Multimed Tools Appl (2018) 77:12679–12699
DOI 10.1007/s11042-017-4910-8

# Automatic segmentation and summarization for videos taken with smart glasses

**Yen-Chia Chiu**[1] · **Li-Yi Liu**[1] · **Tsaipei Wang**[1]

Received: 13 October 2016 / Revised: 8 April 2017 / Accepted: 5 June 2017 /
Published online: 17 June 2017


**Abstract** This paper discusses the topic of automatic segmentation and extraction of important segments of videos taken with Google Glasses. Using the information from both the video images and additional sensor data that are recorded concurrently, we devise methods that automatically divide the video into coherent segments and estimate the importance of the each segment. Such information then enables automatic generation of video summary that contains only the important segments. The features used include colors, image details, motions, and speeches. We then train multi-layer perceptrons for the two tasks (segmentation and importance estimation) according to human annotations. We also present a systematic evaluation procedure that compares the automatic segmentation and importance estimation results with those given by multiple users and demonstrate the effectiveness of our approach.

**Keywords** Google Glass · Smart glasses · Egocentric video · Video abstraction · Video segmentation · Video summarization · Video diary

## 1 Introduction

There has been a large body of research works in the topic of video segmentation and abstraction. Most of the existing works focus on videos that are produced for the mass audience, such as movies and TV programs. Such videos usually consist of short shots. A typical procedure for segmentation starts with shot detection, followed by the grouping of similar shots. The task of abstraction involves selecting a set of key frames or short clips (skimming) to represent the whole video. There are also some types of videos that require specialized segmentation and/or summarization. For example, sports videos usually have relatively clear temporal structures given the rules of that particular sport, and the types of important segments or highlights are more easily defined. For surveillance videos, the main

✉ Tsaipei Wang
wangts@cs.nctu.edu.tw

[1] Department of Computer Science, National Chiao Tung University, Hsinchu City, Taiwan, Republic of China

task is to extract interesting segments (usually some types of events), which is helped by the fact that the whole video covers the same scene. There has been a large body of literature regarding the segmentation and summarization of each of these types of videos [2, 19, 20]. Some examples and surveys include [14, 18] for produced videos such as movies, [16] for newscast, [30] for sports videos, and [6] for surveillance videos.

Home videos, also called user-generated videos, pose a quite different set of challenges for segmentation and summarization. Such videos are mostly unscripted, contain many unintended and meaningless camera motions from, say, shaky hands, and usually have very long shots such that, unlike professional videos, shot detection is basically useless for their segmentation. As devices such as smart phones make it very convenient for generating such videos, researches on their summarization have also increased recently; some representative works include [1, 9, 10]. The majority of works on this topic focus on one of two aspects of home videos: The identification of camera motion as indications of the recorder's intension, and the extraction of important people or objects in the images.

As a special type of user-generated videos, an egocentric video is recorded from a recorder's "first-person" viewpoint, meaning that what is recorded is what the recorder sees. Such videos can be recorded with head-mounted cameras and, more recently, with smart glasses. While such videos have existed for some time, the introduction of Google Glass in 2013 certainly drew a lot of attention to their applications. One attractive characteristics of egocentric videos is that the recorder can proceed with his/her activities without the distraction of having to control the camera, and this provides many interesting new possibilities of video content generation, such as in [23]. The following are several existing research topics related to the content analysis of egocentric videos: The recognition of the recorder's status (such as walking, sitting, etc.) [4, 26], the summary of a video containing a single but complex activity (such as cooking, etc.) [15, 32], the extraction of people or objects of interest to the recorder [13], and the detection of particular recorder events in a given setting, such as viewing a painting in a museum [29].

To the best of our knowledge, this paper is the first attempt to segment and summarize egocentric videos using a subjective importance measure learned from human annotations. The existing works on egocentric videos, particularly those taken with smart glasses, focus on the identification of more well-defined events or objects. This can work well for more limited scenarios, such as for analyzing consumer interests in a store [13]. However, for videos consisting of more diverse activities and environments of everyday life, a recorder's or viewer's idea of importance, when selecting contents from such a video, is likely to be more general and may involve more diverse cues like image features, motions, interactions with people, trajectories, and so on. The subject of modeling how human rate and select such videos has yet to be studied.

Overall, the main contribution of this paper is the implementation and evaluation of a system that attempt to model how human viewers would segment and rate the importance of egocentric videos of everyday lives. The following is the target application scenario of this paper: The recorder, while wearing a smart glass, will go about doing multiple activities, possibly in multiple places while the camera stays on. The recorded video will likely contain segments of various degrees of importance. The objective is to process the video such that the more important segments are retained, and the less important parts (such as when moving between two places) are discarded. The end product is a shortened version of the original video, which is very different from traditional keyframe or skimming based summary. In addition, the technical contributions of this paper also include (1) a new protocol for evaluating

automatic segmentation results against human annotations, including methods for reducing the biases causes by individual preference of level of detail, (2) a procedure that integrates both the estimated cut likelihood and the estimated frame importance for segmentation, and (3) the use of trajectory-based features obtained from GPS, compass and inertia sensors for the purpose of segmenting and rating the importance of egocentric videos.

The overall block diagram of our approach is depicted in Fig. 1. With features extracted from the video frames and other sensors, a per-frame importance measure and a per-frame cut likelihood are computed. These two are used together to identify segments and per-segment importance ratings of the video. Since the extraction of important segments is a major target application of our system, the per-frame importance measure is applied first in the segmentation step, followed by additional cuts according to the cut likelihood measure. More detail on this step is given in Subsection 4.3.

In the rest of the paper, Section 2 contains discussions on previous works, Section 3 covers the features used in our system, and Section 4 describes the dataset and experimental methods. We present the evaluation methods and results in Section 5, followed by the conclusions in Section 6.

## 2 Related works

The segmentation of professionally produced videos, such as movies, has been well studied. Shot detection is the basic step here, and most methods for this task are based on the frame images. For example, Tse et al. [31] used a simple frame differencing method. Differences between color histograms are the basis of the methods used in [3, 7, 21]. Other features that are also commonly applied to video segmentation include motion and accompanying audio. For example, Zhang et al. [33] used camera motion (pan and zoom) to separate video segments. Lienhart [18] used characteristics of accompanying music to identify different segment types in movies. The summarization or abstraction of such videos also has been studied a lot. The typical approach is to extract representative keyframes and/or short clips (skimming) from the shots. Other related works include Fujimura et al. [8], which reduces redundancy by pruning contents from the same scenes, Zhu et al. [35], which utilizes object tracking, and Kannan et al. [12], which allows user-specified preferences to control the proportions of material selected from different scenes.

