Video Summarization by Learning Submodular Mixtures of Objectives

Michael Gygli[1] Helmut Grabner[1] Luc Van Gool[1,2]
[1]Computer Vision Lab, ETH Zurich, Switzerland [2]PSI - VISICS, K.U. Leuven, Belgium
{gygli, grabner, vangool}@vision.ee.ethz.ch

Abstract

We present a novel method for summarizing raw, casually captured videos. The objective is to create a short summary that still conveys the story. It should thus be both, interesting and representative for the input video. Previous methods often used simplified assumptions and only optimized for one of these goals. Alternatively, they used hand-defined objectives that were optimized sequentially by making consecutive hard decisions. This limits their use to a particular setting. Instead, we introduce a new method that (i) uses a supervised approach in order to learn the importance of global characteristics of a summary and (ii) jointly optimizes for multiple objectives and thus creates summaries that posses multiple properties of a good summary. Experiments on two challenging and very diverse datasets demonstrate the effectiveness of our method, where we outperform or match current state-of-the-art.

1. Introduction

With the success of mobile phones, activity cameras, Google Glass, etc. video recording devices have become omnipresent. As a consequence, vast amounts of videos are recorded every day to capture special moments or log daily activities. At the same time, with video capture becoming so easy and cheap, and with the strongly egocentric viewpoints that the devices often induce, videos are recorded casually. As in digital photography, many users follow a *capture first, filter later* mentality, where little thought is spent on timing, cutting, content and view selection. As a result, such casual videos are too long, shaky, redundant and low-paced to watch in their entirety. Therefore, reducing videos to their gist and removing bad parts is of increasing importance. As a result, video summarization, which automates this process, has gained a lot of attention in the last few years [33, 9, 29, 15, 11, 24, 1, 12, 39].

Automatically creating *skims* is challenging, as even a strongly shortened version should still convey the *story* of the initial video. A good summary must comply with at least two objectives [35]. Firstly, it should contain the most interesting parts of a video *e.g.* in a base jumping video one doesn't want to miss highlights such as the start or landing. Secondly, the summary should be representative in keeping the diversity of the original, while removing redundancy.

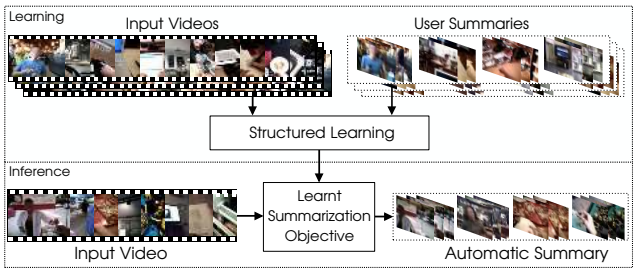

Figure 1: **Overview.** Our method consists of two parts: A supervised learning stage (training) and inference (testing). Given pairs of videos and their user created summaries as training examples, we learn a combined objective. Then, when given a new video as input, our method creates summaries that are both interesting and representative.

Many recent methods predict a score per segment and ignore the structure of the video [33, 9, 29], and therefore have difficulties to jointly optimize both objectives. Methods that go in this direction typically cluster the video into events and select the most important segment(s) per event [15, 11], following a kind of successive optimization of the objectives. Others optimize diversity only locally using a Markov assumption [6]. Instead, our method optimizes for multiple objectives *globally*, avoiding hard decisions early on. Rather than using supervision only for some components [15] or making simplifying assumptions [33, 9, 29], our method learns the importance of summarization objectives directly from reference summaries created by human annotators, as depicted in Fig. 1. Using supervision for the task of video summarization is crucial, since it is extremely complex and highly task-dependent – summaries from surveillance or live-logging data are expected to meet different criteria than summaries of short clips obtained by a mobile phone. Our approach is able to automatically adapt to the type of video and the desired output. It is therefore much more general and can be applied in

all of these settings. Indeed, our experiments show that our method obtains state of the art performance in summarizing hour long life-logging videos [15], as well as short user videos [9].

2. Related Work

Videos can be summarized into many different representations: Keyframes [37, 15, 11, 12], skims [9, 24], storyboards [4], time-lapses [13], montages [34] or video synopses [30]. Here, we focus on approaches for generating and evaluating *skims* (dynamic video summaries)[1], *i.e.* methods that output a shortened version of the initial video, rather than transforming the video into *e.g.* a collection of images. Skims have the advantage that they retain motion information and can provide a nice viewing experience. Following Truong and Venkatesh [35], we review related work categorized into methods optimizing for (i) the preservation of interesting segments and (ii) representativeness of the summary. Further, we (iii) analyze methods optimizing for multiple objectives.

Interestingness/relevance. In order to select keyframes or segments for a summary, many methods predict the importance score for each keyframe or segment. This is typically formulated as a regression (*e.g.* [9]) or ranking problem (*e.g.* [33]). Thereby some features are extracted from a video segment, in order to predict its relevance. For this, Potapov *et al.* [29] use videos annotated for a certain event category. Instead, Sun *et al.* [33] mine YouTube videos in order to train their model. Thereby they use the correspondence between the raw and edited version of a video in order to obtain labels for training. This is based on the assumption, that segments contained in the edited version are more relevant than the ones that are not. Both of these method are however not evaluated in terms of summary quality, but rather in terms of their ability to detect the highlight segment [33] or the most relevant segments for a certain category [29], criteria for which the overall structure of the video and the summary does play no role.

Representativeness. While optimizing for interestingness ignores the global structure of a summary, optimizing for representativeness only risks leaving out the most crucial event(s). Therefore only a few approaches in this area exist. Li and Merialdo [18] adapt the Maximal Marginal Relevance (MMR) approach [2] from the text to the video domain. This approach greedily selects a summary using an objective that optimizes for relevance w.r.t. the input video and penalizes redundancy within the summary. [39] uses sparse coding, in order to create a dictionary that serves as a summary. This method is particularly useful for longer videos, as it can be run in an online fashion.

