Passive Capture and Ensuing Issues for a Personal Lifetime Store

Jim Gemmell*, Lyndsay Williams**, Ken Wood**, Roger Lueder* and Gordon Bell*

*Microsoft Research
455 Market St., #1690
San Francisco, CA
USA 94105

**Microsoft Research
Roger Needham Building
JJ Thomsom Ave.
Cambridge, UK, CB3 0FB

{jgemmell, lyn, krw, gbell, rlueder}@microsoft.com

ABSTRACT
Passive capture lets people record their experiences without having to operate recording equipment, and without even having to give recording conscious thought. The advantages are increased capture, and improved participation in the event itself. However, passive capture also presents many new challenges. One key challenge is how to deal with the increased volume of media for retrieval, browsing, and organizing. This paper describes the SenseCam device, which combines a camera with a number of sensors in a pendant worn around the neck. Data from SenseCam is uploaded into a MyLifeBits repository, where a number of features, but especially correlation and relationships, are used to manage the data.

Categories and Subject Descriptors
H.3.2 [**Information Storage and Retrieval**]: Information Storage – *File Organization*
H.5.4 [**Information Interfaces and Presentation**]: Hypertext/ Hypermedia – *Architectures, Navigation, User issues*

General Terms
Management, Design, Human Factors.

Keywords
SenseCam, MyLifeBits, photo, multimedia, database.

1. INTRODUCTION
"When I had my first child, I bought a camera and took many pictures" said a friend, "but eventually I realized I was living behind the camera and no longer taking part in special events. I gave that up – now I don't have nearly as many pictures of my second child." Nearly every one can identify with this story. There is a strong demand for capture of life experiences, whether in photos, videos, or written accounts. However, few people want to miss the experience in order to be the camera operator. Furthermore, many people who have stuck with photography, and especially digital photography, have ended up feeling overwhelmed by their large collection of photos, and only get enjoyment from a chosen few that are selected for albums.

This paper describes hardware and software developed to address these issues. SenseCam is a device that combines a camera with a number of sensors. It is worn on a pendant hung from one's neck. The sensors are used to automatically take pictures at "good" times. Their sensor information is also recorded, and is uploaded along with the photos to a MyLifeBits repository. MyLifeBits is a system for storing a lifetime's worth of media, with a database at its heart. In this paper, we will describe some aspects of MyLifeBits that enable management of SenseCam photos and data.

Use of SenseCam and MyLifeBits should enable users to fully participate in events and yet have a rich record of those events that can be shared with others. The photos and sensor data recorded may also serve to help the individual, both with memory recall, and with gleaning insight into one's life, for example, how one's time is spent, or how certain factors (like temperature or physical activity) may have an impact on health.

In the remainder of this paper, we describe the design and implementation of SenseCam (Section 2) and MyLifeBits (Section 3). This is followed by a discussion of our experience and ensuing issues (Section 4), a survey of related work (Section 5), and our conclusion.

Figure 1 - SenseCam hardware

2. SENSECAM

The SenseCam prototype is the size of a pager (Figure 1) and is attached via a neck strap or clip to the front of the user's body. The hardware is based on a PIC microcontroller with sensor inputs from a 2-channel accelerometer, digital light sensor, temperature sensor and passive infra red sensor for detecting living beings. A commercial camera module is interfaced to the microcontroller and photos are triggered by sensor data and/or time interval. Table 1 enumerates the hardware used in the SenseCam. We record sensor data every second; a total of 18 bytes (see Table 2). 18 bytes/sec corresponds to only 760 KB in our target of 12 hours of operation, well below the 64MB of MMC memory that SenseCam provides for sensor data storage. The camera module has 128Mbyte of FLASH memory, capable of storing approximately 2000 VGA images. Image capture may be triggered by elapsed time, a change in light level, motion, the heat from a person in front of camera or ambient temperature change.

Our first goal for SenseCam was to record a person's environment as they passed through different rooms. For this purpose, a transition between rooms serves as a good point to take a photo of the room being entered. This transition can be measured by a change in light level as the user walks through doors. After some experimentation, it was observed that an increase of 100% or decrease in 50% in light level from one sample to the next (one second apart) was adequate to detect this room transition.

