Shot Boundary Detection and High-Level Feature Extraction Experiments for TRECVID 2006

Masaki Naito[*1], Kazunori Matsumoto[*1], Masami Shshibori[*2], Kenji Kita[*2], Marco Cuturi[*3], Tomoko Matsui[*3],
Shin'ichi Sato[*4], Keiichiro Hoashi[*1], Fumiaki Sugaya[*1], and Yasuyuki Nakajima[*1]

[*1] KDDI R&D Laboratories, Inc. 2-1-15 Ohara, Fujimino, Saitama 356-8502, JAPAN
[*2] Tokushima University 2-1 Mishimacho Nanjyo, Tokushima, 770-8506, JAPAN
[*3] The Institute of Statistical Mathematics 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, JAPAN
[*4] National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, JAPAN

0. STRUCTURED ABSTRACT

Shot boundary detection

1. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

- abs-1: the system based on 2-stages SVMs technique used in KDDI's last year trial. Only abrupt cuts (hard cuts) and short dissolve transitions with less than 5 frames are detected. This means output includes only CUTs and no GRADs. The performance of this run for CUT is equal to the best CUT detector in TV2005. Hard cuts and short dissolve transitions are trained with TV2004 ref.
- abs-2: modified version of the last year's system. A new feature is added to detect long dissolve transitions. For detecting abrupt cuts, linear combination of multi-kernels is applied instead of 2-stages SVM. Short and long dissolve transitions are detected by 2-stages SVMs respectively. Hard cuts, short dissolve transitions and long dissolve transitions are trained with TV2005 ref.
- abs-3,4,5,6: recall and precision of the 'abs-2' system is controlled in the CUT. Functionality of GRAD detection is common.
- abs-7,8,9,10: recall and precision of the 'abs-5' system is controlled in the GRAD detection.

All these runs are conducted by KDDI Laboratories. In the weights optimization for multiple kernels, ***Institute of Statistical Mathematics*** and ***National Institute of Informatics*** helped KDDI's implementation.

2. *What if any significant differences (in terms of what measures) did you find among the runs?*

Compared with our TRECVID 2005 approach i.e. 2-stage-SVMs, this year's efforts i.e. a new additional feature and the combination of multi-kernels gave significant improvements for CUT.

3. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

New additional feature: is designed to detect long dissolve cuts and to prevent from erroneous cuts, thus this new feature may do harm to the recall of CUT but improve the precision.
Combination of multi-kernels: is just another kind of discriminator. Different training data set are applied in 'abs-1' and 'abs-2,3,4,5,6'. But by our experiments conducted after TREC submission, we estimate the difference of training data gave slight influence for the result. We estimate the difference of the result in 'abs-1' and 'abs-2,3,4,5,6' is caused by using multi-kernels technique instead of 2-stages SVMs.

4. *Overall, what did you learn about runs/approaches and the research question(s) that motivated them?*

The technique of multi-kernels is a promising approach to improve a discriminator for SBD.

High-Level Feature Extraction

1. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

- A_SiriusCy1_1: A color-based image retrieval method using two kinds of image features: the global color distribution feature and the common bitmap feature.
- A_SiriusCy2_2: A contents-based partial image retrieval method that uses two kinds of similarity distance: the template matching based on the Hausdorff distance and the Euclidean distance between color feature vectors.
- C_SiriusCy3_3: Data fusion approach similarity measure based on color histogram and similarity measures from 13 kinds of Haar-like feature detectors. The color histogram is obtained from key frames images. Each Haar feature detectors is trained by a specific object from LSCOM.
- C_SiriusCy4_4: Data fusion approach similarity measure based on color histogram and similarity measure from a Haar feature based face detector. The color histogram is obtained also from key frames images as C_SiriusCy3_3.
- C_SiriusCy5_5: Data fusion approach similarity measures based on 4-color histograms and similarity measures from 13 kinds of Haar-like feature detectors. In this run, color histograms are obtained from foreground, background and panorama of related shot.

- C_SiriusCy6_6: Data fusion approach similarity measures based on 4-color histograms and similarity measure from a Haar feature based face detector. color histograms are the same as run C_SiriusCy5_5.

Run A_SiriusCy1_1 & 2_2 are based on the technique of Tokushima University. Other's are based on that of KDDI.

2. *What if any significant differences (in terms of what measures) did you find among the runs?*

Rather than the template matching with color feature vectors, contents-based partial image retrieval becomes more precise. Method of multiple color histograms i.e. C_SiriusCy5_5 and C_SiriusCy6_6 gave better improvement than that of single image histogram i.e C_SiriusCy3_3 and C_SiriusCy4_4.

3. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

Estimation is the same as above.

4. *Overall, what did you learn about runs/approaches and the research question(s) that motivated them?*

Contens-based approach seems to be promising, but efforts to prepare training set are so hard. Thus, semi-learning algorithm is essential for contents-based approach.

1. INTRODUCTION

This is the forth TRECVID participation for KDDI R&D Laboratories. This year, we have participated in the shot boundary detection (SBD) task and high-level feature extraction (HLFE) task. For the SBD task, our main focus was to apply a discriminator with multi-kernels technique. For the HLFE task, we conduct interest point base approach on uncompressed domain.

2. SHOT BOUNDARY DETECTION

The accurate segmentation of shots in a video sequence is fundamental and an essential functionality for numerous video retrieval and management tasks. Many researchers have proposed algorithms to perform shot boundary detection based on certain features extracted from video frames, such as pixel differences, edge differences, color histograms, etc. From a learning theory perspective, it is a natural approach to combine such promising features in order to decide whether a boundary exists or not within a given video sequence. But naïve feature combination makes an excessive feature space to handle. In order to overcome this space problem, we adopt a 2-stage data fusion approach with a Support Vector Machine (SVM) technique in TV2005 [1]. And in TV2006, we apply a discriminator based on a multi-kernels technique.

2.1 Two-Stages SVMs (abs-1)

The overview of our data fusion approach is as follows[2]: At the first stage, every adopted feature is judged by a specific SVM. This means the number of feature types is equal to the number of SVMs at the 1st stage. And the other SVM at the second stage synthesizes the judgments from the 1st stage.

Figures 1 and 2 show our 2-stage discriminators with SVMs. Figure 1 is the structure of the discriminator in training mode, and Figure 2 is in prediction mode. "F1" ~ "F6" represent the feature types extracted from a video sequence. A conventional and useful *multiple pair-wise* technique [3] is applied for all these features.

"CUT Label," "DSH Label," and "D01L," ~ "D05L" are the label data for training. The values of every label data are assigned frame by frame. The "CUT Label" discriminates whether an abrupt cut occurs just before a relevant frame. "DSH Label" discriminates as to whether the center of a dissolve transition exists at a relevant frame. "D01L" ~ "D05L" discriminates whether the center of a dissolve transition with a specific period exists at a relevant frame.

"SVM1" ~ "SVM6" are Support Vector Machines at the 1st stage. Each SVM is designed to detect an abrupt cut based on a specific feature. "SVMds" is designed to detect a short dissolve cut with any transition span. "SVMd1" ~ "SVMd5" are designed to detect a dissolve transition with a specific length. For example, "SVM1" discriminates the existence of a dissolve transition whose length is 1. Every SVM outputs two kinds of values: the probability that a specified type of cut is detected and the probability that the same is not detected.

"SVM-C" and "SVM-D" are Support Vector Machines at the 2nd stage. "SVM-C" discriminates the existence of an abrupt cut based on the result of the 1st stage, while "SVM-D" also discriminates a short dissolve cut.

The functionality of "MIX" on Figure 2 is an arbitration of "SVM-C" and "SVM-D", based on the four probabilistic values. When "SVM-D" detects a dissolve cut and "SVM-C" does not detect an abrupt cut, "BEST DIS" chooses the most probable length of the dissolve transition.

It is not easy for hand-labelers to specify such a dissolve transition. But through the effort of TRECVID annotators, we can obtain accurate training data of dissolve transitions. Figures 3 and 4 show examples of abrupt and dissolve cuts respectively. In figure 4, the span of the dissolve transition is three frames.

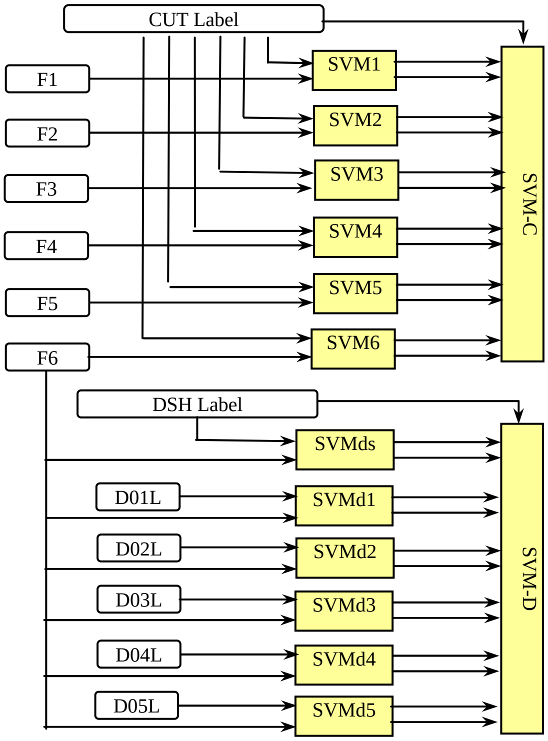

Figure 1: Structure of 2-stages SVMs in training mode.

