A Framework for Semantic Video Transcoding

Rita Cucchiara, Costantino Grana, and Andrea Prati
Dipartimento di Ingegneria dell'Informazione - Unviersità di Modena e Reggio Emilia

Abstract. In this work we present a transcoding framework and an object-based technique to adapt live and stored videos to the user bandwidth and resources capabilities. Multiple transcoding policies are reviewed and a performance evaluation metric based on the *Weighted Mean Square Error* that allows different classes of relevance is presented. We present results for different transcoding policies and for different bandwidth requirements, showing that the use of semantic can improve the bandwidth to distortion ratio.

1 Introduction

Video publishing is becoming a more and more spreading reality on the Web. The main motivations for this are that videos increase the amount and the quality of the information provided with respect to still images and that they call for a more "real" interaction, with the meaning of a more similar reproduction of the way in which we typically look at a scene. Moreover, the improvement of the technology permits now to afford the huge amount of computation, bandwidth and storage required by videos.

Nevertheless, in the video accessibility through the net two important problems arise: the first is the unavailability to all the users of new technologies such as large bandwidth connections to the Internet; the second is the constant growth and diffusion of a diversity of devices to access the Internet. For example, PDAs (personal digital assistant) are devices able to combine computing and communication capabilities in small and handy equipments. Unfortunately, these devices have limited display and resolution resources and have typically a wireless network card with limited bandwidth. With these premises, video delivery and reproduction can be unaffordable tasks.

Many works are present in the literature addressing the problem of the *transcoding of information* [4, 6, 7, 10, 13]. However, the research on the field of video transcoding is quite new. Transcoding is a very popular 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 or as an *intermedia* transcoding when also the media nature changes (for instance transforming audio into text).

The final goal of transcoding remains a suitable adaptation to the client resources, maintaining an acceptable *QoS* whose model definition and performance analysis still remain open problems. This is particularly true in the case of videos where the user satisfaction should be measured in term of perceptive cues.

Video fidelity is hard to define in general, unless the application and its purposes are well known. In this framework, expertise of computer vision community in image and scene understanding could be essential in order to handle the transcoding process. 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's semantic and at the same time the transformation may change the video perception and possibly its appearance, while preserving the semantic. Several approaches have been proposed addressing this topic, mostly dealing with stored videos. They are often associated with a process of *annotation* that takes care of video content, annotated in the video database [8].

Adding the semantic to the video allows the effective scalability of the video on almost every existing device, even in the case in which the limited device capabilities require to change the medium of the information itself. For example, in video-surveillance applications you may want to access the camera installed inside your factory, but you may have only a cell phone to do it. In this case, our claim is that a semantic transcoding of the video will be able to analyze the video and to "transcode" the information of possible intruders into a textual one, visible on your cell phone's display, as in [7, 10].

In [7] a good survey of transcoding products is presented. The authors claim that there are some advantages in designing transcoding capability in multimedia servers especially because the provider keeps the control of distributed data. Moreover on-the-fly transcoding is considered hard to apply on multimedia data such as videos; therefore the authors provide a general framework, named 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 testbed [2]. An alternative solution to storing multimedia data already transcoded is to provide transcoding directly on compressed data, as for instance in [2, 12].

The goal of this work is to propose a transcoding technique able to process videos in order to adapt to the user bandwidth and resource requirements.

2 Definition of the Framework

Commercial video servers for live camera provide a user interface in which the image quality can be modified. Given the fixed compression quality the streaming process adapts the frame rate to the bandwidth. This is often unacceptable for users that need high details only in particular regions of the image; in this case users are forced to choose between frame rate and quality. For this reason, we want to propose a framework in which the transcoding is dynamically and on-the-fly adapted upon the requirements of the clients.

2.1 The transcoding policies

Transcoding has been classified in various ways: Vetro et al in [11] distinguish among *bit-rate conversion* or *scaling, resolution conversion* and *syntactic conversion*: the first copes with bandwidth limitation; the second is used for device limitation, as well as for bandwidth limitation; the third deals with syntactic conversion for protocol layer. By focusing on bit-rate scaling only, the authors propose two solutions: a conservative transcoding varying temporal and spatial quality of multimedia objects and an aggressive model that accepts dropping less relevant object on the scene.