The segmentation and summarization of home videos usually require different sets of techniques than for professional videos [17]. Han [9] discusses the topic of extracting

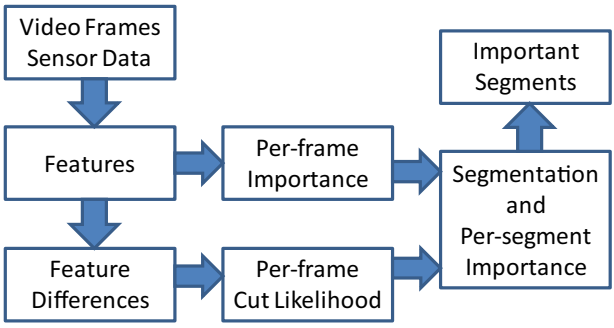


**Fig. 1** The overall block diagram of our system

interesting objects from such videos. Hua et al. [10] introduces a system that selects highlight segments from home videos and combines them with music to form an edited video. An interesting application of creating a summary of multiple recordings of the same event is proposed in [34]. In Abdollahian et al. [1], the topic is how motions in home videos affect viewers. Viewers' responses are furthered considered as cues for summarization in [25]. The work of Cricri et al. [5] studies the usage of inertia sensors to analyze camera motions.

Egocentric videos present a different set of challenges from regular home videos (i.e., those taken with camcorders or smartphones) because they are recorded with no intentional control. As summarized in [29], simple visual saliency is usually insufficient in their analysis. Overall, motion has been the most widely used feature type in their analysis. One main task is to divide a video into short segments, each having a consistent motion pattern; an earlier example is [22]. This is actually similar to some works on home videos, but with a different set of possible motions (for example, no zooming for egocentric videos). Several works propose methods to recognize the recorder's state using the motion patterns. For example, in [4], the states of recorders, who are sightseers, are classified as walking, standing, etc., and [26] attempts to classify states of recorders into classes of walking, riding, etc., based on the distribution of motion vectors that are integrated over time to reduce effects caused by spontaneous and random head movements. In [28, 29], motion features are used to determine whether the recorder is focusing on something in the scene.

Recently, there have also been works that combine more features for more diverse applications of egocentric video analysis. For example, [15] attempts to predict the user's gaze from the recorded images, and [32] uses motion and visual saliency to predict the recorder's attention. In [13], event detection is accomplished by clustering the frames, and multiple cues are integrated to recognize objects in the images that the recorder interacts with.

However, none of the existing works directly address our target problem for egocentric videos. Works on segmentation actually focus on the recognition of the recorder's motion state, such as [26], and have ignored scene characteristics that are essential for segmenting regular videos. On the other hand, works on extracting interesting segments, such as [29], only aim to find a limited set of events that can be defined somewhat objectively, and as a result are not sufficient to handle the diversity of subjectively interesting segments of everyday lives. Therefore, we aim to propose a system here that handles the task of more general segmentation and importance rating of egocentric videos by learning from human annotations.

## 3 Description of features

We focus on the explanation of the features used in our analysis in this section. There are a total of 8 features: Inter-frame color difference, mean color saturation, mean hue, hue consistency, degree of detail, forward motion, duration of stay, and existence of dialog. All these features are used in the task of segmentation. All but inter-frame color differences and degrees of detail are used for the estimation of frame importance. Since we want to learn from the annotators' ideas of important segments, which are purely subjective and can vary between different annotators, the notion of importance is not well-defined this way. As a result, we do not attempt to design features that specialize to some type of importance (except for the dialog feature), and instead just design some features that, from our empirical observations, appear correlated with some manually important video segments. We expect the combination of these features is able to capture more diverse concepts of importance.

The original video taken with Google Glass has a frame size of 1280 × 720 and a frame rate of 30 fps. For better efficiency, we only use one frame per second in our analysis. This is acceptable because we do not intend to find very precise segmentation points as in shot detection.

The first four features are based on colors. To reduce the effect of motions on color features, we choose to first identify per-frame representative colors and then use them to compute these features. For each frame, we first compute a reduced-size image of 128 × 72. Standard k-means clustering is applied to the set of RGB values from five contiguous frames (the current frame plus its two previous and two subsequent frames). Each cluster is a representative color. Figure 2 shows an example frame with its representative colors at various $k$. We empirically choose to use $k = 20$ when computing our color features.

### 3.1 Inter-frame color difference

To estimate the color difference between two frames, we apply the Hungarian Algorithm to match their representative colors. The cost of matching (weighted sum of absolute differences of the matched colors) is used as the measure of frame difference. The resulting values are smoothed temporally using a Gaussian filter with $\sigma$ of 30 s. We use $f_{CD}(t)$ to represent the inter-frame color difference between the frames at time $t$–1 and $t$.

### 3.2 Mean color saturation

Frames with higher color saturation are more likely to represent interesting scenes. In addition, abrupt changes in saturation might indicate scene changes, especially indoor-outdoor transitions. These are our motivations of using this feature.

We first convert the representative colors to the HSV color space, and then compute this feature as the weighted average of their saturation values. The weights are given by the numbers of pixels associated with the representative colors. The resulting values are smoothed temporally using a Gaussian filter with $\sigma$ of 10 s. We use $f_{MS}(t)$ to represent the result at time $t$.

### 3.3 Mean hue

Let us treat each representative color as a 2-D unit vector with the direction given by its hue angle. The mean hue is the direction angle of a weighted average of these vectors. The weight of a color is the product of two factors: The number of its associated pixels, and its saturation

**Fig. 2** An example frame (*left*) with its 20 main colors (*right*)

value. The resulting values are smoothed temporally using a Gaussian filter with $\sigma$ of 10 s. We use $f_{MH}(t)$ to represent the result at time $t$.

### 3.4 Hue consistency

Consider the weighted average of color vectors in Subsection 3.3. When the hues of the colors are more similar to one another, the weighted average vector will have larger magnitude because of constructive addition. Therefore, we can use the magnitude of this average vector to represent as a measure of hue consistency. The resulting values are smoothed temporally using a Gaussian filter with $\sigma$ of 10 s. We use $f_{HC}(t)$ to represent the result at time $t$.

### 3.5 Degree of detail

The degrees of detail of video frames depend on their contents. In general, frames with more details are more likely to be interesting to the user and therefore should be more likely to be important. For a given frame, we simply use the ratio of edge pixels found with the Canny edge detector to represent its degree of detail. The resulting degrees of detail of the frames in a video are smoothed temporally using a Gaussian filter with $\sigma$ of 10 s. We use $f_{DD}(t)$ to represent the result at time $t$.

### 3.6 Forward motion

Motion features are actually intended for the identification of user behaviors. For example, continuous motions may indicate that the user is moving between two places, making the part of video less likely to be important. On the other hand, if the user stays at a place for an extended amount of time, this might indicate the user is at his/her intended location, making the part of video more likely to be important.

