# Automatically Segmenting LifeLog Data Into Events

Aiden R. Doherty and Alan F. Smeaton
Centre for Digital Video Processing and Adaptive Information Cluster
Dublin City University, Glasnevin, Dublin 9, Ireland.
adoherty@computing.dcu.ie

## Abstract

*A personal lifelog of visual information can be very helpful as a human memory aid. The SenseCam, a passively capturing wearable camera, captures an average of 1,785 images per day, which equates to over 600,000 images per year. So as not to overwhelm users it is necessary to deconstruct this substantial collection of images into digestable chunks of information, i.e. into distinct events or activities. This paper improves on previous work on automatic segmentation of SenseCam images into events by up to 29.2%, primarily through the introduction of intelligent threshold selection techniques, but also through improvements in the selection of normalisation, fusion, and vector distance techniques. Here we use the most extensive dataset ever used in this domain, 271,163 images collected by 5 users over a time period of one month with manually groundtruthed events.*

## 1. Introduction

The SenseCam is a small wearable device which incorporates a digital camera and multiple sensors including a 3-axis accelerometer to detect motion, a thermometer to detect ambient temperature, a passive infra red sensor to detect the presence of a person, and a light sensor. It is used to record a lifelog, or detailed visual record of daily activities, and Hodges *et. al.* detail the potential memory benefits of a personal visual lifelog such as that generated by a SenseCam [4].

A SenseCam captures 1,785 images on an average day creating a sizable collection of images even within a short period of time, e.g. over 600,000 images per year (or over 40 million in a lifetime). To manage such information it is important to automatically split these collections into manageable segments by identifing the boundaries between different daily events (Figure 1), e.g. having breakfast, working in front of a computer, attending a game of football, etc. Here we report significant progress in the accuracy of event-based segmentation of SenseCam images.

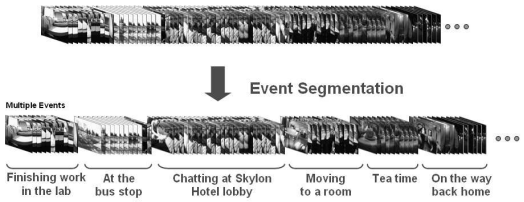


**Figure 1. Segmenting images into events.**

## 2. Related Work

The field of lifelogging is a relatively new but rapidly expanding area of research [1, 5, 8, 2] and it has been recognised that a visual lifelog should be segmented into maneagable shots/activities/events [5] to make it manageable. However to our knowledge no research has been conducted on segmenting lifelogs of data captured for the duration of entire days and from a range of users. Others have dealt with data collected from just one user (capturing for less than 6 hours a day) [5, 8]. In this paper 5 users captured SenseCam image data for an average of 10 hours per day over a 1-month period thus providing a more thorough representation of enduser lifestyles.

Wang *et. al.* segment their lifelog of video into 5 minute clips, however real events are not always 5 minutes in duration [8]. Lin & Hauptmann segment events through a time-constrained clustering technique [5], however they don't identify the boundaries between all events. Instead of having multiple events of "working at PC", they shall cluster all these events together. The time-constrained clustering technique of Yeung & Yeo [9] used in the video domain for story boundary detection is more appropriate. Techniques in the video domain such as motion analysis do not necessarily transfer to the lifelogging domain given the low capture rate (3 per minute on average). In previous work we focused heavily on investigating which individual and combined data sources are best for activity segmentation [2]. However the selection of a threshold for the number of events in a day was fixed at 20, and also we had no detailed groundtruth from which to calculate recall.

## 3. Segmentation Approaches

Our approach to segmentation shall now be explained. Firstly sequences of SenseCam images are broken up into a series of chunks, where the boundary between these chunks corresponds to periods when the device has been turned off for at least 2 hours (e.g. when the user has gone to sleep). Usually each chunk corresponds to a day's worth of SenseCam images. Each image is then represented by MPEG-7 descriptor values and values from SenseCam sensors described earlier. The MPEG-7 descriptors we select are: colour layout, colour structure, scalable colour, and edge histogram.

