On the usefulness of object shape coding with MPEG-4

Andrea Prati
Dipartimento di Scienze e Metodi dell'Ingegneria
Università di Modena e Reggio Emilia
Via Allegri, 13 - 42100 Reggio Emilia - Italy
prati.andrea@unimore.it

Rita Cucchiara
Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia
Via Vignolese, 905 - 41100 Modena - Italy
cucchiara.rita@unimore.it

Abstract

This paper reports the results of an in-depth analysis of the degree of usefulness of object shape coding in video compression. In particular, MPEG-4 is used as reference standard. The influence of different coding parameters on the performance is deeply examined and discussions on the results are provided. Object shape coding is compared with classical (MPEG-2) frame-based coding both at an objective level (by comparing PSNR/quality and bitrate/filesize) and at a subjective level (asking to a set of users to express their opinion on overall quality, cognitive effectiveness, and willingness to pay). In conclusion, this paper aims at answering to the question whether it is convenient to use object shape coding instead of frame-based coding or not.

1. Introduction

The demand for advanced digital services has driven, in the last decade, a wide diffusion of many different types of multimedia contents: from albums of MP3-coded audio streams, to videos on demand, to the Voice Over IP, to the Web-TV, and so on. Some more specific emerging contents, such as videos provided by GIS (Geographical Information Systems), assume great importance due to the large amount of data embedded in them and to the innovative ways that are required to effectively manage them [5].

These contents pose several problems, mainly due to the large amount of data they contain. Making access to these contents in a distributed and portable way is now a "must" to provide innovative services, tools and devices, and to be desirable for the customers. The distributed nature of these contents requires innovative communication protocols that fit the dynamics of the multimedia data to the available network channels. The requirement of portability means wireless communication, low power consumption, and portable devices (such as PDA, HHC, or cell phones), that have also limited capabilities in terms of computational power, available software tools, display size, and so on. Further, these data must be often recorded for storage or future accesses. As a consequence, efficient digital recording has become a relevant topic in multimedia.

Under these pushes, scientific communities and multimedia-related companies have made a tremendous effort to develop innovative solutions for compressing data in order to limit the amount of data required and, thus, reduce the required storage and/or bandwidth. In particular, videos are the most demanding multimedia items in terms of required bandwidth and variability. Reducing the total size of the video and, at the same time, preserving as much quality as possible, enable to relax the above-mentioned problems, keeping user's satisfaction at a reasonable level.

MPEG is, still now, the reference standard for video compression. Earlier MPEG standards, such as MPEG-1 and MPEG-2, have made digital videos available in consumer products. MPEG-1 (completed in 1992) enabled coding of non-interlaced videos at low resolution and bit rates offering VHS-like video quality, while MPEG-2 (completed in 1994) addressed also interlaced videos at a significantly higher resolution and bit rates, allowing digital TV/HDTV quality [13]. MPEG-4 standard was proposed to change the way in which videos are coded, offering new tools and services. MPEG-4 core was delivered in 1999, but many parts of the standard are continuously improved to enhance coding accuracy and to add new capabilities. The MPEG-4 video standard firstly proposed coding of, and thus access to, individual objects, scalability of coded objects, transmission of coded video objects on error prone networks, as well as efficient coding of video objects. Effects of using object-based video coding on the improvement of bitrate and PSNR have been discussed in [4, 11, 17]. However, in these cases object coding does not take alpha planes (i.e., arbitrary shapes) into account. Further, MPEG-4 video also allows higher efficiency coding of rectangular video without the necessity of dividing a scene into video objects. The improvement in coding efficiency w.r.t. previous standards is limited, but modifications to the decoding and encoding

processes are provided within the same coding structure of earlier standards, and, thus, with a limited increase in complexity.