Table 1- SenseCam hardware specification

PIC 16F876 8 bit microcontroller
I2C bus interfaces sensor/peripheral devices to the microcontroller
ADXL210 2 channel accelerometer/ tilt measurement , +/- 10g – used for motion sensing. This also provides static tilt angle of the camera +/- 90 degrees in 2 axis
Passive infrared sensor (Seiko)– detects heat from a person at up to 2.5m
Digital Light Sensor (TCS230) – records light intensity (LUX) and also individual red, green & blue intensities.
Temperature sensor (LM75) - 0 to 70C
Real time clock – to record calendar data of time of image
RS232 serial interface (for loading data to PC)
MMC Card 64 MB (for sensor data)
Power management integrated circuit (Maxim)
Camera module 128 MB – various commercial modules used
Edmunds Optics lens – 2.2mm, f2, ultra wide angle (132 degrees)
2 x AA NiMh cells for recharging after 12 hours.

Image capture is also triggered by a person passing in front of the camera, as detected by the passive infra red (PIR) sensor. This is the same type of sensor as used in indoor burglar alarms. However in the wearable device, false triggers could be obtained from camera motion across the scene. We reduce these false positives by exploiting the motion sensors to record an image only when a PIR event coincides with a relatively stable camera position.

The ADXL210 tilt sensor is used to detect transitions of a person's state, i.e. standing up, lying down, walking etc. However on capturing images using this movement detection, it was discovered that approximately 50% of the images were blurred due to camera motion. However, it is possible to use the tilt sensor to determine an appropriate time to capture a non-blurred image. Some experiments were performed, the results of which established that movement of less than 20 degrees/second resulted in low blur images, even at indoor light levels. Therefore, the SenseCam software was modified so that whenever a trigger event (light level, motion, or PIR) is detected, motion is monitored at 100ms intervals until it is low enough to avoid image blur. This reduced the occurrence of blurred images from 50% to less than 10%.

Table 2 – SenseCam sensor data

SAMPLE DATA	BYTES
Light intensity (LUX)	2
Red, green, blue intensity	6
X tilt (degrees)	2
Y tilt (degrees)	2
Temperature (degrees C)	1
Bit flag for each PIR, light, temperature, tilt change that is detected	1
Corresponding image number	2
Overhead	2
TOTAL	18

SenseCam has no buttons or display. However, it does have a UI to support user-directed tasks in addition to passive capture. In addition to passive capture of images, it is also possible to intentionally capture an image. Intentional capture can be triggered by simply moving a hand across the front of camera: the shadow creates a light change and thus an image is captured. Another aspect of the SenseCam UI is the recognition of a "pouring" gesture: the user tilts the SenseCam so that it is inverted (within 10 degrees of vertical) in order to initiate transfer of data to a PC.

The sensors are also used for power management of the camera module, shutting it down when no photos would be taken. The camera is shut down if movement change is less than 10 degrees and light change is less than +100%/-50%. It takes 4 seconds to power the camera back up. With this power management, two AA NiMh batteries suffice for 12 hours of operation.

To be truly passive, the user must not be worried about pointing the camera in a precise direction. As the SenseCam is worn on the body and the user does not use a viewfinder, the aim can be unpredictable. A normal lens has too narrow a field of view, yielding many photos that "miss the point." We used a 2.2mm ultra wide-angle lens to provide 130 degrees of view and have tested wider-angle lens up to 180 degrees to capture fish eye view. These allow focus from 40cm (or less) to infinity. By way of comparison, the eye typically has 95 degrees of view. The advantage of a very wide-angle lens for the SenseCam is that most or all of the forward view is captured with a large depth of field. With the wide-angle and large depth of field it is rare that the camera misses what the user is seeing.

For location data, we are using a handheld GPS at present but in future we will use an onboard GPS unit with sensor power management. Currently, GPS units consume too much power to be run continuously, but motion and light sensors can be used to power down the GPS and camera when there are reduced sensor stimuli.

The type of scenarios which originally motivated SenseCam center on personal memory recall. For example, where did I leave my spectacles or keys? What was the name of that expensive bottle of French wine I had 3 weeks ago, but have forgotten? What did my family do today? For example, mother (a researcher) is always busy working and father takes daughter ice-skating. All the exciting events (and falls measured by accelerometer and so images of which were captured) can be replayed by mother after she gets home. What happened to me? If one is cycling and falls injured by the side of the road, the emergency medics can review the data in the SenseCam "black box" accident recorder (e.g. images and acceleration as one hits the ground) and give appropriate medical treatment.

