Power to the People: Leveraging Human Physiological Traits to Control Microprocessor Frequency

Alex Shye Yan Pan Ben Scholbrock J. Scott Miller
Gokhan Memik Peter A. Dinda Robert P. Dick
Department of Electrical Engineering and Computer Science, Northwestern University, Evanston IL, USA
{*shye, panyan, b-scholbrock, jeffrey-miller, memik, pdinda, dickrp*}*@northwestern.edu*

Abstract

Any architectural optimization aims at satisfying the end user. However, modern architectures execute with little to no knowledge about the individual user. If architectures could determine whether their users are satisfied, they could provide higher efficiency; improved reliability, reduced power consumption, increased security, and a better user experience. A major reason for this limitation is their input devices. Specifically, the traditional input devices (e.g., the mouse and keyboard) provide limited information about the user. In this paper, we make a case for the addition of new biometric input devices for providing the computer information about the user's physiological traits. We explore three biometric devices as potential sensors: an eye tracker, a galvanic skin response (GSR) sensor, and force sensors. We first present two user studies that explore the link between the sensor readings and user satisfaction when the performance of the processor is varied as a video game is being played. In the first study, we drastically drop the processor clock frequency at a set point in the game. In the second study, we set the clock frequency to randomly-selected levels during game play. Both studies show that there are significant changes in human physiological traits as performance decreases. More importantly, we show that physiological changes correlate strongly to the satisfaction levels reported by the users. Based upon these observations, we construct a Physiological Traits-based Power-management (PTP) system that can be applied to existing dynamic voltage and frequency scaling (DVFS) schemes. We apply PTP to a typical CPU-utilization-based adaptive DVFS policy and evaluate our scheme using a third user study. An aggressive version of our PTP scheme reduces the total system power consumption of a laptop by up to 33.3% for an application averaged across users (18.1% averaged across three applications), while a conservative version reduces the total system power consumption by up to 25.6% across users (11.4% averaged across three applications).

1. Introduction

The ultimate goal of any architectural optimization is to satisfy the end user. However, the design, optimization, and evaluation of modern computer architectures have largely left the user out of the loop. Architects typically envision the computing stack extending from devices at the bottom to applications at the top. The user, who is the true top of the stack, is often not considered during architectural decisions. Similarly, performance evaluation is often simplified to metrics such as instructions per second (IPS). Although such metrics may be convenient and easy to measure, they do not directly correlate to user satisfaction [34].

Several trends are converging to increase the importance of exploring user-aware computer architectures:

User-centric Applications: Batch applications are not the sole workloads for most architectures. An increasing number of modern applications are designed to interact with a user. Many server-side applications exist to provide services to users over the network. Multimedia applications, video games, and web browers are common workloads on desktop machines. In addition, applications executing on embedded and portable devices are inherently interactive. It is important for architectures running such user-centric applications to be optimized with the goal of satisfying the user.

Architectural Trade-offs Exposed to the User: Architectures should not naively execute instructions as fast as possible. Due to thermal and power constraints, architectural trade-offs are now directly exposed to the user in the form of shorter battery life, decreased lifetime reliability, annoying performance-limiting thermal emergencies, and higher operating temperatures (causing "burning-lap syndrome"). To balance the trade-off between performance and thermal/power-related issues, it is important for architectures to tune performance to, but not above, the level necessary to meet user needs and expectations.

Optimization Opportunity: Users differ dramatically from each other. Recent studies have shown that there is considerable variation in user expectation and user satisfaction relative to actual hardware performance [17], [34]. Where there is variation, there is opportunity for optimization. Variation in user expectation has been leveraged for improving power consumption [25] and for efficiently scheduling virtual machines [22]. The benefits result from optimizing to individual users instead of assuming that all users are equal.

We assert that the design of modern architectures makes it difficult (if not impossible) to implicitly infer and reason about the end user. One only needs to observe the current computer usage model to understand this claim. First, the user directs the computer explicitly via input devices (e.g., keyboard or mouse). According to user direction, the computer executes instructions to manipulate machine state. Afterwards, the user obtains information via output devices (e.g., display or speakers). Note that during this human-computer interaction, *there is a considerable asymmetry between the information available to the user and information available to the computer*. Although the user can direct the computer to change/view the system state at any time, the computer executes with little any information about the user state.

In this paper, we make a case for balancing this human-computer information asymmetry by augmenting future architectures with new input devices that provide information

978-1-4244-2837-3/08/$25.00 ©2008 IEEE

on user state. Enabling a computer to sense and perceive user state has a number of benefits. First, understanding user state will enable user-aware optimizations by providing implicit user feedback. Tailoring execution to the individual user's "taste" will result in better efficiency and significant benefits in power savings or increased lifetime reliability. In addition, decisions about resource assignment (i.e., deciding on the level of parallelism of an application running on a chip multiprocessor) can be made more effectively. Most importantly, computer behavior will be personalized based upon individual expectations to improve user satisfaction.

We propose, and evaluate, the use of biometric input devices that provide information on human state by observing physiological traits. Using physiological readings is an intuitive first step in understanding the user; our experiments suggest that a change in user state results in a number of measurable physiological responses. We use an eye tracker to measure pupil dilation and eye movement, a galvanic skin response (GSR) sensor to measure skin resistance/conductance, and force sensors to measure behavior. We begin with two user studies to motivate the use of these additional input devices. In the first, we drastically drop the CPU frequency at a set point while a game is being played. In the second, we randomly vary the CPU frequency across multiple settings during game play. We show that the CPU frequency has a significant impact on the physiological traits of the users. We also show that the changes in the physiological traits correlate with the satisfaction levels reported by the participants.

Based upon these observations, we then construct a **Physiological Traits-based Power-management (PTP)** system to demonstrate an application of these biometric input devices. PTP may augment any existing dynamic voltage and frequency scaling (DVFS) scheme to make user-aware decisions. In its current implementation, PTP adjusts the maximum frequency by incorporating human physiological readings. DVFS is a common power saving technique available on modern microprocessors that scales the frequency (and voltage) of a microprocessor to reduce power consumption. By adding PTP to a typical CPU-utilization-based DVFS scheme, we significantly decrease power consumption with little to no impact on user satisfaction.

It is intuitive to imagine that the computer performance will impact the physiological responses of users. There have been studies showing the relationships between physiological sensor readings and reported user emotions in response to interaction with computer programs [26], [18]. However, to the best of our knowledge, this is the first study in measuring the impact of computer performance on human physiological traits. Specifically, we make the following contributions:

- We make a case for using biometric input devices (such as eye trackers, galvanic skin response sensors, and force sensors) in making architecture-level decisions;
- We show through two user studies that our selected biometric input devices are able to detect changes in human physiological traits as the performance is altered during the run of an application; and
- We demonstrate a user-aware system for augmenting DVFS and evaluate the system with another user study.

The rest of the paper is organized as follows. Section 2 discusses the biometric sensors. Section 3 presents the setup of the user studies. Section 4 describes the first two user studies correlating sensor readings to user satisfaction. Section 5 discusses our prototype DVFS system for leveraging biometric input devices. Section 6 discusses implementation of the system and Section 7 presents our results. Section 8 describes related work and we conclude with Section 9.

2. Biometric Input Devices

To support user-aware computer architectures, computers will require a means to understand user satisfaction. Although it is possible to explicitly ask the user for information, this may be annoying. The ability to implicitly determine the degree of user satisfaction would be ideal. Unfortunately, current architectures are not equipped to implicitly estimate user satisfaction. This is due to a fundamental limitation of current input devices. Traditional input devices mainly exist to allow the user to explicitly control the machine state. However, they provide little information about physiological state. Without any information about user state, it is obvious that a computer cannot reason about user satisfaction. To help bridge this gap, we make a case for the addition of biometric sensors in future architectures. In this work, we explore three biometric sensors: eye trackers, galvanic skin response sensors, and force sensors. These sensors are described in the following sections.

