Active Video Summarization: Customized Summaries via On-line Interaction with the User

Ana Garcia del Molino,[†‡] **Xavier Boix,**[§¶] **Joo-Hwee Lim,**[†] **Ah-Hwee Tan**[‡]
[†]Institute for Infocomm Research, A*STAR, Singapore
[‡]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[§]LCSL, Massachusetts Institute of Technology and Istituto Italiano di Tecnologia, MA
[¶]Center for Brains, Minds, and Machines, McGovern Institute for Brain Research, Massachusetts Institute of Technology, MA
{stugdma,joohwee}@i2r.a-star.edu.sg, xboix@mit.edu, asahtan@ntu.edu.sg

Abstract

To facilitate the browsing of long videos, automatic video summarization provides an excerpt that represents its content. In the case of egocentric and consumer videos, due to their personal nature, adapting the summary to specific user's preferences is desirable. Current approaches to customizable video summarization obtain the user's preferences prior to the summarization process. As a result, the user needs to manually modify the summary to further meet the preferences. In this paper, we introduce Active Video Summarization (AVS), an interactive approach to gather the user's preferences while creating the summary. AVS asks questions about the summary to update it on-line until the user is satisfied. To minimize the interaction, the best segment to inquire next is inferred from the previous feedback. We evaluate AVS in the commonly used UTEgo dataset. We also introduce a new dataset for customized video summarization (CSumm) recorded with a Google Glass. The results show that AVS achieves an excellent compromise between usability and quality. In 41% of the videos, AVS is considered the best over all tested baselines, including summaries manually generated. Also, when looking for specific events in the video, AVS provides an average level of satisfaction higher than those of all other baselines after only six questions to the user.

1 Introduction

The emergence of compact and portable cameras in the consumer market has created the need for automatic and customizable summarization tools. Videos recorded with smartphones and wearable cameras are flooding social networks, and our lives are being recorded in a hands-free and non-intrusive way. As a result, video summarization tools will play a key role to facilitate sharing consumer videos in the near future, as these tools can save a considerable amount of resources to the user, *e.g.* time to create the summary, cost of sharing and keeping long videos, *etc.*

Many state-of-the-art summarization tools select video segments to include in the summary by optimizing a predefined criteria. Such criteria frequently relates to story coherence such as diversity and representativity (Dang and Radha 2014; Zhao and Xing 2014); interestingness from visual aesthetics, attention, importance to external human judges, *etc.* (Ma et al. 2005; Jiang, Cotton, and Loui 2011; Gygli et al. 2014; Potapov et al. 2014; Lin, Morariu, and Hsu 2015; Zhang et al. 2016; Chu, Song, and Jaimes 2015; Yao, Mei, and Rui 2016); or both (Ngo, Ma, and Zhang 2005; Lu and Grauman 2013; Gygli, Grabner, and Van Gool 2015).

Recently, however, several authors stressed the need to take into account the user's preferences, as the usability of video summarization can be significantly improved if customizing the summary. Furthermore, the data show that summaries generated by different people are not consistent between each other (Gygli et al. 2014). Thus, customization might be crucial to effectively summarize consumer videos, as these videos are inherently personal.

Previous work to customize video summaries obtains the user's preferences passively, by analyzing data that the user provided previously to the summarization. Several methods create a customized summary given a text query from the user (Yang et al. 2003; Varini, Serra, and Cucchiara 2015; Sharghi, Gong, and Shah 2016) or a set of video segments that match a set of user's preferences (Tseng and Smith 2003; Han, Hamm, and Sim 2011). Another strand of methods assesses each segment's interest from the user's behavior while watching that video (Peng et al. 2011), or similar videos (Masumitsu and Echigo 2000; Yoshitaka and Sawada 2012).

In this paper, we introduce Active Video Summarization (AVS), which enriches the set of user's preferences while creating the summary. AVS improves the usability of the aforementioned passive approaches, as these approaches are constrained by the initial feedback the user provided. A probabilistic model is used to infer both the customized summary and the next question to ask, which reduces the time the user needs to produce a summary.

We evaluate AVS on two challenging datasets for video summarization: UTEgo (Lee, Ghosh, and Grauman 2012), which is a commonly used egocentric video dataset, and CSumm, a new dataset for customizable video summarization that we introduce. CSumm contains single-shot unconstrained videos of long duration recorded with a Google Glass, which depict a varied range of events and daily life activities. The results show that the summaries generated with AVS exploit better the user's preferences than the state-of-the-art video summarization algorithms. Namely, AVS significantly reduces the time spent by the users to generate their preferred summary. With just six questions to the user, the average level of satisfaction for AVS is higher than those of all other tested algorithms. Also, in 41% of the tested cases, the users consider the summary obtained with AVS better than any other summary, including the summaries generated with manual tools.