For the ever-increasing demand of higher compression rates, new standards come to life. The most representative is H.264/MPEG-4 AVC [13]. This new standard has been developed as a joint work by ISO/IEC MPEG and ITU-T VCEG to bring together the expertise acquired in MPEG and in H26L standards. The achieved gain in compression efficiency is not easily valuable, since it heavily depends on the parameters (i.e., profile and level) of the AVC codec, that, in their turn, depend on the final application. The novelties introduced w.r.t. MPEG-4 are numerous, but the most significant ones are [13]: the use of I-, P- and B- slices together with pictures; use of approximated integer 4x4 DCT transform; multi-reference prediction (allowing P- and B-pictures/slices to use more than one reference); intra-prediction in the pixel domain; fixed scan of DCT coefficients; in-loop filter to reduce block artifacts; slice and macroblock re-ordering; use of context adaptive variable length coding (CAVLC) and context adaptive binary arithmetic coding (CABAC); non linear DC quantization. Unfortunately, the increase in compression efficiency is obtained at the cost of an increase in the complexity, often referred as a factor of 4 for the decoder and a factor of 9 for the encoder (w.r.t. MPEG-2). However, regardless of the single differences, all these standards belong to the class of hybrid approaches based on the dualism of DCT transform and motion compensation.

Among the different features of video codecs, object coding with arbitrary shapes represents a cutting-edge functionality in order to meet both the bandwidth constraints and the demand of advanced services. More specifically, the usefulness of *object shape coding* is twofold: in fact, it allows both to further reduce the required bitrate (thanks to the selective compression of the scene, saving bits by compressing more the background or non-interesting objects) and to provide SNR scalability. As reported above, MPEG-4 is the unique standard that implements this feature. Current version of H.264/MPEG-4 AVC does support neither object-based nor layered scalable coding. A new modification of the standard, called SVC (Scalable Video Coding) [15], is meant to implement these capabilities, but is still a work-in-progress and its efficiency has to be demonstrated. It is worth noting that MPEG-4 does provide only basic tools for shape and object coding, but does not propose a method for extracting the objects from the video content.

In [18], image degradation has been accounted to the image structural distortions, and image quality has been put in direct relationship with the perceptive satisfaction of the user. In [10], a *utility function* has been defined that makes explicit the relationships between resources (bandwidth, display, etc.) and utilities (objective or subjective quality, user's satisfaction, etc.).

With these premises, the main contribution of this work is the analysis of the usefulness of object shape coding, performed by considering the different parameters in shape coding and the trade-off between shape coding overhead and resulting improvements. To do this, a metric for evaluating the quality of the compressed video is proposed. It includes both PSNR-based, objective and user-based, subjective metrics. Eventually, the definition of a possible algorithm for the segmentation of moving objects in scene filmed with a static camera, derived by the SAKBOT system proposed in [6], is proposed and utilized for the tests.

2. Background on object and shape coding in MPEG-4

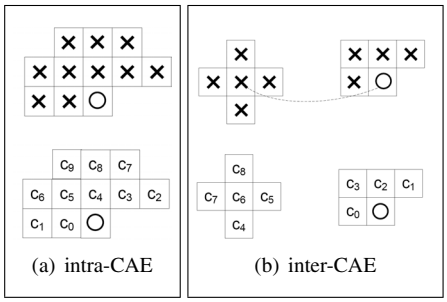


Figure 1. Context number computation in intra-CAE (a) and inter-CAE (b). Circle indicates the current pixel, while 'X' indicates the context pixels.

Object coding in MPEG-4 is based on the concept of *Visual Object* (VO), considered as a temporal sequence of bidimensional images, called *Visual Object Planes* (VOPs), of arbitrary shape. As a particular case, a VOP can have a rectangular shape, also time invariant in terms of dimension and position. Each single VO is defined by its shape and its texture. The texture is defined, as usual, by the YUV components, while the shape is referred as the *alpha* component, composing the so-called *alpha maps*. Basically, three types of VO can be used: *binary objects*, where a binary alpha map is used and no texture information is coded; *textured objects*, in which the texture is associated to a binary alpha map; and *transparent textured objects*, where the alpha map is at gray levels, representing the level of transparency of the VO.

The texture of the VO is coded by using the hybrid approach (DCT combined with motion compensation). In MPEG-4 it is possible to select a different quantization factor (applied in the quantization of the DCT coefficients) for

each VO. Moreover, MPEG-4 includes an additional feature. A significant problem with block-based DCT is that of border blocks. In fact, border blocks can be only partially included in the image and the problem is how to handle transparent pixels out of the image range. Previous standards utilized the *padding* procedure in which the value of the closest opaque pixel is assigned to transparent pixels. MPEG-4 standard includes a new feature, called *shape adaptive DCT*, that modifies the classical fixed block-based DCT with a DCT that adapts to the arbitrary shape of the object, avoiding the use of padding. This functionality can be activated singularly for each VO.