2.1. Eye Tracking

Eye behavior reveals a lot of information about users' state. We are particularly interested in pupil dilation and pupil movement. Pupil dilation, or changes in the pupil radius over time, has been shown to correlate to many external and internal human factors. Studies show pupil dilation to be related to mental workload [19], perceptual changes [10], and positive/negative affect or emotion processing [30]. Pupil movement provides another source of information. Even when viewing a still image, humans do not keep their eyes steady. Instead, the eye constantly looks around finding interesting parts of each scene to create a larger mental map of the whole scene. Changes in the behavior of eye movement may also indicate higher level changes in the scenery, or human interests/state. For example, saccades (fast simultaneous movement of both pupils) have been linked to boundaries of event perception [35].

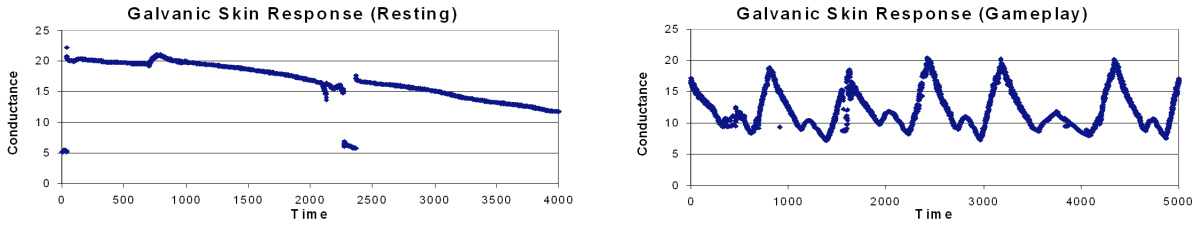
We use the ASL MobileEye eye tracker, shown in Figure 1(a), for collecting eye-related information. The eye tracker uses video-based combined pupil/corneal reflection to track the focus of the user's right eye. A video feed is analyzed to extract the pupil location and pupil radius. The data gathered is in pixels relative to the video feed, and is sampled 30 times per second. Pupil dilation is measured by using the pupil radius samples from the eye tracker. Pupil movement is measured using the Euclidean distance between consecutive samples of the pupil X-Y coordinates.

2.2. Galvanic Skin Response

Galvanic skin response (GSR) [7] measures the skin's ability to conduct electricity. GSR is strongly, but not completely,

(a) Eye tracker. (b) Custom-made GSR sensor. (c) Force sensors.

Figure 1. The biometric sensors used in our experiments: (a) an eye tracker, (b) a custom-made galvanic skin response sensor, and (c) force sensors attached to the arrow keys on the keyboard.



(a) Resting. (b) Playing the Need for Speed video game.

Figure 2. GSR traces of a user that capture (a) the long-term change in the GSR while a user is resting and (b) the short-term effects when playing the Need for Speed game. The existence of the long-term effect motivates the use of the delta GSR metric for measuring user arousal.

correlated to the conductance of sweat in sweat glands in skin [41]. GSR acts as an indicator of the autonomic nervous system reflecting both sympathetic (e.g., fight-or-flight response) as well as parasympathetic (e.g., rest or relaxation) response. In general, a low conductance is a sign of relaxation and high conductance is a sign of mental, emotional, and/or physical arousal. However, different emotions may produce discriminable waveforms [5], [39].

We use a custom-made galvanic skin response (GSR) sensor which is shown in Figure 1(b). The GSR sensor consists of two probes attached to velcro strips that are wrapped around the user's fingers during experiments. The two probes are wired in a voltage divider circuit for measuring the voltage (and therefore the resistance and/or conductance) across the skin.

GSR readings show long-term and short-term effects. For example, two sample GSR traces for one of the authors are shown in Figure 2; Figure 2(a) shows the GSR when resting and Figure 2(b) shows the GSR when playing the Need for Speed computer game. At rest, the GSR does not stay constant. Rather, it slowly decreases over a period of 5–10 minutes and then slowly levels out. When excited during game play, the GSR exhibits a much more varied response. To measure short-term changes in user arousal, and filter out the long-term trends, we employ a metric that we call ***delta GSR***, which resembles the metric "hash GSR" [5]. Delta GSR is computed by taking the difference between consecutive samples and filtering out the negative values. When summed over a period of time, the delta GSR serves as a metric for the total user arousal for the time period. We sample at 30 Hz and use a period of one second.

2.3. Force Sensors

We also use force sensors (shown in Figure 1(c)) to collect behavioral information about the user. Studies in keystroke dynamics have shown that keystroke patterns for a given user are correlated with various emotional states [40]. However, the force of each key press might hold additional information not captured by timing alone. For example, users may press the keys harder to express annoyance, or during times of intense involvement in game play. Also, for some applications, the range of keys involved is quite limited, and force may provide more information than keystroke patterns. In this work, we study the correlation between keystroke force and user satisfaction.

We use force-sensitive resistors to instrument each of the four arrow keys, as shown in Figure 1(c). The force sensors are measured using a voltage divider circuit. The maximum pressure value among all measured keys yields a single metric for comparison, which we will refer to as ***MaxArrow***. The sampling rate is 30 Hz.

2.4. Sensor Metrics

We measure four readings from the biometric input devices: pupil dilation, pupil movement, delta GSR, and arrow-key force. As we gather these readings, we summarize them using various statistics. For each reading, we consider the maximum, arithmetic mean, and the variance of the readings every second. We define the term **sensor metric** to be a specific combination of a statistic and a biometric reading. We format sensor metrics as follows: `<statistic>_<sensor>`. For example, the arithmetic mean of the pupil movement is denoted by `Mean_PupilMovement`.

2.5. Sensor Extensibility and Cost

The intrusiveness of sensors is a major consideration for using them as biometric input devices. Ideally, biometric input devices will (1) not impede the use of the computer in any way, (2) require little effort by the user, and (3) not incur significant financial cost. We select our sensors based on these principles. Consumer "remote eye tracking" products are available which detect eye focus and pupil radius without a head-mounted system. Further research into this area is likely to lower the cost of these systems [6]. Modern laptops contain built-in cameras and image recognition software exists for detecting pupils [28]. The electrical components required to measure GSR are inexpensive. While the velcro-strip contacts may be considered too cumbersome, these contacts have also successfully been integrated into a computer mouse in a way that requires no explicit action by the user [42]. Integrating force sensors into a computer keyboard would do little change to the existing structure and piezoresistive force sensors are inexpensive; the force sensors used for this work are currently available for under $15 per sensor [38].

3. User Study Setup

Our experiments are done using an IBM Thinkpad T61 with a 2.2 GHz Intel Core 2 Duo T7500 processor and 2 GB DDR2 SDRAM running Microsoft Windows XP. The laptop is tethered to power for experiments. The processor supports seven frequency levels using Intel Enhanced SpeedStep Technology (2.2 GHz, 1.6 GHz, 1.2 GHz, 800 MHz, 600 MHz, 400 MHz, and 200 MHz). In our experiments, we use the top five frequencies ranging from 2.2 GHz to 600 MHz.

Data from the GSR and force sensors is collected using a National Instruments 603E data acquisition card connected to the PCI bus of a separate workstation. The workstation then sends the sensor information through a TCP socket to the laptop over a private LAN connection.

In our user studies, we use the following applications:

- **Need for Speed Pro Street** [3]: A 3D driving game against the computer. The game is very CPU-intensive.
- **Tetris Arena** [2]: A 3-D version of the classic puzzle game. The game consumes 100% of the CPU. However it exhibits little performance degradation as the frequency is decreased.
- **Microsoft Word 2000 Version 9.0** [1]: The user is given a document to reproduce in Microsoft Word. In general, Microsoft Word is not CPU intensive. However, we include some high-quality images into the document. Moving the images occasionally causes short bursts of high CPU utilization.

