Semantic Transcoding for Live Video Server

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

Costantino Grana
Dipartimento di Ingegneria dell'Informazione,
Univ. di Modena e Reggio Emilia
Via Vignolese, 905 - 41100 Modena - Italy
grana.costantino@unimo.it

Andrea Prati
Dipartimento di Ingegneria dell'Informazione,
Univ. di Modena e Reggio Emilia
Via Vignolese, 905 - 41100 Modena - Italy
prati.andrea@unimo.it

ABSTRACT

In this paper we present transcoding techniques for a video server architecture that enables the user to access live video streams by using different devices with different capabilities. For live videos, annotation methods cannot be exploited. Instead we propose methods of on-the-fly transcoding that adapt the video content with respect to the user resources and the video semantic. Thus we propose an object-based transcoding with "classes of relevance" (for instance People, Face and Background). To compare the different strategies we propose a metric based on the *Weighted Mean Square Error* that allows the analysis of different application scenarios by means of a class-wise distortion measure. The obtained results show that the use of semantic can improve the bandwidth to distortion ratio significantly.

Keywords

Transcoding, motion segmentation, PSNR, performance evaluation metric

1. INTRODUCTION

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).

In this context, we assist to a tremendous variety of multimedia contents, and, at the same time, to a different panorama of possible clients characterized by specific network bandwidth constraint and display, processing and storage capabilities. Transcoding is consequently an important step to adapt the multimedia content to user requirements: the major effort is devoted to a bandwidth reduction, especially for new terminal types (PDAs, HCCs, smart phones, ...).

The final goal 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, but 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 [6].

In [5] a good survey of '99 transcoding products is presented. The authors claim that there are some advantages in design transcoding capability on the multimedia server 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, 9].

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

2. VIDEO TRANSCODING

We consider video transcoding only, i.e. the process to change the video format having a video both for input and output. Moreover we assume that clients are only interested in *video streaming*, to view the content while the data is downloaded.

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 zones of the image, that are forced to chose between frame rate and quality.

Transcoding has been classified in various ways: Vetro et al in [8] 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 [8] 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, PDA format 96x96 or thumbnails 64x64). 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 chose when frames can be eliminated according with the changes in the motion vectors [4].

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. These methods are not general since they refer to the type of the video. We do not want to classify them according to the application, but instead according to the source type. In fact we can classify live videos depending on the source, that could be: a) fixed camera; b) moving camera.

Fixed cameras are typical cameras for tourist application, entertainment, traffic control, basic video surveillance systems, tele-medicine, where the background can be assumed to be fixed and with a null motion. Therefore different approaches can be provided for extracting the Visual Objects (VOs thereafter) from a basic fixed background suppression (if a-priori known) to more sophisticated approaches. In [3] we defined the Sakbot approach to segment VOs and discriminate them from shadows and other "ghosts". Moving cameras can be either PTSZ cameras with a constrained motion (as used in video-surveillance) or randomly moving ones.

Depending on this classification, different computer vision techniques can be exploited to extract semantically valuable objects. For the test reported in this paper we will focus only on live videos from fixed cameras: in this case objects of interest are objects that are moving, or that have been moving in the past or part of them.

3. 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) [6].

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 [7]), defined as $PSNR = 10 \log_{10} \left(\frac{V_{MAX}^2}{MSE} \right)$,

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:

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

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:

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

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:

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

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

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

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 such that $w_k \geq 0, \forall k = 1, ..., N_{CL}$ and $\sum_{i=1}^{N_{CL}} w_k = 1$. 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.

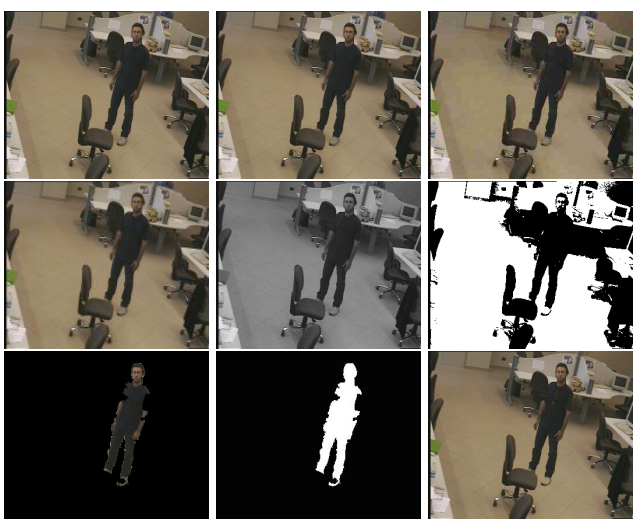

Figure 1: Visual comparison of the transcoded versions. From the upper left to the lower right: original, JPEG (C=20), JPEG (C=80), resize, gray level, binary, VO, alpha planes, VO+background.

