

# A Frame-Based Video Signature Method for Very Quick Video Identification and Location

---

Sang-il Na, Weon-Geun Oh, and Dong-Seok Jeong

**A video signature is a set of feature vectors that compactly represents and uniquely characterizes one video clip from another for fast matching. To find a short duplicated region, the video signature must be robust against common video modifications and have a high discriminability. The matching method must be fast and be successful at finding locations. In this paper, a frame-based video signature that uses the spatial information and a two-stage matching method is presented. The proposed method is pair-wise independent and is robust against common video modifications. The proposed two-stage matching method is fast and works very well in finding locations. In addition, the proposed matching structure and strategy can distinguish a case in which a part of the query video matches a part of the target video. The proposed method is verified using video modified by the VCE7 experimental conditions found in MPEG-7. The proposed video signature method achieves a robustness of 88.7% under an independence condition of 5 parts per million with over 1,000 clips being matched per second.**

**Keywords:** Video signature, frame descriptor, partial query, matching structure, matching strategy.

## I. Introduction

As the network bandwidth accessible by common users is expanding, video is becoming one of the fastest growing data transfer paradigms on the Internet. In particular, with the growing popularity of social media in Web 2.0, there has been an exponential growth in the number of videos available on the Internet. Users can easily download web videos and distribute them again with modifications. As an example, users upload 65,000 new videos each day on such video sharing websites as YouTube; the daily video views are well over 100 million [1]. Digital videos, which have become ubiquitous over the Internet, can be easily duplicated, edited, and redistributed. In considering content management, it would be helpful to devise some tools for use in video copy detection.

If a video copy detection technique is to be effective, the video signature should satisfy the following properties [2], [3].

- *Robustness* (invariance under perceptual similarity): the video signature extracted from a video clip after being subjected to content preserving distortions must be similar to the signature extracted from the original video clip.
- *Pair-wise independence* (collision free): if two video clips are perceptually different, the signatures extracted from them should be significantly different.
- *Fast matching*: the matching speed must be fast.

Content-based schemes extract their signature from the original media [4]-[10]. The extracted signature from the query media is compared with the target media signature to determine if the query is a copy of the target or the query contains a part of the original. The advantage of content-based copy detection over watermarking is that the original information is not changed.

Many different features have been proposed for use in video

---

Manuscript received May 8, 2012; revised Oct. 4, 2012; accepted Oct. 12, 2012.

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program.

Sang-il Na (phone: +82 42 860 1747, sina@etri.re.kr) and Weon-Geun Oh (owg@etri.re.kr) are with the Creative Content Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Dong-Seok Jeong (dsjeong@inha.ac.kr) is with the Department of Electronic Engineering, Inha University, Incheon, Rep. of Korea.

<http://dx.doi.org/10.4218/etrij.13.0112.0286>

and image signatures, for example, the edge histogram and color layout descriptor [4], color (luminance) histogram [5], mean luminance and its variants [6]-[9], dominant color [10], and centroid of gradient orientations [11]. Matching methods have also been proposed [8], [12]. Most of these methods use the frame rate information. However, in real applications, video frame rate information is subject to malicious attack, making the scenario of capturing the frame rate of the query clip stored in the video file unreliable.

If a video signature can be matched to short clips, this has many advantages. First, to determine whether or not a video clip is being used illegally, the duration of the video must be known. Second, many types of videos are short clips, and this method can be used for this type of content. Third, by selecting only a short part of a clip to match, the results can be inferred. This can be very helpful for the operation.

For short clip matching, a video signature using information about space is better than one using information about both time and space. Video consists of sequences of images and audio. In most cases, the timestamp of the video is audio-signal-based. Therefore, the frames per second (fps) of the image sequence changes slightly over time. Another factor is that sometimes the fps rate is altered during transcoding. If the video signature is extracted in the time domain, it will not have any problem when matching it with the long clip because it is possible to compensate. However, in the case of short clip matching, this becomes impossible. These characteristics also need to be considered in matching. In addition, the query parts of the video part of the goal should also be considered in the matching stage.

For the short clip matching, the frame-information-based approach has more advantages. Using the mean luminance and its variants [6]-[9] is fitting for this. However, the order-based approach has error propagation, as explained in section II. Also, the luminance-comparison-based approach does not have enough discriminability because it only uses the mean value and makes the comparison pairs with adjacent blocks. Additionally, these algorithms do not include a matching method for short clips.

In this paper, we present both a descriptor and a matching method. The proposed descriptor uses a block average value comparison that is robust regarding various modifications of the video content. The matching structure's matching speed is fast, even though it does not use the frame rate information. The proposed method is verified using modified video, which is modified by the VCE7 experimental conditions found in MPEG-7 [13].

## II. Proposed Algorithm

In this section, we discuss the proposed algorithm. This

section is divided into four subsections. First, we describe the modeling of our descriptor in relation to the problems found in the previous methods. The proposed frame and spatial descriptors are then presented. Finally, we discuss the matching structure and strategy.

### 1. Frame Descriptor Modeling

To extract the descriptor using the spatial information, it must be extracted using information from the video frames. A frame-based descriptor must be robust against video modifications and sensitive in localization determination. Therefore, in our frame descriptor design, we model the video modifications.

Generally, the luminance component of an image includes more information than the chrominance. Modification using the luminance component can be placed into two classes: geometrical and non-geometrical. Geometrical modification changes the location of a pixel. Scaling, pillarboxing, and letterboxing methods are included in this type of modification. A modified frame can be expressed by

$$I'(x, y) = \alpha I(\bar{x}, \bar{y}) + \beta, \quad (1)$$

where  $I'(x, y)$  is the modified pixel value in the  $(x, y)$  location and  $(\bar{x}, \bar{y})$  is the corresponding location of  $(x, y)$  in an original image. The relationship of the location can be expressed by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \\ 1 \end{bmatrix}, \quad (2)$$

where  $a, b, c,$  and  $d$  represent the position change of the  $x, y$  location and  $e$  and  $f$  represent the translation. Generally, a rotation transformation does not occur in video modifications, so the  $b$  and  $c$  values are 0. The  $a$  and  $d$  values in (2) represent scale factors whose values usually control the values of the modifications resulting from pillarboxing or letterboxing.

