# Rushes Exploitation 2006 By CAS MCG*

*Sheng Tang, Yong-Dong Zhang, Jin-Tao Li, Xue-Feng Pan, Tian Xia, Ming Li ,*
*Anan Liu, Lei Bao, Shu-Chang Liu[1], Quan-Feng Yan, Li Tan*
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
**{ts, zhyd, jtli, xfpan, txia, mli}@ict.ac.cn**

## ABSTRACT

***In our rushes exploitation task of TRECVID 2006, we propose a novel and interactive rushes video selection and editing method based on hierarchical browsing of key frames, where high level features of each key frame such as face, interview, person, crowd, building, outdoor, waterbody, and other information about redundancy and repetition are displayed at same time for helping editors to select what they really want. During high level feature extraction, we propose a multi-modal interview detection method based on audio classification and face detection, and a new repetition detection method based on spatio-temporal slice. We also detect some concepts such as crowd, building, outdoor, waterbody based on SVM classifiers. Additionally, we characterize rushes by categorization camera motion for inferring intention. Due to the difficulty of high level feature extraction and the diversity of editor's requirements, our hierarchical browsing method along with extracted information may be a good choice for rushes exploitation.***

## Keywords

Rushes exploitation, Hierarchical browsing, Interview, Repetition, Concept detection

## 1. Introduction

As pointed out in the guideline [1], rushes are the raw material used to produce a video. They contain many frames or sequences of frames that are highly repetitive. e.g., many takes of the same scene redone due to errors (e.g. an actor gets his lines wrong, a plane flies over, etc.), long segments in which the camera is fixed on a given scene or barely moving, etc. Usually, twenty to forty times as much material may be shot as actually becomes part of the finished product. Watching rushes is boring and time consuming. Manual creating TV programs from rushes is hard and inefficient [2]. However, rushes are potentially very valuable but are largely unexploited because only the original production team knows what the rushes contain [3]. Therefore, with the rapid growth of digital rushes videos, how to mine the rushes and select really interesting clips from them is becoming more prominent.

Considering the peculiar nature of the rushes, and aiming to begin meeting some needs of an intelligence analyst or video producer looking at a large archive of unfamiliar unproduced video, NIST launched rushes task as a "pre-track" in TRECVID 2005. Several very different approaches, such as CUHK's camera motion trajectories analysis, Accenture's search using semantic web, DCU's object-based search, MediaMill's scene categorization, etc., were developed to manage the raw material last year [3].

Up to now, due to the diversity of editor's requirements and the difficulty of high level feature extraction resulted from significant gap between the low-level visual feature and high-level semantic information, fully automatic edition of rushes video seems impossible. In order to meet the needs of an intelligence analyst or video producer or editors, we think that a good system should solve the three following problems. The first one is how to turn a completely unstructured rushes video into a well structured one, and the second is how to remove redundant and repetitive clips and extract semantic features as much/well as possible while the third is how to exhibit the structural and semantic information to editors in a simple and clear way so that editors can easily and quickly select what they really want from rushes content.

To exploit this kind of unproduced video, we propose and demonstrate a novel method with a system which can single out redundant and repetitive rushes data and help editors to find and select what they really want from rushes content by providing structural and semantic information, based on hierarchical browsing of key frames of each shot, where high level features of each key frame such as face, interview, person, crowd, building, outdoor, waterbody, and other information about redundancy and repetition are displayed at same time.

*This work was supported by Beijing Science and Technology Planning Program of China (D0106008040291), and the Key Project of International Science and Technology Cooperation (2005DFA11060).

