Combining Image Descriptors to Effectively Retrieve Events from Visual Lifelogs

Aiden R. Doherty, Ciarán Ó Conaire, Michael Blighe, Alan F. Smeaton, and Noel E. O'Connor
Centre for Digital Video Processing and CLARITY: Centre for Sensor Web Technologies
Dublin City University, Glasnevin, Dublin 9, Ireland
adoherty@computing.dcu.ie

ABSTRACT

The SenseCam is a wearable camera that passively captures approximately 3,000 images per day, which equates to almost one million images per year. It is used to create a personal visual recording of the wearer's life and generates information which can be helpful as a human memory aid. For such a large amount of visual information to be of any use, it is accepted that it should be structured into "events", of which there are about 8,000 in a wearer's average year. In automatically segmenting SenseCam images into events, it will then be useful for users to locate other events similar to a given event e.g. "what other times was I walking in the park?", "show me other events when I was in a restaurant". On two datasets of 240k and 1.8M images containing topics with a variety of information needs, we evaluate the fusion of MPEG-7, SIFT, and SURF content-based retrieval techniques to address the event search issue. We have found that our proposed fusion approach of MPEG-7 and SURF offers an improvement on using either of those sources or SIFT individually, and we have also shown how a lifelog event is modeled has a large effect on the retrieval performance.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Measurement

Keywords

Lifelogging, image retrieval, SenseCam, fusion

1. INTRODUCTION

Almost everything we do these days is in some way monitored or logged. We've come to accept - or maybe just ignore - this massive surveillance because it brings us benefits. We have a more secure feeling when we know there is CCTV present, we get itemised billing from phone companies, and we get convenience and even loyalty bonuses with some of our regular purchases.

Lifelogging is the term used to describe recording different aspects of your daily life, in digital form, for your own exclusive personal use. It is a form of reverse surveillance, sometimes termed *sous*veillance, referring to us - the subjects - doing the watching, of ourselves. Lifelogging can take many forms, such as the application which runs on your mobile phone to 'log' all your phone activities and then present all your phone-based activities in a calendar format.

For over two years we have been working with a small, wearable camera called the SenseCam [11] developed by Microsoft Research in Cambridge, UK that creates a visual record of the wearer's day. The SenseCam is worn on the front of the body, suspended from around the neck with a lanyard as displayed in Figure 1. It is light and compact, about one quarter the weight of a mobile phone and less than half the size. It has a camera and a range of sensors for monitoring the environment by detecting movement, temperature, light intensity, and the possible presence of other people in front of the device via body heat. The SenseCam regularly takes pictures of whatever is happening in front of the wearer throughout the day, triggered by appropriate sensor readings. Images are stored onboard the device, with an average of almost 3,000 images captured in a typical day, along with associated sensor readings.

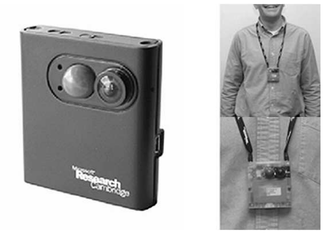


Figure 1: The Microsoft SenseCam

Given that the SenseCam captures 3,000 images in a typical day, we firstly make this information more digestable by seqmenting sequences of images into distinct events/activities e.g. breakfast, meeting, walk in the park, etc.[9, 20] A user will, on average, collect almost 1 million SenseCam images over an entire year which relates to an average of 8,000 events. For a user looking at a particular event (e.g. talking to a friend), they may desire to search for other

MIR'08, October 30–31, 2008, Vancouver, British Columbia, Canada.

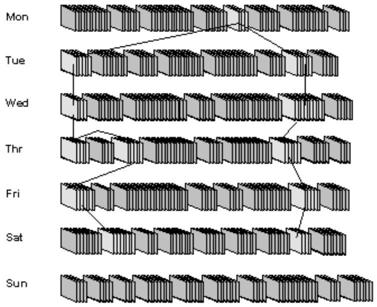


Figure 2: Retrieving other events that are similar to a given lifelog events

similar events and review their past experiences by clicking and viewing on relevant items, which then inspires the user to look at other related events and browse through them. However if a user is capturing approximately 8 thousand events per year, they will soon be overwhelmed by such a volume of information, therefore it will be useful to provide search/retrieval facilities.

Traditionally the lifelogging community have been focused on the minituratisation of capture devices, and also on the problem of storing the very large amounts of data generated by these devices. In this paper we have firstly identified that the community are only now attempting to address the significant challenge of retrieving material from a lifelog that is relevant to the users' information need.

In terms of managing huge volumes of SenseCam events we allow users to filter and browse using the 3 axes of who, where, and when i.e. who was at the event, where was the event, and when was the event. To complement these 3 axes we can also filter lifelog content by using higher level semantic features which has shown great promise for filtering based on concepts other than named people [6]. However whether one's collection of lifelog events is filtered or not, there is a need to either browse all events (which works and is acceptable in some user scenarios, e.g. when the filtering using either of the aforementioned steps has reduced the number of events to something small) or to do searching and event matching. In the case of the latter it is necessary to determine techniques to identify the optimal content-based system to retrieve other relevant events to any given SenseCam event. This case is illustrated in Figure 2 where on a search scenario on the 5^{th} event for the first day, all similar events from other days are linked to it.

