High-Level Feature Extraction Experiments for TRECVID 2007

Masaki Naito[*1], Keiichiro Hoashi[*1], Kazunori Matsumoto[*1], Masami Shishibori[*2], Kenji Kita[*2], Andrea Kutics[*3], Akihiko Nakagawa[*3], 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] Tokyo University of Technology 1404 Katakura, Hachioji, Tokyo 192-0982, JAPAN

0. STRUCTURED ABSTRACT

High-Level Feature Extraction (HLFE)

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

- A_KL1_1: A color-based image retrieval method using three kinds of image features: a global color distribution feature, a common bitmap feature and a Wavelet texture feature. Key-frames generated by our frame clustering method with threshold 5 were used as the input of the feature extraction system.
- A_KL2_2: A color-based image retrieval method in the same way as A_KL1_1, where key-frames generated with threshold 20 were used as the input.
- A_KL3_3: SVMs based on three visual features: a modified MPEG-7-based edge histogram descriptor, a color layout descriptor and an auto-correlogram, where key-frames generated with threshold 5 were used as the input data.
- A_KL4_4: SVMs as for A_KL3_3 and nine kinds of Haar-like feature-based extractors were used.
- A_KL5_5: In addition to A_KL4_4, a Haar-Like feature-based face extractor was applied to extract human related features.
- A_KL6_6: In the same way as A_KL5_5, but the Haar-Like feature-based extractor with lower recall and higher precision was used.

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

The accuracy of SVM based methods A_KL3_3 ~ A_KL6_6 were superior to the color-based image retrieval methods A_KL1_1 and A_KL2_2. In comparison with the inferred average precision of A_KL3_3 which is conducted with only SVM, A_KL4_4, A_KL5_5 and A_KL6_6 which are conducted with Haar-Like feature-based extractors, bring improvements.

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

The precision of HLFE was slightly improved by increasing the number of key-frames in a shot. The introduction of edge histogram descriptor, which describes both shape and textural properties, improves the precision without Haar-like feature-based extractors.

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

Introduction of new video features brings improvement, but they were high-dimensional. Therefore, reduction of the feature vectors is acceptable. A deep analysis such as Haar-like feature-based extraction seems to be promising, but it is difficult to prepare a training set. Thus, a semi-learning algorithm is essential for a contents-based approach.

1. INTRODUCTION

This is the fifth TRECVID participation for KDDI R&D Laboratories. This year, we have participated in a high-level feature extraction (HLFE) task. We obtained key-frames by the frame clustering method. Two types of HLFE methods were tested; a color-based image retrieval method and SVM-based method.

2. KEY-FRAME EXTRACTION

A frame clustering method, originally devised to extract *pseudo-shots* from user generated video content[1], is applied to extract discriminative key-frames for HLFE from a shot.

The first procedure of the proposed method is to extract *pseudo-shots* from shots of clustering frames that have similar color features. First, color layout information is extracted from the frames in the video to be analyzed. Next, the extracted frames are clustered using color layout distance as the measure of distance between frames and/or frame clusters. The number of clusters for each video is based on a pre-defined threshold, which defines the limit of mean distance between every pair of frames belonging to a cluster. Finally, the representative vector for each cluster is calculated, and then the frame with minimum distance to the representative vector is extracted as the key-frame.

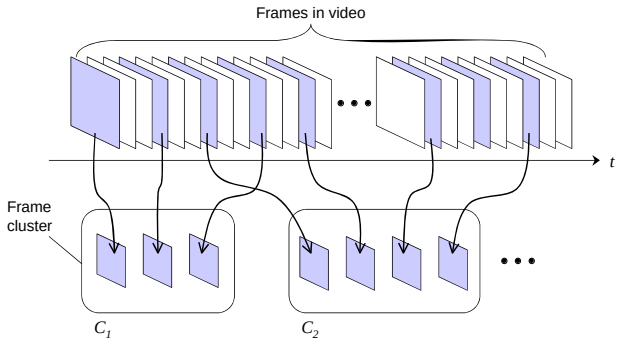


Figure 1. Conceptual illustration of frame clustering.

