Encoded Semantic Tree for Automatic User Profiling Applied to Personalized Video Summarization

Yifang Yin, Roshan Thapliya, and Roger Zimmermann, *Senior Member, IEEE*

***Abstract*— We propose an innovative method of automatic video summary generation with personal adaptations. User interests are mined from their personal image collections. To reduce the semantic gap, we propose to extract visual representations based on a novel *semantic tree* (SeTree). A *SeTree* is a hierarchy that captures the conceptual relationships between the visual scenes in a codebook. This idea builds upon the observation that such semantic connections among the elements have been overlooked in the previous work. To construct the *SeTree*, we adopt a normalized graph cut clustering algorithm by conjunctively exploiting visual features, textual information, and social user-image connections. Using this technique, we obtain an 8.1% improvement of normalized discounted cumulative gain in personalized video segments ranking compared with existing methods. Furthermore, to promote the interesting parts of a video, we extract a space–time saliency map and estimate the attractiveness of segments by kernel fitting and matching. A linear function is utilized to combine the two factors, based on which the playback rate of a video is adapted to generate the summary. We play the less important segments in a fast-forward mode to keep users updated with the context. Subjective experiments were conducted which showed that our proposed video summarization approach outperformed the state-of-the-art techniques by 6.2%.**

***Index Terms*— Semantic modeling, user profiling, video summarization, visual attention.**

I. Introduction

With the rapid development of network techniques and multimedia sharing platforms, posting and watching videos online has become an important way for people to share interests and ideas with each other. However, due to the fast growing video collections, it has become increasingly challenging for users to find the information they desire to view. Moreover, considering the limitation of the available network bandwidth [1], it is important to adapt the content displayed to users to obtain an improved quality of experience (QoE) in video browsing. In the past several years, extensive research has been conducted in video summarization to generate a compact and informative version by extracting the essential information [2]. Such a summarization scheme is a highly important module as it can be used as a first step for many downstream video content management tasks such as video search and delivery.

Traditionally, a video summarization is generated by extracting salient keyframes based on visual features [3], [4]. Various aspects have been utilized including scenes [4]–[6], motions [7], [8], and object-of-interest [9]–[11] in the saliency estimation and frame ranking. However, without prior knowledge of the user preferences, personalized adaptation was never an option in the above approaches. To improve the performance in user-centric applications, studies have been performed on personalized video summarization which is highly challenging due to the difficulties encountered in: 1) user interest modeling and 2) content-based video segment ranking. Most of the current techniques on user profiling involve manual interactions [12], [13], which might be tiresome for people to manage. A user is usually required to input preferences by specifying keywords [14], selecting preferred events [15], or categorizing personal photo libraries [16], [17]. However, the limited descriptiveness of predefined categories may hinder the understanding of a user's intent.

It would be ideal if user preferences could be automatically detected from certain kinds of personal data. One promising source of information is the personal photos available from social sharing applications such as Flickr and Picasa. As pointed out by Takeuchi and Sugimoto [16], personal photos contain rich information about people's tastes and lifestyles. For example, from the photo collections taken during traveling, it is easy to see the type of tourism that a person likes the most. However, tourism videos are usually lengthy by showing all the attractive aspects of a country as illustrated in Fig. 1. If a user shares a great number of images of museums and churches, he or she is more likely to be a fan of cultural tourism. Therefore, video shots that introduce a country's history, architecture, and religions should be ranked higher for personal adaptation. The similarity between video segments and personal photos can be measured with dictionary-based feature extraction techniques such as soft assignment [18] and sparse coding [19]. However, existing approaches mostly adopt a codebook formed by a set of visual descriptors without

Manuscript received March 14, 2016; revised July 10, 2016; accepted August 17, 2016. Date of publication August 25, 2016; date of current version January 5, 2018. This work was supported by the Singapore National Research Foundation under its International Research Centre at Singapore Funding Initiative and administered by the IDM Programme Office through the Centre of Social Media Innovations for Communities. This work was performed during a research internship at the Research and Development Center, Fuji Xerox Co., Ltd., Japan. This paper was recommended by Associate Editor S. Satoh.

Y. Yin and R. Zimmermann are with the School of Computing, Interactive and Digital Media Institute, National University of Singapore, Singapore 117417 (e-mail: idmyiny@nus.edu.sg; rogerz@comp.nus.edu.sg).

R. Thapliya is with the Research and Development Center, Fuji Xerox Company, Ltd., Yokohama 220-8668, Japan (e-mail: roshan.thapliya@fujixerox.co.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2602832

Fig. 1. Example of adapting diverse tourism videos based on personal preferences.

considering any semantic relationships among the elements, which hinders the achievement of improved results. Compared with user specified preferences, one may argue that automatic profiling might be less effective in describing the current intent of a user. Therefore, to avoid mistakenly skipping important parts, content-based adaptation is also necessary in order to generate an informative overview that covers the major subjects in the original video.

To fulfill the above criteria, we propose a video summarization framework that estimates the importance of a segment based on both the user profiles and the visual attention scores. Fig. 2 illustrates the video segment ranking module which is one of the core components highlighted in the flowchart as shown in Fig. 3. Different from traditional approaches where representative images are selected as queries, we model user preferences through a compact representation extracted with a semantic tree. A semantic tree is a hierarchical dictionary that encodes the semantic relationships among the visual scenes, i.e., the images in a branch should be instances of the subconcepts of the root. To construct such a hierarchy, we measure the pairwise similarity between images by conjunctively considering visual features, textual information, and social user-image connections that are available from social sharing platforms such as Flickr. Next, a normalized graph cut clustering approach is applied to generate the semantic tree, based on which both user photos and video keyframes are encoded for personalized saliency score estimation. Moreover, to measure the visual attention score of the content, we compute the spatiotemporal saliency based on the off-the-shelf space–time saliency detection [9]. Note that we carry out an additional step that models the saliency map by a Gaussian kernel to reduce noise. The output kernel can be interpreted as the region-of-interest (ROI) of the input frame. Subsequently, we favor situations where the ROI is close to the frame center by adopting Kullback–Leibler divergence (KLD) as the distance measure. Finally, a linear fusion is adopted to generate the final video summary by adjusting the playback rate. Only the top ranked segments are selected and played in a normal playback speed to improve the QoE in video browsing.

The contributions of this paper are summarized as follows.

1) The introduction of a novel hierarchical dictionary named semantic tree to encode the conceptual relationships among the visual scenes.
2) An automatic content-based feature encoding approach with the semantic tree, which is shown to be more effective for personalized adaptation.
3) The design of a video summarization prototype by conjunctively considering personal interests and visual attention. Experiments show that our proposed method outperforms the state-of-the-art techniques.

The rest of this paper is organized as follows. We first report the important related work in Section II. The construction of the proposed semantic tree is introduced in Section III, followed by the feature encoding technique introduced in Section IV. Next, we apply our user profiling approach to personalized video summarization and present the visual attention model in Section V. The experimental results in Section VI validate the effectiveness of our system. Section VII concludes and suggests future work.

II. Related Work

Video summarization has been extensively studied in the past several years. One of the core problems is to determine the important parts of a video [22], [23]. Traditional video summarization detects a set of salient keyframes based on the visual clues [4], [24]. For example, Zhuang *et al.* [3] proposed a clustering-based approach that summarized a video by a collection of keyframes that were identified as cluster centers. Ngo *et al.* [5] proposed to abstract a video by scene detection using normalized graph-cut algorithm. Guan *et al.* [6] proposed a top-down approach that took both the global and local perspectives into consideration when selecting representative keyframes. Other aspects such as motion [7], [8], title [25], and video categories [26] have also been considered in the user attention model for importance ranking [27], [28]. Nguyen *et al.* [11] presented a unified framework to detect both static and space–time saliency in video sequences. Almeida *et al.* [29] presented a summarization approach for online video applications and addressed the issues in the compressed domain. Some recent work focused on egocentric videos [10], [30], and extracted important objects with which the camera wearer interacts. However, the above work analyzed the video content with little consideration of the user preferences. This greatly hinders the achievement of satisfactory results for practical use.