According with [11] we accept an aggressive model if it is guided by semantic: thus in addition to some downscaling techniques used for reducing bit-rate without relationships with the video content, we propose a semantic transcoding of the video in interesting objects. Transcoding can be classified as:

- *spatial* transcoding (`spat_tr`);
- *temporal* transcoding (`temp_tr`);
- *code* transcoding (`code_tr`);
- *color* transcoding (`color_tr`);
- *object* transcoding (`object_tr`).

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. This approach allows also bandwidth reduction (since the uncompressed amount of data decreases) in most of the cases, but sometime a naive spatial downscaling could increases the file size. For instance it is reported in [1] for still images, especially focusing on the differences between codings such as JPEG and GIF.

Temporal transcoding copes with a reduction of number of frames: this is automatically provided by the streaming process that downscales the number of transferred frames. In other researches dynamic frame skipping techniques have been developed to choose when frames can be eliminated according with the changes in the motion vectors [5].

Color transcoding, like size transcoding, is sometimes requested for specific clients (like gray level PDAs). A color downscaling is automatically performed by all JPEG, MPEG standard with the 4:2:0 YUV code. 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, a too aggressive compression could be unacceptable for many applications due to the lost details.

Finally, the class *object* or *semantic* transcoding comprises some different techniques based on computer vision tasks. Basically the goal is to extract semantically valuable objects from the scene and transfer them with the lower amount of compression in order to maintain both details and speed.

2.2 The architecture

Our aim is to extract the “useful” semantic from a video sequence and to convert the information according to the device type and requirements. Since the useful semantic is strongly application-dependent, we focus on video-surveillance, or more in general motion extraction-based applications, in which the semantic we are interested in is basically the moving objects perceived in the scene. This framework will allow transcoding of video for many different types of Internet clients.

The system architecture is structured in a three layered structure as follows: a web server module will deal with the client requests and with their capabilities; a Transcoding Policy Resolver (TPR) module will choose the best transcoding policy, trying to match the requested video characteristics and the client capabilities; a Video Transcoder module will apply the selected transcoding policy. The communication between the different modules is made by means of XML coded messages to allow easy integration of new modules in the future.

The web server module uses a standard cookie-based system to identify the client so that, after a first description of the system, each request can be associated with its capabilities. The client capabilities are maximum bandwidth, display size and color or black and white display, and videos can be requested from a camera or from a video database. The module creates “Capabilities” and “Request” XML messages, that are then passed to the TPR.

The TPR module gets the “Request” message and queries the video camera properties (from an installation database) or the stored video database. The videos are characterized by frame rate, color depth and video class. Three classes of video are possible: fixed camera, PTZ with constraint motion, PTZ with free motion. In case of stored video a caching policy is also considered. The TPR then makes a match between the actual video characteristics and the client capabilities starting by fitting requested maximum size and color depth, applying an initial JPEG compression and then checking if it fits the bandwidth requirements. For this purpose we choose first to further compress the video, then to apply object-based transcoding and, if necessary, to resize and to skip frames, using a table that considers the typical requirements for a video surveillance sequence. This initial parameters are then sent to the Video Transcoder.

The Video Transcoder applies the conversion and feeds the data back to the web server for transmission. Since the compression depends on the number of objects in the scene and on their size, the output dimension is checked and used to adjust the quality factor of the JPEG compression accordingly.

3 Performance Evaluation Metrics

Evaluating the result of transcoding is not trivial and can be very dependent on the application. In theory, the best possible policy should transcode the video by sending all the *important* information with the highest quality and by reducing the bandwidth not sending (or sending with the lowest quality) the useless data. Unfortunately no robust models for computing the *value* of transcoding have been proposed (i.e. how good is the trade-off between meeting the requirements and preserving the quality of the information) [8].

If the video has no 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. We tested the transcoding policies simulating different applications. In a context of semantic video transcoding we could define “classes of relevance” in order to give a priority in the value of objects that are in the video. Thus we can associate “weights of relevance” to the classes that affect the computation of the distortion produced by transcoding. 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.