2 General Overview of Active Video Summarization

The aim of AVS is to provide a customized summary with as little effort as possible from the user side. The system first asks for the user's initial preferences, selected from a set of items, *i.e.* the most frequent items in the original video. Then, the user's preferences are further refined through a question-asking inference.

AVS asks the user specific questions about segments of the video. It shows one selected segment, and asks the following two binary questions: *Q1: Would you want this segment to be in the final summary?*, and *Q2: Would you want to include similar segments?* Additionally, the user can decide at any time to go through the segments in the summary, and give such feedback about them. Although AVS is not limited to these two questions, experiments show that they are effective in practice, and they serve us as a proof of concept. Note that the original video is not shown to the user, as the segments shown during the interaction provide an accurate idea of the video content in much less time.

Thus, AVS can be divided into two inference problems: *(i)* infer the customized summary, and *(ii)* infer the next segment to show (Alg. 1). We use a probabilistic approach based on active inference in Conditional Random Fields (CRFs) (Roig et al. 2013). to infer the most likely summary, and to estimate the next question to ask. CRFs are sound probabilistic models that have been successfully applied in many computer vision and multimedia problems (Lafferty, McCallum, and Pereira 2001). In the following, we introduce CRFs to infer the customized summary, and then, the algorithm that infers the segments to show.

3 Inference of the Customized Summary

Let $\mathbf{s} = \{s_i\}$ be the set of random variables that represent the summary of the video by indicating whether a segment (or subshot) of the video appears in the summary. Thus, $s_i \in \{0, 1\}$, where s_i is equal to 1 when the segment is included in the summary, and 0 otherwise. We denote $P(\mathbf{s}|\boldsymbol{\theta})$ as the probability density distribution of how likely the summary s is preferred by the user. We model this distribution with a CRF, and $\boldsymbol{\theta}$ are the values of its parameters, that depend on the input video and the user's preferences.

A CRF models the probability density with a Gibbs distribution, *c.f.* (V. and Wainwright 2005). Therefore, $P(\mathbf{s}|\boldsymbol{\theta})$ can be written as the normalized exponential of an energy function, which is denoted as $E_{\boldsymbol{\theta}}(\mathbf{s})$. The energy function is the sum of a set of potentials, which are functions that take as input a subset of $\{s_i\}$. The summary of the video, which is denoted as $\mathbf{s}^\star_{\boldsymbol{\theta}}$, is obtained by inferring the Maximum a Posteriori (MAP), *i.e.* $\mathbf{s}^\star_{\boldsymbol{\theta}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\boldsymbol{\theta})$, or equivalently, maximizing the energy function $E_{\boldsymbol{\theta}}(\mathbf{s})$.

In the following, we first introduce the potentials of the CRF, and then the algorithm to obtain the MAP summary.

3.1 CRF for Customized Summarization

We follow most methods in the literature, that select representative and diverse segments with as little motion as possible. To do so, we define the energy function of the CRF as

$$E_{\boldsymbol{\theta}}(\mathbf{s}) = \lambda \sum_i \underbrace{\phi_u(s_i)}_{\text{unary}} + \sum_{ij} \underbrace{\phi_p(s_i, s_j)}_{\text{pairwise}}, \qquad (1)$$

where the unary potentials enforce the selection of static segments, the pairwise potentials encourage segments with diverse semantic content, and λ is a parameter that weights the unary potentials with respect to the pairwise. There is a unary potential for each segment of the video, and one pairwise potential for each pair of similar segments. The length of the summary is controlled during the inference of the MAP summary by adding additional constraints to the energy function that control the length of the summary, as we show below.

Next, we introduce the potentials, and we make emphasis on the update of the potentials when new user's preferences are known. Note that we omit the dependency of the potentials on $\boldsymbol{\theta}$ for simplicity, and the parameters that we introduce in the following should be considered as part of $\boldsymbol{\theta}$. Also, the values of the parameters of the potentials are introduced in the implementation details in Sec. 5.2.

Unary Potentials. The unary potentials, $\{\phi_u(s_i)\}$, encourage selecting segments that the user will probably like. $\phi_u(s_i)$ is equal to $Q_i\mathbf{I}[s_i = 1] + L\mathbf{I}[s_i = 0]$, in which: $\mathbf{I}[a]$ is an indicator function that is 1 if a is true and 0 otherwise; Q_i is a function representing how well that segment relates to the requirements individually; and L is a constant offset that is set during the MAP inference of the summary in order to adjust the summary length (Sec. 3.2).

