Incorporating Audio Cues into Dialog and Action Scene Extraction

Lei Chen[†], Shariq J. Rizvi[‡*] and M. Tamer Özsu[†]

[†]School of Computer Science, University of Waterloo, Waterloo, Canada
Email:{l6chen, tozsu}@uwaterloo.ca
[‡]Computer Science and Engineering Department, Indian Institute of Technology, Mumbai, India
Email: rizvi@cse.iitb.ac.in

ABSTRACT

In this paper, we present an approach to extract scenes in video. The approach is top-down and uses video editing rules and audio cues to extract simple dialog and action scenes. The underlying model is a finite state machine coupled with audio cues that are determined using an audio classifier.

Keywords: shot, scene, editing rules, finite state machine, support vector machine, audio classification

1. INTRODUCTION

The increasing availability and use of video has raised demands for better modeling of video and the provision of more sophisticated indexing and retrieval techniques. However, compared to text or images, video data are much more complicated. A one minute movie clip may contain about 2,000 video frames (image), a mixture of three types of sounds (audio), and several lines of close caption (text). How to efficiently represent and index video data remains a challenging problem. Early video database systems segment video into *shots*,[1–3] and extract *key frames* from each shot to represent it.[4–6] Such systems have been criticized for three reasons:

- The number of shots becomes very large with the growth of video data, which makes the data difficult to browse;
- A simple shot does not convey much semantics, since it is produced by a single camera operation.
- Using key frames may ignore temporal characteristics of the video.

There have been several attempts[7–10] to cluster semantically related shots into *scenes*. All the scene construction algorithms follow similar steps:

1. Visual features are extracted from shots, such as color histograms, textures and shapes.
2. Shots are clustered based on a similarity measure which is computed from the extracted visual features.
3. Clusters that are temporally close to each other are grouped into scenes.

All of these approaches use a "bottom-up" strategy, clustering the shots into "general" scenes without any knowledge about the semantics and structure of the scenes. However, they only employ low-level visual features, which may cause semantically unrelated shots to be clustered into one unit only because they may be "similar" in terms of their low-level visual features. Furthermore, users may not be interested in the "general" scenes constructed in this way, but may focus on particular scenes. In particular, dialog and action scenes have special importance in video, since they constitute basic "sentences" of a movie that consist of three basic types of scenes[11]: dialogs without action, dialogs with action, and actions without dialog. Automatic extraction of dialog and action scenes from a video is an important issue for practical use of video. There is another shortcoming of

[*]Work performed while the author was visiting the University of Waterloo.

previous approaches[7–10, 12]: most of them only focused on using the visual features ignored the accompanying audio data. However, the audio information contain a number of clues about the semantics of the video data. For example, the audio clips which are extracted from the dialog scenes are mainly speech mixed with some music or environment background sound. Therefore, automatic scene extraction models should take the accompanying audio data into consideration in order to make the extraction results closer to human understanding.

Based on video editing rules and audio cues, we develop a Finite State Machine (FSM) model to extract simple dialog or action scenes† from movies. In previous work[13] we had summarized our approach and reported our preliminary results involving only video editing rules. In this paper, we describe the complete model and discuss our experiments with it.

Audio data of a movie are not simply pure audio such as pure music, pure speech, etc., but are always a mixture of several types of sounds such as speech mixed with music or environment background sound, environment sound mixed with music background. Compared to pure audio classification,[14–17] there has not been much work on mixed audio classification.[18, 19] The audio classification model that we develop in this paper can differentiate between speech mixed with music or environment background sound, and environment sound mixed with music. The incorporation of audio cues into the FSM model based on editing rules results in a powerful extraction system. The experiments that we report indicate that the precision of the prediction model increases significantly with the use of audio data.

The rest of the paper is arranged as follows: Section 2 briefly presents the basic video editing rules which are deduced from the analysis of the dialog and action scenes from the editor's view. These rules guide our extraction model to extract scenes from video. Section 3 explains the audio features that we use for classifying mixed type audio. In Section 4, we present a support vector machine-based classification model to classify mixed audio data. The complete FSM model is presented in Section 5. Experimental results on the efficiency and accuracy of the FSM model coupled with audio classification model are given in Section 6, followed, in Section 7, by a comparison of our work with earlier proposals on video scene detection and audio classification. We conclude, in Section 8 and indicate some further directions that we are pursuing.

