# Semantic Transcoding of Videos by Using Adaptive Quantization

RITA CUCCHIARA, COSTANTINO GRANA, ANDREA PRATI
Dipartimento di Ingegneria dell'Informazione
University of Modena and Reggio Emilia
ITALY
{cucchiara.rita, grana.costantino, prati.andrea}@unimore.it

## Abstract

This paper proposes the use of an approach of video transcoding driven by the video content and provided with the adaptive quantization of MPEG standards. Computer vision techniques can extract semantics from videos according with user's interests: the video semantics is exploited to adapt the video in order to meet the device's capabilities and the user's requirements and preserve the best quality possible. Well assessed video analysis techniques are used to segment the video into objects grouped in *classes of relevance* to which the user can assign a weight proportional to their relevance. This weight is used to decide the quantization values to be applied in the MPEG-2 encoding to each macroblock. A modified version of the PSNR (Peak Signal-to-Noise Ratio) is used as performance metric and comparative evaluation is reported with respect to other coding standards such as JPEG, JPEG 2000, (basic) MPEG-2, and MPEG-4. Experimental results are provided on different situations, one indoor and one outdoor.

**Keywords:** Video transcoding, adaptive quantization, motion detection, video adaptation, performance evaluation.

## 1 Introduction

Universal multimedia accessibility is nowadays mandatory for every system that aims at publishing or distributing multimedia contents. Among multimedia data, videos are particularly challenging due to both the tremendous amount of data to be transmitted and the variability in terms of image size, color depth, frame rate, and so on. First, the amount of data is still a problem since many users still do not have the possibility to access to Internet with high speed connections. This is particularly true in the case of new devices such as PDAs (Personal Digital Assistants), hand-held PCs, and others, that provide portability and ubiquitous access to the network at the expense of limited bandwidth. Moreover, the variability of video format creates another difficulty because these devices have typically very compact sizes, and, therefore, limited screen capabilities.

Adapting videos to the device's capabilities is well known in the literature with the name of *transcoding* [1–6], though *data (video) adaptation* is a more general definition. In the matter of fact, transcoding assumes the presence of a coded video, whose code has to be changed, while the same technique can not be used in live, raw (not coded) videos to adapt its format to the requirements.

Transcoding is a term currently associated with the process of changing a multimedia object format into another: it is referred either as an *intramedia* transcoding when the media nature does not change (e.g. varying the video compression rate, the color description or transforming a video from M-JPEG to MPEG-4) or as an *intermedia* transcoding when also the media nature changes (for instance transforming video in audio by analyzing it and beeping when a certain event happens). In this paper we will focus only on intramedia video transcoding. The selection among the large plethora of multimedia formats must be done in terms of tradeoff costs/benefits: the costs rely on the hardware resources used to provide transcoding (processing capabilities especially for on-the-fly transcoding and storing capabilities if many transcoded videos are available at the same time); the benefits should be measured both in terms of bandwidth saving and video fidelity.

In particular, within video transcoding, we want to adopt the term *semantic* or *content-based transcoding* with the twofold meaning that the transcoding process is guided by the video semantics and, at the same time, the transformation may change the video perception and possibly its appearance, while preserving the semantics. The capability of preserving semantics allows the effective scalability of the video on almost every existing device.

In practice, semantic transcoding assumes that the user does not want to access to all the data, but only to the data semantically useful. The possibility to select *not only what to see* but also *what to see better* should be a winning strategy. Therefore, semantic transcoding techniques are spreading [2,7,8,9], including the support of a previous annotation, for instance with MPEG7 or XML [10], with content

analysis at image- or frame-level [1], or exploiting object detection and tracking approaches working in real-time on videos [8].

In this paper we propose an approach that can be summarized as follows:

1) *Semantics extraction*: first, we use a set of computer vision techniques to extract objects from videos, e.g. to extract moving people in a room, or vehicles in a parking lot, etc. The extracted objects are then classified according to previously defined *classes of relevance* [6]. These classes represent the semantics that our system can extract from the video and are supposed to be a super-set of the semantics to which the users are interested.

2) *Definition of the user's requests*: the user assigns to each class of relevance a weight proportional to the importance (relevance exactly) that the objects belonging to the class have for him/her. For example, in video surveillance applications the user can be very interested in moving people and does not care much of the background. Moreover, the user must specify the constraints/capabilities of the device, he/she will use to see the video, in particular the screen size and the maximum bandwidth of the connection available.

3) *Adaptation (transcoding) of video's content*: based on the information provided in point 1 and 2 the system adapts the video to the user's requests and the device's capabilities, as we will describe in the next section.

In previous works [6,8], we proposed an approach in which the objects extracted are JPEG encoded differently depending on the weights assigned to their class. Each object is sent in a separated image. Also the background is sent (one every *n* frames, with *n* that changes dynamically) in a separated image. At the client side, the decoder takes the current background and superimposes the objects. Thus we proposed an M-JPEG code *without any temporal prediction*. In that proposal, both the encoder and the decoder were not standard, i.e. the user needs to install the decoder at the client side to see the video.

In this paper, instead, we have the twofold goal to add the temporal prediction to our system (in order to further abate the bandwidth required) and to produce a video that can be played by a standard decoder. For these reasons, we address MPEG standard. In this case, we do not send the extracted objects separately, but the semantics extraction part is used to drive the *adaptive quantization* of frame I in the MPEG stream. Moreover, we exploit the temporal prediction of the MPEG-2 (and possibly MPEG-4) standard to both improve bandwidth reduction and obtain a video playable by a standard decoder.

The paper is structured as follows. The next section will present the papers in the literature addressing semantic transcoding and, in particular, the use of adaptive quantization in MPEG coding. Section 3 will briefly describe the classical transcoding policies, concentrating the discussion on our semantic transcoding and on the computer vision techniques used to achieve it. A detailed description on how we use adaptive quantization will be also reported. Section 4 and 5 will describe the system architecture and the performance evaluation metric, respectively, while Section 6 presents our experimental results. Section 7 reports our conclusions.