A common metric to measure the distortion/error in compressed/transcoded images is the *Peak Signal-to-Noise Ratio (PSNR)* (as in [9]), defined as:

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

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:

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

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, that is:

$$d(i,j) = \sqrt{(I_O^R(i,j) - I_D^R(i,j))^2 + (I_O^G(i,j) - I_D^G(i,j))^2 + (I_O^B(i,j) - I_D^B(i,j))^2} \tag{3}$$

being I_O the original image and I_D the distorted version of it. Consequently, V_{MAX} is equal to $\sqrt{3} \cdot 255$.

To account for different classes of object present in the scene, we introduce the $WMSE$ (Weighted MSE) as:

$$WMSE = \sum_{k=1}^{N_{CL}} w_k \cdot MSE_k \tag{4}$$

where N_{CL} is the number of classes of relevance and MSE_k can be written as:

$$MSE_k = \frac{1}{|C_k|} \sum_{(i,j) \in C_k} d^2(i,j) \tag{5}$$

where C_k is the set of the points belonging to the class k and $|C_k|$ is its cardinality. To decide whether a point (i,j) belongs to a certain class or not, we manually segment about forty frames of the video by indicating where the classes are.

The weights w_k are chosen according to the semantic and to the following rules:

$$w_k \geq 0 \quad \forall k = 1, ..., N_{CL} \quad ; \quad \sum_{i=1}^{N_{CL}} w_k = 1 \tag{6}$$

Clearly, in the absence of semantic, $WMSE \equiv MSE$.

Moreover, we use the bandwidth B expressed in Kb/s as a complementary measure. To outline the improvement introduced by the transcoding, the *bandwidth enhancement* ($\frac{B_O}{B_D}$) is also reported.

4 Results Analysis

We compare, both in terms of PSNR and bandwidth, the five types of transcoding reported in Section 2. In the case of spatial transcoding we reduced the size of the image from CIF (352x288) to QCIF (176x144). The temporal transcoding consists in taking one frame each seven (this number has been chosen in order to obtain a bandwidth similar to the other methods). As code transcoding we limited our study to the JPEG compression (with two different compression values). Further work will include more coding methods.

The novelty of this work is the use of object (or *semantic-based*) transcoding with classes of relevance. As above mentioned, we used our own segmenting algorithm to extract the Visual Objects (with the characteristic to be different from a statistical and constantly updated background). As transcoding we tested the performance when sending only these VOs (with the rest of the image in black), or when the VOs are super-imposed to one background frame. We consider four case studies: one application without semantic, one with two classes (people and background) and two with three classes (face and body of the people, and background). In the latter cases, we simulated a surveillance application (in which faces are the most important information) and a “landscape-view” application (in which background is the essential information to be transmitted).

Table 1: Weighted Mean Square Error ($WMSE$) for the transcoding policies with different weights. All the transcoded versions except for the binary image have been compressed with JPEG (C=20). In the case of binary images (1-bit images) the JPEG compression (that does not allow 1-bit compression) results in worst bandwidth performance. The numbers in brackets are the Peak Signal-to-Noise Ratio (PSNR).