2. VIDEO EDITING RULES FOR CONSTRUCT DIALOG AND ACTION SCENES

A given video clip may be (and commonly is) interpreted differently by different users, and this is an example of semantic heterogeneity[20] within the context of video. However, there is only one meaning that a video editor tries to convey to the audience through constructing a semantically meaningful scene from video shots. There are some basic video editing rules that are followed, and developing an scene extraction model based on these rules indirectly solves the semantic heterogeneity problem.

Through the analysis of actor arrangement and camera placement, we find that there are only three basic types of video shot patterns in a two person (call a and b) dialog scene:

- a shot in which only actor a's face is visible throughout the shot (Type A shot);
- a shot in which only actor b's face is visible throughout the shot (Type B shot); and
- a shot with both actors a and b, with both of their faces visible (Type C shot).

In Figure 1(a), three representative frames from three types of shots are shown. These shots are extracted from a short dialog scene between Randall and his ex-girlfriend from the movie "Gone in 60 seconds". In this example dialog, actor a is Randall and actor b is his ex-girlfriend. Thus, in Figure 1(a), the first frame shows that Randall's face is visible; it is an A type shot. Similarly, in the second frame, Randall's ex-girlfriend shows up. The shot which is represented by this frame is a B type shot. In the third frame, both actors appear and with their faces visible. According to the definition above, it represents a C type shot.

In addition to above three types of shots, usually an *insert* or *cut-away* shot is introduced to depict something related to the dialog or not covered by those three types of shots. We use symbol # to represent it. Figure 1(b)

†In this paper, the term action scene is used to address one-on-one fighting action scene for simplicity.

(a) Three types of shots (b) Cut-away shot

Figure 1. Four types of shots in a dialog scene

shows a dialog which contains a cut-away shot. In this dialog, we start with a C type shot showing two actors having a dialog about a wooden sword. After that, a cut-away shot is inserted to show the sword, and finally we get back to a C type shot to re-establish the dialog scenario.

The rules governing the actor arrangement and camera placement in simple action scenes, are the same as those for producing simple dialog scenes. This is true even though, in an action scene, actors move rapidly and cameras follow the actors.

After a set of video shots are obtained from cameras that are used to film dialogs, the issue becomes how these shots can be used to construct a dialog scene to express a conversation. Basically, two steps are followed in editing a dialog scene.[11, 21]

1. Video editors use shots involving both actors (type C shot) or showing alternating actors (i.e, either AB or BA), to set up the dialog scenario, these shots give the audience an early impression of who are involved in the dialog.

   During this setting up process, the basic building blocks of dialog scenes are constructed. We call these basic building blocks *elementary dialog scenes*. An elementary dialog scene includes a set of video shots, and can itself be a dialog scene or be expanded to a longer dialog scene. The set of elementary dialog scenes are determined empirically, based on the analysis of editing rules used to establish dialog scenes and observations of dialog scenes of five movies[‡]. As a result, we have identified eighteen types of elementary dialog scenes as depicted in Table 1 along with statistics about their occurrence frequency in the five movies under consideration. In Table 1, AB and BA elementary dialog scenes do not appear. Through

elementary dialog scenes	appearance percentage
ABAB or BABA	41.21%
CAB or CBA	21.21%
C or C#C	19.39%
ABC or BAC	6.06%
CAC or CBC	3.63%
ABAC or BABC	2.42%
ACC or BCC	2.42%
ACA or BCB	2.42%
ACB or BCA	1.21%

Table 1. Statistical data on elementary dialog scenes

   the analysis of five movies, we find that the single appearance of AB or BA usually acts as a separator to connect two different scenes instead of a single dialog. One pair of shot/reverse shot or a C type shot has to be appended to AB or BA to construct an elementary dialog.

2. Each elementary dialog scene can be expanded by appending three types of shots.

   During this editing process, the basic rule that an editor uses is to give a contrast impression to the audience. For example, if the ending shot of one scene is an A type shot, usually a B type shot is

[‡]1. "Con Air", 1998; 2. "Life is Beautiful", 2000; 3. "First Knight" 1998; 4. "Deconstruction", 1990; 5. "What dreams may come" 1998.

appended to expand the scene. Similarly, the editor can append a C type shot as a re-establishing shot from time to time to remind the audience of the whole scenario surrounding the dialog scene. Table 2 lists expansion rules.

type of end shot in the scene	types of shots that may follow
A	B or C
B	A or C
C	A or B or C

Table 2. Possible types of shots to be appended

Based on these basic techniques, we draw two rules for the constructions of dialog and action scenes:

Rule 1: Dialog or action scenes must start with an elementary dialog scene.

Rule 2: An elementary dialog scene is expanded by appending different types of shots, which follows the expansion rules in Table 2.

