# Content Vs. Context For Multimedia Semantics: The Case of SenseCam Image Structuring
## — Invited Keynote Paper —

Alan F. Smeaton

Adaptive Information Cluster
& Centre For Digital Video Processing,
Dublin City University,
Ireland.
Alan.Smeaton@dcu.ie

**Abstract.** Much of the current work on determining multimedia semantics from multimedia artifacts is based around using either context, or using content. When leveraged thoroughly these can independently provide content description which is used in building content-based applications. However, there are few cases where multimedia semantics are determined based on an *integrated* analysis of content and context. In this keynote talk we present one such example system in which we use an integrated combination of the two to automatically structure large collections of images taken by a SenseCam, a device from Microsoft Research which passively records a person's daily activities. This paper describes the post-processing we perform on SenseCam images in order to present a structured, organised visualisation of the highlights of each of the wearer's days.

## 1 Introduction

When we think of multimedia information retrieval and multimedia semantics we tend to think of fairly standard multimedia artifacts such as still images, music, video and maybe 3D objects. When we think of how to determine the semantic content of such multimedia artifacts we do so because we want to perform a variety of content-based operations on such information including browsing, searching, summarisation, linking, etc. And finally, when we look at *how* we might determine semantics of multimedia objects we find that there are generally two approaches, namely:

1. use the context of the objects such as information gathered at the time of object creation or capture, to help determine some content features;
2. extract information directly from within the content of the objects in order to determine some content aspects.

Trying to determine and then usefully use a user's *context* is a fairly hot topic in information retrieval at the moment with lots of attempts to capture and then apply such context in retrieval [11]. Determining a document or a multimedia object's context has also been explored for a long time and this forms the basis for many current systems for multimedia object management. For example such basic metadata as date and time of creation form the essential content representation for many tools which manage personal photos. Examples of such popular photoware includes Photoshop Album [2], PhotoFinder [16], ACDSee [1], Picasa [9] and others. Other metadata created at the time of photo capture such information as shutter speed and lens aperture, whether a flash was used or not can also be used to support automatic grouping of photos [14]. Finally, there are emerging online photoware systems such as Flickr [5] and Yahoo 360 [19] which support user-supplied context information to help with photo organisation.

What all these applications have in common, apart from the fact that they are all used to manage personal photos, is that they all use semantic information to describe multimedia objects (photos) which are derived from the context of the photo ... either directly from the capture process, or provided by an end-user afterwards.

To complement semantics derived from context we also use semantics derived from content in helping to manage our multimedia objects. Returning to the example of personal photos, this corresponds to extracting features directly from the image contents. An example system which does this is MediAssist which automatically determines whether a picture was taken indoors or outdoors, whether it is of a built or of a natural environment, whether a picture contains faces and if so whether those faces are faces of known individuals [14]. While this is a limited set of descriptive semantics, automatically determining the presence or absence of a larger number of medium and high level semantic features in visual media is notoriously difficult as is shown repeatedly in the TRECVid benchmarking evaluation campaign [17].

Once we have determined some level of semantic representation for multimedia objects we can then use these for content-based operations such as retrieval and we find that those derived from content and from context are almost always used either independently of each other or collaboratively with each other, but rarely are they truly integrated with each other. In other words, because these semantics are derived from different primary sources they maintain and retain their differing heritages when they are used subsequently.

To illustrate this let us examine the different ways in which video shots can be retrieved. In [18] we presented a classification of five different experimental approaches to video shot retrieval, namely:

1. Use metadata determined at the time of video capture/creation to access video by date, time, title, genre, actors, popularity rating, etc. as in [12];
2. Use one or more example query images to match against shot keyframes using whole-image matching approaches based on colour, texture or edges, as shown by many systems in [17];
3. Use text queries to match against text derived from transcriptions of the spoken dialogue of text determined from video OCR, also as shown by many systems in [15];

4. Use video objects, semi-automatically determined from shot keyframes and from user query images, and match these video objects based on shape, colour and/or texture, also as shown by many systems in [17];
5. Use the presence or absence of semantic video features such as indoor, outdoor, beach, sky, boats, motor vehicles, certain named persons, etc. to narrow the scope of shot retrieval to only those shots likely to contain such features;

Many systems have been developed to support video shot retrieval using one, two or perhaps three of the above but none have been developed to support all of them and for those that support multiple modalities for shot retrieval, the user is normally left with responsibility for combining and integrating them.

