

# Compressed Domain Features Extraction for Shot Characterization

Costantino Grana, Roberto Vezzani, Daniele Borghesani, Rita Cucchiara

Department of Information Engineering  
University of Modena and Reggio Emilia, Italy  
{name.surname}@unimore.it

**Abstract.** In this work, we propose a system for shot comparison directly working on the MPEG-1 stream in the compressed domain, extracting both color, texture and motion features considering all frames with a reasonable computational cost, and results comparable to those obtained on uncompressed keyframes. In particular a summary descriptor for each Group Of Pictures (GOP) is computed and employed for shot characterization and comparison. The Mallows distance allows to match different length clips in a unified framework.

**Keywords:** MPEG-1, Shot Characterization, Compressed Domain Feature Extraction, GOP Level Features.

## 1 Introduction

The increasing spread of Video Digital Libraries calls for the design of efficient Video Data Management Systems to manage video access, provide summarization, similarity search, and support queries according with available annotations. General internet users are very demanding in search, so the media search technologies for the mass have to be very simple, intuitive, and easy to use, as text search is [1].

Examples of automatic semantic annotation systems have been presented recently, most of them in the application domain of news and sports video. Most of the proposals deal with a specific context making use of ad-hoc features. In [2] the playfield area, the number and the placement of players on the play field, and motion cues are used to distinguish soccer highlights into subclasses. Differently, a first approach trying to apply general features is described by [3]. Employing color, texture, motion, and shape visual queries by sketches are provided, supporting automatic object based indexing and spatiotemporal queries.

Different systems have been developed to compare shots, and many of these simply extend research results obtained on image retrieval to the analysis of a representative key frame. This is made mainly for computational reasons which make



**Fig. 1.** a) Sample frame with superimposed motion vectors. b) DC Image. c) AC Image. d) Motion Image.

impossible to analyze all frames. A problem with this approach is then how to include motion cues in this analysis.

In this paper we propose a system to generalize this approach by directly working on the MPEG-1 stream in the compressed domain, extracting both color, texture and motion features considering all frames with a reasonable computational cost, and results comparable to those obtained on uncompressed keyframes. In particular a summary descriptor for each Group Of Pictures (GOP) is computed and employed for shot characterization and comparison. The Mallows distance is used to allow different length clips to be compared in a unified framework.

## 2 Similarity of Video Clips

The problem of clip similarity can be seen as a generalization of the problem of image similarity: as for images, each clip may be described by a set of visual features, such as color, shape, texture and motion. These are grouped in a feature vector:

$$\mathbf{V}_i = [F_i^1, F_i^2, \dots, F_i^N] \quad (1)$$

where  $i$  is the frame number,  $N$  is number of features and  $F_i^j$  is the  $j$ -th feature computed at frame  $i$ . However, extracting a feature vector at each frame can lead to some problems during the similarity computation between clips, since they may have different lengths, and, more important, this could lead to an excessive computational load; at the same time keeping a single feature vector for the whole clip cannot be representative enough, because it does not take into account the features' temporal variability. A simple and common solution can be represented by the use of a fixed number  $M$  of feature vectors for each clip, computed on  $M$  frames sampled at uniform intervals within the clip. In previous experiments [4], as a tradeoff between efficacy and computational load we used  $M = 5$  for clips of averaging 100 frames. A major

problem with this approach is that it is not possible to include reliable motion information without considering the continuous variability of the clip.

Here, we propose an intermediate approach which provides a compromise solution to both the problems of computational load and complete motion description, working directly on the compressed MPEG video stream. Indeed the compressed MPEG video stream provides, without the complete decompression process, information about color (DC Image), texture (AC Images) and motion (Motion Vectors). Of course these features are provided at macroblock level (a 16 by 16 pixels square), but are sufficient for feature extraction in the context of video clip similarity search. At the same time, these values are already a summary of the 256 pixels of the macroblock, leading to a twofold advantage: an obvious speedup given by the reduced number of values, and an already summarized view of the blocks information.

Since the MPEG stream structure is composed by a hierarchy of layers, and pictures can be of different types, we selected to cope with the absence of motion information in I type pictures or with directly available color information in P and B pictures, by collecting features at Group of Pictures layer (GOP). The only remaining obstacle is the fact that a different number of GOPs can be found in every shot, since they are of different lengths. To this aim, we use the Mallows distance which allows the comparison of discrete distributions of different lengths.

### **3 GOP feature extraction and comparison**

A GOP is composed by an Intra coded frame (I frame) and a variable number of Predicted frames (P and B frames). I frames have a structure which resembles the JPEG standard, thus color and texture data are readily available for decoding. P and B frames conversely require one or more reference to be decoded, and contain motion vectors, which allow for the reconstruction of a motion compensated image.