3. CHARACTERISTICS OF MIXED TYPE AUDIO DATA

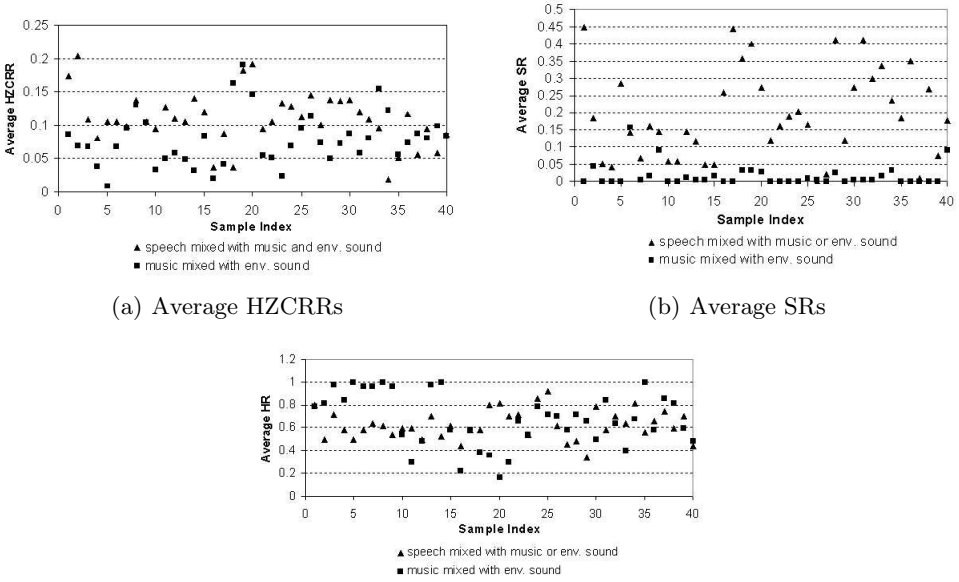
Through the observation of five test movies[§], we find an interesting fact: in most of the dialog scenes, the background audio is music or environment sound, but, compared to the foreground speech signals, these music signals are hidden and very easily ignored. In most of the action scenes, the background audio is also music but it can be noticed together with the action effect sounds. Therefore, audio data which come from dialog and action scenes can be classified into two audio types, which are speech mixed with music or environment background sound (*speech mixture*) and environment sound mixed with music (*env-mus mixture*). Incorporating an audio classifier which can differentiate these two mixed types of audio data is expected to increase accuracy of our FSM model.

A number of audio features provide separability between the different classes involved in pure audio classification. For example, the variance of zero crossing rate (ZCR) is relatively higher for speech than for music, because of the considerable difference between the ZCR values of voiced and unvoiced speech. However, for mixed type audio data, the values of these audio features are clustered, which bring the difficulties in mixed type audio classification. In the following subsections, three audio features, which are commonly used to differentiate music from speech, are presented with their characteristics in mixed type audio data. We extract 80 sample clips (5 seconds long) of speech mixture and env-mus mixture from the movie "Gladiator", 40 for each type. A 1-second window is used to divide each clip into subsegments.

3.1. Variance of Zero Crossing Rate

Due to the sharp difference between the ZCR values for voiced and unvoiced components, speech tends to have a high variance in its ZCR values. Because of the absence of any such phenomenon, music tends to have a lower variance. We calculate a measure of the ZCR variance for a window of N frames, as the ratio of frames having ZCR value more than 1.5 times the average ZCR of the window (*high zero-crossing rate ratio*[17]). In our experiments, we divide the 1-second window into 100 frames. Figure 2(a) shows average HZCRRs of speech mixture and env-mus mixture.

[§]1. "Con Air", 1998; 2. "Life is Beautiful", 2000; 3. "First Knight" 1998; 4. "Deconstruction", 1990; 5. "What dreams may come" 1998.



(a) Average HZCRRs

(b) Average SRs

(c) Average HRs

Figure 2. Characteristics of mixed type audio

3.2. Silence Ratio

Silence Ratio (SR) is defined as the ratio between the amount of silence of an audio piece and the length of the piece. SR is an useful statistics feature for audio classification, it is usually used to differentiate music from speech.[22] Normally speech has higher SRs than music. We divide a 1-second window into 50 frames. For each frame, the root mean square (RMS) is computed and compared to the RMS of the whole window. If the frame RMS is less than 50% of window RMS, we consider it a silence frame. With respect to speech mixture, silent periods can be defined as the time intervals that only background music or environment sound is playing. Therefore, the threshold of 50% is used instead of the 10% used for pure speech.[23] Figure 2(b) shows average SRs of speech mixture and env-mus mixture.