When the person wearing the Google Glass is moving between two places, the motion of the image frame is likely to exhibit the characteristics in Fig. 2, with the motion vectors pointing outward from the image center. Our idea is to compare the estimated motion vectors with the pattern in Fig. 3 to determine the likelihood of whether the recorder is moving forward. The procedure is listed below:

(1) Compute the optical flow between adjacent frames (sub-sampled to 320 × 180).
(2) Divide the frame into 3 × 3 blocks and compute the per-block mean directions. To reduce the interference from moving objects in the frames, we use a robust estimation method: The estimation of the mean direction is refined three times by excluding samples that are beyond a standard deviation from the current mean.
(3) Substract the mean direction of the center block from all the per-block mean directions to exclude the effect of pan/tilt motion.
(4) Compute $d_\theta(t)$, the total absolute difference between the per-block mean directions of the frame at time $t$ and their respective target directions, which are depicted in Fig. 3. The degree of forward motion is given by

$$f_{FM}(t) = \begin{cases} 1 & \text{if } d_\theta(t) \le \pi, \\ \dfrac{4\pi - d_\theta(t)}{3\pi} & \text{if } \pi < d_\theta(t) \le 4\pi, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

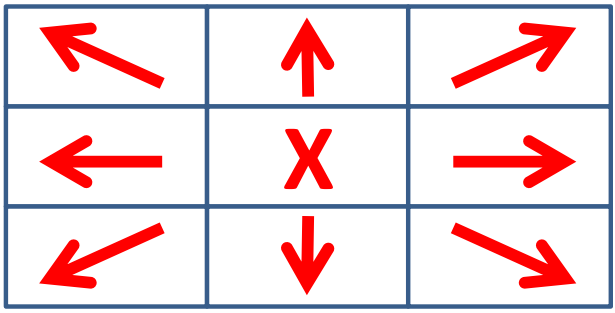

**Fig. 3** The target motion directions of the image regions for forward motion

Finally, the resulting degrees of forward motion of the frames are smoothed temporally using a Gaussian filter with $\sigma$ of 10 s.

We want to note here that the motion features in Fig. 3 are similar for forward motion and zooming-in. This is not a problem here though, as zooming is not available on smart glasses. For devices that allow zooming-in, more sophisticated analysis or the integration with other motion sensors are required to distinguish the two difference cases.

### 3.7 Duration of stay

When the user stays at the vicinity of a location for a significant amount of time, it might be due to some event that is important for the user at that location. Therefore, we use the duration of the user staying at a certain location as a cue of the important of that location in the video.

In order to estimate the duration of stay at different locations, we need to first estimate the trajectory of the user. The procedure is listed below:

(1) Estimate the user's velocity using the accelerometers. Due to the fact that the orientation of the Google Glasses (and therefore the accelerometers) changes constantly over time, the direct integration of accelerometer readings do not yield accurate velocities. As a result, we take a different approach that focuses on the user's walking motion. The velocity is estimated as a multitude of the rate of walking steps. Currently we simply use 0.8 m as the average step size. While this will not work if the user is not moving by walking, such as when the user is riding a bike, such cases usually occur outdoors and can be handled with absolute location information from, say, GPS.
(2) Estimate the relative directional change by integrating the gyroscope readings.
(3) Use the velocity and directional change information to construct a preliminary trajectory.
(4) Refine the locations and directions when absolute localization information (GPS and electronic compass readings) is available.

An example of trajectory estimation is shown in Fig. 4: Fig. 4a is a manually annotated trajectory overlapped on Google Map, and Fig. 4b is the automatically determined trajectory. We can see that they possess very similar characteristics.

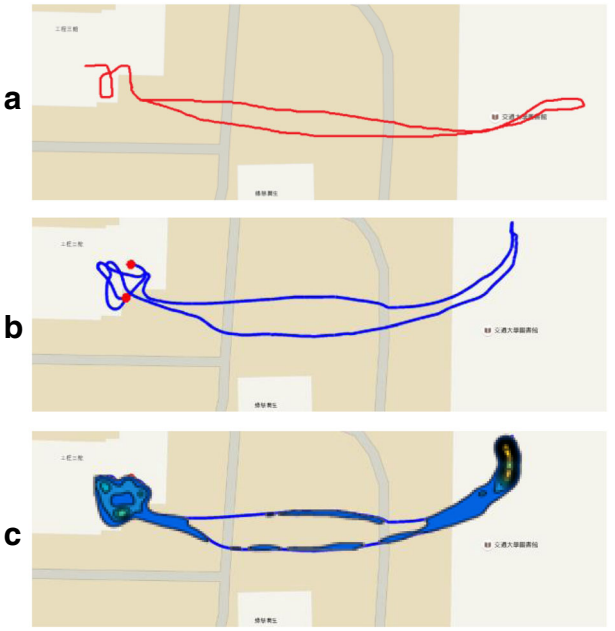


**Fig. 4** Example result of the estimation of the duration-of-stay feature. **a** A snapshot of Google Map with manually marked trajectory. **b** The same as (**a**) with an automatically estimated trajectory. **c** The relative durations of stay at different locations along the trajectory shown as pseudo colors overlaid on (**b**)

Let $\boldsymbol{p}(t)$ be the spatial location of the frame at time $t$ on the estimated trajectory. The duration-of-stay feature of this frame is given by

$$f_{DS}(t) = \sum_{t'} \exp\left(-\frac{\left\|\mathbf{p}(t)-\mathbf{p}(t')\right\|^2}{2\delta^2}\right) \tag{2}$$

where the summation of over all the frames in the trajectory. The factor δ is set at one meter in our experiments. We show in Fig. 4c the estimated relative duration of stay overlapped on the map. We can see that the user spent most of the time at a location to the right.

### 3.8 Existence of dialog

If the video contains a segment where the user is talking with someone, we believe such a segment is more likely to be considered important by the user. Here we discuss the cues we use to identify such segments, including the audio cue, which is about whether the recorded sound contains human speech, and the visual cue, which is about whether the video images contain human faces.

For the audio cue, we apply the short-time Fourier transform to the recorded audio signal with a window size of one second. We add up the squared coefficients in the frequency range of 150 to 500 Hz [24] and use the result as the estimation of the amount of human speech sound at a time point. The resulting values are smoothed temporally using a Gaussian filter with $\sigma$ of 5 s. To identify the range of time in which there exists significant sound level of human speech, we first find continuous segments with energy levels of at least 50,000 and maximum energy level of at least 80,000. These thresholds are estimated empirically. The found ranges are refined with a minimum filter with a half-width of 2 s, followed by a maximum filter with a half-width of 3 s.

For the visual cue, we apply the face detector of OpenCV to the image frames. The intervals of frames with detected faces are refined with minimum and maximum filters in the same way as for the audio cue. This cue is useful to exclude the effect of sounds that are not from a dialog involving the user.