For personalized video summarization, Wei *et al.* [14] proposed to adapt the video content based on both the client-side resource constraints and the keywords provided by users as preferences. Xu *et al.* [15] proposed a personalized video adaptation scheme by mining both cognitive content and affective content. Their system gives high priority to the events and affective level selected by a user. Takeuchi and Sugimoto [16], [17] proposed to mine user preferences from personal photo libraries. However, users are required to cluster photos into several major categories to represent their interests. Zhang *et al.* [13] proposed to generate personalized sketch summarization of events from the interactively selected keyframes. As we can see, most of the personalized summarization techniques require human interactions to acquire the user preferences, which might be tiresome to some extent [12], [13]. Table I shows a comparison of the

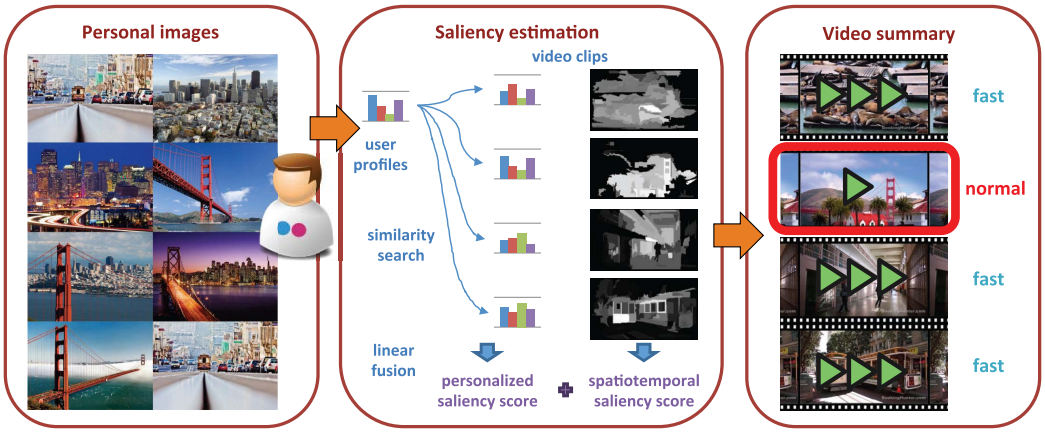


Fig. 2. Video segment ranking in our proposed personalized summarization framework.

TABLE I
COMPARISON WITH PREVIOUS WORK

Work	General-purpose video summarization [4], [20], [21]	Wei *et al.* [14]	Xu *et al.* [15]	Takeuchi and Sugimoto [16], [17]	**Our proposed SeTree method**
Content-based Adaptation	✓				✓
Personalized Adaptation		✓	✓	✓	✓
User Preference	–	Semantic terms specified by a user	Pre-defined classes selected by a user	Clusters derived from personal photo libraries	Feature encoding from personal images

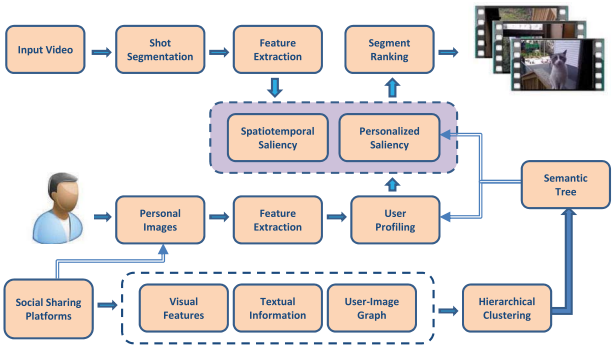


Fig. 3. Illustration of the major components in this paper.

related work. Generally speaking, video abstraction strategies can be roughly divided into content-based and personalized adaptation schemes. Popular techniques belonging to the former include scene clustering [4] and visual attention-based summarization [20], [21]. Such methods are designed without any prior knowledge of user preferences. Comparatively, personalized adaptation usually relies on classifiers trained for a list of predefined concepts [15], [31] or visual example-based similarity search [17], [32], [33]. Due to the unpredictable condition change in videos such as illumination and viewpoint, one of the major issues in this area is to bridge the semantic gap between the visual clues and the semantic concepts. Moreover, with the surge of user-centric applications, challenges have also been posed in the presentation of user's intent by query.

In multimedia search and recommendation [34]–[36], textual clues and user behaviors have long been utilized as supplementary information in addition to visual features. Davidson *et al.* [35] discussed the challenges of the YouTube video recommendation system. They constructed a graph of videos based on covisitation activities. Subsequently, they proposed to generate the recommendation list by expanding from the watched, favored, and liked videos of a user. Liu *et al.* [36] proposed to rerank the video search results from a global perspective. Multifeatures including text, visual, and audio information were used in the neighborhood score propagation. However, different from video recommendation, the ranking of video shots in a summarization system is highly challenging as in most cases only the raw video stream can be used for analysis. The content-based video summarization approaches are still struggling to achieve satisfactory results.

III. SEMANTIC TREE CONSTRUCTION

We propose an unsupervised video summarization framework by predicting the user preference based on personal image collections. Video shots showing conceptually similar content should be given higher weights as the subjects are more likely to arouse the user's interest. To effectively rank video shots *with respect to* personal images, we propose to extract features based on a semantic tree, which is a hierarchical dictionary that describes the semantic relationship among the visual scenes. Previous solutions overlooked the conceptual connections among the elements in the dictionary [32], [37], which hinders the generation of more accurate representations for multimedia documents.

To construct such a hierarchy, we collect social images from Flickr to deem as the leaf nodes in the tree. We compute the pairwise similarity of the leaves by exploiting the implicit relationships among them based on the visual features, the textual information, and the social user-image connections. Thereafter, a normalized graph cut clustering algorithm is applied to generate the semantic tree that will be used as the dictionary for visual feature encoding.

A. Pairwise Image Similarity Measure

1) Visual Similarity: The visual similarity between images is estimated based on the Euclidean distance between their visual descriptors (e.g., HOG, GIST, and SIFT). To preserve the semantic meanings, here we adopt the ObjectBank representation [38]. It is a high-level visual representation that describes an image as a scale-invariant response map of a large number of pretrained generic object detectors.

Let I_i denote an image and X_i be its visual descriptor. The visual similarity between images I_i and I_j is subsequently computed with a Gaussian function as

$$\mathrm{VS}(I_i, I_j) = \exp\left(-\frac{\|X_i - X_j\|_2^2}{\sigma^2}\right) \tag{1}$$

where σ is a smoothing factor. Such content-based image analysis has its own limitations such as the well-known semantic gap. Therefore, it might not be sufficient to utilize only the visual features in the similarity measure. In the following sections, we will introduce how to exploit the textual and social connections between images to tune the visual space defined by the dictionary.