During the on-line interaction phase, when the user recommends to include a segment s_i (an affirmative response to *Q1*), Q_i is increased by Δ to enforce the selection of that segment; otherwise Q_i is decreased by Δ.

Pairwise Potentials. The pairwise potentials, $\{\phi_p(s_i, s_j)\}$, are defined between each pair of similar segments, and enforce selecting segments with diverse content.

Let $d(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)$ be the Euclidean distance between the descriptors of two segments (details in Sec. 5.2). The pairwise potential enforces that similar segments should not be included in the summary. To do so, we define a potential that is weighted by the distance between descriptors, *i.e.* $\phi_p(s_i, s_j) = \exp\left(-d(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)\right)\phi'_p(s_i, s_j)$, in which $\phi'_p(s_i, s_j)$ enforces that both segments should not be selected at the same time, and the term $\exp\left(-d(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)\right)$ reduces the effect of $\phi'_p(s_i, s_j)$ when the segments are dissimilar. In this way, only a representative segment among similar segments is selected.

Specifically, $\phi'_p(s_i, s_j)$ is defined as

$$\phi'_p(s_i, s_j) = \left\{ \begin{array}{rl} L\alpha & \text{if } s_i = s_j = 0 \\ -L\beta & \text{if } s_i = s_j = 1 \\ \gamma & \text{if } s_i \neq s_j \end{array} \right. , \qquad (2)$$

where γ is the cost of selecting only one segment in the pair, α and β are the cost to discard or select both segments, respectively, and L is a variable parameter that controls the length of the summary. Note that when γ, α, and β are positive, the negative sign of the case $s_i = s_j = 1$ implies that selecting two similar segments is discouraged for big values of β.

When new user's preferences are available, we update all the pairwise terms in which the segment in question appears. When the user recommends selecting either the segment (*Q1*) or a similar segment (*Q2*), but not both, γ is multiplied by a $K > 1$ to encourage that one of the two segments in the pair is selected. On the opposite case, if the user recommends discarding or selecting both segments at the same time, we multiply γ by $-K$ to penalize selecting one of them, *i.e.* it penalizes $x_i \neq x_j$. Additionally, if the user recommends selecting both the segment and similar ones, β is enlarged by $-K$, to cancel the negative sign in Eq. (2) and allow that multiple similar segments are selected.

3.2 MAP Inference of the Summary

There are many off-the-shelf algorithms to obtain the MAP summary from the CRF with the energy function we introduced in Eq. (1). We use the implementation of Belief Propagation (BP) (Yedidia, Freeman, and Weiss 2005) implemented by Boykov and Kolmogorov (2004), using a maximum of five iterations.

The summary is generated using a line search algorithm that optimizes the values of L and λ to yield the desired summary duration and balance between visual quality (unary potentials) and diversity (pairwise potentials). Recall that the parameter L encourages excluding segments from the summary when $L > 1$: $s_i = s_j = 0$ is further encouraged and $s_i = s_j = 1$ is further penalized (due to the negative sign). Thus, when L is increased, the summary is shorter; otherwise it is longer. Additionally, the parameter λ is increased when the segments selected do not meet the minimum quality criteria or to better meet the initial requirements, and decreased to facilitate diverse content.

4 Inference on the Next Segment to Show

The formulation with the CRF that we have introduced in the previous section yields the following flow of the algorithm. Initially, the values of the CRF potentials are $\boldsymbol{\theta}_1$, which are estimated from the input video. Then, the summary $\mathbf{s}^\star_{\boldsymbol{\theta}_1}$ (MAP summary) is shown to the user. The algorithm selects a segment to query, and the values are updated, $\boldsymbol{\theta}_2$, to match the user's answer. Thus, after the t-th answer, the potential's values are $\boldsymbol{\theta}_{t+1}$.

We now introduce the inference on the next segment to query. AVS ranks all possible questions with a score, and asks for the one with the highest rank. Let S_k be the score used to rank the k-th candidate segment. Following the dynamic programming formulation (Bellman 1952), the score is based on a reward function that evaluates the change produced in the summary given the answer of the user, *i.e.* it compares $\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}$ to $\mathbf{s}^\star_{\boldsymbol{\theta}_t}$. Since the reward is obtained after an answer of the user, the algorithm can only estimate the expected reward to decide the candidate to query. Thus, the score S_k, is obtained evaluating the expected reward for the k-th candidate.