Non-geometrical modifications can be represented by the following equations:

$$I'(x, y) = I(x, y) + \beta, \quad (3)$$

$$I'(x, y) = \alpha I(x, y), \quad (4)$$

$$I'(x, y) = \sum_{i=-\frac{M}{2}}^{\frac{M}{2}} \sum_{j=-\frac{M}{2}}^{\frac{M}{2}} \gamma_{i,j} I(x, y). \quad (5)$$

These equations represent brightness changes, contrast changes, and convolutional filtering, respectively. These modifications are representative of video modifications to a frame. Most of the non-geometrical video modifications obey

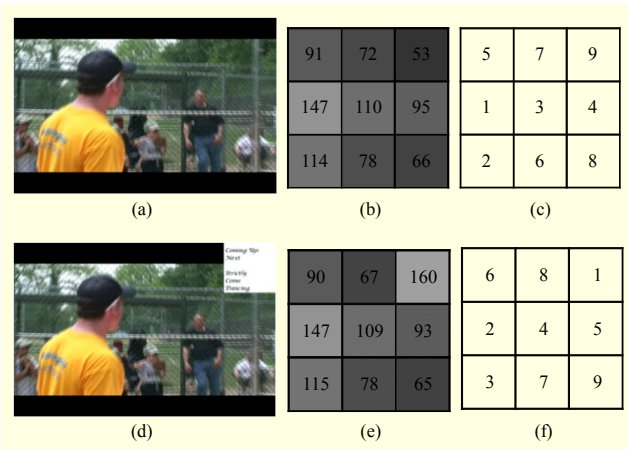


Fig. 1. Example of order in which image is corrupted by text/logo overlay. (a) original image, (b) average intensity value in (a), (c) order of (b), (d) text/logo overlaid image on (a), (e) average intensity value in (c), and (f) order of (e).

linear characteristics, so they can be approximated by (3) through (5).

As shown in (3) through (5), the relationship between pixels does not change, even if the modifications occur in the video frame. Equation (6) shows these relationships.

$$I(x_a, y_a) > I(x_b, y_b), \text{ then } I'(x_a, y_a) > I'(x_b, y_b). \quad (6)$$

Here, different positions are described by  $(x_a, y_a)$ ,  $(x_b, y_b)$ ; the relationships of the different positions do not change even if the modifications occur in the video. However, some modifications do not follow this relationship type, such as the noise addition. In this case, if we use the block average pixel value, the rule is maintained. Equation (7) shows the block average, and (8) shows the expansion of (6) using the block average value.

$$B(x, y) = \frac{1}{w \times h} \sum_{i=1}^w \sum_{j=1}^h I(xw+i, yh+j), \quad (7)$$

$$B(X_a, Y_a) > B(X_b, Y_b), \text{ then } B'(X_a, Y_a) > B'(X_b, Y_b). \quad (8)$$

This comparison method, which uses an image divided into subblocks, has the advantage of being robust to modifications. We obtain a greater advantage if the subblock size is larger because the mean pixel value of a larger area is more robust to noise, so (6) becomes even more suitable.

Kim presented a representative method in [6] based on these properties. However, if one of the blocks becomes corrupted by noise, such as a text or logo overlay, it influences other blocks. Figure 1 shows an ordinal measure example in the case of noise corruption. It shows that if a block gets corrupted by noise, it influences other blocks. Therefore, we need to design our system so that noise does not influence the other descriptor values.

A larger subblock is more robust to geometric modifications.

Table 1. Percentage of region change when divided into five blocks (%).

Subblock position	1	2	3	4	5
Region change	50	30	20	30	50

Table 2. Percentages of region change when divided into 10 blocks (%).

Subblock position	1	2	3	4	5	6	7	8	9	10
Region change	100	80	60	40	20	20	40	60	80	100

Most geometric modifications in video are caused by pillarboxes or letterboxes, which change the aspect ratio. A pillarbox is inserted when the 4:3 ratio is changed to 16:9; a letterbox is inserted when the 16:9 ratio is changed to 4:3. Both of these cases insert 25% more block regions; 12.5% of the new regions are inserted to either the left and right or top and bottom sides. The ratios of the changed regions are as follows. Table 1 shows the region change percentage if the image is divided into five blocks, and Table 2 shows the change for a 10-block division. These values are obtained through manual calculation.

The probability of changing the relative size of a region in the outer block is about 25% if the image is divided by five. This means that it is about 75% robust.

To address these shortages, we need to prevent a noisy block from affecting other blocks. Therefore, we design a binary descriptor using the spatial information, which has a high level of robustness and independence.

## 2. Proposed Spatial Descriptor

As mentioned above, the spatial descriptor must be robust against modification and sensitive to localization. To satisfy these characteristics, the descriptor must encompass enough bits; each bit must be robust and be discriminable. In this subsection, we discuss a binary descriptor that satisfies both robustness and sensitivity.

Generally, adjacent blocks have a similar value, so they have a higher discriminability, but the robustness is low. Table 3 shows the average difference value according to the block distance. For this table, we extract 6,000 frames from a video; each frame divided into  $5 \times 5$  blocks. As shown in Table 3, a distance of more than two block pairs has a greater difference value, making it more robust.

