Fast and effective features for recognizing recurring video clips in very large databases

Ina Dohring, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Dohring, Ina, and Rainer Lienhart. 2007. “Fast and effective features for recognizing recurring video clips in very large databases.” In *14th International Conference of Image Analysis and Processing - Workshops (ICIAPW 2007), 10 - 13 September 2007, Modena, Italy*, 65–70. Piscataway, NJ: IEEE. https://doi.org/10.1109/iciapw.2007.26.

Fast and Effective Features for Recognizing Recurring Video Clips in Very Large Databases

Ina Döhring and Rainer Lienhart
Lehrstuhl für Multimedia Computing
Universität Augsburg
{doehring,lienhart}@informatik.uni-augsburg.de

Abstract

Three different frame features (color patches, color coherence vectors, and gradient histograms) are investigated for their suitability to recognize recurring video clips in very large databases. They are evaluated in a real-time processing and real-time recognition system. Real-time recognition means that each clip must be recognized one second after its start. As the experimental results show, only gradient histograms work satisfactorily across different video material with the same video domain independent parameter set. For instance, they are – in contrast to color features – not negatively affected by dark frame sequences in video clips and the live video stream. By means of precomputation and subsequent table look-ups, gradient histograms can be implemented such that their computational costs come very close to that of color features.

1. Introduction

In the recent past several methods for recognizing reoccurring video clips (such as commercials) in video streams have been developed. Most schemes are based on deriving features from individual video frames, and then search for feature sequences in live-streams that match one of the stored feature sequences of the desired video clips.

For real-time recognition in live-streams color features such as color moments or color histograms are preferred due to their low computational complexity. On live-streams frame features must be computed in a fraction of 40ms (PAL) or 33 ms (NTSC), respectively. In practice, however, the various color features exhibit some serious deficiencies mainly concerning low intensity (i.e., brightness) sequences. Video clips with a large fraction of darker frames cannot be satisfyingly distinguished from each other mostly due to color quantization effects. Especially in the case of commercial recognition there is a need for domain-independent universally working features. Advertisements can be of any style. They may be designed like action/horror/romance movies, comics, newscasts, sportscasts or all other possible genres occurring on TV. They occur in all kinds of broadcast - from feature movies to MTV. This clearly requires highly robust features.

It has been shown that edge-based features may improve video clip recognition over color features [1, 2]. Normally, however, their computational complexity is at minimum a magnitude above that of color features. In this work we introduce an edge feature called *Gradient Histograms* for recognizing video clips which is almost as fast as the various investigated color features, but improves significantly the general precision and recall of video clips across all tested video genres.

Unlike almost all related work the term "'real-time"' has a double meaning in our work: (1) the overall processing time must be faster than 25 fps for PAL videos and (2) known video clips must be recognized with 1 second (i.e., within 25 frames for PAL). All performance numbers are reported of this case.

Related Work: In our work we study two different color features: color coherence vectors (CCVs) and color patches (CPs). CCVs have been introduced by Pass et al. [5] and applied to commercial detection by Lienhart et al. [3]. CPs have been described for instance in [1].

CCVs extend a color histogram by splitting it up into two histograms: One histogram recording the color distribution of so called homogeneous pixels and one histogram recording the color distribution of so-called inhomogeneous pixels. The sum of both histograms becomes a plain-vanilla color histogram. A pixel is regarded as coherent if it belongs to a larger region (larger than a certain threshold) of the same color, while a pixel is regarded as inhomogeneous otherwise.

CP features measure the coarse color distribution of the image. The whole image is divided into $N \times M$ subareas, to each of which the mean intensities are assigned. In difference to [1] we work in the RGB space with both color features.

2. Gradient Histograms

Definition: Different edge-based features have been investigated as sequence fingerprints in the past. Hampapur and Bolle, for instance, compared a gradient-based analog to color patches. In addition to the average gradient magnitude they also used higher order moments for each subarea [1]. In [2] they use amongst others a gradient direction histogram of the whole image.