Alg. 1: Active Summarization

```
$\boldsymbol{\theta}_1 \leftarrow$ initialization from the video
$t = 1$
while user does not stop the loop do
    ▷ Compute customized summary:
    $\mathbf{s}^\star_{\boldsymbol{\theta}_t} = \arg\max_{\mathbf{s}} E_{\boldsymbol{\theta}_t}(\mathbf{s})$
    Display $\mathbf{s}^\star_{\boldsymbol{\theta}_t}$
    ▷ Compute the reward of asking about each candidate:
    forall candidate segments do
        $S_k = \boldsymbol{E}_{\boldsymbol{\theta}_{t+1}}\left[R\left(\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}, \mathbf{s}^\star_{\boldsymbol{\theta}_t}\right) \mid k\text{-th candidate}\right]$
    end
    ▷ Ask about the candidate with the highest $S_k$:
    if users wants to review summary then
        Ask questions about segments in $\mathbf{s}^\star_{\boldsymbol{\theta}_t}$
    else
        $k^\star = \arg\max_k S_k$
        Ask about $k^\star$-th candidate
    end
    $\boldsymbol{\theta}_{t+1} \leftarrow$ adapt from user's answer
    $t = t + 1$
end
```

We use $R(\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}, \mathbf{s}^\star_{\boldsymbol{\theta}_t})$ to denote the reward function, that compares the future summary $\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}$ to $\mathbf{s}^\star_{\boldsymbol{\theta}_t}$. Since we want to prioritize the questions that may yield the largest changes in the summary, we define $R(\cdot,\cdot)$ as the Kendall τ correlation between $\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}$ and $\mathbf{s}^\star_{\boldsymbol{\theta}_t}$ (Deza and Deza 2009).

Also, we only evaluate the expected reward for the next candidate by discarding the reward of future segment queries that are not the next one. Thus, we define S_k as

$$S_k = \boldsymbol{E}_{\boldsymbol{\theta}_{t+1}}\left[R\left(\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}, \mathbf{s}^\star_{\boldsymbol{\theta}_t}\right) \mid k\text{-th question}\right], \tag{3}$$

where the expectation is over all possible answers to querying the k-th candidate, and $\mathbf{s}^\star_{\boldsymbol{\theta}_{t+1}}$ is the MAP summary for an user's answer of the k-th candidate.

Note that to compute the expected value in Eq. 3 we need an estimate of the probability of the user's answers. We can estimate this probability using BP (Sec. 3.2). BP obtains the MAP summary by approximating the marginals of the Gibbs distribution, *i.e.* BP approximates $\{P(s_i|\boldsymbol{\theta})\}$ and $\{P(s_i, s_j|\boldsymbol{\theta})\}$, and then, it takes the s_i's that maximizes $P(s_i|\boldsymbol{\theta})$, independently from the other segments, *c.f.* (Yedidia, Freeman, and Weiss 2005). Thus, we can take the marginals estimated by BP to compute the probability of the user's answers. Note that $\{P(s_i|\boldsymbol{\theta})\}$ is the probability that the user recommends the i-th segment to be included in the summary (an affirmative response to *Q1*). Also, we can estimate the probability that the user will recommend to include similar segments (*Q2*) by averaging the pairwise marginals, $\{P(s_i, s_j|\boldsymbol{\theta})\}$, that refer to the segments similar to s_i.

5 Experiments

In this section, we report results both on a new dataset for customized video summarization and UTEgo (Lee, Ghosh, and Grauman 2012). After introducing the new dataset, and implementation details, we report the results of AVS.

5.1 Datasets (CSumm and UTEgo)

Since current public datasets that provide annotations of the summary contain 1 to 5 minutes videos (*e.g.* SumMe (Gygli et al. 2014), MED (Potapov et al. 2014)), and video summarization is of most use for longer videos, we have obtained annotations for 10 shots of 15 to 30 minutes that we recorded with a Google Glass (29 fps, with resolution of 720×1280 pixels). The videos include a large selection of activities, such as practicing or watching sports, enjoying nature, having dinner, *etc.*

The videos are unconstrained, and include a wide range of viewpoints and motion, as they are first person view, and a large amount of irrelevant moments alongside the recording. This makes our dataset challenging for video summarization, which is supported by our results below. In the Suppl. Material we show several summaries of CSumm.

Additionally, we report results on UTEgo (Lee, Ghosh, and Grauman 2012), an egocentric video dataset commonly used for the evaluation task. Recorded with a Looxcie camera at 15fps and a resolution of 320×480 pixels, this dataset contains four long videos (three to five hours) of daily activities such as cooking, shopping, eating and driving. We have divided three of these videos into two parts, obtaining a total of seven videos of two hours or longer.

5.2 Implementation Details

We now introduce the specifics of our implementation, and the values of the different constants. These values have been manually set during development, prior to the studies with the subjects.