We developed a user pool by advertising our studies within Northwestern University. The participants come from a variety of backgrounds and include males and females, engineers and non-engineers, as well as inexperienced computer users.

4. Correlating Human Physiological Traits with User Satisfaction

The ultimate goal of this paper is show how human physiological traits can be used as an implicit measure for inferring user satisfaction. In this section, we present two user studies exploring the link between human physiological readings and user satisfaction.

4.1. Motivating the Use of Physiological Sensors

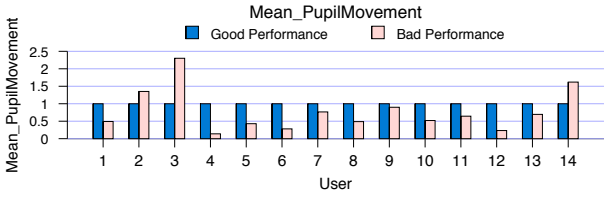
The first user study explores whether there **are** changes in human physiological traits when the performance of the processor is changed. One of our major concerns was that the measurement noise during game play may mask any changes in physiological traits. It is not difficult to imagine possible sources of noise. For example, in a driving game, a difficult section of tight turns may produce different measurements than another section with a long straightaway. Due to this concern, we first conduct a controlled initial user study with 14 users. During the study, we ask the users to play the Need for Speed game twice. Each time, at a predetermined position on the racetrack, we either maintain the highest frequency, or drop the frequency to 600 MHz for 20 seconds. At 600 MHz, the game greatly slows down. During the 20 seconds, we measure statistics from each of the physiological sensors.

Figure 3 shows the data from three of the sensor metrics that display significant changes in the initial user study. Mean eye movement (shown in Figure 3(a)) decreases for the large majority of the users. The maximum force on the arrow keys (shown in Figure 3(b)) also registers a noticeable decrease for most users. The maximum delta GSR (shown in Figure 3(c)) shows a relative change for many of the users. However, it increases for some users and decreases for others. The difference in users may be attributed to varying emotional reactions to a slow system: some users become annoyed and more aroused, while others become bored and less involved. Nevertheless, the results indicate that both arousal-based sensors (e.g., DeltaGSR) and behavioral sensors (e.g., MaxArrow) do indeed change significantly as application performance is decreased.

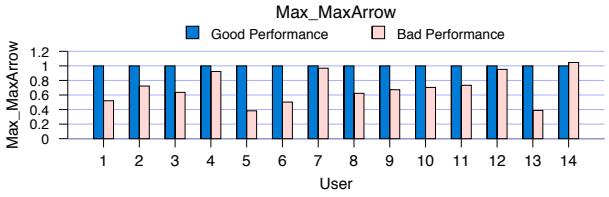
4.2. Physiological Sensors and User Satisfaction

With the knowledge that the sensor metrics do indeed change with performance, we conduct a second study to explore (1) the effect of random game phases and (2) the correlation between physiological readings at different performance levels and user satisfaction. The users play the Need for Speed game. This time, the processor speed is changed to a random frequency at a random point in the game. The change in performance lasts for 30 seconds. We randomly visit each frequency level twice; the first time we collect sensor metric readings, and the second time we verbally ask the user for a satisfaction rating. Users report their satisfaction as follows: 5 (Very Satisfied), 4 (Satisfied), 3 (Indifferent), 2 (Unsatisfied), and 1 (Very Unsatisfied).

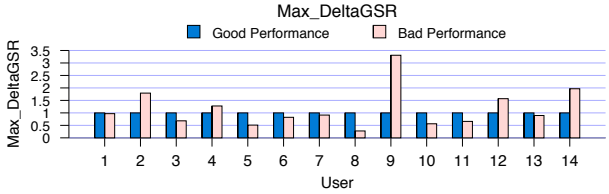
A good sensor metric will report as different when the user satisfaction changes and as similar when user satisfaction remains the same. To distinguish between sensor metrics at different frequencies, we employ a *t-test-based similiarity metric*. As the physiological sensors are noisy by nature, we use multiple samples and statistical methods. Both the data acquisition card (collecting GSR and force information) and the eye tracker sample at 30 Hz. Each second, we compute the sensor metrics based on 30 samples. After discarding the first and last five seconds of each 30 seconds interval, we have 20



(a) Mean eye movement



(b) Maximum force on the arrow keys.



(c) Maximum delta GSR.

Figure 3. (a) Mean pupil movement, (b) maximum arrow force, and (c) maximum delta GSR for the same 20 seconds of game play at a good performance level, and at a bad performance level. Mean pupil movement and maximum arrow force significantly decrease. Maximum delta GSR has more variation across users indicating different responses to a drop in performance.

Sensor Data	Success Rate	False Positive	False Negative
Max_PupilRadius	70.2%	14.3%	15.5%
Max_MaxArrow	69.0%	13.1%	17.9%
Mean_MaxArrow	69.0%	13.1%	17.9%
Mean_PupilRadius	67.9%	11.9%	20.2%
Mean_PupilMovement	57.1%	13.1%	29.8%
Max_DeltaGSR	58.3%	9.5%	32.1%

Table 1. Outcomes of comparing the t-test-based similarity metric and user satisfaction. Success means that the t-test outcome matches the user rating. False negatives occur when the t-test falsely predicts a difference and false positives occur when the t-test falsely predicts similarity with the highest frequency.

calculated values per sensor metric. We then use a t-test, with a 90% confidence interval, as our metric for measuring the similarity between sets of values from different frequencies.

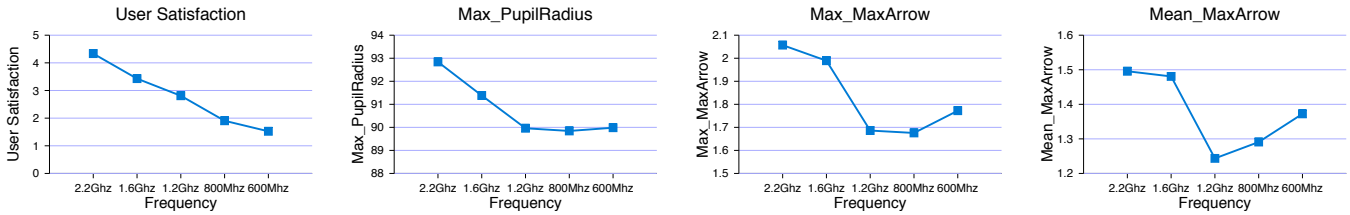
We now evaluate the behavior of our sensor metrics across multiple frequencies. For every sensor metric, we use the t-test-based similarity metric to compare each frequency with the highest frequency. The assumption is that if the user is annoyed, the t-test should indicate that the two sets are different; if the user is not annoyed, the t-test should indicate that the two sets are similar. We then manually compare the t-test results with the reported user satisfaction. The sensor metric a success if (1) the t-test indicates a difference and the user satisfaction changes, or (2) the t-test indicates similarity and the user satisfaction does not change. False positives occur when the t-test indicates a difference, but the user satisfaction is the same. False negatives occur when the t-test indicates similarity, but the user satisfaction is different.

Out of our twelve potential sensor metrics (maximum, mean, and variance for pupil radius, pupil movement, delta GSR, and force feedback), we develop a set of the six best individual sensor metrics (shown with their respective counts in Table 1). The success rates of the six sensor metrics are all above 60% with the top three predicting similar/different user satisfaction with nearly 70% accuracy. The false positive rate ranges from 11.9%–14.3% and the false negative rate ranges from about 15.5%–32.1%.[1] These results show that there is a strong correlation between changes in satisfaction and changes in the physiological readings.