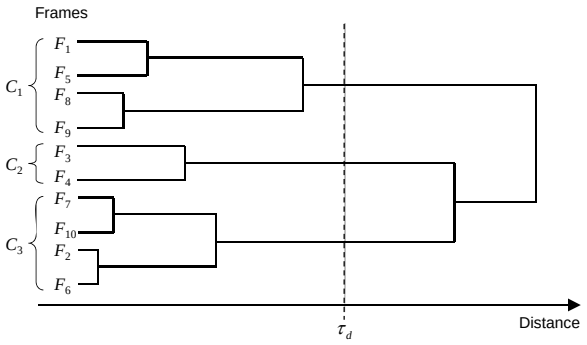


Figure 2. Conceptual illustration of cluster determination step

A conceptual illustration of frame clustering is shown in Figure 1. Note that the frame clustering procedure clusters frames that have similar color features, regardless of the chronological order of frame occurrence. The first step is to extract color layout features from the frames comprising the video. The color layout information, defined in MPEG-7 Visual [2], is extracted based on the algorithm developed by Sugano *et al*. [3]. The color layout information corresponds to 8×8 DCT coefficients of Y, C_b, and C_r components of the 8×8 downscaled image. The numbers of coefficients used here are 6, 3, and 3 for Y, C_b, and C_r, respectively. Since it is obviously redundant to extract features from every frame of the video, the color layout features are extracted from every K-th frame. In the following experiments, K is decided shot by shot as 100 ~ 200 frames are extracted from each shot.

The next step is to cluster the previous frames based on their extracted color layout information. Since the optimal number of frame clusters for a video should be determined based on its visual features, a bottom-up hierarchical clustering algorithm based on Ward's method is applied for this step.

The distance between the color layout of two frames is calculated based on the image similarity measure, which is also defined in MPEG-7 Extraction and Use [4]. Let

$$F_1 = \left(Y_1,\cdots,Y_6,C_{b1},\cdots,C_{b3},C_{r1},\cdots,C_{r3}\right),$$
$$F_2 = \left(Y_1^{'},\cdots,Y_6^{'},C_{b1}^{'},\cdots,C_{b3}^{'},C_{r1}^{'},\cdots,C_{r3}^{'}\right)$$

denote the color layout information (DCT coefficients) extracted from frames F_1 and F_2, respectively. The distance between these two frames $Dist\left(F_1,F_2\right)$ is calculated by:

$$Dist\left(F_1,F_2\right) = \sqrt{\sum_{i=1}^{6}\lambda_{Y_i}\left(Y_i - Y_i^{'}\right)^2} + \sqrt{\sum_{i=1}^{3}\lambda_{C_{b_i}}\left(C_{b_i} - C_{b_i}^{'}\right)^2} + \sqrt{\sum_{i=1}^{3}\lambda_{C_{r_i}}\left(C_{r_i} - C_{r_i}^{'}\right)^2}$$

, where the lambdas, λ_{Y_i}, $\lambda_{C_{b_i}}$, $\lambda_{C_{r_i}}$, denote the weighting values for each coefficient. The actual values of the weights are, following the examples in MPEG-7 Extraction and Use [4], as follows:

$$\left\{\lambda_{Y_1},\lambda_{Y_2},\lambda_{Y_3},\lambda_{Y_4},\lambda_{Y_5},\lambda_{Y_6}\right\} = \{2,2,2,1,1,1\}$$
$$\left\{\lambda_{C_{b_1}},\lambda_{C_{b_2}},\lambda_{C_{b_3}}\right\} = \{2,1,1\}$$
$$\left\{\lambda_{C_{r_1}},\lambda_{C_{r_2}},\lambda_{C_{r_3}}\right\} = \{4,2,2\}$$

Clustering is conducted by calculating the distances between all pairs of frames and merging the two frames with the shortest distance.

Since the element values of the 12-dimension features are DCT coefficients, the representative vector of a cluster cannot be calculated by generating the centroid of all vectors in the cluster. Therefore, the representative vector for a cluster is generated by selecting the coefficient value of each vector element that appears most frequently within the cluster.

This process is repeated until all frames are grouped to a single cluster. The final step is to determine the *pseudo-shots* for the video based on the number of clusters for each video determined based on the average distance between each pair of frames belonging to a cluster, calculated during the previous frame clustering step. Figure 2 illustrates the cluster determination process for a shot consisting of 10 frames $F_1,\cdots,F_{10}$. The horizontal axis of Figure 2 expresses the average distance between two frames belonging to a cluster, and τ_d denotes the threshold to determine the *pseudo-shots*. In this example, the frames are divided into three *pseudo-shots* $\left(C_1,C_2,C_3\right)$.

