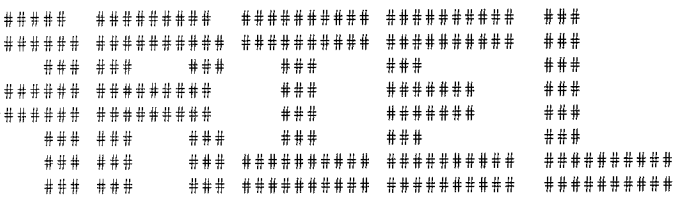


A Novel Video Summarization Based on Mining the Story-Structure and Semantic Relations Among Concept Entities

Bo-Wei Chen, Jia-Ching Wang, and Jhing-Fa Wang, *Fellow, IEEE*

Abstract—**Video summarization techniques have been proposed for years to offer people comprehensive understanding of the whole story in the video. Roughly speaking, existing approaches can be classified into the two types: one is static storyboard, and the other is dynamic skimming. However, despite that these traditional methods give brief summaries for users, they still do not provide with a concept-organized and systematic view. In this paper, we present a structural video content browsing system and a novel summarization method by utilizing the four kinds of entities: who, what, where, and when to establish the framework of the video contents. With the assistance of the above-mentioned indexed information, the structure of the story can be built up according to the characters, the things, the places, and the time. Therefore, users can not only browse the video efficiently but also focus on what they are interested in via the browsing interface. In order to construct the fundamental system, we employ maximum entropy criterion to integrate visual and text features extracted from video frames and speech transcripts, generating high-level concept entities. A novel concept expansion method is introduced to explore the associations among these entities. After constructing the relational graph, we exploit graph entropy model to detect meaningful shots and relations, which serve as the indices for users. The results demonstrate that our system can achieve better performance and information coverage.**

Index Terms—**Concept expansion tree, graph entropy, graph mining, structural video contents, video browsing, video indexing, video summarization.**

I. Introduction

VIDEO summarization aims to generate a series of visual contents for users to browse and understand the whole story of the video efficiently. To date, a great deal of effort has been devoted to providing people with more friendly interfaces and better concept interpretation [1]–[27]. In this paper, we focus on the importance of the semantic presentation of videos. Traditional summarization methods, as illustrated in Fig. 1, usually provide compact static visualization [1]–[12] or concatenated video skims [13]–[18]. In their literatures, the interrelations between shots often rely on visual similarities by extracting salient frames and skipping similar ones. They do not tend to emphasize the relations amongst these shots.

From the perspective of the video summarization, presenting video contents in the form of a structuralized and systematic view is important. Video contents as well as articles are highly organized, and usually these contents can be outlined according to the people involved, where the events took place, the things they related to, and when they happened [28]. Generally speaking, the major difficulties of semantic presentation can be addressed by the following aspects: 1) How to interpret these extracted visual contents in a semantic way. 2) How to calculate the interrelations between these extracted contents. 3) How to organize these contents. 4) How to explore the key points.

To overcome these difficulties, we propose a framework based on the four kinds of entities, "who," "what," "where," and "when," to summarize video contents. Once we extract these entities, we can weave each entity together according to their interrelations and establish a relational graph. The merit of using a relational graph is that it can display indexed entities and relations at the same time.

In this paper, we endeavor to provide: 1) A systematic approach to summarize videos according to the concept entities: "who," "what," "where," and "when." 2) Concept expansion trees and the dependency degree measuring function to discover the relations. 3) The visual presentation based on the relational graph consisting of significant vertices. 4) The strategy to select significant vertices by integrating visual and text information into graph entropy algorithm.

The remainder of the paper is organized as follows: Section II reviews the previous work used in video summarization. Section III elaborates the details of our proposed methods about the system. We exhibit the performance of our system and the experimental results in Section IV. Section V summarizes the conclusions of this paper.

II. Related Work

Depending on the summary type, video summarization techniques can be roughly classified into the two categories: one is static storyboard, and the other is dynamic skimming. The former attempts to extract salient frames in a compact view, whereas the latter offers continuous video fragments, which are playable. In the end of this section, we also investigate the recent researches on browsing and indexing interfaces of video summaries.

Manuscript received April 01, 2008; revised October 07, 2008. Current version published January 16, 2009. This work was supported in part by the National Science Council of the Republic of China under Grant NSC97-2218-E-006-012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jiebo Luo.

The authors are with the Department of Electrical Engineering, National Cheng Kung University, 701 Tainan, Taiwan (e-mail: chenbw@icwang.ee.ncku.edu.tw; wjc@icwang.ee.ncku.edu.tw; wangjf@csie.ncku.edu.tw).

Digital Object Identifier 10.1109/TMM.2008.2009703

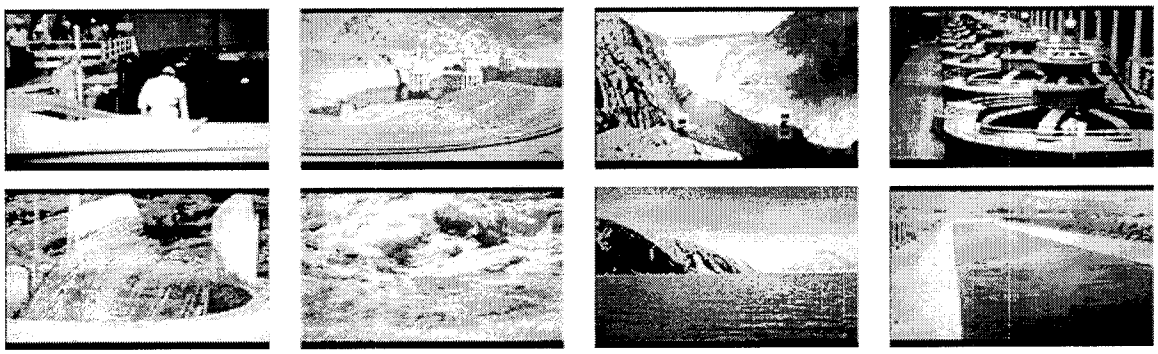


Fig. 1. Examples of the traditional video summarization results.

A. Static Storyboard

Traditional summarization techniques display the representative key-frames according to the designed ranking algorithms. In these methods, ranking algorithms are usually based on maximum frame discrepancy strategies. This means we have to discard similar frames and preserve the frames, which are different from one another and draw attention to people. Therefore, visual features are usually referred to as the principle to compare the relational degree between frames [1]–[4]. In order to shorten the original length to fit the desired time constraint, different shot distortion methods are adopted based on the redundant percentage of frames in shots [2], [5], [6].

Odobez *et al.* [3], [4] utilize the low-level features to compute the similarity matrix between key-frames according to the visual and temporal properties. They employ spectral clustering to analyze the similarity matrix, and similar frames are grouped together based on their feature distances. In another study [7], the algorithm turns to use facial features as the relations among key-frames to classify shots. It can adaptively decide the number of clusters while running spectral clustering. These are key-frame-based clustering approaches. There are also some sophisticated methods applying spatial-temporal relations on video summarizations. Lu *et al.* [2] make use of dynamic programming on the graph structure to find out the optimal skimming length and distribute it to each shot. Their shot-relations are primarily based on spatial-temporal linkages. Recently, more and more researches tend to exploit attention models as the criteria to choose frames, such as [5], which attempts to apply visual attentive indicators to model the relations between scenes. The skim ratio is calculated from the scene level, cluster level, shot level up to subshots level. Cernekova *et al.* [8] use another viewpoint, modeling the shot and scene change by detecting mutual information between frames. The representative key-frame is chosen by the one, which maximizes inter-frame mutual information in the cluster. Li *et al.* [6] proposed MINMAX optimization formulation with viewing time, frame skip and bit rate constraints. They also design new metrics for missing frames and video summary distortions.

In recent researches, text processing plays an important role in video summarization [9]–[11]. They focus on caption and transcript information to judge the boundaries of scenes or stories. Term weighting schemes are employed in their applications. Frames or shots corresponding to higher term weights are then considered as the representative abstracts. The authors [12] discuss the impact on the navigation of videos using various text features.

B. Dynamic Skimming

Although static storyboard summarization can offer users a more comprehensive view, it is susceptible to the smoothness problem, which means users may feel uncomfortable while browsing the results. Dynamic skimming techniques attempt to overcome these problems by concatenating extracted video events into continuous image sequences. The authors [13]–[15] incorporate various attention models with fusion methods to handle linguistic data and generate video clips. Users can grasp more complete information because the duration of the clip matches the verbal sentence length. Li *et al.* [16] turn to take advantage of audiovisual clues and shot-change patterns to extract dialog events. In rush summarization techniques, the authors [17], [18] aim to capture specific motion events, such as fast-moving objects or camera events. Detyniecki and Marsala [18] proposed an adaptive acceleration approach, which is a kind of shot-distortion concepts, to select informative shots.

Generally speaking, smoothness is the major concern in dynamic skimming summarization. The relevant information between these video segments is often based on the attentive curves. However, finding an effective way to transform video data into numeric curves becomes a challenge.

C. Browsing and Indexing Interface of Video Summaries

Although a good video summarization system consists in the algorithm it uses, providing a more friendly and useful browsing interface is indispensable to users. Because video abstracts aim to generate a series of assisting information from videos, a well-designed browsing interface can quickly guide people to access those contents by indexing them.

So far, many applications [19]–[21] deal with the subjects related to the video conferences' minutes or slides of the presentation. The authors [19] create a hierarchical interface, which allows people to identify potentially useful or relevant videos from the database level, single video level, to the key-frame

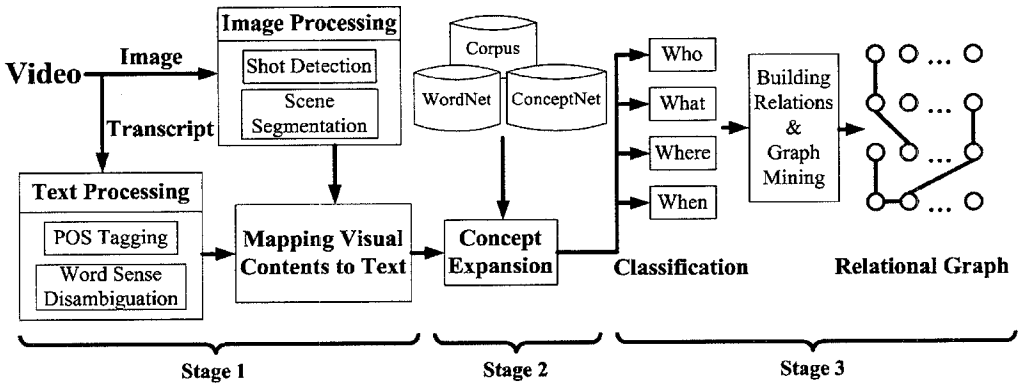


Fig. 2. Proposed system architecture.

cluster level. This tool can provide popup captions, highlighted intervals, and time indices to the selected frames. Haubold and Kender build a navigation system "VAST MM" [20] that supports the indexing of various clues, such as visual speaker lists, topic phases, and thumbnails of presentation videos. It enhances the multi-type searching ability. Another video browser [21] utilizes the table-of-content video mapping technique to search the matching course materials by recognizing the text information in the video. Some researchers [22]–[25] are concerned about indexing news video contents. The common properties of their work are to provide users with news segments along with corresponding news transcripts or documents. Cristel [26] expands the existing storyboard system by establishing a linking graph, where each vertex is the queried result from the relevant videos. It can also display geographic distribution of the locations mentioned in the video set. The authors [27] proposed the cross-referenced video summary, which exploits temporal relations to index related scenes.