To confirm our findings for the entire set of users, we average the sensor metrics across all users and look for trends. Figure 4 shows the averaged data for user satisfaction and the top three sensor metrics. There is a clear correlation between our sensor metrics and user satisfaction. For reference, the rest of the raw data is shown in Figure 10 in Appendix A. The sensor metrics exhibit some noise across users but, overall, these results show that a change in user satisfaction generally results in a change in sensor readings. This behavior, together with the high prediction accuracy, shows that user satisfaction and physiological traits are correlated.

We now consider the confidence level reported by the t-test for each comparison. A high confidence level indicates that the two sets of data being tested are different. Figure 5 shows the average confidence levels across all users for each comparison. As performance decreases, confidence that the user satisfaction is different tends to increase. This signifies that the physiological readings differ more at lower performance levels. However, the lowest frequency level does not follow the same trend. We postulate that at this frequency level, the performance is so low that some users stop caring about the game. During the user studies, we recall users complaining about the performance and talking to the proctor instead of

1. The false positive rate implies a lost opportunity for reducing frequency, but no reduction in user satisfaction. Assuming that the sensors are independent, combinations of them may be used to reduce the false negative rate. Furthermore, any DVFS algorithm based on these sensors could treat the sensor readings conservatively, reducing the effect of false negatives. In the system we describe in Section 5, we use combinations of sensors and evaluate both aggressive and conservative uses of their readings.



(a) User Satisfaction (b) Max_PupilRadius (c) Max_MaxArrow (d) Mean_MaxArrow

Figure 4. Averages of the three best individual sensor metrics and the user satisfaction ratings across all 20 users. The three sensor metrics have a very strong correlation with the reported user rating.

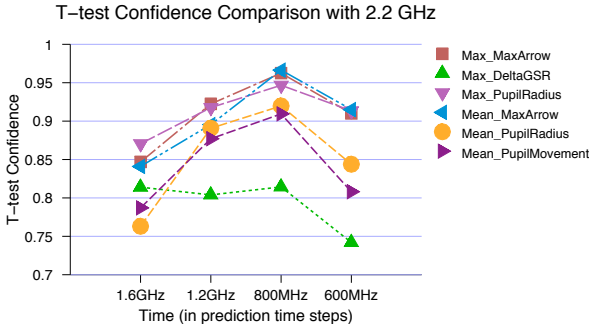


Figure 5. The average confidence provided by the t-test-based similarity metric between a frequency and the highest frequency across all 20 users and all sensor metrics. A high confidence indicates a difference. As frequency difference increases, the sensor metrics differentiate better, except for the lowest frequency.

playing the game. It is possible that the sensor readings may change in such situations. Nevertheless, even for this case, the sensor readings show significantly different behavior when compared to the highest frequency.

An important decision we have to make is how to decide when two readings are different. According to our subjective observations, the Need for Speed game exhibits very similar performance at 2.2 GHz and 1.6 GHz, but the performance quickly decreases at lower frequencies. A confidence level of 85% makes this distinction correctly when averaging across all users, and continues to distinguish correctly for a different set of users in the third study. Thus, we adopt an 85% confidence level in the t-tests for the rest of the paper.

In summary, these two initial user studies indicate that (1) a drastic drop in performance results in noticeable changes in our sensor metrics and (2) physiological readings can be used to infer user satisfaction.

5. Using Physiological Traits for DVFS

To demonstrate a use of empathic inputs, we construct a **Physiological Traits-based Power-management (PTP) system** for inferring user satisfaction from physiological readings and driving a DVFS algorithm.

The goal of PTP is to determine the minimum operating frequency that maintains user satisfaction. Specifically, PTP first runs a training phase with the target application (the algorithm for the training phase is detailed in Algorithm 1). PTP begins by comparing sensor readings at the second-highest frequency and the readings at the highest frequency.

Algorithm 1 PTP training algorithm

```
Frequency: f ← MAX_FREQ−1
while f is in frequency range do
  if TestSame(MAX_FREQ, f) then
    f ← f − 1
  else if Majority vote of 3 calls to TestSame(MAX_FREQ,f) is true then
    f ← f + 1
  else
    while f is in frequency range and Majority vote of 3 calls to TestSame(MAX_FREQ,f) is false do
      f ← f + 1;
    return f
```

Algorithm 2 TestSame: used by the PTP training algorithm

```
Two frequencies to test: f1, f2
Collect sensor metrics at f1 for 20 seconds
Collect sensor metrics at f2 for 20 seconds
t-test each sensor metric at f1 and f2 with confidence level of 85%
if more than 50% of sensor differ then
  return false
else
  return true
```

Each comparison (detailed in Algorithm 2) consists of (1) running for 20 seconds at the highest frequency, (2) running for 20 seconds at the testing frequency, and (3) a t-test between each of the sensor metrics. Initially, the algorithm aims at quickly reducing the frequency, if possible. The algorithm consecutively tests the frequencies for noise in the sensors. If two out of three tests report that the sensor metrics have changed, the majority vote test concludes that the two frequencies are the different; if not, it reports they result in the same user satisfaction. PTP repeats the majority vote for each frequency until it finds a frequency that does not pass. Then, it starts moving up from this point until it finds the level that passes the majority test. This frequency is called the ***settled*** frequency. Settled frequency is used as the maximum frequency during the execution of this application (in other words, the operating frequency is never increased to above the settled frequency).

It is important to note that from the user's perspective, the training and testing phases are not visible. The user simply interacts with the computer as normal.

An example of the interaction between the sensor metrics and PTP training is shown in Figure 6. The figure shows a trace of the algorithm as it settles on a frequency (in this case,

Algorithm 3 Linux ondemand governor algorithm

```
for every CPU in the system do
  if UP_DELAY milliseconds since last check then
    if utilization > UP_THRESHOLD then
      increase frequency to maximum
  if DOWN_DELAY milliseconds since last check then
    if utilization < DOWN_THRESHOLD then
      decrease to lowest frequency that keeps the utilization at 80%
```

1.6 GHz). The x-axis is time. Each step represents a 40 second period: 20 seconds at the highest frequency, and 20 seconds at the test frequency. The bold line with diamonds shows the test frequency, corresponding to the right vertical axis. The confidence levels of the t-tests for each sensor metric is shown in each time step, with the confidence indicated by the left vertical axis. A confidence above 85% indicates that the sensor metric differs between the two frequencies. We begin at 1.6 GHz. At this point, only 2 of the 6 sensors are different so we continue down to 1.2 GHz. At 1.2 GHz, there is a large change in Mean_PupilRadius. In fact, Max_MaxArrow, Mean_PupilRadius, Mean_MaxArrow, and Max_PupilRadius all exhibit high confidence for two tests and therefore reject the majority vote test for 1.2 GHz. The frequency increases to 1.6 GHz, and the sensor metrics return to values indicating that the sensors are the same, therefore predicting the user is satisfied. The algorithm settles at this frequency.

The PTP control algorithm is orthogonal to most other DVFS strategies. Although PTP provides a long-term prediction of user satisfaction, another DVFS strategy can be used for short-term decisions. We build PTP on top of an ***Adaptive*** DVFS strategy that is based upon the Linux ondemand DVFS governor [29]. This strategy is described in Algorithm 3. In short, if utilization increases above `UP_THRESHOLD`, the frequency increases to the maximum frequency. If the utilization is below the `DOWN_THRESHOLD`, the algorithm finds the frequency that maintains above 80% utilization. We use 200 ms for both `UP_DELAY` and `DOWN_DELAY`, 80% for `UP_THRESHOLD` and 30% for the `DOWN_THRESHOLD`.

PTP uses the minimum value of the frequency provided by the PTP control policy and the *Adaptive* control policy. Although the idea of combining the DVFS schemes may seem simple, there are benefits to such a solution. For example, a burst of keyboard or mouse events often cause adaptive DVFS control schemes (e.g., Windows XP DVFS [27] or the Linux ondemand control policy [29]) to unnecessarily raise the frequency to the maximum level. PTP prevents this by limiting frequency at the minimum level necessary to satisfy the user. In other words, PTP allows an adaptive DVFS scheme to make better short-term decisions when the CPU utilization is generally low. For applications that satisfy the user at high utilization, PTP may set the frequency to a lower level (if it predicts that the user is satisfied with that level), saving a significant amount of power.