We also avoid using a block for comparison where the block

**Table 3.** Distribution of average difference value between average block mean by distance.

Distance of block	1	2	3	4	5
Average of difference value	20	39	44	42	40

is already in use for the same comparison pattern. If the same block is used, the comparison pattern correlation becomes too high and makes it difficult to get sufficient discriminability. For example, if the average value of the 0th block is larger than the 1st block, then the probability of the average value of the 1st block is greater than the 2nd block or higher than 50%. In our observation, using 6,000 frames shows about a 60% correlation. It is destructive to independence if the same bits are used.

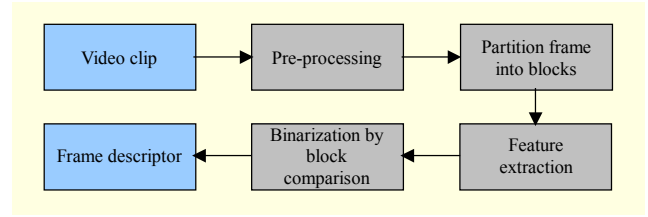
### 3. Proposed Frame Descriptor

Based on these observations, we incorporate comparison pairs, which have a distance of more than two blocks. We also use only a single time for each block in the same comparison pattern. Figure 2 shows an overview of the proposed descriptor. The proposed descriptor uses spatial features from all of the video frames, so the video clip is decoded and the decoded frame is resized to 100×100 pixels. We use the bicubic interpolation method to resize the frames. We convert the resized frames to grayscale (one-channel) images, since the proposed algorithm only uses the luminance component. After the normalization process, we build two levels of descriptors: coarse and fine. In the case of the coarse descriptor, we divide the normalized image into 5×5 subblocks and generate comparison patterns. The comparison patterns are generated using three subblock values. The three subblock values are the average and the x and y directions difference values inside the subblock, as shown in Fig. 3. In Fig. 3, we show the 9th subblock's feature pattern, which is similarly extracted for all of the subblocks. The equations for each of the block features are as follows:

$$Average[n] = \frac{1}{M \times M} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} B_n(i, j), \quad (9)$$

$$DiffY[n] = \sum_{i=0}^{M-1} \sum_{j=0}^{\frac{M-1}{2}} B_n(i, j) - \sum_{i=0}^{M-1} \sum_{j=\frac{M}{2}}^{M-1} B_n(i, j), \quad (10)$$

$$DiffX[n] = \sum_{i=0}^{\frac{M-1}{2}} \sum_{j=0}^{M-1} B_n(i, j) - \sum_{i=\frac{M}{2}}^{M-1} \sum_{j=0}^{M-1} B_n(i, j), \quad (11)$$



**Fig. 2.** Proposed descriptor overview.

where  $Average[n]$  is the average value of the  $n$ -th block, and  $DiffY[n]$  and  $DiffX[n]$  are the x-direction and the y-direction differentiation, respectively.  $M$  is the block width and height; for our purposes, we choose a value of 20 because this value ensures that the change to the region is less than 50%, as shown in Table 1. The pixel value at position  $(i, j)$  in the  $n$ -th block is represented by  $B_n(i, j)$ .

We convert these values into a binary descriptor for comparison. For the binarization, we use 72 subblock pairs and select the values before binarization. The rule used in making the pair is to remove the correlation. So, if feasible, we compare the values once for each block for each feature. Subsequently, for each comparison we carry out binarization, so, if one block is corrupted by noise, it influences just one bit for each feature.

The outline for the procedure for generating the values is as follows.

1)  $DiffY[n]$  and  $DiffX[n]$  values: The  $DiffY[n]$  values are calculated for all subblocks except the top and bottom blocks because these blocks are not robust against letterboxing (15 values); In the same manner,  $DiffX[n]$  values are calculated for all subblocks except the left and right blocks because these blocks are not robust against pillarboxing (15 values).

2) The comparison pairs make use of point symmetry from the origin, which is the 12th subblock. For the same reason as above, the number of  $DiffY[n]$  and  $DiffX[n]$  value comparison pairs is seven for each (14 values). The  $Average[n]$  values of the comparison pairs are the same as those of the  $DiffX[n]$  comparison pairs (7 values). We carry out the binarization on these comparison pairs by following (12).  $Value[x]$  is the resulting binary value and  $x$  is the subtracting value of the comparison pairs.

$$Value[x] = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

3) We generate one value using the sum of the  $DiffY[n]$  and  $DiffX[n]$  values for all subblocks. The comparison pairs for this case are made using point symmetry whose position is oriented in the center region and binarization is carried out using (12) (4 values).

4) We compare the large regions. The comparison regions are



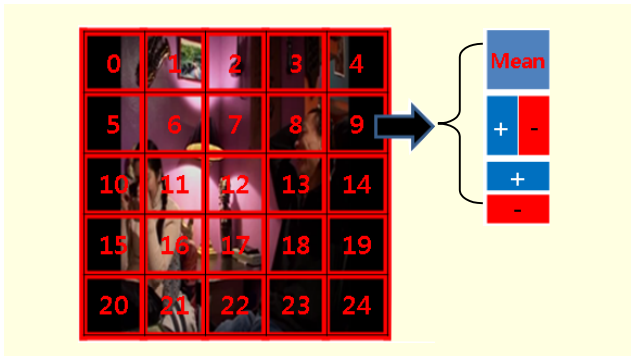


Fig. 3. Subblock pattern.



Fig. 4. Image division in case of fine descriptor.

- (0th, 1st, 5th, 6th, 10th, 11th, 15th, 16th, 20th, and 21st subblocks) versus (3rd, 4th, 8th, 9th, 13th, 14th, 18th, 19th, 23rd, and 24th subblocks);
- 0th through 9th subblocks versus 15th through 24th subblocks; and
- 12th subblock versus the subblocks of its eight neighbors.

We also carry out the binarization on these values using (12) (3 values).