## 2 Related Works

According with Vetro *et al.* [9] we can classify transcoding as *spatial*, *temporal*, *code*, *color*, and *object* transcoding. *Spatial* transcoding is the standard frame size downscaling, from standard formats (as CIF 352x288, QCIF 176x144, etc.). This is necessary for some specific clients with limited display resources and allows also bandwidth reduction. *Temporal* transcoding copes with the reduction of the number of frames, either dynamic (to choose when frames can be eliminated according with the changes in the motion vectors [11]) or fixed. In [12] the composition problem (i.e., how to merge/compose transmitted object with the reference/background image in an effective way) is examined, associated with varying the temporal transcoding of multimedia objects. *Color* transcoding, like spatial transcoding, is sometime requested for specific clients (like gray level PDAs). Using less bits for pixel, chrominance suppression (adopting 8 bits gray level) and a more aggressive binarization (1 bit B/W code) are possible transcoding policies that can reduce bandwidth, but also modify the perception of images. It can be accepted by human users, but sometimes should be avoided if the transferred videos must be processed by computer vision algorithms that typically make a large use of colors. *Code* transcoding, i.e. the change of (standard) coding, has been widely analyzed: increasing the level of compression saves bandwidth and sometimes could be acceptable for the video QoS standard too; however, an excessive compression could be unacceptable for many applications due to the loss of details.

Finally, the class *object* or *semantic* transcoding comprises some different techniques to tract

differently multimedia objects in the video [1,9]. We propose to manage moving objects computed with computer vision processes. Basically the goal is to extract semantically valuable objects from the scene and transfer them with the lower amount of compression in order to preserve both details and speed. In previous works [6,8] we demonstrated that the proposed semantic transcoding can outperform almost always the other methods, at least those that do not use temporal prediction such as MPEG standards.

Several approaches have been proposed addressing semantic transcoding, mostly dealing with stored videos. They are often associated to a process of *annotation* that takes care of video content, annotated in the video database [7]. The standardization work of MPEG-7 with Multimedia Content Description Interface has defined the meta-data description coupled with stored videos, to support transcoding applications [13]. For instance, the IBM's Video Semantic Summarization Systems described in [14] exploits MPEG-7 for semantic transcoding. A good survey of transcoding products is presented in [1], where the idea of preserving multimedia content in Web access is well underlined. Transcoding can be provided at server, proxy or client level; the authors of [1] claim that there are some advantages in designing transcoding capability in multimedia servers especially because the provider keeps the control of distributed data. Therefore the authors provide a general framework, called InfoPyramid, to store annotated information and transcoded versions of the same multimedia content in the server. A similar approach is proposed in Columbia's video on demand test-bed [15]. An alternative solution to storing multimedia data already transcoded is to provide transcoding directly on compressed data [15,16]. It is exploited especially to downscale the compression rate or the video resolution [17].

There have been many papers dealing with adaptive quantization in MPEG standard [18–21]. Most common quantization schemes assume quantizer design based on training sets and/or source models which can represent the statistics of the entire input. For example, after analyzing the images coming from a camera for a certain amount of time, one could assume that a fixed quantization table for an M-JPEG encoder would lead to the desired average bit-rate. The performance of quantizers using such a priori knowledge of the input is largely affected by the choice of the training set or the input model, resulting in a loss of performance if there is a mismatch between the actual input statistics and the design assumptions. In practical situations of quantizing complex data, it may be hard to have a good training set or sufficient knowledge on the input model. Thus there exists a motivation for adaptive quantization schemes which do not require any (or as few as possible) priori information on the signal of interest.

In the literature, adaptive quantization schemes are categorized into two broad classes [22]: *backward adaptation* and *forward adaptation*. In backward adaptation [23], quantizers are updated based only on the previously quantized data which are available to both the encoder and the decoder, and has the advantage of avoiding the need for overhead information transmission to the decoding end.

In forward adaptive quantization [20], the encoder makes a decision on how to update the quantizer by probing current and future inputs. Since the encoder's decision is based on information unavailable to the decoder, side information has to be sent to the decoder to specify the changes. If this information is valuable also for the decoding end, it is no more to be considered useless overhead, but additional semantics that allows more flexible handling of data.

The uses of the MQUANT parameter of the MPEG syntax range from flexible compression level to account for the channel size [18], to perceptually invisible compression gains [24,25]. Also MPEG-2 Test Model 5 rate control algorithm employs considerations based on the human visual system [26].

# 3 Description of the System

## 3.1 System Architecture

The architecture of the proposed system is reported in Fig. 1. The transcoding can be applied both to videos acquired from live cameras and to stored videos. In both cases, the videos are first decoded (unless they are uncompressed). This step is fundamental since our system do not work in the compressed domain. The reason is that, as core of our system for *semantics extraction*, there is a computer vision module able to detect, extract, track and classify objects in video sequences. These techniques are part of a wider system called *Sakbot* (Statistical And Knowledge Based Object Tracker). This system is based on *background suppression* and is able to extract both moving objects and objects that stop after being in motion. The more interesting classes of objects that the system is able to detect, classify and