III. Video Summarization System

The proposed system architecture is shown in Fig. 2, where the input is a video with its speech transcripts, and the output is a graph composed of important shots. To organize traditional summarization results into a relational graph may involve recognizing the meanings of shots, classifying them into the four entities, and linking these correlated shots together. These tasks are fulfilled at the three stages, of which the first one is to map visual contents to text; the second is concept expansion; the last is to structuralize the video contents. Before we continue to introduce the details of the proposed system, our summarization procedures are outlined below (**to avoid ambiguity, we use the terms *vertices* and *edges* to represent a graph, whereas *nodes* and *links* are used to denote a tree**).

1) Using an annotator to link shots with text information.
2) Classifying all shots into the four entities and building a graph based on these shots.
3) Dividing all shots in the graph into groups, each of which represents a video scene.
4) Finding edges within each scene and between consecutive scenes using concept expansion trees.
5) Exploiting graph entropy algorithm to discover significant shots and relations.

A. Mapping Visual Contents to Text

At this stage, the shots in the video are annotated with keywords. The reason we adopt annotation here is that we wish to organize a relational graph in a semantic way. By means of image annotation, visual relations between shots could be turned into high-level notions. Moreover, we may take advantage of annotations to estimate importance-scores of the vertices in a graph.

1) Visual and Text Content Pre-Analysis: We apply an ensemble of methods proposed in [29]–[31] to detect shot boundaries. Our scene analysis is based on the procedures discussed in [29], [32]: The middle frame of a shot is selected to represent the shot. An intermediate structure "video group" between physical shots and scenes is constructed to serve as the bridge between the two. In this approach, shots with similar properties are combined into a video group, and a video scene is composed of several similar video groups. We list the algorithm of the scene construction in Fig. 3. This algorithm iteratively tests each shot to see which group of the scene is more similar to the current shot. After the scene construction, we exploit the scene-segmentation information to incorporate text and visual data to compute the importance degree within scenes in Section III-C4.

In order to analyze images for annotation at the later stage, we follow the steps mentioned in [33] to extract features. Each frame is divided into 6×6 blocks. A block is denoted as a 23-dimensional feature vector, which is calculated from the color, texture, location, and motion vector of the block: 1) We adopt the HVC color histogram and use its mean and variance as 6-dimensional color features; 2) the texture feature has 12 dimensions. We perform the Gabor filter with six orientations on the block and then choose the mean and variance from the histogram of each direction; 3) the location feature is obtained according to the column and the row index of the block in the frame; 4) motion vectors are estimated, and then we use three bins to describe the normalized directions.

ALGORITHM – SCENE CONSTRUCTION

initialization

designating the first shot as the initial group and scene;

for each shot s

begin

calculating the similarity between shot s and each group g by:

$\text{GrpSim}_{s,g} = \text{ShtSim}_{s,g_{\text{last}}}$;

finding the maximal group-similarity by:

$\text{MaxGrpSim}_s = \arg\max_g \text{GrpSim}_{s,g}$;

determining which group to assign by:

if $\text{MaxGrpSim}_s > GrpThd$, merging s to $g_{\max}$;

else s is a new group;

calculating the similarity between shot s and each scene SC by:

$$\text{ScnSim}_{s,SC} = \frac{1}{\text{numGrp}(SC)} \sum_{g}^{\text{numGrp}(SC)} \text{GrpSim}_{s,g};$$

finding the maximal scene-similairty by:

$\text{MaxScnSim}_s = \arg\max_{SC} \text{ScnSim}_{s,SC}$;

determining which scene to assign by:

if $\text{MaxScnSim}_s > ScnThd$, merging s to $SC_{\max}$;

else s is a single group in a new scene;

end

Fig. 3. Scene construction algorithm, where g is the index of the video group, and SC is the index of the video scene. $\text{ShtSim}_{s,g_{\text{last}}}$ is the similarity between shot s and the most recent shot (g_{last}) in group g. The authors [32] claim that choosing the most recent shot can capture the largest temporal association to the current shot s. The variables, $g_{\max}$ and $SC_{\max}$, are the group and the scene that have the maximal similarity between shot s and themselves, respectively. *GrpThd* and *ScnThd* are the thresholds specified by users according to each video.

After extracting the features, we cluster all the blocks of the frames using X-means algorithm [34] (a K-means-like algorithm. See the discussions in Section IV-D) and assign the label of the centroid to each block. These blocks are denoted as $V = \{v_1, \ldots, v_{|V|}\}$, where v_i is a block, $|\cdot|$ means the size operator, and $i = 1, \ldots, |V|$.

All the speech transcripts corresponding to the training videos are collected (we restrict these transcripts to the same domain; the details are described in Section IV). The videos and their transcripts are manually aligned. We make use of the part-of-speech tagger [35] to label the words. Word sense disambiguation [35] is employed to identify the word sense, which is used for the later stages. In order to collect meaningful keywords with respect to the video topic, we use stopwords to filter unnecessary keywords. All the words except nouns are removed from the transcripts. We denote these keywords as $W = \{w_1, \ldots, w_{|W|}\}$, where w_j is a keyword and $j = 1, \ldots, |W|$.

We download the Wikipedia corpus from the internet (www.wikipedia.org). It contains various articles and is suited to provide us background information to calculate relations between annotations. In order to retrieve the articles related to the videos we summarize, the keywords in the transcript-word set W are sorted according to their word occurrences. The highest 25%–35% of the keywords are preserved (the percentage we use is the approximate lower bound suggested in [36]. It helps to decrease the search time without losing too much discriminating ability). Each of them is used to query the Wikipedia corpus. The queried results are regarded as the articles related to the transcripts. We denote them as our knowledge base B.

2) Maximum Entropy Criterion-Based Annotator: Maximum entropy (MaxEnt) method offers a feasible way to model the co-occurrences between visual contents and text [37]–[39]. Its aim is to generate an annotated output y given a frame image x. The co-occurrence relations between images and their annotations can be learned via the training dataset, V and W. The training data are collected by gathering pair-wise data (x, y), and their association degree can be described by a designed function, which is depicted in the following discussions. Assume that we have the training data, V and W. Let k denote the index of the training pair $(v, w) \in \{V \times W\}$, where v represents a block in V derived from the video frames, and w means a keyword in W. Therefore, the number of total combinations is $|V| \times |W|$. We intend to construct a function that can capture the dependency between the image and its correlated keyword. On referring to [33], the function could be given as

$$f_k(x, y) = \delta_{yw} \times \#(v, x),$$
$$\delta_{yw} = \begin{cases} 1, & \text{if } y = w \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $k = 1, \ldots, |V| \times |W|$, δ is the Kronecker delta function, x denotes an image, y means a word in the transcripts when we scan the training videos, and $\#(v, x)$ is the number of matching blocks regarding v in x. This function defines the degree of the relation when we input words of interest. We then proceed to scan the training video to measure the mutual relation degree from each sample pair.

We exploit the following MaxEnt exponential form, which was proposed by Berger *et al.* [40], to model the visual and the text linkages.

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{k=1}^{|V|\times|W|} \lambda_k f_k(x, y) \right) \tag{2}$$

where $Z(x) = \sum_y \exp(\sum_{k=1}^{|V|\times|W|} \lambda_k f_k(x, y))$, and λ_k is the parameter for $f_k(x, y)$. Once each $f_k(x, y)$ is determined from the samples, λ_k can be estimated by using Generalized Iterative Scaling (GIS) algorithm [40]. Let D denote the keyword set in the transcripts of the video we summarize. When the training procedure is complete, the annotation of a frame is decided by $y^* = \arg\max_{y \in D} P(y|x)$. This equation evaluates an unlabeled image by testing all possible keywords in D, and the keyword with the highest score is referred to as the annotation of the

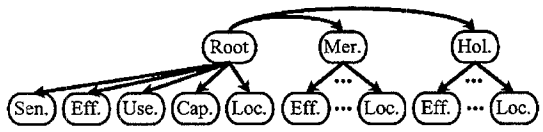


Fig. 4. Structure of the expansion tree. We use ellipses with abbreviations to represent the semantic relations, which we use to query WordNet or ConceptNet. All the ellipses except the root may consist of more than one node. Each child node indicates a correlated keyword expanded from the root. For brevity, we omit the nodes in these ellipses.

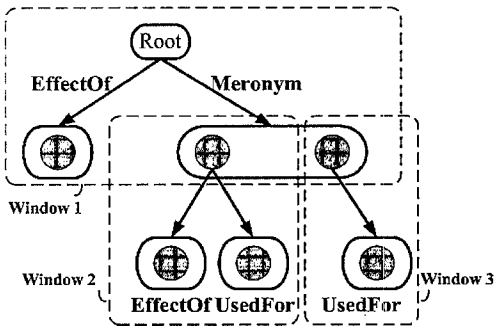


Fig. 5. It shows the partial structure of a concept tree, and some nodes are omitted for the sake of brevity. The ellipse represents the semantic relations we use to query WordNet or ConceptNet. The shading circle denotes one expanded keyword. We use a sliding window of height two, which moves from the top to the bottom of the tree, to calculate the dependency degree between parent and child nodes. Each window is responsible for the computation of the subtree inside.

image. Since we use the middle frame to represent a shot, every shot in the input video can be linked to a proper keyword.

Although multiple annotations would be better for labeling, we use a single keyword instead. The reason is that when there is more than one word belonging to one shot, it is difficult to decide which words we should choose to expand. Furthermore, measuring the relational degree could become more complicated since multiple annotations would result in multiple edges between two shots.

B. Concept Expansion

1) Background: Deciding whether two shots can be linked together when we construct a relational graph depends on the relations of their annotations. Accordingly, we employ concept expansion method, which can be regarded as a refinement to extend the annotations of images. It provides an alterative view and dependency information while measuring the relation of two annotations. Our concept expansion relies on the two knowledge-based lexicons, WordNet [41] and ConceptNet [42]. They are word-reasoning toolkits, which define semantic relations between words. In our work, we only focus on nouns and their semantic relations.

2) Constructing Trees: We use a tree to denote the dependent structure of the expanded words, where the root r represents the annotation of an image. The tree structure is illustrated in Fig. 4. We define the root as the first level, and the height of the tree with only one node is one. Each child node indicates a correlated keyword expanded from r.

While constructing the tree, we retrieve the sense, meronym (parts-of), and holonym (is-a-part-of) words as the child nodes of the root by querying the semantic relations in WordNet. Moreover, we also fetch relative words of r by querying "CapableOf," "UsedFor," "EffectOf," and "LocationOf" in ConceptNet. These relative words are then analyzed according to the text processing procedures stated at the preceding stage, and only nouns and verbs (verbs are converted to verbal nouns) are kept. Besides, those words, which are not relevant to the root, are pruned by querying whether they are in our knowledge base B or not.