3.3. Harmonic Ratio

Spectrum analysis shows that pure music is more harmonic than speech, since pure speech contains a sequence of tonal (vowels) and noise (consonants).[19] Harmonic sound is defined as one which contains a series of frequencies which are derived from a fundamental or original frequency as a multiple of that. For each 1-second window audio clip, we divide it into 10 frames. We compute the harmonic frequency of each frame using the algorithm in.[24] The harmonic ratio (HR) is defined as the number of frames having a harmonic frequency to the total number of frames in the window. HR is a measure of the harmony in the clip. Figure 2(c) shows the average HRs of speech mixture and env-mus mixture.

As shown in Figure 2(b), with our defined threshold, the average SRs of speech mixture are much higher than those of env-mus mixture. Only a few points of speech mixture have lower values. The analysis of those clips shows that their background music or environment sound are loud and speech periods are short. Although Figures 2(a) and 2(c) show that HZCRRs and HRs of two mixed types audio are not separable in their original space, we still choose these two features for our audio classifier based on traits of HZCRR and HR in differentiating speech from music. In the next section, we present a classifier which can transform these mixed data into higher dimensional space and separate them there.

4. SVM-BASED AUDIO CLASSIFIER

In this section, we propose an audio classifier which can differentiate speech mixture and env-mus mixture. Support vector machine (SVM) is selected as the classifier. We select SVM because it always finds a global minimum compared with other classifiers such as neural network[25] and it is capable of learning in sparse high-dimensional spaces with relatively few training examples.[26] Basically, SVM seeks the separating hyperplane that produces the largest separation margin and it is based on the principle of structural risk minimization.[26] If the data vectors are linearly separable in the input space, a simple linear SVM can be used for their classification. However, in general, the input data vectors are not linearly separable. SVM uses a nonlinear transformation to map the input data vectors X into higher-dimensional space (feature space) and attempts to linearly separate the mapped data in higher-dimensional space. The nonlinear transformation is embedded in the kernel function of SVM. Therefore, the kernel function plays an important role in transformation and separation in feature space. As shown in Figures 2(a),2(b),2(c), HZCRRs, SRs, and HRs of mixed type audio data are not linear separable; we expect the kernel function of SVM to transform these data into higher dimension and make them separable. Two kernel functions are frequently used:

1. polynomial kernel:
   $K(X, X_i) = (X^T \cdot X_i + 1)^d$
   where X_i are support vectors which are determined from training data and $d = 1, 2, \ldots$ is the degree of the polynomial.

2. Gaussian Radial Basis Function (RBF) kernel:
   $K(X, X_i) = e^{-\|X - X_i\|^2 / 2\delta^2}$
   where $\delta > 0$ is defined to be the global basis function width.

Besides d and δ, there is another parameter, C, which is a positive penalty component used to control the amount of overlap that is allowed between classes. We select mixed type audio clips from different movies[¶], which are transformed into 16 bit, 44kHz, single channel raw audio clips of 5 second duration. With these clips, we generated a database with a size of around 1600 clips and these clips are manually annotated as "speech mixture" or "env-mus mixture". Half of the clips are used as training data and the rest of the clips are used as test data. We define the accuracy of a SVM-based classifier as the proportion of correctly identified clips to total number of the clips. SVM Light[27] is used to build our classifier. Both of kernel functions defined above are tested. In Figure 3 , we show the accuracy of trained SVM classifier with the polynomial kernel. With various parameter settings, around 86% accuracy can be achieved. These results demonstrate that SVM classifier is rather robust over the choice of model parameters. Similar accuracy can be achieved with Gaussian RBF kernel with $\delta = 5$. Due to the space limitation, these results are not shown. We also test the accuracy of SVM-based

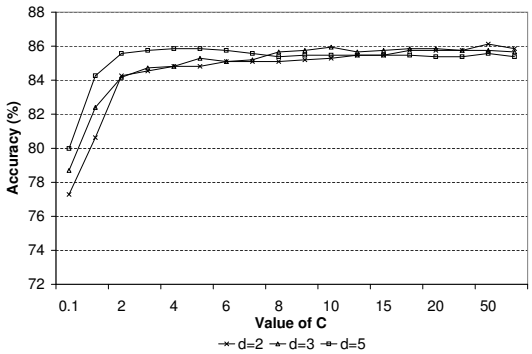


Figure 3. Accuracy versus C using polynomial kernel with $d = 2, 3, 5$

classifier with different audio features in action and the results are reported in Table 3. As we expected, the accuracy increases when one or more features are "turned on". Table 1 also shows that SR is a "better" feature

[¶]1. "Crouching Tiger Hidden Dragon", 2000; 2. "Gladiator", 2000

in differentiating speech mixture and env-mus mixture compared with the other two, which exactly corresponds to the analysis of three features in Section 3.