Video Segmentation The subshot boundaries used for the summarization are estimated with the motion status and changes on the environment. In CSumm, these are obtained through the gyroscope from Google Glass to infer motion, and the illumination sensor to identify abrupt changes in the lighting condition. Each segment is set to be around 2.5 seconds long, and its boundaries to match a change in illumination or motion pattern. In UTEgo, since the sensor data is not available, segments are equally set to be around 2.5 seconds long, with its boundaries matching a change in the image overall illumination, obtained from a quantization of the image mean intensity.

Segment Descriptors The frame descriptors, $\boldsymbol{\psi}_i$, are based on the output of a neural network for object recognition, extracted for each frame. Specifically, we use the last layer from AlexNet (Krizhevsky, Sutskever, and Hinton 2012)) trained in Places dataset (Zhou et al. 2014) and in ImageNet (Russakovsky et al. 2015). We concatenate the output of the neural network for the categories of objects (including animals) and places. Finally, we average the value for each item along all the frames in the video segment.

Unary Potentials Recall that Q_i represents the quality of the video segment for the user. Initially and by default, Q_i depends on the motion and blur. Q_i is inversely proportional to the amount of motion in the segment, which is estimated from a blur detector in UTEgo, and from the gyroscope in CSumm. Q_i is normalized to take values in $[0, 1]$.

Additional passive preferences can be added by the user beforehand. Such preferences are included in the model as constraints for this potential. Before starting AVS, we show the user the list of top ranked *objects* and *places* categories in the video (*i.e.* the categories with higher accumulated activation), and the user can select among them the relevant items and the irrelevant ones. Then, Q_i is increased or decreased, respectively, depending on the activation value of such items in segment i. This is done by multiplying Q_i by $1+\sum_{j\in\text{relevant}} \psi_i(j) - \sum_{j\in\text{irrelevant}} \psi_i(j)$, in which $\psi_i(j)$ is the output of the neural network for the category indexed with j.

In the active interaction phase, Q_i is increased or decreased by Δ, which is set to 100 to ensure that the segments selected by the user appear in the summary, and the discarded do not.

Pairwise Potentials To enforce representativeness of the segments in the summary, we set $\alpha = 5$, $\beta = 1$ and $\gamma = 1$. We can observe by analyzing Eq. (2) that these parameters penalize selecting both segments ($\beta = 1$, and the negative sign). Also, these parameters encourage that both segments are discarded ($\alpha = 5$), or that only one of them is selected ($\gamma = 1$). Note that α is bigger than γ because in most cases the pair of segments in the pairwise potential should be discarded, as only few segments should be selected in the final summary. In the active interaction phase, the multiplier K is set to 5.

To reduce the computational cost of the MAP inference algorithm, we discard the pairwise potentials with smallest influence. Specifically, we discard 30% of the pairwise potentials that encode the largest distances between segments.

Duration of the Summary The duration is variable depending on the length of the original video. It is set to be around 0.1% of the video length, with a minimum of 10 seconds.

Baselines We compare AVS to the following baselines:

- *Uniform*: Summary from uniformly sampled segments.
- *VMMR*: Video Maximal Marginal Relevance, a summarization method which rewards diversity (Li and Merialdo 2010), executed using our deep features.
- Lee *et al.* : Summary extracted with the method presented by Lee, Ghosh, and Grauman (2012). Since this approach obtains a set of key-frames, we have mapped each keyframe to its corresponding segment to obtain the video summary. These summaries are available in UTEgo.
- *Manual*: In CSumm, where results by Lee, Ghosh, and Grauman (2012) are not available, we have replaced such baseline for a manual annotation of the best segments, with a length constraint of 10 seconds. This was performed by two independent subjects (who did not par-

ticipate in the rest of the user study), which were asked to manually summarize the given video to their own liking. The annotation to use as baseline is chosen at random among both.

Additionally, the efficiency of the inference on the segment to query in AVS is compared to that of a random selection of segments (referred as *random*).

5.3 Evaluation methodology

We analyze two scenarios in which AVS can be used in practice. In the first scenario, the user has to summarize a video never seen before. The user has no knowledge of the video essence, and thus does not know yet what are the relevant parts. AVS allows the user to discover his or her own preferences while exploring the video content.

In the second scenario, the user already knows the content of the video (*e.g.* the user was the camera wearer), and already knows his or her preferences. However, due to the length of the original video, looking for such preferences in the video is very time consuming. AVS allows for the user to browse the video and find such events easier and faster. In the following, we provide the details for the conducted user study, related to both scenarios.

Scenario 1: Discovery Task We asked 30 independent participants to summarize two videos they had never seen before. They were given no constrains as to what had to be seen in the summary, other than whatever they were interested in. Then, they were asked to rate how good was that summary, by answering the question *"Did the system manage to provide your ideal summary for that video?"* with a scale of 1 (*"Not at all"*) to 5 (*"Absolutely"*).