3) The Dependency Degree Function: The reason why we conduct the dependency degree function at this level is: When we try to explore the relations between annotations among a graph in Section III-C, those unreliable expanded child nodes can be removed to avoid creating weaker relations. To compute the dependency degree between a parent node and its child node, we use a sliding window of height two started at the root to scan the structure. Consider the subtree rooted at level one in the sliding window. The conditional probability of the leaf node c in this subtree depends on its parent node ρ, the video title T, and the transcript-keyword set D of the video we intend to summarize. Hence, we can describe their dependency according to (3), which is similar to the algorithm proposed in [43], with several changes to support the calculation of the dependency degree:

$$P(c|\rho,T,D) = \frac{P(T|c,\rho)P(D|c,\rho)P(c|\rho)P(\rho)}{\sum_{c_l\in\mathbb{C}} P(T|c_l,\rho)P(D|c_l,\rho)P(c_l|\rho)P(\rho)} \quad (3)$$

where $\mathbb{C}$ represents the collection of the child nodes at the same level, which are obtained by querying the same semantic relations with c, and c_l is a node in $\mathbb{C}$. $P(\rho)$ is calculated based on its term frequency in the Wikipedia corpus. The value of $P(D|c,\rho)$ is decided by the transcript-keywords in D: We choose the highest five terms in D according their word occurrences (the number of selected terms is suggested by the researches [44], [45]. It can reduce the complexity of the search without losing too many relevant documents). Each of them is used to query our knowledge base B. Thus, $P(D,c,\gamma)$ can be determined by querying c and ρ at the same time. We calculate $P(T|c,\rho)$ in the same way. By multiplying the results of two sliding windows illustrated in Fig. 5, we can derive the dependency degree between the root and any child node. Later, in Section III-C, we employ (3) to select child nodes with higher dependency degrees to reduce the complexity while building up relational graphs.

C. Structuralizing Video Contents

In order to structuralize those annotated shots by a relational graph, we have to 1) classify the shots into the proper entities according to their corresponding annotations in WordNet, 2) build vertices in the graph and expand them by their concept expansion trees, and 3) build relational edges by concept expansion trees.

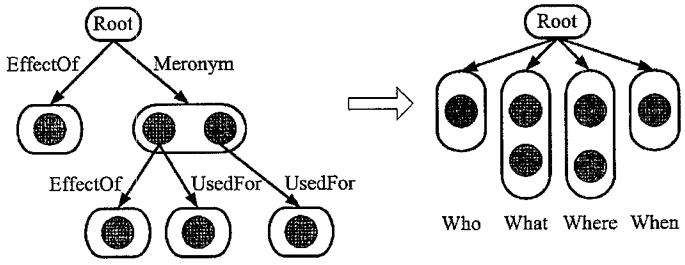


Fig. 6. Example of the reorganization of the tree structure. It displays the classification of child nodes in an expansion tree. The left is a concept expansion tree built in Section III-B, whereas the right is its reorganized structure after classification. All the child nodes adjacent to the root are classified into the four categories. The shading circle denotes one expansion keyword. We exploit this new structure to facilitate the progress of the relational measurement between shots.

1) Annotation Classification Using WordNet: Among all the relations in WordNet, hypernym (is-subset-of) can express the hierarchical semantic relations between two words. Take "teacher" for example: It is in the subset of "person," and this word "person" is one concept category defined in WordNet. Accordingly, "person" is a hypernym of "teacher." With the assistance of WordNet, a word can be classified into a proper entity by querying its hypernym. The following descriptions explain the steps we take in series to classify words.

1) We define that the first entity "who" only includes names of people and those terms in the subset of "person" in WordNet. In other words, such keywords as "teacher," "worker," and "musician" are classified into this entity as long as their hypernyms belong to "person." We make use of a dictionary to assist us to identify names. The words that can not be recognized in this step are passed into the next step.
2) In the second entity "where," we only select the words which belong to one of these three subsets: "social group," "building," and "location."
3) The third entity is "what." All the other words (except the words in the entity "when"), which do not belong to "who" and "where," are classified into this class.
4) The last is "when." This entity is created by simply searching for time-patterns, such as years and any time-period phrase.

2) Building Vertices in the Relational Graph: At the earlier stage, we have built the concept expansion trees. For convenience, we use s to represent a shot, a to represent its annotation, o to represent its concept expansion tree rooted at a, and tuple (s, a, o) to denote them. We proceed to establish the relational graph.

Firstly, we classify all the tuples into the four kinds of entities in accordance with their annotations a. Meanwhile, we build a graph composed of four columns, each of which represents one certain type of entity. When a tuple is assigned to one of the entities, we create a vertex to represent this tuple in the corresponding entity (column).

Secondly, we reorganize each concept expansion tree into a new tree structure. For the concept tree o in every tuple in the graph, all the nodes (except the root) in one tree are also classified into the four kinds of entities. Fig. 6 sketches the example about the reorganization of the tree structure.

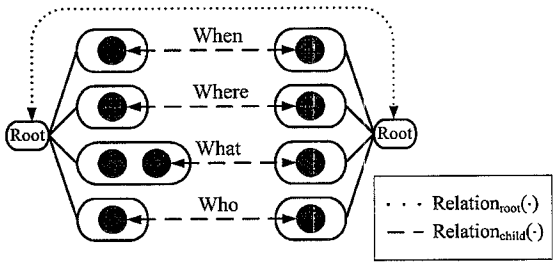


Fig. 7. Relational measurement between two concept trees. The dotted line denotes the measurement between two roots, and the relations between child nodes are represented by dash-lines. The solid lines are the linkages between the root and the child nodes.

3) Building Relations in the Relational Graph: Discovering the relation means to select two tuples (s, a, o) from the graph and use their concept expansion trees o to measure their relations. In short, the associations between two trees are related to the word-comparisons of the roots and their child nodes. Fig. 7 illustrates the detailed information of the comparisons. In this phase, we employ a heuristic estimating function (4) to express the relations of the two trees:

$$\Gamma(\alpha, \beta) = \text{Relation}_{\text{root}}(\alpha, \beta) \times \text{Relation}_{\text{child}}(\alpha, \beta) \quad (4)$$

where α and β are the tuples in the graph. $\text{Relation}_{\text{root}}(\alpha, \beta)$ describes the way to calculate the relations of roots; $\text{Ralation}_{\text{child}}(\alpha, \beta)$ is adopted to measure the relations between two child nodes among these trees. In the following contents, we look more closely at how to assign weights to relational edges.

1) Measuring the relations between two roots:
Since the keyword in the root comes from the transcripts, a better way to find out the relation is utilizing the same transcripts as the knowledge base. This is because they can provide clues that are more straightforward and reliable. The relation of the two roots can be expressed by:

$$\text{Relation}_{\text{root}}(\alpha, \beta) = \frac{\text{sent}_{\alpha,\beta}}{\text{the number of sentences}} \quad (5)$$

where $\text{sent}_{\alpha,\beta}$ uses the roots of the tuples (α and β) to measure the number of sentences containing both of them in the transcripts.
2) Measuring the relations between two child nodes:

In each expansion tree, there are four types of child nodes, as shown in Fig. 7. However, measuring any two nodes between two trees iteratively may cost much complexity. To alleviate this problem, we set a constraint: When the node-types are the same, we could proceed with the measurement. Given two child nodes, the connection is measured by detecting whether their keywords are identical or not. If they are identical, this means we can bridge the two trees via these identical nodes, which also implies the roots are correlated. The relational measurement between the two nodes of the same type can be given as follows:

$$\text{Relation}_{type}(I,J) = \sum_{I,J} \frac{\text{ident}_{I,J}}{\text{the number of pairs}} \quad (6)$$

where I and J are the child nodes of the same type in two different trees, $\text{ident}_{I,J}$ means the number of occurrences when I and J are identical, and *type* represents the child-node type: "who," "what," "where," or "when" in a tree. For instance, assume we have two nodes, node_{11} = student and node_{12} = pupil, in one expansion tree; we also have another node, node_{21} = student, in another tree. All of them belong to the same type "who." Because the total test-pairs are 2 and $\text{node}_{11} = \text{node}_{21}$, the relational degree corresponding to "who" is $1/2$.

In order to explore all possible relations, the four types need to be calculated. Given two tuples α and β, let I denote the child nodes in α, and J denote the child nodes in β. We can rewrite (6) to

$$\text{Relation}_{\text{child}}(\alpha,\beta) = \sum_{type} \left(\sum_{I,J} \frac{\text{ident}_{I,J}}{\text{the number of pairs}} \right)_{\text{type}}. \quad (7)$$

Therefore, the equation above could be used to estimate all the child-node relations.

To reduce the time complexity when calculating the relations of each type, we adopt the dependency degree in the expansion tree, which is depicted in (3), to decrease the number of child nodes. Empirically, we keep the highest two child nodes in each type. When constructing a relational graph, we iteratively select two tuples and connect them with only one edge, the weight of which is then computed according to (4).

There is another time complexity issue. After the construction is finished, the relational graph is a complete graph without any loop edge. To put it another way, it is a fully-connected graph, since each vertex is adjacent to the other vertices in the graph. Nevertheless, if we take all the tuples into consideration, it costs too much computation to determine the weights of edges. A feasible way is to divide all the tuples into several groups according to the video scenes we detect in Section III-A. Thus, the relational measurements are performed: 1) within each scene and 2) between the consecutive scenes, respectively.

4) Mining Important Vertices and Edges: After the structure of the story contents is built up, we extract important vertices and edges from the graph. Mining meaningful vertices in the relational graph is important to video abstracts. This makes it easier for users to understand the main points of the summarization. Since the motivation behind our work is to establish a graph and explore significant subgraphs, the graph mining technique [46], [47] is a practicable way to meet demand. In our system, we conduct graph entropy algorithm [47], which allows us to explore the vertices whose removal has significant impact on the graph. It also helps to analyze paths with various lengths. Let us represent a graph by $G = \langle U, E \rangle$, the vertices by $U(G)$, and the edges by $E(G)$. Conventionally, graph entropy $H(G)$ can be given as follows:

$$H(G) = \sum_{s=1}^{|U|} P(u_s) \log(1/P(u_s)) \quad (8)$$

where u_s is a vertex, $s = 1, \ldots, |U|$, $|U|$ denotes the total number of vertices (shots) in G, and $P(u_s)$ is the probability function of u_s. In Shetty's paper [47], graph entropy is designed to discover a subgraph such that its entropy reaches maximum. High entropy indicates that many vertices are equally important, whereas low entropy indicates that only a few vertices are relevant (weaker relational degree) [48]. The intuition is to find out higher-correlated and significant vertices by checking the entropy difference after we iteratively remove vertices.

The analysis of our relational graph relies on the significance of the vertex and the edge. In a manner similar to the approach in [47], we integrate the vertex (shot) weight into the algorithm when calculating graph entropy. In addition, the importance of edges in the graph is also considered to help search for the important vertices. While calculating the significance of vertices, we take both visual and annotated information into consideration. Because the shot (visual) content as well as text information plays an important role in the summarization, humans may have more comprehensive understanding if we provide visual and text information at the same time. To fully describe the idea, given an annotated shot, we could model the influence degree by using

$$A(u_s) = coef \times \text{VisualAttention}\,(u_s) \times \text{AnnotationWeight}\,(u_s) \quad (9)$$

where $\text{VisualAttention}(u_s)$ returns the visual attractive strength of u_s, $\text{AnntationWeight}(u_s)$ measures the importance of the shot annotation, and $coef$ is a weight specified by users.

