Managing the Storage and Battery Resources in an Image Capture Device (Digital Camera) using Dynamic Transcoding

Surendar Chandra
Dept. of Computer Science
Duke University
Durham, NC 27708-0129
surendar@cs.duke.edu

Carla Schlatter Ellis
Dept. of Computer Science
Duke University
Durham, NC 27708-0129
carla@cs.duke.edu

Amin Vahdat
Dept. of Computer Science
Duke University
Durham, NC 27708-0129
vahdat@cs.duke.edu

ABSTRACT

Advances in hardware imaging technology and user demand for convenient mobile electronic image capture are fueling the development of inexpensive image capture devices that can acquire images rivaling the image quality of photographic film. Improvements in the hardware imaging technology have to be matched with intelligent image storage mechanisms that are aware of local storage and battery constraints. In this paper, we explore using a dynamic, informed image transcoding technique to manage the consumed battery and storage resources in digital cameras. Such application aware technologies are fundamental for the mass consumer acceptance of these newer digital technologies.

We show that this technique can allow the camera to store an order of magnitude more images. For a moderate number of images (e.g. 40), transcoding techniques can also maintain high quality images. The availability of fast wireless networks can allow the camera to capture 58 high quality images (51 uploaded) before running out of battery power. Storage technologies with expensive read and write operations (such as micro disks) can have a minor negative impact on battery life because of the extra read and write operations associated with transcoding operations. We show that the ability to effectively communicate the power vs. size vs. quality tradeoff to the end user is important for applications to adapt to the prevailing operating conditions.

1. Introduction

There is a rapid movement toward embedding powerful processors in a large variety of consumer electronic devices. Applications and devices dealing with multimedia objects (e.g. digital cameras, mp3 players, etc.) are an example. This paper focuses on resource management in such devices that are affected by the unique challenges and opportunities offered by mobile multimedia applications. We use the example of a digital camera to illustrate the issues in a mobile multimedia system.

The proliferation of multimedia capable devices and ubiquitous network technologies to deliver multimedia objects is fueling the demand for multimedia *generation* tools such as digital cameras. Consumer digital cameras with 2.1 megapixel resolution are currently available for under 500 dollars. Cameras offering 6 megapixels are becoming available. By contrast, 35 mm point-and-shoot cameras offer resolutions in the range of 8-9 megapixels and 35 mm SLR cameras offer resolutions of around 35 megapixels. Digital camera resolutions are improving exponentially and are expected to surpass film resolutions within a few years. Commercial image capture devices [9] capable of capturing 16 megapixels are also currently available. The generated image size grows as resolution of the image capture devices improves. For example, images produced by cameras offering 2.1 megapixels occupy around 1.2 MB of JPEG compressed storage. With increasing resolutions, the problem of image size worsens.

As a result of size, power and cost concerns, the storage capacity on these devices is limited. Cameras have used floppy disks, memory sticks, flash memory and microdrives as a storage medium. Each storage technology has its restrictions on storage capacity, battery consumption and dollar cost. Microdrives are expensive for mass market cameras. Floppy disks, though inexpensive, are cumbersome with their limited storage capacity. Even with anticipated improvements in the storage capacity available for future mobile devices, the demand for increasing resolution will likely out-pace the supply of storage.

Except for film transport and focusing mechanisms, traditional photography depends little on battery power. Digital cameras, on the other hand, are totally dependent on battery power to not only capture images, but also to store images, display the images in the view finder and upload images to permanent storage. Digital cameras use conventional primary batteries as well as re-chargeable metal hydride batteries. Future trends in battery technology do not promise dramatic improvements that will make this issue disappear.

A necessary feature for mass acceptance of a digital camera is acceptable battery and storage management. The nature of multimedia objects allows additional degrees of freedom in storage management. Multimedia objects are amenable to lossy transcoding operation that can reduce the file size by trading off information quality. For example lowering the bit rate and clipping are typical audio transcodings; clipping and resolution reduction are typical video transcodings; thumbnailing and cropping are typical image transcodings. In this work, we explore transcoding as a tool to manage multimedia objects in a mobile multimedia capture system. We use the example of a digital camera to illustrate the issues in a mobile multimedia system. We propose utilizing informed transcoding technology in a digital camera to manage its consumed battery and storage space. Vitally important for this application is that such resource management does not unduly sacrifice image quality.

Digital cameras currently offer the means for specifying the image quality so that less storage can be allocated to less important images. However, a-priori specification of the image quality of a picture is not desirable. Frequently, the user is left with a lower quality photograph than desired because they miscalculated the image importance or because they requested high quality versions of pictures that did not turn out as well as expected. We propose an automatic scheme for managing picture quality, available storage and battery life based on user-specified preferences where users specify the image importance (either before or after the picture is taken) with the system dynamically choosing the appropriate quality level based on the current system constraints. This also gives the photographer in the field more flexibility. Nothing is worse for a photographer than to lose the perfect photo opportunity due to resource constraints (traditionally, being out of film). This dynamic approach may allow the photographer to squeeze in *one more shot* without necessarily deleting any existing objects.

A fundamental observation of this work is the importance of communicating this quality vs. battery vs. storage tradeoff to the end user. In this paper, we develop techniques to allow application developers to exploit the storage, quality and batter tradeoffs to store images in the best possible fashion.

In this paper, we describe the model for a digital camera that offers additional features possible with current technology. We use transcoding as the enabling technology to dynamically change the image size using a quality-vs-size tradeoff. We utilize earlier work in quantifying the tradeoffs of an image transcoding, as well as an estimation of the computing overhead and storage gains of a image transcoding [3]. This technology allows informed decisions on the level of transcoding necessary for a particular operating scenario.

The rest of the paper is organized as follows: Section 2 reviews our previous work as the necessary background. We present the system architecture of our digital camera in Section 3. Section 4 describes the evaluation methodologies, measurement metrics and the workloads used in our study. The results of operating the digital camera under various operating environments are described in Section 5. Related work attacking similar problems is briefly discussed in Section 6. We conclude in Section 7.

2. Background: Quality Aware Transcoding

Quality aware transcoding is the enabling technology for our research. Transcoding operations are often performed to fit an object to the characteristics of the display device. Images have been transcoded to thumbnails, grayscale, progressive formats as well as transcoded to textual information. Our focus is on transcoding to reduce the storage requirements of the images. We need to determine the level of transcoding needed to be effective at storage space reduction and in quantifying the actual information loss and computational characteristics of those transcoding operations.