[1]For a systematic and detailed review of the existing techniques, the readers are referred to [35].

Multi-objective. Several methods optimize for multiple objectives. Khosla *et al.* [11] use web priors to predict relevance. Thereby they cluster web images to learn canonical viewpoints as used in a specific domain (*e.g.* cars). In order to create a summary, they select the most central video frame per cluster. This way, the keyframes are similar to web images, while the summary remains diverse. Kim *et al.* [12] combine web priors with sub-modular maximization. They formulate the problem as a subset selection in a graph of web images and video frames. Given this graph, they optimize an anisotropic diffusion objective to select a set of densely connected but diverse nodes. This leads to summaries that strike a balance between relevance to the event and representativeness within the video. Lee *et al.* [15] propose a comprehensive method for summarization of egocentric videos. They introduce a method that clusters the video into events using global image features and a temporal regularization, which ensures that clusters are compact in time. For each cluster they predict the importance of the objects it contains and select the most important ones for the final summary. As our work, Li *et al.* [17] uses a structured learning formulation, but focuses on transfer learning from text and has no approximation guarantees, since it doesn't restrict the objectives to be submodular [22].

We summarize the most related works in a taxonomy in Tab. 1. Thereby we analyze the objectives used for each method and how these objectives are combined and optimized. While existing methods focused on interestingness or representativeness, we also find temporal distribution of the summary to be important. In line with [11], we observe (Sec. 5.1), that uniform sampling provides a strong baseline, typically outperforming clustering based approaches. Uniform sampling, as naïve as it is, retains temporal coherence and thus gives a good sense of the story of the initial video. Many previous methods made simplified assumptions or defined an objective based on heuristics. Instead, we follow a supervised learning approach, where we learn the importance of the different objectives. Given a new video, these objectives are optimized jointly to create a summary.

		Sun [33]	Gygli [9]	Patapov [29]	Lee [15]	Kim [12]	**Ours**
Obj.	Interesting	√	√	√	√	√	√
	Representative	-	-	-	√	√	√
	Uniform	-	-	-	(√)	-	√
Comb.	Learnt weights	-	-	-	-	-	√
	Optimized jointly	-	-	-	-	√	√

Table 1: Taxonomy of the most recent and relevant video summarization methods. We differentiate in terms of objectives they optimize and and how they combine multiple objectives. Many methods score segment locally. Others combine multiple objectives, but do so based on a hand-defined sequential optimization. In opposition, we learn the importance of each objective from data and optimize them jointly.

Evaluation. Objectively evaluating a summary is a hard task, as there is not one true summary, but rather many ways to summarize a video well. Early methods used user studies, where viewers where asked to score [27] or compare [15, 24] automatically generated summaries. A consensus has grown that videos should be evaluated automatically to simplify evaluation and comparison [35, 29, 9, 38]. This is either done in the video [9] or text domain [38] using multiple reference summaries. Gygli *et al*. [9] evaluate using the frame overlap between an automatically generated summary and some reference summaries. As different summaries with a practically equivalent semantic meaning are possible, they use a large number of human annotated reference summaries per video to reflect this ambiguity. Instead, Young *et al*. [38] map a video summary into text and use an existing text summarization evaluation [19]. This has the advantage, that summaries are compared in terms of semantics. It however also means that the evaluation does not take into account visual aspects such as shaky cameras, etc., as long as a certain content is depicted.

3. Structured prediction with submodular functions

We formulate the task of video summarization as a subset selection problem. We are given a video $\mathcal{V}$ and a budget B. Let $\mathcal{Y}_\mathcal{V}$ denote the set of all possible solutions $\mathbf{y} \subseteq \mathcal{V}$ given this constraint.

The task of our method is to select a summary $\mathbf{y}^*$, such that it optimizes an objective o:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\mathcal{Y}_\mathcal{V}} o(\mathbf{x}_\mathcal{V}, \mathbf{y}), \tag{1}$$

where $\mathbf{x}_\mathcal{V}$ are all features extracted from the video $\mathcal{V}$. We define $o(\mathbf{x}_\mathcal{V}, \mathbf{y})$ as a linear combination of objectives $\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y}) = [f_1(\mathbf{x}_\mathcal{V}, \mathbf{y}), f_2(\mathbf{x}_\mathcal{V}, \mathbf{y}), ..., f_n(\mathbf{x}_\mathcal{V}, \mathbf{y})]^T$, each capturing a different aspect of a summary:

$$o(\mathbf{x}_\mathcal{V}, \mathbf{y}) = \mathbf{w}^\mathrm{T}\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y}). \tag{2}$$

The objectives are defined in Sec. 4. Since $\mathcal{Y}_\mathcal{V}$ is growing exponentially with the length of the video, optimally solving Eq. (2) quickly becomes intractable. Therefore, we restrict the objectives $\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y})$ to be monotone submodular and $\mathbf{w}$ to be non-negative. This allows to find a near optimal solution for Eq. (1) in an efficient way [26][2].

Next, we give a brief overview of submodular maximization and show how to learn the weights $\mathbf{w}$. Then, Sec. 4 proposes functions $\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y})$ adapted to the problem of video summarization.

[2]Without constraining $\mathbf{w}$ to be non-negative, $o(\mathbf{x}_\mathcal{V}, \mathbf{y})$ would not be guaranteed to be submodular and thus difficult to optimize.

3.1. Submodular maximization

Set functions are submodular if they fulfill the *diminishing returns* property, *i.e*. given arbitrary sets $T \subseteq U \subseteq V \setminus \{s\}$ and a set function f, f is submodular, if it satisfies: $f(T \cup \{s\}) - f(T) \geq f(U \cup \{s\}) - f(U)$. Linear combinations of submodular functions are also submodular for non-negative weights [14].