Finally, we intersect the sets of frames identified with the audio and visual cues to obtain possible frames of dialogs. The process is illustrated in Fig. 5: Fig. 5a contains a sequence of frame snapshots of a video sequence, Fig. 5b indicates the actual frames that contain dialogs, Fig. 5c and d indicate frames selected with the audio and visual cues, respectively, and Fig. 5e gives their intersection. We define $f_{dlg}(t)$ as +1 if the frame at time $t$ is identified as possibly containing dialogs, and 0 otherwise.

## 4 Experimental methods

### 4.1 Description of test data

There are seven videos recorded using Google Glasses used in our experiments. The videos are indexed from 1 to 7 and represent a diverse set of everyday activities in a campus. Table 1 summarizes the scenarios in the videos. Lengths of the individual videos range from 3 to 15 min, with a total of 48 min. The videos contain both outdoor scenes (labeled as 'C' in Table 1 regardless of the actual locations) and indoor scenes (other labels in Table 1).

### 4.2 Human annotation of test data

The division of a long continuous video into meaningful parts and the estimation of their importance are very subjective. To allow for more comprehensive and reliable evaluation of our methods, we ask a number of people other than the one who takes the video to act as the annotators, who provide manual segmentation and per-segment importance rating for the videos. A labeling program is implemented for this purpose, with an example screenshot shown in Fig. 6.

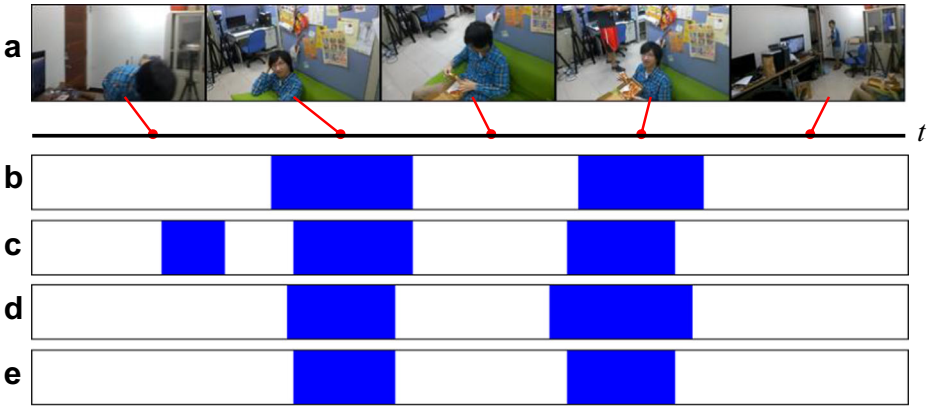


**Fig. 5** The illustration of the procedure to identify video frames with dialogs. **a** Snapshots of a video that contains dialogs. Their temporal locations are marked on the time axis. **b** Human labeled video frames that contain dialogs. **c** Frames identified with the audio cue. **d** Frames identified with the visual cue (*face detection*). **e** The intersection of (**c**) and (**d**)

**Table 1** Description of test videos

| Video no. | Scenes |
|---|---|
| 1 | O → C → L → C → O |
| 2 | C → L1 → L2 → C |
| 3 | (night scene) H → C → F → H |
| 4 | H → C → S → C → H |
| 5 | H → C → H |
| 6 | O → C → L → C → S → C → O |
| 7 | O (with two segments of dialogs) |

Outdoor scenes: C: campus; F: food stand

Indoor scenes: O: office; H: hallway; S: store; L: library

(L1 and L2 indicate two separate areas in the library)

For each video, the annotator is first asked to segment it at three different levels of detail: 3 ~ 5 segments at Level 1, 5 ~ 9 segments at Level 2, and 7 ~ 12 segments at Level 3. The purpose of using these levels is to avoid the biases in segmentation caused by the individual annotator's preference of level of detail. Subsequent evaluations are done for each level separately. After finishing the segmentation, the annotator is asked to rate the importance of each segment as being Low, Medium, or High. For illustration purpose, Fig. 7 displays the three-level segmentation and the importance ratings of a video by a human expert. The importance ratings are color-coded as green, yellow, and red for Low, Medium, and High importance, respectively. We do not specifically instruct the annotators what properties of the videos constitute importance here. This is because our objective is to learn from the annotators' subjective judgments without limiting their considerations to specific events or scenarios.

### 4.3 Classifier design

This subsection covers the core of our algorithm, which is to use the extracted features to generate the video segments and their importance ratings. The overall framework is depicted in Fig. 8. Two quantities are computed for each frame: $S_{cut}(t)$, which is a measure of whether a cut should occur between frames $t$ and $t + 1$, and $S_{imp}(t)$, which is a measure of its importance.

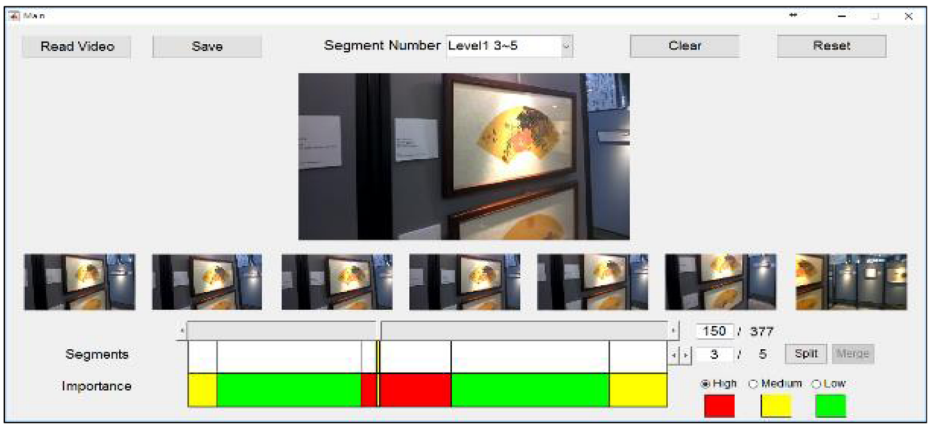


**Fig. 6** A screenshot of the program for the annotators to do manual segmentation and importance rating


**Fig. 7** Example segmentations and importance ratings by human annotators. Two sets of results are shown here, each consisting of results for Levels 1 to 3 (from *top down*) on the same video

Two multi-layer perceptrons (MLPs) are used in the process. We will call them the "cut" and "importance" MLPs, respectively, given the tasks in which they are involved. Both MLPs have two hidden layers with four neurons each and a single output neuron. The outputs for both MLP are between 0 and 1. For the first and last 30 s in each video, we always use 0 as the output from the "importance" MLP because these frames usually involve the user operating the Google Glass, making the results less reliable.

Each sample for the "cut" MLP is a pair of adjacent frames that are one second apart. We train this MLP for a two-class problem, with the classes being "cut" and "no-cut", corresponding to whether the two frames should be put in two different segments. The seven inputs for the "cut" MLP are based on the features described in Subsections 3.1–3.7. For the 6 features other than the inter-frame color difference, the inputs to this MLP are actually their amounts of change between two consecutive frames. All the 7 inputs are normalized so that their minimum and maximum values in the training set are zero and one, respectively. $S_{cut}(t)$ is just the output of this MLP.