To code the shape, instead, a bitmap-based approach, called *CAE* (Context-based Arithmetic Encoding) [1, 3] has been chosen due to its good efficiency and relatively low computational cost. In the CAE algorithm, the bounding box containing the alpha map is divided in blocks (called *Binary Alpha Blocks* - BAB) of 16x16 pixels. The standard formalizes two versions of the CAE, called *intra-CAE* and *inter-CAE*. In the latter case, a motion compensation is also performed on the BAB. Specifically, seven different modes [1] can be used to code a BAB. In addition, similarly to the texture coding, shape coding can be lossless or lossy. In MPEG-4 lossy coding is achieved by subsampling the BABs: for instance, instead of using the full 16x16 BAB, only a subsample of 8x8 or 4x4 can be used. It is worth noting, however, that lossy shape coding can lead to significant distortions, less tolerable for the users.

The CAE algorithm is based on the following process. First, a *context number* is computed for each pixel of the BAB. The context number is a number obtained by analyzing the binary values of neighboring pixels. Referring to Fig. 1, in the case of intra-CAE (Fig. 1(a)), the context number is a 10-bit value composed by $C_{intra} = (c_9, c_8, c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0)$, while in the case of inter-CAE (Fig. 1(b)) the context number C_{inter} is a 9-bit value that takes into account also values from the motion compensated BAB (left of Fig. 1(b)). Pixels out of the BAB are treated as transparent.

The computed context number is then used as an index to access to a table with probability values. Two different tables P_INTRA and P_INTER are used. For instance, $P\_INTRA[11]$ is a 16-bit number that represents the probability (normalized in the range [0-65535]), that a pixel is transparent, given a context number of 11. This probability is used to guide an arithmetic coder in the assignment of the codeword [3]. Since the alpha component is crucial for an efficient coding and since it has been proved that errors in the shape are less tolerable for the user than errors in the texture, its updating should be more frequent than the updating of the YUV (texture) components.

In the case of transparent textured objects, besides the CAE for the shape, the same coding process applied to the texture is also applied to the gray levels of the alpha maps.

3. Object segmentation from fixed camera

As stated in the previous section, MPEG-4 standard does not propose any segmentation algorithm in order to obtain the VOPs.

Hundreds of papers have been reported in the literature on the topic of object segmentation from videos. Some seminal works are reported in [2, 9, 14, 20, 21]. Among them, some researchers proposed object segmentation techniques explicitly for MPEG/MPEG-4 coding [21]. On the other hand, some of these approaches exploit MPEG coded videos to extract video objects in the compressed domain [2, 20]. However, the description of the details of the numerous methods for video object segmentation is beyond the scopes of this paper.

Instead, this section will present our proposal for object segmentation in order to provide the MPEG-4 encoder with the VOPs by means of binary masks. The approach is derived from the system called SAKBOT (Statistical And Knowledge-Based Object Tracker) whose scheme is reported in Fig. 2 and complete details can be found in [6]. Without going into too much details, the motion detection embedded in the SAKBOT system is based on background subtraction and models the background using statistics and knowledge-based assumptions. In fact, the background model is computed frame by frame by using a statistical function (temporal median) and taking into account the knowledge acquired about the scene in previous frames. In practice, the background model is updated differently if the considered pixel belongs to a previously detected visual object: in this case, the background model is kept unchanged because the current value is surely not describing the background. SAKBOT also implements an effective shadow-detection algorithm [12]. Shadows are detected by means of an appearance model that relies on the fact that cast shadows darken the background that they cover, but slightly change the color. An object validation task is performed to remove small objects and to distinguish between real and apparent (ghost) moving objects.

The extracted visual objects are processed by a tracking module that must ensure the maintenance of the tracks also in the case of occlusions due to static or moving objects (e.g., furniture or other moving people). Our tracking module is a suitable adaptation of that proposed by Senior et al. in [16], which suggests the use of an incremental and adaptive definition of tracks using a probabilistic and color-based appearance model of the detected blob. More details can be found in [8]. Eventually, high-level modules have been recently developed, such as a posture classification module able to identified the current posture of a person in the scene (see [7] for further details).