To retrieve events similar to a given event in a lifelog it is necessary to firstly determine *how to represent* SenseCam events, and then *how to compare* those event representations against each other. We compare our proposed approach to a selection of other lifelogging retrieval methods. Two datasets are used, one of 239,033 SenseCam images with a groundtruth which also allows the tuning of retrieval parameters, and one of 1,864,149 SenseCam images where users judged retrieved events to a given query event for each approach. We discuss how we address the challenges of retrieval in lifelogs throughout the remainder of this paper.

2. BACKGROUND

Gemmell & Bell [2] and Tancharoen & Aizawa [22] emerge with a common understanding that the biggest challenge facing the lifelogging community is that of efficient retrieval of information that is useful to the user. The work carried out in this paper is solely focused on the effective retrieval of lifelog data that suits the information need of the user.

Extensive research has taken place on the management of personal image collections [21, 13]. However these applications only consider pictures *manually* taken with current digital camera and mobile phone technology. With passively capturing wearable cameras, like the SenseCam device, the scale and frequency of images is much more significant. This presents a different set of challenges to those posed by current personal photo management applications.

One method to review images captured by the lifelogging devices would be to play all the material through at a high speed [11]. However it takes upwards on 2 minutes to quickly play through a day's worth of SenseCam images, which translates to 15 minutes to review all the images from a week. We believe a one page visual summary of a day containing different images representing encountered activities or events, coupled with the ability to search for events or similar events, provides a much more useful method to manage SenseCam images [14].

Retrieval in the domain of lifelogging has been investigated before, however experiments have generally been on very small datasets confined to the data of one user [24, 15, 22]. The scale of the dataset in our experiments is much greater than that used in lifelogging retrieval experiments carried out by others, and provides a better insight on the challenges facing the community in the future.

In an interesting contrast to the approaches described thus far, Hori & Aizawa actually ignore image content in retrieving related lifelog events because they found it would *probably* not be useful [12]. They retrieve similar events to a given lifelog event using contextual sources only; e.g. brain wave analyser, GPS, accelerometer and gyro sensors. However in experiments detailed later in this thesis we show that content is very important in terms of retrieving relevant lifelog events to given query events captured by SenseCam users.

3. APPROACHES TO EVENT MATCHING

Given that a SenseCam captures an average of 3,000 images per day, we firstly make this more digestable by segmenting sequences of images into distinct semantic events or activities, as detailed in previous work [9]. Having identified those activities, we will now discuss possible approaches towards retrieval of other similar SenseCam events to a given event. The aim is to find similar events to a query event, as displayed in Figure 2. However at the most basic level an event will consist of many images, therefore we also question how to represent events.

3.1 MPEG7+Sensors

There are a number of available MPEG-7 descriptors, and we have selected 4 that are particularly well suited to SenseCam images. The MPEG-7 descriptors we extract are colour layout, colour structure, scalable colour, and edge histogram [4]. In training we investigated which of those sources of information are most useful (when early fused) in comparing images against each other, and found that a concatenation of the edge histogram and scalable colour descriptor performed optimally.

Given that MPEG-7 image values are represented by vector values, we investigated the optimal vector distance com-

Research Variable	Parameter Value
Keyframe selection	middle image
Vector distance	Canberra
Normalisation	Min Max
Fusion	Comb MAX
MPEG7 features	scalable colour, edge histogram
MPEG7 weight	0.293
Accelerometer weight	0.057
Light weight	0.203
PIR weight	0.197
Temperature weight	0.25

Table 1: Best trained "MPEG7Sense" parameters

parison technique to use in the retrieval of lifelog events. We considered the following techniques: *Bray-Curtis*, *Canberra*, *Euclidean*, *Histogram Intersection*, *Kullback-Leiber*, *Jeffrey Modification of K-L*, *Manhattan*, *Square Chi Squared*, *Squared Chord*, and X^2 *Statistics*. After training, the *Canberra* vector distance technique performed optimally.

The SenseCam also has environmental sensors onboard, namely: ambient temperature, passive infrared, light, and motion instruments. Given this information, we also investigated using these sources of information in retrieving events potentially similar to a given event. As these are scalar values, comparison of any two given events on each data source is straightforward. However given that there are now a number of different sources of information available to represent all the events, it is desirable to investigate combining these sources together. This involved investigating various normalisation (*Mean-Shift*, *Min-Max*, *Sum* and *Standard* [18]) and fusion (*CombANZ*, *CombMAX*, *CombMED*, *CombMIN*, *CombMNZ*, *CombSUM* [10]) methods, as well as investigating how much confidence to place on the various sources of information. Table 1 outlines the best trained system after over one thousand parameter variations were evaluated. This approach will be simply referred to as "*MPEG7Sense*" for the remainder of this paper.