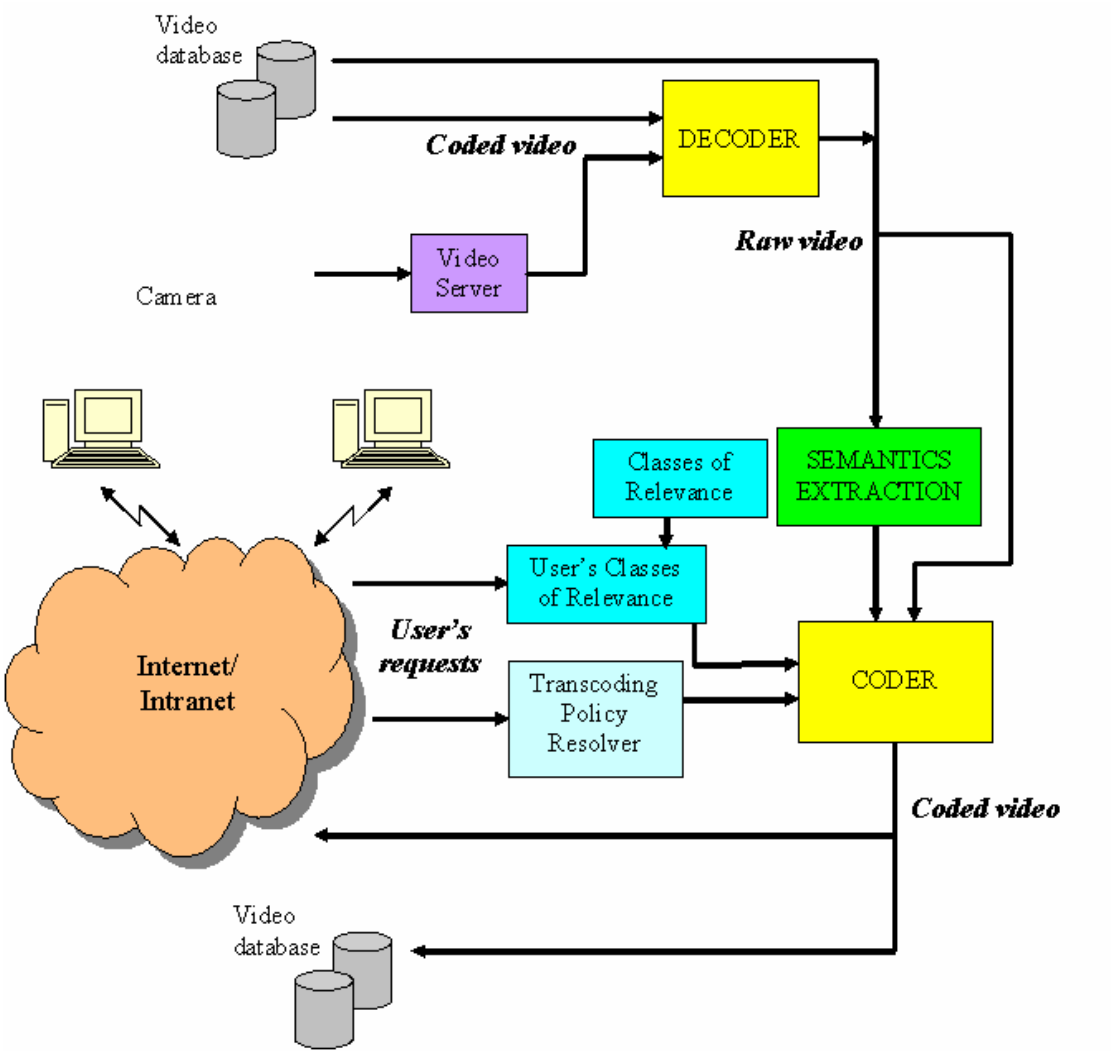



**Fig. 1. Architecture of the system.**

track, are vehicles, people, and pieces of furniture (such as chairs). Moreover, the classification system of Sakbot can further segment a person into "body" and "head/face", or extract the license plate of a vehicle. The detailed description of this system is beyond the scopes of this paper, but further details can be found in [26-28].

The set $C=\{C_1,C_2, ..., C_n\}$ of classes of objects that the system can extract from the video can be further grouped into the *user's classes of relevance* $UC=\{UC_1,UC_2, ..., UC_k\}$, with $k \leq n$. In fact, some classes extracted can result in a too fine classification for the aims of the user: for example, the system is able to segment moving people into body and head, but the user can not be interested in such a distinction. Since the sum of the weights associated to the classes of relevance must be equal to 1, assuming that the user considers the moving person (in its entirety) as very relevant (weight equal to 0.9), he/she could not set to 0.9 the weights of both the class body and the class head, because the sum will be greater than 1. In this case, the user can define a user's class that includes both classes and sets the weight for this class to 0.9. As a consequence, we can define a set of weight $w=\{w_1, w_2, ..., w_k\}$ whose sum is equal to 1.

Thus, the user must define:

i) the maximum bandwidth available;
ii) the maximum screen size available;
iii) his interest in the user's classes of relevance, by means of the set of weights *w*.

This information are used by the *TPR* (Transcoding Policy Resolver) to select the more suitable policies for the transcoding, first to fit the display and computing capabilities of the user's device. Besides standard spatial, temporal, color and code transcoding policies we proposed two object-based transcoding, one based on M-JPEG and one based on MPEG standard.

In the case of M-JPEG, we compress differently the objects depending on the weights associated to their class. In particular, the JPEG compression level

is proportional to the weight $w_i$. This is done for each user's class. Instead, the background class is sent with both code transcoding and temporal transcoding: the latter is achieved dynamically, sending a new background only when it is significantly different from the reference background (with the *Changed* label). The composition (i.e., the merge between moving objects and the background model) is then performed with a method called `Alpha_obj_tr` (object transcoding with alpha planes). In the `Alpha_obj_tr` transcoding for each frame we send a list of bounding boxes compressed in JPEG and the alpha planes coded in RLE (run length encoding) that is more suitable for B/W templates as the alpha planes are. The B/W masks are used to decide singularly which pixels of the object to superimpose to the background. Results of this technique are discussed in [6].

### 3.2 Semantic Adaptive Quantization

One of the main problems in compressing video data is the large variability that can be observed from time to time and the strong dependency of complexity from the objects in the scene. Exploiting high level information can lead to a better choice of the compression codebook.