In our companion work [3], we characterized the tradeoffs inherent to a transcoding that changed a JPEG compression metric, such as the JPEG Quality Factor [3]. Whereas in our digital camera application, we have full quality original images, in the general case (e.g. images found on the web) the images have already been processed to some degree. Reconstructing the original Quality Factor that was used to produce the image is necessary so loss in quality becomes meaningful. Using the quantization tables stored in the JFIF [14] headers, we developed an algorithm to predict the Independent JPEG Group's (IJG) [26] equivalent of the JPEG Quality Factor for images compressed using popular JPEG compressors from IJG, Adobe Photoshop and Paint Shop Pro. We utilized results showing that the information quality loss directly corresponds to the change in the JPEG Quality Factor [12, 20].

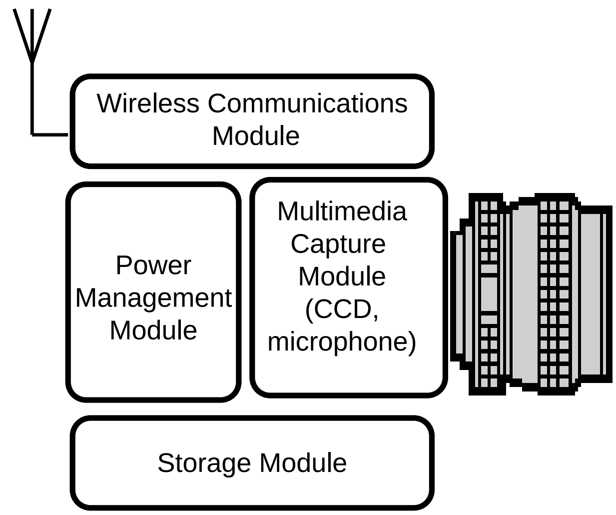


Figure 1: System Architecture

Next, we characterized the computational overhead and the expected change in image size for a particular transcoding. We showed that the computational requirements for a transcoding that changes the JPEG Quality Factor do not depend on the actual Quality Factor change, but on the sum of Minimum Code Unit (MCU) block counts for all the different color space components. We showed that this transcoding can be performed entirely in the frequency domain, avoiding computationally expensive Fourier (FFT) transformations. This aspect of our work has an impact on energy consumption.

We also developed a heuristic to predict if an image will transcode efficiently, wherein it loses more in size than in image quality. We empirically showed that images with high coefficients for low frequency components as well as images with initial JPEG Quality Factor greater than 80 can transcode images efficiently at a significantly better percentage than the base case of all the images. As we will show, the images captured by digital cameras can be efficiently transcoded.

These previous results are the enabling technology for our research effort.

3. Architectural Model

We present the architecture for our digital camera system in Figure 1. Some of the modules such as the wireless communications are not available in contemporary digital cameras. However, we feel that wireless networks are ubiquitous enough that it is a matter of time before wireless communication facilities are added to digital cameras. Our hypothetical digital camera is comprised of four modules:

Optical Module The optical module is comprised of image capture mechanisms such as auto-focus mechanism, light exposure control and flash mechanisms. Our digital camera offers a 2.1 megapixel resolution.

Wireless Communications Module Our digital camera is equipped with a wireless link so that images can be uploaded to a server. Typical wireless links operate at speeds of 19.2 Kbps and hence take a lot of time to upload images. For this study, we assume eventual connectivity (the camera may go through periods of disconnection, but the camera is not being taken on a trip with no connectivity during the entire trip). Even though the users may use this capability to share images by sending them as an email attachment from our camera, for this study, we are only interested in using this wireless link to upload images to reclaim storage space.

Storage Module There are several possible storage technologies that may be used such as floppy disks, memory sticks (flash memory) and microdrives. Each has its unique battery and storage characteristics to quantify the effect of storage management policies on storage space.

Power Management Module For this study, we consider two different battery capacities (typical of existing battery technologies) to see the effect of available energy consumption on the storage management policies.

4. Experiment Objectives and Design

4.1 Objectives

The goal of our digital camera system is to increase the number of images that can be captured and stored without unnecessarily sacrificing image quality. The primary constraint to our ability to continuously capture additional images is the limited storage space and battery capacity available in the mobile camera. We utilize transcoding as the technology to customize image size to fit into the available storage space. We explore the effects of transcoding on increasing the effective storage space subject to constraints on battery power. We also explore the use of wireless networking to improve the effective storage in exchange for the increased battery power requirements.

In such a system, our experiments are designed to answer the following questions:

- Can a digital camera with constrained storage space use dynamic image transcoding to increase the effective storage capacity while still storing reasonable quality images?
- For a digital camera that utilizes dynamic image transcoding, what are the effects of transcoding on battery power consumption? Any energy saving in storing smaller, transcoded images is expected to be offset by the cost to transcode as well as rewrite older images.
- Will the availability of wireless networking enable an increase in the effective storage space visible to the camera? What are the battery power implications of this network connectivity?

4.2 Image Storage Policies

Unlike the storage management requirements in devices supporting productivity applications, the digital camera storage access patterns are dominated by writes with occasional readout of all accumulated data. The major policy issue is what to do when the available storage is exhausted but the demand for the capture of additional images remains. We propose viewing transcoding as a storage reclamation strategy to make space for incoming image data. The initial image placement policy is not an issue we explore.

We envision a system that allows the users to specify four levels of priority on a per-image basis. Images stored with priority 0 are not to be transcoded. Images in priority levels of 1, 2 and 3 may be transcoded to Quality Factor values of 75, 50 and 25, respectively. The transcoder can only transcode images to discrete Quality Factor values of 75, 50 and 25. Typical digital cameras offer such a policy to let the user control the Quality Factor of the images captured at the time the image is captured. For our experiments, we assume the priority of all images is set at the lowest priority level, 3 (Quality Factor 25), since our objective is to study storage management policies rather than model actual user choices.

Specific storage reclamation options include:

No transcoding This policy does not transcode images. As many images are stored at the highest quality as will fit in the limited storage space. When the device is full, additional images are refused. We consider images that do not fit as having zero quality.

Always transcode Conventional digital cameras offer this policy that transcodes all images (as they are generated) to fixed Quality Factor values. For our experiments, we transcode all images immediately to a Quality Factor value of 25.

One shot transcoding This policy initially stores images at highest quality. When additional space is needed, it transcodes images to their specified Quality Factor values to reclaim the required storage space. This policy lowers energy consumption by performing the least number of transcodings. In our experiments, images are transcoded on demand to Quality Factor value of 25.

Gradual transcoding In this policy, images are initially stored at highest quality. When space is needed, this policy gradually reduces the image Quality Factors of the stored images in steps of 25 until images reach their specified Quality Factor values. Just enough transcoding operations are performed on existing images to free up the requisite space for the new image. This policy maintains image Quality Factors at the expense of an increase in the total number of transcodings. In our experiments this means that images are transcoded in steps of 25 down to the minimal Quality Factor of 25.