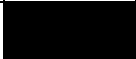




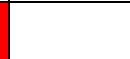
Since we are looking for a summary feature for the whole GOP, while keeping down the computational load, we chose to extract color and texture information from I frames only, and to employ P and B frames, just to provide a characterization of the amount of motion.

#### **3.1 Color description.**

Images are coded in I frames, by subtracting 128 from each color channel, by transforming with the DCT every 8x8 block, then by quantizing the 64 transformed coefficients and finally by Huffman encoding them in Zig-Zag order. Decoding would require the reverse process with the inverse DCT being the bottle neck.

By analyzing the DCT equation it is possible to see that the first coefficient of the transform is simply eight times the average color of the 64 pixels in the block. The quantization value for this coefficient is fixed to 8 by the standard, so we can obtain a so called DC image, that is a scaled down version of the frame, by simply decoding the Huffman codes and collecting the first coefficient for each block. An example of an I frame and the corresponding DC image is shown in Fig. 1b.

**Table 1.** Quantization levels and color hints for the motion amount.

$\mu$ range	0-2	2-5	5-10	10-15	15-20	>20
Level	0	1	2	3	4	5
Color						

### 3.2 Texture description.

Another interesting property of the DCT transform is that the first coefficient after the DC one, both in the horizontal and vertical direction describes the amount of variation in that direction, which is an information close to the spatial gradient. Thus, collecting these two coefficients for each block we can generate the two so called AC images, which can be considered representatives for  $x$  and  $y$  components of a gradient vector. We chose to store the module of the vector in a unique AC image, as in the example of Fig. 1c.

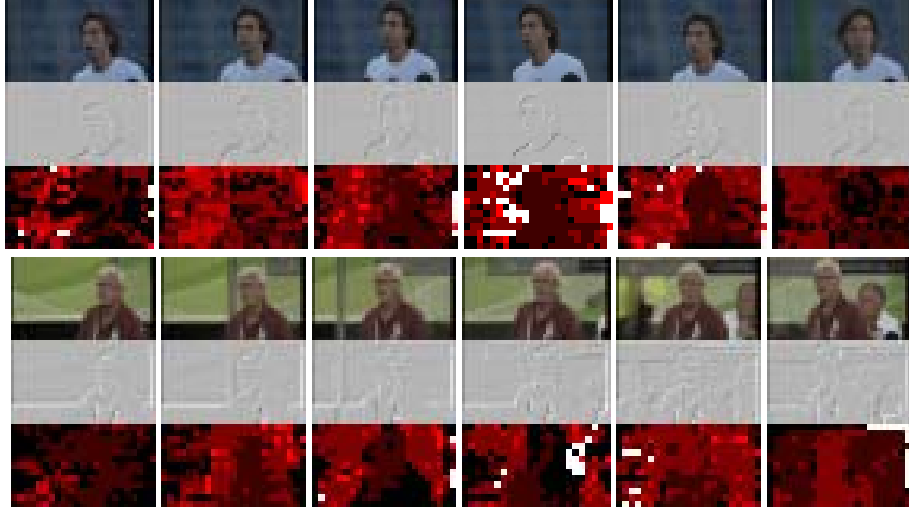
It is worth to highlight that extracting the DC image for the color characterization and the AC image for the texture description is much faster than decompressing the entire I Frame, since we avoid the iDCT operation.

### 3.3 Motion description.

As above mentioned, we extract motion information from the motion vectors coded inside the P and B frames. Both frame types have a prediction with respect to a previous reference, while only B frames have a prediction vector with respect to a reference in the future, so to simplify and make more uniform the process we consider only the *forward motion vectors*, that is those referring to the past. For similarity purposes, we chose to only characterize the amount of motion in the GOP, instead of using a more detailed description, like an affine motion estimation, or a direction histogram. This is motivated by the fact that motion vectors are not always so significant, but even in those cases they still give a rough idea of a moving/non-moving scene.

To produce a summary of the GOP amount of motion, we chose to quantize the modulus of the motion vectors ( $\mu$ ), then for each macroblock we collect the amount of motion distribution (histogram) during the GOP. The most frequently observed level (the mode of the distribution) is selected to build the motion image (Fig. 1d). The quantization scale has been empirically selected and is reported in Table 1.

This process allows to produce a motion image, which does not require to reorder the frames in the GOP, which are stored in a different order with respect to the visualization, providing another tool for faster computation.