Integration with MyLifeBits provides an even wider range of possibilities for exploiting SenseCam data and relating it to other aspects of one's personal history.

3. MYLIFEBITS

MyLifeBits is a lifetime personal digital store. An early version was first described at ACM Multimedia 2002 [5]. The version at that time included core features such as using a database for storage, full-text search, filing in zero or more "collections" instead of strict hierarchy, text and audio annotations. Since then, it has expanded in many directions, including new forms of capture (e.g., radio and TV) and more general use of typed links [3,4]. Most recently, it has been extended to act as the repository for SenseCam photos and data.

An import program uploads SenseCam photos and sensor data into MyLifeBits. The photos are in JPEG format and are stored just like any other JPEG photos in MyLifeBits, with attributes that include date/time taken, location, and camera make. The sensor data is stored in tables in the MyLifeBits database. All sensor values include the date/time of the sensor reading.

GPS location readings, consisting of date/time, latitude, longitude, elevation, and precision, are also loaded into a table in the database. Date/time correlation between photos and GPS readings are then used to set the location in the photos whenever possible. The location is set for the photo rather than relying on time correlation with the GPS table because a user may obtain photos from a third party, in which case a correlated time stamp is not meaningful. Furthermore, these photos may already have their location set by the third party.

One enjoyable use of SenseCam photos is to do a quick "replay" of one's day. We have implemented a simple form of Rapid Serial Visual Presentation (RSVP) [14] that flips through the photos (Figure 2). The speed is adjustable, in either forward or reverse, allowing both slide-show like display, and a rapid visualization that is much like watching a movie in fast forward mode. Using RSVP, one can quickly re-live a day, or search for some moment of interest. There is significant psychological evidence to suggest that RSVP techniques are especially suitable to images which are related and which derive from one's own experience, and SenseCam images fit this bill ideally. See [14] for a good overview of RSVP and its application to computing interfaces. We are also considering additional techniques for presenting the sensor, time and location data alongside an RSVP presentation of the images. By concentrating on the browsing interface, perhaps even using specialized hardware, we could provide the envisaged "black box" memory aid with minimal image processing or machine learning.

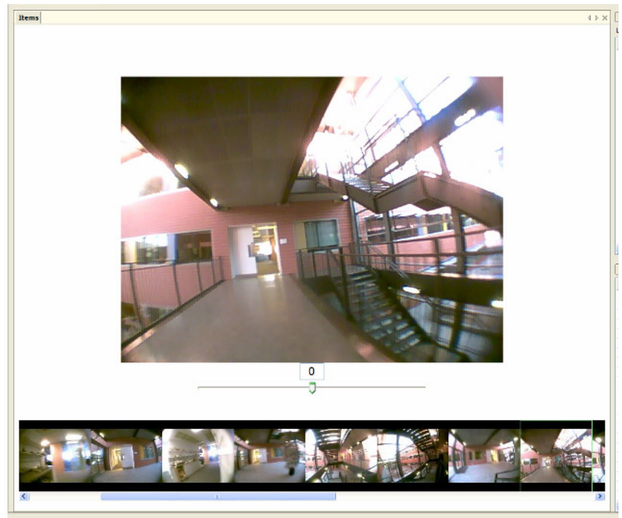

Figure 2 - RSVP viewer for SenseCam images. The slider below the large image controls speed/direction. The "filmstrip" at the bottom shows images adjacent in time.

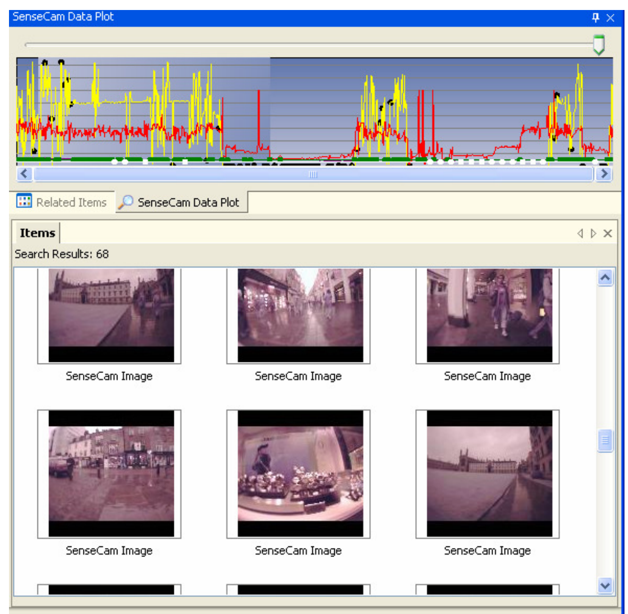

Figure 3 - SenseCam sensor readings plotted in upper pane. Thumbnails of photos in lower pane. Selecting a region of sensor readings updates the contents of the lower pane to only photo from the selected time region.