In this paper we argue for a more integrated approach to using semantic features determined from content and from context, and we illustrate what is possible with a novel application based around sets of images taken with a SenseCam. In the next section we introduce the SenseCam and its possible range of applications and in section 3 we present a summary of our work on structuring SenseCam images based on an integrated combination of content and context features. Section 4 concludes the paper.

## 2 The SenseCam

A SenseCam is a device developed by Microsoft Research in Cambridge, UK, for recording visual images of a wearer's day. It passively captures images through a fisheye lens and stores them on-board for subsequent download to a personal computer [8]. In addition to being a camera, a SenseCam also has other sensors including a light meter, a passive infra-red sensor and a 3-axis accelerometer and sensor readings from all these devices are also stored for later download. However, in addition to recording some elements of the SenseCam environment, the additional sensors are also used in a semi-intelligent way to trigger when photos are to be taken. For example when a person walks in front of the wearer this can be picked up by the passive infra-red sensor to trigger a photo to be taken. Similarly, when the user moves by standing up, or moves from indoor to outdoor or vice-versa, these are picked up by the accelerometer and light level sensors respectively and also trigger taking of photos. As a default, without an explicit triggering from the sensors, or from a user-controlled button on the SenseCam, the device will take a new photo every 45 seconds anyway. In this way a typical day can have up to 3,000 photos taken, which could add up to almost a million images in a year. A SenseCam being worn around a wearer's neck is shown in Figure 1 and a set of sample images taken from the author's use of a SenseCam is shown in Table 2.

The SenseCam device has been used extensively in the MyLifeBits project at Microsoft Research [6], [7] as well as being used in other, exploratory projects at Microsoft Research in Cambridge [10]. Like many other sensor devices, the SenseCam is great at capturing raw data, up to a million images per year for each user, and the main challenge is to effectively manage this huge volume of personal data. This requires automatic analysis and structuring in order to