4. EXPERIMENTAL RESULTS

4.1 The reference architecture

A video server is a system devoted to video management, mainly for Web applications. Basically, it is equipped with standard modules: I) a Web server; II) a video database management system (V-DBMS) for stored videos; III) video-on-demand, streaming, downloading services supports. Our prototype of the reference architecture of the live video server with semantic transcoding, called ImageLab Video Server (`ImaViS`) adds two other modules: IV) a network camera interface; V) a semantic on-the-fly transcoding module that can work both on stored videos and on live videos coming in streaming fashion from a camera through the network camera server. In our protorype we implemented the defined types of transcoding. Ideally the IV) module should work in real time in order not to introduce delay. In our tests we stand for a streaming from the network camera with 10 fps (with a JPEG compressed format) and our transcoding program can work at this frame rate.

4.2 Analysis of the Results

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).

We tested three levels of object transcoding: a) only VOs in their original texture and colors; b) the VO silhouettes (similarly to the alpha planes of MPEG4); c) the VOs with an initial static super-imposed background. As transcoding we chose to send either only these VOs (with the rest of the image in black), or the VOs in white (similarly to the alpha planes of MPEG 4), or only one frame as background with super-imposed the VOs (Fig. 1). 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 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 by 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

Parameters	spat_tr	temp_tr	code_tr		color_tr		object_tr		
	Resize CIF-QCIF	**1 frame each 7**	**JPEG (C=20)**	**JPEG (C=80)**	**Grayscale Image**	**Binary Image**	**Visual Objects**	**Alpha Planes**	**VO + Backgr.**
w/o semantic	285,53	596,07	49,33	176,50	757,83	34233,58	38703,63	45487,01	133,28
(*PSNR* in dB)	(28,35)	(25,15)	(35,97)	(30,43)	(24,11)	(7,56)	(7,02)	(6,29)	(31,65)
P/B [0,9 0,1]	372,68	3695,31	57,84	219,79	281,79	13385,98	5914,18	109719,71	120,62
(*PSNR* in dB)	(27,19)	(17,23)	(35,28)	(29,48)	(28,40)	(11,64)	(15,18)	(2,50)	(32,09)
F/P/B [0,8 0,2 0]	711,27	7559,82	93,71	378,78	702,56	25351,15	1389,93	100348,87	139,59
(*PSNR* in dB)	(24,38)	(14,12)	(33,18)	(27,12)	(24,44)	(8,86)	(21,47)	(2,89)	(31,45)
F/P/B [0,1 0,1 0,8]	336,93	1468,89	54,68	200,95	742,24	32796,13	33716,30	53438,28	133,77
(*PSNR* in dB)	(27,63)	(21,23)	(35,52)	(29,87)	(24,20)	(7,74)	(7,62)	(5,62)	(31,64)

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		
	Original video	**Resize CIF-QCIF**	**1 frame each 7**	**JPEG (C=20)**	**JPEG (C=80)**	**Grayscale Image**	**Binary Image**	**Visual Objects**	**Alpha Planes**	**VO + Backgr.**
Bandw. (Kb/s)	23762,33	438,14	185,50	1282,89	440,78	1228,34	992,26	243,86	261,42	287,55
Bandw. enhanc.	1	54,23	128,10	18,52	53,91	19,35	23,95	97,44	90,90	82,64

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

introduced by sending black pixels 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.

5. CONCLUSIONS

In this paper we reported preliminary experimental results of our transcoding study that aims at proposing a novel object-based transcoding able to maximize the bandwidth reduction and to minimize the error introduced.

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 super-imposed the Visual Objects extracted by our motion segmentation scheme [3], fits both the requirements of bandwidth reduction and high quality of the data.

6. 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] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.