To validate their responses on a semi-blind setting, a week after the experiment we asked them to compare the quality of the different baseline summaries. For the two videos the subject has summarized, we showed the summaries generated with the baselines, and the ones that the subject generated. Then, the subject assessed the quality of the summaries by ranking each of them using one of the following tags: *best*, *good*, *acceptable*, *bad*, and *worst*. We asked them to rate at least one as *worst* and one as *best*. The subjects did not know the baseline corresponding to each summary, and the order was randomized among trials. Note that more than one summary can be rated with the same label, so that there may be more than one *best* or *worst*, if these seem to be equally good or bad.

Scenario 2: Search Task To evaluate the efficiency of AVS, the same participants were asked to find a set of events in 2 videos. Such preferences are given in the form of key-frames, extracted from the original video, and a text description of what needs to be included in the final summary (an example can be seen in Fig. 1).

To do so, we asked three independent subjects (that do not participate in the user study) to agree in the selection of four frames from each original video. This set of four key-frames is then used as guidance and scoring reference to summarize the video. In the user study, each user is asked to generate a summary which includes the four given events or items.

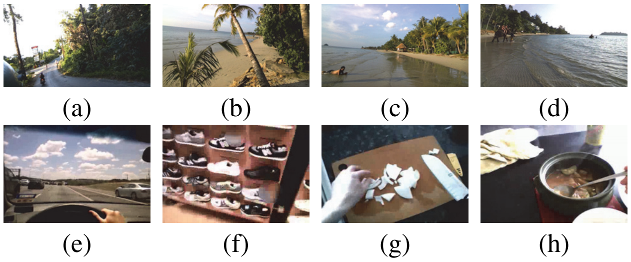


Figure 1: Items to be found in Scenario 2 for an example video. CSumm: (a) Gas station by the road. (b) Beach viewed from the road. (c) Man lying at the shore. (d) Elephants in the water. UTEgo: (e) Driving in highway. (f) Shoe shopping. (g) Chopping vegetables. (h) Serving food.

	CSumm				UTEgo			
	Unif.	Annot.	VMMR	AVS	Unif.	CVPR	VMMR	AVS
Unif.	-	28%	44%	25%	-	29%	41%	24%
An./CV.	66%	-	78%	**50%**	59%	-	71%	41%
VMMR	47%	19%	-	19%	47%	24%	-	24%
AVS	**59%**	34%	**66%**	-	**71%**	**53%**	**76%**	-

Table 1: Percentage of times each method on the left was ranked better than the one on top for the Discovery task (scenario 1). Note that symmetric elements may not add up to 100%, since two summaries can be ranked equally.

The subjects perform this task twice, one time with AVS and the other with AVS with *random* questions. None of them knew anything about AVS during the experiment. The subjects also ignored whether they were using AVS, by randomly changing the order of the algorithms.

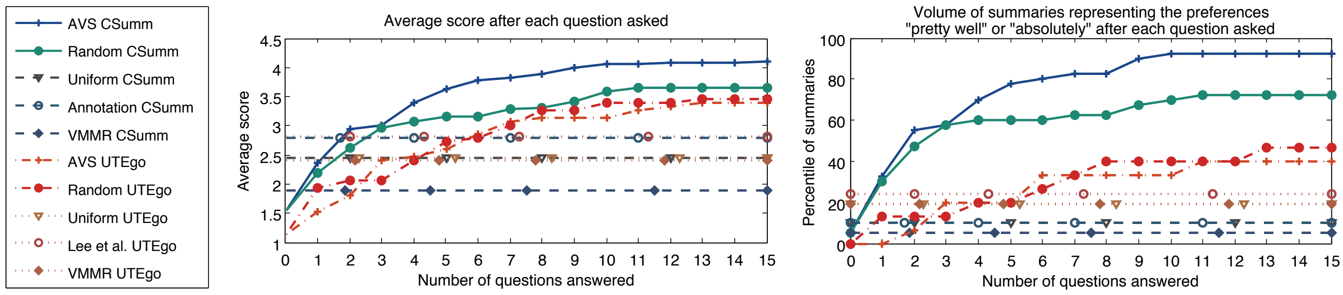
At the end of each summarization, we asked the users to rate how well the final summary represented the given constraints, on a scale from 1, none or only one of the events is found, to 5, all the given constraints are perfectly included in the summary. This experiment also allows obtaining an objective measurement from the amount of interaction needed to reach the target summary. Once performed the summarization task with both approaches, we asked the users to rate the usability of one approach against the other.