Ideally, we would like to explore the combinations of sensor metrics for users and applications as well as search the parameter space for the PTP thresholds, but this would require real users in the loop and therefore be slow. A single user study with three applications takes about an hour of

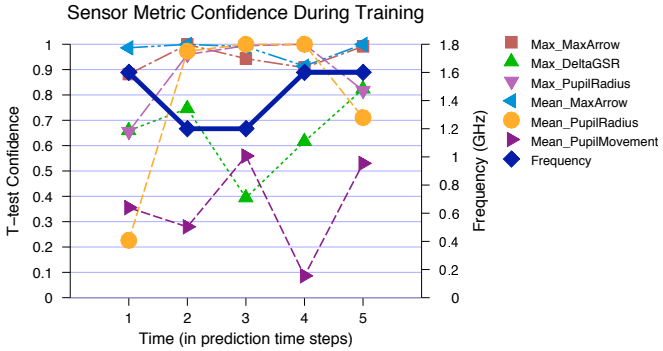


Figure 6. Trace of sensor metrics and the frequency during the training phase of the PTP algorithm. When sensor readings are compared for 1.2 GHz and 2.2 GHz, the majority of the sensors result in a high t-test, indicating that the user's state changes. As the algorithm adjusts to test 1.6 GHz, the physiological traits show less change. PTP chooses 1.6 GHz for the rest of the experiment

experimental lab time, not including the time to schedule the experiment. Therefore, trying multiple combinations quickly becomes very time consuming. We settled on the six most accurate individual sensor metrics listed in Table 1 and close the loop for evaluation with user studies.

Picking one set of sensor metrics opens some questions. Will the sensor metrics generalize across applications? Even for a single application, how does the sensitivity depend on users? By using the same set of sensor metrics across all users and applications, it is very possible that we will occasionally annoy some users. To increase the sensitivity to our experiments, we develop two variations of PTP: an **aggressive PTP** (aPTP) and a **conservative PTP** (cPTP). aPTP operates exactly as the PTP algorithm described in this section. cPTP is similar to aPTP but selects the frequency level one step higher than aPTP.

6. Implementation and Deployment

The PTP system is implemented as a user-space program that executes before each application run in the user studies. Data from the biometric devices are collected on a separate workstation and sent to the experimental laptop via a TCP socket connection. In production systems, we envision biometric input devices being managed by the operating system like traditional input devices. We have designed PTP as a proof of concept for using biometric input devices to improve architecture-level decisions. Other approaches to using biometric data different from ours could potentially lead to even stronger results. Here, we are concerned with providing the first evidence of the clear benefits of using biometric data in architecture-level decision making.

In a real-world implementation, the power consumption of the biometric devices would need to be outweighed by the power savings due to the PTP. The sensors chosen for this work all conform to this requirement. Piezoresistive force sensors may be measured with very little additional energy using a voltage-divider circuit and an analog-to-digital converter, which are both common, low-power circuits. GSR is also a simple resistive measurement, and requires

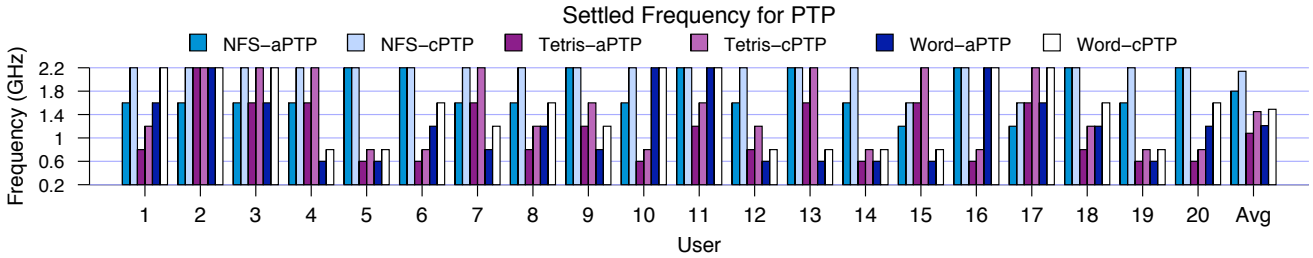


Figure 7. Frequency that aPTP and cPTP settle on for the Need for Speed, Tetris, and Word applications.

only a voltage divider and an analog-to-digital converter. An eye tracker requires an infrared camera, infrared LEDs, and the capacity for image processing. Collectively, the eye tracker sensor could operate on well below a Watt [44], [21]. Although some of these sensors may be expensive today, the technology for producing sensors capable of operating within desirable power constraints and at a low cost has already been developed. Additionally, the processing needs to interpret the sensors could also be assigned to a core of a chip multiprocessor, reducing the additional hardware required.

7. Experimental Results

In this section, we evaluate the aPTP and cPTP systems. We compare both PTP variants with the *Adaptive* scheme described in Section 5. We use the Need for Speed (NFS), Tetris, and Word applications and 20 users. In each run of an application, we begin with the training phase described in Section 5. The training phase varies based upon the number of majority vote tests performed by the PTP strategy. Afterwards, the user continues to use the *Adaptive* scheme and the aPTP scheme for 2.5 minutes each. The order of the aPTP and the *Adaptive* scheme is randomized between experiments. The last 10 users subsequently use the cPTP scheme for 2.5 minutes. At the end of each run, the user is asked to verbally report satisfaction based upon the scale described in Section 4.

During experiments, we capture traces of the frequency. A National Instruments 6034E data acquisition card measures the potential drop across a low-impedance resistor in series with the laptop power cable. This allows us to measure the system power consumption as frequency traces are replayed. The total system power includes the power consumed by the fully-operating laptop including the processor, a fully-lit 15.1" laptop display, network interface, and other peripherals.

The take-away points from our evaluation are:

- User satisfaction for aPTP and cPTP are nearly identical to the underlying adaptive scheme, and
- aPTP and cPTP save 18.4% and 11.4% total system power, respectively.

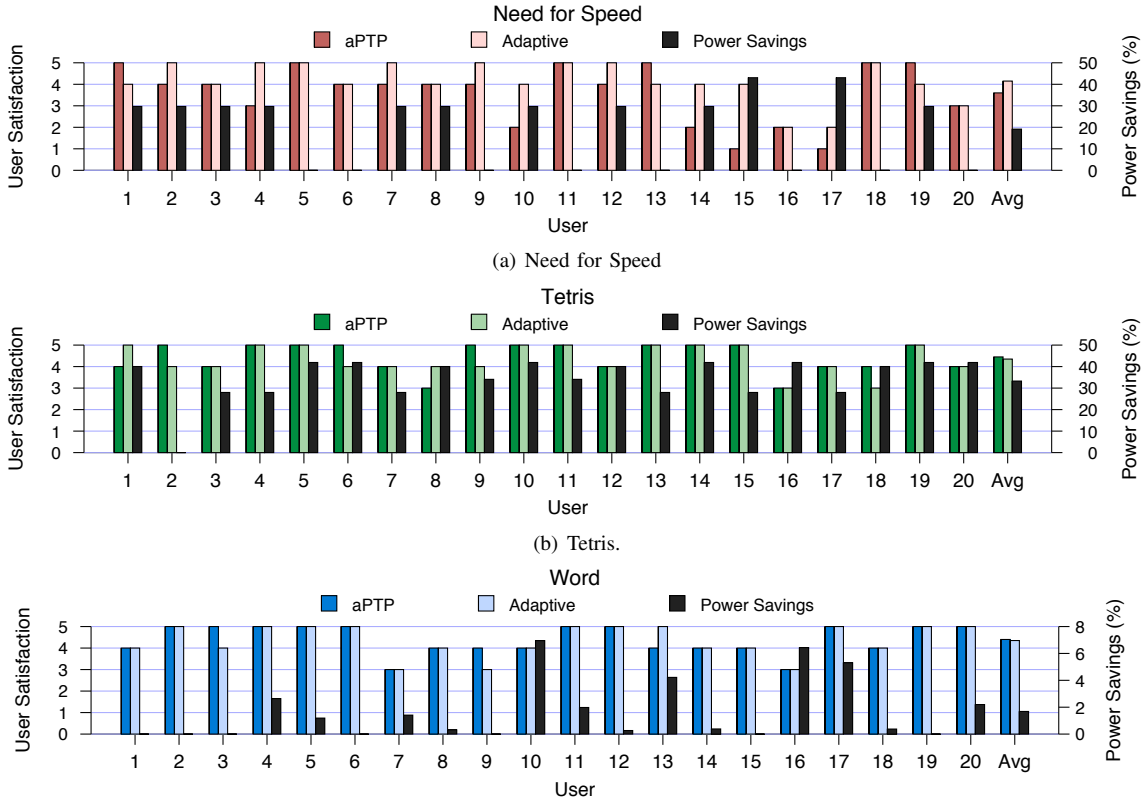
7.1. User Satisfaction and Power Savings