2) Textual Semantics: Nowadays, social sharing platforms allow users to add tags to photos for document search and management. We take advantage of this and compute the semantic similarity between images by analyzing the associated tags based on the WordNet [39]. We filter the raw tags by removing the ones that do not exist in the WordNet, and compute the semantic similarity of the remaining tags with an information-based approach. The key idea is to measure the amount of information two tags share. According to [40], the similarity between two tags t_i and t_j is estimated as

$$\mathrm{Sim}(t_i, t_j) = \frac{2\mathrm{IC}(\mathrm{lso}(t_i, t_j))}{\mathrm{IC}(t_i) + \mathrm{IC}(t_j)} \tag{2}$$

where $\mathrm{lso}(t_i, t_j)$ is the lowest superordinate and $\mathrm{IC}(t)$ denote the information content of tag t. Subsequently, Zhou *et al.* [39] modeled the information content of a tag based on its hyponyms and depth as

$$\mathrm{IC}(t) = k\left(1 - \frac{\log(\mathrm{hypo}(t)+1)}{\log(\mathrm{node}_{\max})}\right) + (1-k)\left(\frac{\log(\mathrm{deep}(t))}{\log(\mathrm{deep}_{\max})}\right) \tag{3}$$

where functions $\mathrm{hypo}(t)$ and $\mathrm{deep}(t)$ return the number of hyponyms and the depth of tag t, respectively. $\mathrm{node}_{\max}$ and $\mathrm{deep}_{\max}$ are constant values set to the maximum number of concepts in the taxonomy and the maximal depth of the taxonomy, respectively. k is a balancing factor that controls the weights of the two aspects. In the experiments, we used the Semantic Measures Library [41], which is an open-source Java implementation of semantic measures.[1]

Let $T = \{t_1, t_2, \ldots, t_n\}$ denote the tag set after filtering with WordNet. We describe the tags associated with image I_i by a vector $T_i = [t_1^i, t_2^i, \ldots, t_n^i]$ where

$$t_j^i = \begin{cases} 1 & \text{if } I_i \text{ is annotated by } t_j \\ 0 & \text{else.} \end{cases} \tag{4}$$

By normalizing T_i by its Manhattan norm, the textual feature of image I_i is represented as $T_i = ((T_i)/(\|T_i\|_1))$. Finally, the textual similarity between images I_i and I_j is computed by

$$\mathrm{TS}(I_i, I_j) = T_i M T_j^{\mathsf{T}} \tag{5}$$

where M is a similarity matrix. The element in the ith row and the jth column of M, denoted by m_{ij}, is the similarity score between tags t_i and t_j computed with (2).

3) Social Graph: Different from the visual and textual similarity where the scores are directly computed between individual images, we analyze the user behaviors on social sharing platforms from a global perspective. The basic idea is that if people who like image I_i also like image I_j but dislike image I_k, the distance between I_i and I_j is likely to be smaller than the distance between I_i and I_k. The photos shared by one user might be diverse, but the implicit connections among images can be derived from multiple user behaviors. For analysis, we build a user-image graph $G = \{V, E\}$, which is undirected bipartite. The vertices V are users and images. Let W denote the weights assigned to the edges in graph G based on user behaviors. Formally, $w_{ij} = 1$ if one vertex of edge e_{ij} is a user node and the other is an image node that has been favored by this user on the social sharing platforms, and $w_{ij} = 0$ otherwise. To capture the global structure of the graph, we compute the relatedness for every pair of images through the random walk and restart (RWR) algorithm [42].

Starting from a vertex i, RWR computes the relevance scores of the nodes in V *with respect to* vertex i as defined by

$$\vec{r_i} = c\tilde{W}\vec{r_i} + (1-c)\vec{e_i} \tag{6}$$

where $\vec{r_i}$ is the vector of relevance scores, $c \in [0, 1]$ is the restart probability, $\tilde{W}$ is the normalized weighted matrix with reference to W, and $\vec{e_i}$ is the starting vector with the ith element set to 1 and 0 for the others. The steady-state probabilities $\vec{r_i}$ can be solved by iteratively applying (6) until convergence. Thereafter, the relevance scores between every pair of images can be obtained by starting from different image vertices.

Let $\mathrm{SS}(I_i, I_j)$ be the social relevance between images I_i and I_j computed by RWR. The similarity between the visual scenes in the dictionary is defined as a linear combination of the scores calculated based on the above three key clues, which is

$$\mathrm{Sim}(I_i, I_j) = \alpha\mathrm{VS}(I_i, I_j) + \beta\mathrm{TS}(I_i, I_j) + \gamma\,\mathrm{SS}(I_i, I_j) \tag{7}$$

where α, β, and γ are positive weighting factors, subject to $\alpha + \beta + \gamma = 1$.

[1] http://www.semantic-measures-library.org/sml/

B. Hierarchical Clustering

As aforementioned, the purpose is to construct a hierarchy that can be used as the dictionary for improved visual feature encoding. It is expected to be more descriptive as it additionally captures the relationships among the visual scenes. To build such a dictionary, we hierarchically cluster the leaf nodes based on the pairwise similarities computed with (7). This process is controlled by the following two parameters: 1) the height of the tree, denoted by H and 2) the number of branches that each node has, defined by B.

Specifically speaking, we first cluster the leaf nodes of images into B groups. Next, we recursively carry out the same process and cluster the images in each group into B subgroups. By doing this, the semantic tree is created from the root to the leaves, until the maximum height H is reached.

Compared with the vocabulary tree proposed in [37], our model differs in several aspects. First, the vocabulary tree is designed for quantizing image local descriptors where the visual feature is the only available clue for clustering. Comparatively, our model targets the hierarchical dictionary construction of visual scenes that carry semantic information [32]. Additional data sources of textual and social cues can also be analyzed for hierarchical clustering. By utilizing such supplementary information in addition to the visual features, the nodes in the same branch of the tree are more likely to be semantically consistent with each other. Recall that the social relevance scores are computed from a global graph without generating feature vectors for images. Therefore, we adopt the normalized graph cut algorithm [43] instead of k-means clustering as the latter is no longer applicable in our case. Moreover, our model adopts an alternative feature encoding technique. The details will be discussed in the following section.

IV. Feature Encoding

Let $\tilde{I} = \{I_1, I_2, \ldots, I_n\}$ denote the set of Flickr images for dictionary construction, and $\tilde{J} = \{J_1, J_2, \ldots, J_m\}$ denote the personal images of a user ($n \gg m$). We first individually encode the personal images and then pool the scores at each node to obtain the final representation. As the coding process is the same for every image J_j, we will omit the subscript in circumstances of no ambiguity.

To utilize the semantic tree for feature encoding, we first generate sparse code on the leaf nodes (Flickr images) and then propagate the scores through the internal nodes up to the root. Formally, we use node$_i^h$ to represent the ith node with a height of h, $h \in \{0, 1, \ldots, H\}$, as the height of a node is the number of edges on the longest downward path between that node and a leaf. The coding process is carried out as follows. First, for every image $J \in \tilde{J}$, we represent it by its visually similar neighbors among the leaf nodes. Let $D(J, I_i)$ be the Euclidean distance between images J and I_i in the visual feature space. The scores we assign to the leaf nodes with reference to J are calculated as

$$s_i^0 = \frac{\mathbb{1}_{\text{nei}_J}(I_i) \cdot K_\sigma(D(J, I_i))}{\sum_{j=1}^{n} \mathbb{1}_{\text{nei}_J}(I_j) \cdot K_\sigma(D(J, I_j))} \tag{8}$$

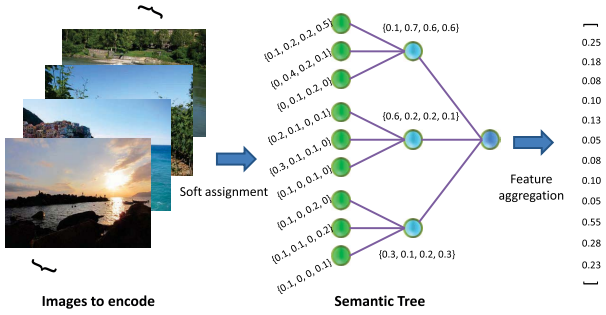


Fig. 4. Illustration of feature encoding using the semantic tree.