An effective method to describe the attractive strength is using the attention model. It focuses on visual perceptron analysis, which utilizes color contrast, luminance, and so forth, to provide numerical processing. In our work, we employ the approach [13], [49] to generate salient maps by analyzing color contrast, intensity contrast, and orientation contrast in each frame of the video. As shown in Fig. 8, salient maps attempt to highlight the regions which attract people.

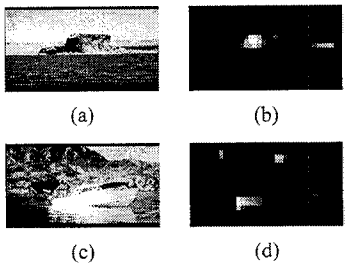


Fig. 8. Examples of the gray-scale salient maps, where each pixel has a brightness value from 0 (black) to 255 (white). (a) and (c) are the original images, and the rest are their corresponding salient maps.

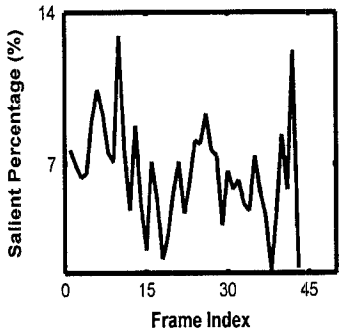


Fig. 9. Attentive curve based on the salient maps of the frames. The horizontal axis denotes the frame index of the video; the vertical axis denotes the salient percentage representing the proportion of the brightness region in the salient map.

ALGORITHM – MINING EDGES

decomposing G into length-one paths;

for each path

begin

calculating graph entropy $H_{\text{edge}}(G)$ according to (12);

obtaining G' by removing the edge of the current path from G;

calculating graph entropy $H_{\text{edge}}(G')$;

calculating the cross entropy $H_{\text{edge}}(G')/\log\left(H_{\text{edge}}(G')/H_{\text{edge}}(G)\right)$;

end

sorting each path's cross entropy;

(a)

ALGORITHM – MINING VERTICES

for each vertex u_s, $s = 1,\ldots,|U|$

begin

calculating graph entropy $H_{\text{vertex}}(G)$ according to (8);

obtaining G'' by removing u_s;

calculating graph entropy $H_{\text{vertex}}(G'')$;

calculating the cross entropy $H_{\text{vertex}}(G'')/\log\left(H_{\text{vertex}}(G'')/H_{\text{vertex}}(G)\right)$;

end

sorting each vertex's cross entropy;

(b)

Fig. 10. (a) Algorithm for mining the important edges in the video summary. (b) The algorithm for mining the important vertices in the video summary.

In order to convert these regions into an attentive curve, we calculate the proportion of the brightness region in each salient map. An example of the attentive curve is illustrated in Fig. 9.

We decompose the attentive curve into segments, and the duration of each is equal to the length of the corresponding shot. Additionally, a histogram with ten bins is also adopted, and the mean value is calculated to generate the attentive degree. Let mean_{u_s} denote the mean of the attentive histogram related to shot s, dur_{u_s} denote the duration of shot s, SC_{u_s} represent the scene which u_s belongs to, $\text{mean}_{\text{SC}_{u_s}}$ denote the mean of the attentive histogram related to SC_{u_s}, and $\text{dur}_{\text{SC}_{u_s}}$ denote the duration of the scene. The visual attractive strength can be expressed using the following formula:

$$\text{VisualAttention}\,(u_s) = \frac{\text{mean}_{u_s} \times \text{dur}_{u_s}}{\text{mean}_{\text{SC}_{u_s}} \times \text{dur}_{\text{SC}_{u_s}}}. \quad (10)$$

On the other hand, to identify the importance of the annotations, we employ TFIDF function to realize AnntationWeight($\cdot$), which could be defined by

$$\text{AnnotationWeight}\,(u_s) = \text{TF}\,(u_s, \text{SC}_{u_s}) \times \text{IDF}(u_s) \quad (11)$$

where $\text{TF}(u_s, \text{SC}_{u_s})$ calculates the frequency of u_s in scene SC_{u_s}, and $\text{IDF}(u_s)$ represents the inverse document frequency. When an annotation appears more frequently, the value of the inverse document frequency becomes lower. Equation (11) is usually used to evaluate the rarity of terms. The inverse document frequency can be defined as $\text{IDF}(u_s) = \log(M)/(\text{DF}(u_s))$, where M is the total number of scenes in the video, and $\text{DF}(u_s)$ is the document frequency, which represents the number of scenes involving the annotation of u_s. Substituting (10) and (11) into (9) yields the complete form of computing the vertex weight in the graph. To estimate the edge importance, we decompose the relational graph into several paths of length one. A path $\overline{u_m u_n}$ is composed of two vertices, u_m and u_n $(m \neq n)$, and a connected edge. Besides, the number of total paths is equal to the number of edges $|E|$. The entropy of these paths can be defined by modifying (8) into the following equation:

$$H(G) = \sum_{m,n=1}^{|U|} P(\overline{u_m u_n}) \log(1/P(\overline{u_m u_n})), m \neq n \quad (12)$$

and $P(\overline{u_m u_n})$ can be determined by combining (4) and (9) into one equation:

$$P(\overline{u_m u_n}) = A(u_m)\Gamma(u_m, u_n)A(u_n), \quad (13)$$

which estimates the co-occurrence between two vertices. We use (9) to express the weight of a vertex, and the relational weight is represented by (4). When vertex u_m is isolated in the graph, its $P(u_m)$ is simply determined by $A(u_m)$. The algorithms shown in Fig. 10 describe the steps we take to discover the important vertices and edges. Before running graph entropy algorithm, we have to decide the total number of important vertices N. Fig. 10(a) shows the algorithm to find out the edges. After filtering the edges, we apply the other algorithm shown in Fig. 10(b) to calculate the remnant graph.

TABLE I
TRAINING VIDEO DATA

Genre	Total lengths	$\|W\|$	$\|V\|$	B
Buildings	274 minutes	1244	120	65133
Ocean	306 minutes	1064	89	100330
Wildlife	302 minutes	1135	100	66151
War	319 minutes	1498	136	71485
Ancient history	318 minutes	1110	99	72699
Space	400 minutes	1211	93	73784
Crime science	313 minutes	1300	125	68000
Art	300 minutes	1189	128	64580

We list the number of transcript-keywords in the third column. The number of block clusters, $|V|$, is determined by running the clustering analysis software, Weka (www.cs.waikato.ac.nz/ml/weka/). For the time complexity issue, the algorithm we choose in Weka is X-means algorithm [34] (using the default parameters). It can automatically decide the number of centroids. The last column is the number of documents in the knowledge bases with respect to each genre.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of our system, we select documentary videos as our experimental data. Documentary videos have several advantages over movies and news videos: In our observations, speech transcripts of documentary videos are highly dependent on the shots because they are usually involved intensive postproduction and editing. Moreover, in each video clip or shots, the terms are highly organized and dedicated to the visual presentation, whereas transcripts in movies are usually too brief, abstract, or even irrelevant to frames. In most cases, transcripts in movies are composed of dialogues, which may contain many meaningless words or pronouns. Besides, their visual effects are usually too complicated and involve the styles of the directors, which may make it difficult to summarize. As for news videos, although the transcripts are more dedicated, they usually include different domain video-segments and have wide diversity. The lengths of these video segments may be too short, which does not provide sufficient training information and degrades the performance.

We choose eight genres of documentary videos: Buildings, ocean, wildlife, war, ancient history, space, crime science, and art as our test data. It is noted that different domain videos are trained separately to avoid low precision while we annotate videos. Henceforth, when selecting training videos and annotating test ones, we only choose the videos in the same domain. These test videos are downloaded from the Open Video Project website (www.open-video.org), and some are collected from the Discovery Channel, British Broadcasting Corporation (BBC), and National Geographic Channel documentaries. Table I summarizes these videos. The first column is the types of videos. Column two shows the total lengths of these videos. The number of keywords, block clusters, and documents in the knowledge bases are listed from column three to column five, respectively.

TABLE II
TEST VIDEO DATA

No.	Subject	Genre	Length
I-1	Dam	Buildings	30 minutes
I-2	Bridge	Buildings	45 minutes
II-1	Sea	Ocean	51 minutes
II-2	Sea	Ocean	51 minutes
III-1	Caribou	Wildlife	57 minutes
III-2	Bear	Wildlife	47 minutes
IV-1	World War II	War	50 minutes
IV-2	World War II	War	50 minutes
V-1	Egyptian	Ancient history	54 minutes
V-2	Mummy	Ancient history	52 minutes
VI-1	Planet	Space	50 minutes
VI-2	Planet	Space	50 minutes
VII-1	Crime	Crime science	55 minutes
VII-2	FBI	Crime science	48 minutes
VIII-1	World art	Art	60 minutes
VIII-2	World art	Art	60 minutes

Table II shows the information of the test videos. We use numbers to represent our test videos in column one.

A. Concept Expansion and Word Relations

In order to validate the effectiveness of the expanded words and the relations between annotations, we design an experiment that exploits the information retrieval technique. The desire is to query a set of articles by using two expanded terms in different trees. If using expanded words retrieves more correlated documents than using non-expanded ones does, we could prove the feasibility of the concept expansion. This is because when concept expansion explores more unknown-relations between entities, their association degree may become stronger.

We collect 200 documents for each test video in Table II. Each document is selected by humans, and it is correlated to the corresponding test video. Therefore, there are 3200 (200×16) articles. In the next step, 50 pairs of words (each pair only contains two different words) are extracted from the transcripts of each test video. These word pairs are chosen according to their high occurrences in the transcripts. Totally, 800 pairs of words are selected. When testing each pair of words, we only search in the 200 documents that belong to the same test video as the query words do.

In this experiment, the assessment is to check the number of queried results before and after the concept expansion. The average number of searched documents is 58 before we expand each pair of words. The searched results using concept expansion are shown in Fig. 11. The vertical means the number of

TABLE III
DETAILED QUERIED RESULTS AFTER CONCEPT EXPANSION

	1.0E-1	1.0E-2	1.0E-3	1.0E-4	1.0E-5	1.0E-6
The average number of queried documents after expansion	48.5	24.5	11.4	7.7	3.8	1.5
The maximal number of queried documents after expansion	89	41	28	13	13	8
The average number of documents that are not queried before expansion	26.4	11.1	5.8	2.7	2.5	0.7
The maximal number of documents that are not queried before expansion	66	25	9	8	8	5

This table shows the numeric data mentioned in Fig. 11. The first row is the dependency degree calculated according to (3). We use scientific notations to represent dependency degrees. The mantissa (coefficient) is omitted, and the remainder is the order of magnitude. Row two and three list the average and maximal number of queried documents after concept expansion, respectively. The rest rows are the number of documents that are not retrieved before concept expansion.