5) The average value of a subblock is compared with the average value of the center region's subblocks, excluding the 12th subblock  $([6\text{th}+7\text{th}+8\text{th}+11\text{th}+13\text{th}+16\text{th}+17\text{th}+18\text{th}+19\text{th}]/8)$  versus each subblock that uses this average) (8 values).

6) Second differentiation values are calculated by using (13) and (14). In (13) and (14),  $n$  is the subblock index, as shown in Fig. 3. These values are calculated in the center region (6th, 7th, 8th, 11th, 12th, 13th, 16th, 17th, and 18th subblocks) (6 values).

$$\begin{aligned} \text{Diff}2X[n] = & \text{Average}[n-1] \\ & + \text{Average}[n+1] - 2 \times \text{Average}[n] \end{aligned} \quad (13)$$

$$\begin{aligned} \text{Diff}2X[n] = & \text{Average}[n-1] \\ & + \text{Average}[n+5] - 2 \times \text{Average}[n] \end{aligned} \quad (14)$$

Using these 72 values, we can make a 72-bit binary descriptor for coarse matching. This 72-bit descriptor is not sufficient to distinguish between two different frames for localization. Therefore, we extend this descriptor by adding a

fine descriptor. We calculate the fine descriptor using the same method used to create the coarse descriptor. We first divide one frame into four regions, as shown in Fig. 4, and obtain a 72-bit descriptor for each region.

The distance measure, which indicates the similarity between the frame descriptors, uses the hamming distance, which means that the number of positions for the corresponding binary bit values of the binarized frame descriptor is different.

#### 4. Matching the Structure and Strategy

By now, every frame in the video clip has two types of binary descriptors, that is, coarse and fine. The binary form of the coarse descriptor is used for the coarse matching. For efficient matching, we use the index table for the coarse descriptor. After coarse matching, we find the candidate segments and then apply the fine descriptor to find the precise time location.

In the coarse matching stage, we find the candidate points, which represent the positions of the matched frames between the query clip and the target clip. The matched frames are selected by comparing the coarse descriptor and calculating the distance between the two frames. We need to define a threshold for the error tolerance of the number of bits difference between the binary descriptors. To empirically calculate the threshold, we assume that the extraction process yields random independent and identically distributed (i.i.d.) bits. The number of bit errors between the descriptors from different frames will then have a binomial distribution  $B(n, p)$ , wherein  $n$  is equal to the number of bits extracted and  $p$  is the probability that a "0" or "1" bit is extracted. If  $n$  is sufficiently large, the binomial distribution can be approximated as a normal distribution. Therefore, its mean is  $np$  and the standard deviation is  $\sqrt{np(1-p)}$ . From this, it can be deduced that the bit error rate (BER) has a normal distribution with mean  $u=p$  and a standard deviation of  $\sigma = \sqrt{p(1-p)/n}$ . In an ideal case,  $p=0.5$ . Through the normal approximation  $N(\mu, \sigma)$ , the probability of false alarm  $P_{FA}$  for the BER is as in (15) [14].

$$P_{FA} = \frac{1}{2} \text{erfc}\left(\frac{u-T}{\sqrt{2}\sigma}\right). \quad (15)$$

In this paper, we use a threshold  $T=0.24$ , which gives a  $P_{FA}$  of less than 0.1%.

After the coarse matching process, we obtain a candidate map, as shown in Fig. 5. In Fig. 5, the red cells indicate the matched point candidates resulting from the coarse matching.

After generating the candidate map, we apply geometric labeling to the matched cells. During the geometric labeling, we connect each label if a consequent label exists on a  $45^\circ$

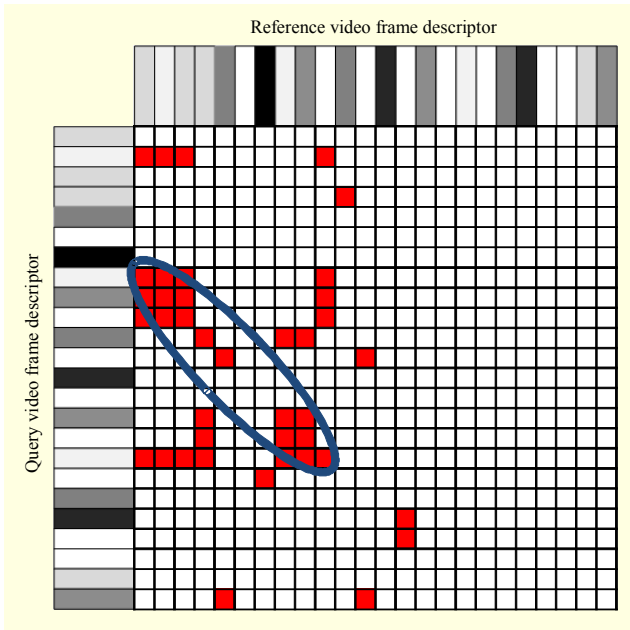


Fig. 5. Candidate cells obtained from coarse descriptor matching.