[1] Beijing University of Posts and Telecommunications, Beijing, 100876, China

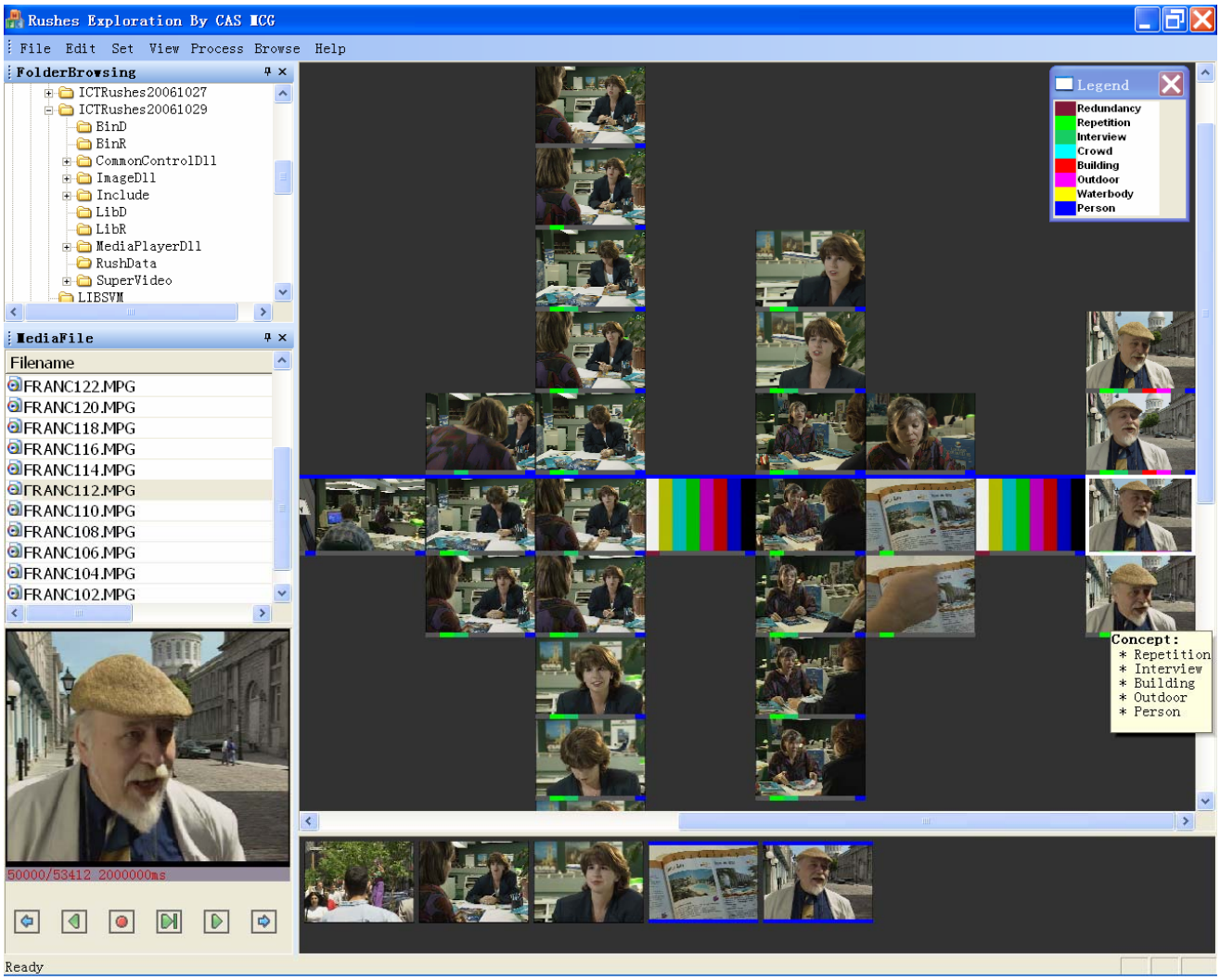


Fig.1 Interactive interface of our system: hierarchical browsing of the rushes video FRANC112.MPG

Due to the difficulty of high level feature extraction and the diversity of editor's requirements, our hierarchical browsing method along with extracted information may be a good choice for rushes exploitation.

The rest of the paper is organized as followed. The overall system, especially the interactive interface is described in Section 2. Then, the structuralization of rushes video is introduced in Section 3. Redundant and repetitive shot detection is described in Section 4, where a new repetition detection method based on spatio-temporal slice is proposed. Section 5 details high level feature extraction, especially presents a multi-modal interview detection method based on audio classification and face detection. For inferring intention, we also characterize rushes content by of categorization camera motion. Finally, we give our conclusion and future direction in Section 6.

## 2. System overview

To solve the aforementioned three questions, our system consists of three steps correspondingly. The first step is structuralization of rushes video, and the second is redundancy detection and semantic features extraction while the third is interactive interface. Since the first two steps will be described the following sections, we only introduce our interactive interface here.

As shown in Fig.1, our interface is composed of five parts: a folder browsing subwindow and a media file subwindow used for users to locate and select the rushes video they want to process; a playing-back subwindow (left-bottom) used for playing back the shot or the whole video; a hierarchical browsing subwindow used for visualizing both the structural and semantic information; and finally a

storyboard subwindow (right-bottom) used for editors to select and reorder the interested shots or clips.