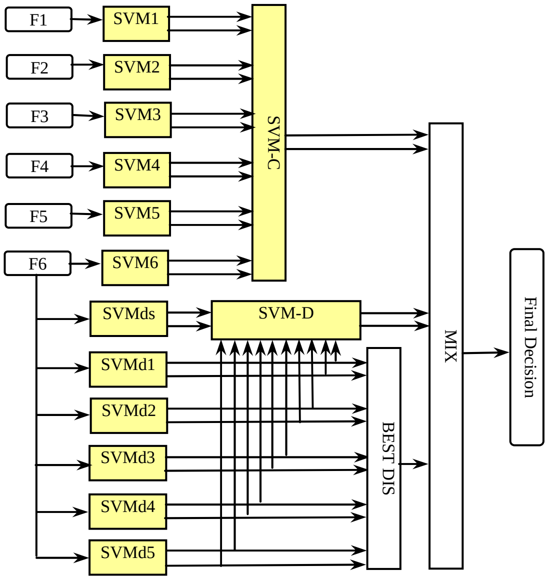

Figure 2: Structure of 2-stages SVMs in a prediction mode.

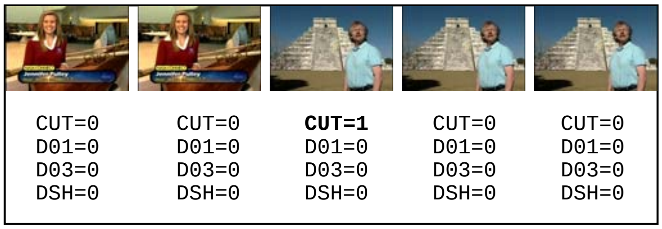

Figure 3: Example of an abrupt cut and values of labels.

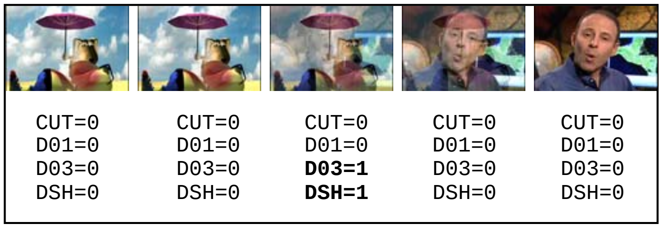

Figure 4: Example of a short dissolve cut (transition span = 3). Please note that this example is also CUT.

The next Table 1 shows the brief description of used features in 'abs-1'.

Table 1: Explanation about adopted features.

Feature ID	**Description**	**# dim(s)**
F1	the number of in-edges and out-edges in divided regions (4 by 4) based on [4].	224
F2	Standard deviation of pixel intensities in divided regions (4 by 4).	224
F3	TRECVID2004 approach by FX PAL[5] with Ohata's color domain, with PAC.	192
F4	TRECVID2004 approach by FX PAL[5] with RGB color domain, with PAC.	192
F5	Edge change ratio described in [6].	192
F6	Novel feature described in [1,2].	210

2.2 Combination of multiple kernels (abs-2,,,)

2.2.1 New Additional Feature

The new additional feature F7 in TV2006 is almost the same as *F6* described above.

A frame image in a dissolve transition is synthesized from two images, which come from different two video sequences respectively [6]. There are two scaling parameters for the synthesis. We estimate these two types of optimal scaling parameters frame by frame with a least-squares technique. And we consider two types of image differences. One is the difference between the target and the synthesized image, and the other is the difference between neighbor images simultaneously. Feature F6 is designed to detect short dissolve, but *F7* is for long dissolve.

