ELSEVIER

Signal Processing: *Image Communication* 7 (1995) 231–248

SIGNAL PROCESSING: IMAGE COMMUNICATION

Automatic face location detection and tracking for model-assisted coding of video teleconferencing sequences at low bit-rates

Alexandros Eleftheriadis[a,*,1], Arnaud Jacquin[b]

[a]*Electrical Engineering Department, Columbia University, New York, NY, USA*
[b]*Signal Processing Research Department, AT&T Bell Laboratories, Murray Hill, NJ, USA*

Received 15 December 1993

Abstract

We present a novel and practical way to integrate techniques from computer vision to low bit-rate coding systems for video teleconferencing applications. Our focus is to locate and track the faces of persons in typical head-and-shoulders video sequences, and to exploit the face location information in a 'classical' video coding/decoding system. The motivation is to enable the system to selectively encode various image areas and to produce psychologically pleasing coded images where faces are sharper. We refer to this approach as model-assisted coding. We propose a totally automatic, low-complexity algorithm, which robustly performs face detection and tracking. A priori assumptions regarding sequence content are minimal and the algorithm operates accurately even in cases of partial occlusion by moving objects. Face location information is exploited by a low bit-rate 3D subband-based video coder which uses both a novel model-assisted pixel-based motion compensation scheme, as well as model-assisted dynamic bit allocation with object-selective quantization. By transferring a small fraction of the total available bit-rate from the non-facial to the facial area, the coder produces images with better-rendered facial features. The improvement was found to be perceptually significant on video sequences coded at 96 kbps for an input luminance signal in CIF format. The technique is applicable to any video coding scheme that allows for fine-grain quantizer selection (e.g. MPEG, H.261), and can maintain full decoder compatibility.

Keywords: Face tracking; Model-based coding; Teleconferencing; Video coding

1. Introduction

In very low bit-rate video teleconferencing situations, state-of-the-art coding algorithms produce artifacts which are systematically present throughout coded images; all the more as the image content in terms of motion and texture is high. These

*Corresponding author. Mailing Address: Room 2D-337, AT&T Bell Laboratories, 600 Mountain Avenue, P.O. Box 636, Murray Hill, NJ 07974-0636, USA

[1] This work was performed while the author was a University Relations Intern in the Signal Processing Research Department at AT&T Bell Laboratories, Murray Hill, NJ, USA, June August 1993.

SSDI 0923-5965(95)00028-3

artifacts usually affect all areas of the image without discrimination. Viewers, however, will mostly find coding artifacts to be more noticeable in areas of particular interest to them. In particular, a user of a video teleconferencing system or video telephone will typically focus his or her attention to the *face*(*s*) of the person(s) on the screen, rather than to areas such as clothing or background. Besides, although fast motion is known to mask coding artifacts, the human visual system has the ability to *lock on* and *track* particular moving objects, such as a person's face. Communication between users of very low bit-rate video teleconferencing systems or video phones will be intelligible and pleasing only when facial features are not plagued with an excessive amount of coding artifacts.[2]

The motivation of this work was to investigate the possibility to detect and track specific moving objects known a priori to be present in a video sequence, and to enable a video coding system to use this information in order to discriminatively encode different areas in typical 'head-and-shoulders' video sequences – an idea which has been proposed in [28, 39, 30], in which, however, only parts of the problem are addressed. The coder would, for example,

- Encode facial features (such as eyes, mouth, nose, etc.) very accurately.
- Encode less accurately the rest of the picture, be it moving or still.

This requires that the coder first detects and models face locations, then exploits this information to achieve *model-assisted coding*. The location detection algorithm should be of fairly low complexity; in addition, if transmission of the model parameters is required, the overhead bit-rate should be minimized. In [39], Ueno et al. propose to use face location information in a preprocessing stage which consists of low-pass filtering the image area outside the face location.[3]

In this work, we show how to exploit and integrate in a novel way techniques derived from *computer vision* (scene analysis, geometric modeling, object recognition) for low bit-rate 3D subband-based video coding. The coding system used operates at 128 kbps, with an input digital color video signal in YUV format, and with a coding rate of 96 kbps for the luminance signal. The video data consists of 'head-and-shoulders' sequences. We describe an automatic face location detection and tracking algorithm which models face contours as ellipses. We also describe two ways to exploit the face location information through *model-assisted motion compensation*, and *model-assisted dynamic bit allocation*. In the former technique, a motion vector field for pixels inside the face region is *automatically* computed from the relative positions of facial models in successive frames. By transmitting only the model parameters, the motion vector field can be easily regenerated at the decoder without any additional motion information needed. The latter technique uses two quantizers per subband: a fine one used for data inside the face location model, and a coarse one used for data outside this region.