Submodular functions offer several properties desirable for optimization. It has been shown by Nemhauser *et al*. [26] that maximizing a monotonous submodular function under cardinality constraints with a greedy algorithm yields a good approximation of the optimal solution: the score of the greedy solution is lower bounded by $\frac{e-1}{e}$ ($\approx$ 63%) times the optimal value [26]. With cost constraints, *i.e*. the submodular knapsack problem, the greedy algorithm can perform arbitrarily bad. However Leskovec *et al*. [16] showed that by solving a standard and a cost-benefit greedy optimization and selecting the solution with the higher score, this is lower bounded by $\frac{1}{2}\frac{e-1}{e}$ times the optimal value. In practice, however, the greedy solution often performs much better, with an approximation factor close to 1 [20] and can be speeded up with lazy evaluations [25]. These properties are crucial for the task at hand, in order to have a scalable algorithm. In our work, we use the algorithm of [16] with lazy evaluations [25] to optimize Eq. (1), shown in Algo. 1.

For more information on submodular function maximization we refer the reader to [14].

3.2. Learning

Given T pairs of a video and a reference summary $(\mathcal{V}, \mathbf{y}_\text{gt})$, we learn the weight vector $\mathbf{w}$ of Eq. (2). Thereby we optimize the following large-margin formulation:

$$\min_{\mathbf{w}\geq 0} \frac{1}{T}\sum_{t=1}^{T} \hat{L}_t(\mathbf{w}) + \frac{\lambda}{2}||\mathbf{w}||^2, \tag{3}$$

where $\hat{L}_t(\mathbf{w})$ is the generalized hinge loss of training example t [22]:

$$\hat{L}_t(\mathbf{w}) = \max_{\mathbf{y}\subseteq\mathcal{Y}_\mathcal{V}{}^{(t)}} (\mathbf{w}^\mathrm{T} f(\mathbf{x}_\mathcal{V}^{(t)}, \mathbf{y}) + l_t(\mathbf{y})) - \mathbf{w}^\mathrm{T} f(\mathbf{x}_\mathcal{V}^{(t)}, \mathbf{y}_\text{gt}^{(t)}), \tag{4}$$

where we use superscript (t) to refer to the features and subsets of video t. The intuition behind this objective is that each human reference summary $\mathbf{y}_\text{gt}^{(t)}$ should score higher than any other summary by some margin. Given the complexity of the subset selection problem, finding the best scoring element in Eq. (4) can only be done approximately, as discussed above. We therefore resort to approximately learning and optimizing the objective using projected subgradient descent [22].

Algorithm 1 Inference algorithm for submodular maximization with approximation bounds and lazy evaluations [25, 16].

function INFERENCE($\mathcal{V}, \mathbf{x}_\mathcal{V}, c, \mathbf{w}, \mathbf{f}, B$)
  $\mathbf{y_{uc}} \leftarrow$LAZYGREEDY($\mathcal{V}, \mathbf{x}_\mathcal{V}, c, \mathbf{w}, \mathbf{f}, B$, uniform cost)
  $\mathbf{y_{cb}} \leftarrow$LAZYGREEDY($\mathcal{V}, \mathbf{x}_\mathcal{V}, c, \mathbf{w}, \mathbf{f}, B$, cost benefit)
  return $\arg\max\left(\mathbf{w}^T\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y_{uc}}), \mathbf{w}^T\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y_{cb}})\right)$
end function

function LAZYGREEDY($\mathcal{V}, \mathbf{x}_\mathcal{V}, c, \mathbf{w}, \mathbf{f}, B, type$)
  $\mathbf{y} \leftarrow \emptyset$ ▷ Start from an empty solution
  $\delta_s \leftarrow \infty, \forall s \in \mathcal{V}$ ▷ Initialize marginal gains
  while $\exists s \in \mathcal{V} \setminus \mathbf{y} : c(\mathbf{y} \cup \{s\}) \leq B$ **do**
    $cur_s \leftarrow$ **false**, $\forall s \in \mathcal{V} \setminus \mathbf{y}$ ▷ Set gains to outdated
    while true do
      if $type =$ uniform cost **then**
        $s^* \in \underset{s \in \mathcal{V}\setminus\mathbf{y}, c(\mathbf{y}\cup\{s\})\leq B}{\arg\max} \delta_s$ ▷ Max gain
      else if $type =$ cost benefit **then**
        $s^* \in \underset{s \in \mathcal{V}\setminus\mathbf{y}, c(\mathbf{y}\cup\{s\})\leq B}{\arg\max} \frac{\delta_s}{c(s)}$ ▷ Max gain / cost
      end if
      if cur_{s^*} **then** ▷ If gain of s^* is up to date
        $\mathbf{y} \leftarrow \mathbf{y} \cup \{s^*\}$; ▷ Select the element
        break
      else ▷ Else, update marginal
        $\delta_s \leftarrow \mathbf{w}^T\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y} \cup \{s^*\}) - \mathbf{w}^T\mathbf{f}(\mathbf{x}_\mathcal{V}, \mathbf{y})$
      end if
    end while
  end while
  return $\mathbf{y}$
end function

For the margin, we propose a recall loss, similar to the one used in [22] for text summarization:

$$l_t(\mathbf{y}) = \frac{1}{B}\left(|\mathbf{y}| - \left|\mathbf{y} \cap \mathbf{y}^{(t)}\right|\right), \tag{5}$$

i.e. it is a count of how many of the candidate summary $\mathbf{y}$ are not represented in the ground truth, normalized by the maximal length of the summary. We found this to work best in our experiments, but other loss functions are also possible.