where $K_\sigma(x) = (1/(\sqrt{2\pi}\sigma)) \exp(-((x^2)/(2\sigma^2)))$ is a Gaussian kernel and $\mathbb{1}_{\text{nei}_J}(I)$ is an indicator function that selects the k-nearest neighbors of image J in the leaf node images as shown in (9). In the experiments, we find that the feature encodings become less effective when $k > 200$. Therefore, we set $k = 100$ and select the top 100-nearest neighbors for every image J

$$\mathbb{1}_{\text{nei}_J}(I) = \begin{cases} 1 & \text{if } I \in \text{k-NN of } J \\ 0 & \text{else.} \end{cases} \tag{9}$$

After soft assignment on the leaf nodes, we propagate the scores to the internal nodes level by level. The score of an internal node s_i^h is defined as the sum of the scores of its child nodes. To aggregate the features of all personal images in set J, we apply the average pooling strategy where only the mean of the scores associated with each node is kept. This process is illustrated in Fig. 4. As can be seen, in this example we encode a set of four images based on the semantic tree. After individually extracting the features for every input image by soft assignment, each node is associated with four scores correspondingly. The next step is to apply a feature pooling strategy to generate a compact representation. Popular pooling strategies include average pooling and max pooling. We adopt the former because personal photo collections are usually quite diverse where individual images should not be emphasized too much. The average pooling scheme generates a single feature vector for a set of images while being able to maintain the feature descriptiveness at the same time.

To further improve the system effectiveness, we additionally carry out a weighting process after the feature aggregation. As suggested in [37], we assign weights w_i^h to each of the nodes in the tree as

$$w_i^h = \ln \frac{N}{N_i^h} \tag{10}$$

where h denotes the height of the node, N is the total number of images in the training data set, and N_i^h is the number of images that are the descendant of node$_i^h$. $\ln(N/(N_i^h))$ is an entropy weighting that promotes the nodes containing descriptive visual scenes. As the nodes at the higher levels are usually associated with larger N_i^h, $\ln(N/(N_i^h))$ also decreases the weights assigned to the nodes close to the root. It is also possible to block the higher levels in the tree by setting their weights to zero as the nodes close to the leaves are

generally more representative in the feature encodings. Finally, we update the node scores by multiplying the weights, that is

$$x_i^h = w_i^h \cdot s_i^h. \tag{11}$$

The final representation is generated by concatenating the scores of leaf and internal nodes on each level into a vector, which is $X^{\mathrm{STr}} = [x_1^0, x_2^0, \ldots, x_n^0, \ldots, x_1^{H-1}, x_2^{H-1}, \ldots, x_B^{H-1}]$.

Similarly, the feature of a video segment can be extracted by carrying out the same process on the set of video frames, $\tilde{F} = \{F_1, F_2, \ldots, F_m\}$, belonging to it.

V. Video Summarization

Previously, we have introduced the construction of the semantic tree and its utilization for feature encoding. Here, we apply the semantic tree to a personalized video summarization system. We extract frames from a video at a sample rate of two per second for feature encoding and shot detection. The frames are then clustered into groups and the shot boundaries are determined whenever two consecutive frames have been clustered into different groups [44]. Next, we estimate the importance of each segment based on which a dynamic video summary is generated to improve the QoE in video browsing.

A. Video Segment Ranking

A high-quality video summary is expected to satisfy a user's needs by personalized adaptation. Compared with pretrained classifiers, unsupervised methods reduce the manual efforts but have one drawback of being comparatively less accurate. Therefore, we would also like to maintain the informativeness of the summary in order to make sure that no important parts will be missed by users. To fulfill the above criteria, we rank video segments by linearly combining a personalized saliency score PS with a spatiotemporal saliency score AS, formally given as

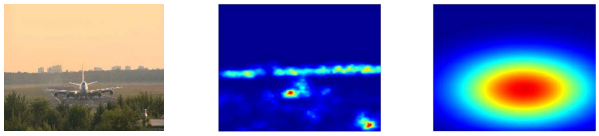
$$S_i = \mathrm{PS}_i + \lambda \mathrm{AS}_i \tag{12}$$

where λ is a balancing factor. The subscript i indicates that the scores are computed with reference to the ith segment in the input video. Next, we introduce the technical details of how to calculate the personalized and the spatiotemporal saliency scores, respectively.

1) Personalized Saliency: The personalized saliency score is estimated by comparing each video segment with the user profile. As introduced in Section IV, the feature of a set of images (or frames) encoded with a semantic tree can be represented as a vector $X^{\mathrm{STr}} = [x_1^0, x_2^0, \ldots, x_n^0, \ldots, x_1^{H-1}, x_2^{H-1}, \ldots, x_B^{H-1}]$, where H is the height of the tree, B is the number of branches that each node has, and n is the total number of leaf nodes in the tree.

Let X_u^{STr} be the feature of the personal images of a user and X_i^{STr} be the feature of the ith video segment. The personalized saliency score for the ith segment, denoted as PS_i, is formulated as

$$\mathrm{PS}_i = 1 - \frac{1}{2}\left\| \frac{X_i^{\mathrm{STr}}}{\|X_i^{\mathrm{STr}}\|} - \frac{X_u^{\mathrm{STr}}}{\|X_u^{\mathrm{STr}}\|} \right\|^2. \tag{13}$$



(a) Video frame (b) Saliency map (c) Kernel fitting

Fig. 5. Illustrations of our proposed attention-based spatiotemporal visual saliency modeling. (a) Video frame. (b) Saliency map. (c) Kernel fitting.

We normalize the feature vectors with L_2-norm and convert the distance measure into the cosine similarity between the two feature vectors with (13). In the experiments, different normalization schemes have been evaluated and L_2-norm has been shown to obtain better results than the L_1-norm.

2) Spatiotemporal Saliency: In recent years, extensive studies have been carried out in the field of visual attention modeling for images and videos [9], [11]. By fusing the spatial and temporal attention values, a static video summary is usually generated by extracting a set of visually salient keyframes from the video [20]. To follow the path of the existing attention-based schemes, we generate the saliency map for each frame by utilizing the off-the-shelf space–time saliency detection approach [9]. As illustrated in Fig. 5, the first picture is an input video frame. Fig. 5(b) shows the estimated saliency map of the input frame by employing the space–time local steering kernels. Instead of obtaining the attention score of a frame by directly aggregating the pixelwise saliencies from the map, we apply an additional step by fitting the data into a Gaussian kernel to reduce the noise [see Fig. 5(c)].

The attention score of a frame is formulated based on two factors. The first factor is the weighted sum of the saliency map. Let $\mathrm{smap}(i, j)$ denote the saliency value of the pixel located at position (i, j) before kernel fitting. The Gaussian kernel estimated based on the saliency map, denoted by Q, describes the distribution of the salient pixels in a frame. Thereafter, the sum of the saliency map weighted by kernel Q is computed as

$$\mathrm{Sum}(\mathrm{smap}, Q) = \sum_{i,j} Q(i, j) \cdot \mathrm{smap}(i, j). \tag{14}$$

The second factor is based on the observation that people tend to focus on the center of an image. Let $P = \mathcal{N}(\mu, \Sigma)$ denote the normal distribution located at the center of an image. We favor the saliency distributions Q that are close to the ideal distribution P by computing the KLD, which is defined to be the integral

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} \ln\left(\frac{p(u)}{q(u)}\right) p(u) du \tag{15}$$

where $p(u)$ and $q(u)$ are the densities of the distributions P and Q. In our implementation, we utilized a MATLAB toolbox for kernel density estimation *(KDE)*[2] for Gaussian kernel estimation and KLD calculation.