3.2 SIFT

SIFT [17] is a method for extracting interest point features from images. It detects interest point locations and also extracts features from around the points that can be used to perform reliable matching between different views of an object or scene. The SIFT features are invariant to image orientation, image scale, and provide robust matching across a substantial range of affine distortions, changes in 3D viewpoint, addition of noise, and changes in illumination. In addition to these properties, they are highly distinctive, relatively easy to extract, allow for correct object identification with low probability of mismatch and are easy to match against a large database of local features. They are also robust to occlusion; as few as three SIFT features from an object are enough to compute its location and pose. In addition to object recognition, the SIFT features can be used for matching, which is useful for tracking and 3D scene reconstruction. Recognition can be performed in close-to-real time for small databases on modern computer hardware. The calculation of the features occurs in a multiphased filtering process that discovers interest points in scale space. Keypoints are generated which account for the local geometric deformations by characterising blurred image gradients in numerous orientation planes and at various scales [16].

For event/keyframe matching, SIFT features are first extracted from a set of reference keyframes and stored in a database. A new keyframe is matched by individually comparing each feature from the new keyframe to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors. In order to match features between keyframes, the distance ratio test was used [5, 3]. To examine whether a point from the 1^{st} keyframe has a match in the 2^{nd}, it's two most similar descriptors in the 2^{nd} keyframe are found. If the ratio of the nearest distance to the second nearest distance is less than 0.7, a match is declared. The number of matches between a keyframe and all other keyframes in the event are summed, and then the average number of matches is calculated. Events are ranked based on the average number of matches calculated, with the most relevant items containing the highest number of matches.

3.3 SURF

Introduced by Bay et al. [1] Speeded Up Robust Features (SURF) are inspired by the SIFT feature approach, but speed up the extraction and description of interest points by exploiting integral images, achieving state-of-the-art performance in feature matching.

While the extraction of SURF features is faster than SIFT, it is still computationally expensive to perform exhaustive matching between a query image and an image collection. Targetting fast image retrieval, Nistér and Stewénius describe an approach to image matching that scales to very large datasets, up to a million images [19]. Following their proposed scheme, we created a hierarchical visual word vocabulary using seven million SURF descriptors extracted from a collection of web images. These descriptors were clustered hierarchically using K-means, to generate a vocabulary tree with 4096 leaf nodes (visual words). An image database was structured by including an inverse file for each visual word, allowing efficient retrieval. Database images were compared to the query image using the L_1 distance measure between their normalised histograms of visual words. This measure performed best in the original work [19]. Unlike our SIFT approach in this paper, which exhaustively compares every image to the query, a vocabulary tree query can be run in a few seconds for a database of thousands of images.

To further improve performance, we rerank the top 20 results by counting bi-directional matches between the query and each database image. In Lowe's work with SIFT features [17], a match between interest points was determined by using the distance ratio test, with $\alpha = 0.6$, as described in section 3.2. Since this measure is asymmetrical, we also compute the matches in the reverse direction (from the target to the query) and we count the matches that occur in both directions (bi-directional matches). Such matches were found to be very stable and strong indicators of a good match. We optimised the value of α using a training set and found that a value of $\alpha = 0.7$ worked best. This is to be expected, as SenseCam images are generally of a lower visual quality than the images used by Lowe.

3.4 Fused Approaches

As the global colour and edge features of MPEG-7 are complementary to the local SURF features, fusing the results of both approaches has the potential to improve the overall event matching performance. We describe here the two main approaches to fusion we investigated.

Baseline Fusion The bi-directional SURF matches that are used for SURF re-ranking are strong indicators of matching confidence. As a baseline fusion scheme, we simply take the *most confident* SURF results and then append the rank-list provided by the *MPEG7Sense* approach. More formally, we choose the top events ranked by the SURF scheme that have at least T bi-directional matches with the query image, then we insert the *MPEG7Sense* results that have not already occurred in our fused rank-list. We trained the parameter T on a training set of 24 queries and found that $T = 2$ optimised the MAP score.

Score Fusion In this approach we rank results in the following order for a given query based on: 1) events proposed by both *MPEG7Sense* and SURF, 2) events proposed by SURF, and 3) events proposoed by *MPEG7Sense*. Parts 2 & 3 have already been ranked, so they are straightforward to order. However for part 1 the scores from both sources must be normalised and fused. After training we selected the *Min-Max* normalisation technique and the *CombSUM* fusion method. A weighting/confidence of 80% was placed on the SURF source, and 20% on the *MPEG7Sense* source.

3.5 Event Representation

On average each event consists of almost 100 images and there are a number of approaches that we have investigated to model an event, including: selecting the middle image; the image with the highest quality; the image within the event that is closest to the event average or all the other images in the event; the image within the event that is not only representative of that event but also most distinct from all the other events [8]. We also investigated whether it would be useful to more strongly weight those images towards the middle of a given SenseCam event more strongly, on the premise that they are more likely to represent the semantic meaning of events, however after training we found that simply selecting the middle image from an event works best with respect to final retrieval performance. We used this as the basis on which to compare the similarity of events using *MPEG7Sense*, SIFT, and SURF features detailed in the preceeding subsections.