Summarizing, the problem of subset selection is difficult to optimize. But if the optimization can be posed as submodular maximization, we have seen that there exist efficient algorithms, which yield good approximations.

4. Submodular functions for video summarization

Submodular functions have already been used for summarization problems, *e.g.* for document [23, 21, 22] and also image collection [31, 36] and video summarization using keyframes [12]. This is not a coincidence, since summarization inherently has a diminishing returns property: The more segments that have already been selected from a video, the less an additional segment helps to get a better overview.

Defining submodular functions for the task of video summarization is not straightforward, however. While sentences of a document can be compared relatively easy, *e.g.* by n-gram overlap, the problem of finding a semantic similarity between video segments is largely unexplored. While the dominant theme of a text can be found based on frequent sentences (n-grams), finding frequent visual content does not suffice to create a good summary. Even persons or objects appearing only for a short period of time can be of high importance for the whole video. It is therefore insufficient to optimize representativeness as for document summarization [22]. Additional measures need to be used to score video segments.

In the following, we define several submodular functions, aimed at capturing the quality of a summary. Since our method creates skims, we use segments as the atomic entities, *i.e.* a video is defined as a set of segments: $\mathcal{V} = \{s_1, s_2, ...s_n\}$ from which we select a subset $\mathbf{y}^* \subset \mathcal{V}$.

Interestingness. Following existing approaches, we predict the importance of a segment locally, *i.e.* without taking into account the rest of the video. Specifically, we want to predict a score $I(k)$ of each frame k given its features $\mathbf{x_k}$. This prediction might come from a general interestingness model as in [15, 9], or from a model that predicts a score of domain relevance, as in [33]. To allow for overlapping segments, we use the union of frames in $\mathbf{y}$ and score them with $I(k)$[3]. We use

$$f^{imp}(\mathbf{x}_\mathcal{V}, \mathbf{y}) = \sum_{k \in \bigcup_{s\in\mathbf{y}} s} I(k), \tag{6}$$

where s is a segment in the solution $\mathbf{y}$. This function is called a weighted coverage function, which is known to be submodular [14].

Representativeness. This function scores how well a summary represents the initial video. While many existing methods clustered the video into *events*, we believe this is not appropriate for raw videos, as they are continuous and therefore have gradual changes between locations and events. Instead, we propose an objective that favors representative solutions while avoiding a hard clustering.

Finding the best k segments to represent a dataset is known as the the k-medoids problem. Its objective is to select a set of medoids, such that the sum of squared errors between the datapoints and the nearest selected medoid is

[3]For the case of non-overlapping segmentations, this simply becomes: $f^{imp}(\mathbf{x}_\mathcal{V}, \mathbf{y}) = \sum_{s\in\mathbf{y}} I(\mathbf{x}_s)$, *i.e.* it is modular and a score can be assigned to a segment directly, which is more computationally efficient.

minimal, *i.e*.

$$L_r(\mathbf{x^r}, \mathbf{y}) = \sum_{i \in \mathcal{V}} \min_{s \in \mathbf{y}} ||\mathbf{x_i^r} - \mathbf{x_s^r}||_2^2, \quad (7)$$

where $\mathbf{x^r}$ are the features used to represent a segment. Here, we use global image features averaged over the segment frames for $\mathbf{x^r}$. The k-medoid objective can be reformulated as a submodular objective as follows:

$$f^{rep}(\mathbf{x}_\mathcal{V}, \mathbf{y}) = L_r(\mathbf{x^r}, \{p'\}) - L_r(\mathbf{x^r}, \mathbf{y} \cup \{p'\}), \quad (8)$$

where p' is a *phantom exemplar* [5], necessary to avoid taking the minimum over an empty set in Eq. (7).

Uniformity. As good summary tells the story of the input video, it needs to retain temporal coherence. Large jumps ahead can confuse a viewer. Similarly, a summary with many temporally adjacent segments risks being redundant. In order to avoid such problems, we propose a uniformity objective, using the same form as representativeness:

$$f^{uni}(\mathbf{x}_\mathcal{V}, \mathbf{y}) = L_r(\mathbf{x^u}, \{p'\}) - L_r(\mathbf{x^u}, \mathbf{y} \cup \{p'\}), \quad (9)$$

where we represent a segment using its mean frame number, i.e the features in $\mathbf{x^u}$ are single scalars in this case. This objective scores how well the temporal dimension is represented by the solution $\mathbf{y}$, effectively leading to solutions that are more uniformly distributed over the video.

Using these objective functions, we can now estimate the summarization objective Eq. (2). Given a set of videos and their summaries as training examples, we learn the importance of each function by optimizing Eq. (3). In the next section, we evaluate the summaries generated by our method and compare them to existing works.

5. Experiments

We evaluate the performance of our method and its individual components using two datasets: (i) the egocentric dataset of [15] and (ii) the SumMe dataset [9]. These datasets are extremely diverse: While the SumMe dataset consists of short user videos, the egocentric dataset contains hour long life-logging data from wearable cameras. Therefore, we analyze them separately in Sec. 5.1 and Sec. 5.2.

Evaluation. We evaluate w.r.t. the nearest-neighbor summary, *i.e*. the one that is the most similar to the automatically created one. This helps to account for the fact that there exists not a single ground truth summary, but multiple summaries are possible. This approach was also used in ROUGE [19], which is the standard metric in document summarization. We follow [9, 19, 38] and report the recall and f-measure, motivated by the fact that including crucially important events in more important than having perfect precision.

Compared methods. We compare to several baselines, as well as state of the art methods: (i) Uniform sampling, (ii) a previous method for the used dataset (SumMe: [9], egocentric: [15]) and (iii) Video MMR (Maximal Marginal Relevance) [2]. Video MMR, initially proposed for document summarization, was adapted to the video domain by [18]. It uses a greedy maximization of an objective that favors representativeness and penalizes redundancy of elements within the summary. We use the approach of [18], but with deep features [3], rather than SIFT+BoW to compute affinities between segments.

