Key Issues in Video Summarization and Its Application

Yueting Zhuang; Ruogui Xiao; Fei Wu

Institute of Artificial Intelligence of Zhejiang University
Zhejiang University, Hangzhou, P. R. China, 310027

yzhuang@cs.zju.edu.cn rouger@163.com wufei@cs.zju.edu.cn

Abstract: The summarization of video content provides an effective way to speed up video browser and assist for video retrieval, this paper proposes a novel method for automatically summarizing multi-view video contents. Three kinds of content views such as semantic-face, video caption and visual features are presented in this paper: First, the human faces in video and images bear significant semantics, thus we can summarize video content by indexing and analyzing video faces, especially for video news; Second, video caption could be used to generate video summarization with high-level semantics since it implied lots of semantics inherently; Finally, the similar video shots could be integrated into concise representation through clustering and mining of visual features.

Keywords: Video summarization, video semantic face, support vector clustering

1. Introduction

A video summarization is a compact representation of video content. How to concisely and informatively summarize video content is extremely important for large-scale multimedia browser and retrieval. For instance, it provides an effective way to quick overview of the video database, fast access to users’ interested video shots and episodes, browsing and retrieval video content.

There are a number of approaches proposed for automatically creating video summaries. The existing video summarization methods can be classified into two categories: key-frame extraction and highlight detection. Selecting a set of key frames as video summary is the most common approach and a great number of key frame extraction methods [1,2] have been proposed. Key frames can help the user identify the desired video shots, but they are insufficient to help the user obtain a general idea of whether the created summary is relevant or not. Video highlight [3,4] uses pre-trained model to extract video events and reduce a long video into a short sequence. Video highlight helps users to determine whether a video is worth viewing in its entirety. However, Video highlights only contain pre-trained video events and loss other video content. Nowadays, automatic generation of video summarization has been applied to various applications, for example, sports video, news video, home video and movies [4,5].

Usually users would browser video content according to different views such as characters, caption and visual features. For examples, users would like to look all of content related to certain characters or visually similar video shots. In this paper, we propose a novel method for automatically generating multi-view video summarization. Three kinds of content views are presented in this paper: video semantic-face, video caption and visual features. First, since the human faces in video and images bear significant semantic contents, thus we can summarize video content by video faces. Second, video caption could be used to generate video summarization with high-level semantics since it implied lots of semantics inherently. Finally, low-level video visual features are used to cluster and mine video content.

The paper is organized as follows: Section 2 presents semantic-face for video content analysis; In Section 3 we present video caption extraction for video content index; In Section 4, we discuss using visual features for video content clustering and mining. Conclusion and future work are given in last section.

2. Semantic-face for video content analysis

People are one of the most important contents in video news and human faces are the most distinct features of people. So using human faces to summarize video content is a possible and meaningful approach.

In traditional video structuring algorithms [6], key frame’s visual features such as color histogram, are used to classify shots and detect anchorperson shots. These visual features, however, are low-level features and do not carry semantic contents. Some other algorithms [7] use the human faces in video news to label video news shots. Although human faces are high-level semantic features, but not all the faces in video news are important and semantic. Only some important faces like semantic-face in video news carry important semantic contents. We define semantic-face as below: Semantic-face is obverse and legible and distinct from the environment. Semantic-face includes anchorperson face, interviewed person face and main news character face. Usually, Semantic-face is a main carrier of semantic contents in video news.

2.1 A hierarchical semantic-face detection algorithm

The basic idea of human face detection is to model human face based on statistics or prior physiology knowledge, and then compares the similarity between a new face and the model to find the possible face area. These traditional approaches detect only the presence and location of faces. They cannot tell whether the faces are legible or important. So they cannot be used to detect semantic-face.

We propose a hierarchical semantic-face algorithm based
on the observation that semantic-face is usually legible and obverse. The hierarchical semantic-face detection algorithm includes three layers. The RGB true color image is turned into YCbCr color space and a face complexion model is used to filter out non-face area coarsely. Then a classify machine based on support vector machine and independent component analysis (SVM/ICA) architecture is trained to detect face patterns in the complexion area. Finally, a semantic-face template is applied to all candidate faces to exactly filter out non-semantic-faces (In Figure 1)

Layer I: complexion model coarse filter
Because RGB color model mixes hue and intensity together, the intensity will affect the detection results the most. We choose YCbCr color space and define the complexion model as the two-dimension Gaussian distribution of the two hue factors Y, Cb, and Cr.