Traditionally a single "keyframe" image has been selected as the event unit of retrieval. However given that SenseCam events can be highly variable in terms of visual and semantic content quite often [8], we investigate constructing a unit event based not just on 1 image, but on a number of images present in the event. We investigate the following two techniques:

1. **Event Average** - Get the average value of all the features across all the images in the event

2. **Middle N** - Select the average value of all the features across the middle n images in the event. Given that sequences of SenseCam images can be segmented into distinct events or activities exceptionally quickly using sensor values [9], we investigate extracting the MPEG-7 image descriptor values only from a selection of images in the middle of each event. The premise of this is that images in the middle of an event are more likely to be representative of the semantic meaning of that event. Also by taking the average of a number of images, the effect of single poor quality or outlier images is less damaging on retrieval performance.

ID	**General** Images	**General** Events	**Specific** Images	**Specific** Events
1	79,595	1,071	1,686,424	19,995
2	76,023	892	92,837	1,182
3	42,700	409	44,173	443
4	40,715	505	40,715	505

Table 2: User Statistics

4. EXPERIMENTAL SETUP

In experiments to investigate the effectiveness of our retrieval approaches, we asked 4 different users to collect SenseCam data. We collected two datasets, one of 239,033 images, and another of 1,864,149 images. The first and smaller dataset was used to construct an extensive groundtruth of relevance judgements. To have a sufficient number of relevant events to train parameters on, it was decided to use more general queries in this dataset e.g. driving, at work on PC, eating, etc. The purpose of the second and larger dataset was to investigate how the performance of the best systems on the smaller dataset, translates to a larger dataset. Also the queries used on the larger dataset were much more specific e.g. "what other times was I talking to John?", "what other times was I on an aeroplane?", etc. We will now describe the experimental set up of firstly the smaller dataset with the relevance judgements, and secondly the very large dataset with the very specific user-generated queries.

4.1 Dataset with Relevance Judgements

In experiments to investigate the effectiveness of our retrieval approaches, we asked 4 different users to collect SenseCam data over a period of 30 days. A total of 239,033 SenseCam images were used in this experiment (all images of less than 4KB in size were ignored, as these are invariably images of total darkness, i.e. when SenseCam lens is behind a coat, etc.). Sequences of these images were segmented into events using only the sensor values [9]. Table 2 summarises some of the main statistics broken down by user.

Each and every image has sensor values associated with it, and also 4 MPEG-7 descriptor values were extracted. It takes approximately 30 minutes to process a normal day of 2,500 images (on a 2.4GHz Pentium 4 machine with 512MB RAM). Therefore to process all the images it took approximately 75 hours (150 days data, and 30 minutes to process each day).

10 diverse query events were selected for each user, thus giving a total of 40 queries/topics. The users were then asked to judge a large number of potentially relevant events against each query event to build up a groundtruth of data. This was done in a TRECVid style pooling approach [23]. The experiments in this paper are carried out on a groundtruth of 17,637 judgments from 4 different users across 4 different datasets. Compared to early TRECVid systems, the scale of this groundtruth is not insignificant, thus giving the retrieval results in this paper a certain degree of gravity.

After the groundtruthing stage the 40 queries (10 x 4 users) were then divided into a training and test set. The result is each user providing 6 queries for training, and 4 queries for testing. In terms of overall system judgement this means that there will be 24 queries (6 × 4 queries) for training and 16 queries for testing purposes.

4.2 Large Dataset with Specific User-Generated Queries

A disadvantage of the dataset used in section 4.1 is that while it was necessary to select very general queries to produce a sufficient number of relevant events on which to tune retrieval parameters, these queries are not entirely representative of *all* possible user query classes. Therefore we decided to create a second dataset on which users were asked to construct real world queries with specific information needs. This dataset is also much more extensive in terms of size, thus adding to the challenge of producing good retrieval results, but which also provides a more realistic evaluation of how such systems may perform in the real world.

In experiments to investigate the effectiveness of our retrieval approaches for real user-generated queries on extensive datasets, we asked 4 different users to collect SenseCam data over a long period of time. A total of 1,864,149 SenseCam images were used in this experiment. These images were segmented into 22,125 events using the optimal segmentation approach identified in the previous chapter. Table 2 summarises some of the main statistics broken down by user. Two MPEG-7 features (scalable colour and edge histogram) were extracted for all of the images, taking an average of 13 minues to process a typical day of 2,500 images, thus taking a total of 156 hours.

Users were asked to select a number of query events. They were presented with an event based browser to sift through their SenseCam images. In total, 23 events were selected as query topics.