Implementation details. To extract the representativeness of a segment, we compute deep features trained on ImageNet [3]. We use deep features, as they are the state of the art visual features. Since they are trained for object classification, they capture objects of a scene. We used layer 6 of DeCAF [3], which has show the best performance on various recognition tasks. For Eq (8) and Eq. (9), we use a phantom element p', which has the same distance to all points in the dataset. For this, we take the mean distance of the data points.

Since the learning process receives the data points in random order, the output is also non-deterministic. Therefore we run learning and inference 100 times and average the results. We do the same for all objectives, since some might give the same score to multiple segments, *i.e*. there multiple elements might have a maximal gain (see Algo. 1, Line 13/15) We use cross-testing with 4 and 12 splits, respectively. All objectives were normalized such that the function values lie within $[0, 1]$.

5.1. Egocentric daily life dataset

The egocentric dataset of [15] contains 4 videos from wearable cameras. These videos log the day of the camera-wearer and have a duration of 3-5 hours, each, amounting to over 17 hours of video. The dataset does not include video reference summaries, but was annotated in [38] using text. Given the textual annotations for each segment of the video, a video summary can be mapped into the textual domain. There, it is compared to reference summaries using the ROUGE [19] evaluation package[4]. We use the same ROUGE parameters as [38]. Since our method requires reference summaries to train, it also requires an inverse mapping. We follow [38] and generate video summaries using a greedy bag of words and an ordered subshot method. In order to obtain multiple summaries, we vary the parameters (the n for the n-gram scoring as well as the order and maximal jump in the ordered subshot). We score these and remove the bottom 25%. Finally, we obtain 60 reference summaries (15 per video).

In order to predict the interestingness of a segment

[4] http://www.berouge.com/

	Method	**Short** (≈ 1min 20sec)		**Long** (2min)	
		F-measure	Recall	F-measure	Recall
Others	Random	$19.44 \pm 2.56\%$	$13.76 \pm 1.99\%$	$25.34 \pm 2.54\%$	$22.91 \pm 2.47\%$
	Uniform	$21.37 \pm 1.88\%$	$15.06 \pm 1.48\%$	$28.21 \pm 2.68\%$	$25.37 \pm 2.58\%$
	Lee *et al*. [15]	$17.40 \pm 4.07\%$	$12.20 \pm 3.30\%$	-	-
	Video MMR [18]	$17.73 \pm 0.00\%$	$12.49 \pm 0.00\%$	$25.57 \pm 0.00\%$	$23.10 \pm 0.00\%$
Ours	Uniformity	$18.75 \pm 1.36\%$	$12.92 \pm 1.11\%$	$25.41 \pm 1.35\%$	$22.27 \pm 1.56\%$
	Interestingness	$20.93 \pm 0.00\%$	$15.15 \pm 0.00\%$	$27.07 \pm 0.00\%$	$24.78 \pm 0.00\%$
	Representative	$19.08 \pm 0.00\%$	$12.95 \pm 0.00\%$	$27.02 \pm 0.00\%$	$23.51 \pm 0.00\%$
	Combined	$\mathbf{21.91 \pm 0.06\%}$	$\mathbf{15.73 \pm 0.04\%}$	$\mathbf{29.01 \pm 1.18\%}$	$\mathbf{26.61 \pm 1.23\%}$

Table 2: **Egocentric dataset.** Performance of the individual objectives and previous methods vs. our approach. We report results for short (≈ 1 minute and 20 seconds) as used in [15] as well as longer (2 minute) summaries.

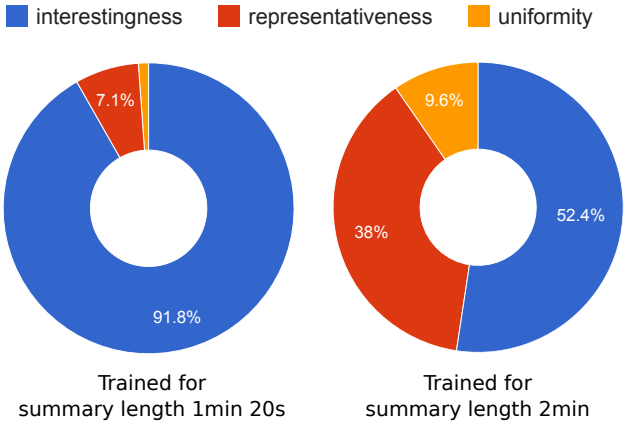

Figure 2: **Learnt weights per objective:** We can observe how the learning algorithm adapts to the specific summary length: While interestingness, *i.e*. a local prediction of importance for each segment, is the most important objective for shorter lengths, having a representative and well distributed solution becomes more important, as the summaries get longer.

(Eq. (6)), we train an classifier using deep features [3] and the training data provided by [15]. Rather than learning to classify an image region as [15], we only learn to classify whether a frame contains important objects or not. We learn a linear classifier and use its prediction confidence as an importance estimate. While more sophisticated temporal segmentation's are possible, *e.g*. [28], we use uniform segments with a length of 5 seconds. Since [38] provides annotation for the same segmentation, this allows for a non-ambiguous mapping to the textual domain in the evaluation.