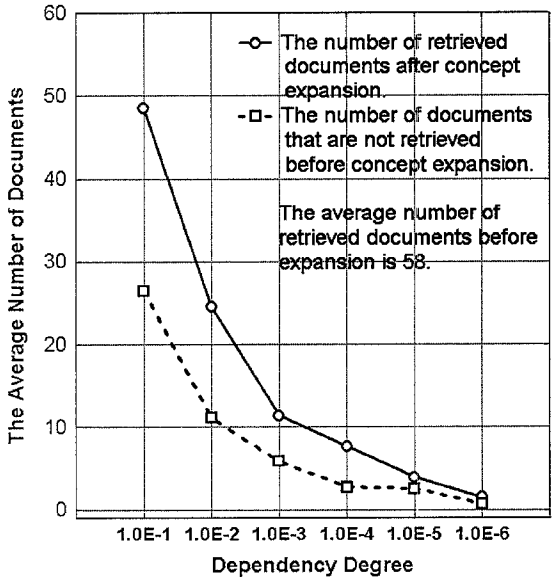


Fig. 11. Average number of retrieved documents using expanded words. We use scientific notations to represent dependency degrees. The mantissa (coefficient) is omitted, and the remainder is the order of magnitude. The circle means the number of queried documents after concept expansion. The square is the number of documents that are not retrieved before concept expansion. In other words, the square $= |\{\text{docs after expansion}\}| - |\{\text{docs after expansion}\} \cap \{\text{docs before expansion}\}|$. Besides, the average number of documents before expansion is 58.

documents, and the horizontal is the dependency degree calculated according to (3).

This figure reflects the discrepancy between whether the two words are expanded or not. If the two words in the pair are both expanded, the average number of queried articles would be increased by at least 24.23%. These detailed data can be referred to Table III. On the other hand, the number of queried articles represents the co-occurrence strength between the words in the pair. Thus, this also indicates that our concept expansion approach can assist in creating relational edges while we construct graphs.

B. Informativeness and Interrelation

In our proposed system, each test video can be summarized into a relational graph. In order to observe its influence on humans, we use *informativeness* and *interrelation* [14]–[16] to evaluate the results. The criterion *informativeness* measures whether the result can bring users as much information as the original video does or not. *Interrelation* means the mutual relation between the selected shots. Empirically, we found that preserving 10%–20% of the total shots in the test videos yields satisfactory performance. Accordingly, we choose the first Nth important shots and the first C_2^Nth important edges for the users. To compare the performance with our proposed method, we implement two storyboard systems as the baselines. Their detailed settings are shown in Table IV. Baseline I is a simple version of our proposed system. The major difference is that although its important shots are calculated using annotations and our proposed algorithm, we remove all the contextual information, including annotations, edges, and entity tags, from its result at the evaluation stage. The goal is to test the text influence on users. As for the other storyboard system, baseline II, we replace the annotation method with the word importance (WI) estimation. Besides, we calculate the cosine-similarity score between the literal information of two shots using the vector space model (VSM) to measure the relations. Both two methods, the WI and VSM, are widely used in the traditional information retrieval area [50]. The significant shots can be chosen according to the ranks of $\text{VisualAttention}(u_s) \times \text{WI}(u_s)$, where $\text{WI}(u_s)$ denotes the word importance function. Each shot in baseline II is endowed with a keyword, and this word is selected from the corresponding transcripts. The word importance is estimated by

$$\text{WI}(u_s) = \arg \max_{\varepsilon} \log(P(\varepsilon)^{-1}), \tag{14}$$

where ε is the word in the transcripts along with shot u_s, and $P(\varepsilon)$ is derived from the term frequency of ε in the transcripts.

We evaluate *informativeness* between the two baselines and our method. In our experiments, we invite ten users, and each of them is asked to give a score ranging from 1 (worst) to 100 (best) to the criterion. After browsing the video summaries, they can watch the original video and write down their decisions. We follow the same procedure to assess *interrelation* between shots in the graph. Users can give scores ranging from 1 (worst) to 5 (best) to the criterion.

TABLE IV
EXPERIMENTAL SETTINGS

		Baseline I	Baseline II	Proposed
Algorithm settings	Type	Storyboard	Storyboard	Storyboard
	Shot-selection algorithm	Visual attentive degree, annotation weight, and graph entropy	Visual attentive degree, word importance, and sorting	Visual attentive degree, annotation weight, and graph entropy
	Using transcripts	Yes	Yes	Yes
	Annotated shots	Yes	No	Yes
	Entity classification	No	No	Yes
	Concept expansion	No	No	Yes
	Shot-relations	No	VSM	Concept expansion
Evaluation settings	Providing keywords to users	No	Yes (selected by the word importance)	Yes (annotation)
	Providing original transcripts to users	No	Below the corresponding shots	Below the corresponding shots
	Interrelations between shots	N/A	Yes	Yes

TABLE V
EVALUATION RESULTS

Video No.	*Informativeness* (%)			*Interrelation*	
	Proposed	Baseline II	Baseline I	Proposed	Baseline II
I-1	84.6	80.1	72.2	4.1	1.7
I-2	82.9	75.5	70.3	4.5	1.7
II-1	83.3	79.1	76.2	4.3	1.9
II-2	84.0	82.5	75.9	3.9	1.6
III-1	83.5	80.6	68.0	3.1	1.9
III-2	80.5	71.3	65.0	3.0	1.5
IV-1	75.8	71.0	67.3	4.1	2.0
IV-2	76.8	70.8	71.2	3.9	2.9
V-1	85.4	79.3	74.7	2.3	1.4
V-2	85.9	75.2	76.0	2.9	1.3
VI-1	81.1	75.9	64.4	3.3	1.5
VI-2	82.3	80.7	79.7	3.7	2.5
VII-1	74.7	63.3	59.0	2.3	1.1
VII-2	70.0	60.0	58.2	2.5	1.1
VIII-1	79.8	74.6	64.7	3.2	2.1
VIII-2	81.6	68.5	62.7	3.0	2.6
Average	80.8	74.3	69.1	3.4	1.8
Drop	19.2	25.7	30.9	1.6	3.2

The table is the evaluation results in Section IV-B. The *informativeness* score ranges from 1 (worst) to 100 (best), whereas the *interrelation* score is from 1 (worst) to 5 (best).

Table V lists the results of the video summaries. Among all the test data, we get better outcomes when summarizing most of the videos. Their average *informativeness* and *interrelation* scores are 80.8% and 3.4. However, we observe that some of the results degrade, such as video VII-1 and VII-2. This is because these videos contain scenes that are more complicated. The narrations comprise technical terms and notions, which need more descriptive words to elaborate the meanings. It is difficult for the users to grasp the whole concept of the summary, only depending on the relational graph and snippets of annotations. Their *interrelation* scores also reflect the phenomena. On the other hand, compared with the two baselines, the performance of our system can be seen obviously in column one of Table V. The *informativeness* difference between the two baselines and ours reaches 6.5% and 11.7% on average. Such results may imply that our summary has better interpretive ability to summarize stories than the others do.

The detailed comparison results of *informativeness* can be referred to the box plot in Fig. 12(a). The smallest rectangle in the plot denotes the mean, and the standard deviation is represented by the large rectangle. H-shape denotes the range from the minimum to the maximum value. The horizontal axis represents the test video number; the vertical represents the *informativeness* score. The distribution of the users' scores about *informativeness* is manifested in this figure [see Fig. 12(a)]. We observe that the variations of the users' opinions about the other two kinds of video summaries are higher than those opinions about our summaries. Fig. 12(b) presents the distribution of the scores with respect to *interrelation*. As can be seen, most of the scores are above 3. The *interrelation* scores become higher if the relations between adjacent shots are closer. The users prefer our results due to the rationality of the interrelationships. In summary, judging from the evaluations in Fig. 12, we may see that baseline I and II yielded limited information about the story, but our system has better organizing ability.

Figs. 13–15 illustrate the examples of the summarized results using the two baselines and our system, respectively. For the sake of clarity and better viewing, we use a seven-minute segment of video I-1 containing 131 shots to demonstrate the results. Only the most important vertices of the summaries are shown in these figures. Fig. 13 is the control group that is used to assess the impact of text information. The results of baseline II can be seen in Fig. 14. Their relations are measured by the VSM, and each keyword below the shot is selected by the WI function. We illustrate the results of our system in Fig. 15. Fig. 15(a) is

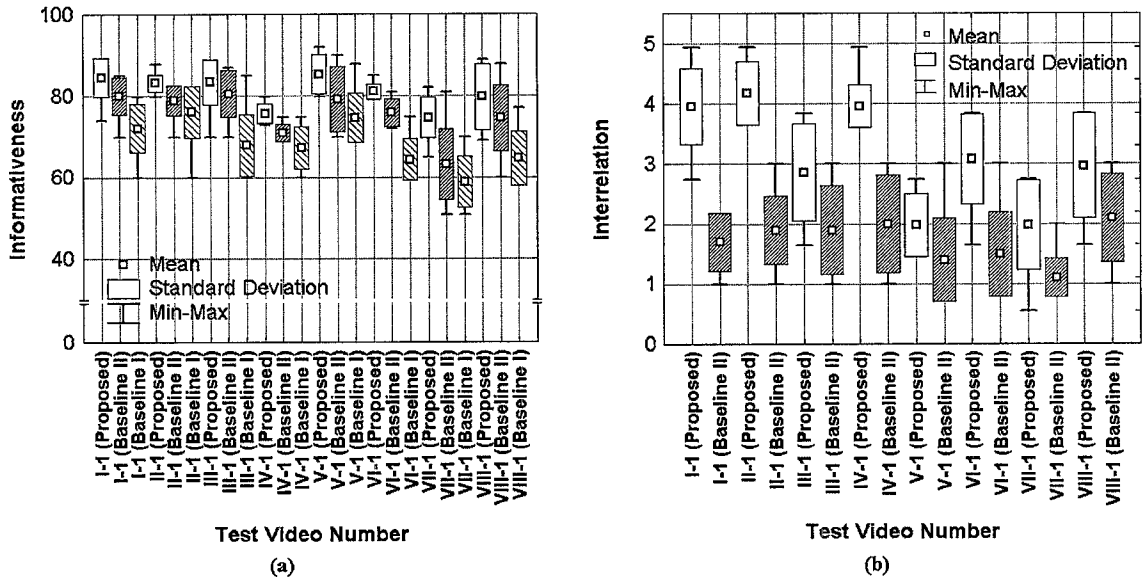


Fig. 12. (a) *Informativeness* comparison results between the baselines and our relational graphs. Each user can give a score ranging from 1 (worst) to 100 (best) after browsing the video summaries. (b) *Interrelation* comparison results between baseline II and our relational graphs. The score ranges from 1 (worst) to 5 (best). The horizontal axis represents the test video number and each system; the vertical represents the score.

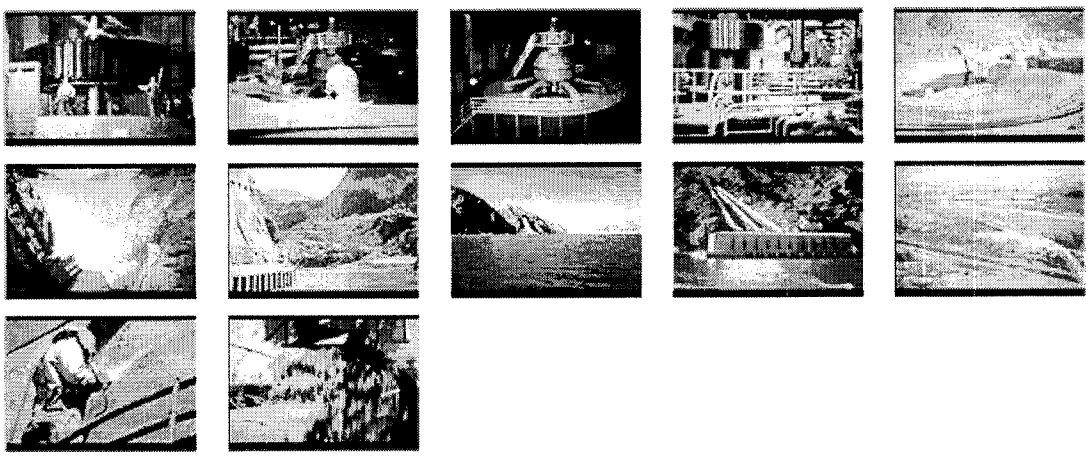


Fig. 13. Summarized results using baseline I (the settings of this algorithm can be referred to Table IV). The shots are arranged according to the time index (from left to right; from top to bottom). The goal of baseline I is to assess the influence of the contextual information on humans. Thus, we remove those data from its results in contrast to the proposed results.