Sensor readings are displayed in a graph, with black dots for every picture that was taken (Figure 3). These values may be interesting in their own right: one may wonder "Just how hot was it in the backyard yesterday?" or, want to glance at light level to get an indication of time spent indoors vs. outdoors. A range of the graph may be selected using the mouse. The pane of photos below is then updated to only include those photos taken during the time range selected. This allows the user to look for photos based on sensor values.

GPS locations can also be used to browse photos. A window pane shows a map with photo locations marked (Figure 4, Figure 5). Zooming and panning on the map issues a new query for photos from the region of the visible map to be shown in the corresponding pane. Photos that have no location (e.g., old photos) can be dragged and dropped onto the map to set their location. The mapping interface was ported from the WWMX, and more details on its appearance and operation can be found in a paper describing WWMX [15].

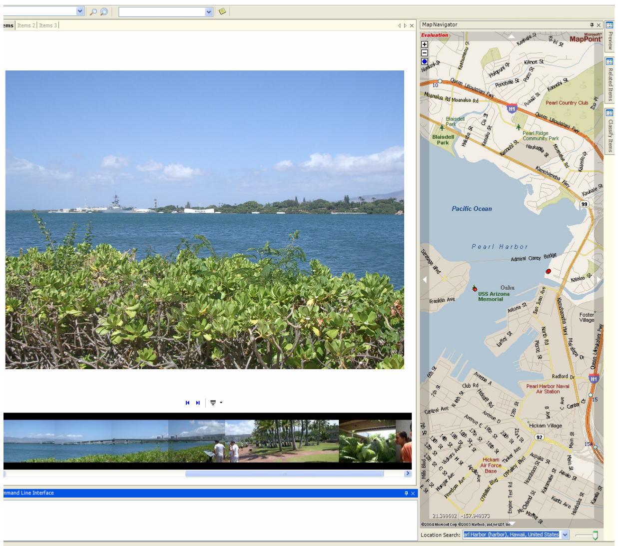

Figure 4 - Location user interface: map on right shows dots where photos are taken. Pane on left only shows photos taken in the are shown on the map.

Figure 5 – Close-up of map from Figure 4. Red dots show photo location.

Location & time allow for powerful story-telling of events, especially vacations. MyLifeBits makes it trivial to annotate photos and view them on a map. Thus, with little effort, one may create a travelogue.[1]

We have shown how sensor values and location can be used to browse and find photos. These values are potential "memory handles" for the user. Perhaps you remember the photo you want was taken on a spring day that was unseasonably hot, so a graph of temperature will help you find it. Or you might remember that it was taken on a certain area of the city, so that zooming in with the map will suffice to find it.

There are many other possible memory handles that MyLifeBits can exploit. For example, suppose that you are discussing selling your home with your realtor over the telephone, and she instructs you to look at a certain web page of an interesting property for sale. Months later, you may not remember any good search terms to find this page ("home" or "real estate" will likely produce too many hits to be useful). But you do remember who your real estate agent is, and MyLifeBits will have a log of telephone calls, with caller-ID used to automatically associate the calls with your real estate agent. It then gives you the ability to see what happened during a call, which will reveal the web page visited (including a copy MyLifeBits has made, in case it is no longer on the Web, or you are currently offline). Note that finding the desired web page involved using the relationship between the person and the phone call, and then being able to quickly find time-related events. Similarly, it may be an event on your calendar that you recall. A quick recall of all photos taken during the event helps find the photo.

There are three important features in MyLifeBits to support these operations. The first is quick lookup by time. The MyLifeBits database has indices for time that enable very rapid queries to find overlaps in time, or occurrences close in time. It is also important that a log of events is kept to help find useful overlaps, including documents being opened, telephone conversations beginning and ending, play/pause events in movie playback, etc.