[2]http://www.ics.uci.edu/ ihler/code/kde.html

Subsequently, the attention score of a frame F is formulated by

$$\mathrm{AS}(F) = \frac{\mathrm{Sum}(\mathrm{smap}_f, Q_f)}{D_{\mathrm{KL}}(P \parallel Q_f)} \tag{16}$$

where smap_f and Q_f indicate the saliency map and the Gaussian kernel associated with frame F.

Let $\tilde{F}_i$ represent the frames of the ith video segment, the attention-based spatiotemporal saliency score for this segment, denoted by AS_i, is computed by averaging the frames it contains

$$\mathrm{AS}_i = \frac{1}{\|\tilde{F}_i\|} \sum_{F \in \tilde{F}_i} \mathrm{AS}(F). \tag{17}$$

B. Dynamic Summary Generation

To improve the QoE for video browsing, we generate a dynamic video summary based on the saliency estimation introduced in the previous sections. The top-ranked video segments are selected and displayed to users at the normal playback rate. To keep users updated with the context between the selected shots, we play the rest of the video in a fast-forward mode instead of completely cutting off the less interesting parts. The length of the final summary can be controlled by a parameter ratio $\in (0, 1)$. The length of the selected salient video segments should not exceed ratio times the total length of the input video.

Traditional video summarization methods can be divided into two categories, namely, the static keyframe abstraction and the dynamic video skimming. Our strategy belongs to the latter. It has the advantage of presenting the users an informative video summary. Due to the limitations of the content-based video analysis, people might be afraid of missing any of the interesting scenes or events captured in the video. Therefore, a better strategy is to enable rapid skimming at a fast playback speed in order to ensure that no important segments are mistakenly skipped [45].

Please note that the major issue we studied in this paper is how to effectively estimate the importance of each segment. In practice, people may have different preferences on how to present the summary to users. This part can be easily customized by letting users choose the way they prefer. It could be simply presenting the keyframes of important segments, skipping the less interesting parts, or displaying the video at a customized fast playback speed as we did in the experiments.

VI. Evaluation

To evaluate the effectiveness of our proposed approach, we collected 41 212 images of 100 users from Flickr as the experimental data set. After manually filtering out valueless images (e.g., screenshots), we randomly selected 20 users and used their image collections as test queries for personalized video segment ranking in Section VI-B. The rest of the Flickr images were used as the training samples for the semantic tree construction and the parameter tuning. We utilized a public video data set SumMe [21] to test the proposed attention-based ranking model. Additionally, we prepared a new YouTube data set due to the lack of public videos for personalized ranking evaluation. To keep the data set diverse and manageable, we collected 25 videos of five categories including animal, natural scene, cityscape, food, and landmark, the size of which is similar to other summarization papers [20], [21].

TABLE II
Average nDCG Comparison Based on Different Clues and Their Fusion

Method	Random	Visual	Textual	Social	Fusion
nDCG	0.412	0.525	0.459	0.504	0.547

A. Parameter Tuning

As aforementioned, we examined the three key clues for image similarity estimation. The parameters α, β, and γ in (7) should be set according to the quality of their corresponding data sources. To measure the reliability of the three information sources in our training set, we randomly sampled 100 images from the training data set. Each of them was used as a query to rank the rest of the images based on the similarity scores derived from different clues. The measure we used for comparison is the normalized discounted cumulative gain (nDCG). It is designed for evaluating the ranking quality. For a query q, if rel_i denote the relevance score of the ith item in the result list, the DCG is calculated as

$$\mathrm{DCG} = \sum_i \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)}.$$

To normalize over queries, we compute the ideal DCG (IDCG), which equals the maximum possible DCG produced by the ideal ranking list. Subsequently, nDCG is computed as

$$\mathrm{nDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}.$$

The ground-truth relevance scores between images were defined by four levels: 0.9, 0.6, 0.3, and 0.1 (0.9-most, 0.1-least) based on the relevance rank estimated by humans. The image similarity of each pair was judged by a total of 21 people and the average score over all the subjects was adopted as the ground-truth annotation. Irrelevant images were assigned a score of zero.

We carried out 100 queries with similarity measures as introduced in Section III-A based on visual, textual, and social features, respectively. The comparison of the average nDCG over the queries is given in Table II. To show the statistics of the data set, we also report the result achieved by random permutations in the first column. As can be seen, the visual clues were more reliable than the others as the ObjectBank representation carried semantics of the image content to some extent. The tags of Flickr images were added by their uploaders and therefore might be inaccurate and incomplete. After filtering using the WordNet, a number of images were associated with little textual information, resulting in a less effective approach in the similarity ranking. One way to overcome this problem is to use images with

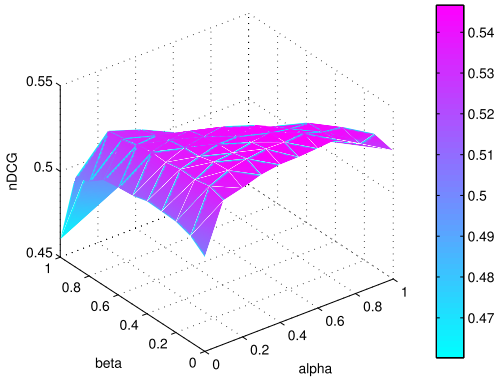


Fig. 6. nDCG plot based on variations of parameters α and β.

TABLE III

AVERAGE nDCG COMPARISON WITH SEMANTIC TREES IN DIFFERENT SHAPES. LEVEL: NUMBER OF LEVELS OR TREE HEIGHT (H), BRANCH: NUMBER OF BRANCHES (B), NORM: THE NORMALIZATION METHOD USED IN (13), AND SCORING: NUMBER OF LEVELS (STARTING FROM THE LEAF NODES) USED FOR SCORING

Run	Level	Branch	Norm	Scoring	nDCG
1	2	100	L1	2	0.621
2	2	100	L2	2	0.660
3	2	500	L2	2	0.667
4	2	1000	L1	1	0.622
5	2	1000	L2	1	0.664
6	**2**	**1000**	**L2**	**2**	**0.677**
7	3	10	L1	2	0.615
8	3	10	L2	2	0.622
9	3	50	L2	2	0.658
10	3	50	L2	3	0.633

ground-truth labels (e.g., the ImageNet [46]), but such data sets lack the social clues. The exploration of other image sources for the semantic tree construction will be considered as a part of our future work.

Considering the three features were extracted from different clues, better results can be obtained by fusing such information with low correlations. Recalling that $\alpha+\beta+\gamma = 1$, we linearly combined the three features and plotted nDCG in Fig. 6 by changing the values of parameters α and β in (7). The highest score of 0.547 was obtained when $\alpha = 0.6$ $\beta = 0.1$ and $\gamma = 0.3$. This is consistent with the reliability of the features, and we kept this setting fixed in the following experiments.

B. Personalized Ranking

We evaluated the effectiveness of our proposed semantic tree in the personalized video segment ranking. The 25 videos we collected from YouTube were further segmented into 751 shots [44]. For each of the 20 users, we ranked the video segments with reference to their personal photo collections. The ground-truth relevance scores were generated in the same way as described in the previous section by manual annotations. Table III gives the average nDCG obtained with different SeTree settings. As the nodes close to the leaves are generally more powerful for feature encoding, improved results have been reported by scoring with only the last two levels of the nodes. Moreover, L_2-norm gives better personalized ranking than L_1-norm as the distance measure in (13). As can be seen, the best ranking result has been reported when $H = 2$ and $B = 1000$, with nDCG equal to 0.677.