**Fig. 2.** DC Images, AC Images, and Motion Images for the GOPs of two shots. The camera is panning to follow the player/coach; this produces a lower amount of motion corresponding to the foreground region. White blocks are in correspondence with large motion vectors which are likely not significant for motion descriptions.

### 3.4 Distance between GOPs.

The distance between two GOPs is computed by linearly combining the distances between the DC images, the AC images and the motion images. For AC and motion images, we choose to keep the spatial information, so these images are compared by a point by point difference and the sum of the absolute values is collected as a measure of dissimilarity. In the case of the DC image, we only consider the color distribution and choose to use a three dimensional color histogram in the YCbCr color space, which is the native color space of the MPEG standard, and does not require any color space conversion. The YCbCr color space is quantized as in [5] and the distance is computed as the histogram intersection.

## 4 Mallows Distance

To describe a shot, we want to take into account the number of GOP features obtained as described previously. Each shot is thus characterized by a discrete distribution of features:

$$\beta_i = \{(V_i^1, P_i^1), \dots, (V_i^N, P_i^N)\} \quad (2)$$

where  $V_i^k$  is the vector of features extracted for shot  $i$  at GOP  $k$ , and  $P_i^k$  is the associated probability, which in our case we fixed at  $1/N$ , that is all GOPs are equally weighted.

To compute the distance  $D(\beta_1, \beta_2)$  between two distributions  $\beta_1, \beta_2$ , we use the Mallows distance [6,7] introduced in 1972. Consider two probability distributions  $P$  and  $Q$  on  $\mathbb{R}^n$ . Define

$$M = \left\{ \text{probability distribution } \mu(x, y) \text{ on } \mathbb{R}^n \times \mathbb{R}^n \mid \int_y d\mu(x, y) = P(x), \int_x d\mu(x, y) = Q(y) \right\} \quad (3)$$

Mallows proposed to measure the difference between two probability distributions as:

$$Mallows_p(P, Q) = \min_{\mu} \left( E_{\mu} \|x - y\|_p^p \right)^{1/p} \quad (4)$$

subject to the constraints of Eq. 3, where the  $\|\cdot\|_p$  denotes the  $L_p$  norm, and  $1 \leq p \leq +\infty$ . For two discrete distributions  $P = \{(x_1, p_1), \dots, (x_n, p_n)\}$  and  $Q = \{(y_1, q_1), \dots, (y_m, q_m)\}$ , with  $\sum p_i = 1$  and  $\sum q_i = 1$ , minimizing the cost functional reduces to