The second important feature is support for typed links. MyLifeBits has entities such as documents, photos, contacts and calendar events. It is possible to create typed links between these entities. For example, a "photo of event" link may be created between a photo and an event. Our photo wizard allows such links to be created automatically between photos and events when the time overlaps. The explicit link is needed for the same reason that photos need explicit location values: photos from third parties taken at the same time as your calendar events should not be interpreted as being photos of your events. Other examples are "person in photo" (between a contact and a photo) or "attendee" (between a contact and an event).

The third feature is the ability to quickly cluster and refine search results. Suppose, for example, that your memory is rather vague: you want to find that interesting email by a person whose name you don't recall. You recall that it was from a long thread with many contributors, and you recall the subject, but don't recall any unique words from the particular email. You do remember that the sender was not a regular contributor to the thread. A search for the subject string of the email thread may return, say, two hundred email messages. MyLifeBits can quickly cluster these emails by sender, and by eliminating the most common senders, you might end up with a short enough list to scan and then recognize the name. Or, you can use MyLifeBits to cluster the search results by time. The time clusters may help you recall when the email was sent. Naturally, the same techniques apply to cluster photos by time taken, or who is in them, or who sent them to you in an email, or in the case of SenseCam, what the temperature was at the time, etc.

[1] The WWMX team already has a publicly available tool to create such travelogues at wwmx.org.

SenseCam is not the only source of passive capture for MyLifeBits. There is also automatic recording of web pages, instant messenger logs, telephone calls, and usage of computer media. The impact of such passive capture is just as important as SenseCam is for photo/sensor capture. It may seem like just a novelty to record web pages, but in fact it quickly becomes indispensable and behavior changing. MyLifeBits users come to rely on the fact that any document they retrieved from the World Wide Web or their corporate web will be on their notebook while they are flying cross-country. They visit web pages just to have a copy in their lifetime store, without a firm plan for reading them, but wanting a copy "just in case." Likewise, knowing that other things are logged relieves the user of worrying about keeping track of details that they know can be easily found later in MyLifeBits.

An additional example of correlating sensor data with media in MyLifeBits comes from TechFest, a sort of science fair that Microsoft Research puts on for Microsoft employees and members of the press annually. At TechFest, employees may swipe their badge on a reader at a demo booth to indicate interest. At the MyLifeBits booth, photos and video are passively captured. When a visitor returns the following year, it is possible to search on their email name, look up the badge swipe, and cross-reference the time to find photos and video of the visitor.

4. DISCUSSION

4.1 SenseCam Issues

Thus far, we have built two prototype SenseCams as described above and these have been used by a few users to capture test data sets for evaluation within the MyLifeBits framework. We are interested in the issues of wider deployment of SenseCam, and so the next phase of the project is to build several more thoroughly engineered devices and to conduct formal user trials with them on a range of subjects. This work is now underway. The trials will encompass both the use of the device itself and the utility of the various MyLifeBits mechanisms described for searching and browsing the images and data. Specific application areas for investigation include tourism and support for memory-impaired patients.

The next version of SenseCam will include audio capture, and will trigger image capture based on audio events. Eventually we would also like to record audio clips surrounding image capture events.

One unexpected success of SenseCam was its capability to capture handwritten notes, both on paper and on the whiteboard. Even at its current modest resolution, SenseCam images of written text and diagrams are often readable, and this feature proved useful for recalling the content of meetings, for instance. With the next version of SenseCam, we intend to explore this application domain further.

In the longer term, we intend to develop battery management techniques to allow GPS, audio, and other additional sensors to be put onboard the device. We are also interested in adding physiological sensors (heart rate, skin resistance, respiration) to complement the environmental ones.

Table 3 – Some examples of MyLifeBits link types

Link type	Source	Target
Annotates	Document, audio, image	Any
Author of	Person	Any
Capture of	Person, Event	Image, video, audio, phone-call
Attendee	Contact	Event
Organizer	Person	Event

4.2 MyLifeBits Issues

For MyLifeBits to adequately handle the output of SenseCam and other passive capture, two requirements are key: (1) a data schema that can express the appropriate relationships, and (2) quick lookup of items by common or similar values.

Of course, any logical data model will have entities and relationships. However, MyLifeBits must deal with special entities and relationships that form part of the user's mental model. For example, entities that might appear as results to a search query, in contrast with obscure system values.