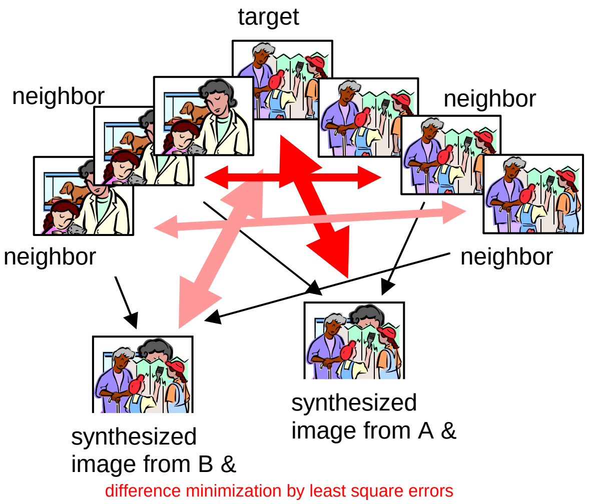
target
neighbor
neighbor
neighbor
neighbor
synthesized image from B &
synthesized image from A &
difference minimization by least square errors

Figure 5: Concept of Generating Feature F6 and F7.

2.2.2 Weight optimization of kernels

Kernel methods, especially SVM, are a powerful class of machine learning algorithms. Classical kernel-based learning algorithms are based on a single kernel, but recent developments in the literature on SVMs and other kernel methods have shown the need to consider multiple kernels.

The result of SVM learning is an α-weighted linear combination of kernel elements and the bias *b*:

$$\Phi(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i y_i \mathrm{k}(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Where $\mathbf{x}_i$ is labeled training examples ($y_i \in \{1,-1\}$). In our approach we consider linear combination of multiple kernels, i.e.

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}) = \sum_{k=1}^{K} \beta \mathrm{k}_k(\mathbf{x}_i, \mathbf{x})$$

with $\sum \beta_k = 1$, where each kernel k_k uses only a distinct set of features of each frame image. Lankriet et al. showed the efficient algorithms to optimize weight coefficients β [7]. According to this algorithms, we made optimized linear combination of F1, F2, and so on.

2.3 Evaluation of SBD

Our main focus is to validate the performance improvement of multi-kernel technique for SBD application. Therefore the most important result is the difference between *abs-1* and *abs-2,3,4,5,6*. By our careless consideration, different training data set are applied in *abs-1* and *abs-2,3,4,5,6*. But by our experiments conducted after TREC submission, we estimate the difference of training data gave slight influence for the result. We estimate the difference of the result in 'abs-1' and others is caused by using multi-kernels technique instead of 2-stages SVMs.

Table 2. Recall, precision and F-measure of CUT

#	RunID	Recall	Precision	F1
1	abs-1	0.815	0.925	0.867
2	abs-2	0.766	0.950	0.848
3	abs-3	0.820	0.935	0.874
4	abs-4	0.848	0.913	0.879
5	**abs-5**	**0.866**	**0.904**	**0.885**
6	abs-6	0.876	0.890	0.883
7	abs-7	0.766	0.950	0.848
8	abs-8	0.766	0.950	0.848
9	abs-9	0.766	0.950	0.848
10	abs-10	0.766	0.950	0.848

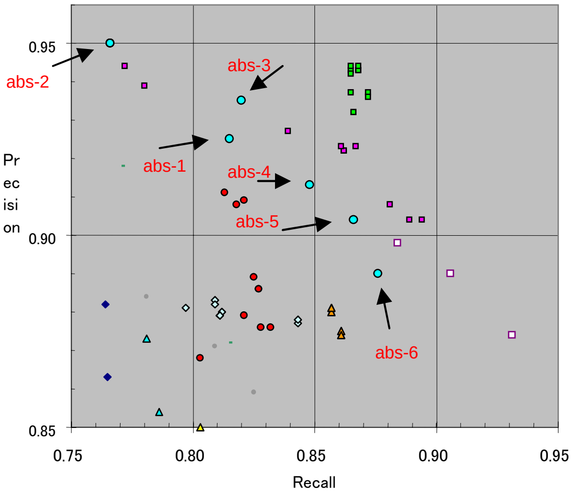

Figure 6: Recall and precision of CUT in SBD task.

3. HIGH-LEVEL FEATURE EXTRACTION

3.1 A_SiriusCy1_1

The A_SiriusCy1_1 system adopts a simple approach based on color-based image retrieval that uses two kinds of image features: the global color distribution feature and the common bitmap (CBM) feature [8]. Its main characteristic is the speed with which it takes, on average, only 2～4 minutes to retrieve results for each high-level feature.

To reduce the influence of telop texts, we first removed marginal pixels from an image, and then partitioned the image into 8×15 non-overlapping blocks (see Figure 7).