Finally, to obtain a blind test against the baselines, the user is also asked to rate the summary that another user generated, and the baseline summaries, using the same scale and criteria as used in his or her summaries.

5.4 Results

The customization potential of AVS is evaluated through the quality of the final summary and the usability of the system, using the data and feedback obtained from the user study (Sec. 5.3). Examples of the summaries can be found in the supplementary material.

Quality of the Summary Table 1 describes the percentage of times that the subjects have ranked a summary better than another summary in the discovery task (Scenario 1). We



(a) (b) (c)

Figure 2: Evaluation of the summary after each question in the Search task (scenario 2). The score is given by the answer to the question *"Did the system manage to provide your ideal summary for that video?"*, which accepted as responses *"Not at all"* (1), *"Not much'* (2), *"So-so'* (3), *"Pretty much'* (4) *"Absolutely'* (5). (a) Legend. (b) Mean score of the summaries. (c) Percentage of summaries for which the summary obtained a score greater or equal to 4. Figure best viewed in color.

can see that AVS is largely preferred over two of the tested baselines, *uniform* and *VMMR* for both datasets. For half of the summaries in our dataset, AVS is preferred or equally preferred to *manual* annotation. In UTEgo, AVS is also preferred to the method by Lee, Ghosh, and Grauman (2012).

Note that the comparison between *manual* and *uniform* in CSumm shows that in not all cases the subjects prefer the *manual* summary over *uniform*. This shows that the summarization of the videos in CSumm is highly subjective, as a subject may prefer the *uniform* summary over a *manual* annotation from another person (recall that the subject that is assessing the *manual* summary is not the author of this summary, but of the summary with AVS). This proves the challenging nature of CSumm, and it gives more reassurance that the inference of the user's preferences is a key component for video summarization.

When searching for specific events (Scenario 2), Fig. 2 reports the percentage of users satisfied after each question answered. We observe that the AVS summary after two questions is rated better than any of the baselines for CSumm. For the videos of UTEgo, the user needs 6 questions to reach this level of satisfaction. Thus, with small interaction with the user, AVS achieves better results than any of the baselines.

However, we observe that for UTEgo, AVS obtains only slightly better performance than AVS with *random* questions. We investigated this, and we found that the performance of AVS highly depends on the image quality of the input data. UTEgo has a resolution of 320×480px (more than four times inferior to the 720×1280px of CSumm, recorded with a Google Glass). As a consequence, the descriptors extracted with neural networks results in an almost flat output vector, making it difficult for AVS to distinguish among the different events.

Usability We compare the time needed to generate a customized summary with AVS and *manually* (only in CSumm, as we did not obtain the manual annotation for UTEgo) in the discovery task (Scenario 1). In Table 2, we can see that the users are 4 times faster creating the summary with AVS than with the *manual* baseline. This is a significant improvement of the usability of the *manual* annotation, since the quality of AVS is competitive with the quality of the *manual* annotation as shown before.

AVS	Manual
5.89 ± 3.85 min.	21.66 ± 6.59 min.

Table 2: Time to generate a summary in CSumm.

	Much worse	Worse	Similar	Better	Much better
CSumm:	5.4%	16.2%	18.9%	43.2%	16.2%
UTEgo:	6.7%	13.3%	26.7%	40%	13.3%

Table 3: Subjective perception of the usability of AVS against the random baseline: amount of summaries that obtained each possible score.

Looking for specific events (Scenario 2), we can compare AVS and AVS with *random* questioning under the same search constraints. We show such subjective assessment in Table 3. We can see that the majority of the subjects prefer active inference over the *random* baseline in both datasets, which is in accordance with Fig. 2. These results demonstrate the usefulness of estimating the next questions to ask, as opposing to selecting *random* segments.

6 Conclusions

We presented Active Video Summarization (AVS), which is an approach to interact with the user to customize a video summary based on Conditional Random Fields. To evaluate our approach, we introduced a challenging dataset for customizable summaries of consumer videos, which we called CSumm. In a series of experiments, we have demonstrated that AVS strikes a balance between usability and quality of the summary.

In the future, the user's previously generated summaries will be used to learn his or her preferences. The summaries will also be used to learn to better interact with the user. A component we are investigating to further improve the usability of AVS is a rich set of questions to ask the user —including relations among the semantic content and the human actions in the video.

Acknowledgments

This material is based upon work supported by the CBMM, funded by NSF STC award CCF-1231216; The MOE AcRF Tier 2 grant R-263-000-B32-112; the Singapore DIRP grant 9014100596; A*STAR JCO REVIVE project grant 1335h00098; the Singapore International Graduate Award (SINGA); and the Obra Social "La Caixa" and Casa Asia Fellowship.