After structuralization, the video is segmented to scenes, a scene is segmented to shots, and key frames are extracted from each shot. In order to visualize this structure clearly, we put the video into a two-dimensional Cartesian Coordinates in the hierarchical browsing subwindow. Along the vertical dimension is the first key frame of each shot inside the same scene while along the horizontal dimension is the linear temporal dimension of scene sequences. Through this subwindow, editors can browse the selected video briefly.

This structural presentation also supports a shot-wise video browsing. If editors are not sure about the contents of a certain shot, they can double click the corresponding key frame to launch a video player for playing back the shot in the left-bottom playing-back subwindow.

To express the redundant and repetitive shot and semantic features or concepts, we use a concept legend and a color bar under each key frame to display whether the corresponding key frame is redundant or repetitive, or contains the concepts: face, interview, person, crowd, building, outdoor, and waterbody respectively. Furthermore, for editor's convenience, these concepts about a shot can also be displayed as mouse moving to the corresponding key frames.

Video editors can select their desired shots only by dragging and dropping the corresponding key frames to the right-bottom storyboard subwindow. The shots in the box will be connected together from left to right to form a new video clip after the video editor has completed the shot selection. Editors can also reorder and delete shots on the storyboard using drag-and-drop. Consequently, editors can select what they really want from the rushes video easily.

## 3. Structuralization

Structuralization of rushes includes two steps: shot boundary detection and key frame extraction, and scene boundary detection.

### 3.1 Shot boundary detection and key frame extraction

Firstly, we detect the shot boundary using RGB histogram and segment the video into shots, and then extract key frames of shots using the adaptive key frame extraction using unsupervised clustering method proposed in [4].

We use an unsupervised clustering based approach to determine key frames within a shot. The similarity of two frames is defined as the similarity of their feature histogram. In our system, we select the color histogram of a frame as our visual content. After the clusters are formed, we simply choose the first frame of every cluster as its key frame.

The clustering based key frame extraction approach is not only efficient to compute, it also effectively captures the salient visual content of the video shots. For low-activity shots, it will extract less key frames than for high-activity shots. It can automatically extract multiple key frames depending on the visual complexity of the shot.

### 3.2 Scene boundary detection

We use the key frames of shots to detect scenes in videos. We transform problem of shot clustering into a graph partitioning problem as mentioned in [5]. This is achieved by constructing a weighted undirected graph called a shot similarity graph (SSG) [5]. The SSG is then split into sub graphs by applying the normalized cuts for graph partitioning. A partition is created for minimizing the disassociation between the groups and maximizing the association within the groups. Especially, when there is a redundant shot, it is treated as an individual scene.

Furthermore, we use the results of repetitive shot detection to remove the over segmented boundary in scene partition results. If some clips in scene A are the repeat of any clips in scene B, the scene A and scene B are emerged to form a new scene.

## 4. Redundant and repetitive shot detection

Since there are many shots in rushes which are useless for the video editors, such as color-bar, black or gray background, we consider these shots as redundant shots and develop three types of redundant shots: color-bar shot, black or gray background shot and very-short shot (less than 10 frames). There are also some highly repetitive shots in rushes, such as many takes of the same scene redone due to errors (e.g. an actor gets his lines wrong, a plane flies over, etc.). Taking the motivation of rushes task into consideration, we think it helpful for the editors to highlight the redundant shots and repetitive shots.

Since detection of the very-short shot is very simple, our work of redundant shots detection is concentrated on the classification of color-bar shot, black or gray background shot and other normal shots. It is easy to figure out redundant shots by extracting their uniform visual features as template. As for repetitive shots detection, our approach is based on spatiotemporal slice instead of key frames.

### 4.1 Redundant Shot Detection Based on Template Matching

We use the method of template matching to distinguish color-bar shot and black or gray background shot from normal shot.

The template of color-bar is the RGB normalized histogram of color-bar frames. If the difference of the RGB normalized histograms between query frame and template is less than a given value such as 1, we consider the frame as color-bar frame. The template of black and gray

background is the mean and standard deviation of the pixels of its frames. The mean and standard deviation of the black background pixels are 0 and 0 respectively. The average and standard deviation of the gray background pixels are 127 and 15. If both differences of the average and standard deviation between query frame and template are less than 5, we consider the frame as black or gray background frame. If query frame is neither color-bar frame nor black or gray background frame, we consider it as normal frame. Finally, we turn the frame-level result to shot-level result based on majority voting.