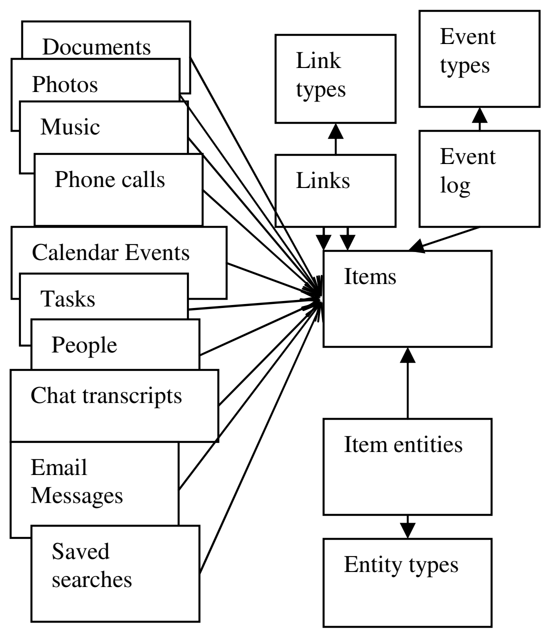

Figure 6 – Key elements of the MyLifeBits schema. Each box is a database table. Arrows indicate foreign keys.

A number of user-level entities have been created in MyLifeBits including documents, photos, phone calls, calendar events, people, and email messages. We call the base type for a user-level entity *item.* Every item (and hence entity) has a universal unique identifier (UUID), and no other fields need be unique (in particular, display name need not be unique, in contrast to a file system). The UUID and some other common fields are stored in the Items table. For all other fields particular to a given entity type, there is a specific table (e.g., the Phone Calls table).

User-level relationships are called *links.* Links in MyLifeBits relate exactly two items (a source and target), and have a type.

Each link type has a name, and allowable entity types for the two entities being related. Table 3 lists some of the MyLifeBits link types.

Figure 6 illustrates the core of the MyLifeBits schema, as implemented in SQL Server. At the bottom the "Entity types" table, which has a row for each valid entity type, and for each there is a corresponding table (shown at the left of the diagram). Each row in the "Links" table indicates the link type (from the "Link Types" table) and two UUIDs of the items that are linked. The "Event log" table records the time of cyber-events performed on the items, for example, when a file is opened.

This schema is sufficient to express the relationships described in section 3: a photo may be linked to a calendar event with a "capture of" link type[2], a photo may be linked to a person with a "capture of" link, or a person may be linked to an event with a "attendee" link.

We are still experimenting with how to best handle sensor and GPS values in our schema. As items, one can select, for example, a list of GPS points and annotate, which may be valuable. On the other hand, they will inflate the size of the items table, which will degrade performance. They will also potentially appear in search results, potentially swamping other values. It is hard to imagine good values for display name, thumbnail or icon when they are listed as search results. On the other hand, if they are not user-level entities, only special UI elements will know about them and special case code will need to be written to handle searching over their values. There is no clear answer to this, and it may be that we settle on some middle road, where they are second class items, not treated as system data, but also not elevated to the same status as an image or document.

In order to perform quick lookup of entities by common or similar values, MyLifeBits relies on database indices. When there is a large result set for such a lookup, MyLifeBits is designed to quickly load the first page of results with minimal overhead; it would be fatal to wait to retrieve all results before displaying the first page to the user with the size of a lifetime store. Because this operation is so common, we have created a paging library that any visualization can call to efficiently retrieve just the results needed for their current page of display.

In order to demonstrate the performance of MyLifeBits for SenseCam results, we replicated one day's worth of SenseCam data across a year. The result was 318,000 SenseCam samples and over 55,000 SenseCam images (81,000 total images). We then performed some of the operations mentioned in Section 3 as important to supporting SenseCam. For each experiment, we first flushed the database buffers so that nothing would be cached ("cold") and then repeated the operation with cached values ("hot"). The time measured includes: (1) issuing the query and getting a list of UUIDs that match (2) sorting the results by display name (the default sort order for the results window pane) (3) retrieving the entity types that apply to the resulting items (4) retrieving all the attribute values for each item in the first page of results. The results are shown in Table 4.

[2] The name of the link is important: some other photo may be linked with a "supporting materials" link, being a photo that was presented at the event rather than one that was taken at the event.

Table 4 – MyLifeBits performance results. Time in seconds. Cold time is with database buffers pre-flushed, hot allows indices to be cached. Experimental system is Dual Xeon 2.8 GHz with 1.5 GB of RAM. Experimental database of size 19 GB with 640,000 items; 40,000 files of total size 133 GB indexed.