In our work we deal with a combination of these two variants. Each image is described by a vector of gradient orientation histograms for each of the $N \times M$ subareas of an image. This gradient histograms were inspired by the SIFT feature introduced in [4]. For each image pixel its orientation and magnitude of the gradient is calculated. In the gradient direction histogram each sample point is weighted by the gradient magnitude.

Let

$$\begin{array}{rcl} I(\boldsymbol{x}) & - & \text{Grayscale intensity value, } I(\boldsymbol{x}) \in (0, 255)\,, \\ & & \text{and} \\ \nabla I\left(\boldsymbol{x}\right) & = & \left(\dfrac{\partial}{\partial x_1} I\left(\boldsymbol{x}\right), \dfrac{\partial}{\partial x_2} I\left(\boldsymbol{x}\right)\right) \\ & - & \text{the gradient intensity at point } \boldsymbol{x}. \end{array}$$

The magnitude of gradient $\nabla I\left(\boldsymbol{x}\right)$

$$M_g\left(\boldsymbol{x}\right) = \sqrt{\left(\frac{\partial I\left(\boldsymbol{x}\right)}{\partial x_1}\right)^2 + \left(\frac{\partial I\left(\boldsymbol{x}\right)}{\partial x_2}\right)^2} \qquad (1)$$

and orientation

$$\Theta_g\left(\boldsymbol{x}\right) = \arctan\left(\frac{\dfrac{\partial I\left(\boldsymbol{x}\right)}{\partial x_2}}{\dfrac{\partial I\left(\boldsymbol{x}\right)}{\partial x_2}}\right) \qquad (2)$$

are calculated by using pixel differences as gradient approximation

$$\begin{array}{rcll} \dfrac{\partial I\left(\boldsymbol{x}\right)}{\partial x_1} & \approx & I\left(x_1+1, x_2\right) - I\left(x_1-1, x_2\right), & (3) \\ \dfrac{\partial I\left(\boldsymbol{x}\right)}{\partial x_2} & \approx & I\left(x_1, x_2+1\right) - I\left(x_1, x_2-1\right). & (4) \end{array}$$

Thus, we use the following discrete representation for intensity gradient magnitude and orientation:

$$\begin{array}{rcll} m_g & = & \sqrt{\left(I\left(x_1+1, x_2\right) - I\left(x_1-1, x_2\right)\right)^2} & \\ & & \overline{+\left(I\left(x_1, x_2+1\right) - I\left(x_1, x_2-1\right)\right)^2}, & (5) \\ \theta_g & = & \arctan\left(\dfrac{I\left(x_1, x_2+1\right) - I\left(x_1, x_2-1\right)}{I\left(x_1+1, x_2\right) - I\left(x_1-1, x_2\right)}\right) & (6) \end{array}$$

For histogram evaluation we divide the whole image into $N \times M$ subareas I_{nm} with

$$I_{nm}\left(x_1, x_2\right) = I\left(x_1, x_2\right) \qquad (7)$$

with

$$\begin{array}{rcl} x_1 & = & X_n, \ldots, X_n + H_n - 1, \\ x_2 & = & Y_m, \ldots, Y_m + W_m - 1 \end{array}$$

$$\begin{array}{rcl} \left(X_n, Y_m\right) & - & \text{first sample point of image part } I_{nm}, \\ H_n & - & \text{height of } I_{nm}, \\ W_m & - & \text{width of } I_{nm}, \\ n & = & 1, \ldots, N, \\ m & = & 1, \ldots, M, \end{array}$$

over each of which we accumulate gradient magnitude values in K bins, covering the range of possible gradient orientation.