Results. We evaluated our method using two summary lengths: Longer summaries of 2 minutes and shorter summaries as generated by [15]. On this dataset, our method outperforms all compared methods (see Tab. 2). It is able to learn the importance of the individual objectives for this difficult task. Furthermore, our method adapts to different summary lengths (see Fig 2). While for shorter summaries, interestingness is dominant, representativeness and uniformity get more weight for longer summaries. Thus, in short summaries the method focuses more on highlights, while it avoids getting redundant in longer summaries and therefore gives more weight to selecting representative and well distributed segments (the effect of this regularization is shown in Fig. 3). In opposition to our work, all previous methods are outperformed by uniform sampling. While this seems surprising, it can be explained by the type of video: The videos in this dataset are very slow paced and contain only few highlights. The main goal is therefore to give an overview over a camera-wearers day, for which uniform sampling is a simple, but reasonable approach. Another reason might be that the used evaluation metric only measures semantic summary quality (See Fig. 4 for an example). Thus, it ignores whether a particular segment is a good representative for a certain event (or has bad quality/motion blur). There, our method, as well as [15], have an additional advantage over uniform sampling. We show a visual comparison between [15], uniform sampling and our approach in Fig. 5.

5.2. User video dataset

The SumMe dataset [9] consists of short user videos (1 to 7 minutes). These depict a certain event of interest, *e.g*. a plane landing or a base jump. The dataset contains 25 videos, each annotated with ≥ 15 user summaries (390 reference summaries). The annotation was created in a controlled environment, where users were asked to create their own summary for a given video. To evaluate the generated summaries, we compute the overlap with these user summaries using the code provided[5]. For learning, we can directly use the user summaries of the training videos, as they already are in the video domain.

In order to predict the interestingness of a segment, we use the method of [9] with the same superframe segmentation. Given our submodular formulation however, it is not necessary to pre-commit to a fixed set of disjoint segments

[5] http://vision.ee.ethz.ch/~gyglim/vsum/

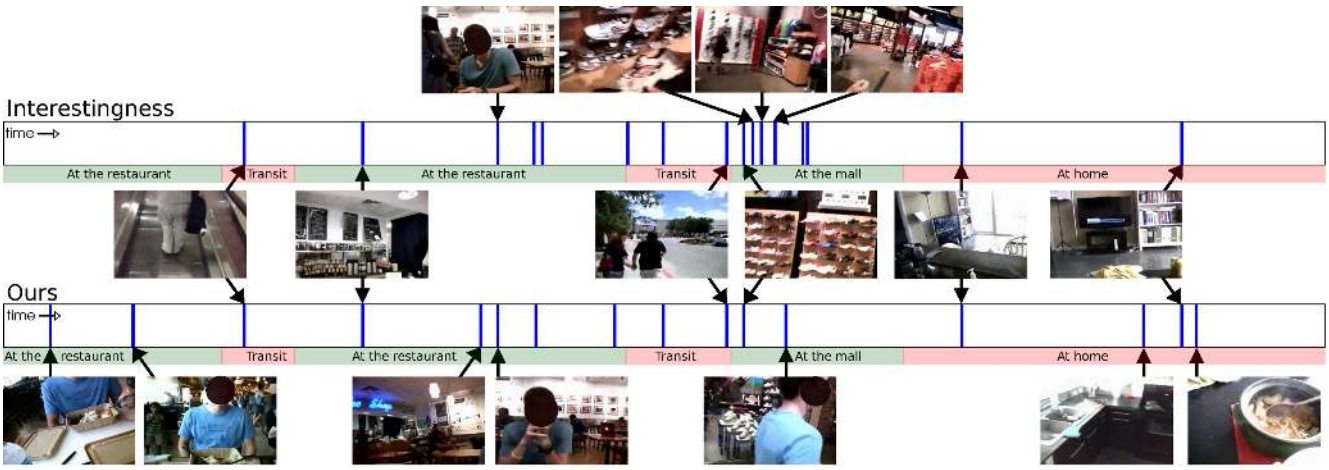

Figure 3: **Egocentric dataset, Video P1:** Selected segments of the interestingness objective and our method (shown in blue). Using multiple competing objectives helps to regularize the summarization. This leads to summaries that are more representative, while still laying focus on the most most "interesting" parts (Also see Fig. 2). Using the interestingness objective only leads to a redundant summary, where, in this example, 4 of 16 segments depict the visit in a shoe store. Thus, it misses other substantial event from the initial video. **For video results please see** `http://people.ee.ethz.ch/~gyglim/vsum_struct/`

A Reference Summary		**Ours**
I waited in line with my friend. I used the card machine to pay. I walked through the grocery store with my friend. My friend and I sat at the table and ate a meal together. My friend and I sat at the table and talked.	**Restaurant** 39 minutes	My friend and i sat at the table and ate a meal together. My friend and i sat at the table and talked. My friend and i sat at the table and talked.
I walked outside. I went down the escalator. I drove the car. I walked into the store. I walked through a cafe.	**Transit** 11 minutes	I went down the escalator. I waited in the car.
I looked at my laptop while talking to my friend. My friend and I sat at the table and talked. I wrote on my notepad. I sat at the table with my friend and drank tea.	**Restaruant** 59 minutes	I looked at the menu on the wall. I looked at the wall. I waited in line. My friend and i sat at the table and talked. My friend and i sat at the table and talked. My friend and i sat at the table and talked.
I drove the car.	**Transit** 18 minutes	I drove the car. I drove the car.
I walked into the mall. I looked at shoes on the wall. My friend and I looked at the sunglasses. I walked through the mall with my friend. I watched children bounce on the trampoline. I walked through the video game store.	**At the mall** 31 minutes	I looked at shoes on the wall. I walked in the mall. I watched children bounce on the trampoline. I looked around the store.
I washed dishes. I sliced onions. I peeled the potato.	**At home** 68 minutes	I put cleaner on the sponge. I washed a knife. I watched the television in my living room. I rinsed off the chopsticks. I ate my meal. I watched television. I washed the utensils in the sink.