Obviously, the simplest encoding algorithm is that which maps each of the input blocks into a codeword regardless of the context. In other words, each block symbol will be considered independently of the others and will be mapped to the nearest codeword as in case of the choice of a quality factor in the JPEG standard. Conversely, optimality is not guaranteed and different proposal have been presented to circumvent this problem. For instance in [21] thresholding is used to remove, in a rate-distortion optimal way, coefficients after having quantized all the image using a single table. Similar approaches have been proposed to improve the performance of wavelet-based encoders.

We can define adaptivity as the ability to change the choice of codeword for a given block depending on the context, so, following the lines given by the MPEG standard, we assume that only a single codebook, defined by the weights in the quantization matrices, is available, but that we can modify it by choosing different multipliers for each macroblock. This leads to the production of a standard MPEG stream, but with greatly different compression, depending on the image region that is under examination. Of course, artifacts will be easily noticeable on transition macroblocks, but the user gains the ability to specify where to put most effort, in consideration of his viewing and connection capabilities. We thus adopt the Sakbot system to automatically identify and classify different objects, or classes, in the scene and then provide them to an encoder that adapts the quantization multiplier to the identified object (or to the most represented one for the specific macroblock).

The scheme of this process is reported in Fig. 2. Sakbot extracts from the raw, decoded video, the objects present in the I ($F_i$) frames. Moreover, it computes the background with an adaptive model based on temporal median filtering and on a feedback on the objects extracted in the preceding frame. In this way, as it is possible to see in Fig. 2, there are no holes in the background image (even where the background is currently covered by an object) and we can obtain a better composition. Then, the classifier assigns each object to the previously defined user's classes of relevance $UC_i$. Using the weights assigned by the user, TPR computes the quantization multipliers $QS_i$ (see above).

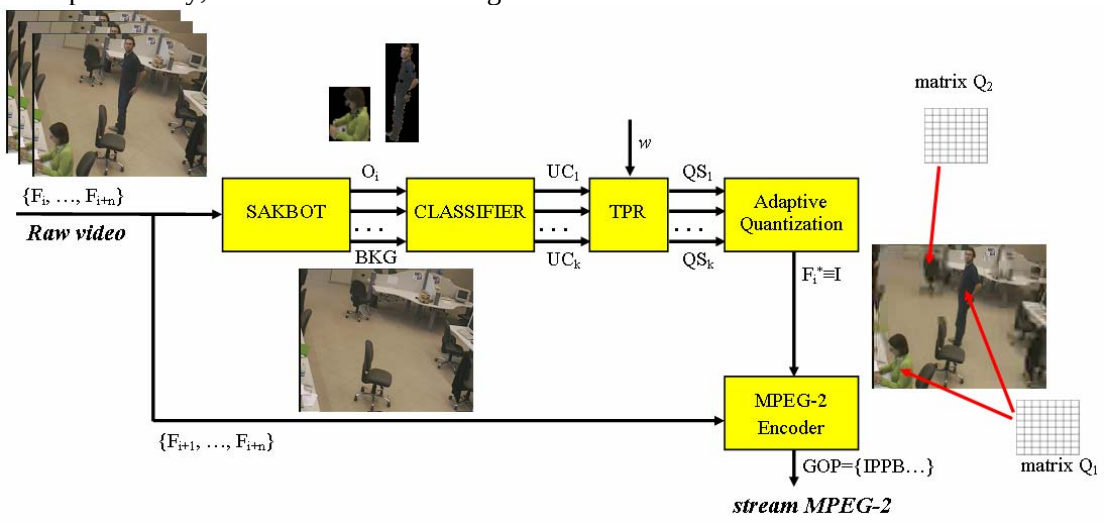



**Fig. 2. Scheme of the adaptive quantization used.**

By multiplying them with the main quantization matrix we can obtain a quantization matrix for each class of relevance defined by the user. In the test described in the next section, for instance, we considered only two classes $UC_1$={moving people} and $UC_2$={background}. The first is very relevant and thus the multiplier is low (quantization matrix $Q_1$), whereas for the background we use a large multiplier, that will result in a more strong quantization (matrix $Q_2$). Finally, a standard MPEG-2 encoder uses this coded frame as I frame and reconstructs the GOP (Group Of Pictures) of the stream.

One last consideration: we choose to use MPEG-2 standard instead of the newer MPEG-4 due to a technical reason. In MPEG-4 the different quantization values for the macroblocks *within the same Video Object Plane (VOP)* are sent in a differential format: each value for a macroblock (except for the first) is coded as {-2,-1,1,2} with respect to the previous one. This allows MPEG-4 to reduce the bandwidth required for the adaptive quantization (2 bits for each quantization values w.r.t. 5 bits), but restricts the flexibility and thus is not suitable for our technique. We could have used an adaptive quantization among different VOPs (similarly to what we did among different macroblocks in MPEG-2), but the MPEG-4 source code we have includes only Simple Profile, i.e. only one VOP for each frame. Therefore, tests with MPEG-4 are less interesting than the ones with MPEG-2.

## 4 Performance Metric

An universally recognized and utilized metric to evaluate the performance of video transcoding does not exist. This is due to the fact that to evaluate the goodness of a video is not a trivial (and standard) task. What can be satisfactory for a user could be unacceptable for another. Maybe, the best choice would be to show the videos to be compared to a set of different users on different devices and to ask them an evaluation. Unfortunately, this is an unsuitable task to be accomplished, both for the need of large number of users and the subjectivity of the evaluation.

From signal processing theory, one of the possibilities for an automated (fast) and objective evaluation is to compute the distortion that the image/video introduces after the processing (in this case, the transcoding). If the video has not an associated semantic, i.e. there is no distinction between important and useless information, the trade-off between the bandwidth reduction and the minimal distortion of the information is typically the best choice. On the other hand, in real applications the limited bandwidth of the connection is the key constraint and, therefore, the distortion should be minimized.