$$H_{nm}^k\left(I\left(x_1, x_2\right)\right) = \frac{1}{F} \sum_{x_1=X_n}^{X_n+H_n-1} \sum_{x_2=Y_m}^{Y_m+W_m-1} \mathcal{M}_g\left(x_1, x_2\right) \qquad (8)$$

with

$$\mathcal{M}_g = \begin{cases} m_g\left(x_1, x_2\right) & \text{if } \theta_k \leq \theta_g\left(x_1, x_2\right) < \theta_{k+1}, \\ 0 & \text{else,} \end{cases}$$

$$\begin{array}{rcl} k & = & 1, \ldots, K, \\ \theta_k & = & \left(k-1\right) 360^\circ / K, \end{array}$$

and normalisation factor

$$\begin{array}{rcll} F & = & \displaystyle\sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{x_1=X_n}^{X_n+H_n-1} \sum_{x_2=Y_m}^{Y_m+W_m-1} \mathcal{M}_g\left(x_1, x_2\right), & \\ & = & \displaystyle\sum_{x_1=1}^{H} \sum_{x_2=1}^{W} m_g\left(x_1, x_2\right). & (9) \end{array}$$

We measure the distance between two images I_1 and I_2 with the L_1-Norm

$$D_{GH}\left(I_1, I_2\right) = \frac{1}{NMK} * \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} \left|H_{nm}^k\left(I_1\right) - H_{nm}^k\left(I_2\right)\right|. \qquad (10)$$

The size of the gradient histogram fingerprint is $N \times M \times K$ values H_{nm}^k.

Reduction of fingerprint size: Due to Equations 8 and 9 we deal with normalized histograms, i.e., gradient histograms whose component values sum up to 1 (L1-norm). The component values are naturally represented by floating point numbers resulting in relatively large feature vectors. For real-time search in large databases smaller feature sizes are preferred. Therefore we map all floating point values to 1-byte integer values.

Empirical analysis shows that the fingerprint values H^k_{nm} are not uniformly distributed in $(0, 1)$. Especially values near 1 are very unlikely to be observed and can be lumped up into one value. To keep computational mapping cost low we apply a linear mapping from range $(0, L) \to (0, 255)$, $L \in (0, 1)$ with saturation at the higher bound. Thus, we use a scaling which maps only the lower values near 0, i. e. the interval $(0, L)$, to the destination interval; all values greater than L are assigned to the highest value 255. The concrete choice of L depends on the values of N, M, and K. With the use of 1-byte integer values we can reduce the fingerprint size to a quarter of its former size.

Reduction of computational costs: The computation of edge-based features is normally more than 10 times more time consuming than the computation of color-based features. According to Eqs. 5 and 6 we deal with the square root for the gradient magnitude and the really expensive arc tangent function for the estimation of the gradient orientations. To circumvent the repeatedly evaluation of such complex functions the use of look-up tables is a proper alternative. In our case we work on a discrete and limited range of values represented by all possible differences of two 1-byte unsigned integer values. That means both of our two arguments in Eqs. 5 and 6 are in the range $(-255, 255)$, and a 511×511 - table for each of the possible difference value pairs is sufficient for holding all correct values. Moreover, we include the estimation of the bin number of the orientation histogram in our look-up table. In result, we completely avoid the computation of the arc tangent - after building the look-up table - and estimation of the orientation histogram bin directly through a table look-up. By the this technique the computation of the gradient histogram is accelerate by more than a factor of 10 on current Dual-Core processor machines. We want to point out that size of the look-up table can be reduced by quantizing all possible difference values of two 1-byte integers to fewer bits.

3. Experiments

For our investigations we deal with two MPEG-2 test videos of different properties: one 4-hour NTSC video (720x480) from US television (431,540 frames) and one approximately 3-hour PAL video (720x576) from UK (281,896 frames). Both videos differ not only in their TV standard, but also in their content and visual fidelity. Because these differences impact the recognition results, we need to discuss the characteristics of our test videos in more detail: The US test video contains a mixture of news, sports, movies and commercials, whereas the UK test video is taken from SkySports TV. The main part of the video are sports news paused by commercial blocks. Visually the UK video is of inferior quality and of higher average intensity (brightness) compared to the US video. Commercials are mostly separated by hard cuts and one-frame dissolves in the UK video, whereas in the US video commercials are mostly separated by a couple of black frames. As we will see the various tested frame features exhibit different sensitivity to these individual differences of the two test videos. Figure 1 and Figure 2 show four sample frames from the UK and USA video.