40	44	54	69	78	80	93	95	89	92	92	104	102	92	97	92	101	112	120	126	132
24	28	42	57	66	68	83	89	83	84	86	100	96	90	93	88	101	112	118	126	130
30	22	36	57	66	70	83	85	87	88	94	102	100	92	95	94	107	118	124	128	134
43	35	17	44	51	59	74	76	76	85	97	101	101	95	96	93	108	115	125	131	135
61	55	45	18	33	43	66	76	78	91	107	123	119	115	114	109	120	123	135	141	145
89	63	53	36	19	31	54	66	76	89	105	121	117	113	110	105	118	119	129	133	137
75	71	65	46	33	21	48	60	76	85	101	123	119	115	112	107	114	115	125	129	131
87	81	77	72	59	45	24	40	58	67	85	117	111	109	108	103	110	105	113	111	111
92	84	78	77	70	54	35	21	45	56	78	106	100	98	97	92	93	90	102	100	106
90	88	78	81	80	70	53	47	23	34	56	86	84	84	83	78	85	78	88	94	94
91	89	87	92	93	81	68	60	34	27	47	75	71	73	72	67	76	77	81	89	87
90	92	98	109	110	96	81	79	55	40	28	62	62	68	65	64	67	70	73	80	78
102	102	104	123	126	120	109	107	83	74	58	28	32	44	47	54	64	65	72	84	90
101	101	105	120	121	117	108	106	86	75	67	45	29	39	42	53	64	65	72	85	89
99	97	105	120	121	117	110	106	88	79	71	53	37	37	40	51	64	69	75	87	87
97	95	101	116	113	111	106	100	84	75	65	55	45	41	26	39	68	73	81	95	103
93	93	97	112	107	105	100	96	80	71	65	65	61	59	52	45	50	63	67	85	93
101	101	105	114	113	103	100	92	84	71	65	65	61	59	52	45	24	41	55	71	77
112	112	116	123	122	110	101	93	81	70	70	70	64	66	67	68	45	24	44	60	66
121	121	125	132	129	117	108	102	88	79	77	77	73	75	76	69	46	43	29	53	59

Fig. 6. Distance table for fine descriptor and example of possible fps.

diagonal axis. The numbers in the red boxes are the assigned label numbers in Fig. 5. We roughly decide the matched segments from the labeled box positions. The matched candidate region is presented from the top left corner to bottom right corner. The matched segments that are small in size are likely to have the same contents in the query clips and target clips, so we discard them from the candidate segments.

Although we find roughly matched candidate segments of the target and the query through the coarse descriptor matching, we must verify that the matched segments truly consist of the same contents in a specific time location. To find the exact position, we use the fine descriptor in the matched segment. The difference between the target and the query pair is calculated for the fine descriptor just as the difference is calculated in the coarse descriptor matching. After calculating the distances of all of the possible pairs in the fine descriptor, we find the line from the fine descriptor distance in the

candidate segment.

To calculate the fine distance, we must limit the fps range. Figure 6 shows an example of a fine descriptor distance table and its possible fps. We already have a candidate segment, so we have knowledge regarding the number of query frames that matched the target frames. We want to find one matching segment that is longer than the given durations. From this information, we can estimate the minimum and maximum boundaries of the frame rate for the query clip according to

$$fps_{\max} = \frac{N_{MQ}}{T_{\min}}, \quad (16)$$

$$fps_{\min} = \frac{N_{MR}}{T_R}, \quad (17)$$

where  $N_{MQ}$  is the matched number of frames in the query,  $N_{MR}$  is the matched number of frames in the reference,  $T_{\min}$  is the minimum matching time, and  $T_R$  is the matched time in the reference video.

Using this information, we can determine a distance for each estimated frame rate by using

$$D_{\text{line}}(x) = \frac{1}{K} \sum_{i=0}^{K-1} D_{\text{Hamming}}(F_Q(i * x), F_T(i)), \quad (18)$$

where  $K$  represents the matched query frames,  $D_{\text{Hamming}}$  is the Hamming distance between the descriptors, and  $F_Q(i)$  and  $F_T(i)$  are the query fine descriptor and target fine descriptor, respectively, at the  $i$ -th position within the candidate. The value obtained by dividing the target clip frame rate value by the estimated query frame rate is  $x$ . The distance found using the target clip frame rate and the estimated query clip frame rate is  $D_{\text{line}}$ .

When calculating the distance stage, one must calculate the distance for all possible frame rates. However, this is too exhaustive an approach when we consider all of the cases of the estimated query frame rate, so we adopt the coarse to fine estimation approach. We first calculate the increment distance with a step size of five. In the previous example, we calculated the distances for 15 fps, 20 fps, 25 fps, and 30 fps. We then choose their minimum distance and carry out a refinement using a step size of one in the neighborhood of the selected data.

### III. Experiment

#### 1. Experiment Conditions

In this paper, we use MPEG-7's VCE7 for the experiment conditions [13]. In VCE7, for each comparison between the query clip and the target clip, the proposed algorithm is required to give a binary decision that classifies clips as related

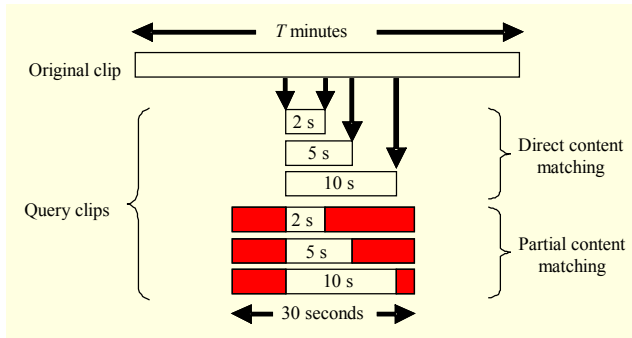


Fig. 7. Different query scenarios.

or unrelated. In the case of related clips, additional information is needed regarding the time position of the matched position.

The experiments use two different query scenarios: direct content matching and partial content matching. Direct content matching uses the case in which the entire query clip is matched to a certain part of the original clip. The algorithm is required to give the starting point of the matched fragment in the target clip. Partial content matching uses the case in which a part of the query clip matches with a part of the original clip. In this case, the query clip may contain content not present in the original clip and the original clip may contain content not present in the query clip. The algorithm provides only the minimum duration of the segment to be matched, and the actual duration of the matching part is unknown. The algorithm searches for any matching parts longer than this minimum duration. It is required to give the starting point and the end point of only one matched part between the target clip and the query clip. Figure 7 shows six different query types. The white blocks in the query clips reflect the corresponding parts of the original clip; the red blocks reflect the differences between the query clips and the original clip.