As the global color distribution feature, we used the mean (μ_L, μ_U and μ_V) and the standard deviation (σ_L, σ_U and σ_V) of Luv values for the entire image. Furthermore, we used the common bitmap feature to capture the spatial layout of the image. The common bitmap feature was derived by quantizing the image block into a two-level bitmap as follows:

$$CBM_L(i,j) = \begin{cases} 1 & if\ \mu_L(i,j) \geq \mu_L \\ 0 & otherwise \end{cases}$$

, where $\mu_L(i, j)$ is the mean L value for block (i, j). Similarly, $CBM_U(i, j)$ and $CBM_V(i, j)$ can be defined. Figure 8 shows the example of CBM, where the entire image is divided into 2×2 non-overlapping blocks.

Based on the global color feature and the common bitmap feature, the overall image similarity is obtained by linearly combining two different distances. The first distance is the Euclidean distance, which is used for comparing μ and σ, and the second is hamming distance for comparing two CBMs.

In the run, we first picked up images that were the representative for each high-level feature as query images (8 images on average per HLF), and then retrieved similar images from the whole test-set database.

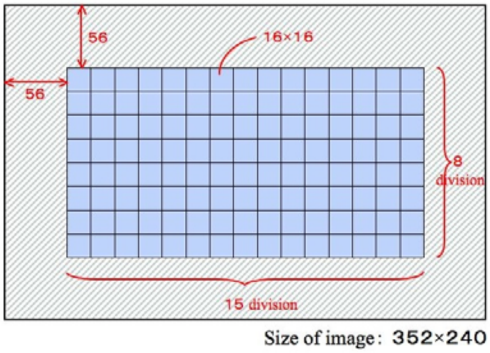

Figure 7. Partition of the image into 8×15 non-overlapping blocks.

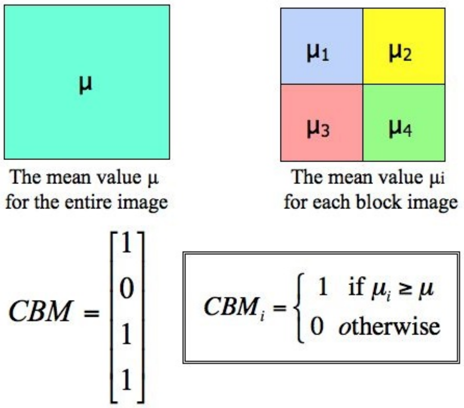

Figure 8. Example of CBM for 2×2 block images.

3.2 A_SiriusCy2_2

The A_SiriusCy2_2 system adopts a contents-based partial image retrieval approach that uses two kinds of similarity distance: the template matching based on the Hausdorff distance [9] and the Euclidean distance between color feature vectors. The Color feature vector is created from a patch image, whose center is a feature point extracted using a Harris operator [10].

The outline of the A_SiriusCy2_2 system is shown in Figure 9, and its algorithm is indicated as follows:

(1) A suitable key-frame image for each high level feature in development data is selected. In order to make a model image, the suitable part of the image is cut by the trimming manually.

(2) Some feature points are extracted from the model image using a Harris operator [10]. For example, Figure 10-(b) shows the extraction result from the model image of Figure 10-(a), where a red point represents a feature point.

(3) A center of all feature points obtained in step 2 is calculated, and then all feature points are moved to the center by a few pixels as shown in Figure 10-(c).

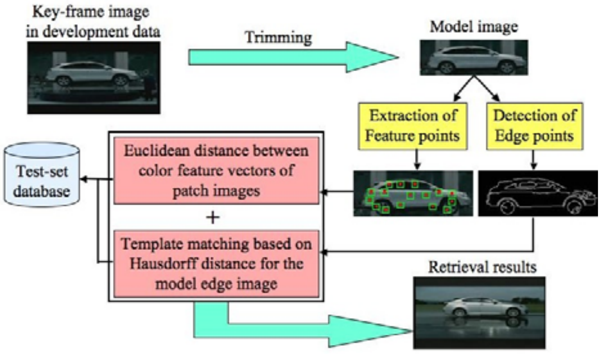

Figure 9. Outline of the A_SiriusCy2_2 system.

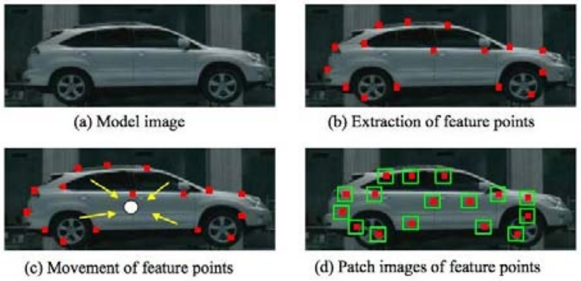
(a) Model image
(b) Extraction of feature points
(c) Movement of feature points
(d) Patch images of feature points

Figure 10. Generation of the patch image and its color feature for the model image.