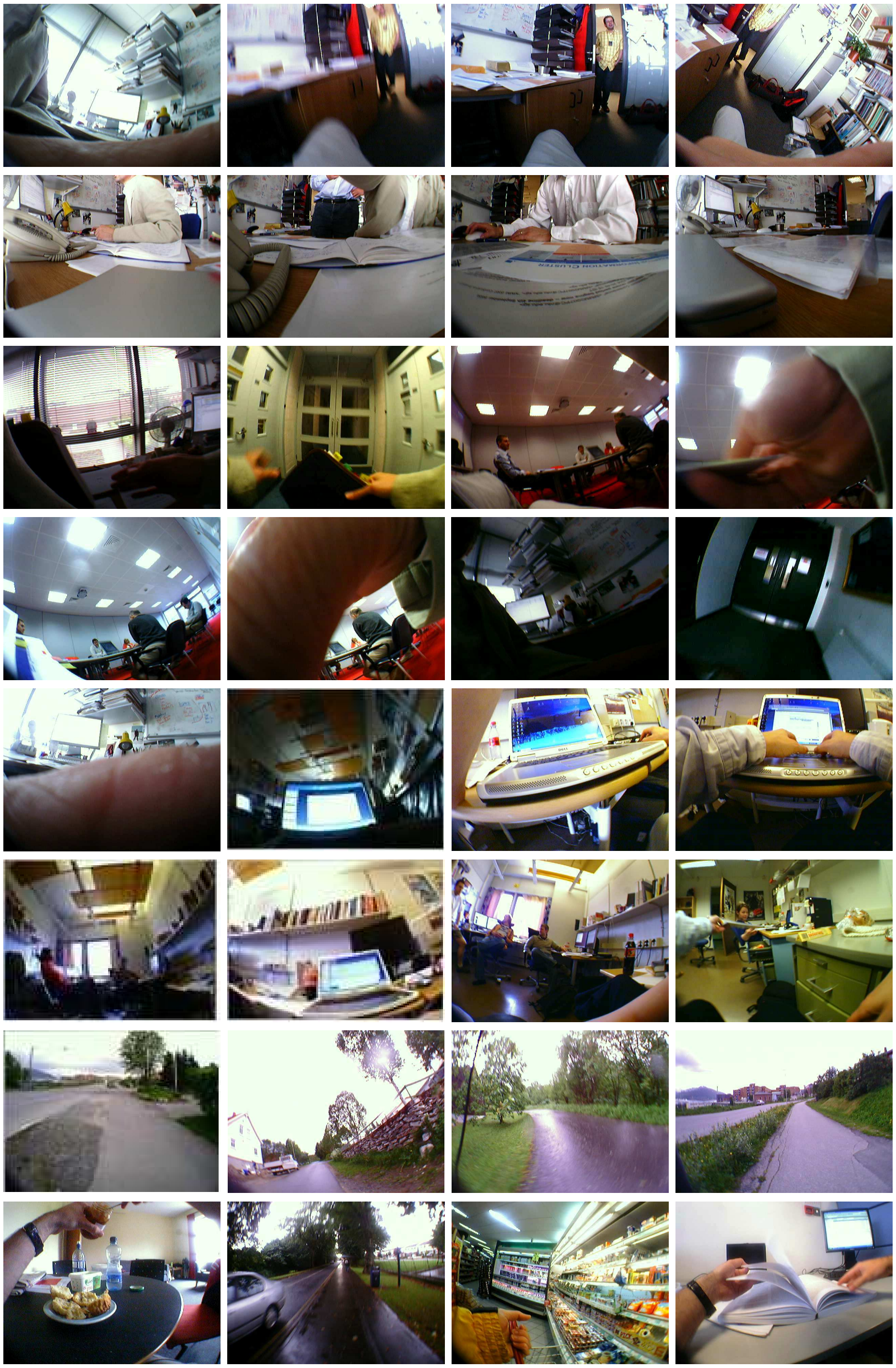


**Table 1.** Sample SenseCam images

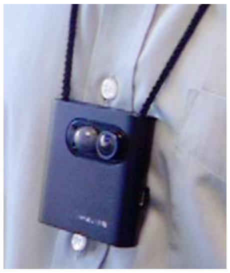


**Fig. 1.** A SenseCam worn around the neck

impose some organisation on the raw images and this is the challenge we address as we seek to determine semantics for these multimedia objects and to use both context (date, time, sensor readings) and content (image processing) to achieve this. In the next section we describe how we do this.

## 3 Structuring SenseCam Images

Effectively managing a growing collection of up to 3,000 images taken per day is physically impossible unless the images are structured in some way. Within our daily lives, our activities can be broken down into "events" corresponding to things like having breakfast, walking to the bus stop, travelling to work on the bus, walking to our workplace, making coffee as soon as we arrive at work, sitting at our desk reading email, drinking coffee and starting to write a report, breaking to have a short meeting with colleagues, going to the canteen to have a morning coffee break, returning to work at the desk, having lunch with a group of friends, back to the desk in the afternoon and finishing work with a one-on-one meeting with the boss, getting the bus to the gym, having a workout there, going to a movie, taking the bus home, making and eating dinner, watching TV, and finally going to bed.