Layer II: face detection based on SVM/ICA
Our face detection algorithm uses such SVM/ICA architecture. We use human faces’ color features in layer one, the face complexion areas are firstly converted into gray-level images. Then we use SVM/ICA architecture to detect presence and location of the face.

Statistics shows that the ratio of height and width of human face is roughly $4/3$. So we travel through each candidate image block with this proportion in certain step to get all the possible face sub-blocks. Then all the sub-blocks’ ICA features are extracted and input into the SVM classifier. Accordingly, the output of SVM, we can find out whether the human face exists and its location. If there are more than one candidate faces within an image block, the SVM output can be converted into probability result through a sigmoid function. We choose the candidate face with the highest probability as detected result.

Due to the interference of background color, the output area of complexion model sometimes contains extra surface around the human face. True faces are hard to detect if detection algorithm is only applied to the full output area. So, in the candidate complexion area, the face sub-block must be adjusted smaller and repeat the detection process until the sub-block size reaches a low limit or face is detected.

Layer III: semantic-face detection based on semantic-face template
Candidate faces have been detected and their positions have been confirmed in layer II. All the candidate faces, however, have the same semantic importance because we do not know which face is more significant. We cannot distinguish between semantic-face and non-semantic-face, and there may also be mistakes in our detection result in layer II. In this layer, semantic-face template is applied to measure each candidate face’s semantic importance and produce the final results of semantic-face detection.

As discussed above, semantic-face’s major characteristic is obverse and legible. When a face is obverse or legible, its facial organs should be legible first. The facial organs such as mouth, nose, left eye and right eye in semantic face must be obvious and legible. The semantic-face template is based on detection of these facial organs.

Semantic-face template is made up of two sub-templates, facial organ distribution probability template (FODPT) and facial organ likelihood probability template (FOLPT). The FODPT is a two-dimension Gaussian distribution of each facial organ in the two-dimension human face plane based on statistics results of a great amount of semantic-faces; the FOLPT uses the SVM/ICA to detect a image sub-blocks’ probability of being a certain facial organ. Both of the two sub-templates deal with four facial organs: mouth, nose, left eye and right eye.

3. Video caption extraction for video content indexing
Video caption could also be used to generate video summarization with high-level semantics since it implied lots of semantics inherently. For example, captions in news broadcasts and documentaries usually annotate information of the reported events. There are two kinds of text in videos, video caption and scene text. Scene text is part of the environment and captured by the camera along with the rest of scene. Because video caption contains more semantic information than scene text, the objective of our algorithm is to locate the caption text.

3.1 Extraction of Video Caption Features
Pixels of an image are not independent of themselves. Deploying the correlation between them is a way to reduce the computation cost. Observing the fact that strokes of characters are usually arranged in eight directions, that is, east, southeast, south, etc and caption text itself form a rather homogenous region, we can just extract a particular set of pixels to speed up processing. Here, we divide an image into $N \times N$ sub blocks (Here $N=11$). For every block, we annotate it caption text or non-caption text. Using the mask in Figure 2, we only have to extract gray level scale of those pixels of a sub block corresponding to the black squares in the mask (Figure 2). This reduces the dimensionality of the feature of a sub block image from
\begin{align}
N \times N & \text{ to } 4 \times N - 3. \text{ For } N = 11, \text{ it reaches a reduction factor to } 3.
\end{align}

Fig. 2 mask structure

3.2 Location of Video Caption

1) Selection of SVM kernel function
In support vector machine, the most used kernel functions are polynomial, Gaussian radial basis function (RBF) and sigmoidal neural network. Here we use RBF with the form:

\[ K(x, y) = \exp\left(-\frac{|x-y|^2}{\sigma^2}\right) \]  

(1)

2) Pyramid Model
In real applications, the size of caption text region can reach out this 11×11 sub block. A “p-step” pyramid is adopted here to gradually reduce the resolution of original image for p times. At each step, both the length and width of the sub block are reduced to \(\sqrt{2}\) of their original value, and trained SVM is used to detect caption text region with the same mask. The final result is obtained by taking the outcomes of each step into consideration.