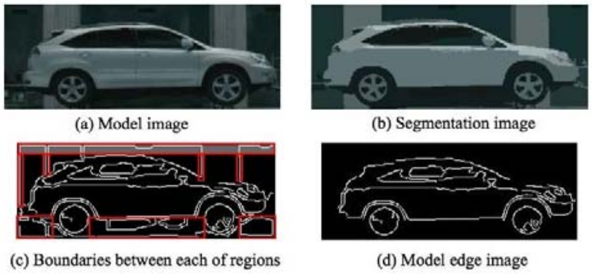

Figure 11 Generation of the model edge image.

(4) Color feature vectors are generated from each patch image as shown in Figure 10-(d), whose center is allocated by each of feature points obtained in step 3.

(5) In order to generate the color segmentation image as shown in Figure 10-(b), the color image segmentation algorithm [11] is applied to the model image of Figure 11-(a).

(6) All edge points of boundaries between each of regions in the segmentation image are detected as shown in Figure 11-(c).

(7) For the edge image obtained in step 6, noisy edge points, that is background edge points in red areas of Figure 11-(c), are removed manually, and then the image that consists of remaining edge points is called a model edge image as shown in Figure 11-(d).

(8) As for the shape similarity, the template matching based on the modified Hausdorff distance [9] between the model edge image and the corresponding edge image of each image in the test-set database.

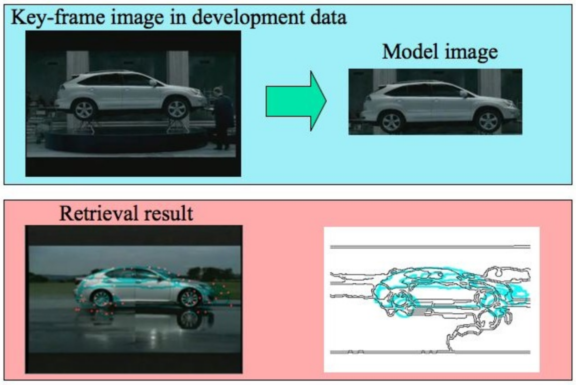

Figure 12. Example of retrieval results.

For two sets of edge points $A = a_1, \dots , a_m$ and $B = b_1, \dots , b_n$, the modified Hausdorff distance is defined by the following expressions.

$$H(A,B) = \max\,(\,h(A,B), h(B,A)\,)$$

where

$$h(A,B) = \frac{1}{m}\sum_{a\in A}\min_{b\in B}\| \, a - b \, \|$$

(9) As for the color similarity, the Euclidean distance of the color feature vector between the model image and each key-frame image in the test-set database, where the position to be allocated the model image in each test image set to the coordinates of the point detected by the template matching in step 8.

(10) The final similarity value is the weighted average of the shape similarity distance obtained in step 8 and the color similarity distance obtained in step 9.

Figure 12 shows the example of retrieval results using TRECVID2006 HLFE test data. In this run, we applied the A_SiriusCy2_2 system to only following 8 high-level features: “weather”, “office”, “building”, “face”, “person”, “airplane”, “car”, “explosion”, and the number of representative images picked up as model images is 5, 3, 1, 3, 3, 25, 15 and 1, respectively. The A_SiriusCy2_2 results of remaining high-level features were same as the A_SiriusCy1_1.

3.3 C_SiriusCy3_3 - C_SiriusCy6_6

3.3.1 Kernel based similarity by color histogram

We apply Harris feature points matching method to separate foreground images and background images, and to generate background panorama images. The method [1] is developed in TV2005’s Low-Level Feature Extraction task. By the method we generates three image i.e. foreground, background and panorama, for each key frame in each shot in training phase. Shot boundaries and key frames are obtained from common shot boundary information provided by TRECVID. These three images and original image are segmented by color information. After obtaining these four images for key frame, the relation between color histograms of 4-images and objects, that is high level feature, is trained by SVM.

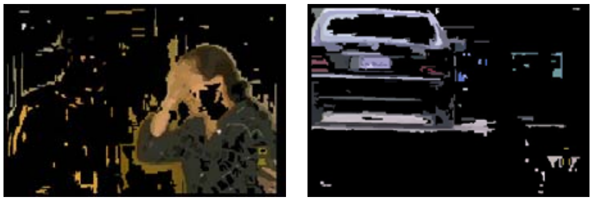
shot148_197_RKF shot148_238_RKF

Figure 13. Example of foreground images.