For each of the two query scenarios, this experiment is evaluated using three different durations ( $D$ ) during which the segment should be matched, that is,  $D=2$  s,  $D=5$  s, and  $D=10$  s. In the case of the partial scenario, the total duration of the clips is 30 seconds.

For these scenarios, we must define a threshold. We obtain the threshold using the independence test. In the independence test, all of the comparisons that are declared to be related are counted as false positives. The operational settings that achieve a false positive rate of less than 5 parts per million (ppm) can be identified in this manner.

In the robustness test, the detection capabilities in the presence of various modifications are evaluated. Table 4 shows the various modifications and levels that are used in the experiment.

The percentage used to define the values in the text/logo overlay indicates how much of the area is corrupted by the

Table 4. Modifications and levels (9 modifications, 22 categories).

Modifications \ Levels	Heavy	Medium	Light
Text/logo overlay	30%	20%	10%
Severe compression (at CIF resolution)	64 kbps	256 kbps	512 kbps
Resolution reduction (from SD)	N/A	QCIF	CIF
Frame-rate reduction (from 30/25 fps)	4 fps	5 fps	15 fps
Capturing on camera (at SD resolution)	10%	5%	0%
Analog VCR recording & recapturing (100% of image captured)	3 times	2 times	1 time
Color to monochrome conversion	N/A	N/A	$I = 0.299R + 0.587G + 0.114B$
Brightness change	+36	-18	+9
Interlaced/progressive conversion	N/A	N/A	$P \rightarrow I \rightarrow P$ $I \rightarrow P$

text/logo. The percentage used in capturing content on camera correlates to the percentage of the extra background area.

In the robustness test, we have two different types of queries, so we have two different types of success conditions: direct success and partial success. In the direct success condition, the difference in the starting position between the ground truth position and the estimated position must be less than 1 second. In the partial success condition, the difference in the duration of the matched positions and the difference of the start and end positions between the ground truth position and the estimated position must be less than 1 second.

In this study, we develop the frame descriptor and the matching structure and strategy for the video signature. The video signature is completely verified using the above conditions; however, the frame descriptor cannot be verified in that manner, so we test it with the help of captured images from both the original video clips and all of the modified video clips except the clip resulting from frame rate reduction modification because the captured still image from the frame rate reduction clip is exactly the same as the original.

## 2. Frame Descriptor Results

To evaluate the performance of our proposed frame descriptor, we use captured frame images from video sequences in an experiment dataset. We extract frame descriptors from 6,000 frames captured from original video clips and 90,000 modified frames captured from modified

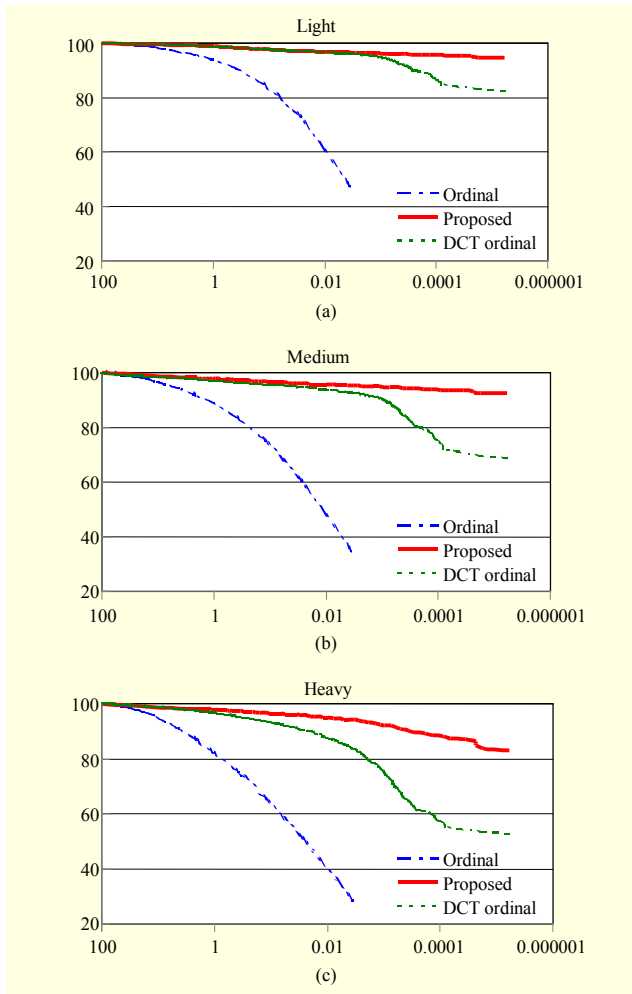


Fig. 8. ROC curves for proposed descriptor and other methods: (a) light level modification average, (b) medium level modification average, and (c) heavy level modification average.

video clips. To test the independence, we compare the 6,000 original images for possible pairs. To show the robustness performance, the results from the levels and modifications are averaged for each level and region of convergence (ROC) curves are drawn. Figure 8 shows the ROC curves for the proposed method and other methods. In Fig. 8, the  $x$  axis represents the independence value in a log scale; the  $y$  axis represents the robustness value. Both axes' units are in percent. Because our proposed video signature is based on the frame image descriptor, we compare our proposed algorithm to conventional image copy detection methods: the ordinal method [9] and the discrete cosine transform (DCT) ordinal method [15]. The DCT ordinal method also uses image signatures; however, our proposed method is based on the frame descriptor, so the DCT ordinal method can be used to compare the performances.

The proposed method uses coarse and fine matching. As

Table 5. Average success ratio for all modifications (%).