3) Post-processing
A post-processing is needed here to remove such noises and merge the individual recognized sub blocks into a homogenous caption text region. Morphology suggests that caption text tend to cluster horizontally. Based on this assumption, most isolated mismatched sub blocks can be removed as follows:

(1) compute all candidate sub blocks from each video frame;
(2) construct extended block \((i,j)^*\) for each candidate block \((i,j)\), where \((i,j)^*\) consists of \((i,j)\) and its two horizontally adjacent blocks \((i-1,j)\) and \((i+1,j)\). If the extended block \((i,j)^*\) is connected with the extended block of any other candidate block, then \((i,j)\) is said to be a true block consisting caption text, otherwise \((i,j)\) is a noisy block and is removed from the set of candidates;
(3) After removing noisy blocks, the only thing remaining is to find a rectangle to enclose as much horizontally clustering true candidate as possible. The candidates locating in a rectangle are final identified caption text region.

4. Video shot clustering and mining
This section talks about the generation of video summarization based on the clustering and mining of low-level visual features. After shot boundary detection, we extract the key frames from the shots, and perform support vector clustering (SVC) [8] to merge similar shots. SVC can handle high dimensional data and arbitrary boundaries in data space. We get video abstraction by clustering and mining the shots, as well as removal of the visual-content redundancy among video frames.

4.1 Support vector clustering
After shot segmentation, we use support vector clustering to group similar shot, and for the sake of simplicity, we chose the first frame of the shot as the key frame. In the SVC algorithm data points are mapped from data space to a high dimensional feature space using a Gaussian kernel. In the feature space we look for the smallest sphere that encloses the image of the data. This sphere is mapped back to data space, where it forms a set of contours, which enclose the data points. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour are associated with the same cluster.

The clustering level can be controlled by changes in the width parameter of the Gaussian kernel (q). As this parameter is increased, the number of disconnected contours in data space increases too, leading to an increasing number of clusters.

SVC is a kernelized unsupervised learning technique. Given a set of data points \(\{x_j\}_0^N, x_j \in R^n (1 \leq j \leq N)\), in order to formulate a support vector description of this data set, a nonlinear mapping is employed to map X into some high dimensional feature space. The next step is to find the smallest enclosing hypersphere:

\[ \phi(x_j) - a \in R^l + \varepsilon, \]  

(2)

where \(R\) is the radius, \(a\) is the center and \(\varepsilon\) are some slack variables allowing for soft boundaries (some data points can be allowed to lie outside the sphere). The problem (2) is usually solved in its dual by introducing the Lagrangian and a regularization constant in the penalty term:

\[ L = R^2 - \sum_j \left[ R^2 + \varepsilon_j - \phi(x_j) - a \right]^2 \beta_j - \sum_j \varepsilon_j \mu_j + C \sum_j \varepsilon_j \]  

(3)

where \(\beta_j \geq 0, \mu_j \geq 0\) are Lagrangian multipliers, and \(C \sum_j \varepsilon_j\) is a penalty term. Also the Karush-Kuhn-Tucker condition and the Mercer kernel \(K(x, x_j)\) allows the problem to be rewritten as:

\[ W = \sum_j k(x, x_j) \beta_j - \sum_j \beta_j k(x, x_j) \]  

(4)

Those points lying on the boundary of the sphere are called Support Vectors (SVs). Points lying outside the sphere are called Bounded Support Vectors (BSVs), and are treated as noise.

One of the key features of kernel methods is that they do not require an explicit calculation of the feature map \(\phi\) but only use the values of the dot products between mapped patterns. It provides a way to deal with outliers and by using kernel method; therefore explicit calculations in feature space are not necessary.

4.2 Construction of cluster boundaries
Support vectors can be used to describe the hypersphere in feature space. For each point x the distance of its image in the feature space from the center of the hypersphere is given by:

\[ R^2(x) = K(x, x) - 2 \sum_j \beta_j K(x, x_j) + \sum_{ij} \beta_j \beta_i K(x_i, x_j) \]  

(5)
The cluster description itself does not differentiate between points that belong to different clusters. To do this, an adjacency matrix $A_x$ is defined based on geometric observation: given a pair of data points that belong to different clusters, for any path in data space connecting them, the corresponding path in feature space must have an intersection with the outside of the hypersphere. For each pair of points $A_x$ takes a binary value:

$$A_x = \begin{cases} 1, & \text{if} \exists \lambda \in [0,1], \langle \lambda \xi_i + (1-\lambda)\xi_j \rangle \in D, \forall \lambda \in [0,1] \\ 0, & \text{otherwise.} \end{cases}$$

Clusters are now defined as the connected components of the graph induced by. Checking the line segment is implemented by sampling a number of points (where in this paper we use 10 points).