We test all the 48 rushes test videos, and the average precision and recall are 99.05% and 100.00% respectively. As shown in Table 1, Franco83.mpg and Franco112.mpg are the only videos whose precisions are below 100%, which indicates that all of the error detections are very-short shots. All the lengths of the error shots are less than 10 frames but the content of these shots is useful for the editors. Actually, these mistakes are due to the weak shot boundary detection algorithm.

Table 1: Redundant shot detection results of Franco83.mpg and Franco112.mpg

| Video | Color-bar Shot | | | Black or gray background Shot | | | Very-short Shot | | | Precision | Recall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | T | D | C | T | D | C | T | D | C | | |
| Franco83.mpg | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 7 | 6 | 90.9% | 100.0% |
| Franco112.mpg | 4 | 4 | 4 | 1 | 1 | 1 | 2 | 6 | 2 | 63.6% | 100.0% |

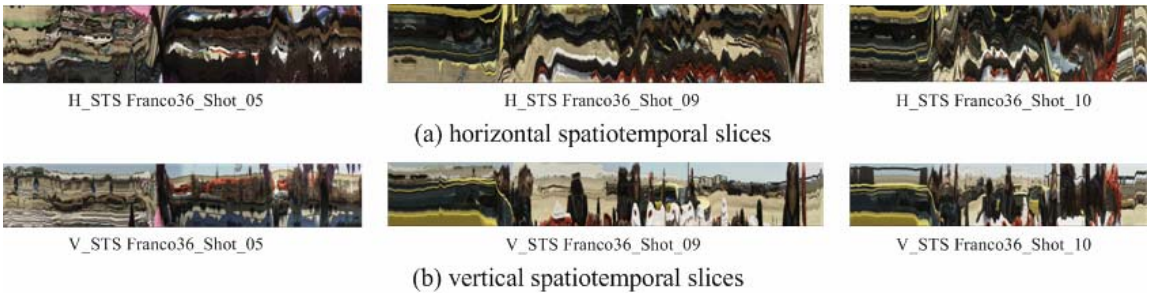


Fig.2 Examples of horizontal and vertical spatiotemporal slices

## 4.2 Repetitive Shot Detection Based on Spatiotemporal Slice

There is little research on repetitive shot detection for rushes. The similar work is content-based copy detection. Most of the works on content-based copy detection are focused on the characteristic of every single video frame. Jain [6] proposed a sequence matching method, based on a set of key frames. Although motion information was included with the key frames, it is not yet clear if the selected frames are appropriate to fully reflect the "action" within the video sequence. Hampapur [7] uses the ordinal measure originally proposed by [8] to retrieve video clips that depict similar actions. But the number of partitions is critical because the discriminability of system will be weakened as the number of partitions is reduced as pointed out in [9]. All of these methods ignore the temporal information of video which is an important feature of video and makes video difference from static picture.

For the task of repetitive shot detection for rushes, our approach is based on spatiotemporal slice [10] which is a set of two dimensional (2-D) images extracted along the time dimension of an image sequence. It reflects the variation in time dimension of video content in one static picture. That is the reason we choose spatiotemporal slice to represent a shot rather than key frames of it.

### *4.2.1 Repetitive shot detection on spatiotemporal slice*

As aforementioned, we define repetitive shots as many takes of the same scene redone due to errors. We propose an approach to detect repetitive shots based on spatiotemporal slice as follows.

As indicated in Fig.2, we set two rules to detect the repetitive shot for query shot: (1) its H_STS and V_STS are the most similar with those of query shot; (2) its interval between the query shot is no more than 2 shots.