Here we describe how we construct the training set for the "cut" MLP. For a given video used for training, we initialize $n_{cut}(t)$ to be the total counts of human-labeled cuts at time $t$, summed over the results of all the annotators and all the three levels. The results are smoothed

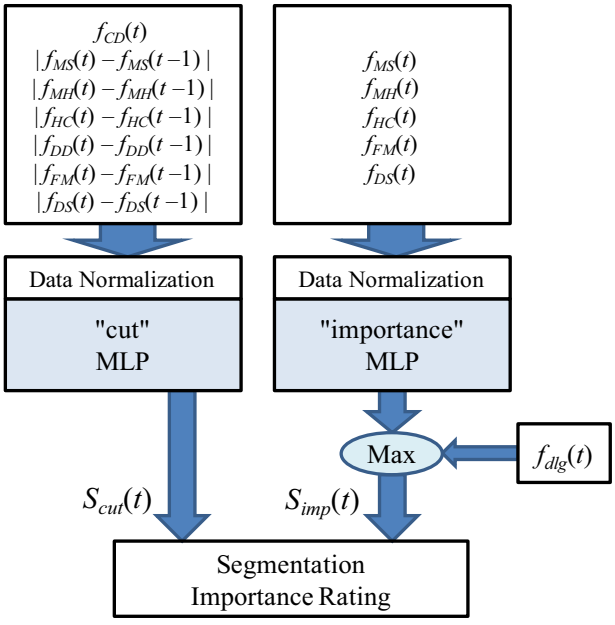


**Fig. 8** The framework for automatic segmentation and importance rating from the computed features

temporally using a Gaussian filter with σ of 10 s. A threshold of 0.5 is applied to the smoothed $n_{cut}(t)$ to separate the samples into the two classes. Due to the fact that the "no-cut" class has many more samples than the "cut" class, we choose to duplicate the samples in the "cut" class so that the two classes have approximately equal numbers of training samples, with the duplicating ratios proportional to $n_{cut}(t)$.

Each sample for the "importance" MLP is a single frame. We also train this MLP as a two-class problem, with the two classes being "important" and "unimportant". This MLP uses only 5 inputs, with the inter-frame color difference, degree of detail, and dialog features excluded. The inputs here are normalized as well.

To label a frame as one of the two classes for training purpose, we compute its average importance rating over all the annotations and all the three levels of detail, followed by Gaussian smoothing with σ of 10 s. A threshold of 0.5 is applied to label this frame as in the "important" or "unimportant" class.

The information about the existence of dialog is integrated into the importance ratings, as we want to keep segments that contain dialogs as important ones. We first smooth $f_{dlg}(t)$ with a Gaussian filter with σ being 10, 6, and 4 s for the three levels of detail, respectively. We then compute $S_{imp}(t)$ as the maximum of $f_{dlg}(t)$ and the output of the "importance" MLP. The following is the rationale of treating the dialog feature separately, instead of using it as just one of the features for the MLPs: As long as dialog exists in a segment, we consider the segment as being important regardless of what else are in the image.

### 4.4 From classifier outputs to video segments

This subsection explains how we combine $S_{cut}(t)$ and $S_{imp}(t)$ to divide a video into segments and estimate their importance. Given the main target application of extracting important segments from a video, we start with $S_{imp}(t)$ and then applies $S_{cut}(t)$ to identify finer segmentation if necessary. The procedure is explained below:

(1) The first step is to extract the "important" segments, as illustrated in Fig. 9. We first identify maximal intervals of the video where $S_{imp}(t)$ stays above a threshold (we use 0.5). The area under $S_{imp}(t)$ is then computed for each of these intervals. Assuming that the largest area of such intervals is A, we identify each interval with an area of at least A/3 (for Level 1), A/5 (for Level 2), or A/10 (for Level 3) as a segment. Cutting points are placed at the beginning and end of such a segment.

(2) If the number of segments is below the lower bound specified for the given level of detail after the previous step, additional cut points are added at the local maximums of $S_{cut}(t)$. These local maximums of $S_{cut}(t)$ are selected in the descending order. To prevent the generation of spuriously short segments, we will skip a local maximum if it is too close to

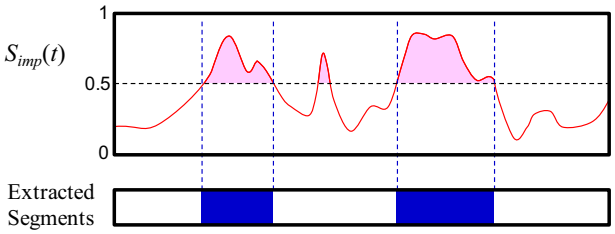


**Fig. 9** An illustration of the initial extraction of important video segments based on $S_{imp}(t)$

any existing cutting point; the required temporal separation is 20, 10, and 6 s for Levels one to three, respectively. This is repeated until the lower bound is reached or the local maximums are exhausted.

(3) Finally, we assign the importance rating of each segment based on the mean $S_{imp}(t)$ within that segment. Let $S_{imp}^{*}$ be the mean $S_{imp}(t)$ of a segment, its importance rating is High, Medium, and Low for $S_{imp}^{*} \geq 2/3$, $2/3 > S_{imp}^{*} \geq 1/3$, and $S_{imp}^{*} < 1/3$, respectively.

## 5 Experimental results

In this section, we present the evaluation of our results in three different aspects: per-frame importance rating, segmentation accuracy, and important segment extraction. The evaluation of our results is always obtained with leave-one-video-out cross-validation. Furthermore, the last video (no. 7) is not used for training because it contains much higher proportion of dialog than other videos, making the annotations of its segmentation and importance rating less applicable to other videos.

To compare our method with existing related techniques, several existing approaches are implemented and applied to the same set of videos. Here are the approaches used for comparison:

Per-frame importance rating based on visual saliency, using the method of [27]. The code is made available by the authors. We use the abbreviation SAL (meaning SALiency) to represent the results of this method.

Per-frame importance rating based on inverse motion magnitude [28]. From the experiments in [29], this is better than visual saliency in detecting "engagement" in egocentric videos. We use the abbreviation IMM (meaning Inverse Motion Magnitude) to represent the results of this method.

An adaptive fusion method that combines static attention, motion attention, and face detection to compute per-frame saliency from [19]. We use the results of [27] for static attention and use the method in [19] for computing motion attention. The audio part in [19] is ignored. We use the abbreviation AFF (meaning Adaptive Fusion Function) to represent the results of this method.

When comparing the results of the methods above with our results, their outputs are used in place of the output of our "importance" MLP as in Fig. 7. The other elements of our system ($f_{dlg}(t)$ and $S_{cut}(t)$) are not changed. Even so, as we will see in our analysis (Subsection 5.4), the per-frame importance rating carries the most weight when extraction important segments.