References

Bellman, R. 1952. On the theory of dynamic programming. *Proc. Natl. Acad. Sci.* 38(8):716–719.

Boykov, Y., and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(9):1124–1137.

Chu, W.-S.; Song, Y.; and Jaimes, A. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*. IEEE. 3584–3592.

Dang, C. T., and Radha, H. 2014. Heterogeneity image patch index and its application to consumer video summarization. *IEEE Trans. Image Process.* 23(6):2704–2718.

Deza, M. M., and Deza, E. 2009. Encyclopedia of distances. In *Encyclopedia of Distances*. Springer. 1–583.

Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating summaries from user videos. In *ECCV*. Springer. 505–520.

Gygli, M.; Grabner, H.; and Van Gool, L. 2015. Video summarization by learning submodular mixtures of objectives. In *CVPR*. IEEE. 3090–3098.

Han, B.; Hamm, J.; and Sim, J. 2011. Personalized video summarization with human in the loop. In *WACV*. IEEE. 51–57.

Jiang, W.; Cotton, C.; and Loui, A. C. 2011. Automatic consumer video summarization by audio and visual analysis. In *ICME*. IEEE. 1–6.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. 1097–1105.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*. Vol. 1, pp. 282–289

Lee, Y. J.; Ghosh, J.; and Grauman, K. 2012. Discovering important people and objects for egocentric video summarization. In *CVPR*. IEEE. Vol. 2, No. 6, p. 7.

Li, Y., and Merialdo, B. 2010. Multi-video summarization based on video-mmr. In *WIAMIS*. IEEE. 1–4.

Lin, Y.-L.; Morariu, V.; and Hsu, W. 2015. Summarizing while recording: Context-based highlight detection for egocentric videos. In *ICCVW*. IEEE. 51–59.

Lu, Z., and Grauman, K. 2013. Story-driven summarization for egocentric video. In *CVPR*. IEEE. 2714–2721.

Ma, Y.-F.; Hua, X.-S.; Lu, L.; and Zhan, H.-J. 2005. A generic framework of user attention model and its application in video summarization. *IEEE Trans. Multimedia* 7(5):907–919.

Masumitsu, K., and Echigo, T. 2000. Video summarization using reinforcement learning in eigenspace. In *ICIP*. IEEE. Vol. 2, pp. 267–270.

Ngo, C.-W.; Ma, Y.-F.; and Zhang, H.-J. 2005. Video summarization and scene detection by graph modeling. *IEEE Trans. Circuits Syst. Video Technol.* 15(2):296–305.

Peng, W.-T.; Chu, W.-T.; Chang, C.-H.; Chou, C.-N.; Huang, W.-J.; Chang, W.-Y.; and Hung, Y.-P. 2011. Editing by viewing: automatic home video summarization by viewing behavior analysis. *IEEE Trans. Multimedia* 13(3):539–550.

Potapov, D.; Douze, M.; Harchaoui, Z.; and Schmid, C. 2014. Category-specific video summarization. In *ECCV*. Springer. 540–555.

Roig, G.; Boix, X.; De Nijs, R.; Ramos, S.; Kuhnlenz, K.; and Van Gool, L. 2013. Active map inference in CRFs for efficient semantic segmentation. In *ICCV*. IEEE. 2312–2319.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* 115(3):211–252.

Sharghi, A.; Gong, B.; and Shah, M. 2016. Query-focused extractive video summarization. In *ECCV*. Springer. 3–19.

Tseng, B. L., and Smith, J. R. 2003. Hierarchical video summarization based on context clustering. In *ITCom*, 14–25. International Society for Optics and Photonics.

V., K., and Wainwright, M. J. 2005. On optimality properties of tree-reweighted message-passing. In *UAI*.

Varini, P.; Serra, G.; and Cucchiara, R. 2015. Egocentric video summarization of cultural tour based on user preferences. In *ACMMM*. ACM. 931–934.

Yang, H.; Chaisorn, L.; Zhao, Y.; Neo, S.-Y.; and Chua, T.-S. 2003. Videoqa: question answering on news video. In *ACMMM*. ACM. 632–641.

Yao, T.; Mei, T.; and Rui, Y. 2016. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*. IEEE.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2005. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inf. Theory* 51(7):2282–2312.

Yoshitaka, A., and Sawada, K. 2012. Personalized video summarization based on behavior of viewer. In *SITIS*. IEEE. 661–667.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016. Summary transfer: Exemplar-based subset selection for video summarization. *arXiv preprint arXiv:1603.03369*.

Zhao, B., and Xing, E. 2014. Quasi real-time summarization for consumer videos. In *CVPR*. IEEE. 2513–2520.

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using places database. In *NIPS*. 487–495.