This approach to conducting hierarchical clustering, and determining *pseudo-shots* for each video based on a threshold that makes it possible to generate pseudo-shots adaptively, depending on the visual features of the video. For a video that consists of a wide variety of content, the number of *pseudo-shots* is expected to be high, while for video which is mostly still, the number of extracted *pseudo-shots* is expected to be low.

Finally, a frame that has minimum distance $Dist(F_1,F_2)$ to a representative vector is selected as a key-frame for each cluster.

This method is applied to extract key-frames from TRECVID2007 test data. The relation between threshold τ_d, which determines number of key-frames, and the average number of key-frames in a shot are shown in Figure 3. Key-frames obtained using thresholds $\tau_d = 5$, $\tau_d = 10$ and $\tau_d = 20$ are used in the following experiments. The distribution of number of key-frames in a shot is described in Figure 4. The average number of key-frames in a shot obtained by frame clustering with threshold $\tau_d = 5$, $\tau_d = 10$ and $\tau_d = 20$ are 3.31, 2.00 and 1.23, respectively.

3. HIGH-LEVEL FEATURE EXTRACTION

After we obtained key-frames by the frame clustering method, HLFE is conducted. We tested to types of extraction methods; a color-based image retrieval method and SVM-based method.

3.1 Color-based image retrieval method

3.1.1 Visual features

The A_KL1_1 and A_KL2_2 systems adopt a simple approach based on color-based image retrieval that uses three kinds of visual features: the global color distribution feature, the common bitmap (CBM) feature [5] and the wavelet texture feature [6]. Its main characteristic is its speed: 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 (each 56 pixels) from an image, and then partitioned the image into 8×15 non-overlapping blocks. As a result, the size of a non-overlapping block becomes 16×16.

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.

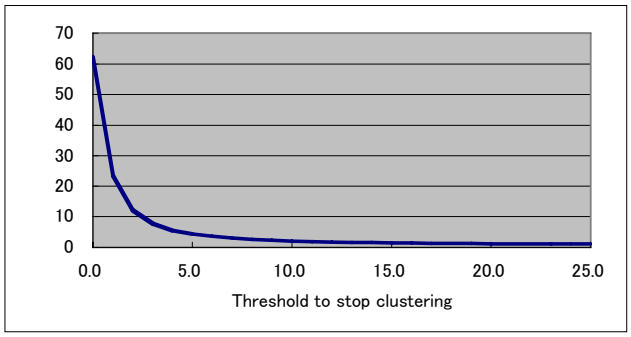
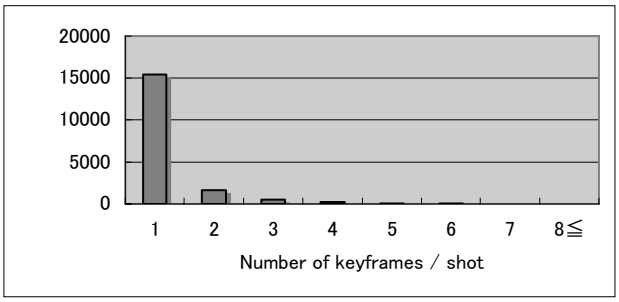
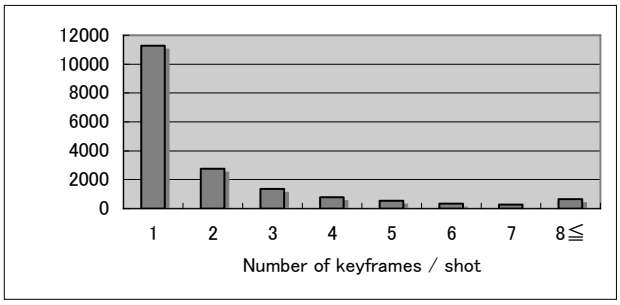


Figure 3. Average number of key-frames in a shot obtained by frame clustering with several threshold.