Figure 4: **Egocentric dataset, Video P1:** Textual representation of a summary created with our method. Our method selects expressive segments from each of the main events and the travel between them. It tells the same story as the summary created by a human annotator. Please also be aware of the fact that the reference summaries are not extractive, *i.e.* they can contain formulations and sentences that are not in the annotation and can thus never be selected. One consequence of this are more repetitive sentences in the automatic summary. We give a keyframe vizualisation for the same video in Fig. 3.

(see Eq. (6)). We therefore run the superframe segmentation with multiple initializations. We follow [9] and generate summaries with a maximal length of 15%.

Results. Quantitative results are shown in Tab. 3. Given the content of the videos (they mostly capture some interesting event) and the length of the videos in this dataset, interestingness prediction is the dominant objective. Therefore, adding structure only leads to marginal gains over a local prediction approach. This is also reflected in the learnt weights, where interestingness has a weight of 97.5%. Nonetheless, our method learns the dominance of interestingness over the other objective and still obtains a marginal improvement over each individual objective. Please note that while [9] uses a knapsack optimization, we use a greedy procedure. On these short videos with segments of variable length, a knapsack optimization yields lengths closer to the given maximum. Therefore our method produces slightly shorter summaries, which is reflected in the recall.

	Method	F-measure	Recall
Others	Random	$28.55 \pm 1.70\%$	$30.16 \pm 1.88\%$
	Uniform	$27.07 \pm 0.94\%$	$29.42 \pm 0.86\%$
	Gygli *et al.* [9]	$39.34 \pm 0.00\%$	$\mathbf{44.44 \pm 0.00\%}$
	Video MMR [18]	$26.58 \pm 0.00\%$	$26.37 \pm 0.00\%$
Ours	Uniformity	$24.68 \pm 0.04\%$	$27.08 \pm 0.08\%$
	Interestingness	$39.52 \pm 0.00\%$	$42.50 \pm 0.00\%$
	Representative	$26.69 \pm 0.00\%$	$26.65 \pm 0.00\%$
	Combined	$\mathbf{39.68 \pm 0.09\%}$	$43.01 \pm 0.08\%$

Table 3: **User video dataset.** Performance of the individual objectives and previous methods vs. our approach.

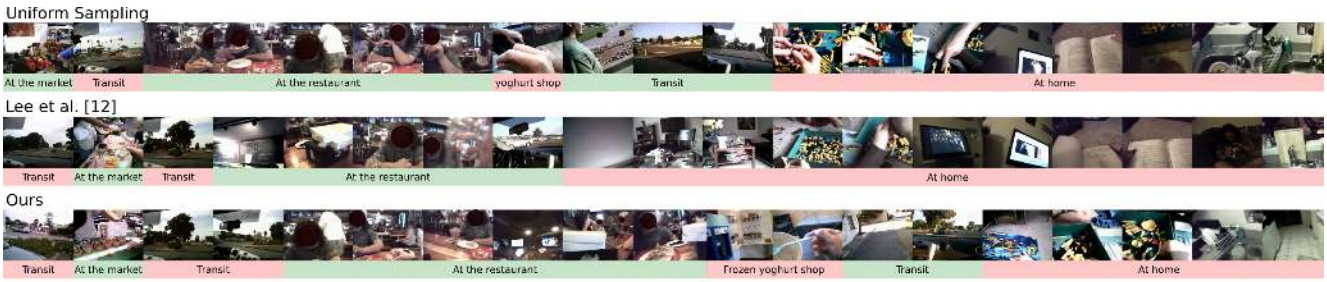

Figure 5: **Egocentric dataset, Video P2:** Qualitative comparison of uniform sampling, [15] and our method. [15] is prone to include duplicates and misses the visit of the frozen yogurt shop. While uniform sampling includes this, it is hardly visible for the selected segments/frames: Uniform sampling more often selects frames that are non-informative. Instead, our method selects a diverse set of informative segments.

6. Discussion & Conclusion

Based on our experiments and observations, we now discuss some of the insights on the advantages and limitations of our work.

Summarization Objective. We introduced a method to learn the a linear combination of summarization objectives from user summaries and jointly optimize for them. Given the complexity of the task, using hand-defined objectives as previous works [15, 12, 11] is often more difficult to implement. Indeed, our experiments have shown the advantage of our approach.

What is a good summary cannot be defined absolute. It depends on multiple factors, were the intention which a video was taken is the most obvious (live-logging vs. professional videos). We have seen in our experiments that the summary length is also important. Short summaries focus on highlights, longer summaries are expected to capture more diverse content. Furthermore, each person has different expectations when it comes to an optimal summary. Therefore an automatic summarization method should be able to adapt to user preferences. Our method could incorporate this by learning from summaries of a specific user.

Datasets and ground truth collection. For evaluation and training, human reference summaries are crucial. There is however still a shortage on the availability of large datasets for summarization. Several datasets have recently been introduced or annotated for video skimming [33, 9, 29, 38] or keyframe extraction [12, 11]. But regrettably, they are limited in size and many of them are not publicly available [39, 12, 11]. Others are not annotated for the task of summarization, but only for highlight/relevance prediction [33, 29].

Collecting summarization ground truth is time consuming, especially for longer videos. Furthermore, multiple reference summaries are necessary per video, to account for the ambiguity of the task. An ideal solution would be to have pairs of raw and professionally edited videos. An approach in this direction is of Sun *et al.* [33] who mine YouTube videos for training. Thereby, they used the correspondence between the raw and edited version of a video. Even if the quality is not perfect, such an approach could potentially be used for evaluation and would provide a way to create large datasets without explicitly letting users annotate the videos.