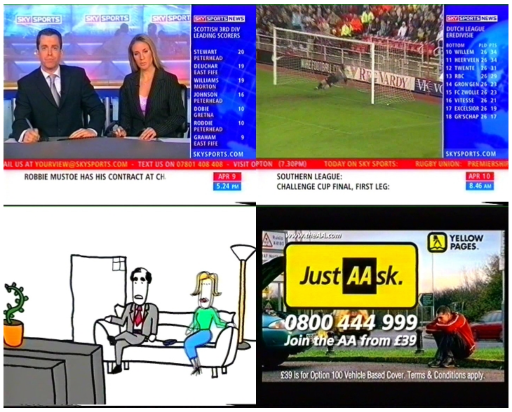
SKY SPORTS
ROBBIE MUSTOE HAS HIS CONTRACT AT CH
SOUTHERN LEAGUE:
CHALLENGE CUP FINAL, FIRST LEG:
YELLOW PAGES
Just AA sk.
0800 444 999
Join the AA from £39
£39 is for Option 100 Vehicle Based Cover. Terms & Conditions apply.

Figure 1. Sample frames from the UK video.

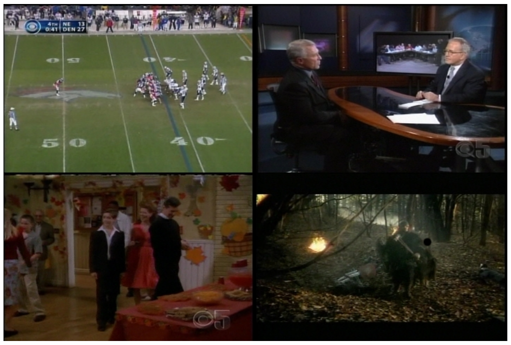

Figure 2. Sample frames from the US video.

The US video contains 152 manually detected commercials, intros/outros, and previews (in the following all test sequences are called commercials), 97 of them are different. We take one sample of each of the 97 commercials as query for building our database, even if there is no duplicate in the video, because one critical point for color-based features is precision. We do not want to get false alarms. So, we test against all entries in our database. The query set of the UK video consists of 65 distinctive test sequences; including all duplicates we found 104 commercials.

We use the approximate substring matching algorithm described in Lienhart et al. [3] for finding similar video sequences: We construct a fingerprint of each commercial by calculating important features per frame and then represent a commercial's fingerprint as a sequence of these features. We call the representation of the value of a feature a character, the domain of possible values an alphabet, and the sequence of characters a string. Approximate substring matching then solves the following problem: Given a query string A of length P and a (much) longer subject string B of length N, the approximate substring matching finds the substring of B that aligns with A with minimal substitutions, deletions and insertions of characters [6]. A cost of 1 is assigned to deletions and insertions, while the cost of substitutions is based on the distance between both feature vectors under comparison. We need to design a feature distance function there similar frames have a distance less than 1, and arbitrarily selected frames should have a distance value greater than 1. All distance values above 1 are saturated to 1. To meet these requirements we use a scaled L_1-Norm as our feature distance function, because Eqn. 10 gives much too small values (always less equal 1). The minimal number of substitutions, deletions and insertions transforming A into the best matching substring of B is called the minimal distance D between A and B. Two fingerprint sequences A and B are regarded as identical if the minimal distance D normalized by the query length P between query string A and subject string B does not exceed the threshold $t_{stringDist}$.

For the computation of the three different frame features the parameters are listed in the following: We downscale video frames to half the resolution 360 × 240 for NTSC, and 360 × 288 for PAL, respectively, in order to reduce the computing time.