A variation of any of these policies, to be discussed later, is to use the wireless link to upload images for the purpose of freeing up local storage space.

4.3 Metrics

The important measures for the utility of our approach include (i) the number of images successfully stored (in the limited storage case), (ii) the distribution of Quality Factors after attempting to capture some number of images, n, and (iii) the energy consumed to capture n images. The goals are to achieve a high number of images with high quality, subject to storage and battery life constraints.

In our system, the battery energy required to take a single picture

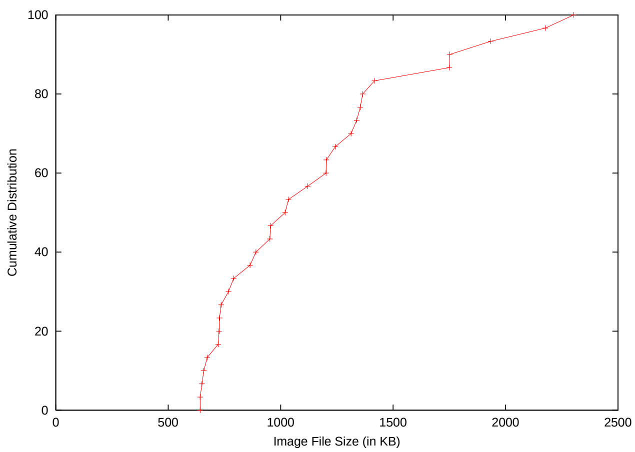


Figure 2: Image File Size Distribution

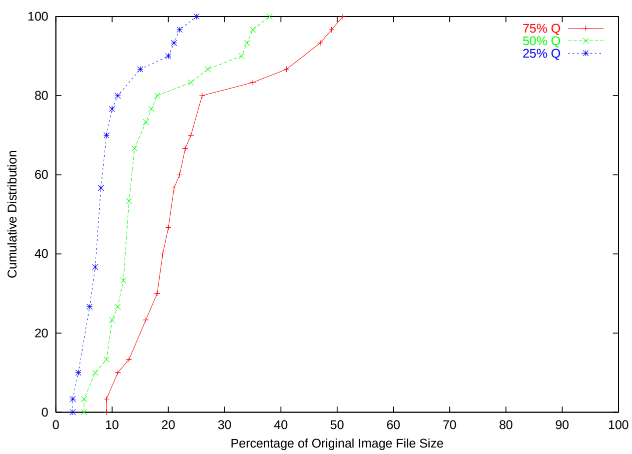


Figure 3: Efficient images for a given loss in image quality

can be quantified as:

$$requiredEnergy = A + S + x * T + y * U$$

A is the energy consumed by the optical module, *S* is the energy consumed to store the image (i.e., writing to the storage module), *T* is the energy consumed to transcode an image and *U* is the energy consumed to upload an image. We could potentially transcode *x* images and upload *y* images to make enough storage space available to store the new image. This calculation does not explicitly address the energy consumed in periods when the camera is on and ready to take a picture, but is not currently being used to do so. This factor is folded into *A*, assuming a fixed time between shots.

Battery lifetime is abstracted as a bound on cumulative energy consumption. The rated capacity of batteries is typically reported in mAh. Consumed energy is measured in Joules (or Ws). We make the simplifying assumption that voltage is constant in order to relate the two for a coarse approximation of battery life in terms of the number of images captured. This is not unreasonable for the type of batteries we are considering.

4.4 Experimental Workload

For our experiments, we utilize 30 high quality images captured with 2.1 megapixel cameras such as Nikon Coolpix 950 [19], Olympus C-2020 and Canon Powershot S10. Most of the images are photographs of test objects under varying lighting and camera settings. Each camera was also used to capture a test pattern. The image samples are available from http://www.imaging-resource.com/. All the images are of the same image geometry (1600x1200) and captured using the highest Quality Factor settings. This has been validated using the algorithm developed in [3]. We computed the IJG equivalent Quality Factor for 5 images to be 92 and the rest to be 97. The file size distribution of the images is shown in Figure 2. From Figure 2, we note that most images are smaller than 1.5 MB with a few images as large as 2.3 MBs. The algorithm developed in [3] predicts that all the images will transcode efficiently, wherein they lose more in image file size for a given loss in image Quality Factor. To verify this prediction, we transcoded the images to Quality Factor values that are 25%, 50% and 75% of the original JPEG Quality Factor values. The image file sizes should at least reduce by 25%, 50% and 75% respectively for the transcoding to be efficient. We plot the results of our experiment in Figure 3. From Figure 3 we note that the transcodings are indeed efficient. This means that using transcoding for storage reclamation will be effective in gaining additional space.

4.5 Architectural Parameters

For our experiments, we use the power and performance specifications of commercially available components. This particular set of devices has not been combined in any digital camera product of which we are aware. Table 1 summarizes the parameters we use.

For the battery, we utilize a NiMH re-chargeable battery similar to the Duracell DR30 [8]. This NiMH battery has a capacity of 2.4 Ah and provides a high rate of discharge. For comparison, we also consider a Lithium primary cell (non-rechargeable) with a lower (220 mAh) rated capacity.

For the storage, we utilize a 8 MB flash memory [10] that draws 325 mW while reading and 400 mW while storing images. This flash memory can transfer images at rates of over 8 MB/s. A separate erase operation is required which uses the same power as the write operation. We also consider a 512 MB PCMCIA microdisk [30] that consumes 3450 mW for spinup and 2125 mW while active. The card supports transfer rates of over 3.7 MB/s.

For the wireless network, we assume a wireless LAN connectivity similar to Rangelan 2 [35]. It consumes 1325 mW to transmit. We experimentally verified that Rangelan 2 supports through-puts in the range of 35 to 120 KB/s within a small apartment. That translates to an energy consumption of 37.8 to 11 mWs/KB. A 600 KB image would consume between 22.7 Ws and 6.6 Ws. We also assume a wireless WAN connectivity using CDPD [1]. It consumes 2500 mW to transmit. CDPD network operate at speeds upto 2.4 KB/s (www.wirelessu.com). This translates to an energy consumption of 1042 mWs/KB. CDPD offered by our local service provider (GTE) costs between 6 and 12 cents per Kbyte depending on the level of usage. For the median images in our workload, this translates to a cost of 63 to 126 dollars per picture, respectively.