Algorithm		Levels			
		Heavy	Medium	Light	Mean
Direct 2 s	Proposed	81	89	93	88
	Kim's ordinal	53	58	65	59
	Ordinal	39	45	55	46
	Tomography	61	68	72	67
Direct 5 s	Proposed	86	93	96	92
	Kim's ordinal	55	61	67	61
	Ordinal	41	48	57	49
Direct 10 s	Proposed	87	93	96	92
	Kim's ordinal	56	62	69	62
	Ordinal	41	48	57	49
Partial 2 s	Proposed	80	86	90	85
	Proposed	83	90	92	88
	Proposed	81	87	93	87

Table 6. Matching speed test results.

Algorithm	Matches per second	Matching time
Proposed	5,574	1
Kim's ordinal	1,475	3.78
Ordinal	1,621	3.44
Tomography	1,818	3.07

shown in Fig. 8, the robustness of the proposed method is better than that of the other methods under various modifications. We show the average robustness result for each modification level to save space.

### 3. Video Signature Results

The proposed algorithm is evaluated using the specifications found in [13]. The proposed algorithm is fully tested; however, since the comparison methods do not focus on partial query scenarios, the comparison for the proposed method and the other methods are tested using only the direct query scenario. We set the threshold as described previously. However, in the case of some of the compared algorithms, when the result of the independence test satisfies the 5-ppm condition, the distance value used to determine the threshold is close to zero. Therefore, we adopt a 500-ppm condition for the threshold if the algorithms fail to satisfy the 5-ppm condition.

Table 5 shows the success ratio of our algorithm and the



compared algorithms. We only show the averages of the overall results for each modification level and query type using the VCE7 database to save space. Our algorithm is evaluated for all six query types. However, the compared methods are evaluated for only three query types: the direct two-second (Direct 02), the direct five-second (Direct 05), and the direct ten-second (Direct 10). This is because the compared algorithms are difficult to apply in the partial query scenarios. Our algorithm and the tomography algorithm [16] set the threshold with independence testing using the 5-ppm condition, and the ordinal algorithm [9] and Kim's ordinal algorithm [6] use the 500-ppm threshold. Our algorithm does not use frame rate information, but all of the compared algorithms do. We show the full experiment results of our algorithm in the appendices.

Table 6 shows the results of the matching speed test. The matches per second in the table tell us how many clips are matched per second. The matching time is calculated using the results of our algorithm as the base to which the results of the other algorithms are compared. In this case, we use direct 10-second queries and three-minute target clips. We match 100 query clips to 100 target clips and check the average matching time.

#### IV. Conclusion

This paper presented a frame-based video signature method and a coarse-to-fine matching structure. The signature was designed for all frames that are pair-wise independent and robust against various modifications. The performance was evaluated using MPEG-7's VCE7 database and experimental conditions. As shown in the experiment, our algorithm achieves a 90% average success ratio in direct queries as compared to the 69% achieved by the tomography algorithm and the 61% and 68% achieved by other approaches. In addition, the matching speed is about three times faster than that found for the other algorithms. Another merit of the proposed algorithm is the discarding of the frame rate information. Even though the performance of our algorithm is slightly deteriorated under the partial query situations, we can find matches for both query types by using the proposed matching structure and strategy, which is an additional advantage offered by our algorithm.

#### Appendix

Direct 02

Modifications	Levels			
	Heavy	Medium	Light	Mean
Analog VCR recording & recapturing	0.89	0.92	0.96	0.92
Brightness change	0.94	0.91	0.95	0.93
Capturing on camera	0.65	0.87	0.86	0.79
Frame-rate reduction	0.8	0.85	0.93	0.86
Interlaced/progressive conversion	N/A	N/A	0.94	0.94
Color to monochrome conversion	N/A	N/A	0.94	0.94
Resolution reduction	N/A	0.93	0.94	0.94
Severe compression	0.91	0.93	0.94	0.93
Text/logo overlay	0.65	0.81	0.92	0.79
Average				0.89

Direct 05

Modifications	Levels			
	Heavy	Medium	Light	Mean
Analog VCR recording & recapturing	0.95	0.96	0.98	0.96
Brightness change	0.96	0.94	0.97	0.95
Capturing on camera	0.76	0.93	0.93	0.88
Frame-rate reduction	0.95	0.95	0.96	0.95
Interlaced/progressive conversion	N/A	N/A	0.96	0.96
Color to monochrome conversion	N/A	N/A	0.96	0.96
Resolution reduction	N/A	0.96	0.96	0.96
Severe compression	0.94	0.96	0.96	0.95
Text/logo overlay	0.62	0.82	0.94	0.79
Average				0.93

Direct 10

Modifications	Levels			
	Heavy	Medium	Light	Mean
Analog VCR recording & recapturing	0.97	0.98	0.99	0.98
Brightness change	0.96	0.94	0.96	0.95
Capturing on camera	0.82	0.94	0.95	0.9
Frame-rate reduction	0.93	0.94	0.96	0.94
Interlaced/progressive conversion	N/A	N/A	0.96	0.96
Color to monochrome conversion	N/A	N/A	0.96	0.96
Resolution reduction	N/A	0.96	0.96	0.96
Severe compression	0.94	0.96	0.96	0.95
Text/logo overlay	0.57	0.78	0.94	0.76
Average				0.93

Partial 02

Modifications \ Levels	Levels			
	Heavy	Medium	Light	Mean
Analog VCR recording & recapturing	0.88	0.9	0.93	0.9
Brightness change	0.94	0.9	0.94	0.92
Capturing on camera	0.65	0.85	0.85	0.78
Frame-rate reduction	0.87	0.88	0.92	0.89
Interlaced/progressive conversion	N/A	N/A	0.94	0.94
Color to monochrome conversion	N/A	N/A	0.94	0.94
Resolution reduction	N/A	0.93	0.93	0.93
Severe compression	0.86	0.92	0.91	0.9
Text/logo overlay	0.6	0.65	0.73	0.66
Average				0.87