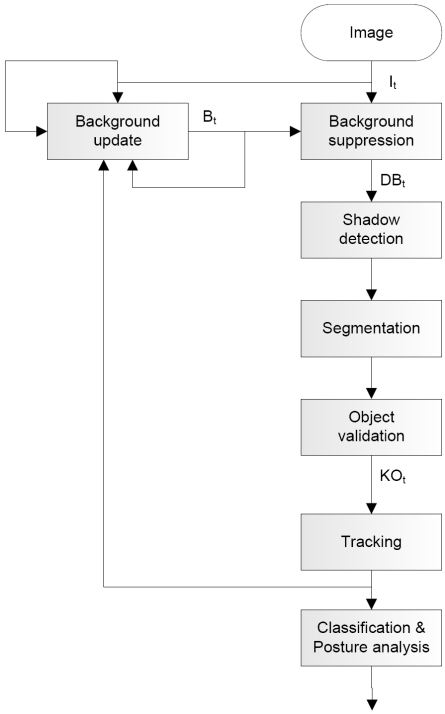


Figure 2. The process scheme of the SAKBOT algorithm.

4. Analysis of the results

The usefulness of object shape coding depends mainly on which are the application and the final users of the system. As stated in the introduction, the application we focus on is the transmission/storage of videos, in which the aim is to achieve the best trade-off between filesize (or bitrate, in the case of transmission) and quality of significant entities of the video. Since in typical applications (such as video surveillance or news broadcasting) motion is the key feature, a motion detector (described in the previous section) has been used to segment video moving objects with respect to the background. While the filesize is an objective, measurable value, quality is a more fuzzy concept. For this reason, in this work we propose to exploit both a quantitative, objective metric (PSNR), and a qualitative, subjective metric, defined in the following.

For all our experiments, we used a open-source version of the Microsoft MPEG-4 codec, developed during the realization of MPEG-4 Video standard ISO/IEC 14496-2.

This section is divided in three subsections. The first will analyze the effect of shape coding parameters, while the second will compare object shape coding with standard, frame-based coding. Eventually, the third subsection will present the results of the subjective tests.

4.1. Effect of parameters

This first phase of the analysis has been carried out on two videos from fixed camera (two example snapshots are reported in Fig. 3).

(a) Video1 (b) Video2

Figure 3. Two snapshots of the videos used for the first phase.

As reported in section 2, MPEG-4 allows both lossless and lossy shape coding. Our software contains a parameter with which it is possible to set a threshold for the maximum error in shape coding, and the encoder automatically selects "how much lossy" the shape coding must be. The error is computed with the SAD (Sum of Absolute Distance) with respect to the original shape. Therefore, we compared lossless coding with lossy coding with different thresholds (0, 16, 32, 64, 128, and 256). It must be noticed that, in these tests, the shape adaptive DCT has been enabled.

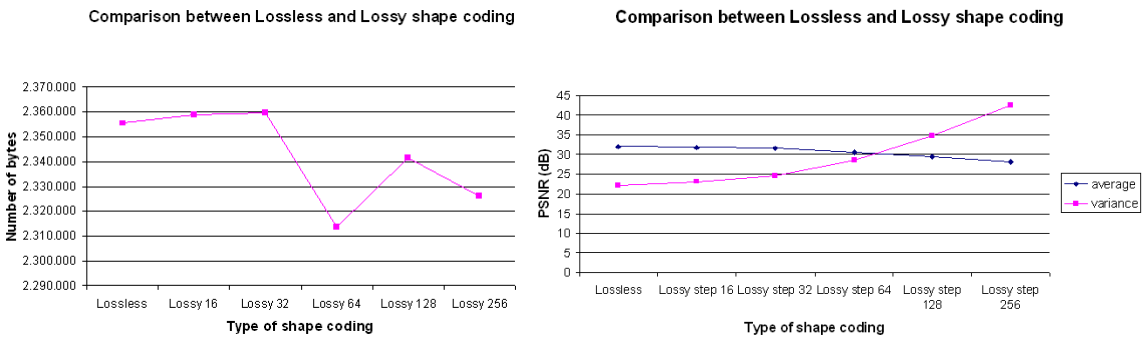
The results are summarized in the graphs of Fig. 4. Examining them, it is straightforward to notice that lossy shape coding is not convenient, since it assures a saving of few bytes (or evens some increase, probably due to a less effective entropy coding of the DCT coefficients) at the cost of up to 4 dB in the average PSNR.