To segment a day of images into distinct events, processing follows these steps:

- Compare adjacent images (or blocks of images) against each other to determine how dissimilar they are.
- Determine a threshold value whereby higher dissimilarity values indicate areas that are likely to be event boundaries. e.g. a boundary is more likely to occur at a time of significant visual or sensory change as opposed to when little change occurs.
- Remove successive event boundaries that occur too close to each other.

### 3.1. Comparing adjacent (blocks of) images against each other

The first research question we address is, for each available source of information, whether to compare adjacent individual images, or blocks of images, against each other (like an adaptation of Hearst's Texttiling algorithm [3] used previously in this domain [2]). If the TextTiling techinque is found to be optimal for a given source, it must be determined what the best block size should be for that given source, e.g. blocks of 3, 7, 10, etc. images?

The second research question is to investigate the optimal vector distance metric to compare images against each other using MPEG-7 low-level descriptors. A total of 10 distance metrics are investigated in this work (listed later in Table 2).

Sensor readings are simply scalar values and thus the difference between adjacent readings is calculated using normal subtraction. To calculate the overall dissimilarity score for each image it is then necessary to normalise and fuse the various sources of information (MPEG-7, accelerometer, light level, ambient temperature, and passive infrared). Due to space constraints we are unable to describe how this was done in detail, but it has been empirically determined that *Min-Max* normalisation and *CombMIN* are the optimal techniques [6].

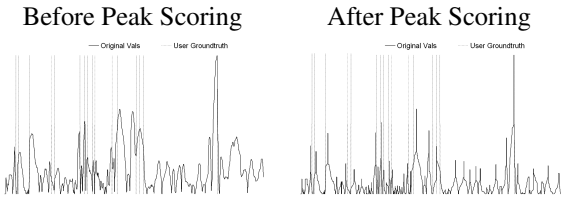


**Figure 2. Effects Of Peak Scoring.**

Higher dissimilarity scores between adjacent images indicates a greater likelihood of an event boundary taking place. In this case it is desireable to emphasise the instances where peaks occur (which makes a dissimilarity graph more "spiky" such as in Figure 2). To achieve this we get the total difference for each data point against it's neighbouring left- and right-most trough values. Consider Figure 3, the likelihood that image *n* will be an event boundary is (*h1+h2*), whereas the likelihood that image *n+1* is an event boundary shall just be *h3* alone (the likelihood score to the left of this image is not considered as it is of a greater value). We refer to this approach as *peak scoring* for the remainder of this paper.

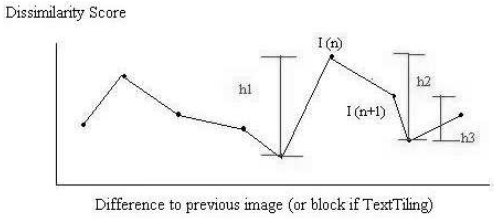


**Figure 3. Peak Scoring Diagram.**

### 3.2. Determining a threshold to select images that are event boundaries

After the first stage we are left with an event boundary likelihood score, for each image. The task is then to automatically determine which images should be chosen as boundaries. Previous work in this domain has only selected an arbitrarily fixed number of boundaries [2, 8], meaning that no allowance is made for the fact that, for example, on busy social days there may be many more events in the wearer's life than on other mundane workdays. As a result we investigate two thresholding techniques, one non-parametric (Kapur) and one parametric (Mean) [7].