$\tau_d = 20$:



$\tau_d = 10$:



$\tau_d = 5$:

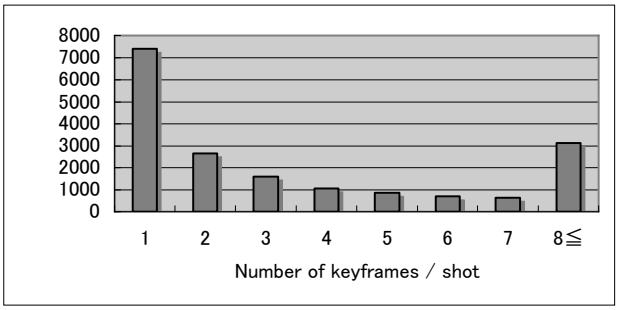


Figure 4. Distribution of number of key-frames in a shot obtained by frame clustering with threshold $\tau_d = 20$, $\tau_d = 10$ and $\tau_d = 5$.

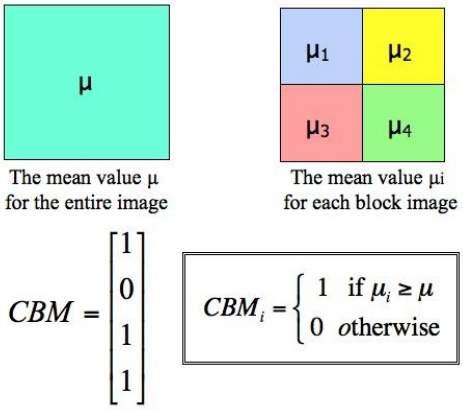


Figure 5. Example of CBM for 2×2 block images.

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). $CBM_U(i, j)$ and $CBM_V(i, j)$ can be similarly defined. Figure 5 shows the example of CBM, where the entire image is divided into 2×2 non-overlapping blocks. Finally, we used the wavelet texture feature to capture the spatial texture of the image. The wavelet texture feature was generated using Daubechies Wavelet toward each non-overlapping block.

3.1.2 Retrieval method

Based on the global color feature, the common bitmap feature and the wavelet texture feature, the overall image similarity is obtained by combining three different distances. The first distance is the Euclidean distance, for comparing μ and σ; the second distance is the hamming distance, for comparing two CBMs; and the third distance is the Euclidean distance, for comparing two wavelet texture features. These three distances are combined by the Gaussian normalization.

The retrieval algorithms of the A_KL1_1 and A_KL2_2 systems are the same, but the difference between the two systems is the number of key frames extracted from the video data. The A_KL1_1 system used the set of key frames generated by threshold 5 on the clustering process. This key frame set includes an average of 3.31 frames per shot. On the other hand, the A_KL1_2 system used the set of key frames generated by threshold 20 in the clustering process. This key frame set includes an average of 1.23 frames per shot.

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.

3.2 SVM-based extraction method

In A_KL3_3 and A_KL6_6, SVMs based on three types of visual features and Haar-Like feature-based extractors were used.

3.2.1 Visual features

In this step, we extract discrete still images such as video frames from continuous video data used as input data. Next, we extract mainly MPEG-7 compliant visual features from each image frame. In our experiments, we used three main visual features: a modified MPEG-7-based edge histogram descriptor, a color layout descriptor [7] and a descriptor called a color correlogram [8]. The color correlogram is extracted to capture spatial color information of the frames. We selected these three features as they can provide useful information about color, shape, texture and layout properties of the extracted video frames.

The edge histogram descriptor (called here ehd) is a powerful descriptor that works by detecting edge points and their directions in images, as the existence and location of edges and their directional information are very important features for describing both shape and textural properties. In order to extract this descriptor, we first divide the image into 4x4 blocks and then further divide the resulting sub-images into smaller sub-blocks. When the size of a sub-block is larger then 2x2, it is further subdivided into 2x2 blocks and average gray-scale values are calculated for each part of the sub-block window for further filtering and thus detecting edges. Next, four directional edge detectors, covering 0, 45, 90, 135 degrees, and the non-directional edge detector are applied to the sub-blocks. In this way, we obtain several edge points according to each of the five directions. We determine a histogram consisting of 80 bins according to the 5 edge directions calculated over the sub-images in order to obtain local edge histograms. Here, we use an eight-bit representation to express each histogram bin value as we found that using a more compact and thus lower resolution representation results in much lower retrieval as well as annotation accuracy. Furthermore, we compute an extended histogram by grouping the image blocks and calculating so-called semi-global and global histograms. The semi-global histograms are calculated as follows: (1) we divide the image into four evenly divided horizontal and vertical blocks, thus creating eight different edge histograms; (2) we divide the image into four evenly divided blocks and an overlapping block of the same size in the middle of the

image, thus creating five different edge histograms. Finally, we measure the similarity by using a weighted *L1* distance by applying a larger weight (5) to the global histogram, which is calculated by collecting the directional information of the edge points detected for the whole image.