3.3.2 Object detection using Haar-like features

In some high-level features, the shapes of objects may offer important clues for detecting target objects. Therefore, we apply HLFE methods using shape information as well as those using color information.

Methods

We apply an object detection method using Haar-like features proposed by Voila et al [12], followed by the reclassification of the order of detected shots, including target objects, so that shots with larger objects are higher ranked (in **size-order**). We apply this method to detect the following 13 features; **Face, Person, Government-Leader, Corporate-Leader, Military, Animal, Computer_TV-screen, Flag-US, Airplane, Car, Bus, Truck** and **Boat_Ship**.

Furthermore, we applied the results of the "person" detection to detect certain other features. In this case, detected shots are reordered so that shots with a larger number of objects are higher ranked (in **number-order**).

Evaluation using TV2005 development data

Unfortunately, most of the high-level features detected by these methods were not evaluated in TRECVID2006. Therefore, we show the evaluation results using TRECVID2005 development data. 26 files of TRECVID2005 development data are used to evaluate the accuracy, while the remaining 109 files and TRECVID2003 development data were used to train each object detector. Each object detector was trained according to the following routine. Firstly, a primal detector is trained by using TRECVID2003 data. In this step, the region of key-frames in which the target object exists was used as positive training data and shots without target objects were used as negative training data (**Trainigset1**). Positive and negative data were separated based on common annotation [13]. In the next step, feature detection from key-frames of TRECVID2005 development data was performed by using primal object detectors. Subsequently, regions which were wrongly detected from shots that target features not involved were added to **Trainigset1** as negative training data (**Trainigset2**). This decision was performed based on the annotation offered by the LSCOM workshop [14]. Subsequently, refined object detectors were trained by using **Trainigset2**.

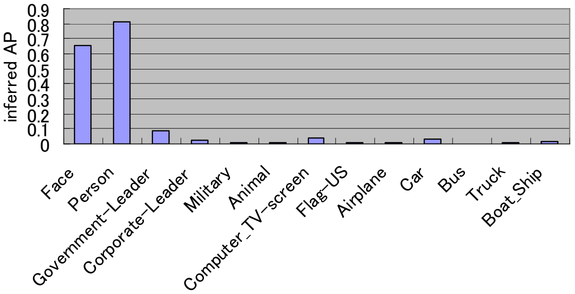

Figure 14. Inferred average precision for high-level feature detection using Haar-like feature based object detectors.

Table 3. Inferred average precision obtained by applying "person" detection to detect other features.

Feature	Criterion for reordering	
	size-order	number-order
Person	0.81	0.74
Studio	0.21	0.14
Outdoor	0.29	0.39
Crowd	0.10	0.34

The accuracy of object detection is described in Fig. 14. A vertical axis of this graph indicates the inferred average precisions (infAP) for each object. The results show that detection of human-related feature, e.g. face and person, were detected relatively correct. However, detection of another feature were not works well. Comparing with human-related objects, image of these objects appear with various direction and this may make accurate detection difficult.

The accuracy obtained by applying "person" detection to detect certain other features is described in Table 3. For each feature, the inferred average precision obtained by reordering detected shots in **size-order** and **number-order** is described in the table. The results show that detection works well by reordering the detected shots based on an appropriate criterion.

3.3.3 Integration of color-based detection and Haar-like feature based detection

To improve the accuracy of high-level feature detection, we integrate color-based detection and Haar-like feature based detection respectively, via the following simple method. Only shots detected by both color-based detection and Haar-like feature based detection are assumed to include a target feature. The rank of detected shots is then decided, based on the SVM score for color-based detection.

Although the introduction of positional relations between the front image obtained using color-based detection and objects detected using Haar-like feature based detection may achieve further improvement, it remains untested to date.

Table 4. Relation between the runID of submitted HLFE results with the use of color-based detection and Harr-like feature based detection.

runID	Type of detection	
	Color	**Haar-like**
C_SiriusCy3_3	**color-org**	**Haar-13**
C_SiriusCy4_4	**color-org**	**Haar-face**
C_SiriusCy5_5	**color-4img**	**Haar-13**
C_SiriusCy6_6	**color-4img**	**Harr-face**

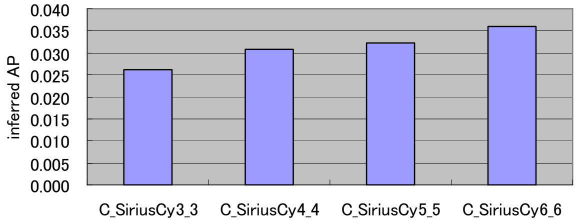

Figure 15. Inferred average precision of each run (average of all 20 features) .