TABLE IV

nDCG COMPARISON OF THE METHODS ON RANKING VIDEO SEGMENTS

Methods	Pairwise	BoS_{sa}	BoS_{sc}	SeTree
U1	0.626	0.607	0.569	**0.669**
U2	0.607	0.595	0.657	**0.691**
U3	0.575	0.544	**0.634**	0.491
U4	0.726	0.828	**0.840**	0.796
U5	0.508	0.565	0.527	**0.742**
U6	**0.691**	0.643	0.647	0.576
U7	0.395	0.436	0.554	**0.738**
U8	0.713	0.704	0.710	**0.717**
U9	**0.710**	0.687	0.686	0.662
U10	**0.646**	0.534	0.569	0.468
U11	0.786	0.768	**0.828**	0.753
U12	0.759	0.850	**0.861**	0.771
U13	0.196	0.726	0.658	**0.833**
U14	0.589	0.591	0.595	**0.724**
U15	0.215	0.225	0.421	**0.549**
U16	0.721	0.733	0.634	**0.812**
U17	**0.681**	0.561	0.610	0.564
U18	0.815	**0.851**	0.846	0.837
U19	**0.690**	0.579	0.491	0.631
U20	**0.570**	0.486	0.355	0.520
Avg.	0.611	0.626	0.635	**0.677**

Next, we compared our approach with the following three methods: 1) random; 2) pairwise distance; and 3) bag-of-scene (BoS) signature [32]. *Pairwise* computes the Euclidean distance between every pair of images and uses the average value for ranking. *BoS* generates a dictionary of scenes, each of which represents a specific semantic concept. Next, it assigns frames to one or more visual scenes, followed by a pooling step to generate the final representation. However, one limitation is that it assumes the basis vectors in the codebook to be independent without modeling the semantic relationship among the scenes. We evaluated two advanced coding techniques, namely, the soft assignment (BoS_{sa}) and the sparse coding [19] (BoS_{sc}) with the BoS signature. A dictionary size of 1000 was adopted as a larger codebook did not impact the results much. The average nDCG comparison is given in Table IV and the **best** and the second best results are highlighted. The result of *Random* was computed as the nDCG of a random permutation of the segments. It is provided as baseline that shows the characteristics of experimental data.

As can be seen from the detailed results on each of the users, our proposed approach outperformed the other methods in most of the cases. The *Pairwise* method worked well when users' personal images captured consistent content on one topic. However, this method is time-consuming as it computes the pairwise distance between the high-dimensional visual features of frames. This drawback hinders its utilization in real-time video summarizations. Comparatively, *BoS* and *SeTree* overcome this issue by generating high-level semantic video representations. BoS_{sc} utilizes the sparse coding technique [19], which improves the soft-assignment coding [18] by

TABLE V
f-MEASURE COMPARISON AT 15% SUMMARY LENGTH

	Video Name	Random	Attention [20]	Superframe [21]	ST [9]	STKernel
ego.	Base jumping	0.144	**0.194**	0.121	0.119	0.171
	Scuba	0.138	**0.200**	0.184	0.120	0.180
	Valparaiso Downhill	0.142	0.231	0.242	0.275	**0.277**
moving	Bearpark climbing	0.147	0.227	0.118	0.194	**0.234**
	Bus in Rock Tunnel	0.135	0.112	0.135	**0.147**	0.145
	Cockpit Landing	0.136	0.116	0.172	**0.225**	0.182
	Excavators river crossing	0.144	0.041	**0.189**	0.117	0.139
	Kids playing in leaves	0.139	0.084	0.089	0.073	**0.225**
	Notre Dame	0.137	0.138	**0.235**	0.054	0.135
	Playing on water slide	0.134	0.124	**0.200**	0.038	0.063
	Saving dolphins	0.144	**0.154**	0.145	0.121	0.113
	St Maarten Landing	0.143	**0.419**	0.313	0.319	0.396
	Statue of Liberty	0.122	0.083	0.192	0.141	**0.217**
	Uncut Evening Flight	0.131	0.299	0.271	**0.308**	0.246
	paluma jump	0.139	0.028	**0.181**	0.114	0.115
	playing ball	0.145	0.140	**0.174**	0.155	0.134
static	Air Force One	0.144	0.215	0.318	**0.389**	0.328
	Fire Domino	0.145	**0.252**	0.130	0.220	0.249
	Paintball	0.127	0.281	0.320	**0.353**	0.340
	car over camera	0.134	0.201	0.372	0.356	**0.427**
	mean	0.139	0.177	0.205	0.192	**0.216**

approximating a feature as a linear sum of a sparse set of the basis vectors in the dictionary. As both BoS_{sa} and BoS_{sc} use a codebook of single-level structure, it neglects the conceptual relationship among the visual scenes. Our proposed *SeTree* addresses this issue using a tree structure, and has been verified to be effective as it outperformed BoS_{sa} and BoS_{sc} by 8.1% and 6.6%, respectively.

C. Attention-Based Ranking

We verified our spatiotemporal saliency estimation approach (see Section V-A2) by comparing it with three state-of-the-art methods. *ST* denotes the off-the-shelf technique of space–time saliency detection [9]. The importance of a frame is estimated by averaging the saliency scores of all the pixels. We refer to our approach as *STKernel* to emphasize the KDE and the KLD distance calculation. The other two competitors are based on visual attention [20] and superframe [21], respectively. We carried out experiments on 20 videos from the public data set SumMe introduced in [21]. Based on the camera characteristics, the videos were divided into three categories: static, moving, and egocentric. A group of study subjects were asked to produce video summaries that contain most of the important content, and each video in the data set was summarized by 15 to 18 different people. We selected segments around the top-ranked frames and generated summary with length set to 15% of the input video. Additionally, we report the results of assigning random scores to characterize the data set. Here, we have followed Gygli *et al.* [21] and adopted the pairwise f-measure as the measurement. For each ground-truth generated by humans, we computed precision and recall on a per-frame basis. Subsequently, the f-measure was computed as

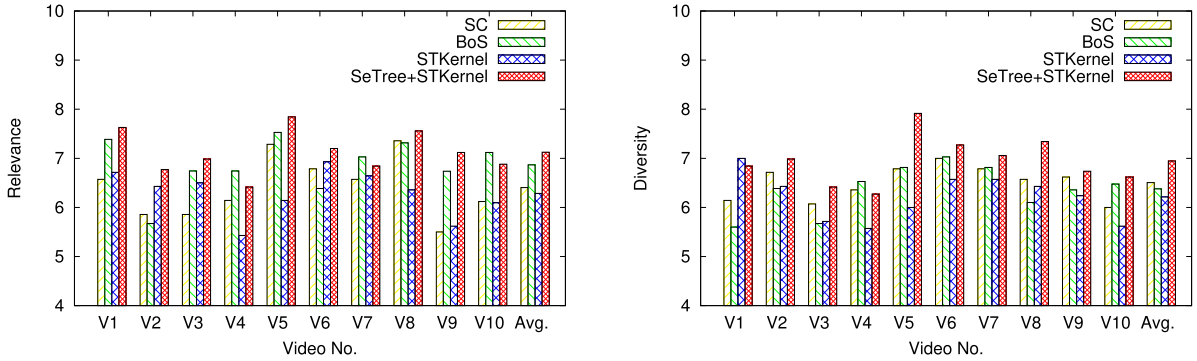
$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Finally, we averaged the f-measures over the ground-truth selected by different people as the final measure for evaluation. The comparison of the f-measures at 15% summary length is given in Table V, where the **best** and second best results are highlighted.