Query	Hits	Time (cold)	Time (hot)
Time overlap with badge swipe	3	1.9	0.2
Time overlap with video clip	213	6.5	0.5
Photos in San Francisco region	344	7.2	0.7
One day of SenseCam photos	153	9.7	1.0
Items related to calendar event	346	5.6	1.4
Photos in North America region	2325	12.4	1.9
Photos taken in temperature range	4745	8.5	1.9
All SenseCam photos	55756	16.2	9.1

The hot performance is, not surprisingly, much better than the cold performance, and MyLifeBits users have come to expect that the first few operations after booting a PC will be slow, and then the speed will improve markedly. Clearly, large RAM to increase the database cache size is an important hardware requirement.

The worst performance, cold or hot, is from the query for all SenseCam photos. A breakdown of the time spent in this query is given in Table 5. The dominant cost is actually in sorting and retrieving the entity types, not in the initial query itself, which explains why the query with the most hits has the worst response time. Databases are designed to be fast, and we can see that any given operation is fast, but the combination of operations required are a result of choices made in the MyLifeBits schema, software library, and UI. While we are satisfied with MyLifeBits performance now, it is clear that more work will be required to scale up as we collect more years of data, and likely at a higher rate per year.

Table 5 – Breakdown of time for all SenseCam photos query (times in seconds).

Operation	Time (cold)	Time (hot)
Query	4.5	2.1
Sort	7.6	4.8
Retrieve entity types	3.0	1.6
First page of results	1.1	0.6

For more details on the MyLifeBits schema, and techniques it supports for managing a lifetime store, see [3].

5. RELATED WORK

Wearable digital systems performing passive image capture date at least back to work at MIT in the 1980's. A 1996 version included sensors to detect force and velocity in the shoes, as well as heart rate, respiration, and skin resistance [11]. This line of research places a greater emphasis on capture and image processing rather than storage and retrieval, making a simple “image stack” in time order and retrieval by location sufficient to their needs.

Hori and Aizawa et al developed a wearable system that continuously captures video, along with sensors that include GPS, gyroscope, accelerometer, and a brain wave sensor that has produced promising results for indicating “interesting” scenes [8]. While investigating the same area as us, including associative retrieval, this project is clearly aimed at further out scenarios in hardware, requiring the user to wear a backpack with a notebook computer in it. SenseCam is also distinguished in what it does *not* record, e.g. blurry images.

The pervasive computing community has a natural interest in passive capture, usually in the form of surveillance. In-room capture can also be augmented with robotic capture [6]. However, the focus is on capture and event detection rather than long term storage and management.

StartleCam [7] is a sensor-triggered wearable camera, but is different in significant ways to SenseCam. It uses only one sensor, skin conductivity, and is much more invasive to wear than SenseCam, comprising a chest-worn camera, a separate computer carried in a shoulder bag, and multiple conductivity sensors worn on the fingers and connected to the computer. The idea is to detect the startle response in the wearer, and save the recently captured images which will presumably be of events which aroused the user’s attention. As mentioned above, we intend to add physiological sensors to SenseCam in order to pursue similar ideas in the context of a more readily worn device.

Other projects seek to provide more expressiveness for a personal store than traditional files systems. For example, “Placeless docs” supports links and filing multiple locations [2]. The Haystack project [9] uses RDF to build an ontology for one’s personal storage. PhotoFinder reaffirms the importance of metadata and annotations for photos and attacks the UI problem [10]. Shoebox manages photos, and supports searchable audio annotation as well as visual similarity algorithms [13].

There are many social and legal issues arising from image and sensor capture that are just starting to be explored. While privacy is a key concern, Steve Mann suggests there is also an empowering element to “inverse surveillance” or “shooting back” [12].

6. CONCLUSION

Vannevar Bush, whose visionary 1945 article “As We May Think” has served as a manifesto for MyLifeBits and other related projects, said “The difficulty seems to be, not so much that we publish unduly … but rather that publication has been extended far beyond our present ability to make real use of the record.” In the case of passive capture, including SenseCam, one might say the difficulty is not that we capture too much, but rather that the volume of captured material might extend beyond our ability to make real use of the record.

We believe there are two keys to making use of the record. The first is using relationships between items and the power of databases to quickly sort, cluster and cross-reference in order to perform retrieval and browsing. We have described a number of ways in which MyLifeBits supports these operations. The second is using recorded information to home in on interesting items, and suppress uninteresting or poor items in the record, perhaps to ensure that they are not even recorded in the first place. SenseCam takes this approach when it uses IR sensors to take pictures when people appear, and motion detection to avoid taking blurry pictures.