Even though the work reported here focuses on 3D subband-based video coding algorithms, face location information can be used for similar discriminative quantization strategies in other video coding algorithms. In particular, and if one dispenses with the model-assisted motion compensation scheme which requires transmission of model parameters to the decoder, any coding scheme that allows selection of quantization parameters at a fine scale can be accomodated with full decoder compatibility (e.g. MPEG [3], H.261 [12, 29], in which quantizers are selectable down to the macroblock level). Using facial area locations for region-selective quantization in the context of a CCITT-compatible low bit-rate codec was initially proposed by Badiqué in [4].

The organization of this paper is the following. In Section 2, we briefly review the concept of model-based video coding, and define our model-assisted coding approach. In Section 3, we describe the model adopted for the representation of face location information and the integration of face location information in a low bit-rate 3D subband-based video coding system. In Section 4, we

[2] In some situations, a very good rendition of facial features is paramount to intelligibility. The case of hearing-impaired viewers who would mostly rely on lip reading is one such example.

[3] The authors do not, however, disclose how the face location detection is performed.

describe the automatic face location detection and tracking algorithm, and illustrate the quality improvement in image sequences coded with a model-assisted coder.

2. Model-based and model-assisted video coding

It is widely agreed upon that 'classical' (i.e. purely waveform-based) coding techniques alone may not be sufficient for high-quality coding of digital signals at very low bit-rates (e.g. 128 kbps and below for a color video signal [41, 31, 23]. Thus, *model-based* approaches to very low bit-rate coding of video, also referred to as *knowledge-based coding*, *semantic coding*, or *analysis–synthesis coding*, have been receiving a great deal of attention [26, 16, 7, 2, 1, 33, 34, 8, 40, 6, 27, 21, 32, 20, 11]. For a detailed overview of state-of-the-art model-based video coding techniques, the reader is referred to [7, 1, 13].

The principle of a fairly generic model-based coding system is illustrated in Fig. 1. Each input video frame to the encoder is analyzed, and a geometric model of the data is constructed (this model being either three-dimensional and fitted[4] to the data [2, 34], or two-dimensional and obtained from segmentation of the input image into *objects* represented by models of the 'contour-texture' type [21, 32, 20, 11]. The parameters of the model are transmitted on the channel along with an appropriately coded error signal. The latter is necessary in order to mitigate quality loss in regions of the image – typically complex, highly detailed ones – where the model does not give a sufficiently good fit, or simply 'fails.'

The signal is reconstructed (synthesized) at the receiving end from the model parameters and the decoded error signal. Since the bit-rate required to transmit the model parameters can be extremely low, very low coding rates can be achieved for very specific scenes, usually fairly low in texture and motion content. This approach, however, apart from its inherently very high complexity, suffers also from a lack of flexibility: the models are usually tailored to a specific sequence content. Whenever the input video data differs substantially from what can be modeled by the encoder, a model breakdown will occur with serious consequences for the coded signal.

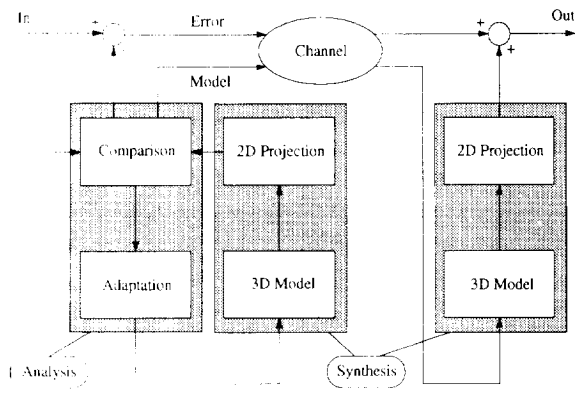

Fig. 1. A generic model-based video coding system.

Rather than relying on ubiquitous data models for head-and-shoulders video sequences, our approach has been to only *partially model* the data, i.e. model the *location* of specific objects known a priori to be present in the scene, and integrate this partial model to a 'classical' video coding system. For the purposes of very low bit-rate coding of video teleconferencing scenes, where typically one person is shown from the waist up moving in front of a still background, we propose to model the location of the face[5] of the person present in the scene, rather than model the face itself.

This location model, which is obtained automatically and reliably, is used to *improve* in an area-selective fashion the image quality given by a classical video coder. In effect, the coder is assigned to transfer a relatively small fraction of the available bit-rate from the coding of the non-facial area[6] to

[4] The automatic fitting (i.e. not involving any human interaction) of models – such as the wireframe models of Harashima et al. [2, 34], in real-time, to video data is far from being a solved problem.