Another feature whose effect on performance can be interesting to analyze is the shape adaptive DCT. In Table 1 it is evident that the enabling of the shape adaptive DCT slightly improves the PSNR (approx. 0.5 dB), but requires additional bytes (even though less than 2% of the total bitrate). Since the difference with respect to disabled SA-DCT is negligible, we conclude that enabling SA-DCT does not influence much the performance. These effects increase if many objects are present in the scene simultaneously.

	Filesize (bytes)	**PSNR (dB)**
SA-DCT enabled	8201672	29,21
SA-DCT disabled	8080266	28,62

Table 1. Performance analysis for enabling or disabling the shape adaptive DCT.

As final test of this first phase, we examined the bitrate



(a) Filesize (b) PSNR

Figure 4. Performance analysis for lossless vs. lossy shape coding.

differences between coding only the shape or also the texture. In particular, we selected three profiles:

- *shape only*: only the shape is coded; arbitrary shapes are allowed;
- *shape+texture*: both shape and texture are coded; arbitrary shapes are allowed;
- *texture only*: arbitrary shapes are not allowed; objects are coded with bounding (rectangular) boxes, with no alpha planes; this is similar to what MPEG-2 already did.

The graph in Fig. 5 shows the cost (at different level of shape coding quantization) of using the (arbitrary) shape in object coding. Comparing the filesize required for "texture only" profile with that required for coding the shape too, it is evident that the required amount is limited (ranging between 12% and 16%). On the other hand, coding only the shape requires about half the bytes of coding shape and texture together. This is due to the fixed overhead needed for the MPEG-4 syntax.

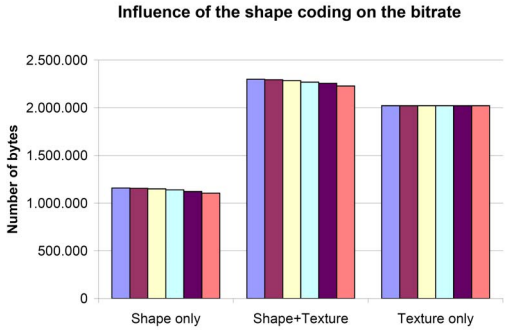


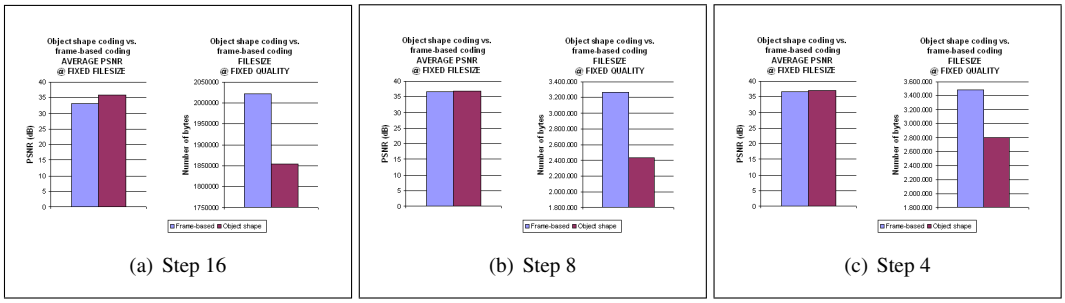
Figure 5. Shape coding vs. the filesize.

4.2. Object shape coding vs. frame-based coding

In comparing object shape coding with frame-based coding, we must first decide how the comparison must be carried out. Suppose the user has a maximum bandwidth or a maximum filesize allowed for a video. It can be useful to know which of the two coding methods can obtain the higher quality. It is worth noting that, in our study, we consider a user-centered approach. The user is interested in some parts (either objects or events) of the video more than in others (such as the fixed background). For this reason, we propose the use of a *weighted PSNR (WPSNR)* as quantitative quality measure. Basically, the WPSNR weights more the pixels belonging to significant entities. In our experiments, we fixed the weights to 80% for the VOs and 20% for the background. In parallel, object shape coding exploits the MPEG-4 functionality described in section 2 to compress more the background (saving bits) and less the VOs (improving the quality).

The dual comparison is at fixed quality. Suppose the user accepts at least a certain level of quality for the video. We want to know which between the object shape coding and the frame-based coding requires less bits to grant this level of quality.