We use the MPEG-7-based color layout descriptor (called here cls), described in section 2, and the similarity between two images is determined by using the *L2* distance and assigning larger weights to the lower frequency coefficients. However, to obtain more accurate information about the color layout of the frames, we apply a modified color layout descriptor (called here clsv). We define this descriptor via a localization carried out in a similar manner to the extraction method applied to the edge histogram descriptor. That is, we further divide the image into sub-blocks and determine the average color and DCT coefficients for them as well.

Several other MPEG-7 compliant features and other extensions or newly defined visual features are also provided to the user. For example, here we use color correlograms [8] and try to capture the relation between similar color agglomerates inside the image, thus using the very important spatial distribution of colors, which is normally lacking when using normalized color histograms only. A descriptor constructed of multiple color co-occurrence matrices is determined in order to obtain information about the spatial distribution of colors appearing in the video frame. That is, the co-occurrence of each color pair located a given distance *(k)* in the frame is calculated and saved as an integer number. In this way, we can obtain multiple co-occurrence matrices for varying distances between pixels. This is a very powerful descriptor, but it has the disadvantage of heaviness even when the maximum value for distance *(k)* is small. Thus, in our experiments we applied a simplification and used auto-correlograms only. That is, we calculated only the co-occurrences of the same colors for several different distances varying from *1* to *max(k)*. Thus, the calculation is simplified to determining only the diagonal of the matrices according to a given distance (k). We applied a maximum number of seven for the distance *(k)* in our experiments and we applied the *L2* distance to measure the similarity between video frames.

Annotation example results obtained as ranked video frame lists by using these features are shown in Figure 6. In these experiments we used three frames as training data for annotation "golf". Figure 6(a) illustrates a ranked video frame list obtained by using all of the three features, the auto-correlogram, the color layout descriptor and the edge histogram descriptor, by applying the same weight for each feature. In Figure 6(b), (c) and (d) ranked video frame lists are shown and these were obtained by using only one feature. In Figure 7(a) and (b), results obtained for annotation "soccer" and "waterfront" are illustrated by using all of the three features. These example results, especially those shown in Figure 6(a), and in Figure 7 demonstrate the robustness of these features even when only very simple ranking-based training is applied.

3.2.2 Feature extraction method

Based on the video features described in 3.2.1, SVMs for feature extraction were trained. The feature vector consisted of the following four features (880-dimensional in total): (1) a color layout feature for frame clustering (12-dimensional), (2) an edge histogram descriptor (150-dimensional), (3) a color layout descriptor[1] (270-dimensional) and (4) an auto-correlogram (448-dimensional). Gaussian Radial Basis Function (RBF) kernel was used instead of similarity, L1 and L2 distance, as used in 3.1.1.

For some high-level features, the shapes of objects may offer important clues for detecting target objects. Therefore, we applied an object extraction method using Haar-like features proposed by Voila et al [9] in TRECVID2006 HLFE task [10]. In TRECVID 2007, we applied this method to detect the following 9 features: **Face, Police-Security, Military, Animal, Flag-US, Airplane, Bus, Truck** and **Boat_Ship**. Furthermore, we applied the results of the "Face" extraction to detect human-related features.

The results of SVM-based extraction and Haar-like feature-based extraction were integrated by the following simple method. Only shots detected by both SVM-based extraction and Haar-like feature-based extraction are assumed to include a target feature. The rank of detected shots is then decided based on the SVM score. When multiple key-frames exist in a shot, shots in which at least one key-frame includes a target feature, are assumed to include a target feature. The rank of detected shots is decided based on the maximum SVM score of key-frames belonging in the shot.

Only SVM-based extraction was used in run A_KL4_4. Haar-like feature-based extractors used in each run, A_KL4_4, A_KL5_5 and A_KL6_6, are described in Table 1.

[1] Lower three frequency coefficients were used for each sub-block and then the total number of feature vector dimensions was reduced from 5760 to 270.