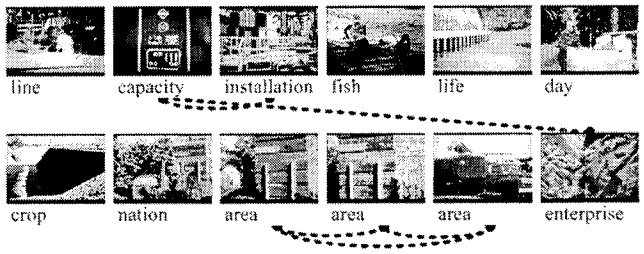


Fig. 14. Summarized results using baseline II (the settings of this algorithm can be referred to Table IV). The shots are arranged according to the time index (from left to right; from top to bottom). The dotted line represents that two shots are correlated, and it is measured by the cosine-similarity score between corresponding transcripts using the VSM. These transcripts are listed in Table VI. There is one interesting observation related to the VSM: It does not demonstrate its feasibility to measure correlated shots because we found that the average number of words corresponding to a shot is fewer than 3.5. This makes it difficult to compute cosine-similarity scores between shots. Accordingly, the relational edges are fewer than those edges in our proposed results.

the diagram of the inter-connections between each entity, and the connections between shots in the entity "what" are shown in Fig. 15(b). In contrast to the results in Fig. 14, we observe that the keywords chosen by the annotation scheme are more

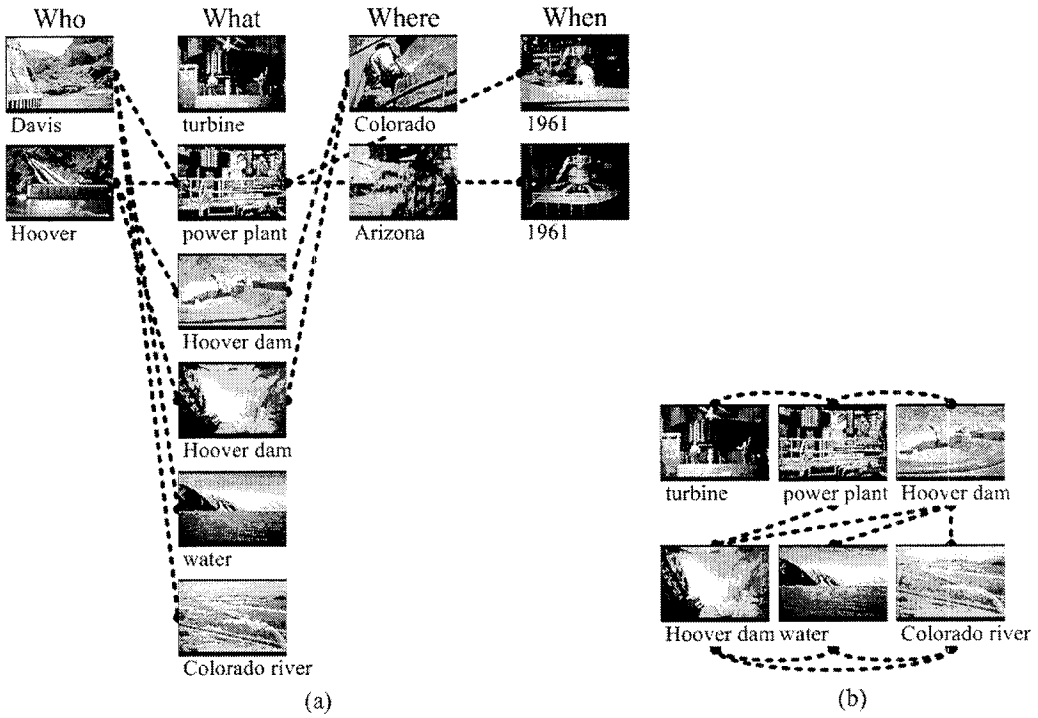


Fig. 15. (a) Inter-connections between entities in our relational graph. The dotted line denotes the connection between two shots, which means they are highly correlated. The tags below the shots are the words correlated to the stories or the contents of the shots. This video segment contains 131 shots totally (before summarization), and we only display some important shots for the sake of clarity and better viewing. (b) This diagram shows the detailed connections between the shots among the entity "what." Their transcripts are listed in Table VI. Compared to the two baselines in Figs. 13 and 14, users can get more information from our relational graph. Moreover, from the viewpoint of the information indexing, our relational clues and keywords bring more benefits to users.

reasonable and suitable for humans. Additionally, these annotations reflect more information about the topic of the video. The advantage of using concept expansion can be seen in Fig. 15. Our system offers more associations between each entity, while baseline II cannot. More discussions about the performance of baseline II and ours are depicted in the next subsection.

C. Contextual Information Among Shots

The contextual information among shots may consist in the low-level features of images, the semantic meanings, the annotations we give, the corresponding transcripts, and so forth. In order to evaluate whether contextual information helps with indexing or not, we employ the four metrics (depicted later) to query related shots in the dataset using a given shot. This dataset consists of different shots from the test videos. To derive the dataset, we pick 30 relevant shots (from every test video), each of which has the following contextual data: the annotation, the significant word determined by the word importance function, and the corresponding transcripts. It is noted that the keywords of the selected shots do not repeat for more than five times in the same test video (to avoid biased searched results while we use keywords to search the dataset). Each time, we select one shot from the dataset and employ the following four metrics to find the relevant shots: 1) Low-level features: This method uses the low-level features extracted from shots according to the steps mentioned in Section III-A1 and computes the similarity between shots based on the Euclidean distance. 2) Low−level features + VSM: In addition to the first one, the similar shots can be determined by calculating cosine-similarity scores between the transcripts of shots. 3) Low−level features+VSM + WI: This approach attempts to search the dataset by making use of the important word selected by the WI function. If the important words of the shots are the same, then a matching shot is found. 4) Low−level features + Annotation + Concept expansion: Instead of using the important words of the shots, we adopt annotation and expanded words to search the dataset.

After the search is complete, we take the top ten shots and check whether they are correlated to the query or not. Fig. 16 is the searched result that shows the partial queried examples using the different four metrics (we have totally 80 queries; some are omitted for the sake of better viewing). The vertical axis represents the number of relevant shots queried by the metric, and the horizontal is the metric we use. In Fig. 16, we observe that most of the curves are rising, which implies that the fourth metric has the better outcome. This accords with the results shown in Fig. 17 and Table VII. In contrast to the other three measurements, our proposed method (the fourth one) has improved the performance by 38% (versus the third one, which is the kernel of baseline II) and 53% (versus the first). There is another interesting observation related to the second metric "Low−level features + VSM." It does not demonstrate its feasibility of searching correlated shots because we found that the average number of words corresponding to a shot is fewer than

3.5. This makes it difficult to compute cosine-similarity scores between shots.

D. The Adaptation of the Proposed System to the Traditional Storyboard

Our proposed system can adapt itself to the conventional one by changing some parameters. The traditional storyboard often uses a technique that removes redundant or less important shots from the video sequences. To achieve such results, we may divide the shot sequences into several parts according to a sliding time-window. Each part could contain various numbers of shots, and the bordering ones are partially overlapped. The next step is to apply graph entropy algorithm on each part, where (9) can be modified to $A(u_s) = \text{VisualAttention}(u_s) \times \text{AnnotationWeight}(u_s)$ or $A(u_s) = \text{VisualAttention}(u_s)$, depending on the existence of transcripts and annotations. Since traditional methods usually do not emphasize the relations, we may skip the procedures for estimating the relational edges. Unlike the traditional ones, our system cannot determine the shot distortion ratio (the number of frames that have to be discarded in a shot) as aforementioned in the related work. This is because our system uses a fixed key-frame to represent a shot. However, our system still can change the number of important shots, which are presented to users, by manually setting a desired number.

In contrast to those static video summarization approaches that do not require the training phase, the computational complexity at our training phase mainly consists in the two parts: one is X-means algorithm, which is used to cluster the blocks, and the other is maximum entropy criterion.

X-means algorithm: According the analysis in [34], [51], it requires two kinds of operations for each iteration.

1) Running K-means algorithm to convergence: The authors [34] speed up K-means algorithm by embedding the dataset into a multiresolutional kd-tree [52] and storing sufficient statistics at its nodes. The computational complexity for a naive K-means algorithm is $\mathrm{O}(K \times Dim \times |V| \times |W|)$, where K is the number of clusters, Dim is the dimensions of the data, and $|V| \times |W|$ is the number of data. With these acceleration modifications, the complexity in this step reduces to $\mathrm{O}(Dim)$.
2) Determining the number of clusters by continuously splitting existing cluster centroids and comparing K-means results using the Bayesian information criterion [53]. The complexity of this step is $\mathrm{O}(\log K_{\max})$, where $K_{\max}$ is a large number specified by users.

Accordingly, by combining the two results mentioned above, the complexity of X-means is $\mathrm{O}(Dim \times \log K_{\max})$.

Maximum entropy criterion: This algorithm exploits Generalized Iterative Scaling algorithm to estimate the parameter λ_k in (2). On referring to [54], its complexity for an iteration depends on the number of parameters λ_k, the dimensions of the data, and the number of data. Consequently, the step needs $\mathrm{O}(|V|^2 \times |W|^2 \times Dim)$.

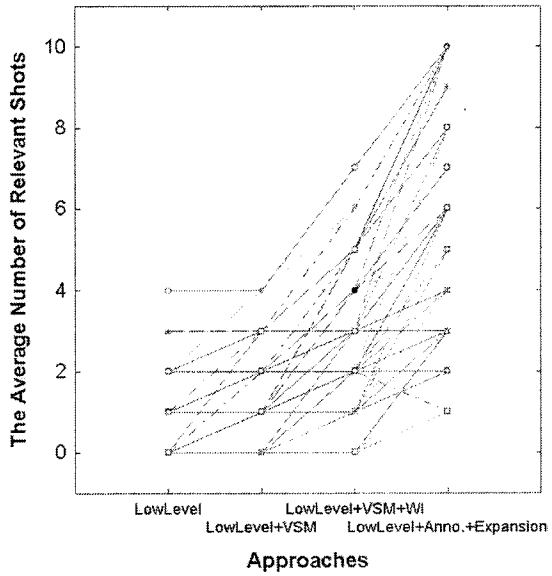


Fig. 16. Searched results using different search strategies. This figure only shows the partial cases for the sake of better viewing (we have totally 80 queries). The vertical axis represents the number of relevant shots queried by the metric, and the horizontal is the metric we use. The detailed information of each metric can be referred to Section IV-C. This figure mainly conveys that the fourth metric has better outcomes than the others do.