For the definition of the similarity of two spatiotemporal slices, we extract color and texture feature to present their content. The color feature is described by three global color moments $(C_{avg}, C_{\text{var}}, C_{ske})$ and 512-dimesional normalized local HSV histogram $H = (h_1, h_2, \cdots h_{512})$. And the texture feature is described by 80-dimesional normalized local edge histogram $E = (e_1, e_2, \cdots e_{80})$ [11]. Finally, the similarity of spatiotemporal slices x and y is defined by the equation (1):

$$
\begin{aligned}
&Sim(x,y) = W_C \cdot Sim_{color}(x,y) + W_T \cdot Sim_{texture}(x,y) \\
&Sim_{color}(x,y) = w_1 \cdot S_1(C_{avgx}, C_{avgy}) + w_2 \cdot S_1(C_{\text{var}\,x}, C_{\text{var}\,y}) \\
&\qquad + w_3 \cdot S_1(C_{skex}, C_{skey}) + w_4 \cdot S_2(H_x, H_y) \\
&Sim_{texture}(x,y) = S_3(E_x, E_y) \\
&S_1(C_x, C_y) = 1 - \frac{|C_x - C_y|}{\max(C_x, C_y)} \\
&S_2(H_x, H_y) = \sum_{i=1}^{512} \min(h_{ix}, h_{iy}) \\
&S_3(E_x, E_y) = \sum_{i=1}^{80} \min(e_{ix}, e_{iy})
\end{aligned}
\tag{1}
$$

Where $W_C$ and $W_T$ are the weights given to color feature and texture feature such that $W_C + W_T = 1$. We choose $W_C = W_T = 0.5$. $w_1, w_2, w_3, w_4$ are the weights given to three global color moments and local HSV histogram such that $w_1 + w_2 + w_3 + w_4 = 1$. In our experiments, we choose $w_1 = 0.18, w_2 = w_3 = 0.06, w_4 = 0.7$.

### *4.2.2 The results of repetitive shot detection*

We test all the 48 test rushes videos, and some results are shown in Table 2. We can see that the results of Franco77 and Franco116 are much better than those of Franco99 and Franco104. The more repetitive shots the video has, the better the result is. The fact lies in our detection rules. We choose the shots whose H_STS and V_STS are the most similar with those of query shot as candidate. If the query shot has repetitive shot, this rule is sound because the repetitive shot must be the most similar one. But if there is no repetitive shot, it is possible that the most similar shot content the rule (2), and this will cause error detection. Future work includes more research on the detection rules based on spatiotemporal slices and representative feature extraction for spatiotemporal slices.

Table 2: Results of Repetitive Shot Detection Based on Spatiotemporal Slice

| Video | Truth | Detect | Correct | Recall (%) | Precision (%) |
|---|---|---|---|---|---|
| Franco77.mpg | 10 | 10 | 9 | 90.0 | 90.0 |
| Franco99.mpg | 6 | 8 | 4 | 66.7 | 50.0 |
| Franco104.mpg | 5 | 3 | 3 | 60.0 | 100 |
| Franco116.mpg | 17 | 15 | 14 | 82.4 | 93.3 |
| Franco124.mpg | 8 | 8 | 7 | 87.5 | 87.5 |
| Average Recall (%) | | | 77.0 | | |
| Average Precision (%) | | | 84.0 | | |

Table 3: Results of shot-level face detection

| Face concept detection | Labeled shots with face concept | Detected shots with face concept | Missing detection shots | Error detection shots |
|---|---|---|---|---|
| Total shots number | 394 | 368 | 60 | 34 |
| Precision (%) | 90.8 | | | |
| Recall (%) | 84.8 | | | |

## 5. High level feature extraction

Among the development data and test data provided for rushes exploitation, the main content can be classified into the following four kinds: interview scenes, person activity scenes, natural scenes and some redundancies. Firstly, interview scenes and person activity scenes are the most important parts for the editor to make new videos. By detecting these scenes, advanced video abstraction can be

achieved to generate brief news clips. Secondly, natural scenes, for example, waterbody, building and outdoor, are also important material for video editing. They depict the surrounding when the detected events happen. Thirdly, another important concept, camera motion characterization, plays an auxiliary role in video editing. This concept can suggest the location of the video segments with the desired camera motion, and can be used to infer intention [3]. Consequently, after removing the redundant shots in section 4, we detect the following concepts: interview; person; crowd, waterbody, building, outdoor, and camera motion characterization.

## 5.1 Face detection

Active Appearance Model (AAM) is very powerful to extract good facial features for success of applications such as face recognition, expression analysis and face animation. It is composed of two parts: the AAM subspace model and the AAM search. The superiority of AAM mentioned in [12] is that an approach for optimizing the parameterization of the AAM subspace model according to the search procedure is proposed while in the conventional methodology, the two sections are treated separately. The detailed procedure is described in [12].

Forty-eight rushes test videos are evaluated for face detection, and the results are shown in Table 3.