Concerning the computational cost, currently the processing is approximately 0.1 s per frame; the environment is a PC with Intel i7 CPU and MATLAB 7. The majority of the time is spent on the feature computation part, especially on motion estimation.

### 5.1 Evaluation of per-frame importance ratings

Here we use mean absolute difference (MAD) as the difference between two sets of per-frame importance ratings of a video. To evaluate the accuracy of a set of rating for a test video, we

compare it to the ratings of that video from all the annotators and take the average MAD. A computed set of rating is first scaled so that its minimum and maximum are 0 and 1, respectively. On the other hand, the annotated rating is set at 0, 0.5, and 1 for a frame in a low-, medium-, and high-importance segment, respectively.

The average MAD values of our results against the annotations are shown in Table 2. Values for each test video and each of the three levels are listed separately. For comparison purpose, we also list in Table 2 the average pairwise MAD between all the annotations. When comparing the automatic and annotators' results, the performances are approximately the same for Levels 1 and 2 and only different by 0.04 for the more fine-grained Level 3. This indicates that our system is able to approximate the human annotators' importance ratings up to the variations among the annotators themselves.

The comparisons between multiple methods for importance ratings are listed in Table 3. For brevity, the MAD values of the seven videos are aggregated together weighted by their numbers of frames. The method marked as "noDS" is a version of our method that excludes the use of DS features (Subsection 3.7). This allows us to determine whether this new feature of ours improves the importance rating accuracies. Based on the results in the two rows marked as "Ours" and "Ours_noDS", we can see that this feature does produce moderate improvements.

Also listed in Table 3 are results from the three reference methods from [19, 27, 28]. We can see that our method performs much better than the other methods. It is also interesting to see that SAL (visual saliency) is the best among the three methods, indicating a correlation between visual saliency and what the annotators consider to be important. This observation is apparently different from that in [29], where motion-based importance measures like IMM works much better than saliency. The following are two possible explanations:

- Unlike the problem considered by [28, 29], what viewers' consider important in a video may not be related to the recorder's actions. An example is walking in a garden. Therefore, a scene that is simply more interesting, which yields higher saliency grades, may be more important in some scenarios.
- A significant portion of our videos are taken outdoors where the surroundings (buildings, etc.) are quite far away. The motion magnitudes estimated from the frames are quite small even if the recorder is moving. This makes it difficult to distinguish them from motion

**Table 2** Evaluation of importance rating results (MAD)

| | Computed results | | | Manual annotations | | |
|---|---|---|---|---|---|---|
| Video no. | L1 | L2 | L3 | L1 | L2 | L3 |
| 1 | 0.11 | 0.14 | 0.13 | 0.18 | 0.19 | 0.09 |
| 2 | 0.23 | 0.22 | 0.20 | 0.07 | 0.12 | 0.10 |
| 3 | 0.37 | 0.39 | 0.40 | 0.33 | 0.25 | 0.23 |
| 4 | 0.12 | 0.12 | 0.13 | 0.12 | 0.09 | 0.09 |
| 5 | 0.19 | 0.20 | 0.18 | 0.35 | 0.33 | 0.26 |
| 6 | 0.11 | 0.12 | 0.11 | 0.10 | 0.12 | 0.10 |
| 7 | 0.29 | 0.22 | 0.20 | 0.33 | 0.26 | 0.21 |
| Average | **0.17** | **0.17** | **0.17** | **0.17** | **0.16** | **0.13** |

**Table 3** Comparison of frame importance measures (MAD) for different methods

| Method | L1 | L2 | L3 |
|---|---|---|---|
| Ours | 0.17 | 0.17 | 0.17 |
| Ours_noDS | 0.18 | 0.19 | 0.19 |
| SAL | 0.28 | 0.28 | 0.29 |
| IMM | 0.36 | 0.35 | 0.32 |
| AFF | 0.40 | 0.37 | 0.37 |

magnitudes computed when the recorder is engaged with something close, hence leading to reduced effectiveness of such features.

### 5.2 Evaluation of segmentation results

Since it is difficult to define an "average" segmentation of a video from multiple sets of annotations, we choose to compare the segmentation result with each annotator's segmentation separately. The evaluation metric we use here is the adjusted Rand index (ARI) [11], which is widely used in measuring the similarity between two ways of partitioning a common set of data. ARI values range between −1 and +1, with +1 resulting from two identical partitions and 0 being the expected value for two random partitions.

We show in Table 4 the average ARI between our segmentation results and all the annotators' results. Values for each test video and each of the three levels are listed separately. For comparison purpose, we also list the average pairwise ARI between all the annotations. When comparing the computed and annotated segmentations, the average ARI difference is only around 0.1. This indicates that our segmentations are relatively close to the human annotators' segmentations up to the variations among the annotators themselves.

### 5.3 Evaluation of the extraction of important segments

In this subsection, we consider the problem of selecting only part of the segments of a video according to their estimated importance. This is evaluated as a retrieval problem using precisions and recalls as the metrics. For a video with multiple sets of annotations, we compute the confusion matrix of our extracted segments against each set of annotation, and then use the

**Table 4** Evaluation of segmentation results (ARI values)

| | Computed results | | | Between annotations | | |
|---|---|---|---|---|---|---|
| Video no. | L1 | L2 | L3 | L1 | L2 | L3 |
| 1 | 0.68 | 0.56 | 0.55 | 0.62 | 0.61 | 0.72 |
| 2 | 0.46 | 0.66 | 0.62 | 0.72 | 0.70 | 0.70 |
| 3 | 0.45 | 0.46 | 0.58 | 0.48 | 0.70 | 0.73 |
| 4 | 0.58 | 0.57 | 0.49 | 0.75 | 0.79 | 0.69 |
| 5 | 0.58 | 0.49 | 0.47 | 0.77 | 0.62 | 0.63 |
| 6 | 0.84 | 0.76 | 0.64 | 0.75 | 0.85 | 0.70 |
| 7 | 0.35 | 0.68 | 0.64 | 0.49 | 0.57 | 0.63 |
| Average | **0.63** | **0.62** | **0.57** | **0.69** | **0.74** | **0.69** |

aggregated confusion matrix elements to compute the overall precision and recall. This allows us to avoid the problem of determining the average of multiple annotations as the ground truth.

We employ two approaches of computing the precisions and recalls:

- The frame-based metric: We simply use frames as the unit of evaluation. For example, a frame is considered a true positive if it belongs to segments of high importance in both the annotation and the computed result.
- The segment-based metric: Here the segments are the unit of evaluation. We follow the metric given in [29], where precision is the ratio of extracted segments with at least 50% overlap with the annotator-selected segments, and recall is the ratio of annotator-selected segments with at least 50% overlap with the extracted segments.

The results for these two metrics as well as the F1 scores are given in Tables 5 and 6, respectively. The evaluation includes only the segments with importance rated as High. Our method again performs better than the reference methods. The improvement of "Ours" results over "Ours_noDS" results is more evident than that in Table 3.