[5] Throughout this paper, the term 'face location' is slightly abused for the sake of simplicity. This term is meant to encompass the case of people turning their head to their left or right – thereby appearing in a profile, or even turning their back to the camera, where 'face location' should read 'location of the head outline'.

[6] The image area surrounding the face location.

that of the facial area, thereby providing images with sharper facial features. Note that in cases where the a priori assumptions with respect to the source content are not satisfied (model breakdown), the classical video coder can be used as 'fall-back' coding mode. We refer to this approach as *model-assisted* video coding, in order to distinguish it from *model-based* coding which relies more heavily on the data model. The benefits of our approach are (i) it guarantees an acceptable lower bound in coding quality since it relies on a good fall-back mode, (ii) the coding of images of faces of people with glasses and/or facial hair presents no additional difficulty as it does in the case of wireframe model-based methods, (iii) it is compatible with existing decoders, and (iv) its requirements in terms of model-fitting accuracy are significantly reduced. In what follows we concentrate on a specific type of video data (head-and-shoulders sequences), partial models (face locations), and fall-back video coders (3D subband based), with a global coding rate of 96 kbps for a luminance signal in CIF format. Despite the specificity of this framework, however, the concept is quite general. It could be used in the context of other video coders working at other rates, and the object tracking algorithms could also be redesigned for different applications where objects other than faces are of interest.

3. Using face location information for model-assisted video coding

In Section 3.1 we describe the model adopted for the representation of face location information. Manually derived location information can be used to both benchmark any automatic detection algorithm, as well as provide an upper bound to the effectiveness of our model-assisted approach in improving perceptual image quality. Then, in Section 3.2, we discuss in detail the way this information is utilized in a subband-based video coding scheme.

3.1. Face location modeling

The model we adopted in order to represent the location of a face was simply that of an ellipse, as shown in Fig. 2. Although the upper (hair) and lower (chin) areas in actual face outlines can have quite different curvatures, ellipses provide a good trade-off between model accuracy and parametric simplicitly. Moreover, due to the fact that this information is not actually used to regenerate the face outline, a small lack of model-fitting accuracy does not have any significant impact in the overall performance of the coding process.

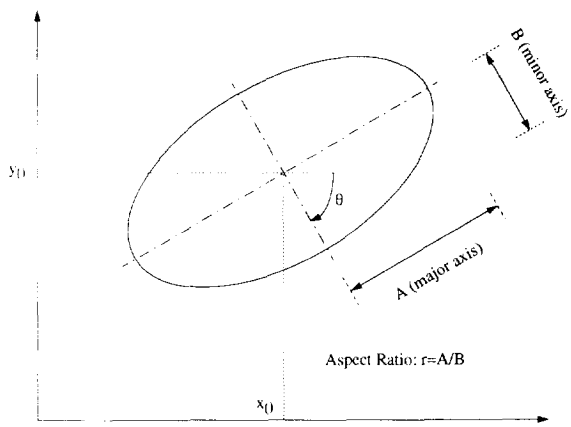

Fig. 2. Elliptical face location model.

An ellipse of arbitrary size and 'tilt' can be represented by the following quadratic, non-parametric equation (implicit form) [5]:

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0,$$
$$b^2 - ac < 0. \tag{1}$$

The negative value of the discriminant $D = b^2 - ac$ is a necessary condition, as other values are associated with different quadratic curves. The parameters of this implicit form are made available to the encoder, to be used as described in the following section. In Fig. 10, we show manually traced outlines of faces in still frames from the video sequences 'jelena' and 'jim'.

3.2. Model-assisted 3D subband-based video coding

The face location information for successive frames of a head-and-shoulders video teleconferencing sequence can be utilized in two different

components of a 3D subband-based video teleconferencing system, such as the one described in [24]. Firstly, it is used to devise a novel *model-assisted, pixel-based motion compensation* scheme in the spatio-temporal baseband which involves no transmission of motion vectors, and which is compatible with conditional replenishment. Secondly, it is used to enable the dynamic bit allocator (DBA) of the encoder to selectively use two different quantizers Q_i and $Q_e - Q_i$ being finer than Q_e - in the two areas of the subband signals delimited by an elliptical face outline. Q_i is used in the interior region of the ellipse, whereas Q_e is used in the exterior one.

3.2.1. *Low bit-rate 3D subband-based coding of digital video with a dynamic bit allocation*

We briefly review the structure of the 3D subband-based video teleconferencing system described in [24], functioning at the rate of 128 kbps with the luminance signal encoded at 96 kbps. The input luminance signal (in CIF format, consisting of images of size 360×240 pixels and temporally subsampled at 7.5 fps), is decomposed in a separable fashion into seventeen spatio-temporal subbands organized according to Fig. 3. Sample pairs of subband frames for the sequences referred to as 'jelena', and 'jim', are shown in Fig. 5.