Results are reported in Fig. 6. Each pair of graphs includes, on the left, the comparison at fixed filesize (obtained by using in the frame-based coding a fixed quantization step, respectively, of 16, 8 and 4, on the range 1-31, where 31 means maximum compression), and, on the right, the comparison at fixed quality (achieved by adjusting the quantization steps of VOs and background in the object shape coding in order to obtain the same average PSNR of the frame-based version with the quantization step as above). It is evident that *the shape coding outperforms the frame-based by achieving higher PSNR at fixed filesize and lower filesize (i.e., required bandwidth) at fixed quality*.



(a) Step 16 (b) Step 8 (c) Step 4

Figure 6. Object shape vs. frame-base coding.

The differences are more prominent at higher compression (step 16). This means that object shape coding is more useful at lower bitrate, or, in general, when the video needs to be compressed more, that is the emergent requirement in the multimedia research field. In particular, object shape coding can reach level of compression (at reasonable quality) that frame-based coding can not reach, thanks to the saving of bits in compressing more the background.

Unfortunately, this conclusion is true only for fixed camera. In [8], we demonstrated that in case of moving camera a frame-based (or event-based, i.e. with different quantization steps in different frames corresponding to different events) approach is more convenient. This is due mainly to two reasons: the first is that, if the camera is moving, the background must be re-sent each time it changes, and the required number of bytes increases; the second is that if the camera moves the shape prediction is not effective and shape coding overhead is increased.

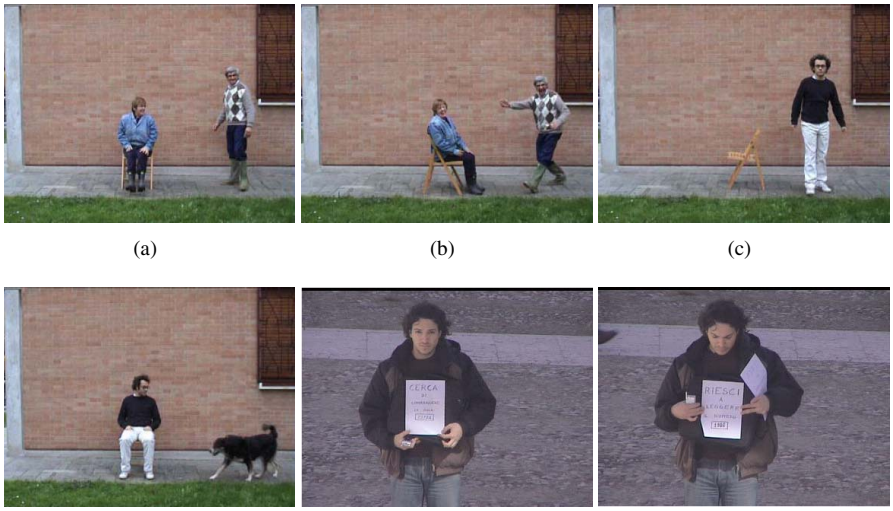
4.3. Qualitative analysis

Quantitative analysis based on PSNR provides interesting cues on the quality of the video. Unfortunately, it does not take into account the human visual system (HVS). For example, PSNR is not sufficiently effective in considering MPEG typical artifacts, such as *blocking* (the presence of 8 x 8 pixel pattern blocks in the compressed video stream that were not part of the original source) and *ringing* (the presence of a blurring, or out of focus effect around the edges of an object that is moving from frame to frame; it occurs more often when there is a large amount of motion between frames of the video).

To take into account the quality as perceived by the user, we have prepared a subjective test. A benchmark of differently-compressed videos has been seen by a set of 25 users (mainly students of multimedia-based courses and experts in the field). The benchmark consists in 18 videos obtained from the six original videos whose snapshots are reported in Fig. 7.

Each video has frames of 384x288 pixels and 24 bits per pixels. For each of them, three versions are created with SAKBOT and MPEG-4 encoder: the first version is compressed with the frame-based encoding at a very low bitrate (ranging from 22 to 29 kbps) and 10 fps (frames per second); the second is compressed with object shape coding with high compression (step 31) for the background and low compression (step 1) for the objects, the bitrate is approximately 100 kbps and the frame rate is 25 fps; the third version is, instead, a frame-based version at higher bitrate (ranging from 133 to 153 kbps) and 25 fps.