The results in Tables 5 and 6 are aggregated over all the seven videos. However, actual performances for different videos do vary quite significantly. To better understand the causes of the differences, we choose two videos with quite different performances and display in Fig. 10 their detailed results for qualitative analysis. There two videos are video no. 6 (the better one, with segment-based F1 scores of 0.92, 0.82, and 0.70 for Levels 1 ~ 3, respectively) and video no. 5 (the worse one, with segment-based F1 scores of 0.26, 0.46, and 0.46 for levels 1 ~ 3, respectively). We can see that it appears that video no. 6 contains two important segments that are quite well-defined, and there appears to be a high degree of agreement among the annotators (except for the last one). Our method is able to identify the same set of important segments in this case. On the other hand, the important segment in video no. 5 is less well defined since the annotators do not agree on that very well. The actual video content of the detected important segment is when the recorder stopped in a walk, in an outdoor scene, and took a photo with a smartphone. There are a lot motions (the recorder pulling out the phone and looking around to choose a view) without a change of scene, and this might be why the annotators do not agree well on the range of frames that constitute the important segment. Considering the results for Level 1, the important segment found by our method overlaps with the important segment found by four of the annotators. However, when computing the segment-based recall/precision metric, only for one of them (the last annotator) is our segment counted as a successful detection, and the other three (annotators 1, 2, and 5) are considered as missed detections because our segment covers less than 50% of the annotated important segments. This observation may indicate a need for more sophisticated evaluation metrics.

**Table 5** Comparison of segment extraction accuracy (Frame-based metric)

| | L1 | | | L2 | | | L3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | REC | PRE | F1 | REC | PRE | F1 | REC | PRE | F1 |
| Ours | 0.77 | 0.68 | 0.72 | 0.75 | 0.66 | 0.70 | 0.81 | 0.63 | 0.71 |
| Ours_noDS | 0.59 | 0.72 | 0.65 | 0.56 | 0.67 | 0.61 | 0.67 | 0.63 | 0.65 |
| SAL | 0.71 | 0.52 | 0.60 | 0.58 | 0.50 | 0.54 | 0.75 | 0.58 | 0.65 |
| IMM | 0.36 | 0.35 | 0.35 | 0.32 | 0.34 | 0.33 | 0.21 | 0.36 | 0.27 |
| AFF | 0.26 | 0.35 | 0.30 | 0.21 | 0.35 | 0.26 | 0.10 | 0.18 | 0.13 |

**Table 6** Comparison of segment extraction accuracy (Segment-based metric)

| | L1 | | | L2 | | | L3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | REC | PRE | F1 | REC | PRE | F1 | REC | PRE | F1 |
| Ours | 0.68 | 0.80 | 0.74 | 0.73 | 0.75 | 0.74 | 0.66 | 0.71 | 0.68 |
| Ours_noDS | 0.56 | 0.79 | 0.66 | 0.57 | 0.74 | 0.65 | 0.48 | 0.62 | 0.54 |
| SAL | 0.58 | 0.60 | 0.59 | 0.56 | 0.56 | 0.56 | 0.69 | 0.60 | 0.64 |
| IMM | 0.24 | 0.38 | 0.29 | 0.31 | 0.41 | 0.35 | 0.17 | 0.44 | 0.24 |
| AFF | 0.20 | 0.32 | 0.25 | 0.16 | 0.22 | 0.19 | 0.08 | 0.13 | 0.10 |

### 5.4 Effects of combining both $S_{imp}$ and $S_{cut}$

Currently, both $S_{imp}$ and $S_{cut}$ are utilized in the segmentation of a video and the extraction of its important segments. It is interesting to investigate their separate effects on these tasks. In this subsection, we repeat the experiments in Subsections 5.2 and 5.3 with two variations: The first is to use only $S_{imp}$, meaning that the second step in Subsection 4.4 is skipped and $S_{cut}$ has no effect at all. The second variation is to use only $S_{cut}$ to segment the video, skipping the first step in Subsection 4.4, and $S_{imp}$ values are only used to determine the importance ratings of the segments. The results are listed in Table 7, together with the results of the original method that uses both.

Regarding the ARI metric, the performance of using both $S_{imp}$ and $S_{cut}$ at levels 2 and 3 is somewhat better that of using only $S_{imp}$. This improvement results from the cases when $S_{cut}$ introduces cuts that separate segments of similar importance. On the other hand, using only $S_{cut}$ yields clearly worse numbers. When considering the metrics on importance segment extraction, using only $S_{imp}$ is no worse than, and sometimes even slightly better than using both. A likely reason is that the use of $S_{cut}$ (step 2 of Subsection 4.4) might unnecessarily divide an important segment. This observation seems to indicate that, when the sole purpose is to find important segments in a video, using only $S_{imp}$ is sufficient. This is consistent with event detection works such as [29], where only the per-frame event likelihood is used to generate the segments.


**Fig. 10** Example color-coded comparison of the computed segmentation and importance rating results with annotations. Results on two videos are shown here: video no.6 (*top*) and video no. 5 (*bottom*). Results and annotations of all three levels are displayed. For each level, the top plot is the result from our algorithm, and the other 7 plots are human annotations

**Table 7** Comparison of results with and without combining $S_{imp}$ and $S_{cut}$

| | Segmentation (ARI) | | | Segment extraction (Frame-based F1) | | | Segment extraction (Segment-based F1) | | |
|---|---|---|---|---|---|---|---|---|---|
| | L1 | L2 | L3 | L1 | L2 | L3 | L1 | L2 | L3 |
| Combined | 0.63 | 0.63 | 0.57 | 0.72 | 0.70 | 0.71 | 0.74 | 0.74 | 0.68 |
| $S_{imp}$ only | 0.63 | 0.57 | 0.50 | 0.72 | 0.70 | 0.71 | 0.74 | 0.77 | 0.73 |
| $S_{cut}$ only | 0.43 | 0.57 | 0.52 | 0.35 | 0.44 | 0.51 | 0.34 | 0.37 | 0.44 |

## 6 Conclusion

In sum, this paper describes a set of automatic methods to segment videos taken using smart glasses, and for generating importance ratings of the extracted segments. The results can be applied straightforwardly to the generation of a video summary by selecting and combining only the segments with importance ratings above some given threshold. We also present a systematic evaluation framework of these tasks, and the results indicate that our results are highly consistent with those given by the annotators. When compared with several existing approaches, the experimental results show that our method performs best at modeling the segmentation and importance rating of human viewers.

We believe that videos taken with wearable devices, such as smart glasses, provide a lot of new possibilities and challenges for video processing techniques. This work is just a beginning in this direction. So far, we have limited the videos to contain lives in a college campus in order to limit the complexity. A possible direction for future work is to expand the variety of videos, and to investigate the relations between the contexts and the suitable features for video segmentation and summarization, including the possibility of automatic or semi-automatic identification of the contexts, including recognition of scenes and activities. Another interesting subject is to extract information about people interactions and relations, using both verbal and visual information, from egocentric videos.