Ejaz *et al.* [20] obtained the static attention score of a frame by averaging the nonzero values in the saliency map. Comparatively, our method *STKernel* extracted the spatiotemporal ROI from each frame. The saliency map was next characterized by a Gaussian kernel to create summary statistics that are less sensitive to the high-frequency noise. In addition to the weighted average of pixel saliency values, we also promote frames where the spatiotemporal ROI is close to the image center. As indicated by Table V, our method improved the f-measure by 22.0% and 12.5%, respectively, compared with the attention-based approach proposed in [20] and the original space–time saliency detection method [9]. Gygli *et al.* [21] segmented videos into superframes and predicted the interestingness by fusing scores of human attention, video quality, presence of landmarks, faces, and objects. To combine the above features, they used a linear model where a great number of parameters needed to be trained. Comparatively, our method only introduces two parameters (the center region of an image characterized by a Gaussian distribution $P = \mathcal{N}(\mu, \Sigma)$) that can be heuristically decided. We speculate *Superframe* would require a comparatively larger processing time than our approach due to the utilization of complex features for video analysis, while our method, however, may have a tradeoff in accuracy for some of the cases, as shown in Table V.

In this experiment, we resized the input frames to 64×64 for space–time saliency map extraction. Subsequently, parameter μ was set to $(32, 32)$, and the center region was defined as a circle with standard deviation set to 5.

Fig. 7. Illustrations of the frame samples selected by our algorithm SeTree+STKernel.



(a) Relevance (b) Diversity

Fig. 8. Subjective evaluation of the performances of summarization schemes. (a) Relevance. (b) Diversity.

D. User Study

We have evaluated the effectiveness of our proposed personalized and spatiotemporal scoring in the previous sections, respectively. Here, we generated ten video summaries for ten users with different interests by combining the above two scores and performed a user study. The duration of the videos varies from 2 min and 2 s to 27 min and 29 s. To illustrate, Fig. 7 shows an example of the summary generated by our approach. In this case, the user's interest is the Golden Gate Bridge in California. The parameter λ in (12) was set to 0.3. Subjective tests were conducted with users to compare the effectiveness of our scheme with the following three approaches.

1) *SC:* A video summarization scheme based on the scene clustering algorithm. The video segments around the keyframes of the top salient scenes are selected [4].
2) *BoS:* An example-based video summarization scheme. The similarity between video segments and query examples is computed based on a compact video representation called BoS [32] with soft-assignment coding.
3) *STKernel:* The importance of a video segment is determined by the visual attention score estimated based on the space–time saliency map [9] with kernel fitting.

For each of the above methods, we selected the top-ranked segments, the total length of which was set to no longer than 15% of the input video. A group of 21 subjects (different from the eight users in the test set) participated in this user study. They were requested to watch the summaries very carefully, and rank the results on a scale of 1–10 (1—the worst and 10—the best) based on the following two aspects: 1) the *relevance* of the summary to the user's personal image collections (i.e., user interest) and 2) the *diversity* of the summary with reference to the video content. To reduce the carryover effect, we randomized the orders of the summaries generated by different methods before presenting them to the participants. The results are illustrated in Fig. 8(a) and (b), respectively.

Method *STKernel* calculated the importance of video segments based on the space–time saliency map estimation as introduced in Section V-A2. This method is capable of extracting the interesting parts of a video. However, the generated summary is not adapted to the user preferences. Such a strategy is suitable for the SumMe data set where the test videos are strongly concentrated on specific topics (object or event). It might not be equally effective for more complex videos that are formed by multiple shots with a longer duration. Comparatively, method *SC* is designed to automatically determine the representative scenes of a video by clustering. The importance of a scene was measured by the number of frames quantized to it. Subsequently, we chose the frame with the maximum membership grade for each cluster as the keyframe, and selected the segments around the keyframes of the top salient scenes to play at the normal speed. Therefore, it can be seen from Fig. 8(b) that *SC* works generally well in terms of maintaining the diversity of the video summaries, but it sometimes may include lengthy but less important scenes such as a person talking in front of the camera. Method *BoS* applied the high-level BoS representation to personalized

TABLE VI
EFFECTIVENESS COMPARISON BASED ON THE AVERAGE SATISFACTION SCORES EVALUATED BY USERS

Method	SC	BoS	STKernel	SeTree+ STKernel
Mean	6.45	6.62	6.25	7.03
Standard Dev.	$\pm$0.51	$\pm$0.58	$\pm$0.50	$\pm$ 0.52

video summarization. It selected the segments that are the most visually similar to the user's personal images as the salient parts. Thereby, it obtained relatively high scores in terms of relevance in the user study.

To select the important shots with personal adaptation, we fused *STKernel* with our proposed *SeTree*. It extracted the visual features with a hierarchical dictionary that encapsulates the conceptual links between scenes to improve the similarity search accuracy. The overall satisfactory of each summary was measured by the average value of the relevance and the diversity. As shown in Table VI, our proposed approach is more effective than its competitors. It achieved an average satisfaction score of 7.03, outperforming the second best method *BoS* by 6.2%. Moreover, we measured the consistency of the user ratings by computing Cronbach's alpha and obtained an average value of 0.7, which is considered to be acceptable for a good test [47]. These results indicate that our method is able to effectively adapt a video's content to personal interests and generate a diverse and satisfactory summary.

VII. CONCLUSION

We presented a visual approach for user profiling with a hierarchical dictionary termed *SeTree*. The feature encoded by *SeTree* is more descriptive than the *BoS* representation as it additionally captures the semantic relationships among the visual scenes in the codebook. To construct the semantic tree, we perform a normalized graph cut clustering by conjunctively utilizing visual, textual, and social clues. Our tentative results show that an improvement of 8.1% compared to BoS with the sparse coding technique was observed with our test data. We predict, however, that higher accuracy can be further obtained by using pictures with less noisy text compared with Flickr images. Next, we apply our proposed model to personalized video summarization. To promote the important segments of a video, we also estimate the attention-based spatiotemporal saliency score by modeling the ROI in a frame with a Gaussian kernel. We linearly fuse the personalized score and the spatiotemporal saliency for video segment ranking. The less important shots will be fast-forwarded in the generated dynamic summary to improve the QoE. In our initial experiments with viewing quality analysis, our results are found to be 6.2% better than the summaries generated by standard methods such as *SC* and *BoS*.

In the future, we will carry out experiments by utilizing other image sources for *SeTree* construction in order to enhance clustering accuracy. To improve its descriptiveness, more advanced coding techniques will be studied and evaluated. Moreover, we will apply our proposed *SeTree* encoding to other applications such as video retrieval and classification to evaluate its effectiveness.

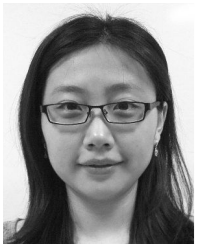
ACKNOWLEDGMENT

The authors would like to thank H. Kashimura and the entire Incubation Center, Fuji Xerox Company Ltd., Kanagawa, Japan, for the continuous support and encouragement. All brand names and product names are trademarks or registered trademarks of their respective companies.