Partial 05

Modifications \ Levels	Levels			
	Heavy	Medium	Light	Mean
Analog VCR recording & recapturing	0.94	0.95	0.96	0.95
Brightness change	0.94	0.93	0.93	0.94
Capturing on camera	0.8	0.92	0.91	0.88
Frame-rate reduction	0.86	0.93	0.92	0.92
Interlaced/progressive conversion	N/A	N/A	0.93	0.95
Color to monochrome conversion	N/A	N/A	0.93	0.95
Resolution reduction	N/A	0.95	0.93	0.95
Severe compression	0.92	0.94	0.93	0.94
Text/logo overlay	0.41	0.66	0.91	0.63
Average				0.9

Partial 10

Modifications \ Levels	Levels			
	Heavy	Medium	Light	Mean
Analog VCR recording & recapturing	0.94	0.96	0.96	0.95
Brightness change	0.94	0.91	0.93	0.93
Capturing on camera	0.8	0.91	0.91	0.87
Frame-rate reduction	0.86	0.9	0.92	0.89
Interlaced/progressive conversion	N/A	N/A	0.93	0.93
Color to monochrome conversion	N/A	N/A	0.93	0.93
Resolution reduction	N/A	0.88	0.93	0.91

Resolution reduction	N/A	0.88	0.93	0.91
Severe compression	0.92	0.93	0.93	0.93
Text/logo overlay	0.41	0.58	0.91	0.63
Average				0.89

## References

- [1] Wikipedia. <http://en.wikipedia.org/wiki/YouTube>
- [2] T. Kalker, J.A. Haitisma, and J. Oostveen, "Issues with Digital Watermarking and Perceptual Hashing," *Proc. SPIE, Multimedia Syst. Appl. IV*, Nov. 2001, pp. 189-197.
- [3] J.S. Seo et al., "Audio Fingerprinting Based on Normalized Spectral Subband Moments," *IEEE Signal Process. Lett.*, vol. 13, no. 4, Apr. 2006, pp. 209-212.
- [4] S.M. Kim, S.J. Park, and C.S. Won, "Image Retrieval via Query-by-Layout Using MPEG-7 Visual Descriptors," *ETRI J.*, vol. 29, no. 2, Apr. 2007, pp. 246-248.
- [5] S.S. Cheung and A. Zakhor, "Efficient Video Similarity Measurement with Video Signature," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, Jan. 2003, pp. 59-74.
- [6] C. Kim and B. Vasudev, "Spatiotemporal Sequence Matching for Efficient Video Copy Detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, Jan. 2005, pp. 127-132.
- [7] J. Oostveen, T. Kalker, and J. Haitisma, "Feature Extraction and a Database Strategy for Video Fingerprinting," *Proc. Int. Conf. Recent Adv. Visual Inf. Syst.*, 2002, pp. 117-128.
- [8] X. Hua, X. Chen, and H. Zhang, "Robust Video Signature Based on Ordinal Measure," *Proc. Int. Conf. Image Process.*, Singapore, Oct. 24-27, 2004, pp. 685-688.
- [9] R. Mohan, "Video Sequence Matching," *Proc. Int. Conf. Audio, Speech, Signal Process.*, IEEE Signal Processing Society, vol. 6, 1998, pp. 3697-3700.
- [10] A. Hampapur and R.M. Bolle, "VideoGREP: Video Copy Detection Using Inverted File Indices," Technical Report, IBM Research, 2001.
- [11] S. Lee and C.D. Yoo, "Robust Video Fingerprinting for Content-Based Video Identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, July 2008, pp. 983-988.
- [12] M.-C. Yeh and K.-T. Cheng, "Video Copy Detection by Fast Sequence Matching," *Proc. ACM Int. Conf. Image Video Retrieval*, Apr. 2009.
- [13] MPEG Video Sub-Group, "Description of Core Experiments in Video Signature Description Development," ISO/IEC JTC1/SC29/WG11, w10345, Lausanne, Switzerland, Feb. 2009.
- [14] J.S. Seo et al., "A Robust Image Fingerprinting System Using the Radon Transform," *Signal Process.: Image Commun.*, vol. 19, 2004, pp. 325-339.
- [15] C. Kim, "Content-Based Image Copy Detection," *Signal*

*Process.: Image Commun.*, vol. 18, no. 3, Mar. 2003, pp. 169-184.

- [16] P. Sebastian and H. Kalva, "Accuracy and Stability Improvement of Tomography Video Signatures," *IEEE Int. Conf. Multimedia and Expo*, July 2010, pp. 133-137.



**Sang-il Na** received his BS, MS, and PhD from Inha University, Incheon, Rep. of Korea, in 2002, 2004, and 2010, respectively. He now works at ETRI, Daejeon, Rep. of Korea, as a research engineer. His research interests include image and video processing and image and video signature and retrieval.



**Weon-Geun Oh** received his BS from Chungbuk National University, Cheongju, Rep. of Korea, in 1979 and his MS from Youngnam University, Gyeongsan, Rep. of Korea, in 1981. He received his PhD from Osaka University, Osaka, Japan, in 1988. He now works at ETRI as a principal research engineer. His research interests include computer vision, pattern recognition, mobile visual search, and digital rights management.



**Dong-Seok Jeong** became a member (M) of IEEE in 1983 and a senior member (SM) in 2000. Prof. Dong-Seok Jeong received his BSEE from Seoul National University, Seoul, Rep. of Korea, in 1977 and MSEE and PhD from Virginia Tech, Blacksburg, VA, USA, in 1985 and 1988, respectively. He is also a member of SPIE and HKN. From 1977 to 1982, he was a researcher at the Agency for Defense Development of Korea, and he has been a professor at Inha University, Incheon, Rep. of Korea, since 1988. Additionally, he served as the president of the Institute of Information and Electronics Research from 2000 to 2004. His research interests include image and video processing and image and video signature and forensic watermarking.