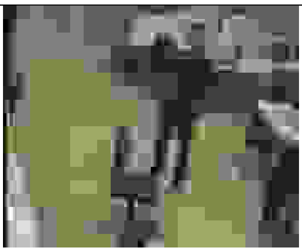
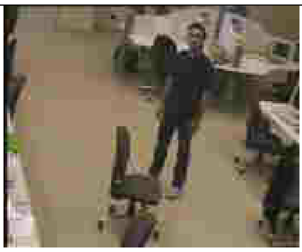
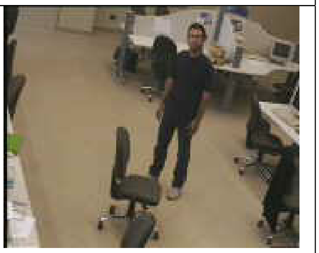
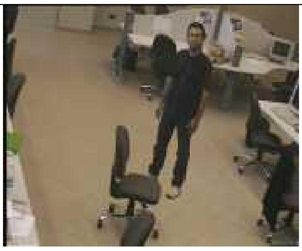
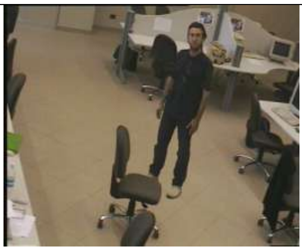
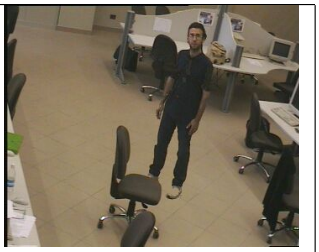
	spat_tr	temp_tr	code_tr		color_tr		object_tr	
Parameters	**Resize CIF-QCIF**	**1 frame each 7**	**JPEG (C=20)**	**JPEG (C=80)**	**Grayscale Image**	**Binary Image**	**Visual Objects**	**VO + Backgr.**
w/o semantic (*PSNR* in dB)	285,53 (28,35)	596,07 (25,15)	49,33 (35,97)	176,50 (30,43)	757,83 (24,11)	34233,58 (7,56)	38703,63 (7,02)	133,28 (31,65)
P/B [0,9 0,1] (*PSNR* in dB)	372,68 (27,19)	3695,31 (17,23)	57,84 (35,28)	219,79 (29,48)	281,79 (28,40)	13385,98 (11,64)	5914,18 (15,18)	120,62 (32,09)
F/P/B [0,8 0,2 0] (*PSNR* in dB)	711,27 (24,38)	7559,82 (14,12)	93,71 (33,18)	378,78 (27,12)	702,56 (24,44)	25351,15 (8,86)	1389,93 (21,47)	139,59 (31,45)
F/P/B [0,1 0,1 0,8] (*PSNR* in dB)	336,93 (27,63)	1468,89 (21,23)	54,68 (35,52)	200,95 (29,87)	742,24 (24,20)	32796,13 (7,74)	33716,30 (7,62)	133,77 (31,64)

Table 2: Bandwidth requirement in Kb/s and enhancement introduced by the various transcoding policies.

		spat_tr	temp_tr	code_tr		color_tr		object_tr	
	Original video	**Resize CIF-QCIF**	**1 frame each 7**	**JPEG (C=20)**	**JPEG (C=80)**	**Grayscale Image**	**Binary Image**	**Visual Objects**	**VO + Backgr.**
Bandw. (Kb/s)	23762,33	438,14	185,50	1282,89	440,78	1228,34	992,26	243,86	287,55
Bandw. enhanc.	1	54,23	128,10	18,52	53,91	19,35	23,95	97,44	82,64

Table 1 reports the value of $WMSE$ and $PSNR$ for the different transcoding policies for the four cases. The values reported in the first column between square brackets are the weights w_k applied to the different classes (P=people, F=faces, B=background). As foreseeable, the best $PSNR$ (that is the lowest distortion, measured as $WMSE$) is achieved by using code transcoding. Nevertheless, also object transcoding preserves data from distortion. Obviously, in this type of transcoding the $PSNR$ depends on the weights. This is well shown in the column entitled ”Visual Objects” in which the performance is low in the case of no semantic or background’s relevance. In fact, in this case the error introduced by sending black pixels

Table 3: Visual comparison of JPEG Compr. vs. Vo+Backgr. policy. In this case a constant bit rate (CBR) has been imposed by considering three standard bit rate (56, 128 and 256 Kbps). Additional transcoding policies have been used to be able to fit the bandwidth requirements. Also the $PSNR$ in the case of no semantic is reported. Note that in the case of JPEG Compr. video at 56 Kbps the actual bandwidth obtained is 72 Kbps. Temporal downscaling is mandatory in this case.

	56 Kb/s	**128 Kb/s**	**256 Kb/s**
JPEG Compr.			
Transcoding policy	Resize QCIF	Resize QCIF	Resize QCIF
PSNR w/o semantic	19,74	25,27	27,30
VO+Backgr.			
Transcoding policy	Resize QCIF	Resize 264x216	none
PSNR w/o semantic	26,63	28,91	31,65

as background can be mitigated only if the background is not relevant. In the case that a static background is sent together with the VOs, though it is not always correct (we moved a chair in the scene to change the background), the $PSNR$ reaches values close to the one of code transcoding.