REFERENCES

[1] R. Thapliya and C. Hu, "AdapComm: A bandwidth allocation methodology for multimedia applications in wireless networks," in *Proc. ACM SIGCOMM Workshop Future Human-Centric Multimedia Netw.*, 2013, pp. 27–32.
[2] Y. Zhang and R. Zimmermann, "Efficient summarization from multiple georeferenced user-generated videos," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 418–431, Mar. 2016.
[3] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. Int. Conf. Image Process.*, Oct. 1998, pp. 866–870.
[4] S. S. Bucak and B. Gunsel, "Online video scene clustering by competitive incremental NMF," *Signal, Image Video Process.*, vol. 7, no. 4, pp. 723–739, 2013.
[5] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.
[6] G. Guan, Z. Wang, S. Mei, M. Ott, M. He, and D. D. Feng, "A top-down approach for video summarization," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 1, 2014, Art. no. 4.
[7] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907–919, Oct. 2005.
[8] M. Rodriguez, "CRAM: Compact representation of actions in movies," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3328–3335.
[9] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, p. 15, 2009.
[10] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE CVPR*, Jun. 2012, pp. 1346–1353.
[11] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, and S. Yan, "Static saliency vs. dynamic saliency: A comparative study," in *Proc. ACM Multimedia*, 2013, pp. 987–996.
[12] B. Han, J. Hamm, and J. Sim, "Personalized video summarization with human in the loop," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2011, pp. 51–57.
[13] Y. Zhang, C. Ma, J. Zhang, D. Zhang, and Y. Liu, "An interactive personalized video summarization based on sketches," in *Proc. ACM SIGGRAPH Virtual-Reality Continuum Appl. Ind.*, 2013, pp. 249–258.
[14] Y. Wei, S. M. Bhandarkar, and K. Li, "Video personalization in resource-constrained multimedia environments," in *Proc. ACM Multimedia*, 2007, pp. 902–911.
[15] M. Xu, J. S. Jin, and S. Luo, "Personalized video adaptation based on video content analysis," in *Proc. Workshop Multimedia Data Mining*, 2008, pp. 26–35.
[16] Y. Takeuchi and M. Sugimoto, "Video summarization using personal photo libraries," in *Proc. Workshop Multimedia Inf. Retr.*, 2006, pp. 213–222.
[17] Y. Takeuchi and M. Sugimoto, "User-adaptive home video summarization using personal photo libraries," in *Proc. 6th ACM Int. Conf. Image Video Retr.*, 2007, pp. 472–479.
[18] J. C. van Gemert, J.-M. Geusebroek, C.-M. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proc. ECCV*, vol. 5304. 2008, pp. 696–709.
[19] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. CVPR*, 2009, pp. 1794–1801.
[20] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Process., Image Commun.*, vol. 28, no. 1, pp. 34–44, 2013.
[21] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Proc. ECCV*, 2014, pp. 505–520.

[22] J. You, G. Liu, L. Sun, and H. Li, "A multiple visual models based perceptive analysis framework for multilevel video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 3, pp. 273–285, Mar. 2007.
[23] Y. Peng and C.-W. Ngo, "Clip-based similarity measure for query-dependent clip retrieval and video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 612–627, May 2006.
[24] S. E. F. de Avila, A. P. B. Lopes, A. da Luz, Jr., and A. de Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 56–68, 2011.
[25] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVsum: Summarizing Web videos using titles," in *Proc. CVPR*, 2015, pp. 5179–5187.
[26] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Proc. ECCV*, 2014, pp. 540–555.
[27] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2069–2077.
[28] M. Gygli and H. G. L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proc. CVPR*, 2015, pp. 3090–3098.
[29] J. Almeida, N. J. Leite, and R. da S. Torres, "VISON: Video summarization for online applications," *Pattern Recognit. Lett.*, vol. 33, no. 4, pp. 397–409, 2012.
[30] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *Proc. CVPR*, 2015, pp. 2235–2244.
[31] Y. Yin, B. Seo, and R. Zimmermann, "Content vs. context: Visual and geographic information use in video landmark retrieval," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 3, 2015, Art. no. 39.
[32] O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres, "A visual approach for video geocoding using bag-of-scenes," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2012, Art. no. 53.
[33] Y. Yin, Y. Yu, and R. Zimmermann, "On generating content-oriented geo features for sensor-rich outdoor video search," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1760–1772, Oct. 2015.
[34] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of WWW image search results using visual, textual and link information," in *Proc. ACM Multimedia*, 2004, pp. 952–959.
[35] J. Davidson *et al.*, "The YouTube video recommendation system," in *Proc. ACM Recommender Syst.*, 2010, pp. 293–296.
[36] C. Liu, S. Jiang, and Q. Huang, "Personalized online video recommendation by neighborhood score propagation based global ranking," in *Proc. Internet Multimedia Comput. Service*, 2009, pp. 244–253.
[37] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE CVPR*, Jun. 2006, pp. 2161–2168.
[38] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Object bank: An object-level image representation for high-level visual recognition," *Int. J. Comput. Vis.*, vol. 107, no. 1, pp. 20–39, 2014.
[39] Z. Zhou, Y. Wang, and J. Gu, "A new model of information content for semantic similarity in WordNet," in *Proc. Future Generat. Commun. Netw. Symp.*, 2008, pp. 85–89.
[40] D. Lin, "An information-theoretic definition of similarity," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 296–304.
[41] S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain, "The semantic measures library: Assessing semantic similarity from knowledge representation analysis," in *Proc. NLDB*. Montpellier, France: Springer, Jun. 2014.
[42] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proc. Int. Conf. Data Mining*, 2006, pp. 613–622.
[43] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
[44] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video Manga: Generating semantically meaningful video summaries," in *Proc. ACM Multimedia*, 1999, pp. 383–392.
[45] K.-Y. Cheng, S.-J. Luo, B.-Y. Chen, and H.-H. Chu, "SmartPlayer: User-centric video fast-forwarding," in *Proc. ACM CHI*, 2009, pp. 789–798.
[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE CVPR*, Jun. 2009, pp. 248–255.
[47] M. Tavakol and R. Dennick, "Making sense of Cronbach's alpha," *Int. J. Med. Edu.*, vol. 2, pp. 53–55, Jun. 2011.



Yifang Yin received the B.E. degree from the Department of Computer Science and Technology, Northeastern University, Shenyang, China, in 2011 and the Ph.D. degree from National University of Singapore, Singapore, in 2016.

She was a Research Intern with the Research and Technology Group, Incubation Center, Fuji Xerox Company, Ltd., Tokyo, Japan, from 2014 to 2015. She is currently a Research Fellow with the Interactive and Digital Media Institute, National University of Singapore. Her research interests include geotagged video annotation and retrieval, geometadata correction, and video summarization.

Roshan Thapliya received the B.Eng., M.Eng., and Ph.D. degrees in electronic engineering from University of Tokyo, Tokyo, Japan, in 1994, 1996, and 1999, respectively.

He is a Research and Development Manager with the Research and Technology Group, Fuji Xerox Company, Ltd., Tokyo, the Principal Researcher, and the Project Manager of Intelligent Robotics and Systems. He has led interdisciplinary research groups in optical device/systems and advanced media analytics/delivery technologies. He has authored major OSA, IEEE, ACM, and AIP journals, and holds over 25 patents. His current research interests include physical optics, content analytics, cellular networks, distributed Internet of Things analytics networks, robot group association systems, and robot intelligence.

Roger Zimmermann (S'93–M'99–SM'07) received the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, CA, USA, in 1994 and 1998, respectively.

He is currently an Associate Professor with the Department of Computer Science, National University of Singapore, Singapore, where he is also the Deputy Director with the Interactive and Digital Media Institute and the Co-Director of the Centre of Social Media Innovations for Communities. He has co-authored a book, more than 200 conference publications, journal articles, and book chapters, and holds six patents. His current research interests include streaming media architectures, distributed and peer-to-peer systems, mobile and georeferenced video management, collaborative environments, spatiotemporal information management, and mobile location-based services.

Mr. Zimmermann is a member of ACM.