Each pair of low-pass temporal (LPT), high-pass temporal (HPT) subband frames is allocated a fixed number of bits which is given by the global coding rate. These bits are dynamically allocated to the various subbands according to an *encoding priority list* shown in Fig. 4(a). For any given pair of subband frames, the *dynamic bit allocator* (DBA) first orders the subband data blocks[7] which cannot be repeated from the previous pair in a list of blocks with decreasing mean-square energy. The dynamic bit allocator may run out of bits at any point in the list, as the signal content of the various subbands depends on the nature of the original input sequence (close-up, far-away shot, more than one person in scene, presence of textures, motion, etc.). Whenever the bit allocator runs out of bits within

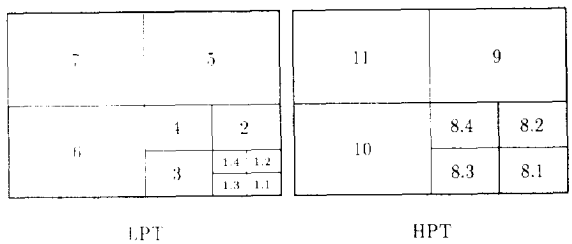

Fig. 3. 3D subband decomposition of a video signal (17-subband scheme for 128 kbps coding). Left: Low-pass temporal subsequence. Right: High-pass temporal subsequence.

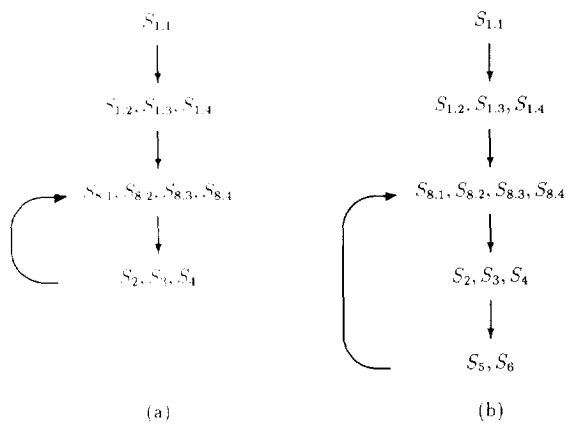

Fig. 4. Encoding priority lists. Left: No model-assisted DBA. Right: With model-assisted DBA.

a subband cluster, the blocks with the highest mean-square energy are coded; the remaining blocks are discarded. The 'feedback loop' in Fig. 4(a) indicates that in cases where bits are left over after the encoding of the subband cluster $\{S_2, S_3, S_4\}$, these bits can be used to encode more data in a particular subband cluster such as the 'motion subbands' $\{S_{8,1}, S_{8,2}, S_{8,3}, S_{8,4}\}$, resulting in a bit allocation with two passes through the data.

The various quantizers used to code the subband data on a pixel or block basis are described in [24, 37]. The quantization strategy is recalled in Table 1. The use of *conditional replenishment* (CR) and zeroing of low-energy subband data blocks implies the generation of side information which specifies for each pixel or block in a non-discarded

[7] This is done for every subband except $S_{1,1}$, which is encoded in a pixel-based fashion. The blocks are of size 4×4.

subband whether it is (i) repeated from the same spatial location in the previous subband frame pair, (ii) coded, or (iii) zeroed-out. Fig. 8 shows a template image for the storage of the side information arising from quantization.

3.2.2. *Model-assisted pixel-based motion compensation*

In [24], the encoding of subband $S_{1,1}$ was performed on a pixel basis, with use of conditional replenishment in order to repeat still background from one subband to the next at a low bit-rate. The pixels which could not be repeated were *replenished*, and quantized with 5-bit PCM. The coding algorithm is simply:

$$\hat{x}_t(i,j) = \begin{cases} \hat{x}_{t-1}(i,j) & \text{if } |x_t(i,j) - x_{t-1}(i,j)| \leqslant T_{cr}, \\ Q\{x_t(i,j)\} & \text{otherwise.} \end{cases} \quad (2)$$

where $x_t(i,j)$ denotes the value of the pixel $p_t(i,j)$ in the ith row, jth column in subband $S_{1,1}$ at instant t.

Table 1
Quantization strategy for 3D subband coding with DBA at 96 kbps

Subbands	Quantization	Bit-rate
$S_{1,1}$	5-bit PCM	5 bpp
$S_{1,2}, S_{1,3}, S_{1,4}$	4-level GVQ	2.5 bpp
$S_{8,1}, S_{8,2}, S_{8,3}, S_{8,4}$	3-level GVQ	1.9 bpp
S_2, S_3, S_4	3-level GVQ	1.9 bpp
S_5, S_6	zeroing	0 bpp

$\hat{x}_t(i,j)$ is the quantized pixel value, and $Q\{\cdot\}$ denotes PCM quantization. The scalar threshold T_{cr} threshold is empirically derived.