Similarly to other papers in the literature [1,9,30], we define a model of performance analysis based on the user interests by means of classes of relevance to test the transcoding policies simulating different applications. In the context of semantic video transcoding the defined classes of relevance give a priority in the value of objects that are in the video. Think for instance to video-surveillance applications in which a video from live camera is transmitted remotely to a human operator. In these applications the operator can be interested in seeing only the moving people inside a room: the best transcoding policy in this case should be the one that sends the moving people without any compression and does not send the static part (background) of the scene at all. For this reason the distortion introduced in the background should not be considered (weight equal to 0) or should have a very small weight. Another example can be biometric-based surveillance in which the face of moving people can be the more important region of the scene. Thus we use classes of relevance and their weight not only for deciding the transcoding policy but also for evaluating performance.

A common metric to measure the distortion/error in compressed/transcoded images is the *Peak Signal-to-NoiseRatio (PSNR)* [31], defined as:

$$PSNR = 10 \log_{10} \left( \frac{V_{MAX}^2}{MSE} \right) \qquad (1)$$

where $V_{MAX}$ is the maximum (peak-to-peak) value of the signal to be measured and *MSE* is the Mean Square Error, typically computed as:

$$MSE = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} d^2(i,j) \qquad (2)$$

with *d(i,j)* a properly defined distance to measure the error between original and distorted images. As distance, we used the *Euclidean distance* in the RGB color space.

We define a performance evaluation model that accounts for classes of relevance. To this aim, we call *WMSE* (*Weighted MSE*) the following measure (being $N_{CL}$ the number of classes):

$$WMSE = \sum_{k=1}^{N_{CL}} w_k \cdot MSE_k \qquad (3)$$

with

$$MSE_k = \frac{1}{|UC_k|} \sum_{(i,j)\in UC_k} d^2(i,j) \qquad (4)$$

where $UC_k$ is the set of the points belonging to the class $k$ and $|UC_k|$ is its cardinality.

To decide whether a point *(i,j)* belongs to a certain class or not, we compare with a manually segmented ground truth for the tested videos. The weights $w_k$ are chosen by user (or tuned a-priori) and with the following rules:

$$w_k \geq 0 \quad \forall k = 1,\ldots,N_{CL}; \quad \sum_{i=1}^{N_{CL}} w_i = 1 \qquad (5)$$

Clearly, in the absence of semantics, $WMSE \equiv MSE$. We use PSNR as in Eq. (1), with WMSE in place of MSE.

Moreover, the bandwidth $B$ expressed in Kb/s is also used.

## 5 Experimental Results

In previous works [6, 8], we compared classical transcoding policies (such as spatial, color, temporal, and coding downscaling) with our M-JPEG semantic transcoding. The results were promising and our approach achieved the best performance as trade-off bandwidth/distortion. However, the MPEG standards with temporal prediction showed better results in most of the cases [6]. This is straightforward since in typical applications such as video surveillance, the motion is localized and of limited amount, and motion prediction techniques are particularly effective. With these premises, in this paper we changed our approach by introducing the temporal prediction in our system by means of adaptive quantization. Moreover, our experimental setup will compare only coding policies, assuming that other policies are already applied or useless. As coding techniques we compare JPEG (actually, M-JPEG) taken by the *Independent JPEG group*'s library (`http://www.ijg.org/files`), JPEG 2000 (M-JPEG 2000) from *JasPer* library version 1.600.0 (`http://www.ece.uvic.ca/~mdadams/jasper`), MPEG-2 taken from the *MPEG Software Simulation Group* open source library (`http://www.mpeg.org/MPEG/MSSG`) version 12, MPEG-4 taken from the open source *Xvid* software (`http://www.xvid.org`) and MPEG-2 with our semantic adaptive quantization.

As a benchmark, we used two example sequences, whose example images are reported in Fig. 3. Indoors and outdoors have different characteristics both from the computer vision and the possible applications' point of view. For this reason,

(a) (b)

**Fig. 3. Example images of the two test sequences.**

we present results from an indoor and an outdoor sequence. The outdoor sequence is composed of 300 frames of CIF (352x288) size and it has been taken from a freeway. The only moving objects are vehicles. The indoor sequence is composed of 425 frames of size QCIF (176x144) and is from our lab. Two people and a chair are moved in the video. Please note that in Fig. 3(b) the picture is shown at CIF size. In both sequences, we simulated the request of a user with two classes of relevance $UC_1$ and $UC_2$ and weights $w=\{w_1, w_2\}=\{0.9, 0.1\}$. In the outdoor sequence $UC_1$ is represented by moving vehicles, whereas in the indoor sequence all the moving objects (people and chair) are considered relevant.

The performance comparison on the outdoor sequence is summarized in Fig. 4. These results are obtained as follows. Within the parameters' space we can select the screen size, the maximum bandwidth and the weights for the classes of relevance. All these parameters can affect the output by changing the quantization values and they are in theory independent one each other. In our experiments, however, we set the screen size as an initial, mandatory constraint, i.e. a priori fixed. MPEG standard (as well as JPEG) decreases bandwidth occupation by increasing the quantization values (that is eliminating more frequencies in the DCT domain) at the expenses of the quality of the image. In our case, we can move along the bandwidth's range of values by changing the quantization values of both classes. In MPEG-2 these values are provided as an index from 1 to 31 of a multiplier vector (that can increase in a non-linear relation with the index), where 1 indicates the lowest multiplier (i.e., the lowest quantization values, best quality).

To obtain the graph of Fig. 4 we first start with both values set to 1, and then keep the $UC_1$ value fixed and start increasing the $UC_2$ value. Once we reach the value 31, we set both values to 2, and start increasing $UC_2$ value, keeping once again $UC_1$ one fixed. This process stops when both the values are set to 31. Please note that in Fig. 4 shows only lowest bandwidth values and that when both quantization values are equal to 31 our semantic MPEG-2 and normal MPEG-2 obtain the same result.