HZCRR	SR	HR	*Accuracy*
√	√	√	85.94%
	√	√	85.29%
√		√	72.15%
√	√		85.42%
√			69.35%
	√		84.17%
		√	63.25%

Table 3. Classification accuracy with different combinations of audio features

5. EXTENDED FSM MODEL

In this section, we present how to build dialog or action scene extraction model based on the video editing rules mentioned in Section 2. The proposed FSM model is extended by incorporating audio cues.

5.1. Video Shot String

We introduce the concept of a *video shot string* (VSS) to represent the temporal presentation sequence of different types of shots in a video.

Definition 1: $V = \{A, B, C, \#\}$ is the set of video shot types whose members are types of video shots defined in section 2.

Definition 2: A VSS is a string which is composed of symbols from V. Each symbol in VSS represents a shot in a video. The ordering of symbols in the string is from left to right, which represents the shot presentation sequence.

Based on the analysis of the above-discussed two editing steps, we define a VSS of a dialog scene as a $VSSDS$:

Definition 3: A $VSSDS$ is a VSS whose prefix is one of the elementary dialog scenes that can be expanded by rule 2.

The starting elementary dialog scene classifies a $VSSDS$ as well. Consequently, there are eighteen types of $VSSDS$s corresponding to those types of dialog scenes. It is easy to prove that these are regular languages over set V. We do not give a complete proof due to lack of space, but the following is the proof of one of these cases, namely the $VSSDS$ whose prefix is $ABAB$. Proof of other cases are similar.

$\{A\}$, $\{B\}$, $\{C\}$ are regular languages over V. $\{ABAB\}$ is a product of regular languages $\{A\}$ and $\{B\}$: $\{ABAB\} = \{A\} \bullet \{B\} \bullet \{A\} \bullet \{B\}$, so $\{ABAB\}$ is a regular language over V, too. A $VSSDS$ that starts with $ABAB$ includes string $ABAB$ and all the strings which are expanded from $ABAB$ by appending A, B or C using rule 2. Appending a shot to a scene is a concatenation operation ($\bullet$). Therefore, by definition of a regular language,[28] a $VSSDS$ of a dialog scene that starts with $ABAB$ is a regular language over V.

By taking the union of the eighteen types of $VSSDS$s, we again obtain a regular language over set V. Therefore, $VSSDS$s that are used to represent the temporal appearance patterns of video shots in dialog scenes are regular languages over set V.

5.2. Finite State Machine to Extract $VSSDS$s of Dialog Scenes

Since $VSSDS$s of dialog scenes are regular languages, the next issue will be how to automatically extract the $VSSDS$s which correspond to dialog scenes from VSSs of the whole video. In other words, how to extract specified regular languages from VSSs? In this paper, we propose a finite state machine (FSM) model to extract dialog scenes from videos. Note that we are not using the FSM to determine whether a language is a regular language over V, but constructively to extract those parts of a VSS that form regular languages with certain properties. In our proposed FSM, a VSS is used as an input to the FSM. A state is used to represent the status after a number of shots have been processed. An edge between states will determine an allowable transition from current state to another state under a labelled condition. The label of the arc is a symbol which is used to represent a type of shot. A sub-string of the VSS will be extracted by the FSM if and only if there exists a path from initial state to one of final states. The symbols on the path correspond to sequence of the shots in that sub-string of VSS. Figure 4 shows the transition diagram of our proposed FSM which is used to extract $VSSDS$s of simple dialog scenes between two actors: A video editor follows similar rules that are

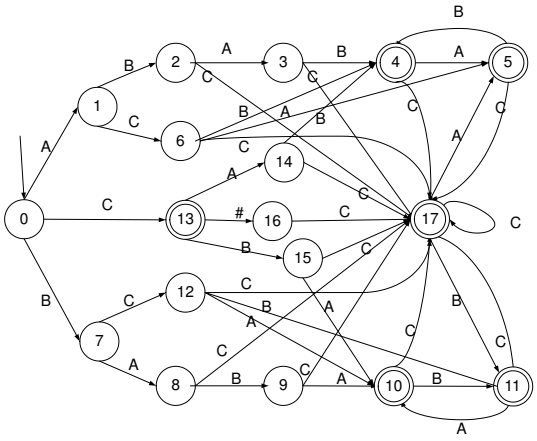


Figure 4. A FSM extracts $VSSDS$s of dialog scenes between actor a and actor b

used to construct dialog scenes to compose simple action scenes,[11,21] therefore, temporal appearance patterns of video shots in simple action scenes are similar to those of dialog scenes. The FSM model discussed above is also suitable for extracting simple action scenes.