Selecting the right questions and the way to answer them is of fundamental importance in subjective tests. We prepare our test with an approach similar to that reported in [19]. The test is divided in three sections: the first contains questions on the *overall quality* of the video, the second reports questions aiming at catching the *cognitive effectiveness* of the video, and the third is related to the *willingness to pay*. Specifically, the overall quality is estimated on a Likert attitudinal seven-point scale, where 1 means “awful quality” and 7 means “perfect quality”. The second part, instead, is different from video to video and contains questions on the recognition of the video scene. For example, referring the video in Fig. 7(d), the question is “Does the dog wear a yellow collar?” and the possible answers are “Of course”, “I think so”, “Surely no”, “It doesn’t seem so”, “It is not possible to see it”. On the videos of Fig. 7(e) and 7(f), instead, the question is to report the codes in the box at the bottom that contain numbers and letters, respectively, and the correctness is evaluated on how many numbers/letters have been guessed (normalized to the interval [0-1]). In addition, as second question of this section, the user has been asked to quantify (in a seven-point Likert scale) how much the given version of the video has helped in answering to the previous question. Obviously, these questions are directly related to the cognitive effectiveness granted by the video and are implicit subjective measures of the video quality. Eventually, the last section associates to each version a cost (computed considering to access to the video with a last-generation cell phone and on the basis of the average current costs of Italian telecommunication companies). The user is asked to choose which version he prefers considering the costs.



(a) (b) (c)

(d) (e) (f)

Figure 7. Snapshots from the benchmark used in qualitative tests.

The results collected on the whole user set for the overall quality have been compared with the average WPSNR of the videos (Fig. 8). It is evident that the subjective quality grade reflects the trend and the ranking among versions of the PSNR, although the difference between the frame-based version 3 and the shape-based version 2 is reduced. This confirms that the PSNR (even if modified in WPSNR) does not consider distortion effects (such as the blocking due to the hybrid macroblocks in the border within the background and the VOs) that are annoying for the user.

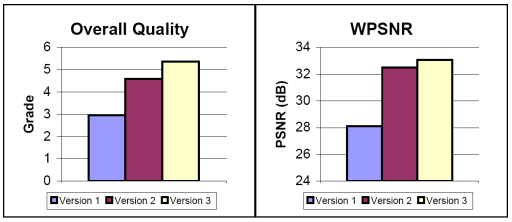


Figure 8. Overall quality.

The graphs in Fig. 9 show the average grade selected in the second section of the test. These results demonstrate that the version 2 (shape-based) allows a better recognition of the scene, though at the subjective question (right graph) the user prefers (slightly) the third version. This unexpected result can lead to the conclusion that object shape coding allows a better recognition of the video scene thanks to the lower compression of significant parts, though the HVS is not enough sensitive to notice it.

When we asked to the users to choose the best version, knowing the price they have to pay to obtain it, the most chosen version results to be version 1 (see Fig. 10). This

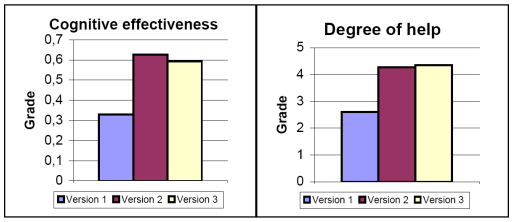


Figure 9. Cognitive effectiveness.

is due to the fact that the version 1 costs significantly less than the other two, and to the fact that the users still do not know if they answered correctly to the question in the second section. Thus, they are not aware of the limitations in scene recognition generated by using version 1.

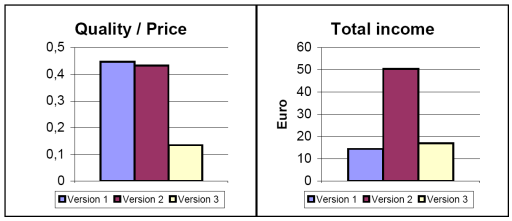


Figure 10. Quality-price trade-off.

The rightmost graph in Fig. 10 reports the overall income that a company will earn on the different versions. In practice, this value is computed as the product of the number of times that the version has been selected by the price of each single video of that version. It is evident that the version 2 is the more convenient.