The availability of face location models for consecutive subband frames makes it possible to perform a type of pixel-based motion compensation which supplements – and is compatible with – the above scheme. In cases where the orientation of the person's head does not change too much from one pair of subband frames to the next, we may assume

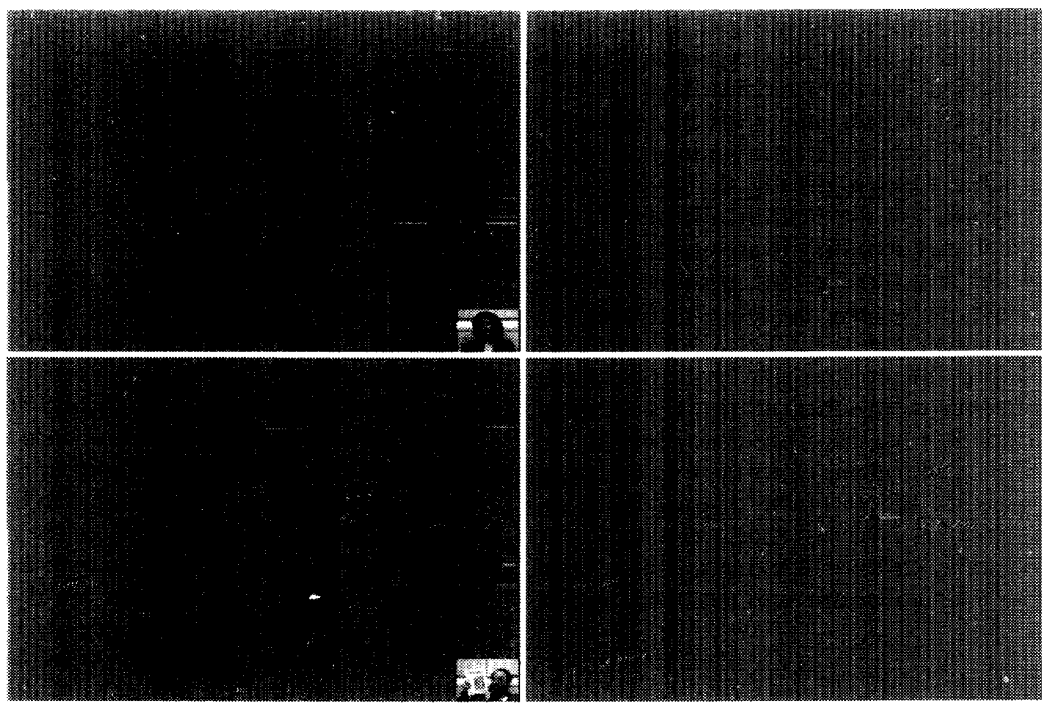

Fig. 5. Pairs of subband frames for sequences 'jelena' and 'jim'.

that the location of facial features can be traced back to the previous pair. The use of more traditional block-based motion compensation schemes in a subband coding context is currently the subject of active research [17, 10, 38].

Let $\mathcal{C}_{t-1}$ and $\mathcal{C}_t$ denote the ellipse contours which are good approximations of face locations in two consecutive subbands S_{t-1}. A two-dimensional affine mapping from one contour to the other is unambiguously[8] defined by mapping the major and minor axes of the ellipses onto one another. Let τ_t indicate this mapping from $\mathcal{C}_t$ *to* $\mathcal{C}_{t-1}$. The application of the mapping to each pixel inside the ellipse contour $\mathcal{C}_t$, generates a pixel-based (affine) motion field which will in general outperform the simple conditional replenishment strategy described above, provided that the ellipses fit reasonably tightly and consistently to the actual face outlines. This idea is illustrated in Fig. 6. The coding algorithm now becomes:

If $p_t(i,j)$ is inside $\mathcal{C}_t$

compute the motion vector $\underline{V}_t(i,j) = [\Delta i, \Delta j]^T$ for $p_t(i,j)$ from

$$[\Delta i, \Delta j, 1]^T = (\tau_t - I)[i, j, 1]^T, \tag{3}$$

where I denotes the identity matrix,

compute $\hat{x}_t(i,j)$ from

$$\hat{x}_t(i,j) = \begin{cases} \hat{x}_{t-1}(i+\Delta i, j+\Delta j) & \text{if } |x_t(i,j) - x_{t-1}(i+\Delta i, j+\Delta j)| \le T_{mc}, \\ Q\{x_t(i,j)\} & \text{otherwise.} \end{cases} \tag{4}$$

else

compute $\hat{x}_t(i,j)$ as specified in (2).