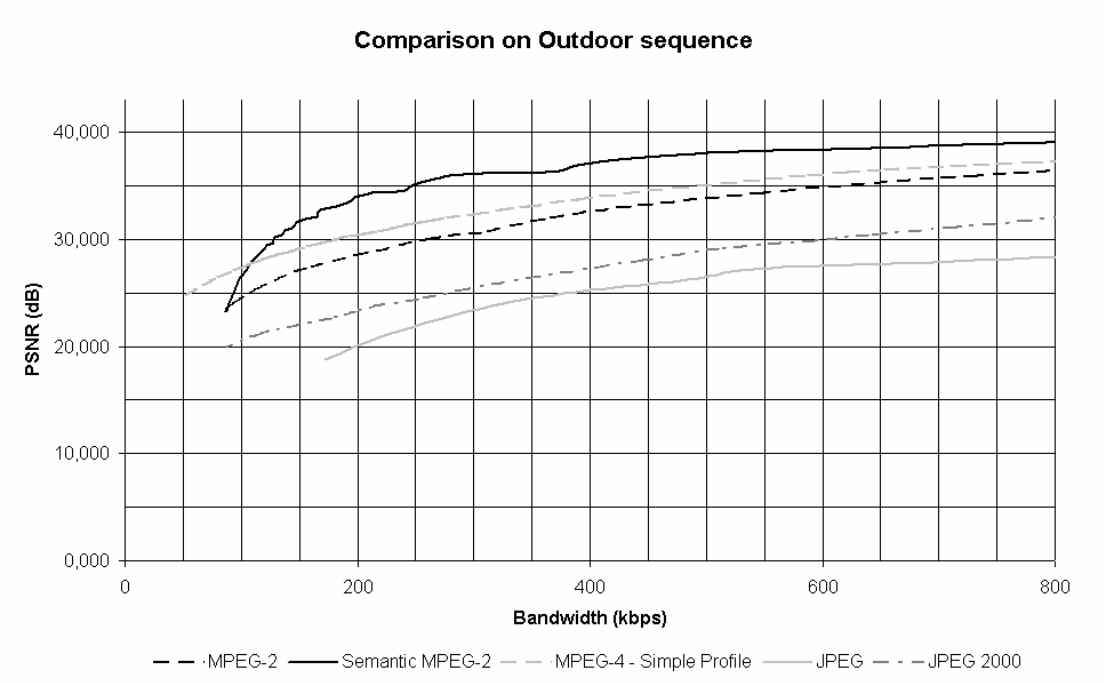



**Fig. 4. Comparison on the Outdoor sequence.**

Moreover, in Fig. 4 it is possible to note that, except for very low bandwidth (remember that this is a CIF format video whose initial - uncompressed - bandwidth occupation is of about 66 Mb/s), our semantic transcoding outperforms the other methods. Example data are reported in Table 1, in which the first row reports the average PSNR given a fixed bandwidth of 256 kbps and the second row shows the bandwidth occupation given a fixed required PSNR of 35 dB.

Fig. 5 reports the same data for the indoor sequence. Note that, in this case, results are worst for our method. In particular, MPEG-4 is often more effective than our method. This is due to the minor number of moving objects present in this video. As a consequence, semantics is less used and the improvements included in MPEG-4 w.r.t. to MPEG-2 allow it to obtain a better performance. Nonetheless, the results of our method are encouraging since similar to those of MPEG-4.

Finally, we report a visual comparison of these methods on single frame of the outdoor sequence (Fig. 6). JPEG (Fig. 6(b)) degrades deeply the image, as well as JPEG-2000 (Fig. 6(c)), with the difference that JPEG-2000 does not introduce any block artifacts. It should be easy to see also that MPEG-2 (Fig. 6(d)) achieves worst performance than MPEG-4 (Fig. 6(f)) and semantic MPEG-2 (Fig. 6(e)). Between these last two algorithms the visual difference is less straightforward to be seen, but it is evident in the quantitative PSNR measure.

# 6 Conclusions

The aim of this paper was to show how the use of semantics can be exploited to better adapt video to device's capabilities and user's requirements. In other words, if a user with a PDA is particularly interested in viewing (from a camera installed in a room) people laying on the floor, we want to adapt the video to the limited screen capabilities of the PDA and meet the bandwidth constraints by preserving as much as possible of the meaningful (for the user) semantics. To do this, in previous works we proposed to use computer vision techniques to extract classes of objects in the scene, then compress them differently in accordance with the relevance that the user assigned to that class.

| | **JPEG** | **JPEG-2000** | **MPEG-2** | **Sem. MPEG-2** | **MPEG-4** |
|---|---|---|---|---|---|
| Fixed bandwidth (256 kbps) | 22 dB | 24.47 dB | 29.93 dB | **35.31 dB** | 31.55 dB |
| Fixed PSNR (35 db) | 937-78 kbps | 1150.61 kbps | 627.16 kbps | **248.24 kbps** | 490.60 kbps |