### 4.3 Video Association Mining

By support vector clustering, we put the video shots to different video clusters, and now we can use video mining to get the video summarization. For example, in video dialog news, if we assume that all the host’s shots denoted as “A”, and all the visitor’s shots denoted as “B”, an other shots denoted as “C”, then the video can denoted as the sequence “ABABABCAB”. Therefore, we use the video association mining based on Apriori [9,10] to analyze the shot sequence and get video summarization. Before we introduce our mining techniques, let’s define the terminology at first:

- **An item** is a basic unit that denotes a shot cluster in this paper.
- **An X-ItemAssociation** is a sequential association that consists of $X$ sequential items. E.g., “ABC” is a 3-ItemAssociation.
- **The X-ItemSet** is an aggregation of all $X$-ItemAssociations, with each of its members being an X-ItemAssociation.
- **The support of an association** is the number of times that this association appears sequentially in the sequence.
- **The X-ItemAssociation** is an aggregation of all X-ItemAssociations that each of their support is no less than a given threshold.

To mine sequential association from sequence $D$, we use the AprioriVS algorithm. The main procedure is depicted in Fig.3. In the first stage, we sequentially scan the sequence $D$ and find the items with their support larger than a given threshold. The aggregation of these items forms the 1-ItemSet. We use the 1-ItemSet as input to generate the candidates of 2-ItemSet by using the candidate generation algorithm. We then scan $D$ again to calculate the support of each association in 2-ItemSet. The associations with their support larger than the threshold are collected to from 2-ItemSet and generate the candidates of 3-ItemSet. We will repeatedly execute this process until no more non-empty ItemSet can be found.

The AprioriVS candidate generation is similar to the method in [9]. This function takes the set of all $k$-ItemAssociations in $L_{k-1}$ and all their items as input. It first join the $L_{k-1}$ with $L_{k-1}$, then insert the result into $I_k$, and delete any member $x \in I_k$ that some $k$-ItemAssociation of $x$ not in $L_{k-1}$.

### 4.4 Summary Creation

As we can see from the procedures above, the items in mined video associations have to meet two requirements below: They should appear in the video frequently and they should appear alternate with other items. One can find that these two features imply that the items of video associations are good units for video summarization. The intuition is that the clusters that have been shown in longer sequential associations with larger supports would be more important in conveying the video scenario information.

Accordingly, to construct a video summary, our algorithm first selects candidate clusters and ranks their importance for summarization. Given video $F$, assume we get at most $X$-ItemSets, our summarization algorithm is given in Fig. 4.

**Begin:**

```
IFCluster[] ← ∅; IPClusterNum ← 0; Level L ← X.
```

For ($L = X; L ≥ 1; L --$)

Begin:

```
Count how many times a cluster has appeared in all L-ItemAssociations of L-ItemSet;
```

Rank the clusters in descent order, and denote them by $C_1, ..., C_k$ (assume $K$ clusters are found in L-ItemSet);

For ($k = 1; k ≤ K; k ++$)

```
{ If $C_k \not\in$ IFCluster[]
  { IFCluster[IFClusterNum] ← $C_k$
    IFClusterNum++;
  }
}
```

End

**Fig.4 Summary candidate clusters generation**

### 5. CONCLUSIONS

Automatic video summarization is a powerful tool for video browsing and accessing. In this paper, we generate video summary through semantic-face, video caption as well as visual feature clustering and mining. Future work includes research on a general framework it can generate a visual-audio summarization.

### Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 60272031), the National Key Technologies R & D Program of China during the 10th Five-Year Plan Period (No.2001BA101A07-03), Key Research Programs of Science and Technology Plan of Zhejiang Province (No.2003C21010), and Doctorate Research Foundation of the State Education Commission of China (No. 20010335049).
6. REFERENCES