The attractive feature of this scheme is that it does not require transmission of the motion field. Instead, the motion field is recomputed at the decoder based on the parameters of the affine transformations which map consecutive elliptical face location models onto one another. Unfortunately, the bit savings resulting from using this scheme (as opposed to conditional replenishment) in the low-pass spatio-temporal subband S_{t-1} was found to be

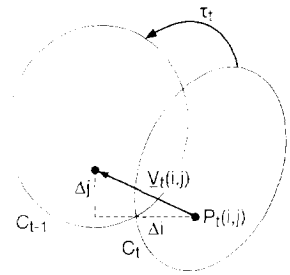

Fig. 6. Pixel-based motion compensation within elliptical face-outlines.

fairly low – in the order of 5% of the bit-rate required to code this subband. This is due to the fact that this particular motion field cannot efficiently capture the (3D) motion of a person's head and the deformability of a person's facial features. The dynamic bit allocation described in the next section has a more significant impact.

3.2.3. *Model-assisted dynamic bit allocation*

Face location information was incorporated to a modified version of the dynamic bit allocation algorithm of the 3D subband-based video teleconferencing system described in [24]. The new dynamic bit allocator, which we call *model-assisted* since it utilizes face location information, is based on a slightly different encoding priority list (shown in Fig. 4(b), as well as on a modified quantization strategy (shown in Table 2). In subbands $\{S_2, S_3, S_4\}$, two block quantizers are used, depending on whether a data block is inside or outside the face location (appropriately scaled to the resolution of these subbands). The finer of the two quantizers, denoted by Q_i, is used for blocks inside the face location. By using a coarser quantizer (Q_e) outside the face location – in the 'diagonal subband' S_4 the blocks are simply zeroed-out – a saving of bits occurs. These bits can be used to encode perceptually important data blocks in the high-pass spatial subbands $\{S_5, S_6\}$, which previously had to

[8] This only assumes that people in the scene do neither turn their backs to the camera, nor appear upside down; frontal shots as well as profiles are allowed.

Table 2
Model-assisted area-selective quantization

Subbands	Quantization
$S_{1,1}$	5-bit PCM
$S_{1,2}, S_{1,3}, S_{1,4}$	4-level GVQ
$S_{8,1}, S_{8,2}, S_{8,3}, S_{8,4}$	3-level GVQ
S_2, S_3	4-level GVQ *inside* face location 3-level GVQ *outside* face location
S_4	4-level GVQ *inside* face location zeroing *outside* face location
S_5, S_6	3-level GVQ *inside* face location zeroing *outside* face location

be discarded altogether. Since the number of bits freed up is fairly small, and since the focus is on improving facial detail in coded sequences, only high-energy blocks that are inside the scaled face location in $\{S_5, S_6\}$ are coded. The 'feedback loop' to the motion subbands takes effect after the encoding of this data. We call this type of dynamic bit allocation *model-assisted* to account for the fact that the bit allocator switches between two quantizers based on its knowledge of the location of a particular object in the subband data – in this particular case a person's face. A block diagram of the coding system with model-assisted DBA is shown in Fig. 7.

How this model-assisted dynamic bit allocation operates is illustrated in Fig. 9, where the side information images on the left were obtained from the scheme described in [24] (96 kbps coding of a CIF luminance signal), and where the images on the right were obtained by using the scheme described in this section,[9] with face location information obtained manually. In the images on the right, the two quantizers are indicated by two colors: white for the finer quantizer (4-level GVQ on 4×4 blocks), and grey for the coarser one (3-level GVQ on 4×4 blocks) in subbands $\{S_2, S_3, S_4\}$, grey for the finer quantizer (3-level GVQ on 4×4 blocks), and black for the coarser one (zeroing) in subbands $\{S_5, S_6\}$. Note that the side information required to transmit the parameters of the elliptical face location models amounts to less than 0.4 kbps,[10] i.e. a negligible 0.4% of the total bit-rate.

The improvement in the rendition of facial detail in sequences coded with model-assisted dynamic bit allocation is illustrated in Fig. 10. The coded images on the left were obtained from 3D subband coding at 96 kbps, as described in [24]; the images on the right, coded at exactly the same rate, were obtained using model-assisted DBA. The eyelids, lips, face texture for 'jelena' are all noticeably sharper in the images on the right. The eyes, spectacles, mouth and beard for 'jim' are also better reproduced in the images on the right. The data blocks in subbands $\{S_5, S_6\}$ which produce the improvement on these particular frames can be traced back to the side information images of Fig. 9. These results are also noticeable in the coded video, albeit differently. In the sequence 'jelena', the increased sharpness of facial features is fairly steady throughout the sequence. In 'jim', however, the very high motion content of the sequence leaves few bits that can be used to improve facial details. Jim's face therefore only appears sharper as long as the motion is low – i.e. at the beginning of the sequence, from which the still of Fig. 10 was extracted.