Interestingness. Interestingness is context dependent [35, 8]. In sports games it might be a specific semantic event, such as a goal or a foul, while in the case of a static camera in your home, a summary should contain what is rare and unusual. In the general setting, it is harder to grasp what is interesting, even though first attempts have been made, at least for images, *e.g.* [7, 8, 10]. For videos, this however remains a largely unexplored problem. But even more than in text summarization, it is important to understand what is interesting, to avoid non-informative segments or junk and instead spot the highlights. Thus far, these problems are often circumvented by using additional information on the content in the video, such as the content category [11, 33, 29] or video titles [32]. Typically, this information is used to obtain a web prior on this topic [11, 12, 32].

Conclusion. We have proposed a new method for video summarization, where we formulated the problem as a subset selection problem. Using submodular maximization, a good approximate solution can be found. We have proposed adapted submodular functions and learnt a linear combination of them using structured learning with a large-margin formulation. Our experiments have shown the potential and generality of our method. In the future, it would be interesting to apply the method on more specific problems, such as sports games, where domain-knowledge can be incorporated.

Acknowledgements. We thank Andreas Krause and Till Kroeger for their inputs. This work was supported by the European Research Council (ERC) under the project VarCity (#273940).

References

[1] I. Arev, H. Park, and Y. Sheikh. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics*, 2014. 1

[2] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM SIGIR*, 1998. 2, 5

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv preprint arXiv:1310.1531*, 2013. 5, 6

[4] D. Goldman and B. Curless. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics*, 2006. 2

[5] R. Gomes and A. Krause. Budgeted Nonparametric Learning from Data Streams. *ICML*, 2010. 5

[6] B. Gong, W. Chao, K. Grauman, and F. Sha. Diverse Sequential Subset Selection for Supervised Video Summarization. *NIPS*, 2014. 1

[7] H. Grabner, F. Nater, M. Druey, and L. V. Gool. Visual Interestingness in Image Sequences. *ACM MM*, 2013. 8

[8] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. Van Gool. The interestingness of images. *ICCV*, 2013. 8

[9] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating Summaries from User Videos. *ECCV*, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[10] A. Khosla, A. Das Sarma, and R. Hamid. What makes an image popular? In *International Conference on World Wide Web Conferences (WWW)*, 2014. 8

[11] A. Khosla, R. Hamid, C. Lin, and N. Sundaresan. Large-Scale Video Summarization Using Web-Image Priors. *CVPR*, 2013. 1, 2, 8

[12] G. Kim, L. Sigal, and E. P. Xing. Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction. *CVPR*, 2014. 1, 2, 4, 8

[13] J. Kopf, M. F. Cohen, and R. Szeliski. First-person hyperlapse videos. *ACM Transactions on Graphics*, 2014. 2

[14] A. Krause and D. Golovin. Submodular Function Maximization. *Tractability: Practical Approaches to Hard Problems*, 2011. 3, 4

[15] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. *CVPR*, 2012. 1, 2, 3, 4, 5, 6, 8

[16] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. *ACM SIGKDD*, 2007. 3, 4

[17] L. Li, K. Zhou, G. Xue, H. Zha, and Y. Yu. Video summarization via transferrable structured learning. *International Conference on World Wide Web (WWW)*, 2011. 2

[18] Y. Li and B. Merialdo. Multi-video summarization based on Video-MMR. In *WIAMIS*. IEEE, 2010. 2, 5, 6, 7

[19] C. Lin. Rouge: A package for automatic evaluation of summaries. *Workshop on Text Summarization Branches Out (WAS)*, 2004. 3, 5

[20] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *NAACL/HLT-2010*, 2010. 3

[21] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *ACL/HLT-2011*, 2011. 4

[22] H. Lin and J. Bilmes. Learning mixtures of submodular shells with application to document summarization. In *Uncertainty in Artificial Intelligence (UAI)*, 2012. 2, 3, 4

[23] H. Lin, J. Bilmes, and S. Xie. Graph-based submodular selection for extractive summarization. *ASRU*, 2009. 4

[24] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. *CVPR*, 2013. 1, 2, 3

[25] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, 1978. 3, 4

[26] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14, 1978. 3

[27] P. Over, A. F. Smeaton, and G. Awad. The TRECVID 2008 BBC rushes summarization evaluation. *Proc. ACM WS on Video summarization*, 2008. 3

[28] Y. Poleg, C. Arora, and S. Peleg. Temporal Segmentation of Egocentric Videos. *CVPR*, 2014. 6

[29] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. *ECCV*, 2014. 1, 2, 3, 8

[30] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *PAMI*, 2008. 2

[31] I. Simon, N. Snavely, and S. M. Seitz. Scene Summarization for Online Image Collections. *CVPR*, 2007. 4

[32] Y. Song, J. Vallmitjana, A. Stent, and A. Jamies. TVSum: Summarizing Web Videos using Titles. *CVPR*, 2015. 8

[33] M. Sun, A. Farhadi, and S. Seitz. Ranking Domain-Specific Highlights by Analyzing Edited Videos. *ECCV*, 2014. 1, 2, 4, 8

[34] M. Sun, A. Farhadi, B. Taskar, and S. Seitz. Salient Montages from Unconstrained Videos. *ECCV*, 2014. 2

[35] B. T. Truong and S. Venkatesh. Video abstraction. *ACM TOMCCAP*, 2007. 1, 2, 3, 8

[36] S. Tschiatschek and J. Bilmes. Learning Mixtures of Submodular Functions for Image Collection Summarization. *NIPS*, 2014. 4

[37] W. Wolf. Key frame selection by motion analysis. *Acoustics, Speech, and Signal Processing*, 1996. 2

[38] S. Yeung, A. Fathi, and L. Fei-Fei. VideoSET: Video Summary Evaluation through Text. *arXiv preprint arXiv:1406.5824*, 2014. 3, 5, 6, 8

[39] B. Zhao and E. P. Xing. Quasi Real-Time Summarization for Consumer Videos. *CVPR*, 2014. 1, 2, 8