We assume a processor similar to the Hitachi Super-H SH-4 (SH7751V) [40] RISC processor. The SH7751V operates at 133 MHz (240 MIPS) and consumes 240 mW under normal operating conditions. For the images in our workload which are of size 1200x1600, JPEG transcoding would need 150*200*3*64 basic computations or about 5.76 million basic computations (based on earlier results [3]). The basic computations consists of 1 multiplication and one arithmetic shift. The exact number of CPU instructions per basic computation depends on the specific processor used. We make a conservative estimate of 10 instructions for the basic computations, or about 60

Component	Operation	Power consumption	Energy Consumption
Flash memory	Read (8 MB/s)	325 mW	39.7 μWs/KB
	Write (8 MB/s)	400 mW	48.8μWs/KB
	Erase (8 MB/s)	400 mW	48.8μWs/KB
Microdisk	Read (3.7 MB/s)	2125 mW	560μWs/KB
	Write (3.7 MB/s)	2125 mW	560μWs/KB
	Spinup (1s)	3450 mW	3450 mWs
Network (Rangelan 2 - 7410)	Transmit - Slow (35 KB/s)	1325 mW	38 mWs/KB
	Transmit - Fast (120 KB/s)	1325 mW	11 mWs/KB
(Aircard 350 PCMCIA CDPD)	Transmit (2.4 KB/s)	2500 mWs	1042 mWs/KB
CPU (Hitachi SH 7751V)	1M instructions	1000 MIPS/W	1 mWs
Optical (LCD,CCD)	One image (5s)	4941 mW	24705 mWs

Battery	Voltage	Rated Capacity
Ni-MH - DR30	6V	14400 mWh
Li/MnO_2- CR 2016	3V	660 mWh

Table 1: Energy characteristics

million instructions. On the SH7751V processor, this should consume about 60mWs.

Based on measurements in [36] we assume the optical module to consume 4941 mW to capture a new image. The optical module in conventional digital cameras have high energy requirements. However, techniques that can intelligently turn-off the CCD and backlight in the LCD monitor can provide significant energy savings. On average, we assume that the user spends 5 seconds to compose and capture each image.

4.6 Implementation Details

For our experiments, we developed a simulator that models the architecture described in Section 3. The simulator models the power consumption of the various units using parameters described in Table 1. The simulator captures images from our collection, performs transcoding operations as needed according to the policies developed in Section 4.2. Actual transcoding operations are performed on a per-image basis with the simulator accounting for available storage and consumed power under a number of different circumstances as described below.

5. Results

We first explore how dynamic image transcoding can increase the effective storage capacity while still storing high quality images. Next we analyze how dynamic transcodings affect battery consumption. Finally, we explore whether the presence of a wireless link can improve the number of images that can be saved subject to battery constraints.

5.1 Storage Constraints

We perform experiments to measure the average image Quality Factor realized while capturing *n* images. For our experiments, we simulate capturing images until the camera runs out of storage space to store the image. We repeatedly use images from the set of 30 described in Section 4.4. Standard flash memory for digital cameras offer 8 MB of storage space. For ease of comparison, we assume an available storage space of 8 MB, both for the flash and microdisk. We explore the storage policies described in Section 4.2.

We plot the average image Quality Factor to capture *n* images in Figure 4. From Figure 4, we note that a policy that does not transcode images can only store about 7 high quality images within the limited storage. The *Always transcode* policy that statically transcodes images to a Quality Factor value of 25 as well as those policies

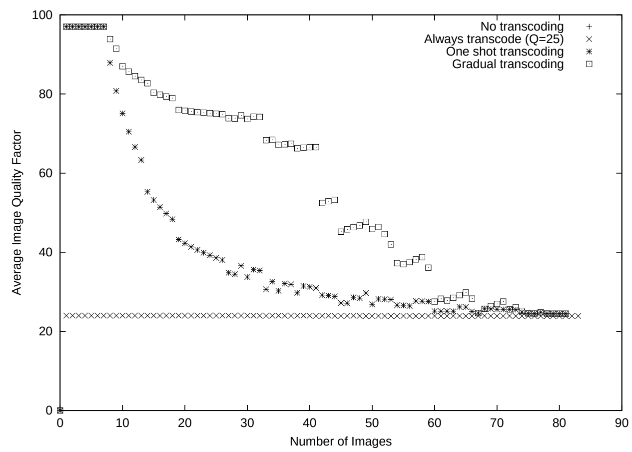


Figure 4: Average Quality to capture *n* images

that dynamically transcode images based on the available storage space, either *One shot* or *Gradual*, can store about eighty five images with an average Quality Factor value of 25. The policy that statically transcodes the images always stores images at a Quality Factor of 25, even when fewer images are desired and adequate storage remains available. The *One shot* policy that transcodes images to Quality Factor values of 25 just when needed to reclaim space reduces the average image Quality Factors rapidly once the available storage space to store original images is exhausted (after 7 pictures). The average Quality Factor value reflects the relative number of original images to images which have already been transcoded to Quality Factor 25. By the time 40 images have been acquired, the average Quality Factor is near 30. On the other hand, the dynamic policy that gradually reduces the image Quality Factor values can offer reasonable Quality Factors of 75 for capturing up to 40 images. As more images are added, the *Gradual* transcoding policy has to be more aggressive to free up enough space. Once it frees up enough space for an incoming image it often creates sufficient room to accept a few more. Thus, we observe a pattern where the average Quality Factor value drops in steps and then increases a bit before another drop. *Gradual* still delivers better average quality than the *One shot* or *Always transcode* policies until around sixty images are captured.

5.2 Battery Constraints

Next, we analyze the effects of transcoding on the limited battery power available in the digital camera. We repeat the earlier simulation experiments using the energy consumption characteristics of flash memory and PCMCIA microdisk. For simplicity, both the storage technologies are assumed to offer a storage space of 8 MB. The cumulative energy consumption using a flash memory and microdisk, as a function of the number of images captured, is plotted in Figures 5(a) and 5(b) respectively.

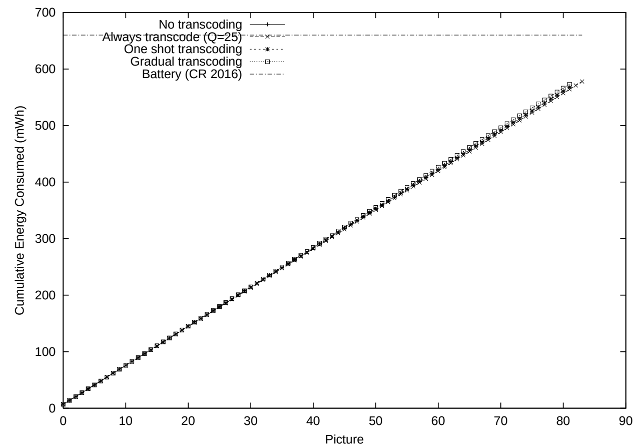
From Figure 5(a), we note that transcoding, both statically and dynamically, has little effect on the overall energy consumption of the digital camera using flash memory. This is to be expected given that the energy consumption of the optical module far outweighs the energy consumption of the storage module (Table 1). However, techniques that can intelligently turn-off the CCD and back-light in the LCD monitor can provide significant energy savings for the optical module. A coin cell such as CR 2016 will be able to supply enough energy before the camera completely runs out of storage space. The NiMH battery is more than adequate to supply the energy needs of the images that can fit in this constrained storage space.