4. Automatic detection and tracking of face location

The detection of head outlines as well as outlines of persons (silhouettes) in still or moving images has been the object of active and recent research in computer vision [15, 9, 18, 19]. Only very recently was it realized that such tasks, when performed totally automatically, could be helpful in low bit-rate coding environments [28, 30]. The task of detecting and tracking face locations in a sequence of images is facilitated by both the fact that people's head outlines are fairly consistently roughly elliptical, and by the temporal correlation from frame to frame. In this section, we describe a totally auto-

[9] The difference between the images in the lower-right corners corresponding to the encoding of $S_{1,1}$ is due to the use of model-assisted pixel-based motion compensation along with model-assisted DBA for the images on the right.

[10] This number assumes four bytes of data per floating point parameter.

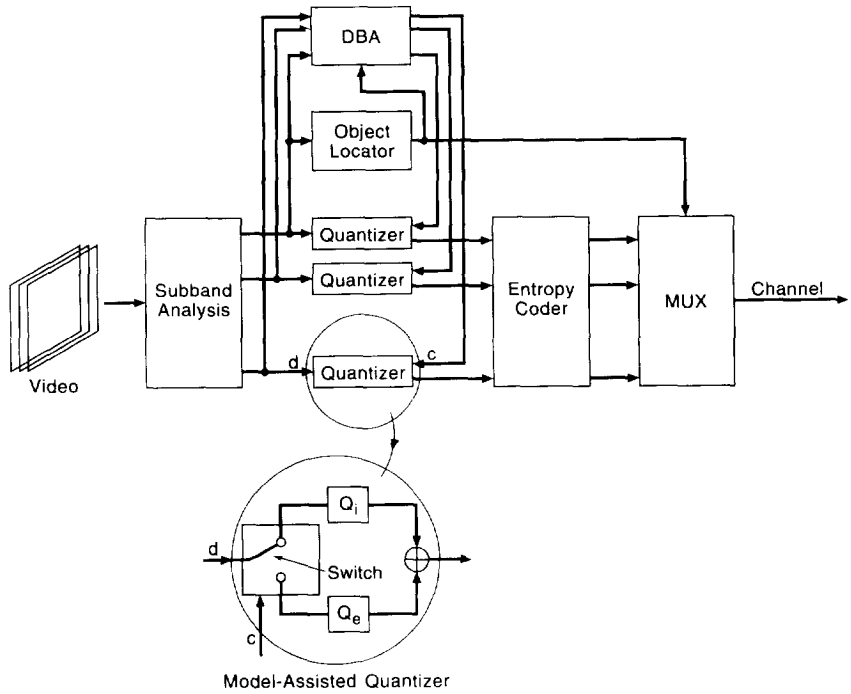

Fig. 7. 3D subband based video coder with model-assisted dynamic bit allocation.

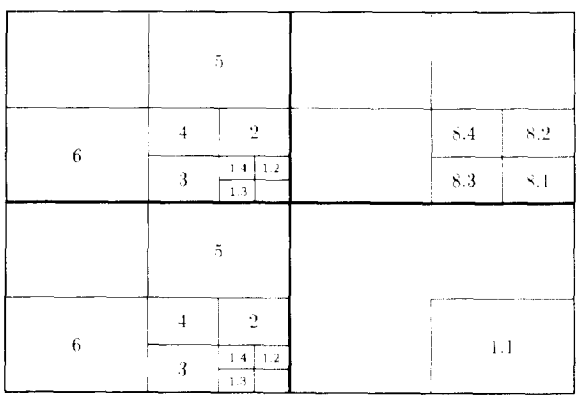

Fig. 8. Template image for side information arising from subband quantization. The side information is binary for $S_{1,1}$, $S_{8,1}$,$S_{8,2}$, $S_{8,3}$, $S_{8,4}$, ternary for $S_{1,2}$, $S_{1,3}$, $S_{1,4}$, S_2, S_3, pixel-based for $S_{1,1}$, and block-based for the other subbands. Pixels which are quantized will appear in white or grey (depending on the quantizer used) in the upper-left (LPT) and upper-right (HPT) quarter images. Blocks which are zeroed-out will appear in white in the lower-left quarter image.

matic low-complexity algorithm which was designed to perform the detection and tracking task in head-and-shoulders video sequences under minimal assumptions regarding sequence content. The algorithm belongs to a broad class of pattern-matching algorithms used for object detection [36, 35].