5.3. Incorporating Audio Cues into FSM Model

Since we use the same FSM (which implements the same set of rules) to detect both types of scenes, we must find a mechanism to separate the dialog scenes from action scenes. In our previous work,[13] we used the average shot length. However, this metric does not perform very well on some action movies (see the results of experiment 1 in Section 6). The main reason is that we only use the visual effect feature (shot length) of dialog and action scenes to differentiate them. Here, the visual effect refers the impression that a video editor wants to bring to the audience by using different shot lengths in constructing a scene.[11] In some action movies, in order to show the action clearly, the video editor selects longer shots and this introduces errors into our model causing it to misclassify these types of action scenes as dialog scenes. Based on the observation of the five test movies mentioned in Section 3, we draw another rule for dialog and action scenes detection in movies.

Rule 3: The audio clips along with dialog scenes are usually speech mixture and the audio clips along with action scenes are env-mus mixture.

With rule 3, we propose the algorithm to incorporated audio cues into our FSM model as follows:

Step 1: For each video clip V_i, extract the audio clip A_i along with the V_i.

Step 2: Construct a VSS_i for each video clip V_i.

Step 3: Divide each audio clip A_i into segments, each of which lasts 5 seconds. The averages of three audio features (HZCRR, SR, and HR) are computed for each audio segment and used as inputs to the trained SVM audio classifier.

Step 4: Create audio meta data file M_i for each A_i by storing the classification results of each segment in A_i.

Step 5: Feed each VSS_i to the FSM.

Step 6: For each scene (dialog or action) extracted by the FSM, the corresponding audio meta data of the audio clip along with the scene is checked. If the audio clip has more speech mixture segments than env-mus segments, it is classified as a speech clip and the corresponding video scene will be considered as a dialog scene; if the audio clip is determined to have more music or environment segments, the corresponding video scene is detected as an action scene.

6. EXPERIMENT RESULTS

In this section, we present the results of some extraction experiments that were conducted using our extended model. We still use the two movies that have been used in our previous work[13] as shown in Table 4, since both of them contain dialog and action scenes. Video are first segmented into shots and appearances of actors are manually marked.

movie title	genre	year	duration (min)	No. shots
Gladiator	Action	2000	154	1363
Crouching Tiger and Hidden Dragon	Action	2000	120	1575

Table 4. The experiment data

We use precision and recall to measure our retrieval results which are well known metrics originally defined in the information retrieval literature. Precision measures the proportion of correctly recognized scenes, while recall measures the proportion of scenes that are recognized. The correctness of the detection results and missing detection of the correct scenes are judged by humans. In order to test the effect of incorporating audio classifier, two approaches have been designed:

1. Apply FSM using shot length without any audio cues (our previous approach[13]).
2. Apply FSM coupled with a SVM-based audio classifier.

Table 5 shows the results of extracting simple dialog scenes and simple action scenes using approach 1. This is the baseline for the other experiments. These results indicate that several scenes which are extracted by the

movie title	No. detected dialogs	precision (%)	recall (%)	No. of detected actions	precision (%)	recall (%)
Gladiator	95	89.47	96.60	25	84	84
Crouching Tiger and Hidden Dragon	154	80.52	90.51	64	76.56	81.6

Table 5. Dialog and action scenes extracted by the FSM with shot length checking

FSM model are neither dialog nor action scenes. Examples are the scenes in which two people stare at each other, the scenes in which two people appear in an interleaving pattern to show the occurrence of two parallel things. However, most of the errors come from the misclassification of dialog and action scenes. From Table 5,

we can see that the precision of retrieval results of the movie "Crouching Tiger and Hidden Dragon" is low. In this movie, there are several action scenes that show the action effect, which increases the average shot length above the threshold. There are also several scenes that mix with action and dialog scenes on which the FSM model does not perform very well.

movie title	No. detected dialogs	precision (%)	recall (%)	No. of detected actions	precision (%)	recall (%)
Gladiator	91	93.40	96.60	29	86	100
Crouching Tiger and Hidden Dragon	144	86.11	90.51	74	81.08	100

Table 6. Dialog and action scenes extracted with FSM coupled with the audio classifier

In our second experiment, the audio classifier is coupled with the FSM model to differentiate dialog or action scenes (which is approach 2). Table 6 shows the extraction results of approach 2. Comparing the results shown in Table 6 with those in Table 5, we find that the FSM coupled with audio classifier can achieve both higher precision and recall. Investigation of the results of both movies reveals misclassified dialog or action scenes in experiment 1 are correctly extracted with the help of audio cues.