The retrieval of potentially relevant events was then processed for each query event in our dataset. Users were presented with a screen full of candidate events (20 from each approach) in which they had the option to select other events relevant to their lifelog query event, which were in turn highlighed with a green background (see Figure 3). The 6 keyframe images chosen for each candidate event were spread evenly e.g. first is image 16% through event, second 33%, third 50%, etc. Below each candidate image there is a "Link" button which opens all the images of that event in a new browser window. 1,736 unique candidate events were retrieved for presentation to users for their judgement.

5. RESULTS

Our six approaches were evaluated on a set of 39 queries, 16 on a set of very general queries (from the 220k images discussed in section 4.1) with associated relevance judgements, and 23 on a set of queries with a specific information need (from the 1.8M images discussed in section 4.1) with the the top 20 results of each query judged by SenseCam users. We now detail the most significant findings from this extensive experimentation.

5.1 Comments on Individual Performance of *MPEG7Sense*, SIFT, & SURF

The *MPEG7Sense* approach worked slightly better overall on both the set of 16 general queries and on the set of 23 specific queries than either SIFT/SURF. On the set of general queries it was the best performing individual approach on 9 of the 16 queries (0.188 MAP vs 0.177 for SURF vs 0.133 for SIFT), and on the set of specific queries it was the best performing individual approach on 16 of the 23 queries (0.126 P@20 vs 0.089 for SURF vs 0.033 for SIFT).

Examining results on the set of "general" queries, all methods perform well on finding *driving events*, with a P@10 of 0.7 or greater. These events are common in the data, and the reliable detection of such events would allow these commonplace events to be removed from view, facilitating more efficient user browsing. SURF performs best overall with a MAP of 0.417 (SIFT: 0.4137, *MPEG7Sense*: 0.1121). For an *eating event* query, the retrieval was quite poor, with a maximum MAP score was 0.061 (MPEG7). The query keyframe was an image of the wearer's living room (while the SenseCam wearer is eating), and many returned results are of the same location, though often not involving *eating*. The low performance can be attributed to the difficulty in determining the semantic meaning of a query. Reasonably good results are obtained on querying *shopping events*. While the relevant events are visually quite different, they are usually cluttered scenes with many colours and edges. The *MPEG7Sense* colour and edge descriptors generalise well, and so do the SURF visual words, probably because some visual words capture properties common to shopping events. MAP scores for this query are SURF: 0.2835, *MPEG7Sense*: 0.2059 and SIFT: 0.0335.

On the 23 "specific" topics it is interesting to consider Figure 4 which plots the precision at 20 score of the 3 approaches. Selecting 3 sample events where each approach works clearly better than the other two, it is somewhat unsurprising that the *MPEG7Sense* approach works best on the "beaches" query, as returning back other events full of predominantly blue and yellow/orange are quite likely to be relevant, while there were not many features present in the given images explaining the relatively lower performance for SIFT and SURF. The SIFT approach works particularly well (with respect to the other approaches) on the "Lynda" query, where one user wished to compile all times he was talking to his colleague Lynda. These images were all taken in the same building where objects within the office would all have a similar look and feel. SIFT worked particularly well in this case because the objects, and their spatial arrangement, within the images are visually similar to the query event. The SURF approach works particularly well (with respect to the other approaches) on the special dinners query. The SURF encoded feature points may detect items such as elegant glasses or salt cellars, and then by reranking the top results based on bi-directional matches, there may be a number of other dinner events excluded as they are not wholly visually similar to the query event "special dinner".

5.2 Benefits of Fusing *MPEG7Sense* & SURF

Both the *baseline fusion* and the *score fusion* approaches offer improvements over individual runs in terms of MAP on the "general" set of queries. The best performing individual source on the set of general queries had a MAP score of 0.188 (*MPEG7Sense*). The *baseline fusion* approach had a MAP score of 0.203 and was at least as good as the the best result offered by either the individual *MPEG7Sense* or SURF approaches on 7 out of 16 "general" topics. Meanwhile the *score fusion* approach had a MAP score of 0.201 and was