On the other hand, from Figure 5(b) we note that a microdisk consumes more energy for transcoding images. In fact, using the CR 2016 battery, the dynamic *One shot* policy and *Always transcode* policy will run out of battery after 81 pictures, even though there is enough storage for 85 pictures. A dynamic policy that gradually transcodes the image will run out of battery power after only 76 pictures.

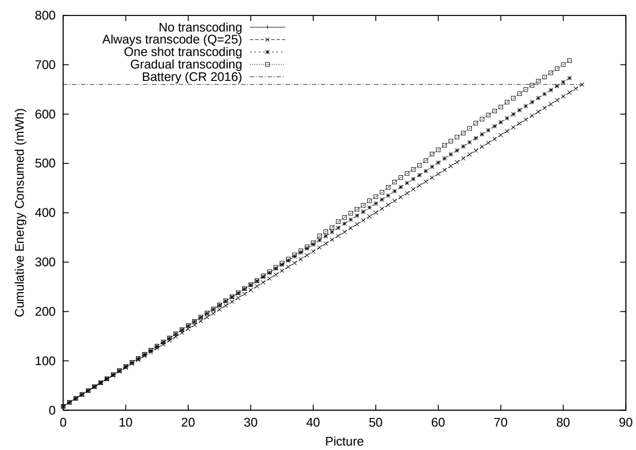
If we increase the disk capacity to 260 MB and use the NiMH battery , then the higher capacity battery edges out storage as the limiting factor on the number of transcoded images (to Quality Factor 25) that can be captured. However, with the estimated ability to take and store over 1800 images, it is hardly a bottleneck.

While it is clear that the optical component dominates the energy consumption, Figure 5(b) shows that storage management has some noticeable impact. By removing the contribution of the optical system, we can understand the role played by the storage module and associated computation. We plot the cumulative energy consumed by the storage module to read, write, erase, spinup and transcode *n* images using the microdisk for the different storage reclamation policies (Section 4.2) in Figure 6. Note the different *y* axis scale for the different graphs.

For a policy that always transcodes the images statically (Figure 6(a)), we note that disk spinup operations dominate the power consumption of the storage module. For a policy that dynamically transcodes the images to a fixed Quality Factor value of 25 (Figure 6(b)), disk spinup still dominates the power consumption, but there is noticeable power consumption from the extra read and write operations associated with the transcoding operations applied to previously stored images. However, a policy that gradually reduces the image Quality Factor (Figure 6(c)) consumes significant power to read and write objects, especially after capturing 40 images. The images in our collection lose relatively more storage space for a transcoding that reduces the Quality Factor of an image in steps of 25 than a transcoding that reduces the Quality Factor of an image in a single step to 25 (Figure 3). Hence more images need to be transcoded to effect similar reclamations in storage space. These results suggest that a disk-based camera could benefit from using some amount of memory buffering to avoid spinup of the disk on a per-image basis as an effective way to improve battery life.



(a) Flash memory (8 MB)



(b) Microdisk (8 MB)

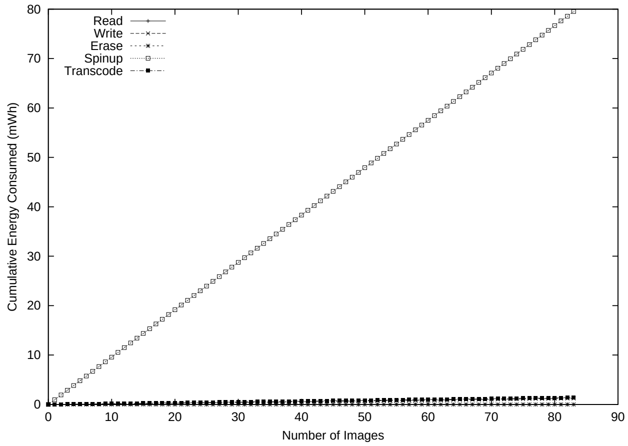
Figure 5: Cumulative Energy Consumption

We plot the individual power consumption components for the flash memory in Figure 7. Note that flash memory does not consume any spinup power. The magnitudes of energy required to read and write data are much smaller than the microdisk. In fact, the energy to transcode dominates the power consumption. For a policy that gradually transcodes the images (Figure 7(c)), the storage and transcoding components consume about 13 mWh of energy to capture 80 images (as compared to 175 mWh while using the microdisk).

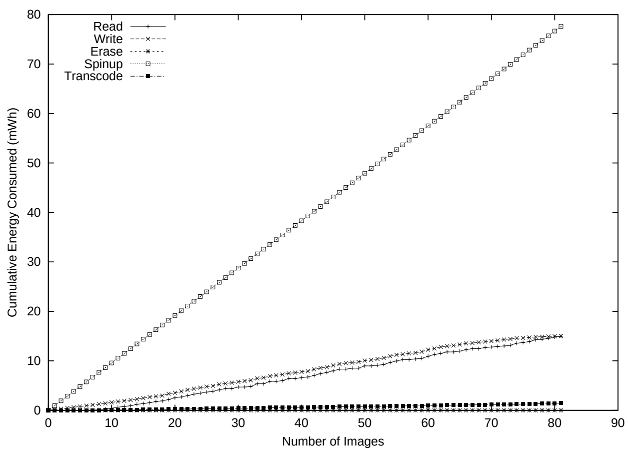
We note that a dynamic policy that transcodes the images to a fixed Quality Factor value of 25 (Figures 6(b),7(b)) consumes much less energy than a policy that gradually transcodes the images (Figure 6(c),7(c)).