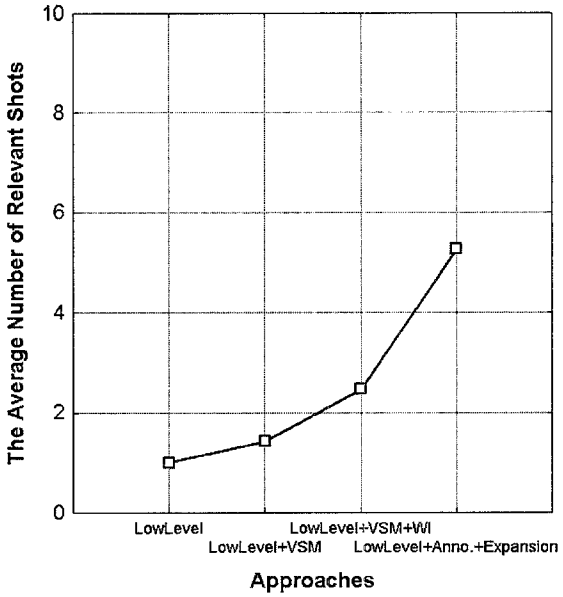
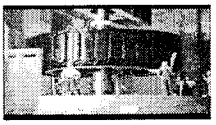
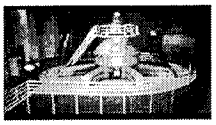
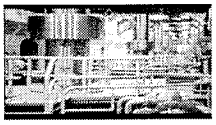
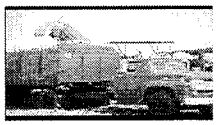
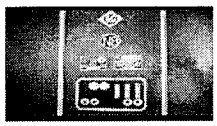
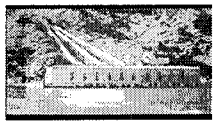


Fig. 17. Average number of searched results using different search strategies. The vertical axis represents the average number of relevant shots queried by the metric, and the horizontal is the metric we use. The detailed information of each metric can be referred to Section IV-C.

E. Discussions About the Browsing and Indexing Interface

Since the functionality of the video summary browsing interface is to offer people the indexed events efficiently, the following descriptions use an example to demonstrate our system.

TABLE VI
EXAMPLES OF THE RANKED RESULTS

Baseline II			Proposed		
Rank	**Shot**	**Corresponding transcripts**	**Rank**	**Shot**	**Corresponding transcripts**
1		...and has required ten days to make the trip...	1		...and lowered inside the stator. The rotor was then joined to the turbine water wheel...
2		...are shipped to dinner tables across the nation...	2		After test runs, N Eight went on the line December first, nineteen sixty-one...
3		The non-surplus food, fiber, and forage crops...	3		After test runs, N Eight went on the line December first, nineteen sixty-one...
4		...these irrigated areas buy farm machinery and other products from the manufacturing centers...	4		...raising its capacity to one and one-third million kilowatts...
5		...these irrigated areas buy farm machinery and other products from the manufacturing centers...	5		As the last sounds of construction faded into history Hoover Dam had cost one hundred seventy-five million dollars...
6		...has helped develop America's free enterprise prosperity.	6		Hoover Dam had cost one hundred seventy-five million dollars. Less a deferred payment of twenty-five million dollars allocated to flood control...
7		...raising its capacity to one and one-third million kilowatts...	7		Lying calmly behind the dam, these waters await need by downstream users. Water is released through the Hoover power plant turbines in a year-round flow...
8		...these irrigated areas buy farm machinery and other products from the manufacturing centers...	8		Sixty-seven miles downstream, Davis Dam re-regulates the Colorado's flow...
9		...to picnic, go boating, swim, fish...	9		...electrical energy from Parker and Hoover power plants...
10		...raising its capacity to one and one-third million kilowatts...	10		...oldest irrigation development on the Colorado River...

This table shows the top 10 important shots in the segment of video I-1 summarized by baseline II and our proposed system, respectively. The first column is the ranking, the second one is the shot, and we list its corresponding transcripts in the last column.

Fig. 15 illustrates the summary of the test video I-1. The topic of video I-1 is about Hoover Dam and the Colorado River, and it mainly describes the construction and the benefits the reclamation brings. In comparison to the original story topic, our re-

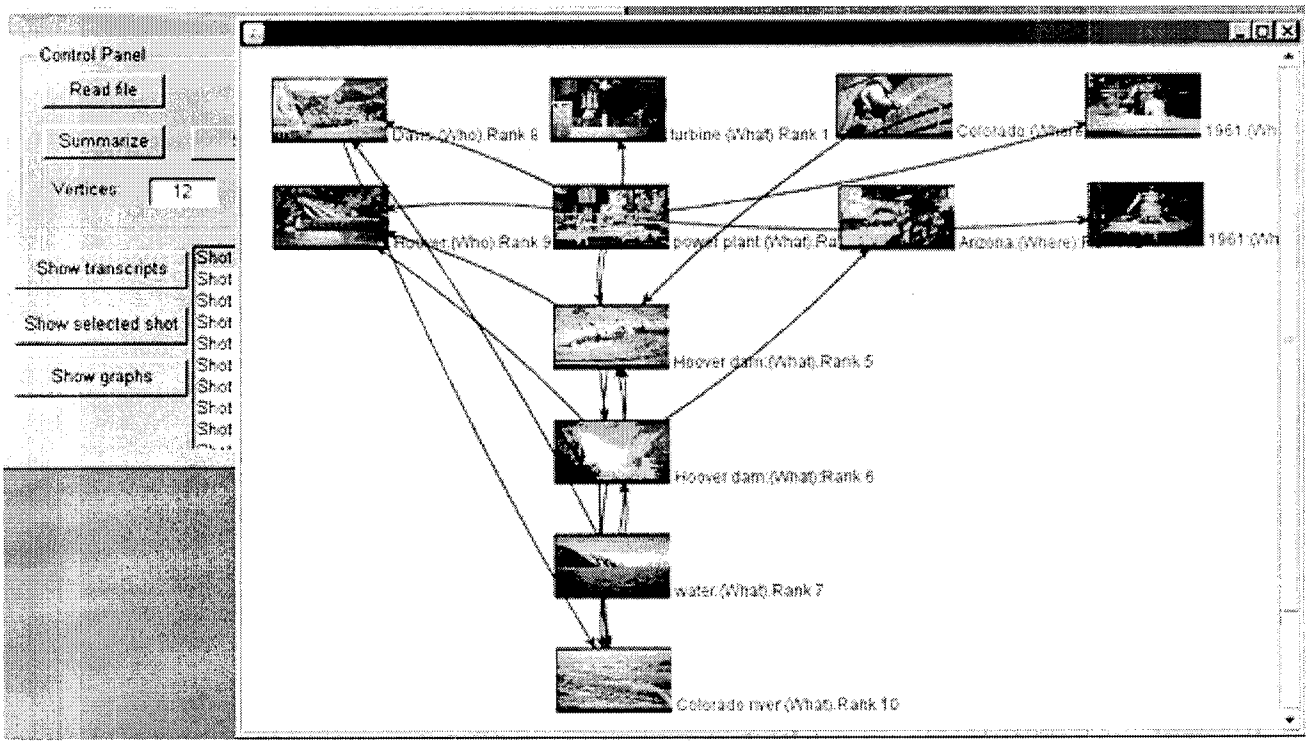


Fig. 18. System snapshot.

TABLE VII
QUERIED RESULTS USING THE DIFFERENT METRICS

	Low-level	Low-level + VSM	Low-level + VSM + Word importance	Low-level + Annotation + Concept expansion
Average number	1.0	1.4	2.5	6.3
Maximal number	4	4	7	10

The first row denotes the metrics we employ. We list the number of relevant shots, which are queried by each approach, in column two and three. The detailed information of each metric can be referred to Section IV-C.

lational graph is capable of summarizing the outlines. We may view the results in Figs. 13 and 14 as the contrast to Fig. 15. One advantage of our relational graph is that as long as users select shots of interest in the graph, the system can offer the other clues, which are correlated with the selection. Thus, users can simply click on the shots and browse their corresponding video segments to obtain complete information. This makes users to comprehend the story efficiently. For instance, in Fig. 15, if a certain user wants to learn about the relevant information with respect to "1961," our system can provide with the correlated clues "power plant" and "turbine." These terms exactly represent one event in the video because "1961" is the year when the engineers activate the power plant. However, the other two approaches cannot provide users with such information. As shown in Fig. 14, the indexing ability of baseline II is not better than ours. This point has been proved in the experimental results in Figs. 16 and 17. In our observations, we found that the keywords indexed in shots in our proposed method are more correlated to the shots and their transcripts. This reflects that the content of the shot is more consistent to the index terms. From the viewpoint of the video indexing, it is much easier for people to navigate the summary.

V. CONCLUSION

In this paper, we present a novel summarization system that exploits concept expansion trees and the mining scheme to construct a relational graph. We design concept expansion trees incorporated with the dependency degree function to support the relational exploration of the graph, and we integrate visual and context weights into graph entropy algorithm to discover the significant shots of the summary. Our proposed method not only structuralizes the video contents but also discovers important shots and their semantically correlated shots in the video. By indexing these semantic clues, people can follow the guided relations in the graph to browse the other ones, which are associated with the selected shots of interest. Thus, with this system and its interface, they can efficiently grasp the comprehensive structure of the story and understand the main points of the contents. The simulation also tells us the performance of the system, which can achieve 80.8% and 3.4 on *informativeness* and *interrelation*, respectively. Compared with baseline II, we improve *informativeness* by 6.5% and *interrelation* by 32% on average. In the concept expansion experiment, we demonstrate the effectiveness of the expanded words. The result shows the average number of queried articles can be increased by at least 24.23%. The contextual information test is to assess the indexing ability

while people search for things of interest. Our results outperform the other two baselines and improve the performance by 38% (versus baseline II) and 53% (vs. baseline I), respectively. These experiments also indicate the organizing ability of our system and prove the necessity of additional indexed information while we summarize the videos.

REFERENCES

[1] 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.

[2] S. Lu, I. King, and M. R. Lyu, "A novel video summarization framework for document preparation and archival applications," in *Proc. 2005 IEEE Aerospace Conf.*, Big Sky, MT, Mar. 5–12, 2005, pp. 1–10.

[3] J.-M. Odobez, D. Gatica-Perez, and M. Guillemot, "Spectral structuring of home videos," in *Proc. 2003 ACM Int. Conf. Image and Video Retrieval*, Urbana, IL, July 24–25, 2003, pp. 310–320.

[4] J.-M. Odobez, D. Gatica-Perez, and M. Guillemot, "Video shot custering using spectral methods," in *Proc. 3rd Int. Workshop on Content-Based Multimedia Indexing*, Rennes, France, Sep. 22–24, 2003, pp. 94–102.

[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, 2005.

[6] Z. Li, G. M. Schuster, and A. K. Katsaggelos, "MINMAX optimal video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1245–1256, 2005.

[7] K. A. Peker and F. I. Bashir, "Content-based video summarization using spectral clustering," in *Proc. 2005 Int. Workshop on Very Low Bit-Rate Video-Coding*, Sardinia, Italy, Sept. 15–16, 2005.

[8] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 82–91, 2006.