In Figure 7, we present the frequencies that aPTP and cPTP settle on for NFS, Tetris, and Word. The x-axis corresponds to the users and the y-axis is the settled frequency. Each cluster shows the settled frequency for both PTP variants and all applications.

NFS is a CPU-intensive application for which observable performance is sensitive to CPU frequency. aPTP picked either 1.6 GHz or 2.2 GHz for 18 out of the 20 users. This is drastically different from Tetris, where the observable performance is less sensitive to CPU frequency. The average frequency chosen by aPTP for Tetris is 1.08 GHz. Similarly, for Word, the average frequency chosen is 1.2 GHz. This clearly demonstrates aPTP's ability to intelligently detect the cases where CPU frequency can be lowered. Since for the Tetris and Word application, the lower frequencies and higher frequencies result in similar physiological responses, aPTP lowers the frequency. As indicated by user satisfaction levels, this achieves significantly higher efficiency without causing any dissatisfaction. Note that a user-specific customization is achieved purely based on the physiological readings from the users, without explicit input or knowledge of program phase.

There are some cases in Tetris and Word (14 out of 40 cases altogether), where a higher frequency of 1.6 GHz or 2.2 GHz is picked by aPTP. We checked the logs of physiological readings and found that the eye tracking data was missing in 4 of these 14 cases. This occurs when the user shifts in a manner such that pupil is not captured by the eye tracker camera. This introduces significant noise to the decision making system and results in a higher frequency being chosen. Another 3 cases correspond to self-admittedly inexperienced users. These users show erratic behavior. Thus, the sensor readings are noisy and our system conservatively sets the frequency at a high level. We must note that, although this looks like a lost opportunity for power saving, it is an interesting feature of the overall scheme: if for one reason or another, the sensor readings become noisy, our system conservatively sets the maximum allowed frequency to a high one, thereby avoiding false negatives (i.e., cases where the user is dissatisfied and our system predicts them to be otherwise). For Word, we are limited to utilizing only 4 metrics, compared to the 6 used in NFS and Tetris, because Max_MaxArrow and Mean_MaxArrow cannot be used (the user does not press the arrow keys often). Nevertheless, with Word, aPTP succeeds in picking low CPU frequencies (1.2 GHz and below) for 13 out of the 18 users with valid sensor readings. Similarly, for Tetris, aPTP picks a low frequency for 13 out of 15 users with valid sensor readings.

The reported user satisfaction ratings and power savings for each of the applications comparing aPTP and the *Adaptive* scheme are presented in Figure 8. The figure shows clustered bars for each user. The left two bars in each cluster represent



(a) Need for Speed

(b) Tetris.

(c) Microsoft Word.

Figure 8. User satisfaction and power consumption for the Need for Speed, Tetris, and Word applications. The left two bars per cluster show the user satisfaction for aPTP and the *Adaptive* DVFS schemes. The right bar in each cluster shows the total system power savings.

the user satisfaction with aPTP and with the *Adaptive* scheme and correspond to the leftmost vertical axis. The right bar in each cluster represents the total power savings corresponding to the vertical axis on the right. For our two CPU-intensive applications, PTP saves a considerable amount of total power. On average, for NFS (presented in Figure 8(a)), aPTP reduces power consumption by 19.2%, and for Tetris (presented in Figure 8(b)), aPTP reduces total power consumption by 33.3%. Word (presented in Figure 8(c)) is only CPU-intensive in short bursts and aPTP only saves 1.7% system power. For both Tetris and Word, aPTP also does not impact user satisfaction. However for NFS, aPTP trades off a small amount of user satisfaction for power savings. For this application, aPTP is too aggressive for some users. Averaged across three applications, aPTP saves 18.4% system power when compared to the *Adaptive* scheme.

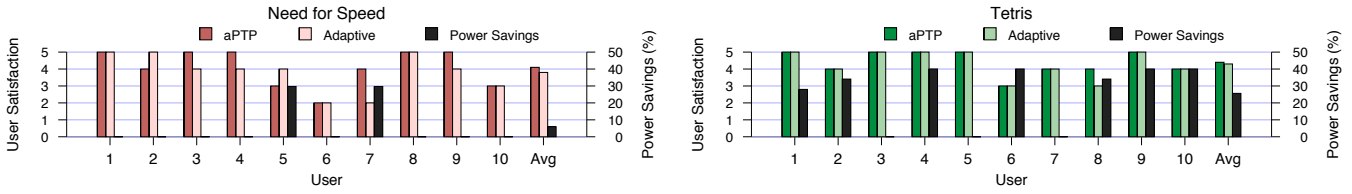
To explore a more conservative PTP scheme, we evaluate cPTP with 10 users. Figure 9 presents the results of this study. The graph is in the same format as Figure 8. By using cPTP, we trade off improved user satisfaction with power savings. cPTP tends to maintain the highest frequency for NFS and saves 5.9% system power, while maintaining the same satisfaction level as the *Adaptive* scheme. cPTP trades off the decreased power savings with an improved average user satisfaction rating compared to aPTP. cPTP also maintains a high user satisfaction for Tetris, and the power savings drop from 33.3% to 25.6%. Averaged across three applications, cPTP saves 11.4% system power while maintaining the user satisfaction.

Overall, our results are very encouraging: they show that PTP can successfully sense physiological traits, predict user satisfaction, and drive a DVFS scheme that saves considerable power while maintaining user satisfaction.

8. Related Work

At the architecture level, there has been work that takes user perception into account. Endo et al. [12], [11] uses latency as a performance metric and for detecting performance anomalies in operating systems. Vertigo [15] monitors application messages to measure user-perceived latency. Vertigo proposes a layered frequency scaling scheme similar to PTP. Other DVFS algorithms use task information, such as measuring response times in interactive applications or rate of change in the display [23], [24] as a proxy for the user. These studies rely on high-level metrics as proxies for user satisfaction. To the best of our knowledge, this is the first work that correlates human physiological data to user satisfaction for making architecture-level decisions.

Dynamic voltage and frequency scaling (DVFS) is an effective technique for microprocessor energy and power control for most modern processors [8], [9], [13], [14], [16],



(a) Need for Speed

(b) Tetris.

Figure 9. User satisfaction and power consumption of cPTP for the Need for Speed and Tetris applications. Word is not included because power savings and user satisfaction levels are nearly identical to aPTP. The left two bars per cluster show the user satisfaction of cPTP and the *Adaptive* DVFS schemes. The right bar in each cluster shows the total system power savings. Using cPTP, we trade-off a decreased power savings with improving user satisfaction when compared to aPTP.