$$\min_{\mu} \sum_{i=1}^n \sum_{j=1}^m \mu(i, j) C(x_i, y_j) \quad (5)$$

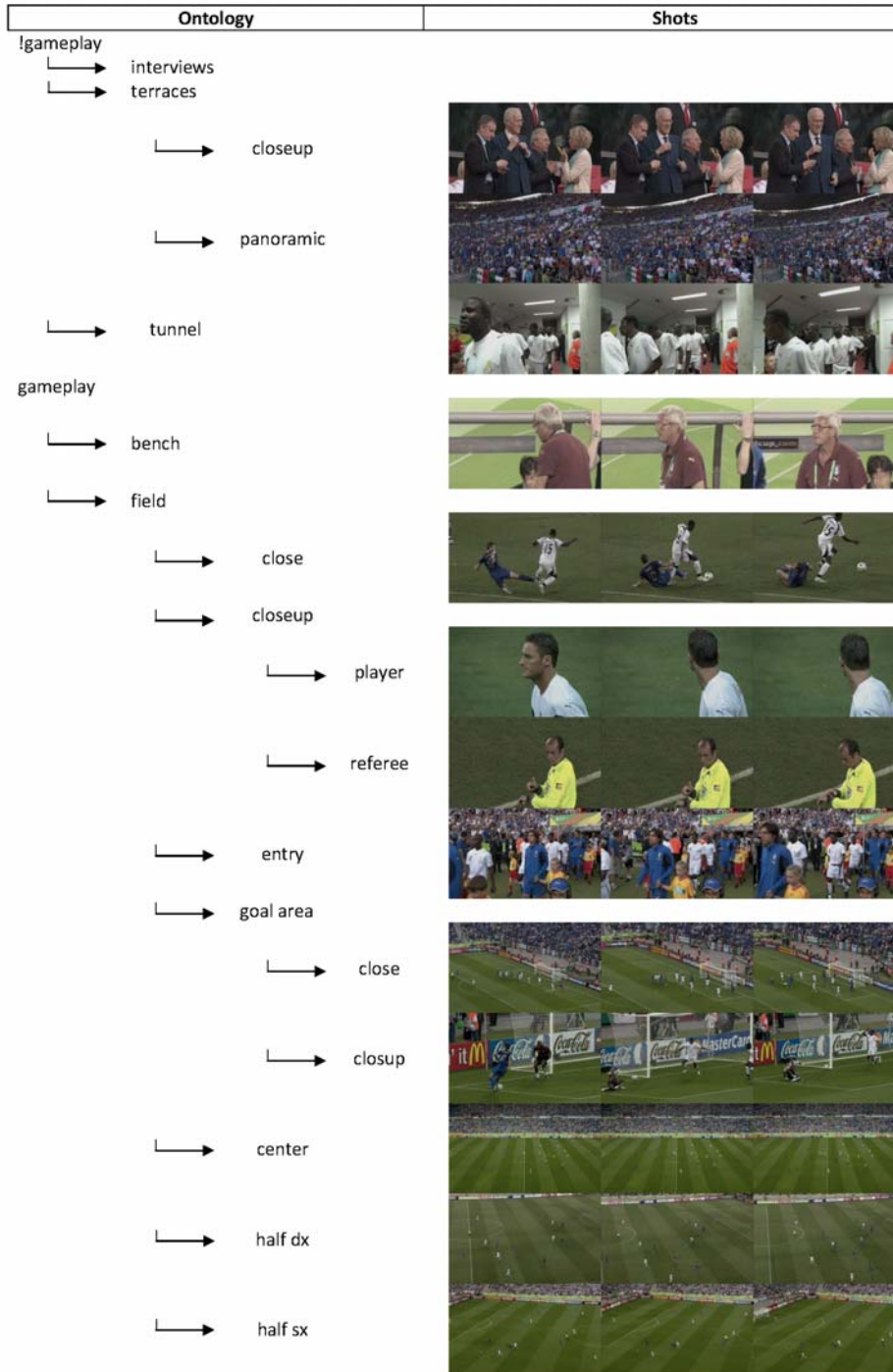
subject to

$$\begin{aligned} \mu(i, j) &\geq 0; \sum_{j=1}^m \mu(i, j) = p_i \\ \sum_{i=1}^n \mu(i, j) &= q_j; 1 \leq i \leq n; 1 \leq j \leq m \end{aligned} \quad (6)$$

where  $C(x_i, y_j)$  is the distance matrix between the elements of the distribution. Note that there is no constraint on  $C$ , i.e. we can use any formulation we have to compute the base distance between frame features. The dual of the linear programming problem of Eq. 5 is to find  $u = [u_1, \dots, u_n]$  and  $v = [v_1, \dots, v_m]$  in order to solve the problem:

$$\max \sum_{i=1}^n p_i u_i + \sum_{j=1}^m q_j v_j \quad (7)$$

subject to  $u_i + v_j \leq C(x_i, y_j), 1 \leq i \leq n; 1 \leq j \leq m$ . By solving the dual problem, we achieve better computational efficiency. In our implementation, we used the simplex algorithm to solve the problem. Note that the above formulation gives a set of  $n \times m$  constraints (the size of the distance matrix), which may prove difficult to solve, since the required simplex tableau will be  $(n \times m) \times (1 + n + m + n \times m)$ .



**Fig. 3.** Hierarchical Soccer Domain Ontology, used in the experiments.

By employing the Mallows distance we can compare two shots, by comparing the associated discrete distributions of features. Since the two distribution may be of different lengths, no constraint is posed on the number of GOPs we consider for the shot. Moreover, the distance measure between two GOPs, which has been previously defined, may be computed with whatever other distance we may think of, and the only requirement is to provide a distance matrix between all GOPs of the distributions, leaving space for future improvements.

## 5 Automatic annotation

We suppose to have a domain-specific video digital library, in which all the videos are referred to a specific context. Each video has be split into clips (or shot) using an automatic or manual shot segmentation process. We assume that it is possible to partition these clips into a set of  $L$  classes  $\mathbf{C} = (C_1, \dots, C_L)$ , which are characterized by different contents or camera views. Given a large set of training clips, we implemented an interactive user-friendly interface to quickly assign each clip of the digital library to a specific class  $C_k$  and then employ it for automatic annotation purposes. A new clip can be compared with the training clip set and classified using a nearest neighbor approach and the similarity measure above defined.

The integrated framework we developed allows opening videos, provides fast browsing capabilities, allows moving with single frame steps and contains all the described modules. In particular, shot detection [8] may be performed also in batch mode, in order to apply the system to a large set of videos (we used it on TRECVID 2005 and 2006 dataset). The system has a classification scheme manager which allows to define a taxonomy, its classes and to manually associate a shot to a defined concept, describing its contents.