**Table 1. Numerical results on Outdoor sequence.**

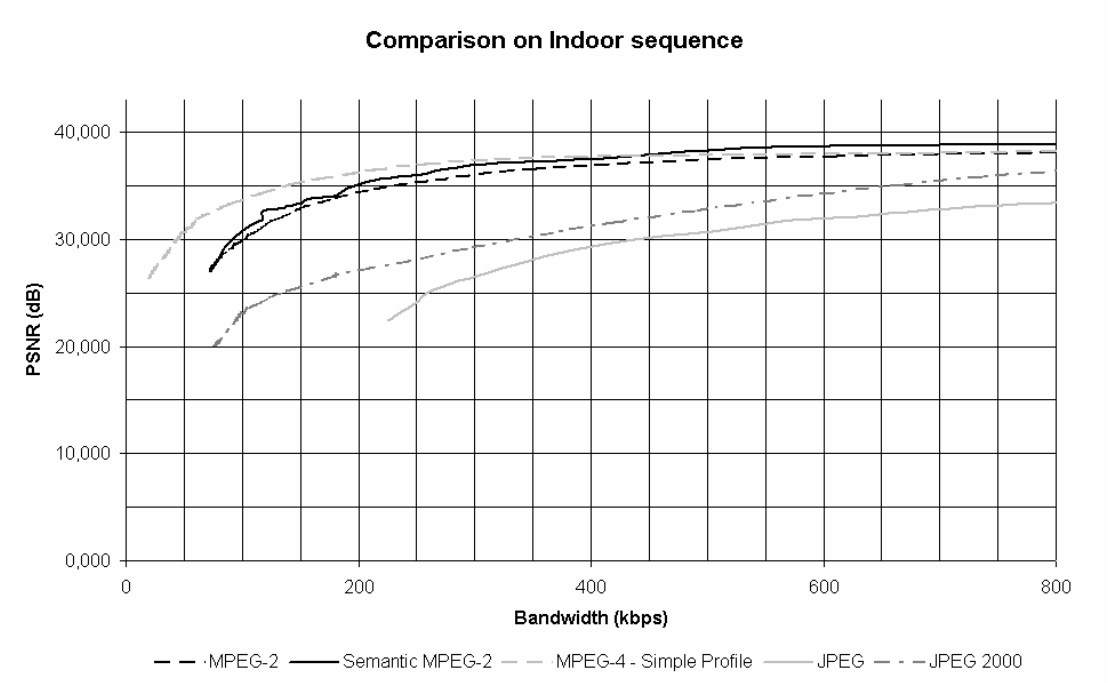
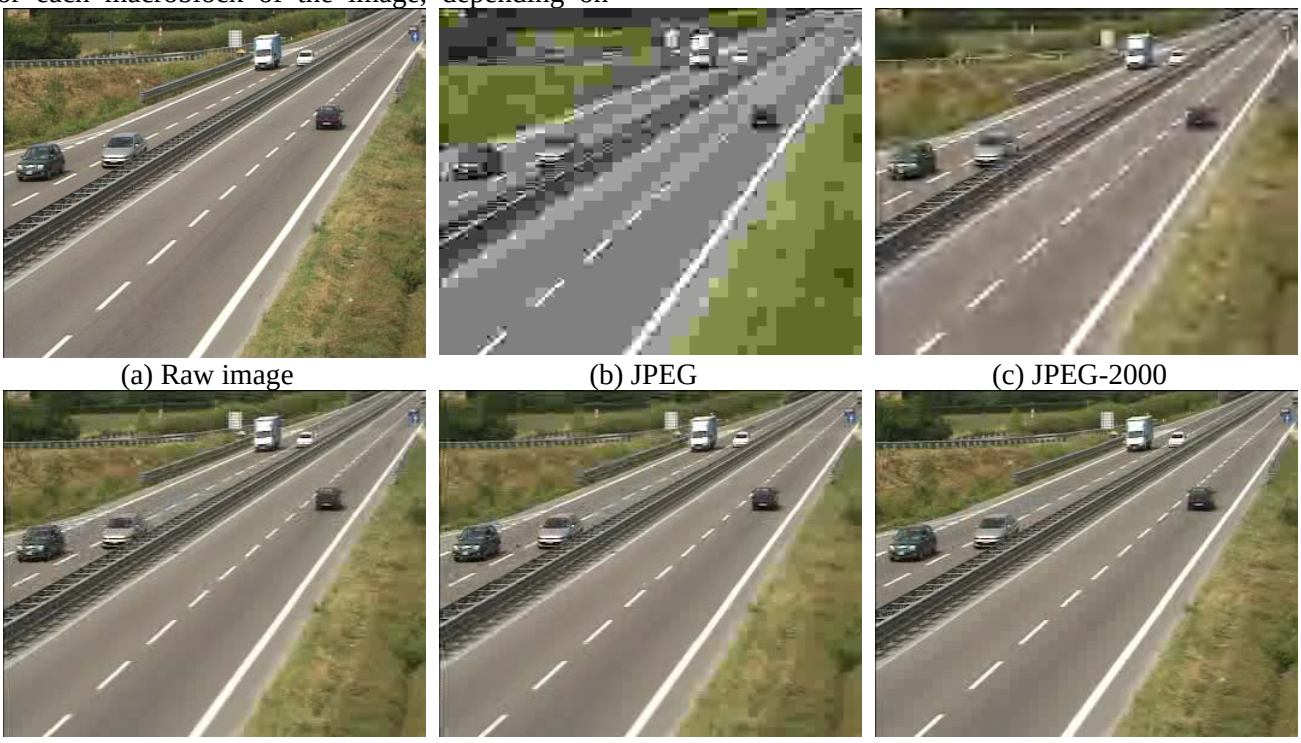



**Fig. 5. Comparison on the Indoor sequence.**

In this work, we extended our previous study by applying also temporal prediction (namely by means of MPEG standard) to the system. In particular, we exploited adaptive quantization of the MPEG standard to selectively choose the quantization values for each macroblock of the image, depending on which class is dominant in that macroblock. The results we reported demonstrate that our method outperforms standard MPEG-2 and in some cases (for some bandwidth) it even outperforms MPEG-4.



(a) Raw image (b) JPEG (c) JPEG-2000

(d) MPEG-2 (e) Semantic MPEG-2 (f) MPEG-4

**Fig. 6. Visual comparison on a single frame of the Outdoor sequence.**

## Acknowledgements

This research is founded by the national project FIRB PERF project and by the European project DELOS (contract no. G038-507618). We would like to thank them for all the financial support. We would also like to thank Emiliano Zoboli for his help in implementing the adaptive quantization.

## References

[1] Mohan, R., Smith, J.R. and Li, C., "Adapting multimedia internet content for universal access," *IEEE Transactions on Multimedia*, Vol. 1, No. 1, March 1999, pp. 104–114.