[43]. Energy efficiency has been a major concern for mobile computers. Mallik et al. [25] and Shye et al. [34] show that it is possible to utilize user feedback to control a power management scheme. However, both schemes require explicit user feedback that may be an inconvenience to the user. Our work provides an implicit mechanism for inferring user satisfaction that is orthogonal to these approaches.

The Affective Computing Group at MIT has worked to develop emotion-aware computers [31]. They have proposed devices such as HandWave GSR [36] with a squeezable mouse [32]. Their most related work is concerned with creating [33] or detecting [20] user frustration with learning software. There is also work on relating posture to persistence in puzzle games [4], and using face recognition software to improve social-emotional learning for autistic children [37]. Other researchers, such as Mandryk and Atkins [26] and Hazlett and Benedek [18], have also shown that physiological measures (e.g., GSR, EMG sensors, and heart rate) can be used to predict emotion when playing games. Our work measures physiological responses in the face of changes in computer performance and utilize real-time sensing of physiological traits in making architectural decisions.

9. Conclusion

In this paper, we made a case for the addition of new input devices that provide information on human state in future computer architectures. Specifically, we explored the use of three biometric sensors: an eye tracker to measure pupil dilation and pupil movement, a galvanic skin response sensor for sensing user arousal, and force sensors on the keyboard for sensing behavioral traits. We have conducted multiple user studies. The first showed that human physiological readings do in fact change with changes in performance. The second shows that biometric readings are correlated with user satisfaction. Based upon the observations in these initial studies, we constructed a Physiological Traits-based Power-management (PTP) system for driving dynamic voltage and frequency scaling on a processor. PTP was designed to be orthogonal to most other DVFS techniques. We built our system in combination with an adaptive DVFS scheme based on the Linux ondemand governor. An evaluation using an additional user study showed that an aggressive PTP scheme reduced the total system power consumption of the laptop by up to 33.3% for an application averaged across users (18.1% averaged across three applications), while a conservative PTP scheme reduced the total system power consumption by up to 25.6% across users (11.4% averaged across three applications). Overall, these results show that a robust system can be built that makes decisions based upon observing biometrics sensors. This demonstrates the potential for incorporating biometric information into the architecture-level decision making process.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and are grateful to the users who volunteered their time for the user studies. This work is in part supported by DOE Awards DE-FG02-05ER25691 and DE-AC05-00OR22725 (via ORNL), NSF Awards CNS-0720691, CNS-0721978, CNS-0715612, CNS-0551639, CNS-0347941, CCF-0541337, CCF-0444405, CCF-0747201, IIS-0536994, IIS-0613568, ANI-0093221, ANI-0301108, and EIA-0224449, by SRC award 2007-HJ-1593, by Wissner-Slivka Chair funds, and by gifts from Symantec, Dell, and VMware.

References

[1] Microsoft word 2000. Microsoft Corporation.
[2] Tetris arena. Terminal Studio.
[3] Need for speed prostreet, 2007. Electronic Arts.
[4] H. I. Ahn, A. Teeters, A. Wang, C. Breazeal, and R. W. Picard. Stoop to conquer: Posture and affect interact to influence computer user's persistence. In *Proceedings of the 2nd Intl. Conf. on Affective Computing and Intelligent Interaction*, September 2007.
[5] A. Ax. The physiological differentiation between fear and anger in humans. *Psychosomatic Medicine*, 15(5):433–442, July 1952.
[6] M. Bohme, A. Meyer, T. Martinetz, and E. Barth. Remote eye tracking: State of the art and directions for future development. In *Proc. of the 2006 Conference on Communication by Gaze Interaction (COGAIN)*, pages 12–17, 2006.
[7] W. Boucsein. *Electrodermal Activity*. Plenum Press, 1992.
[8] B. Brock and K. Rajamani. Dynamic power management for embedded systems. In *Proceedings of IEEE SOC Conference*, 2003.
[9] S. Dhar, D. Maksimovic, and B. Kranzen. Closed loop adaptive voltage scaling controller for standard cell asics. In *Proceedings of Intl. Symp. on Low Power Electronics and Design*, 2005.
[10] W. Einhauser, J. Stout, C. Kock, and O. Carter. Pupil dilation reflects perceptual selection and predicts subsequent stability in perceptual rivalry. In *Proceedings of the National Academy of Sciences*, pages 1704–1709, 2008.
[11] Y. Endo and M. I. Seltzer. Using latency to evaluate interactive system performance. In *Proceedings of the Intl. Conf. on Measurements and Modeling of Computer Systems*, 2000.

[12] Y. Endo, Z. Wang, J. B. Chen, and M. I. Seltzer. Using latency to evaluate interactive system performance. In *Proceedings of the USENIX Symp. on Operating Systems Design and Implementation*, 1996.
[13] D. Ernst, N. S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge. Razor: A low-power pipeline based on circuit-level timing speculation. In *Proceedings of the Intl. Symp. on Microarchitecture*, 2003.
[14] Y. Fei, L. Zhong, and N. K. Jha. An energy-aware framework for coordinated dynamic software management in mobile computers. In *Proceedings of Intl. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, 2004.
[15] K. Flautner and T. N. Mudge. Vertigo: Automatic performance setting for linux. In *Proceedings of the Symposium on Operating Systems Design and Implementation*, 2002.
[16] S. Gochman and R. Ronen. The Intel Pentium M processor: Microarchitecture and performance. *Intel Technology Journal*, 2003.
[17] A. Gupta, B. Lin, and P. A. Dinda. Measuring and understanding user comfort with resource borrowing. In *Proceedings of the Intl. Symp. on High Performance Distributed Computing (HPDC)*, 2004.
[18] R. L. Hazlett and J. Benedek. Measuring emotional valence to understand the user's experience of software. *International Journal of Human-Computer Studies*, 65:306–314, 2007.
[19] S. T. Iqbal, P. D. Adamczyk, Z. S. Zheng, and B. P. Bailey. Towards an index of opportunity: Understanding changes in mental worklad during task execution. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 311–320, April 2005.
[20] A. Kapoor, W. Burleson, and R. W. Picard. Automatic prediction of frustration. *Intl. Journal of Human-Computer Studies*, pages 724–736, August 2007.
[21] J.-O. Klein, L. Lacassagne, H. Mathias, S. Moutault, and A. Dupret. Low power image processing: Analog versus digital comparison. In *CAMP '05: Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception*, pages 111–115, Washington, DC, USA, 2005. IEEE Computer Society.
[22] B. Lin and P. A. Dinda. Towards scheduling virtual machines based on direct user input. In *Proceedings of the 1st International Workshop on Virtualization Technology in Distributed Computing*, Nov 2006.
[23] J. Lorch and A. Smith. Using user interface event information in dynamic voltage scaling algorithms. Technical Report UCB/CSD-02-1190, University of California at Berkeley, Berkeley, CA, 2002.
[24] A. Mallik, J. Cosgrove, R. Dick, G. Memik, and P. Dinda. PICSEL: Measuring user-percieved performance to control dynamic frequency scaling. In *Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems*, March 2008.
[25] A. Mallik, B. Lin, G. Memik, P. A. Dinda, and R. P. Dick. User-driven frequency scaling. *Computer Architecture Letters*, 5(2), July–December 2006.
[26] R. L. Mandryk and M. S. Atkins. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International Journal of Human-Computer Studies*, 65:329–347, 2007.
[27] Microsoft. Windows native processor performance control. In *Windows Platform Design Notes*, November 2002.
[28] V. G. Moshnyaga and E. Morikawa. Reducing energy consumption of computer display by camera-based user monitoring. In *Lecture Notes in Computer Science*, pages 528–539, 2005.
[29] V. Pallipadi and A. Starikovskiy. The ondemand governor: Past, present, and future. In *Ottawa Linux Symposium*, July 2006.
[30] T. Partala and V. Surakka. Pupil size variation as an indication of affective processing. *Int. J. Human-Computer Studies*, 59:185–198, 2003.
[31] R. W. Picard. *Affective Computing*. MIT Press, Cambridge, 1997.
[32] C. J. Reynolds. The sensing and measurement of frustration with computers. Master's thesis, Master of Science in Media Arts and Technology at the MIT, Cambridge, MA, 2001.
[33] J. Scheierer, R. Fernandez, J. Klein, and R. W. Picard. Frustrating the user on purpose: A step toward building an affective computer. *Interacting with Computers*, 14(2):93–118, 2002.
[34] A. Shye, B. Ozisikyilmaz, A. Mallik, G. Memik, P. A. Dinda, R. P. Dick, and A. N. Choudhary. Learning and leveraging the relationship between architecture-level measurements and individual user satisfaction. In *Proceedings of the 35th International Symposium on Computer Architecture*, June 2008.
[35] T. J. Smith, M. Whitwell, and J. Lee. Eye movements and pupil dilation during event perception. In *Proceedings of the Eye Tracking Research and Applications Conference*, March 2006.
[36] M. Strauss, C. Reynolds, S. Huges, K. Park, G. McDarby, and R. W. Picard. The handwave bluetooth skin conductance sensor. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, October 2005.
[37] A. Teeters. Use of a wearable camera system in conversation: Towards a companion tool for social-emotional learning in autism. Master's thesis, MIT, 2001.
[38] Tekscan. Flexiforce: System and sensor pricing. http://www.tekscan.com/flexiforce/pricing.html.
[39] M. Toyokura. Waveform and habituation of sympathetic skin response. *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, 109(2):178–183, 1998.
[40] G. A. Tsihrintzis, M. Virvou, E. Alepis, and I. Stathopoulou. Towards improving visual-facial emotion recognition through use of complementary keyboard-stroke pattern information. In *Proceedings of the Fifth International Conference on Information Technology: New Generations*, pages 32–37, April 2008.
[41] R. Vetrugno, R. Liguori, P. Cortelli, and P. Montagna. Sympathetic skin response: Basic mechanisms and clinical applications. *Clinical Autonomic Research*, pages 256–270, June 2003.
[42] M. Whang. The emotional computer adaptive to human emotion. *Phillips Research: Probing Experience*, 8:209–219, 2008.
[43] Q. Wu, V. Reddi, Y. Wu, J. Lee, D. Connors, D. Brooks, M. Martonosi, and D. W. Clark. A dynamic compilation framework for controlling microprocessor energy and performance. In *Proceedings of the Intl. Symp. on Microarchitecture*, November 2005.
[44] D. Yang, A. Gamal, B. Fowler, and H. Tian. A 640× 512 CMOS image sensor with ultrawide dynamic range floating-point pixel-level ADC. *Solid-State Circuits, IEEE Journal of*, 34(12):1821–1834, 1999.