While we could argue about the definition of an event, whether travel to-from work is one event or divided into walking to the bus stop, travelling on the bus and waking to work which are each events, in general we can say that each of the above is characterised by being visually different form the preceeding and succeeding events. What the user (and SenseCam wearer) sees will be different for each event because the location will change or the people present will change. In theory, such changes in location are detectable through processing the sets of images taken during each event. In a way this is analogous to the task of shot boundary detection in video where we also wish to find the boundary between different shots by comparing images, but in the case of SenseCam event segmentation the task is more difficult because adjacent images may be quite different from each other but still part of the same event. These image differences will be caused by the user turning around towards/away from a window or light source or facing in a different direction, looking at different people, or a different part of the same room. However the *set* of images constituting an event will be globally similar to each other. In contrast, adjacent images in video will only have small differences, unless there is a photo flash or some very rapid camera and/or object movement.

In work reported elsewhere we have addressed the problem of event segmentation by comparing temporally adjacent and temporally nearby SenseCam images using conventional low-level image features like colour and texture [3], as well as spatiograms [4], and our results on this to date indicate that using image processing techniques alone we can achieve useful results. When we then incorporate evidence for event boundaries taken from other SenseCam sensor readings and even from detection of local Bluetooth devices such as people's mobile phones [13] then the reliability of event detection improves further.

In our work to date we have found that SenseCam "events" can contain anything from some tens of images to several hundred, depending on the activity taking place as well as the duration of the event. Once events have been detected then we can then further structure a user's SenseCam images by manipulating and reasoning about events themselves. A schematic overview of how we process SenseCam images is shown in Figure 2.

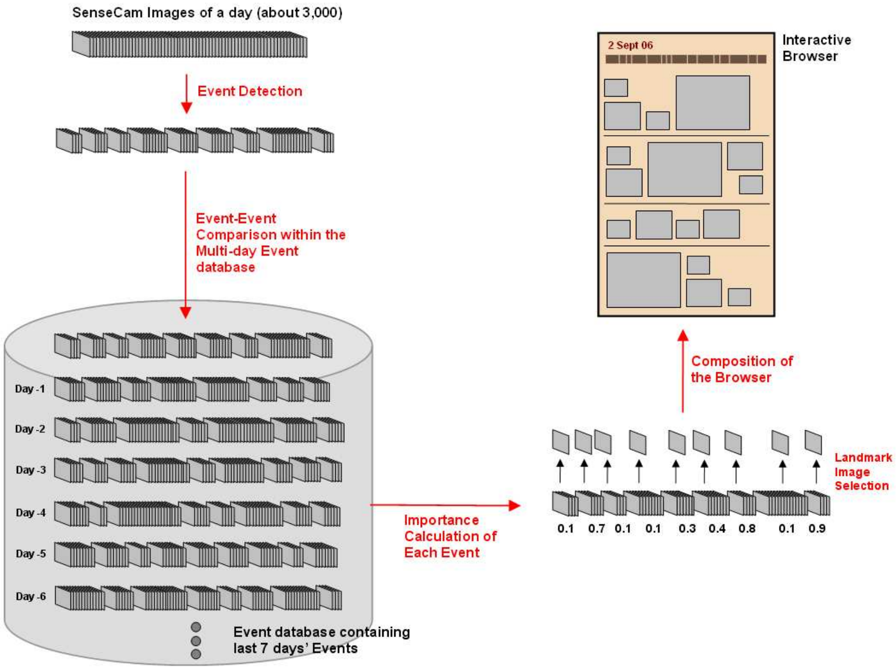

SenseCam Images of a day (about 3,000)
Event Detection
Event-Event Comparison within the Multi-day Event database
Day -1
Day -2
Day -3
Day -4
Day -5
Day -6
Event database containing last 7 days' Events
Importance Calculation of Each Event
0.1
0.7
0.1
0.1
0.3
0.4
0.8
0.1
0.9
Landmark Image Selection
Composition of the Browser
2 Sept 06
Interactive Browser