The system was tested on a DVD source, a documentary video about the Italian victory of last world cup, so it contains different kinds of generic soccer scenes (classical game framing like short or long views, half body views, faces, little or large groups of fans, interviews with players or coaches, graphical animations). The video is stored in PAL format, so it is a 720x576 interlaced video. Prior to elaboration, it was deinterlaced by discarding field 2, horizontally reduced by a factor of 2, and cropped by 40 pixels from top and bottom to remove the useless black stripes.

A specific soccer domain ontology has been created in order to support the visual querying. The ontology is hierarchically structured in four levels, which encompass different semantic levels of detail. The first level is a binary categorization in gameplay vs. non-gameplay scenes. Non gameplay actions are divided by the zone captured in the video and the third level provides distinction between the zoom level. Similarly the gameplay scenes are divided in field and bench shots, further detailed based on zoom or subject. An overview of the ontology is provided in Fig. 3, together with some example shots (shots are summarized by their first, center and last frames). More specific details were omitted, since our aim is focused on general purpose systems, avoiding the use of domain specific feature.

For testing purposes, we compared the results with our previous system which was described in [4], where color and motion features were extracted in the uncompressed



domain. The results are comparable and the differences among the different classes were not significant, but the computation times are about 2 orders of magnitude lower. The compressed domain feature extractor works at about 550 fps, that is a processing speed of 22 times real time.

## 6 Conclusions

We presented a novel approach for fast shot characterization, which is based on the direct analysis of the MPEG-1 stream. Color, texture and motion information are extracted and summarized at GOP level without uncompressing the video. This allows to process about 550 frames per second.

This preliminary work has been tested on MPEG-1 streams only, using a custom made software library to fully take advantage of the MPEG structure. The library has been developed in C++ language and it is fully portable. Of course, the library can be extended also to MPEG-2 streams, which fundamentally share the same compression techniques and frame structure.

A much more difficult problem will be the adoption of the MPEG-4/AVC standard [9]. Differently from the MPEG-1/2 standards, the block size can vary between macroblocks; in order to compare different frames, we have to adapt to the smallest size possible which is 4x4 and to split larger blocks. Another problem is that motion vectors of the same frame can have different references (up to 16 frames away). Thus, a normalization step will be probably required in order to provide a measure of the frame level motion. Finally, we have to cope with the presence of I-blocks inside B-frames which was forbidden in previous versions of the standard.

**Acknowledgments.** This work is supported by the DELOS NoE on Digital Libraries, as part of the IST Program of the European Commission (Contract G038-507618).

## 7 References

- [1] Jaimes, A., Christel, M., Gilles, S., Sarukkai, R., Ma, W.: Multimedia information retrieval: what is it, and why isn't anyone using it? In: 7th ACM SIGMM international Workshop on Multimedia information Retrieval, pp. 3--8. ACM, New York (2005)
- [2] Bertini, M., Cucchiara, R., Del Bimbo, A., Torniai, C.: Video Annotation with Pictorially Enriched Ontologies. In: IEEE International Conference on Multimedia and Expo, pp. 1428--1431. IEEE Press, New York (2005)
- [3] Chang, S., Chen, W., Meng, H.J., Sundaram, H., Zhong, D.: A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries. *IEEE Trans. Circ. Sys. Video Tech.* 8(5), 602--615 (1998)
- [4] Vezzani, R., Grana, C., Bulgarelli, D., Cucchiara, R.: A semi-automatic video annotation tool with MPEG-7 content collections. In: IEEE International Symposium on Multimedia, pp. 742--745. IEEE Press, New York (2006)

- [5] Sang-Kyun, K., Doo Sun, H., Ji-Yeun, K., Yang-Seock, S.: An Effective News Anchorperson Shot Detection Method Based on Adaptive Audio/Visual Model Generation. In: Leow, W.-K.; Lew, M.S.; Chua, T.-S.; Ma, W.-Y.; Chaisorn, L.; Bakker, E.M. (Eds.) Image and Video Retrieval. LNCS, vol. 3568, pp. 276--285. Springer, Heidelberg (2005)
- [6] Mallows, C.L.: A note on asymptotic joint normality. *Annals of Mathematical Statistics*. 43(2), 508--515 (1972)
- [7] Zhou, D., Li, J., Zha, H.: A New Mallows Distance Based Metric For Comparing Clusterings. In: 22nd International Conference on Machine Learning (2005)
- [8] Grana, C., Cucchiara, R.: Linear Transition Detection as a Unified Shot Detection Approach. *IEEE Transactions on Circuits and Systems for Video Technology*. 17(4), 483—489 (2007)
- [9] ISO/IEC 14496-10:2005 – Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding (2005)