Appendix A

This appendix expands upon discussion in Section 4.2. Figure 10 presents the raw data for six of the sensor metrics. The results for each user is presented in a row in the table of graphs and each column corresponds to a different sensor metric (the first column presents the reported user satisfaction level). In each of the graphs, the x-axis represents the frequency with 1 being the highest (2.2 GHz) and 5 being the lowest frequency (600 MHz). The y-axis represents the user satisfaction rating for the first column and the mean of the sensor readings for the remaining columns. The raw data shows that the sensor metrics can be noisy. However, in general, a change in the user satisfaction is reflected by a change in sensor metrics. If we consider the average behavior (presented in the last row), we see that most sensors show a strong relation to the user satisfaction levels.

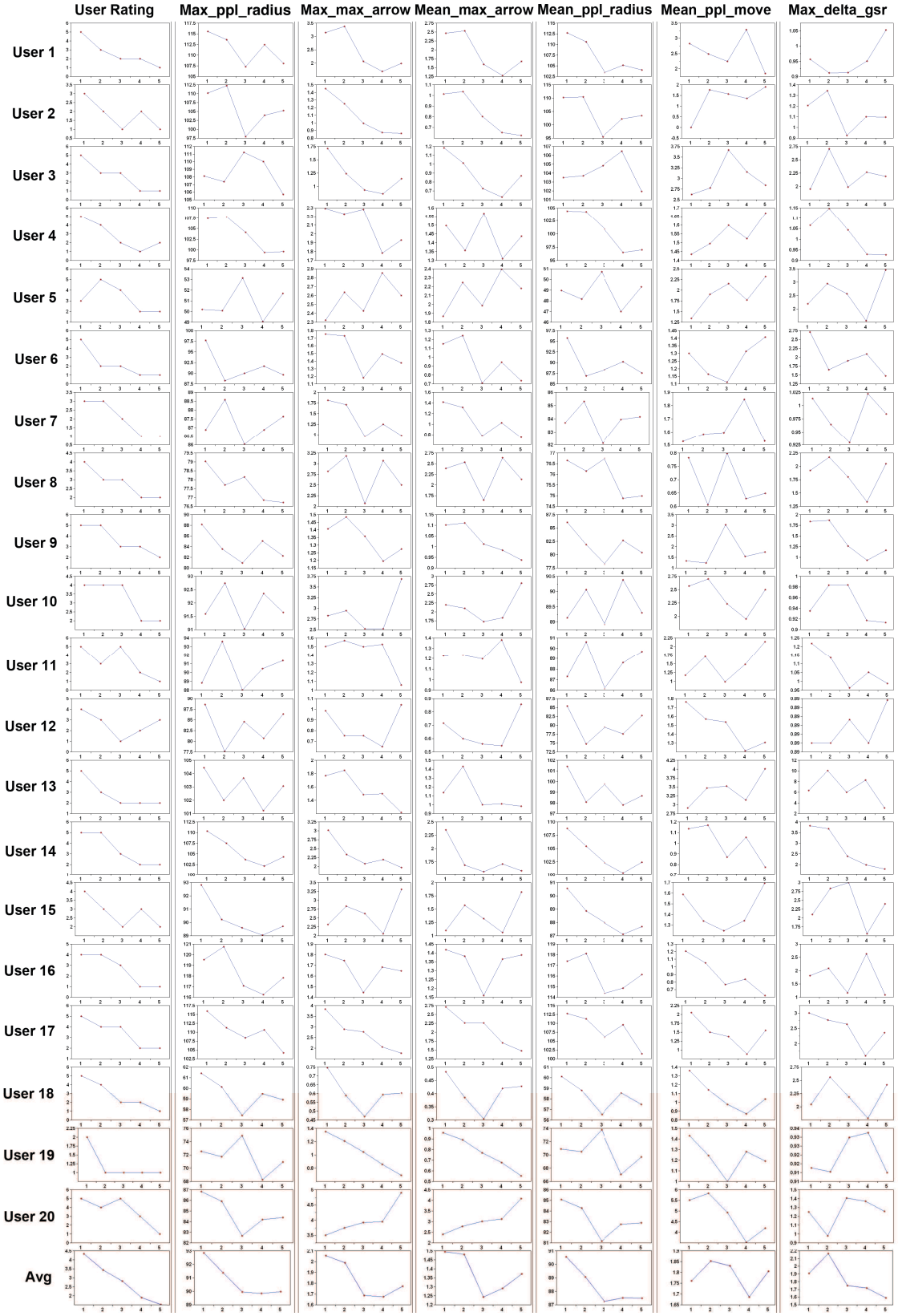


Figure 10. Physiological traits and user satisfaction when randomly changing to multiple frequencies at different points in Need for Speed. In each of the graphs, the x-axis represents frequency with 1 being the highest (2.2 GHz) and 5 being the slowest (600 MHz). The leftmost column shows user satisfaction and the others show data for each of 6 sensor metrics. The rows represent each user with all users averaged at the bottom.