7. RELATED WORK

There is very limited work on extracting the semantic scenes using a "top-down" approach. Yoshitiaka et.al[12] propose an approach similar to ours to extract semantic scenes (conversation, tension rising and action) based on the grammar of the film. However, in their approach, only the repetition of similar shots is employed to detect conversion scenes.

Lienhart et al.[29] develop a technique to extract dialog scenes with the aid of a face detection algorithm. However, they only extract dialog scenes which show shot/reverse shot patterns.

Neither of these approaches address the extraction of action scenes. Compared to their models, our model has the following advantages:

- We can, in addition to shot/reverse shot dialogs, detect single shot dialogs, dialogs with insertions and cuts, and dialogs with shot/reverse shots and recovering shots.
- Our model is rule based, which is very suitable for on-line content-based query processing.
- Our model can be easily extended to extract group conversions.
- Our model uses audio cues, which improves the system performance.
- Our model is based on the editors' point of view, rather than relying on individual users' interpretation of the video, which can solve the semantic heterogeneity problems caused by different interpretations.

As indicated earlier, audio cues play an essential part in an extraction model. The accuracy of audio classifiers directly affects the extraction results. Several audio classification models have been proposed for classifying pure type audio. Saunders[14] addresses the issue of pure type audio classification for FM radio. The idea is to allow automatic switching of channels when music is interrupted by advertisements. ZCR and short time energy are used to classify input audio into two classes: speech and music. Scheirer and Slaney[15] use thirteen features in time, frequency, spectrum and cepstrum domains and achieved better classification, concluding that not all of the audio features are necessary to perform an accurate classification. In addition, they claim that they improved the error rate to 1.4 % for a 2.4s window compared to 2.8% of Saunders' approach. Based on Scheirer's conclusion, Carey et al.[30] make a comparison study on audio features for speech and music discrimination. They figure out that simple audio features, such as pitch and amplitude, have significant differences between music and speech. Since then, many approaches have been proposed to classify pure type audio using different audio features

and classifiers.[16,17,30–33] Very few attempts have been made to classify mixed type audio. Srinivasan et al.[19] propose a rule-based model with empirically determines thresholds to classify mixed audio into discrete classes. Zhang and Kuo[34] propose a method for classification into speech, music, song, environmental sound, speech mixed with background music, silence, etc. Again, empirically determined values for classification thresholds have been proposed. Our SVM-based audio classification model does not require setting up empirical thresholds between classes and focuses on differentiating speech-mixture from mus-evn mixture.

8. CONCLUSION AND FUTURE WORK

Many approaches have been proposed to cluster video shots into scenes by computing the similarity of the shots based on their low level visual features. However, these clustered shots may not be similar in the semantics they convey, which makes the resulting scenes meaningless. These "bottom-up" approaches do not consider the knowledge that is used to construct scenes, which may also cause the constructed scenes to be too "general" for normal users of video database. In this paper, we extend our previous work on rule-based detection using a FSM model by incorporating audio cues. The extended model considers not only the editing rules that a video editor follows, but also the understanding of humans on audio cues of dialog and action scenes. A SVM-based audio classification model is created to differentiate speech mixed with music or environment background sound from environment sound mixed with music. The proposed audio classification model is evaluated with manually annotated audio data, nearly 86% accuracy can be achieved. The experimental results on dialog and action scene detection indicate that the FSM model, coupled with the audio classifier, can achieve much better results confirming the advantage of incorporating audio cues.

We believe, with help of the domain knowledge, our model can be easily extended to detect more types of semantically meaningful scenes, such as car chase, violence scenes, etc. Our future work will focus on developing more models to extract these scenes and a more general audio classification model which can classify more mixed types of audio data.

ACKNOWLEDGMENTS

This research is funded by Intelligent Robotics and Information Systems (IRIS), a Network of Center of Excellence of the Government of Canada.