## 5.2 Interview Detection

Interview can be seen as a high level semantic concept containing both face and speech information. In our study, we investigate the application of existing face detection and audio classification techniques to interview detection.

The audio stream is firstly segmented into non-overlapped 20-ms short time frame (ST frame). Then five frame-level audio features [13, 14]: Short-Time Energy, Short-Time Zero-Crossing Rate, Frequency energy, Sub-band energy ratio, Mel-frequency cepstral coefficients, are extracted from each ST frame. Finally, each audio clip is classified into four kinds: silence, speech, music and background.

Forty-eight videos in rushes test data are all used to detect the shot-level interview. The results of using audio cue, visual cue and intersection fusion method are compared in Table 4, which shows the superiority of integrating audiovisual cues. It is very probable that in an interview shot both speech and face concepts can not be detected simultaneously. Therefore the recall of fusion method is worse than the other methods. However, the audiovisual information effectively denotes the meaning of interview. As a result, the precision of the fusion method strongly outperforms the others. Further analysis indicates that only five videos, in which the background and complexion severely affect the precision of face detection, have great influence on the recall by using fusion method. Excluding the five videos, precision is 81.9% and recall is up to 87.7%.

Table 4: Comparison of interview detection

| Interview detection | Precision | Recall |
|---|---|---|
| Audio cue (%) | 30.8 | 98.3 |
| Visual cue (%) | 62.3 | 78.9 |
| Fusion method (%) | 83.7 | 76.8 |

## 5.3 Person Detection

We use the method proposed in [15, 16] to judge if key frames of each scene depicting human or not. The features used in our system are Histograms of Oriented Gradients (HOG) of variable-size blocks that capture salient features of humans automatically. We adopt linear SVM based human detection for robust person detection.

We test the method on 2006 rushes data. As shown in the equation (2), the precision is defined as the ratio of the number of detected scenes depicting human N(q) over the number of total detected scene number M. On the other hand, the recall is the ratio of the number of retrieved relevant objects N(q) over the actual number of scenes depicting human G(p). The results are shown in Table 5.

$$Pr(q) = N(q)/M, \qquad Re(q) = N(q)/G(q) \tag{2}$$

Table 5: Results of scene-level human detection

| Interview concept detection | Precision | Recall |
|---|---|---|
| Results | 86.3 % | 95.1% |

## 5.4 Other Concept Detection

We also detect some concepts such as crowd, building, outdoor, waterbody, etc, based on SVM classifiers. This module consists of three parts: Part I, Part II and Part III.

Part I is Visual Feature Extraction. We extract six visual features from each video frame include Color Correlogram, Color Histogram, Color Moment, Co-occurrence Texture, Wavelet Texture Grid and Edge Histogram Layout according to [17].

Part II is Mapping. SVM is used to map Visual Features to Video Concepts. We use LibSVM [18] toolkit to do SVM training and classifying. The SVM type is C-SVC. Before training SVM models, we firstly use cross-validation to get the training parameters. For C-SVC, the training parameters are $\gamma$ and C. Cross-validation and scaling the Visual Features before training and classifying evidently improves the precision of mapping.

Part III is fusing the results of all the SVMs for each concept. We have tried some fusion methods including Max, Min and Average. By experiment, average fusion is the most stable one, so we choose it. The training set is the development set of High-level Feature Extraction in TRECVID 2005. For test, we randomly select 1000 key frames from test data of Rushes 2006 and manually label the four concepts for them. Experimental results are shown in Table 6.

Table 6: Experimental results of concepts detection

| Concept | Crowd | Building | Waterbody | Outdoor |
|---|---|---|---|---|
| Precision | 22.6% | 16.4% | 35.8% | 91.0% |

## 5.5 Camera Motion Classification

We characterize rushes by categorization camera motion, i.e., categorize the motion of camera into eight types: still, pan left, pan right, tilt up, tilt down, zoom in and zoom out, based on the motion vectors in compressed domain according to the method in [19].

To test the method, we build a ground truth by manually labeling ten videos selected randomly from the whole test rushes data as follows: FRANC035, FRANC039, FRANC045, FRANC058, FRANC064, FRANC077, FRANC089, FRANC095, FRANC108, FRANC126. The experimental results are shown in Table 7.