Figure. 6(a) Example list of frames obtained for annotation "golf" (TRECVID 2005 data) using all three features

Figure. 6(b) Example list of frames obtained for annotation "golf" (TRECVID 2005 data) using only auto-correlogram feature

Figure. 6(c) Example list of frames obtained for annotation "golf" (TRECVID 2005 data) using only color layout feature

Figure. 6(d) Example list of frames obtained for annotation "golf" (TRECVID 2005 data) using only EHD feature

Figure. 7(a) Example list of frames obtained for annotation "soccer" (TRECVID 2005 data) using all three features

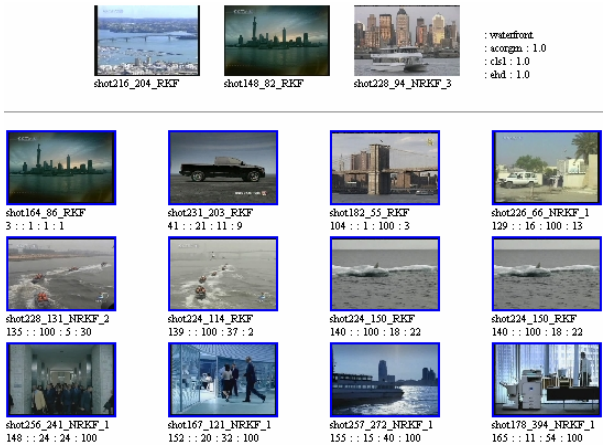


Figure. 7(b) Example list of frames obtained for annotation "waterfront" (TRECVID 2005 data) using all three features

Table 1. Harr-like feature-based extractors integrated with SVM-based extractors on each run.

Feature	runID	
	A_KL4_4	A_KL5_5, A_KL6_6
4. Court	-	∩Face
5. Office	-	∩Face
6. Meeting	-	∩Face
7. Studio	-	∩Face
18. Crowd	-	∩Face
19. Face	∩Face	∩Face
20. Person	∩Face	∩Face
23. Police	∩Police	∩Police∩Face
24. Military	∩Military	∩Military∩Face
25. Prisoner		∩Face
26. Animal	∩Animal	∩Animal∩¬Face
28. Flag-US	∩Flag-US	∩Flag-US
29. Airplane	∩Airplane	∩Airplane
31. Bus	∩Bus	∩Bus
32. Truck	∩Truck	∩Truck
33. Boat	∩Boat	∩Boat
34. Walking	-	∩Face
35. Marching	-	∩Face

SVMs were trained using TRECVID2005 and TRECVID2007 development data. Haar-like feature-based extractors were trained using TRECVID2003, TRECVID2005 and TRECVID2007 development data. Annotations of Video Collaborative Annotation Forum [11] and LSCOM Annotations [12] were used to select positive and negative training data. By comparison with A_KL4_4 and A_KL5_5, Haar-Like feature extractors with lower recall and higher precision were used in A_KL6_6.

4. EVALUATION RESULTS

This section describes results of each run that we submitted to the TRECVID2007 HLFE task and some additional experiments conducted after submission. Twenty features evaluated by NIST are used in the following evaluation.

Figure 8 shows inferred average precisions obtained with the color-based image retrieval method where key-frames are generated by the frame clustering method with threshold 5(A_KL1_1) and 20(A_KL2_2). Figure 9 shows inferred average precisions obtained with SVMs where key-frames are generated with threshold 5(A_KL3_3), 10 and 20. These results show that the precisions of SVM-based methods are superior to that of color-based image retrieval method. The precisions were improved by increasing the number of key-frames in a shot.

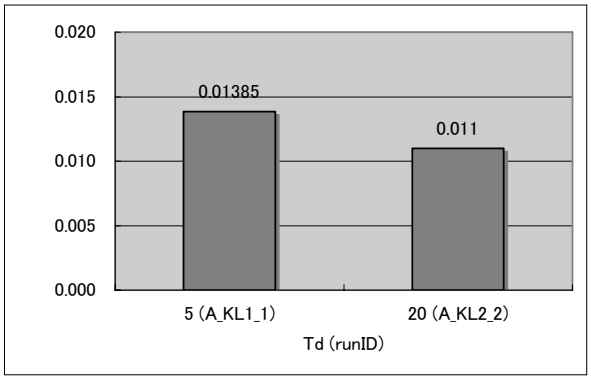


Figure 8. Inferred average precision obtained using color-based image retrieval methods on several thresholds for key-frame extraction (average of all 20 features).