Color Coherence Vector (CCV): We take the two most significant bits of each RGB color component for creating the histograms of the coherent and incoherent pixels. A pixel is regarded as coherent, if the region of the color it belongs to is greater than 1% of the whole image. Fingerprint size is 2×2^{3B} ($B = 2$ - number of significant bits). For measuring the distance between two color coherence vectors $V(I)$ we use the L_1-Norm normalized by its components

$$D_{CCV}(I_1, I_2) = \frac{1}{2^{3B+1}} \sum_{n=1}^{2^{3B+1}} \frac{|V_n(I_1) - V_n(I_2)|}{V_n(I_1) + V_n(I_2) + 1}. \tag{11}$$

The distance from Eq. 11 is multiplied by 10, so that the scaled function approximately meets the requirements by the approximative substring algorithm, that non-matching frames have a distance greater than 1.

Color Patches (CP): The CP features operates on $N \times M$ subareas as the GH features do too. It is formed by the averaged RGB color intensities C_{nm} ($C \in (R, G, B)$ on these subareas. The notation concerning these subareas follows Equation 7. The whole CP feature $C(I)$ has a size of $3 \times N \times M$, and we measure the distance between two vectors with the L_1-Norm

$$D_{CP}(I_1, I_2) = \frac{1}{3NM} * \sum_{C \in (R,G,B)} \sum_{n=1}^{N} \sum_{m=1}^{M} |C_{nm}(I_1) - C_{nm}(I_2)|. \tag{12}$$

For our experiments we use $N = M = 8$ and divide the distance from Eq. 12 by 80.

Gradient Histograms (GH): For the GH features we take the same spatial resolution, i. e. $N = M = 8$. We set the number of bins of the GH to 8. Experiments yield for $N = M = K = 8$ to a value $L = 0.2$ for mapping to 1-byte integer values. So, we only differentiate H_{nm}^{k} values in the range $(0, 0.2)$, all values greater than 0.2 are considered to be equal. We need to modify our distance function to meet the requirements of the approximative substring algorithm and divide values from Eq. 10 by 12, which gives us the best results.

4. Results

Quality: Based on the manually labeled ground truth of our test videos we compute the performance measures recall R and precision P:

$$R = \frac{\text{positive matches}}{\text{all relevant sequences}}, \tag{13}$$

$$P = \frac{\text{positive matches}}{\text{all found sequences}}. \tag{14}$$

We define a found sequence as a positive match, if the estimated end position does not differ more than 5 frames from the actual end position. Although we have tested with every single replication of each commercial for statistical evaluation we limit the test sequence ensemble to only one representative for each commercial. We only take into account the worst case for each one, i. e. only the test sequence which gives the lowest recall factor among all representatives of this commercial.

Fig. 3 and 4 show the recall-to-precision graph for the US and the UK video, respectively. We can recognize strong

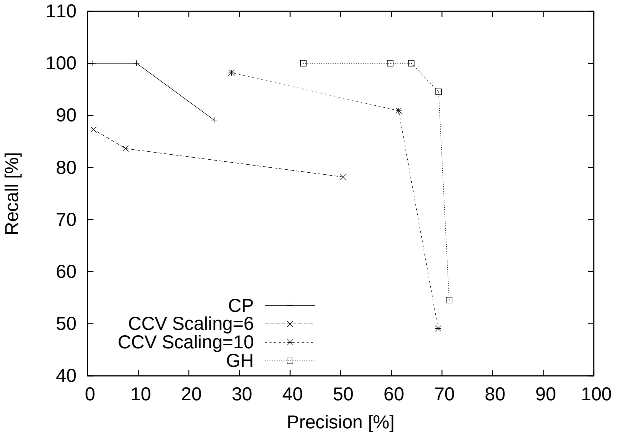

Figure 3. Recall-precision graph for different fingerprints of 97 US commercials.