5.3 Network Connectivity

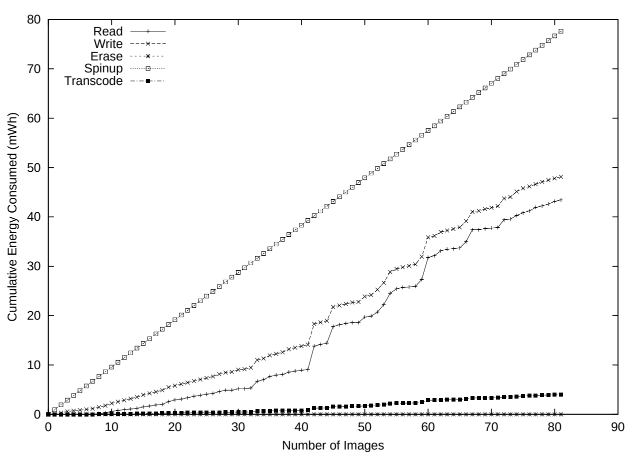
With the availability of ubiquitous network technologies, we explore the usage of wireless networking on the digital camera as a storage management technique. However, there is also an inherent battery cost in transmitting images. The camera stores images in the highest Quality Factor in the local storage and once it runs out



(a) Always transcode

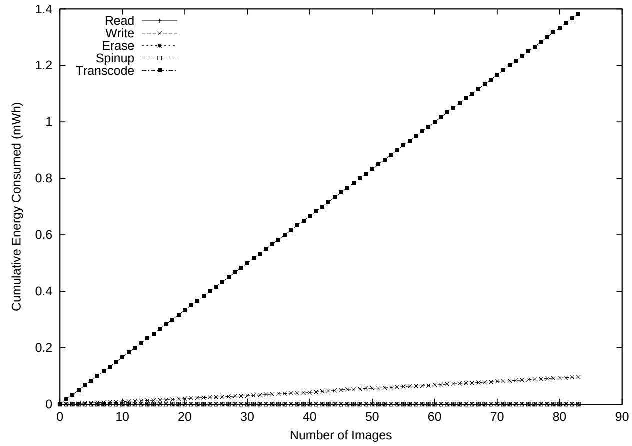


(b) Transcoding (one shot)

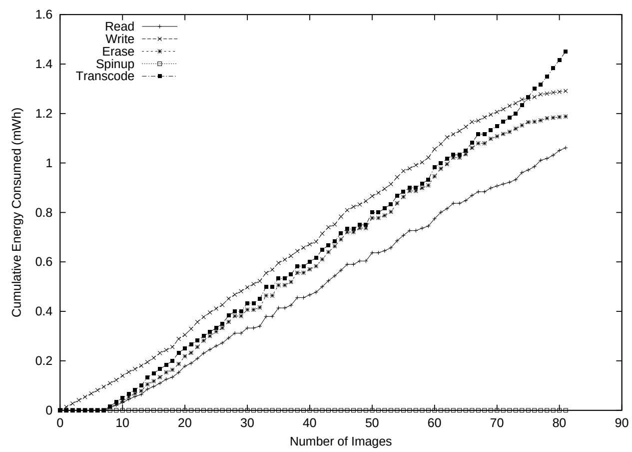


(c) Transcoding (gradual)

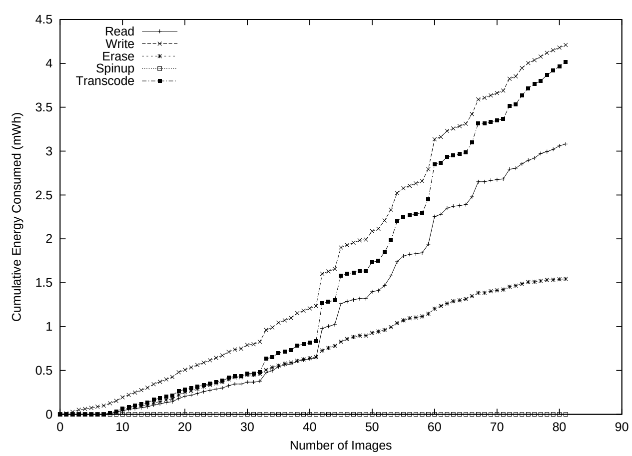
Figure 6: Energy consumed by the storage operations for a Microdisk (8 MB)



(a) Always transcode



(b) Transcoding (one shot)



(c) Transcoding (gradual)

Figure 7: Energy consumed by the storage operations for a Flash (8 MB)

of storage space will transmit any future images over the wireless network. For this experiment, we assume that there is a secure, dedicated remote storage area that the camera is aware of. We assume that there is enough time between image captures to offset the delay in transmitting the images. However, for the median size image in our workload, these transmission delays are estimated to be between 9 (120 KB/s) and 30 seconds (35 KB/s) for the Rangelan 2 network and 7.25 minutes (2.4 KB/s) for the CDPD network.

The cumulative energy consumed by the camera using slow connectivity and fast connectivity using the battery power characteristics of the Rangelan 2 network and CDPD network (Table 1) are plotted in Figures 8(a), 8(b) and 8(c) respectively. For the Rangelan 2 network, using the CR 2016 battery (which can capture over 76 images using transcoding techniques), we can capture at most 35 images (only 28 images can be transmitted). From Figure 8(b), we note that faster connectivity reduces the battery consumption considerably with the camera able to capture 58 images at high quality (51 transmitted). For the CDPD network (Figure 8(c)), using the CR 2016 battery, we can only capture 9 pictures (2 transmitted). Using the high capacity NiMH DR30 battery, we can capture 44 pictures. At a cost of 63 to 126 dollars to transmit an image, CDPD networks may only be used for very important images. For example, a news reporter may upload an important news picture directly to the printers' without concern for the cost to upload images.

Hence, we note the need to capture the power tradeoffs for various operations to allow the designers of multimedia devices to effectively communicate these tradeoffs to the end user. The decision on whether to locally store a lower quality image or a higher quality image over the network has to be presented to the end user with information about the energy/quality tradeoff involved.

6. Related Work

6.1 Power Aware Systems

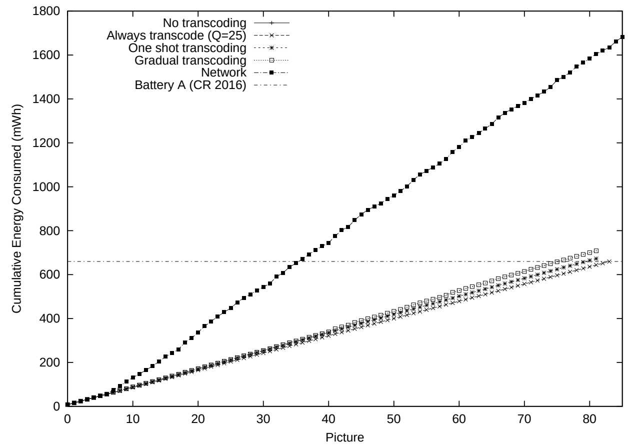
There has been considerable work on power management for components of our digital camera. This work includes spindown policies for disks and alternatives [2, 5, 6, 7, 16, 24, 28, 44], and managing wireless communication [18, 23, 41, 39]. Lorch et al. [29] present a survey of the various software techniques for energy management. Our camera should be able to exploit power management techniques for the components upon which it depends. Unfortunately, our usage model differs in significant ways from those assumed in most of those studies.