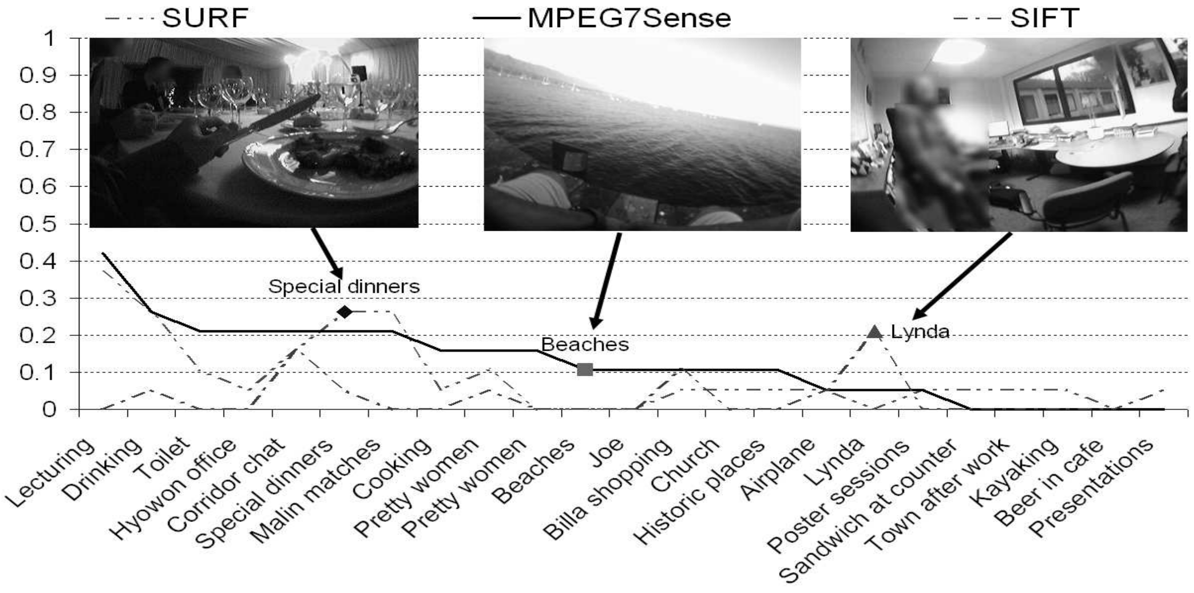


Figure 4: P@20 performance of *MPEG7Sense*, SIFT, and SURF on "specific" queries

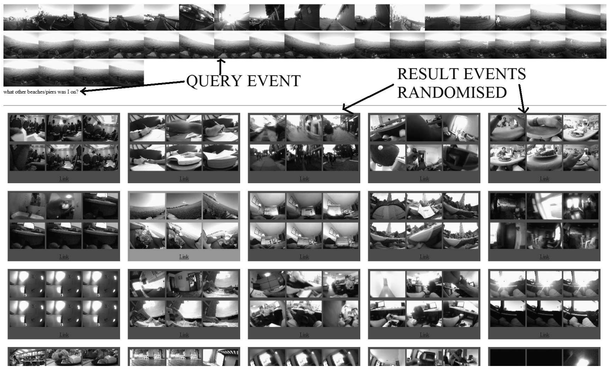


Figure 3: Judging results on specific user-generated queries from large dataset

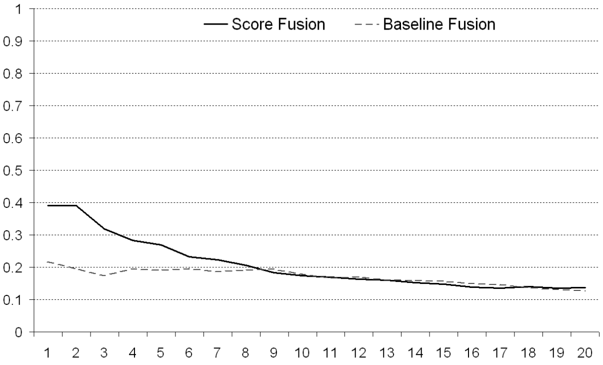


Figure 5: Comparing fusions at different "precision at" levels (x-axis) on "specific" queries

also at least as good as the best results offered by either the individual *MPEG7Sense* or SURF approaches on 5 out of the 16 "general" topics.

The performance of both fusion techniques is perhaps even more effective when dealing with the difficult queries that have a very "specific" information need. The best performing individual source on the set of "specific" queries had a P@20 score of 0.126 (*MPEG7Sense*). The *baseline fusion* approach had a P@20 score of 0.130 and was at least as good as the best of the *MPEG7Sense* and SURF approaches on 16 of the 23 queries. Meanwhile the *score fusion* approach marginally outperforms the *baseline fusion* approach with a P@20 score of 0.135 and improves on the performance of any individual SURF or *MPEG7Sense* approach on 15 of the 23 "specific" queries.

It is interesting to note that the *score fusion* approach returns more relevant documents earlier than the *baseline fusion* approach on the "specific" queries as illustrated in Figure 5. This effect is also present on the set of "general" queries also. An explanation for the phenomenon is that the *score fusion* approach firstly ranks candidates that are returned by both SURF and *MPEG7Sense*, and ranks those firstly. This may explain why for P@1, P@2 . . . P@8 scores are all better for the *score fusion* approach than the *baseline fusion* approach. However for candidates returned at ranks 10 and above the *baseline fusion* is the best performing approach. The likely explanation for this is that the *score fusion* strategy takes a more integrated approach to combine the retrieval scores, instead of the *baseline fusion* strategy of using SURF then switching to *MPEG7Sense* when the number of bi-directional matches drops below a threshold.

5.3 Benefits of Better Representing Events

As discussed in section 3.5 we now investigate the benefits of representing an event by considering *all* of its images, and not just one single keyframe image.