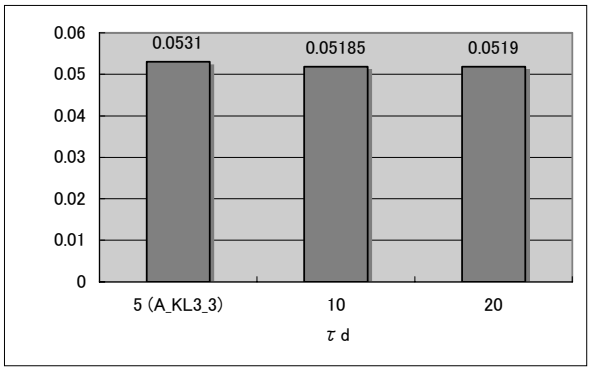


Figure 9. Inferred average precision obtained using SVMs on several thresholds for key-frame extraction (average of all 20 features).

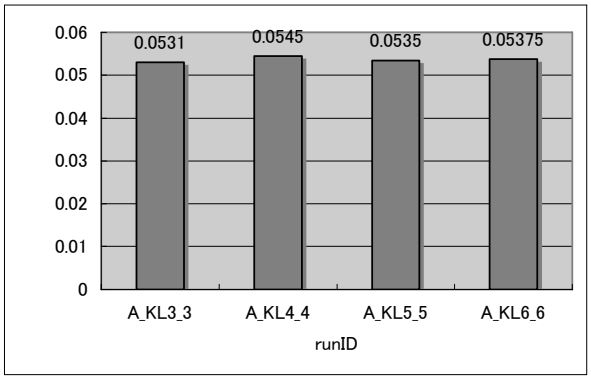


Figure 10. Inferred average precision obtained by integrating SVM-based extractors and Haa-like feature-based extractors (average of all 20 features).

Figure 10 shows the precisions obtained by integrating SVM-based extraction and Haar-Like feature extraction, where key-frames generated with threshold 5. In comparison with the precisions of A_KL3_3 which is conducted with only SVM, A_KL4_4, A_KL5_5 and A_KL6_6 which are conducted with Haar-Like feature-based extractors, bring improvements. The precision of HLFE was slightly improved by increasing the number of key-frames in a shot. The introduction of edge histogram descriptor, which describes both shape and textural properties, improves the precision without Haar-like feature-based extractors.

5. CONCLUSION

In this paper, key-frame extraction using a frame clustering method and two types of feature extractions were tested. The precision of HLFE was improved by increasing the number of key-frames in a shot. The introduction of edge histogram descriptor also improves the precision without Haar-like feature-based extractors.

However, the newly introduced features tend to become high-dimensional, therefore, reduction of the feature vector size will be better. A deep analysis such as Haar-like feature-based extraction seems to be promising, but it is difficult to prepare a training set. Thus, a semi-learning algorithm should be investigated for a deep analysis.

REFERENCES

[1] K. Hoashi *et al*.: ``Content-based Retrieval of User Generated Video Using Frame Clustering," Proc. of 2nd Korea-Japan Joint Workshop on Pattern Recognition (KJPR2007), 2007. (In Japanese)

[2] ISO/IEC 15938-3, ``Information technology --- Multimedia content description interface --- Part 3: Visual," 2002.

[3] M. Sugano *et al*.: ``MPEG content summarization based on compressed domain features analysis," Proc. SPIE Int'l Symposium ITCom2003, Vol.5242, pp.280-288, 2003.

[4] ISO/IEC 15938-8, ``Information technology --- Multimedia content description interface --- Part 8: Extraction and Use of MPEG-7 Descriptions," 2002.

[5] 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.

[6] K. Mail, and R. D. Gupta, "A Wavelet Based Image Retrieval," *Lecture Notes in Computer Science*, Springer-Verlag, No 3776, pp.557-562, 2005.

[7] B. S. Manjunath *et al*.: Color and Texture Descriptors. IEEE Transaction on Circuits and Systems for Video Technology, Vol. 11, No.6, pp. 703-715, 2001.

[8] J. Huang *et al*.: Image indexing using color correlograms, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 762-768, 1997.

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

[10] M. Naito, K. Matsumoto, M. Shishibori, K. Kita, M. Cuturi, T. Matsui, S. Sato, K. Hoashi, H. Kato, Sugaya, and Y. Nakajima: Shot Boundary Detection and High-Level Feature Extraction Experiments for TRECVID 2006, http://www-nlpir.nist.gov/projects/tvpubs/tv6.papers/kddi.pdf, 2006.

[11] 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

[12] 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