### 3.3. Post-Processing: Bounday Gaps

At certain times, such as running to catch a bus, the SenseCam may record a large amount of visual and sensory change in successive images, and neighbouring images may

| User | Total Num Images | Groundtruthed Events | Avg Daily Duration |
|---|---|---|---|
| 1 | 80,934 | 995 | 13h 08m |
| 2 | 76,810 | 875 | 9h 27m |
| 3 | 44,447 | 348 | 10h 41m |
| 4 | 27,929 | 329 | 7h 45m |
| 5 | 41,043 | 439 | 9h 15m |

**Table 1. User Statistics**

be proposed as discrete event boundaries, even after thresholding. In reality only one of these should be selected as the event boundary. Therefore we investigate, for an image proposed as a boundary, the optimum amount of time to ignore subsequent images proposed as boundaries, e.g. 2, 3, 4, 5, 10, etc. minutes. We refer to this approach as *post-processing boundary gap* for the remainder of this paper.

## 4. Experimental Setup

5 users each wore a SenseCam over a one month period. To create a groundtruth, each user reviewed their collection and manually marked the boundary image between all events. It is important for privacy reasons that only the owner of the lifelog images is the person who groundtruths those images, as by their nature lifelog images are highly personal to that user. Table 1 provides a breakdown of the 271,163 images captured by our 5 users.

It takes approximately 30 minutes to process a busy day of 2,500 images on a 2.4GHz Pentium 4 machine with 512Mb RAM. The data was split equally into a training and test set, with training taking approximately 20 hours. The complete set of test images (around 135,000 in total) can be segmented into events in less than 2 minutes of CPU time and over 3,000 parameter combinations were tested and measured against the groundtruth.

## 5. Results

The effects of each of the approaches discussed earlier were investigated experimentally and unless stated, results are reported in terms of the F1-Measure.

### 5.1. TextTiling

The TextTiling approach was found to perform better on averge (than non-TextTiling) for the MPEG-7 (0.6023 vs. 0.5387), passive infrared (0.5151 vs. 0.0844), and temperature (0.4854 vs 0.4218) data sources. The optimal texttiling block size for the MPEG-7 source was to use the average value of 5 images grouped together, while for the temperature and passive infrared sensors it was optimal to use a block size of 8 images. The latter are sources of information that naturally change more slowly, e.g. the ambient temperature value will change relatively slowly over time.

| Vector Distance Method | F1-Measure |
|---|---|
| Histogram Intersection | 0.6271 |
| Euclidean | 0.6253 |
| Manhattan | 0.6166 |
| Squared Chord | 0.6023 |
| Jeffrey Mod KL | 0.6020 |
| Bray Curtis | 0.6013 |
| Square Chi Squared | 0.5907 |
| X2 Statistics | 0.5905 |
| Kullback Leiber | 0.5869 |
| Canberra | 0.5684 |

**Table 2. Vector Distance Methods Results**

| Thresholding Method | Precision | Recall | F1-Measure |
|---|---|---|---|
| Mean (k = 3.4) | 0.6294 | 0.6249 | 0.6271 |
| Kapur (64 bins) | 0.4891 | 0.7121 | 0.5799 |
| RIAO (top 20) | 0.6789 | 0.4642 | 0.5514 |

**Table 3. Overall Thresholding Performance**

However comparing individual adjacent values performs better than texttiling for the sources of information that do change quickly (e.g. motion values change very quickly when user is sitting down and then decides to walk to another location) namely the accelerometer (0.5284 vs 0.3307) and light (0.5209 vs. 0.3988) sensors.

### 5.2. Best Vector Distance Method

Table 2 shows that the performances of the 10 similarity measures for image comparison which we investigated and shows the Histogram Intersection method based on MPEG-7 features performs best.

### 5.3. Peak Scoring

We found that on average it is better to use our propsed peak scoring method which boosts the F1-Measure figures for overall boundary identification from 0.5378 to 0.6271. Out of the 63 days of test data from the 5 users, on 60 of these days the peak scoring method resulted in better segmentation performance.

### 5.4. Optimal Thresholding Technique

While the thresholding approach used in our previous work [2] performs best overall in terms of the number of true positives returned, and the Kapur method performs best in terms of recall; but the performance of the mean thresholding approach performed best on the test dataset in terms of producing both high recall and precision values, as is evident in Table 3.