It can be seen that across many of the "'general query" topics the *"Event Average"* approach works best (in 14 out

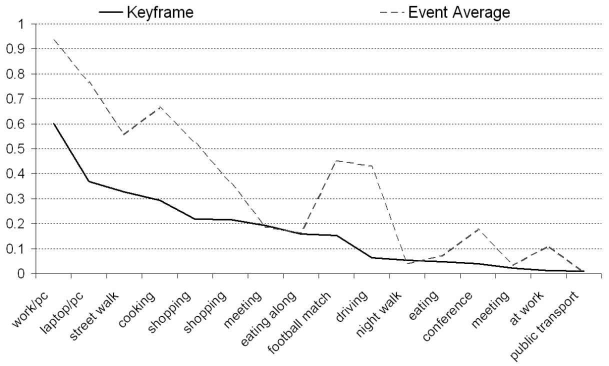
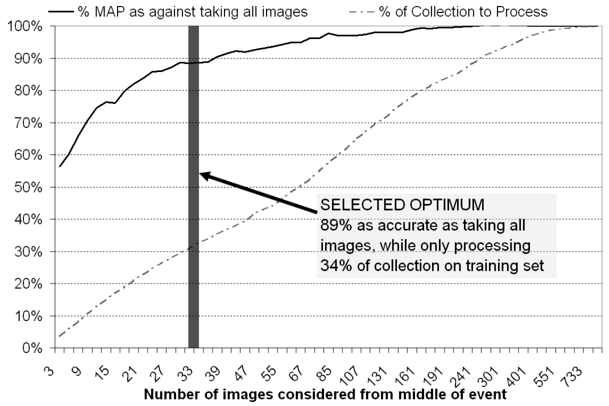


Figure 6: Retrieval performance of using single keyframe image vs. using average of features from all images in event (on *MPEG7Sense*)



SELECTED OPTIMUM
89% as accurate as taking all images, while only processing 34% of collection on training set

Figure 7: Performance in processing only a section of the images to represent an event

of 16 topics as illustrated in Figure 6). Overall the MAP score of the *"Event Average"* approach is 0.387 as against a MAP score of 0.188 for the *"Keyframe"* approach on "general" topics. On the "specific" topics the difference between both approaches is not as pronounced (0.174 P@20 for *event average* vs 0.126 for *keyframe*). However still on 14 of the 23 topics the *event average* approach performs best, and overall we believe this is the superior approach.

It is interesting to note that taking only a small selection of the images from around the middle of the event can perform *almost* as good as considering all the images from the given event. Figure 7 illustrates how considering more images around the middle of an event (as we move right on the x-axis) leads to a performance improvement. However, in the training set if we just extract MPEG-7 features from the middle 35 images in an event, we will perform 90% as well as when all the images in the event are processed. Instead of processing each and every image in the SenseCam collection, we can process just 34% of that collection, and still achieve 90% of the retrieval performance.

This approach of selecting the MPEG-7 features of the middle 35 images in an event, in combination with using sensor sources, performs 51% better than considering only the sensor sources only, thus proving that an analysis of the image content is necessary for better performance. While we need visual features for acceptable event retrieval performance, we have found that it is not necessary to extract features from each and every image, as after event segmentation (performed on only sensor source of information), we can get good retrieval performance by just extracting the middle 35 images from each SenseCam event.

6. CONCLUSIONS & FUTURE WORK

In this work we to address the significant challenge of retrieving material from a lifelog that is relevant to the users' information need. We set up two datasets to consider a broad range of user information needs, one containing 40 (24 for training, 16 for testing) "general" queries/topics (e.g. driving, eating, etc.), and another containing 23 more "specific" queries/topics (e.g. events of talking to Joe, visits to the museum, etc.). We have detailed results illustrating the *MPEG7Sense*, SIFT, and SURF are broadly comparable, and also highly complementary.

After training, we investigated two fusion techniques in this work, whereby we combined the results from the SURF and the *MPEG7Sense* approaches. These approaches offered improved retrieval performance. In future we intend to explore a new fusion method that firstly ranks results based on the *score fusion* approach and then by the *baseline fusion* approach.

It is very interesting to note that the approach towards modelling the event is highly influential in terms of retrieval performance. Representing events by taking the average feature vector value of all the images present in the events, rather than taking a single keyframe image, led to a 38% increase in retrieval performance on the set of "specific" queries, and to over twice the performance (106%) on the set of "general" queries. However as displayed in Figure 7, by only processing the middle 35 images of each event (just over 30% of the entire set of images), 90% of the retrieval performance of considering all the event's images is achieved. The SURF strategy of using a visual vocabulary lends itself to such extension and we will investigate the effectiveness of this in the future.

While performance is quite good with respect to the "general" queries (e.g. P@5 for *event average* approach = 0.69 overall), it is challenging to retrieve relevant results for queries seeking a very "specific" information need (where P@5 for *event average* approach = 0.30 overall). Firstly this is most likely due to the fact that there are less relevant potential results in the users' lifelog for the "specific" queries. Another reason for the relatively poor performance of the "specific" queries is that many of them may require an associated semantic meaning. In future we plan to extend previous exploration work in extracting the semantics of SenseCam images [6], and also in considering other contextual source of information such as GPS and Bluetooth [7].