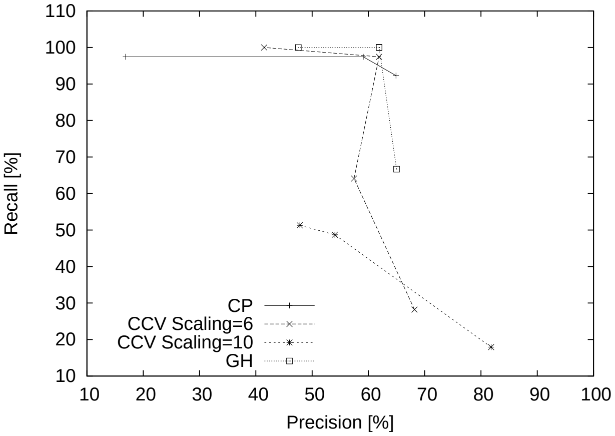

Figure 4. Recall-precision graph for different fingerprints of 65 UK commercials.

differences between all three types of fingerprints, especially for the US video.

The color patches feature consistently gives good recall values, but comes with very low precision for the US video. Most problems are caused by relatively dark commercials of low intensity, mostly due to the letterbox format adopted wide screen film sequences. In our US test video for 10 out of our the 97 analyzed commercials the Color Patches fingerprint produces false alarms due to the above-named reasons. Additionally, in the US video commercials are often separated by black frames which could be accidentally misrecognized as parts of very dark sequences. For the UK video the Color Patches fingerprint works almost flawlessly. Thus, CP fingerprints allow retrieval of all commercials, but with a high risk of false matches for very dark sequences such they might show up more frequently in videos of more diverse content than the UK test video.

With the color coherence vector we are not able to catch all occurring commercials in the US video, but with moderate recall values we reach acceptable precision. Nevertheless, the US video provides some difficulties for color based features. However, with the same scaling of the distance function (Eq. 11) we get unsatisfactory results for the UK video. A scaling factor of 10 reduces the recall up to 50%. If we use a factor of 6 we get comparable results with the color patches and the gradient histogram fingerprints. But different scaling factor for different video content are an indication for instability of this feature. Thus, CCV fingerprints are not appropriate for domain-free video clip recognition.

The gradient histogram algorithm gives excellent recall values at decent precision values across both test videos. False matches are limited to matches of similar spots with identical scenes, but of significantly different duration. For instance, one commercial is a shorter version of the other. Due to the real-time recognition requirement which demands to recognize a commercial within one second (i.e., the first 25 and 30 features of each PAL and NTSC commercial, respectively), such partly identical commercials cannot be distinguished. They can only be separated by search for the whole fingerprint against the video, not just the fingerprint of the first second. However this was not done in this evaluation.

Note that precision is influenced by the fact that we also search for sequences which have no duplicates in the test video. These queries may also contribute to the false positive, but not to the true positive matches, resulting in a lowered precision.

Overall the gradient histogram fingerprint was the most stable method, not only resulting in comparable results for both of our test videos, but also regarding the choice of the threshold parameter. We reach identical results for the UK video for thresholds in the range from $0.4 \ldots 0.8$.

Figures 5 and 6 show the recall and precision for different threshold values $t_{stringDist}$ for the US and the UK video. The relative stability of the gradient histogram feature can be clearly seen.

Computational Costs: Table 1 shows computational resources needed by the different fingerprint methods. The average feature computation time and its vector size per frame for the UK video are given on an AMD Athlon X2 4400+. The color coherence vector has fewer components than the gradient histogram vector, however required 4-byte integer values instead of just 1-byte values. The color patches fingerprint is the fastest method, followed by the