Table 7: Some Results of Camera Motion Classification

| Motion Type | Recall (%) | Precision (%) |
|---|---|---|
| PAN | 99.88 | 93.75 |
| TILT | 91.67 | 73.33 |
| ZOOM | 85.19 | 95.83 |
| STILL | 78.43 | 88.89 |

## 6. Conclusion

In our this year's exploitation in rushes, we propose and demonstrate a novel method with a system which can single out redundant and repetitive rushes data and help editors to find and select what they really want from rushes content by providing structural and semantic information, based on hierarchical browsing of key frames of each shot, where high level features of each key frame such as face, interview, person, crowd, building, outdoor, waterbody, and other information about redundancy and repetition are displayed at same time.

Due to the difficulty of high level feature extraction and the diversity of editor's requirements, our hierarchical browsing method along with extracted information may be a good choice for rushes exploitation.

Since it is the first time we participate in TRECVID, our work on rushes task is very preliminary due to lack of experience and limitation of time. Future work will be devoted to extensive study on semantic feature extraction and search on the whole rushes data.

## 7. REFERENCES


[1] http://www-nlpir.nist.gov/projects/tv2006/.

[2] Bradley P. Allen and Valery A. Petrushin , Searching for Relevant Video Shots in BBC Rushes Using Semantic Web Techniques,http://www-nlpir.nist.gov/projects/tvpubs/tv5.papers/

[3] Paul Over, Tzveta Ianeva, Wessel Kraaij, and Alan F. Smeaton, TRECVID 2005 - An Overview, http://www-nlpir.nist.gov/projects/tvpubs/tv5.papers/

[4] Yueting Zhuang, Yong Rui, Thomas S. Huang and Sharad Mehrotra. Image Processing, 1998. ICIP 98. Proceedings. 1998. Pages: 886-870

[5] Zeeshan Rasheed and Mubarak Shah. Detection and Representation of Scenes in Videos. IEEE Trans. Circuits Syst. Video Technol., Vol. 7, No. 6, Dec. 2005, Pages: 1097-1105

[6] A. K. Jain, A. Vailaya, andW. Xiong, "Query by clip," Multimedia Syst. J., vol. 7, no. 5, pp. 369–384, 1999.

[7] Arun Hampapur, Ki-Ho Hyun, Ruud Bolle: Comparison of Sequence Matching Techniques for Video Copy Detection. Proc. Storage and Retrieval for Media Databases, Jan. 2002, Page(s): 194-201

[8] D. Bhat and S. Nayar, "Ordinal measures for image correspondence," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 Issue: 4, pp. 415–423, April 1998.

[9] Changick Kim, Bhaskaran Vasudev: Spatiotemporal Sequence Matching for Efficient Video Copy Detection, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 1, January 2005, Page(s):127-132

[10] Peng. S. L, Medioni. G, Interpretation of image sequences by spatio-temporal analysis, Workshop on Visual Motion, March 1989. Page(s):344 – 351

[11] Park D K, Jeon Y S, Won C S. Efficient use of local edge histogram descriptor [A], in Proceedings of the ACM Workshops on Multimedia[C]. Los Angeles: [s. n.], 2000.51 - 54.

[12] Zhao Ming,Chen Chun,Li S Z,et al.Subspace analysis and optimization for AAM based face alignment. In: Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition. Seoul,South Korea,2004.290-295.

[13] Bai Liang; Hu Yaali, Feature analysis and extraction for audio automatic classification, Proc. of IEEE International Conference on Systems, Man and Cybernetics, vol.1, pp:767-772,2005

[14] Guodong Guo; Li S.Z, Content-based audio classification and retrieval by support vector machines, IEEE Transactions on Neural Networks,2003,14(1):209～215

[15] Qiang Zhu, Shai Avidan, Mei-Chen Yeh, and Kwang-Ting Cheng. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2) 2006, Pages:1491-1498

[16] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005. Pages:886-893

[17] Arnon Amir, Janne Argillander and etc. IBM Research TRECVID-2005 Video Retrieval System. In NIST Text Retrieval Conference (TREC).

[18] http://140.112.30.28/~cjlin/libsvm/index.html

[19] Xingquan Zhu, Ahmed K. Elmagarmid, Xiangyang Xue, Lide Wu, and Ann Christine Catlin, "InsightVideo: Toward Hierarchical Video Content Organization for Efficient Browsing, Summarization and Retrieval", IEEE Transactions on Multimedia, vol. 7, no. 4, Aug. 2005.