In this work we try to identify the similarity between two events, however we consider both events autonomously. In most instances this is fine, but there are occassions when considering the temporal order of events may be worthwhile e.g. while two events of going through an airport may initially appear similar, one may be of arrival (events of in house and taxi before, and meal then plane after) and the other of departure (event of in plane before then luggage collection after). In future we will investigate the merits of considering the temporal order of other events adjacent to the query event and the candidate event.

Acknowledgments

We would like to extend our thanks to the participants in these experiments. We are grateful to the AceMedia project for equipment. This work is supported by Microsoft Research under grant 2007-056; the Irish Research Council for Science Engineering and Technology; and by Science Foundation Ireland under grant 07/CE/I1147.

7. REFERENCES

[1] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV'06)*, May 2006.

[2] G. Bell and J. Gemmell. A digital life. *Scientific American*, 2007.

[3] M. Blighe, A. Doherty, A. Smeaton, and N. O'Connor. Keyframe detection in visual lifelogs. In *PETRA - Conference on Pervasive Technologies Related to Assistive Environments*, July 2008.

[4] M. Blighe, H. le Borgne, N. O'Connor, A. F. Smeaton, and G. Jones. Exploiting context information to aid landmark detection in SenseCam images. In *ECHISE 2006 - 2nd International Workshop on Exploiting Context Histories in Smart Environments (Ubicomp 2006)*, Orange County, CA, 2006.

[5] M. Blighe, S. Sav, H. Lee, and N. O'Connor. Mo músaem fíorúil: A web-based search and information service for museum visitors. In *International Conference on Image Analysis and Recognition (ICIAR)*, 2008.

[6] D. Byrne, A. R. Doherty, C. Snoek, G. J. Jones, and A. F. Smeaton. Validating the detection of everyday concepts in visual lifelogs. In *submission*, 2008.

[7] D. Byrne, B. Lavelle, A. R. Doherty, G. Jones, and A. F. Smeaton. Using bluetooth and GPS metadata to measure event similarity in SenseCam images. In *IMAI'07*, pages 1454–1460, Salt Lake City, Utah, 2007.

[8] A. R. Doherty, D. Byrne, A. F. Smeaton, G. J. Jones, and M. Hughes. Investigating keyframe selection methods in the novel domain of passively captured visual lifelogs. In *CIVR: ACM International Conference on Image and Video Retrieval*, Niagara Falls, Canada, 2008. ACM Press.

[9] A. R. Doherty and A. F. Smeaton. Automatically segmenting lifelog data into events. In *WIAMIS: Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt, Austria, 2008.

[10] E. Fox and J. Shaw. Combination of multiple searches. In *TREC 2 : Text REtrieval Conference*, Gaithersberg, Maryland, USA, 1993.

[11] 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: 8th International Conference on Ubiquitous Computing*, volume 4602 of *LNCS*, pages 177–193, California, USA, 2006. Springer.

[12] T. Hori and K. Aizawa. Context-based video retrieval system for the life-log applications. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 31–38, Berkeley, California, 2003. ACM Press.

[13] A. Jaffe, M. Naaman, T. Tassa, and M. Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *MIR : The 8th ACM SIGMM international workshop on Multimedia information retrieval*, Santa Barbara, California, USA, 2006.

[14] H. Lee, A. F. Smeaton, N. O'Connor, G. Jones, M. Blighe, D. Byrne, A. Doherty, and C. Gurrin. Constructing a SenseCam visual diary as a media process multimedia systems. *Multimedia Systems Journal, Special Issue on Canonical Processes of Media Production (in press)*, 2008.

[15] W.-H. Lin and A. Hauptmann. Structuring continuous video recordings of everyday life using time-constrained clustering. In *Multimedia Content Analysis, Management, and Retieval : SPIE IST Electronic Imaging*, San Jose, California, USA, 2006.

[16] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60(2), pages 91–110, 2004.

[18] M. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 427–433, New York, NY, USA, 2001. ACM Press.

[19] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2006.

[20] C. Ó Conaire, N. E. O'Connor, A. Smeaton, and G. J. F. Jones. Organising a daily visual diary using multi-feature clustering. In *Proc. of 19th annual Symposium on Electronic Imaging*, 2007.

[21] N. O'Hare, H. Lee, S. Cooray, C. Gurrin, G. Jones, J. Malobabic, N. O'Connor, A. F. Smeaton, and B. Uscilowski. Automatic text searching for personal photos. In *SAMT 2006*, Athens, Greece, 2006.

[22] D. Tancharoen, T. Yamasaki, and K. Aizawa. Practical experience recording and indexing of life log video. In *CARPE : Second ACM workshop on Capture, Archival and Retrieval of Personal Experiences*, Singapore, 2005.

[23] E. M. Voorhees. Trec: Continuing information retrieval's tradition of experimentation. *Commun. ACM*, 50(11):51–54, 2007.

[24] Z. Wang, M. Hoffman, P. Cook, and K. Li. Vferretă: Content based similarity search tool for continuous archived video. In *CARPE*, Santa Barbara, California, USA, 2006.