6.2 Mobile Storage Systems

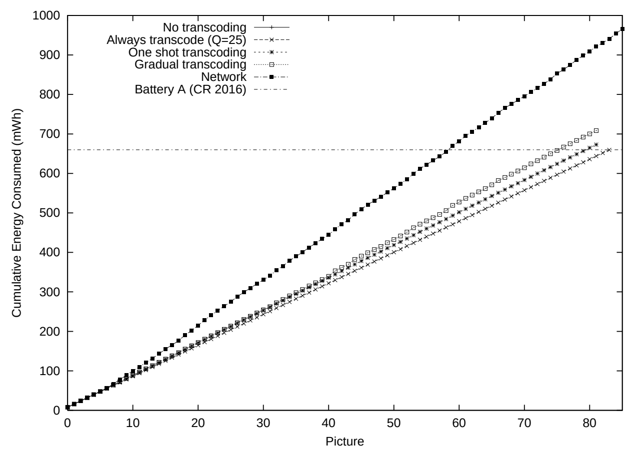
Previous work on storage systems for mobile computing devices has focused on disconnected/ weakly connected file systems [38, 22, 37]. The issues investigated have been hoarding [25, 27, 42] and consistency [43, 34]. Mobile web browsing applications face similar problems with prefetching data for use while disconnected [17, 21]. These studies assume a very different access pattern from the create-dominated workload of this application. Thus, the issue of reclaiming storage has never been of central interest.

The most similar work is the Compression Cache, proposed by Douglis [4]. The goal was to fit more data into the small memories of mobile computers, reducing reliance on secondary memory and connectivity. In the compression cache, lossless compression was necessary. We exploit the feature of multimedia data that makes it amenable to lossy compression.

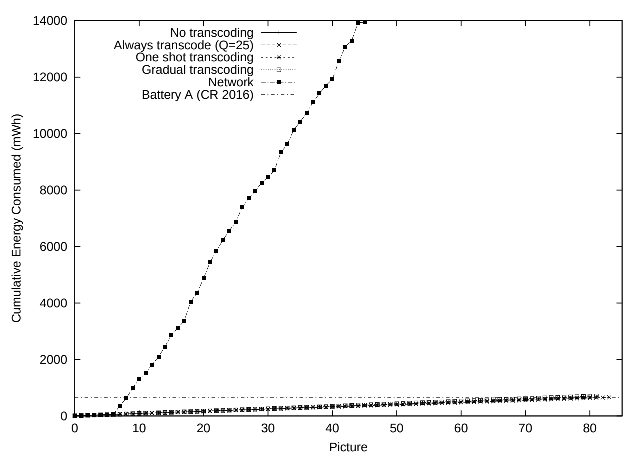
6.3 Transcoding



(a) Slow connectivity - Rangelan 2



(b) Fast connectivity - Rangelan 2



(c) Cellular connectivity - CDPD

Figure 8: Energy consumed by digital camera using Rangelan 2 and CDPD network

A number of systems [13, 31, 32, 33] have used transcoding to fit images to the current operating environment. However, there has been little formal work in conducting a systematic study to measure the information loss associated with a given transcoding, so previous systems performed ad hoc transcoding without an explicit understanding of the tradeoffs and potential gains.

Han et al. [15] present an analytical framework for determining whether to transcode and how much to transcode an image. However, their quantification does not take the image information quality into account and hence the information quality loss is not quantified. Our work relies on a study [12] of various objective image quality and color reproduction measures to quantify the subjective effects of lossy image compression which concluded that the JPEG Quality Factor is a good representation of the subjective, perceived image quality.

The work described in [3] gives assurance that a transcoding operation applied to the kind of images we consider will be efficient in terms of reclaiming storage effectively compared with the associated quality lost. It can be shown that without such a characterization, transcoding operations can be applied which are counterproductive, producing *larger* file sizes and lower quality.

Recent work with Odyssey [11] demonstrates a link between adjusting the fidelity of data and energy consumption. Similarly, our *Always transcode* policy should benefit from smaller image sizes on the initial write operation, but the effect was imperceptible.

7. Conclusion

In this paper, we explore an informed, dynamic transcoding policy as a storage management technique for an image capture application. We use the specific example of a digital camera, although the ideas are relevant to other devices that capture and store multimedia data that is amenable to transcoding. This might include various sensors, video cameras, or audio recording devices. During data capture, the access patterns are dominated by writes of newly generated data. We consider the impact of storage and battery constraints on the number and quality of images that can be stored. We also investigate the option of using wireless networking to store image data remotely. We show that:

- *Gradual* transcoding can simultaneously increase the number of images that can be stored, over a policy of doing *No transcoding*, while preserving reasonable quality, with a moderate number of images (e.g. 40), as compared to a policy that *Always transcodes* to the lower Quality Factor. Pushing the limit of the storage with the maximum number of pictures, all transcoding policies deliver the same average Quality Factor. The *Gradual* transcoding policy can be viewed as an adaptive strategy that tries to reflect the user's apparent preference for quality vs. quantity.
- The optical system currently dominates the energy consumption. In perspective, transcoding has minimal impact on battery life. However, the disk spinup for every image in a camera based on a microdisk for storage has a noticeable effect, as does the extra reads and writes for transcoding previously stored images in the *Gradual* transcoding policy. The lower capacity Lithium battery that we consider in our experiments becomes a constraint for this case. One possible interpretation may be that flash memory is preferable in such an application.
- While exploiting wireless networking is an attractive alternative for preserving high quality images in spite of limited local storage, the impact on the battery lifetime is significant with the existing wireless technology. The battery capacity becomes a significant constraint. The prohibitive cost to use wireless links (63 to 126 dollars per image) forces users to use such network for only the most important images.

8. Acknowledgments

This work was supported in part by a graduate fellowship from North Carolina Networking Initiative (NCNI).

9. REFERENCES

[1] Air Card 350 Type II CDPD PCMCIA Card.

[2] M. Baker, S. Asami, E. Deprit, J. Ousterhout, and M. Seltzer. Non-volatile Memory for Fast, Reliable File Systems. In *Proceedings of the 5th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 10–22, October 1992.

[3] S. Chandra and C. S. Ellis. JPEG Compression Metric as a Quality Aware Image Transcoding. In *2nd Symposium on Internet Technologies and Systems*, Boulder, CO, Oct. 1999. USENIX.

[4] F. Douglis. The compression cache: Using on-line compression to extend physical memory. In *Proceedings of 1993 Winter USENIX Conference*, pages 519–529, January 1993.

[5] F. Douglis, R. Caceres, B. Marsh, F. Kaashoek, K. Li, and J. Tauber. Storage Alternatives for Mobile Computers. In *Proceedings of the First Symposium on Operating Systems Design and Implementation (OSDI)*, pages 25–37, November 1994. Monterey, CA.