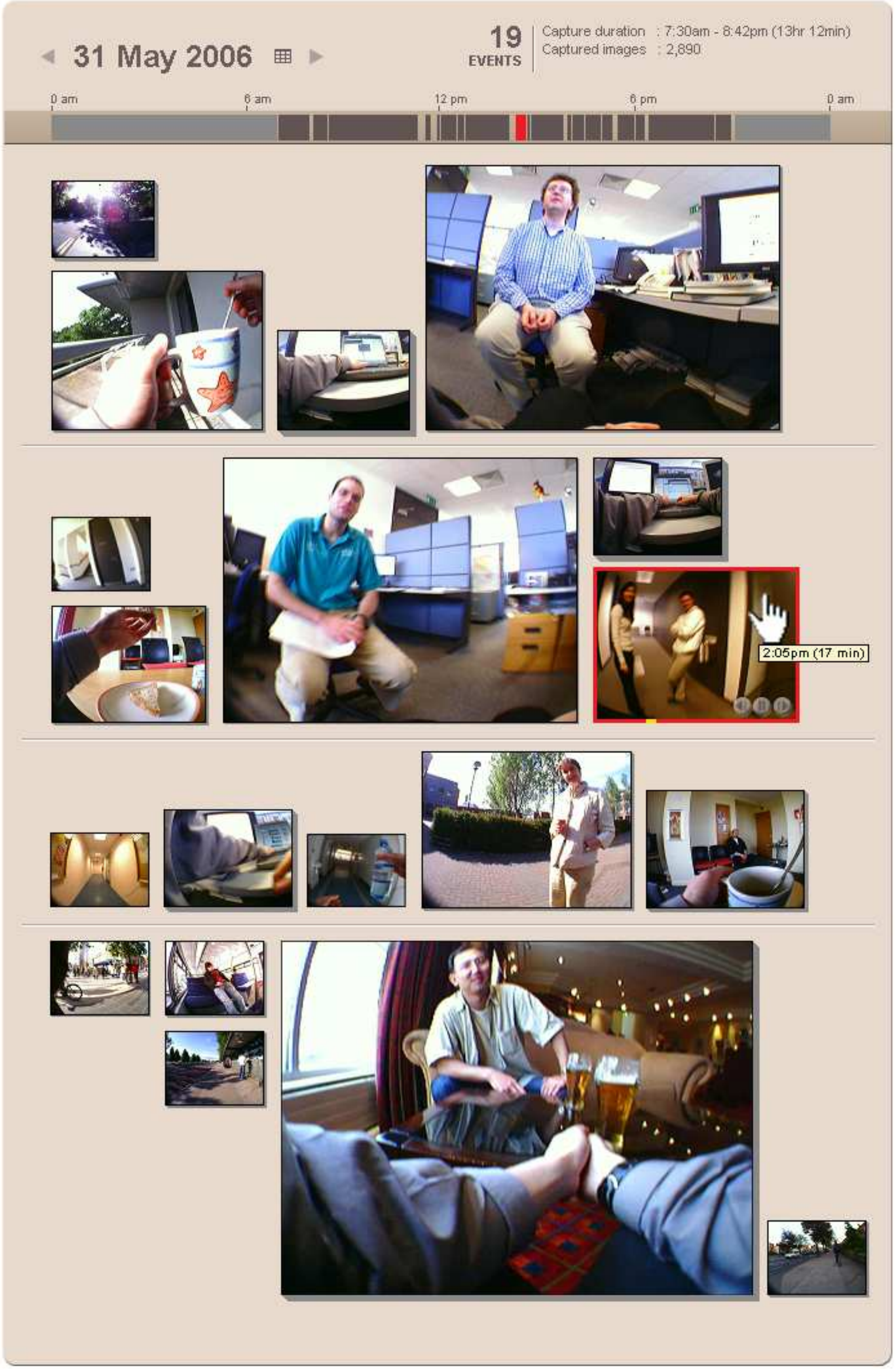
**Fig. 2.** Schematic for processing SenseCam images