Table 2 shows the bandwidth occupation of the original video and of those transcoded. It is possible to note that the best bandwidth enhancement is obtained by performing temporal transcoding, but this degrades heavily the data (25 dB of $PSNR$ in the best case). The proposed transcoding of VOs plus background can reduce the bandwidth from 23 Mb/s to 287 Kb/s by maintaining most of the information.

Transmitting only objects with a given class of relevance can be useful both for reducing bandwidth and limiting the distortion. This improvement will be more valuable with a transcoding with objects and MPEG4.

In addition, we simulated the behaviour of the framework described in Subsection 2.2 by considering three standard bandwidth and comparing our method (VOs+Backgr.) with normal coding compression. Results are visually presented for comparison in Table 3. Note that with JPEG compression the $PSNR$ is lower than in the case of our method. Moreover, even reducing the size from QCIF to CIF and using the greater compression available, JPEG in not able to meet the 56 Kb/s requirement. Finally, a result worthy of note is that our method can operate at full resolution when the bandwidth is 256 Kb/s or greater.

5 Conclusions

In this paper we reported a framework for live video transcoding. We classified existing transcoding policies and compared them with a novel object-based transcoding.

We proposed a performance evaluation metric based on the $PSNR$ and that takes into account different weights in accordance with the semantic of the scene and the relevance of the classes of the objects for that application. By using this metric we demonstrated that our method, based on the transmission of a first, static background image on which are superimposed the Visual Objects extracted by our motion segmentation scheme [3], is able to maximize the bandwidth reduction and to minimize the error introduced.

References

[1] Surendar Chandra, Ashish Gehani, Carla Schlatter Ellis, and Amin Vahdat. Transcoding characteristics of web images. In *Proceedings of the SPIE Multimedia Computing and Networking Conference*, January 2001.

[2] Shih-Fu Chang, Dimitris Anastassiou, Alexandros Eleftheriadis, Jianhao Meng, Seungyup Paek, Sassan Pajhan, and John R. Smith. Development of advanced image/video servers in a video on demand testbed. In *Proceedings of the IEEE Visual Signal Processing and Communications Workshop*, September 1994.

[3] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati. *The Sakbot system for moving object detection and tracking*, chapter 12. Kluwer Academic, 2001.

[4] A.W. Huang and N. Sundaresan. A semantic transcoding system to adapt web services for users with disabilities. In *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies*, pages 156–163, 2000.

[5] Jenq-Neng Hwang, Tzong-Der Wu, and Chia-Wen Lin. Dynamic frame-skipping in video transcoding. In *Proceedings of the IEEE Second Workshop on Multimedia Signal Processing*, pages 616–621, December 1998.

[6] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1385–1422, October 1993.

[7] Rakesh Mohan, John R. Smith, and Shung-Sheng Li. Adapting multimedia internat content for universal access. *IEEE Transactions on Multimedia*, 1(1):104–114, March 1999.

[8] Katashi Nagao, Yoshinari Shirai, and Kevin Squire. Semantic annotation and transcoding: Making web content more accessible. *IEEE Multimedia*, 8(2):69–81, April-June 2001.

[9] Jerome M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[10] John R. Smith, Rakesh Mohan, and Chung-Sheng Li. Content-based transcoding of images in the internet. In *Proceedings of IEEE Int'l Conference on Image Processing*, volume 3, pages 7–11, October 1998.

[11] Anthony Vetro, Huifang Sun, and Yao Wang. Object-based transcoding for adaptable video content delivery. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):387–401, March 2001.

[12] Jeongnam Youn, Ming-Ting Sun, and Chia-Wen Lin. Motion vector refinement for high-performance transcoding. *IEEE Transactions on Multimedia*, 1(1):30–40, March 1999.

[13] Y. Yu and Chen C.W. Snr scalable transcoding for video over wireless channels. In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1396–1402, 2000.