[2] Huang, A.W. and Sundaresan, N., "A semantic transcoding system to adapt web services for users with disabilities," in *Proc. of the ACM SIGCAPH Conference on Assistive Technologies*, 2000, pp. 156–163.

[3] Jayant, N., Johnston, J. and Safranek, R., "Signal compression based on models of human perception," *Proceedings of the IEEE*, Vol. 81, No. 10, October 1993, pp. 1385–1422.

[4] Smith, J.R., Mohan, R., and Li, C., "Content-based transcoding of images in the Internet," in *Proc. of IEEE Int'l Conference on Image Processing*, Vol. 3, October 1998, pp. 7–11.

[5] Yu, Y. and Chen, C.W., "SNR scalable transcoding for video over wireless channels," in *Proc. of the Wireless Communications and Networking Conference*, Vol. 3, 2000, pp. 1396–1402.

[6] Cucchiara, R., Grana, C. and Prati, A., "Semantic video transcoding using classes of relevance", *International Journal of Image and Graphics*, Vol. 3, No. 1, January 2003, pp. 145–169.

[7] Nagao, K., Shirai, Y. and Squire, K., "Semantic annotation and transcoding: Making web content more accessible," *IEEE Multimedia*, Vol. 8, No. 2, April-June 2001, pp. 69–81.

[8] Cucchiara, R., Grana, C. and Prati, A., "Semantic transcoding for live video server," in *Proc. of ACM Multimedia 2002 Conference*, Dec. 2002, pp. 223–226.

[9] Vetro, A., Sun, H. and Wang, Y., "Object-based transcoding for adaptable video content delivery", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 3, March 2001, pp. 387–401.

[10] Vetro, A., Divakaran, A., Sun, H. and Poon, T., "Adaptive transcoding system based on MPEG-7 meta-data," in *Proc. of Pacific-Rim Conference on Multimedia*, 2002.

[11] Hwang, J., Wu, T. and Lin, C., „Dynamic frame-skipping in video transcoding," in *Proc. of the IEEE Second Workshop on MultimediaSignal Processing*, Dec. 1998, pp. 616–621.

[12] Vetro, A. and Sun, H., "Encoding and transcoding multiple video-objects with variable temporal resolution," in *Proc. of Int'l Symposium of Circuit and Systems*, May 2001.

[13] ISO/IEC 15938-3:CD. *"Information Technology - Multimedia Content Description Interface. Part 3: Visual"*

[14] IBM research. *http://www.research.ibm.com/MediaStar/VideoSystem.html.*

[15] Chang, S., Anastassiou, D., Eleftheriadis, A., Meng, J., Paek, S., Pajhan, S. and Smith, J.R., "Development of advanced image/video servers in a video on demand testbed," in *Proc. of the IEEE Visual Signal Processing and Communications Workshop*, Sept. 1994.

[16] Youn, J., Sun, M. and Lin, C., "Motion vector refinement for high-performance transcoding," *IEEE Transactions on Multimedia*, Vol. 1, No. 1, March 1999, pp. 30–40.

[17] Bjork, N. and Christopoulos, c., "Video transcoding for universal multimedia access," in *Proc. of ACM Multimedia Conference*, Jan. 2000.

[18] Westerink, P.H., Rajagopalan, R. and Gonzales, C.A., "Two-pass MPEG-2 variable-bitrate encoding," *IBM Journal of Research and Developement*, Vol. 43, No. 4, July 1999, pp. 471–488.

[19] Farin, D., Ksemann, M., de With, P.H.N. and Effelsberg, W., „Rate-distortion optimal adaptive quantization and coefficient thresholding for MPEG coding," in *23rd Symposium on Information Theory in the Benelux*, May 2002, pp. 131–138.

[20] Ortega, A. and Ramchandran, K., "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to MPEG video coders," in *SPIE Digital Video Compression: Algorithms and Technologies*, Vol. 2419, February 1995, pp. 129–138.

[21] Ramchandran, K. and Vetterli, M., "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Transactions on Image Processing*, Vol. 3, No. 5, Sept. 1994, pp. 700–704.

[22] Gersho, A. and Gray, A., *Vector Quantization and Signal Compression*. Kluwer Academic Press, 1992.

[23] Yoo, Y. and Ortega, A., "Adaptive quantization without side information using SVQ and TCQ," in *29th Asilomar Conference on Signals, Systems, and Computers*, Nov. 1995.

[24] Watson, A.B., "DCT quantization matrices visually optimized for individual images," in J.P. Allebach and B.E. Rogowitz, editors, *Human Vision, Visual Processing and Digital Display IV*, Vol. 1913 of *SPIE Proceedings*, Feb. 1993, pp. 202–216.

[25] Rosenholtz, R. and Watson, A.B., "Perceptual adaptive jpeg coding," in *Proc. of IEEE Int'l Conference on Image Processing*, Vol. I, 1996, pp. 901–904.

[26] Test model 5, April 1993. ISO/IEC JTC1/SC29/WG11 N0400.

[27] Cucchiara, R., Grana, C., Neri, G., Piccardi, M. and Prati, A., "The Sakbot system for moving object detection and tracking". In *Video-based Surveillance Systems - Computer Vision and Distributed Processing*. Kluwer Academic, 2001.

[28] Cucchiara, R., Grana, C., Piccardi, M. and Prati, A., "Detecting Moving Objects, Ghosts and Shadows in Video Streams", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10, Oct. 2003, pp. 1337-1342.

[29] Cucchiara, R., Grana, C. and Prati, A., "Detecting moving objects and their shadows: an evaluation with the PETS2002 dataset," in *Proc. of Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2002)*, May 2002, pp. 18–25.

[30] Divakaran, A. and Sun, H., "A descriptor for spatial distribution of motion activity," in *Proc. of Storage and Retrieval from Image and Video Databases*, January 2000.

[31] Shapiro, J.M., "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, Dec. 1993, pp. 3445–3462.