**Acknowledgements** This work is supported by the Ministry of Science and Technology of Taiwan under grant number MOST-104-3115-E-009-001.

## References

1. Abdollahian G, Taskiran CM, Pizlo Z, Delp EJ (2010) Camera motion-based analysis of user generated video. IEEE Transactions on Multimedia 12:28–41
2. Ajmal M, Ashraf MH, Shakir M, Abbas Y, Shah FA (2012) Video summarization: techniques and classification. In: LNCS, vol 7594, pp 1–13
3. Boreczhy JS, Wilcox LD (1998) A hidden Markov model framework for video segmentation using audio and image features. In: Proc. IEEE 1998 conference on acoustics, speech and signal processing, vol. 6, pp 3741–3744
4. Cheatle P (2004) Media content and type selection from always-on wearable video. Proc ICPR 4: 979–982
5. Cricri F, Dabov K, Curcio ID, Mate S, Gabbouj M (2014) Multimodal extraction of events and of information about the recording activity in user generated videos. Multimedia tools and applications 70: 119–158

6. Damnjanovic U, Fernandez V, Izquierdo E, Martínez JM (2008) Event detection and clustering for surveillance video summarization. In: Proc. 9th international workshop on image analysis for multimedia interactive services, pp. 63–66
7. Ferman M, Tekalp AM, Mehrotra R (2002) Robust color histogram descriptors for video segment retrieval and identification. IEEE Trans Image Process 11:497–508
8. Fujimura K, Honda K, Uehara K (2002) Automatic video summarization by using color and utterance information. In: Proc. ICME, pp 49–52
9. Han J (2009) Object segmentation from consumer videos: a unified framework based on visual attention. IEEE Trans Consum Electron 55:1597–1605
10. Hua XS, Lu L, Zhang HJ (2004) Optimization-based automated home video editing system. IEEE Transactions on circuits and systems for video technology 14:572–583
11. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2:193–218
12. Kannan R, Ghinea G, Swaminathan S, Kannaiyan S (2013) Improving video summarization based on user preferences. In: Fourth National Conference on computer vision, Pattern Recognition, Image Processing and Graphics, pp 1–4
13. Lee YJ, Ghosh J, Grauman K (2012) Discovering important people and objects for egocentric video summarization. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1346–1353
14. Li Y, Lee SH, Yeh CH, Kuo CC (2006) Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques. IEEE Signal Process Mag 23:79–89
15. Li Y, Fathi A, Rehg JM (2013) Learning to predict gaze in egocentric video. In: Proc. IEEE International Conference on Computer Vision, pp. 3216–3223
16. Lie WN, Lai CM (2004) News video summarization based on spatial and motion feature analysis. Advances in multimedia information processing – PCM 2004, LNCS 3332:246–255
17. Lienhart R (1999) Abstracting home video automatically. In: Proc. 7th ACM international conference on multimedia (part 2), pp 37–40
18. Lienhart R, Pfeiffer S, Effelsberg W (1997) Video abstracting. ACM Communications Magazine 40:54–62
19. Ma YF, Hua XS, Lu L, Zhang HJ (2005) A generic framework of user attention model and its application in video summarization. IEEE Transactions on Multimedia 7:907–919
20. Money AG, Agius H (2008) Video summarisation: a conceptual framework and survey of the state of the art. J Vis Commun Image Represent 19:121–143
21. Nagasaka A, Tanaka Y (1992) Automatic video indexing and full-video search for object appearances. In: Proceedings of the IFIP TC2/WG 2.6 second working conference on visual database systems II, pp 113–127
22. Nakamura Y, Ohde JY, Ohta Y (2000) Structuring personal activity records based on attention-analyzing videos from head mounted camera. Proc ICPR 4:222–225
23. [online] http://ggdiaries.com/
24. [Online] http://eshare.stust.edu.tw/EshareFile/2016_1/2016_1_3dca35cb.pptx
25. Peng WT, Chang CH, Chu WT, Huang WJ, Chou CN, Chang WY, Hung YP (2010) A real-time user interest meter and its applications in home video summarizing. In: Proc. 2010 I.E. international conference on multimedia and expo, pp 849–854
26. Poleg Y, Arora C, Peleg S (2014) Temporal segmentation of egocentric videos. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2537–2544
27. Rahtu E, Kannala J, Salo M, Heikkilä J (2010) Segmenting salient objects from images and videos. In: Proc. ECCV, pp. 366–379
28. Rallapalli S, Ganesan A, Chintalapudi K, Padmanabhan VN, Qiu L (2014) Enabling physical analytics in retail stores using smart glasses. In: Proc. 20th annual international conference on mobile computing and networking, pp. 115–126
29. Su YC, Grauman K (2016) Detecting engagement in egocentric video. Proc. ECCV, In, pp 454–471
30. Takahashi Y, Nitta N, Babaguchi N (2005) Video summarization for large sports video archives. In: Proc. ICME, pp. 1170–1173
31. Tse K, Wei J, Panchanathan S (1995) A scene change detection algorithm for MPEG compressed video sequences. In: Proc. IEEE 1995 Canadian conference on electrical and computer engineering, vol. 2, pp 827–830
32. Yamada K, Sugano Y, Okabe T, Sato Y, Sugimoto A, Hiraki K (2012) Attention prediction in egocentric video using motion and visual saliency. In: Proc. PSIVT, pp. 277–288
33. Zhang HJ, Low CY, Gong YH, Smoliar SW (1994) Video parsing using compressed data. In: Proc. SPIE, vol. 2182, pp 142–149
34. Zhang L, Xia Y, Mao K, Ma H, Shan Z (2015) An effective video summarization framework toward handheld devices. IEEE Trans Ind Electron 62:1309–1316
35. Zhu B, Liu W, Wei G, Yuan L (2014) A method for video synopsis based on multiple object tracking. In: Proc. 5th IEEE international conference on software engineering and service sciences, pp 414–418

**Yen-Chia Chiu** received his B.S. degree from National University of Kaohsiung, Taiwan, in 2014 and his M.S. degree in Multimedia Engineering from National Chiao Tung University, Taiwan, in 2016. He is currently an employee of Phison Electronics Corporation, Taiwan.

**Li-Yi Liu** received his B.S. degree from National Taipei University, Taiwan, in 2014. He is currently a master's student at the Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan.

**Tsaipei Wang** received the B.S. degree in physics from National Tsing Hua University, Taiwan, in 1989, the Ph.D. degree in physics from the University of Oregon in 1999, and the Ph.D. degree in computer engineering and computer science from the University of Missouri, Columbia, in 2005. He is currently an Associate Professor with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. His current research interests include computational intelligence and computer vision.