REFERENCES

1. T. Kikukawa and S. Kawafuchi, "Development of an automatic summary editing system for the audio-visual resources," *Transactions on Electronics and Information* **J72**(A), pp. 204–212, 1992.
2. B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proceedings of IS&T/SPIE 2419*, pp. 2–13, 1995.
3. H. Zhang, A. Kankanhalli, and S. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems* **1**, pp. 10–28, 1993.
4. A. Nagasaka and Y. Tanaka, "Automatic video indexing and full video search for object appearances," in *Proceedings of 2nd Working Conference on Visual Database System*, pp. 119–133, 1991.
5. Y.Tonomura, A. Akutsu, K. Otsuji, and T.Sadakata, "VideoMap and video SpaceIcon: Tools for anatomizing video content," in *Proceedings of ACM INTERCHI*, pp. 131–141, 1993.
6. H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, "Video parsing, retrieval and browsing: An integrated and content based solution," in *Proceedings of ACM International Conference on Multimedia*, pp. 15–24, (San Francisco, CA), 1995.
7. M. Yeung and B.-L. Yeo, "Time-constrained clustering for segmentation of video into story units," in *Proceedings of 13th International Conference on Pattern Recognition*, **3**, pp. 375–380, 1996.
8. Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring video structure beyond the shots," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp. 237–240, 1992.
9. A. Hanjalic, R. Lagendijk, and J.Biemond, "Automatically segmenting movies into logical story units," in *Proceedings of International Conference on Visual Information Systems*, pp. 229–236, 1999.

10. W. Mahdi, M. Ardebilian, and L.M.Chen, "Automatic video scene segmentation based on spatial-temporal clues and rhythm," *Networking and Information Systems Journal* **2**(5), pp. 1–25, 2000.
11. D. Arijon, *Grammar of the Film Language*, Focal Press, 1976.
12. A. Yoshitaka, T. Ishii, and A. Hirakawa, "Content-based retrieval of video data by the grammar of film," in *Proceedings of IEEE Symposium on Visual Languages*, **3**, pp. 310–317, 1997.
13. L. Chen and M. T. Özsu, "Rule-based scene extraction from video," in *Proceedings of IEEE International Conference on Image Processing*, pp. 737–740, September 2002.
14. J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 993–996, May 1996.
15. E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 21–24, April 1997.
16. K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2445–2448, June 2000.
17. L. Lu, H. Jiang, and H. J. Zhang, "A robust audio classification and segmentation method," in *Proceedings of ACM International Conference on Multimeida*, 2001.
18. T. Zhang and C. C. J. Kuo, "Audio content analysis for online audiovisual data," *IEEE Transaction on Speech and Audio Processing* **9**, pp. 619–625, 5 2001.
19. S. Srinivasan, D. Petkovic, and D. Ponceleon, "Towards robust features for classifying audio in the CueVideo system," in *Proceedings of ACM International Conference on Multimedia*, 1999.
20. Y. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-oriented conceptual modeling of video data," in *Proceedings of the Eleventh International Conference on Proceedings on Data Engineering*, pp. 401–408, March 1995.
21. S. D. Katz, *Film Directing shot by shot visualizing from concept to screen*, Michael Wiese Productions, 1991.
22. G. J. Lu and T. Hankinson, "A technique towards automatic audio classification and retrieval," in *Proceedings of IEEE International Conference on Signal Processing*, pp. 1142–1145, Oct 1998.
23. M. C. Liu and C. Wan, "A study on content-based classification and retrieval of audio database," in *Proceedings of IEEE International Symposium on Database Engineering and Applications*, pp. 339–345, July 2001.
24. S. Pfeiffer, S. Fischer, and W. Effelsberg, "Automatic audio content analysis," in *Proceedings of ACM International Conference on Multimedia*, 1996.
25. C. J. C. Burges., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **2**(2), pp. 121–167, 1998.
26. V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
27. T. Joachims, "SVMlight support vector machine." http://svmlight.joachims.org/.
28. J. L. Hein, *Theory of Computation: An introduction*, Jones and Bartlett Publishers, 1996.
29. R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Scene determination based on video and audio features," in *Proceedings of International Conference on Visual Information Systems*, pp. 685–690, 1999.
30. M. Carey, E. S. Parris, and H. Lloyd-Thomas, "A comparison of features for speech, music discrimination," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 149–152, April 1999.
31. G. Lu and H. Templar, "An investigation of automatic audio classification and segmentation," in *Proceedings of WCCC-ICSP 2000, 5th International Conference on Signal Processing*, pp. 776–781.
32. E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia* **3**(3), pp. 27–36, 1996.
33. S. Z. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Transaction on Speech and Audio Processing* **8**(5), 2000.
34. T. Zhang and C. C. J. Kuo, "Audio content analysis for online audiovisual data," *IEEE Transaction on Speech and Audio Processing* **9**, pp. 619–625, 5 2001.