It is difficult to validate our approach to passive capture. Simple and short user tests are inadequate because until one has a large volume of truly personal data the benefits and shortcomings cannot be evaluated. We have only three full time users at present, but hope to add more so as to gain real world results from our system. However, our early experience with SenseCam and MyLifeBits has proved very promising. Our users are excited about the system, and the most common question after a demo is “When can I get this?”

Passive capture clearly enables scenarios that people are excited about such as event capture, story-telling, and memory assistance. With the myriad of sensors and cyber-events that may be recorded, we believe that the whole will be greater than the sum of the parts by allowing us to make real use of both passively recorded information, and intentionally recorded information.

7. ACKNOWLEDGEMENTS

Ron Logan and Kentaro Toyama contributed the WWMX mapping software for MyLifeBits. Jim Gray has provided many helpful suggestions.

8. REFERENCES

[1] Bush, Vanneva,. As We May Think, *The Atlantic Monthly*, 176(1), July 1945, 101-108.

[2] Dourish, Paul, Edwards, Keith, LaMarca, Anthony, Lamping, John, Petersen, Karin, Salisbury, Michael, Terry Douglas and Thornton, Jim, *Extending Document Management Systems with User-Specific Active Properties*, ACM TOIS, 18(2), 2000, pp. 140-170.

[3] Gemmell, Jim, Bell, Gordon, Lueder, Roger, Drucker, Steven, and Wong, Curtis, MyLifeBits: Fulfilling the Memex Vision, *Proceedings of ACM Multimedia '02*, December 1-6, 2002, Juan-les-Pins, France, pp. 235-238.

[4] Gemmell, Jim, Lueder, Roger, and Bell, Gordon, Living With a Lifetime Store, *ATR Workshop on Ubiquitous Experience Media*, Sept. 9-10, 2003, Keihanna Science City, Kyoto, Japan, pp.69-76.

[5] Gemmell, Jim, Lueder, Roger, and Bell, Gordon, The MyLifeBits Lifetime Store (demo description), *ACM SIGMM 2003 Workshop on Experiential Telepresence (ETP 2003)*, November 7, 2003, Berkeley, CA.

[6] Hagita, N., Kogure, K., Mase, K., Sumi, Y., Collaborative capturing of experiences with ubiquitous sensors and communication robots, *Robotics and Automation*, 2003. (Proceedings of ICRA '03)., Volume: 3 , 14-19 Sept. 2003 pp. 4166 – 4171.

[7] Healey, J. and Picard, R.W., StartleCam: A Cybernetic Wearable Camera. *Proceedings of the Second International Symposium on Wearable Computing*, Pittsburgh, PA, October 19-20, (1998).

[8] Hori, Tetsuro and Aizawa, Kiyohara, Context-based video retrieval system for the life-log applications, *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, Berkeley, California, USA, 2003.

[9] Huynh, David, Karger, David, and Quan, Dennis, Haystack: A Platform for Creating, Organizing and Visualizing Information Using RDF, *Semantic Web Workshop*, 2002.

[10] Kang, Hyunmo and Schneiderman, Ben, Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder, *Proc. IEEE International Conference on Multimedia and Expo*, 2000, pp. 1539-1542.

[11] Mann, Steve, "Smart Clothing": Wearable Multimedia Computing and Personal Imaging, *Proceedings of ACM Multimedia '96*, Boston, Mass., USA, Nov. 1996, pp. 163-174.

[12] Mann, Steve with Niedzviecki, Hal, *Cyborg: Digital Destiny and Human Possibility in the Age of the Wearable Computer*, Doubleday Canada, 2002.

[13] Rodden, Kerry and Wood, Kenneth R., How do people manage their digital photographs? *Conference on Human Factors in Computing Systems (CHI 2003)*, April 2003, pp 409-416.

[14] Spence, Robert, Rapid, serial and visual: a presentation technique with potential, *Information Visualization*, 1:1 (March 2002) pp 13-19.

[15] Toyama, Kentaro, Logan, Ron, Roseway, Asta, and Anandan, P., Geographic location tags on digital images, *Proceedings of ACM Multimedia '03*, November 2003, Berkeley, CA, USA, pp. 156–166.