5. Concluding remarks

The main objective of this paper was to provide a (partial) answer to the question "Is object shape coding convenient for video compression?". The answer depends on many factors, but the analysis reported in this paper tries to draw the path towards the complete answer.

The conclusions drawn in this paper lead to suggest the use of object shape coding as far as low bitrate are available. In fact, it has been demonstrated that object shape coding can achieved better results, in terms of both saved bits and gained average quality. In particular, it provides, filesize being equal, higher quality for significant parts of the video, and, consequently, a better recognition of the scene. The subjective test, however, has proved that the possible artifacts (especially blocking artifacts) introduced by the selective compression of different parts of the same frame can affect the user's satisfaction more than in the case of frame-based coding, where the average quality is worst, but equally distributed in the image.

The analysis is performed on a benchmark of videos from fixed camera and using a specific segmentator for extracting the VOs for the MPEG-4 encoder. However, besides the fixed camera constraint, we feel the considerations are almost general and can be applied also to other videos and different scenarios.

Acknowledgment

The project is funded by the European VI FP, Network of Excellence DELOS (2004-06) on digital libraries.

References

[1] Information technology coding of audio-visual objects. Technical Report 14 4962, 2nd ed., ISO/IEC, Switzerland, 2001.

[2] R. Babu, K. Ramakrishnan, and S. Srinivasan. Video object segmentation: a compressed domain approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):462–474, Apr. 2004.

[3] N. Brady. MPEG-4 standardized methods for the compression of arbitrarily shaped video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(6):11701189, Dec. 1999.

[4] A. Cavallaro, O. Steiger, and T. Ebrahimi. Semantic segmentation and description for video transcoding. In *Special Session on Video Segmentation for Semantic Annotation and Transcoding at International Conference on Multimedia & Expo (ICME)*, pages 597–600, 2003.

[5] C. Crockford and H. Agius. Modelling VCR-like video content navigation. *Displays*, 26(2):79–96, 2005.

[6] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, Oct. 2003.

[7] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human behaviour analysis. *IEEE Transcations on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35(1):42–54, Jan. 2005.

[8] R. Cucchiara, C. Grana, G. Tardini, and R. Vezzani. Probabilistic people tracking for occlusion handling. In *Proc. of IEEE Int'l Conference on Pattern Recognition*, volume 1, pages 132–135, 2004.

[9] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, Aug. 2000.

[10] J.-G. Kim, Y. Wang, and S.-F. Chang;. Content-adaptive utility-based video adaptation. In *Proc. of IEEE Int'l Conference on Multimedia & Expo*, pages 281–284, July 2003.

[11] M. Kunt. *Object-based Video Coding*, chapter 6.3, pages 585–596. in 'Handbook of Image and Video Processing'. Academic Press, 2000.

[12] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, July 2003.

[13] A. Puri, X. Chen, and A. Luthra. Video coding using the H.264/MPEG-4 AVC compression standard. *Signal Processing: Image Communication*, 19:793–849, 2004.

[14] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, Mar. 2005.

[15] J. Reichel, H. Schwarz, and M. Wien. Scalable video coding - working draft 1. Technical Report JVT-N020, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Hong Kong, Jan. 2005.

[16] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Tracking people with probabilistic appearance models. In *Proceedings of International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) systems*, 2002.

[17] A. Vetro, T. Haga, K. Sumi, and H. Sun;. Object-based coding for long-term archive of surveillance video. In *Proc. of IEEE Int'l Conference on Multimedia & Expo*, volume 2, pages 417–420, 2003.

[18] Z. Wang, A. C. Bovik, and L. Lu. Why is the image assessment so difficult? In *Proc. of the IEEE Conference on Acoustics Speech and Signal processing*, May 2002.

[19] G. Wikstrand and J. Sun. Determining utility functions for streaming low bit rate football video. In *Proc. of IASTED Intl Conf on Internet and Multimedia Systems and Applications (IMSA)*, 2004.

[20] X.-D. Yu, L.-Y. Duan, and Q. Tian. Robust moving video object segmentation in the MPEG compressed domain. In *Proc. of IEEE Int'l Conference on Image Processing*, volume 3, pages 933–936, 2003.

[21] D. Zhong and S.-F. Chang. AMOS: an active system for MPEG-4 video object segmentation. In *Proc. of IEEE Int'l Conference on Image Processing*, volume 2, pages 647–651, 1998.