[9] A. Bagga, J. Hu, J. Zhong, and G. Ramesh, "Multi-source combined-media video tracking for summarization," in *Proc. 16th IEEE Int. Conf. Pattern Recognition*, Quebec, QC, Canada, Aug. 11–15, 2002, pp. 818–821.

[10] C. M. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. J. Delp, "Automated video program summarization using speech transcripts," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 775–791, Aug. 2006.

[11] X. Zhu, J. Fan, A. K. Elmagarmid, and X. Wu, "Hierarchical video content description and summarization using unified semantic and visual similarity," *Multimedia Syst.*, vol. 9, no. 1, pp. 31–53, 2003.

[12] M. G. Christel and A. S. Warmack, "The effect of text in storyboards for video navigation," in *Proc. 2001 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 7–11, 2001, pp. 1409–1412.

[13] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *Proc. 10th ACM Int. Conf. Multimedia*, Juanles-Pins, France, Dec. 1–6, 2002, pp. 533–542.

[14] 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, 2007.

[15] 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.

[16] Y. Li, S.-H. Lee, C.-H. Yeh, and C.-C. J. Kuo, "Techniques for movie content analysis and skimming: Tutorial and overview on video abstraction techniques," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 79–89, 2006.

[17] B. T. Truong and S. Venkatesh, "Generating comprehensible summaries of rushes sequences based on robust feature matching," in *Proc. Int. Workshop on TRECVID Video Summarization*, Bavaria, Germany, Sep. 23–28, 2007, pp. 30–34.

[18] M. Detyniecki and C. Marsala, "Video rushes summarization by adaptive acceleration and stacking of shots," in *Proc. Int. Workshop on TRECVID Video Summarization*, Bavaria, Germany, Sept. 23–28, 2007, pp. 65–69.

[19] A. Girgensohn, J. Boreczky, and L. Wilcox, "Keyframe-based user interfaces for digital video," *Computer*, vol. 34, no. 9, pp. 61–67, 2001.

[20] A. Haubold and J. R. Kender, "VAST MM: Multimedia browser for presentation video," in *Proc. 6th ACM Int. Conf. Image and Video Retrieval*, Amsterdam, The Netherlands, July 9–11, 2007, pp. 14–18.

[21] L. Tang and J. R. Kender, "Designing an intelligent user interface for instructional video indexing and browsing," in *Proc. 11th ACM Int. Conf. Intelligent User Interfaces*, Sydney, Australia, Jan. 29–Feb. 1 2006, pp. 318–320.

[22] D. C. Gibbon, "Generating hypermedia documents from transcriptions of television programs using parallel text alignment," in *Proc. 8th Int. Workshop Continuous-Media Databases and Applications*, Orlando, FL, Feb. 23–24, 1998, pp. 26–33.

[23] J. Graham and J. J. Hull, "Video paper: A paper-based interface for skimming and watching video," in *Proc. 2002 IEEE Int. Conf. Consumer Electronics*, Auckland, New Zealand, Dec. 3–6, 2002, pp. 214–215.

[24] N. Katashi, O. Shigeki, and Y. Mitsuhiro, "Annotation-based multimedia summarization and translation," in *Proc. 19th Int. Conf. Computational Linguistics*, Taipei, Taiwan, Aug. 24–Sep. 01 2002, pp. 1–7.

[25] J.-Y. Pan, H. Yang, and C. Faloutsos, "MMSS: Multi-modal story-oriented video summarization," in *Proc. 4th IEEE Int. Conf. Data Mining*, Brighton, U.K., Nov. 1–4, 2004, pp. 491–494.

[26] M. G. Christel, "Supporting video library exploratory search: When storyboards are not enough," in *Proc. 2008 ACM Int. Conf. Content-Based Image and Video Retrieval*, Niagara Falls, ON, Canada, Jul. 7–9, 2008, pp. 447–456.

[27] A. Aner, T. Lijun, and J. R. Kender, "A method and browser for cross-referenced video summaries," in *Proc. 2002 IEEE Int. Conf. Multimedia and Expo*, Lausanne, Switzerland, Aug. 26–29, 2002, pp. 237–240.

[28] S. Bageshree, S. Hari, and X. Lexing, "Modeling personal and social network context for event annotation in images," in *Proc. 2007 Conf. Digital libraries*, Vancouver, BC, Canada, Jun. 18–23, 2007, pp. 127–134.

[29] Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring video structure beyond the shots," in *Proc. 1998 IEEE Int. Conf. Multimedia Computing and Systems*, Austin, TX, Jun. 28–Jul. 1 1998, pp. 237–240.

[30] B. T. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *Proc. 8th ACM Int. Conf. Multimedia*, Marina del Rey, CA, Oct. 30–Nov. 3 2000, pp. 219–227.

[31] T.-H. Tsai and Y.-C. Chen, "A robust shot change detection method for content-based retrieval," in *Proc. 2005 IEEE Int. Symp. Circuits and Systems*, Taoyuan, Taiwan, May 23–26, 2005, pp. 4590–4593.

[32] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for videos," *Multimedia Syst.*, vol. 7, no. 5, pp. 359–368, 1999.

[33] A. Velivelli and T. S. Huang, "Automatic video annotation by mining speech transcripts," in *Proc. 2006 IEEE Int. Conf. Computer Vision and Pattern Recognition*, New York, NY, June 17–22, 2006, pp. 115–122.

[34] D. Pelleg and A. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters," in *Proc. 17th Int. Conf. Machine Learning*, Standord, CA, Jun. 29–Jul. 2 2000, pp. 727–734.

[35] S. Patwardhan, S. Banerjee, and T. Pedersen, "SenseRelate: TargetWord—A generalized framework for word sense disambiguation," in *Proc. 43rd Annu. Meeting of the Association for Computational Linguistics*, Michigan, MI, Jun. 25–30, 2005, pp. 73–76.

[36] S.-S. Kang, "Keyword-based document clustering," in *Proc. 6th Int. Workshop on Information Retrieval with Asian Languages*, Sapporo, Japan, Jul. 7–12, 2003, pp. 132–137.

[37] J. Argillander, G. Iyengar, and H. Nock, "Semantic annotation of multimedia using maximum entropy models," in *Proc. 2005 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Philadelphia, PA, Mar. 18–23, 2005, pp. 153–156.

[38] S. Gao, J.-H. Lim, and Q. Sun, "An integrated statistical model for multimedia evidence combination," in *Proc. 15th ACM Int. Conf. Multimedia*, Augsburg, Germany, Sep. 25–29, 2007, pp. 872–881.

[39] W. H.-M. Hsu and S.-F. Chang, "Generative, discriminative, and ensemble learning on multi-modal perceptual fusion toward news video story segmentation," in *Proc. 2004 IEEE Int. Conf. Multimedia and Expo*, Taipei, Taiwan, Jun. 27–30, 2004, pp. 1091–1094.

[40] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Computat. Linguist.*, vol. 22, no. 1, pp. 39–71, Mar. 1996.

[41] G. A. Miller, "Nouns in WordNet: A lexical inheritance system," *Int. J. Lexicography*, vol. 3, no. 4, pp. 245–264, 1990.

[42] H. Liu, "Unpacking meaning from words: A context-centered approachto computational lexicon design," in *Proc. 4th Context Int. Interdisciplinary Conf. Modeling and Using Context*, Stanford, CA, Jun. 23–25, 2003, pp. 218–232.

[43] A. Hoogs, J. Rittscher, G. Stein, and J. Schmiederer, "Video content annotation using visual analysis and a large semantic knowledgebase," in *Proc. 2003 IEEE Int. Conf. Computer Vision and Pattern Recognition*, Madison, WI, Jun. 18–20, 2003, pp. 327–334.

[44] L. Khan, D. McLeod, and E. Hovy, "Retrieval effectiveness of an ontology-based model for information selection," *Int. J. Very Large Data Bases*, vol. 13, no. 1, pp. 71–85, 2004.

[45] A. Hulth and B. B. Megyesi, "A study on automatically extracted keywords in text categorization," in *Proc. 21st Int. Conf. Computational Linguistics and 44th Annu. Meeting of the Association for Computational Linguistics*, Sydney, Australia, Jul. 17–18, 2006, pp. 537–544.

[46] L. Singh, M. Beard, L. Getoor, and M. B. Blake, "Visual mining of multi-modal social networks at different abstraction levels," in *Proc. 11th Int. Conf. Information Visualization*, Zürich, Switzerland, July 4–6, 2007, pp. 672–679.
[47] J. Shetty and J. Adibi, "Discovering important nodes through graph entropy the case of Enron email database," in *Proc 3rd ACM Int. Conf. Link Discovery*, Chicago, IL, Aug. 21–25, 2005, pp. 74–81.
[48] R. Navigli and M. Lapata, "Graph connectivity measures for unsupervised word sense disambiguation," in *Proc. 20th Int. Joint Conf. Artificial Intelligence*, Hyderabad, India, Jan. 6–12, 2007, pp. 1683–1688.
[49] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Netw.*, vol. 19, no. 9, pp. 1395–1407, 2006.
[50] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[51] B. Upcroft, S. Kumar, M. Ridley, L. L. Ong, and H. Durrant-Whyte, "Fast re-parameterisation of gaussian mixture models for robotics applications," in *Proc. 2004 Australasian Conf. Robotics and Automation*, Canberra, Australia, Nov. 11–30, 2004.
[52] M. D. Berg, M. V. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*, 2nd ed. Berlin, germany: Springer-Verlag, 2000.
[53] A. D. R. McQuarrie and C.-L. Tsai, *Regression and Time Series Model Selection*. Singapore: World Scientific, 1998.
[54] L. Tan and D. Taniar, "Adaptive estimated maximum-entropy distribution model," *Inform. Sci.*, vol. 177, no. 15, pp. 3110–3128, 2007.

Bo-Wei Chen received the B.S. degree in computer science from National Cheng Chi University (NCCU), Taiwan, in 2003, and the M.S. degree in computer science and information engineering from the National Cheng Kung University (NCKU), Tainan, Taiwan, in 2004. He is currently pursuing the Ph.D. degree in the electrical engineering at NCKU. His research interests include speech signal and multimedia processing.

Jia-Ching Wang received the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1997 and 2002, respectively.

His research interests include signal processing and VLSI architecture design. He is an honor member of Phi Tau Phi.

Dr. Wang is a member of the ACM and the Institute of Electronics, Information and Communication Engineers (IEICE).

Jhing-Fa Wang received the B.S. and M.S. degrees from the Department of Electrical Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan, in 1973 and 1979, respectively, and the Ph.D. degree from the Department of Computer Science and Electrical Engineering, Stevens Institute of Technology, Hoboken, NJ, in 1983.

He is currently a Chair Professor at NCKU. He developed a Mandarin speech recognition system called Venus-Dictate, known as a pioneering system in Taiwan. He is currently leading a research group of different disciplines for the development of "Advanced Ubiquitous Media for Created Cyberspace." He has published about 91 journal papers and 217 conference papers, and has obtained five patents since 1983. His research areas include wireless content-based media processing, image processing, speech recognition, and natural language understanding.

Dr. Wang was an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS. He was invited to give the keynote speech at the 12th Pacific Asia Conference on Language, Information and Computation, Singapore, and has served as the General Chairman of the 2001 International Symposium on Communication, Taiwan. He received outstanding awards from the Institute of Information Industry in 1991 and the National Science Council of Taiwan in 1990, 1995, and 1997, respectively.