In order to manipulate SenseCam events we need some representation for the event itself so we compute a virtual image as the average of all the SenseCam images within an event. This is a crude first approximation and could be refined by detecting outlier SenseCam images within an event and removing them, or removing or down-weighting SenseCam images towards the beginning and end of detected events as they are more likely to be close to event transitions, which will probably involve the user moving location and this will generate SenseCam images which are not really part of the preceeding or succeeding events. However investigating this aspect is part of our future work. In fact to reduce processing time the event representative is generated as part of the event detection process, so there is little overhead in computing this. Once the virtual representative image from an event is computed we then locate the actual SenseCam image which is visually closest to the virtual centroid and we term that a "landmark" image. The reason for doing this is to use an actual SenseCam image as a representative for presentation of the event. In future work we would like this to be the SenseCam image which has the greatest number of faces present, but for now we base our landmark detection on selection of the image most similar to the virtual average of those in the event.

When a day's SenseCam images are uploaded and the virtual representative for each detected event is available we then add it to a database of event representations. Our task now is to determine which of the day's events are more important than the others. For example, having breakfast, travelling to/from work, having coffee with the same colleagues and working at the same desk are all regular events which happen daily and are not very different from one day to the next, even visually, yet going to the movies, visiting the gym or having lunch in a different restaurant or with different people will all be unusual events for this wearer's lifestyle.

We determine an event's importance by comparing the visual representatives for each event over a fixed 7-day window and examining an event's duration. Basically, if an event is unusual in terms of a given week's activities it will not appear to have any visually similar events or a similar duration and it will then be assigned a high importance or novelty rating. On the other hand if an event is one of a series of regular and repeating events during that 7-day period it will have many similar events, both visually and perhaps in terms of duration also. This is quite an heuristic step and could be refined by considering the time of day for example, but as with landmark detection, using these event features is sufficient for now and a possible topic for future work.

Finally, once a day's SenseCam images have been segmented into events with landmark images and importance ratings determined automatically we can present the day's activities in the browser shown in Figure 3. This browser configuration lays out landmark images from the most important or highly novel events from each day with the size of the landmark image being indicative of the importance rating of the event. In this way the unusual activities are highlighted by being bigger yet the complete set of a day's activities are shown. In Figure 3 we can see that the most unusual events for that day – 31 May 2006 – appear to



31 May 2006

19 EVENTS

Capture duration : 7:30am - 8:42pm (13hr 12min)

Captured images : 2,890

**Fig. 3.** Interface for reviewing a single day's SenseCam images

be the wearer drinking beer with a friend (2nd last landmark image on bottom row) and having meetings with 2 different colleagues as shown in Rows 1 and 2. A timeline bar on the top of the browser indicates the ranges of times during the day when the SenseCam was recording images and also indicates the sizes and relative durations of segmented events. When the user mouses over an event landmark, all the images from that landmark play within the frame of that landmark like a video playback.

Using this browser the wearer can get a complete picture of the day's activities with the set of 2,890 images in the case of Figure 3 being easily navigable because of the way they are structured.

In summary, the processing on a single day's SenseCam images proceeds as follows:

1. Segment the set of images into events using low-level image features and spatiograms, combined with temporal ordering of the images;
2. Generate a virtual event representative as the average of all images in the event;
3. Identify the landmark image for each event as the SenseCam image most visually close to the virtual event representative;
4. Assign each event an importance or novelty rating based on comparing the visual representatives for each event over a fixed 7-day window and examining an event's duration;

On closer examination of the different steps in this process we can see that almost all involve operating on content description derived from both content, and context without any differentiation as to whither the source of that content description. So in this example we make no distinction between content and context in deriving content description and this integration of the sources is to everyone's advantage.