[6] F. Douglis, P. Krishnan, and B. Bershad. Adaptive Disk Spin-down Policies for Mobile Computers. In *2nd USENIX Symposium on Mobile and Location Independent Computing*, April 1995. Monterey CA.

[7] F. Douglis, P. Krishnan, and B. Marsh. Thwarting the Power Hungry Disk. In *Proceedings of the 1994 Winter USENIX Conference*, pages 293–306, January 1994.

[8] Duracell, duracell.com/OEM/Pdf/others/TECHBULL.pdf. *Ni-MH Rechargeable Batteries*.

[9] Eastman Kodak Company, Microelectronic Technology Division, Rochester, NY. *KAF 16800 4096x4096 Pixel Full Frame CCD Image Sensor*, revision 2 edition, Apr. 1999.

[10] Flash digital camera memory card. www.actiontec.com.

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

[12] A. M. Ford. *Relations between Image Quality and Still Image Compression*. PhD thesis, University of Westminster, May 1997.

[13] A. Fox and E. A. Brewer. Reducing www latency and bandwidth requirements via real-time distillation. In *Proceedings of Fifth International World Wide Web Conference*, pages 1445–1456, Paris, France, May 1996.

[14] E. Hamilton. *JPEG File Interchange Format - Version 1.02*. C-Cube Microsystems, 1778 McCarthy Blvd, Milpitas, CA 95035, Sep. 1992.

[15] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas. Dynamic adaptation in an image transcoding proxy for mobile web browsing. *IEEE Personal Communications Magazine*, 5(6):8–17, Dec. 1998.

[16] D. Helmbold, D. Long, and B. Sherrod. A Dynamic Disk Spin-Down Technique for Mobile Computing. In *Proc. of the 2nd ACM International Conf. on Mobile Computing (MOBICOM96)*, pages 130–142, November 1996.

[17] B. C. Housel and D. B. Lindquist. ARTour Web Express: A system for optimizing web browsing in a wireless environment. In *MOBICOM 96*, November 1996.

[18] T. Imielinski, M. Gupta, and S. Peyyeti. Energy Efficient Data Filtering and Communications in Mobile Wireless Computing. In *Proceedings of Usenix Symposium on Location Dependent Computing*, April 1995.

[19] N. U. Inc. Digital camera E950. nikonusa.com, 1999.

[20] R. E. Jacobson, A. M. Ford, and G. G. Attridge. Evaluation of the effects of compression on the quality of images on a soft display. In *Proc. of SPIE: Human Vision and Electronic Imaging II*, San Jose, CA, Feb 1997.

[21] F. Kaashoek, T. Pinckney, and J. Tauber. Dynamic documents: Mobile wireless access to the www. In *Proc. of IEEE Workshop on Mobile Computing Sys. and Apps*, December 1994.

[22] J. J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. *ACM Trans. on Computer Systems*, 10(1):3–25, February 1992.

[23] R. Kravets and P. Krishnan. Power Management Techniques for Mobile Communication. In *Proc. of the 4th International Conf. on Mobile Computing and Networking (MOBICOM98)*, pages 157–168, October 1998.

[24] P. Krishnan, P. Long, and J. Vitter. Adaptive Disk Spin-Down via Optimal Rent-to-Buy in Probabilistic Environments. In *Proceedings of the 12th International Conference on Machine Learning*, pages 322–330, July 1995.

[25] G. Kuenning. The design of the seer predictive caching system. In *Proc. of IEEE Workshop on Mobile Computing Sys. and Apps*, December 1994.

[26] T. Lane, P. Gladstone, L. Ortiz, J. Boucher, L. Crocker, J. Minguillon, G. Phillips, D. Rossi, and G. Weijers. The independent JPEG group's JPEG software release 6b. ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz.

[27] H. Lei and D. Duchamp. An analytical approach to file prefetching. In *Proc. 1997 USENIX Technical Conf*, January 1997.

[28] K. Li, R. Kumpf, P. Horton, and T. Anderson. A Quantitative Analysis of Disk Drive Power Management in Portable Computers. In *USENIX Association Winter Technical Conference Proceedings*, pages 279–291, 1994.

[29] J. R. Lorch and A. J. Smith. Software strategies for portable computer energy management. *IEEE Personal Communications Magazine*, 5(3):60–73, Jun. 1998.

[30] C. T. Ltd. PCMCIA II 260 MB disk drive (CT260T2). www.callunacard.com. IBM Microdisk.

[31] M. S. Mazer, C. Brooks, J. LoVerso, L. Theran, F. Hirsch, S. Macrakis, S. Shapiro, and D. Rockwell. Distributed clients for enhanced usability, reliability, and adaptability in accessing the national information environment. Technical report, The Open Group Research Institute, 1998.

[32] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Application-aware adaptation for mobility. In *Proceedings of the 16th ACM Symposium on Operating Systems and Principles*, Saint-Malo, France, Oct. 1997.

[33] A. Ortega, F. Carignano, S. Ayer, and M. Vetterli. Soft caching: Web cache management techniques for images. In *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, Princeton NJ, Jun 1997.

[34] G. Popek, R. Guy, T. Page, and J. Heidemann. Replication in ficus distributed file systems. In *Proc. Workshop on Management of Replicated Data*, November 1990.

[35] Proxim Inc., www.proxim.com. *RangeLAN 2 - PC Card*.

[36] J. M. Rommel. Power measurements for Apple Quicktake 200.

[37] M. Satyanarayanan, J. Kistler, L. Mummert, M. Ebling, P. Kumar, and Q. Lu. Experience with disconnected operation in a mobile computing environment. In *Proc. USENIX Symp. on Mobile and Location-Independent Computing*, August 1993.

[38] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere. Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on Computers*, 39(4), Apr. 1990.

[39] S. Singh, M. Woo, and C. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of MOBICOM*, pages 181–190, October 1998.

[40] J. Slager. *Hitachi's SuperH Risc Family - The SH7751 (SH4 with PCI) microprocessor*. Advanced Microprocessor Core Development, Hitachi Semiconductor (America) Inc, Hitachi Semiconductor (America) Inc, San Jose, CA, May 1999.

[41] M. Stemm and R. Katz. Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices. In *Proceedings of 3rd International Workshop on Mobile Multimedia Communications (MoMuC-3)*, September 1996.

[42] C. Tait, H. Lei, S. Acharya, and H. Chang. Intelligent file hoarding for mobile computers. In *ACM Conference on Mobile Computing and Networking (MOBICOM 95)*, November 1995.

[43] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser. Managing update conflicts in a weakly connected replicated storage system. In *Proc. 15th SOSP*, December 1995.

[44] J. Wilkes. Predictive Power Conservation. Technical Report HPL-CSP-92-5, Hewlett-Packard Labs, February 1992.