3.3.4 Evaluation results

We made 4-runs by combining two types of color-based detection and two types of Haar-like feature based detection respectively. In the color based detection, detection using only the color information of the entire image of a shot (**color-org**), and detection using the color information of all four types of images, original entire image, foreground image, background image and generated background panorama image **(color-4img**) are tested. In Haar-like feature based detection, the case applying detectors for all 13 features, as described in section 3.3.2 (**Haar-13**) and the case detector for human-related objects (**Haar-face**) were tested. The relation between the submitted runID and the kind of color based and Harr-like feature based detector which is used, is described in Table 4, while the inferred average precision for each runID is described in Fig. 15. The best performance was obtained by integrating color-based detection (**color-4img**) and Haar-like feature based detection (**Haar-face**). Most of the objects detected by using Haar-like feature based methods were not evaluated in TRECVID2006, which may make the accuracy of **C_SiriusCy3_3** and **C_SiriusCy5_5**, **C_SiriusCy4_4** and **C_SiriusCy6_6** similar.

REFERENCES

[1] K. Matsumoto, M. Sugano, M. Naito, K. Hoashi, H. Kato, M. Shishibori, K. Kita, F. Sugaya, and Y. Nakajima: Shot Boundary Detection and Low-Level Feature Extraction Experiments for TRECVID 2005, http://www-nlpir.nist.gov/projects/tvpubs/tv5.papers/kddilabs.pdf, 2004.

[2] K. Matsumoto, M. Naito, K. Hoashi, and F. Sugaya, “SVM-Based Shot boundary detection WITH a novel feature,” , *Proceedings of International Conference on Multimedia and Expo*, 2006.

[3] Amir, A., Berg, M., Chang, S.-F., Hsu, W., Iyengar, G., Lin, C.-Y., Naphade, M., Natsev, A. P., Neti, C., Nock, H., Smith, J. R., Tseng, B., Wu, Y., and Zhang, D. “IBM research TREC-2003 video retrieval system,” *TREC Video Retrieval Evaluation (TRECVID 2003)*, Gaithersburg, MD, NIST, 2003

[4] Rainer Lienhart. “Comparison of Automatic Shot Boundary Detection Algorithms,” *Storage and Retrieval for Still Image and Video Databases* VII 1999, Proc. SPIE 3656-29, Jan. 1999.

[5] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, E. Rieffel, “FXPAL Experiments for TRECVID 2004,” *TREC Video Retrieval Evaluation (TRECVID 2004)*, Gaithersburg, MD, NIST, 2004

[6] Lienhart, R. “Reliable Transition Detection In Videos: A Survey and Practitioners Guide,” International Journal of Image and Graphics (IJIG), 1(3):469–486, 2001

[7] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In ICML ’04:Twenty-first international conference on Machine learning. ACM Press, 2004.

[8] C. C. Chang, and T. C. Lu, “A Color-Based Image Retrieval Method Using Color Distribution and Common Bitmap,” *Information Retrieval Technology*, - Second Asia Information Retrieval Symposium, AIRS 2005, (G. G. Lee, A. Yamada, H. Meng and S. H. Myaeng), Springer-Verlag Berlin Heidelberg, Germany, Vol. 3689, pp. 56-71, 2005.

[9] W. J. Rucklidge, “Efficient Visual Recognition Using the Hausdorff Distance,” *Lecture Notes in Computer Science*, No 1173, Springer-Verlag, 1996.

[10] C. Harris, and M. Stephens, “A Combined Corner and Edge Detector,” *Proceedings of the 4th Alvey Vision Conference*, pp.147-151, 1988.

[11] D. Comaniciu, and P. Meer, “Robust Analysis of Feature Spaces: Color Image Segmentation,” *Proceedings of 1997 IEEE Conference Computer Vision and Pattern Recognition*, pp. 750-755, 1997.

[12] P. Viola and M. J. Jones, “Robust Real-time Object Detection,“ Cambridge Research Laboratory Technical Report Series, CRL-2001-1, Feb. 2001.

[13] C.-Y. Lin, B. L. Tseng and J. R. Smith, "Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets," NIST TREC-2003 Video Retrieval Evaluation Conference, Gaithersburg, MD, November 2003. http://www-nlpir.nist.gov/projects/tvpubs/papers/ibm.final.paper.pdf

[14] LSCOM Lexicon Definitions and Annotations Version 1.0, DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia, Columbia University ADVENT Technical Report #217-2006-3 , March 2006