## 4 Conclusions

In this paper we have examined the sources of information from which multimedia semantics can be derived and categorised them into either content-based or context-based. We have also argued for a more *integrated* approach to determining multimedia semantics where the heritage or origin of the information, whether derived from content or from context, is ignored. To illustrate this we have presented an overview of a complex tool we have developed which ingests a set of several thousands of SenseCam images per day, as a summary of the wearer's daily activities. The interesting aspect of this tool, and the analysis of information gathered by the wearer of the SenseCam, is that the analysis is performed on a combination of context and content based information, with no distinction made between the two sources. This, we believe, is a model of where multimedia semantics should be derived for other applications.

## Acknowledgements

This work is partly supported by Science Foundation Ireland under grant number 03/IN.3/I361 with support from Microsoft Research. The work described here is the result of the collaborative research activities of the author and his colleagues, postdoctoral researchers and students.

## References

1. ACDSee. Available at http://www.acdsee-guide.com/ (last visited september 2006).
2. Adobe Photoshop Album. Available at http://www.adobe.com/products/-photoshopalbum/ (last visited September 2006).
3. M. Blighe, H. Le Borgne, N. E. O'Connor, A. F. Smeaton, and G. J. F. Jones. Exploiting context information to aid landmark detection in SenseCam images. In *2nd International Workshop on Exploiting Context Histories in Smart Environments (ECHISE)*, Irvine, Calif., USA, September 2006.
4. C. Ó. Conaire, N. E. O'Connor, A. F. Smeaton, and G. J. F. Jones. Organising a Daily Visual Diary Using Multi-Feature Clustering, 2006. submitted for publication.
5. Flickr. Available at http://www.flickr.com/ (last visited September 2006).
6. J. Gemmell, A. Aris, and R. Lueder. Telling Stories with MyLifeBits. In *ICME '05: IEEE International Conference on Multimedia and Expo*, 2005.
7. J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. MyLifeBites: Fulfilling the Memex vision. In *Proceedings of ACM Multimedia*, December 2002.
8. J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell. Passive capture and ensuing issues for a personal lifetime store. In *CARPE'04: Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences*, pages 48–55, New York, NY, USA, 2004. ACM Press.
9. Google Picasa. Available at http://picasa.google.com/ (last visited september 2006).
10. S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. SenseCam: a Retrospective Memory Aid. In *UBICOMP 2006: The 8th International Conference on Ubiquitous Computing*, September 2006.
11. P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer: the Kluwer International Series on Information Retrieval, 2005.
12. Internet Archive: Moving Image Archive. Available at http://www.archive.org/details/movies (last visited september 2006).
13. B. Lavelle. SenseCam Social Landmark Detection using Bluetooth, 2006. M.Sc. in Software Engineering Practicum Report, Dublin City University.
14. N. O'Hare, H. Lee, S. Cooray, C. Gurrin, G. J. Jones, J. Malobabic, N. E. O'Connor, A. F. Smeaton, and B. Uscilowski. MediAssist: Using content-based analysis and context to manage personal photo collections. In *CIVR2006 - 5th International Conference on Image and Video Retrieval*, 2006.
15. S. Sav, G. J. Jones, H. Lee, N. E. O'Connor, and A. F. Smeaton. *Interactive Experiments in Object-Based Retrieval*, volume 4071 / 2006, pages 1–10. Springer, Berlin/Heidelberg, Germany, 2006.

16. B. Shneiderman, H. Kang, B. Kules, C. Plaisant, A. Rose, and R. Rucheir. A photo history of SIGCHI: evolution of design from personal to public. *interactions*, 9(3):17–23, 2002.
17. A. F. Smeaton. Large scale evaluations of multimedia information retrieval: The TRECVid experience. In W.-K. L. et al., editor, *CIVR 2005 - International Conference on Image and Video Retrieval*, volume LNCS 3569, pages 11–17, Singapore, July 2005. Springer.
18. A. F. Smeaton. Techniques Used and Open Challenges to the Analysis, Indexing and Retrieval of Digital Video. *Information Systems*, 2006. (in press).
19. Yahoo ! 360. Available at http://360.yahoo.com/ (last visited September 2006).