turning off the backlight and display after a certain period of time with no user input [10]. The Compaq iPAQ PocketPC has an additional method of display power management using an ambient light sensor to allow for adaptable display brightness. The only other novel policy ideas we found for reducing the power consumption of the display were zoned backlighting proposed by Flinn and Satyanarayanan, a method which presupposes hardware that is not yet available [5], and the recent work on Energy-Adaptive Display System Designs [8] that proposes software optimizations called *dark windows*.

There are several projects that involve face detection and face, gaze and eye tracking for perceptual user unterfaces. The Smart Kiosk System uses vision to detect potential users and decide whether the person is a good candidate for interaction. [12]. CAMSHIFT is a face tracker that is being used to control games and 3D graphics by defining head movements to perform specific actions [1]. Another related project is a perceptual user interface for recognizing predefined head gesture acknowledgements. The face detection is performed by using an IBM PupilCam to locate the pupils in the image and then uses simple image processing techniques to detect the upper face region [3]. A series of articles on Attentive User Interfaces discusses several projects that use eye tracking to design context-based user interfaces [14]. To our knowledge there are no other projects that are integrating face detection and power management.

8 Conclusion

Sensors as components of sensor networks have recently become an interesting target domain for operating systems research (e.g., TinyOS [6]). In this position paper, we turn this around and consider low-power sensors as tools in the service of OS-based energy management for mobile computers. As a case study, we consider sensors providing information from which to infer user intention and user context as it affects energy management of the display – capturing the direct dependency that looking at the screen suggests a need for it to be illuminated. Intuitively, this is attractive as a more direct indication of the user's need for display power consumption than the keyboard and mouse input events used in traditional timeout-based strategies.

Our preliminary exploration of this use of sensors to inform the OS combines currently available technology that allows software to switch the display power state, low-power sensors, and face detection techniques. We have proposed a method of reducing display power consumption by turning the display off in the absence of a user. Face detection, while a computationally intensive technique, can be optimized for the simplified problem of detecting an upright, frontal face of an approximate size indicating the presence of a user looking at the display. For our FaceOff prototype, a web cam acquires images and the computer's own CPU performs skin color based face detection.

Measurements of power consumption using the prototype system indicate the promise of significant energy savings from this type of context-based display power management scheme. Camera technology trends indicate that cheap, very low power cameras are becoming more readily available and could produce greater net energy savings in the future using our technique. Furthermore, the specific task of user detection could potentially be optimized using additional low power sensors combined with less computationally intensive techniques to further reduce overall energy consumption.

While the most obvious immediate benefit of our display power management system would be extending the battery life of mobile devices, the method could also provide the basis for energy savings on larger scale displays. One interesting possibility is to apply a similar technique to very large displays, using gaze tracking to determine what part of the display to turn on.

In this position paper, we have demonstrated that exploiting low-power sensors to assist the OS in inferring user intention and context for the benefit of energy management is a fruitful direction.

9 Acknowledgments

This work is supported in part by the National Science Foundation (ITR-0082914,CCR-0204367).

References

[1] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2(2):12–21, 1998.

[2] Glolab Corporation. Parts for pyroelectric infrared motion detector. http://www.glolab.com/pirparts/pirparts.html.

[3] James W. Davis and Serge Vaks. A perceptual user interface for recognizing head gesture acknowl-

edgements. In *Workshop on Perceptive User Interfaces 2001*, Orlando, FL, 2001.

[4] C. S. Ellis. The Case for Higher-Level Power Management. In *Proceedings of the 7th Workshop on Hot Topics in Operating Systems*, Rio Rico, AZ, March 1999.

[5] Jason Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. In *Symposium on Operating Systems Principles (SOSP)*, pages 48–63, December 1999.

[6] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for network sensors. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 93–104, October 2000.

[7] Wilderness Security Inc. Bd-127 high resolution cmos camera. http://www.wildernesssecurity.com/bd127.htm.

[8] Subu Iyer, Lu Luo, Robert Mayo, and Parthasarathy Ranganathan. Energy-adaptive display system designs for future mobile environments. In *Proceedings of The First International Conference on Mobile Systems, Applications, and Services*, May 2003.

[9] J. Kolinski, B. Press, and A. Henroid. *Power Management History and Motivation*, chapter 2, pages 7–17. Intel Press, September 2001.

[10] Jacob Lorch and Alan J. Smith. Software Strategies for Portable Computer Energy Management. *IEEE Personal Communications Magazine*, 5(3):60–73, June 1998.

[11] Space Television Ltd. Color and b/w cmos board camera. http://www.spacetv.co.za/YOKO/PG22-23.htm.

[12] J. Rehg, M. Loughlin, and K. Waters. Vision for a smart kiosk. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 690–696, 1997.

[13] Alexey Rudenko, Peter L. Reiher, Gerald J. Popek, and Geoffrey H. Kuenning. The remote processing framework for portable computer power saving. In *Selected Areas in Cryptography*, pages 365–372, 1999.

[14] Roel Vertegaal. Attentive user interfaces. *Communications of the ACM*, 46(3), 2003.

[15] Gregory F. Welch. A survey of power management techniques in mobile computing operating systems. *Operating Systems Review*, 29(4):47–56, 1995.

[16] Heng Zeng, Xiaobo Fan, Carla Ellis, Alvin Lebeck, and Amin Vahdat. ECOSystem: Managing energy as a first class operating system resource. In *Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)*, October 2002.