### 5.5. Optimal Post-Processing Boundary Gap Method

As the magnitude of this parameter is increased, precision also increases accordingly, as the effects of over-segmentation (and thus false positives) is nullified (i.e. one false boundary triggered very soon after a true boundary will be ignored). However this naturally leads to a negative change in the level of recall too. Through experimentation we found that a gap of 3 minutes was best.

### 5.6. Overall Best Approach

Finally 3 different systems are compared: 1) Best trained system using a fusion of only sensor sources (only 3.3% worse than including MPEG-7 sources, but much quicker to process); 2) The temporal segmentation system of Wang *et. al.* [8]; 3) The system from our previous work [2]; 4) Time-constrained clustering system of Yeung & Yeo [9]

As shown in Figure 4 our proposed method (solid black line) offers a significant performance advantage over all other approaches in past literature, performing better on 45 out of 62 days and 29.2% better overall than the next best system (Yeung & Yeo [9]).

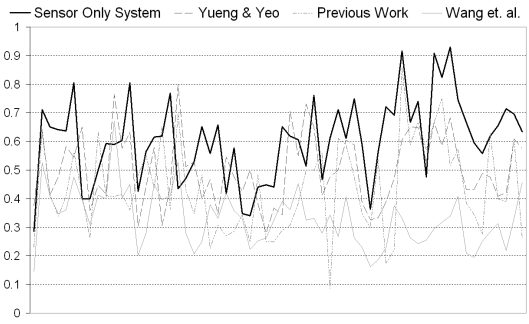


**Figure 4. Comparison of Other Systems**

To determine exactly where the main performance improvements came from (relative to our previous work[2], various facets of system 1 (Sensor Only) was applied to system 3 (RIAO) yielding the following results:

- Mean thresholding boosts performance by 14.7%
- Peak scoring resulted in an improvement of 9.4%
- CombMIN fusion as opposed to CombMNZ fusion resulted in a 6.3% improvement
- A post processing boundary gap of 3 minutes as opposed to 5 minutes relates to a 3.5% improvement

## 6. Conclusions

Considering the time penalty required to extract MPEG-7 features from images it is recommended to carry out event segmentation on a lifelog of images through the sensor sources alone, as processing is practically instant. As illustrated in Figure 4, improvements of 29.2% (using sensor sources only) have been made against previous work in the area of automatically segmenting SenseCam images. To all intents and purposes we can now regard the segmentation of a lifelog of images into events as a solved problem. With the ability of identifying events with reasonable accuracy, in future we intend to concentrate on numerous retrieval techniques to find similar events to a given event. Another challenge shall be to determine routine/mundane events as well as events of great importance.

**Acknowledgements:** Thanks to Gareth Jones, Georgina Gaughan, and Sandrine Áime for their advice. We are grateful to the AceMedia project and Microsoft Research for equipment; and to the Irish Research Council for Science, Engineering and Technology and Science Foundation Ireland under grant number 03/IN.3/I361 for support. We would also like to acknowledge the constructive comments provided by the reviewers.

## References

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

[2] A. R. Doherty, A. F. Smeaton, K. Lee, and D. P. Ellis. Multimodal segmentation of lifelog data. In *RIAO 2007 - Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, 2007.

[3] M. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *SIGIR - The 16th Annuan ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1993.

[4] 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*, 2006.

[5] 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*, 2006.

[6] 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, 2001.

[7] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electonic Imaging*, 13(1):146–168, 2004.

[8] Z. Wang, M. Hoffman, P. Cook, and K. Li. Vferret: Content-based similarity search tool for continuous archived video. In *CARPE Third ACM workshop on Capture, Archival and Retrieval of Personal Experiences*, 2006.

[9] M. Yeung and B.-L. Yeo. Time-constrained clustering for segmentation of video into story units. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 3:375–380, Aug 1996.