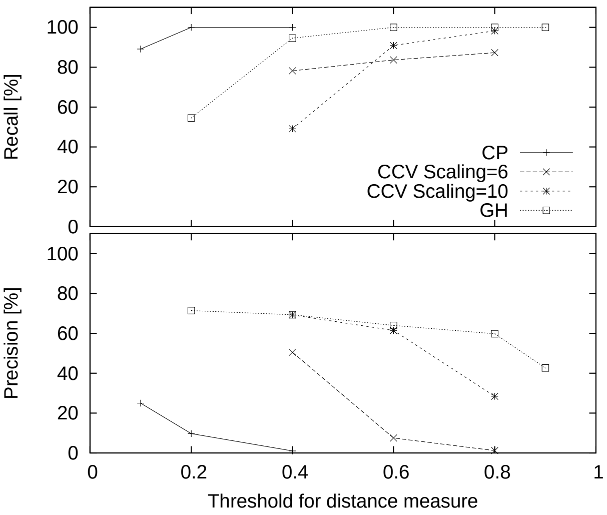

Figure 5. Recall vs. precision in dependence on threshold $t_{stringDist}$ for the US video.

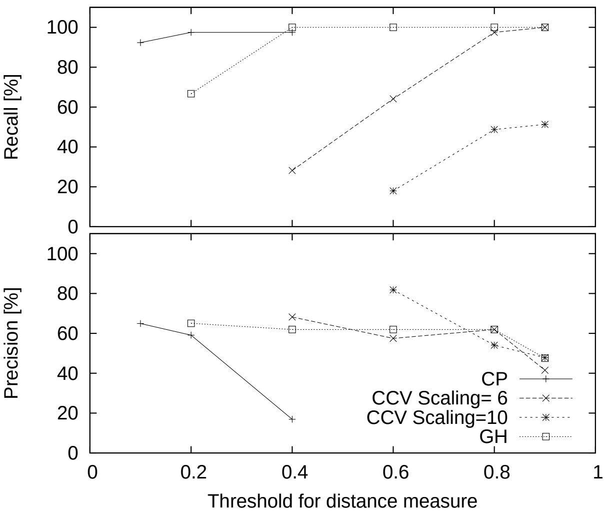

Figure 6. Recall vs. precision in dependence on the threshold $t_{stringDist}$ for the UK video.

color coherence vector, but the computation of our edge based feature is only 3-4 times slower than that of the color based feature and it is usable for real time applications.

Feature	Time [ms]	Feature size [bytes]
Color Patches (CP)	3.0	196
Color Coherence Vector (CCV)	3.9	512
Gradient Histogram (GH)	11.4	512
Gradient Histogram (slow)	23.3	512

Table 1. Avg. feature computation time and its vector size (per frame) for the UK video. GH (slow) gives the time for the GH code without the use of a look-up table.

5. Conclusion

In this work we compared three different frame-features for real-time video clip recognition: color patches, color coherence vectors, and gradient histograms. The edge-based gradient histograms outperformed CCVs and CPs, and worked with the same parameters with both of our two different test videos. The performance of color-based features, however, was dependent on the type and style of video. Gradient histograms combine the spatial information provided by the image partition into subareas with the information about the type of gradients represented by the orientation histograms and are thus an improvement over the two gradient-based features used in [1] and [2]. Gradient histograms provide detailed information about the image structure. Therefore, we could not verify the doubts mentioned by Hampapur et al. [1] about mismatching frames with plain background and limited text.

Acknowledgments

This project was supported by Half Minute Media Ltd.

References

[1] A. Hampapur and R. Bolle. Feature based indexing for media tracking. In *Proceedings of International Conference on Multimedia and Expo*, New York, 2000.

[2] A. Hampapur and R. M. Bolle. Comparision of distance measures for video copy detection. Computer Science RC 22056, IBM Research, 2001.

[3] R. Lienhart, C. Kuhmünch, and W. Effelsberg. On the detetction and recognition of television commercials. Reihe Informatik 16, Universität Mannheim, 1996.

[4] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[5] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Conference on Multimedia*, pages 65–74, Boston, Massachusetts, 1996.

[6] G. A. Stephen. *String Searching Algorithms*, volume 3 of *Lectures Notes Series on Computing*. World Scientific Publishing, first edition, 1994.