4.1. Detection and tracking algorithm

The algorithm detects and traces the outline of a face location, geometrically modeled as an ellipse, using as input data difference images obtained by subtracting consecutive low-pass spatio-temporal subbands $S_{1,1}$ and thresholded to produce binary images. Input images for the algorithm are therefore of size 45×30; typical input images are shown in the lower-right quarter of the images on the left side in Fig. 9. Our face location detection algorithm

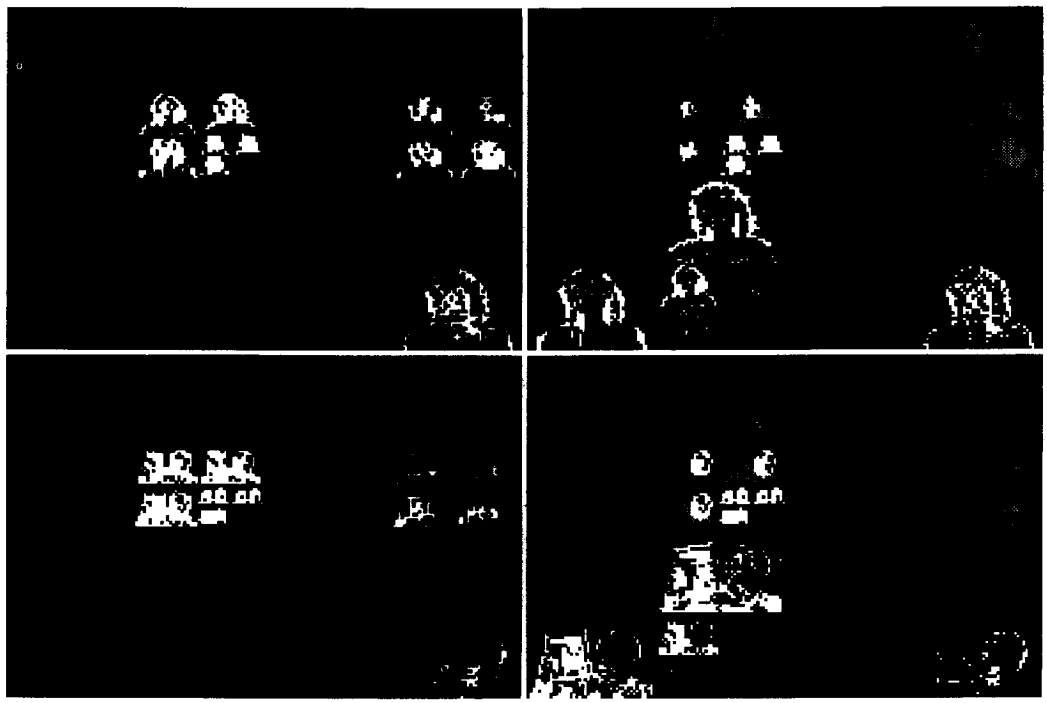

Fig. 9. Side information for a pair of subbands arising in the encoding at 96 kbps of the video sequences 'jelena' (upper images) and 'jim' (lower images). Left: without model-assisted DBA. Right: with model-assisted DBA.

was designed to locate both oval shapes (i.e. 'filled') as well as oval contours partially occluded by data. The algorithm is organized in a hierarchical three-step procedure: coarse scanning, fine scanning, and ellipse fitting. A final step consists of selecting the most likely among multiple candidates. This decomposition of the recognition and detection task in three steps, along with the small input image size,[11] make the algorithm attractive for its low computational complexity; exhaustive searches of large pools of candidates were thereby avoided. The different steps are described below, and are illustrated in Fig. 11.

[11] This input data to the algorithm is readily available at the encoder in our 3D subband coding framework. With a full-band video coding system such as one based on the H.261 standard [12, 29], similar input data can easily be generated.

Step 1: Coarse scanning. The input signal – the binary edge image corresponding to subband $S_{1,1}$ is segmented into blocks of size $B \times B$ (typically 5×5). The block size is a tunable design parameter. Each block is marked if at least one of the pixels it contains is non-zero. The block array is then scanned in a left-to-right, top-to-bottom fashion, searching for contiguous runs of marked blocks. One such run is shown in the small circle, in Fig. 11(a). For each such run the following two steps are performed.

Step 2: Fine scanning. Fig. 11(b) shows the two circled blocks of the run of Fig. 11(a), appropriately magnified. The algorithm scans the pixels contained in the blocks of a run, again in a left-to-right, top-to-bottom fashion. Here, however, the algorithm is not interested in contiguous runs of pixels, but rather in the first line that contains non-zero pixels. The first and last non-zero pixels of that line,

Fig. 10. Stills from sequences 'jelena' and 'jim' coded with 3D SBC at 96 kbps, respectively without (left), and with (right), model-assisted DBA, with the face location models in white obtained manually.

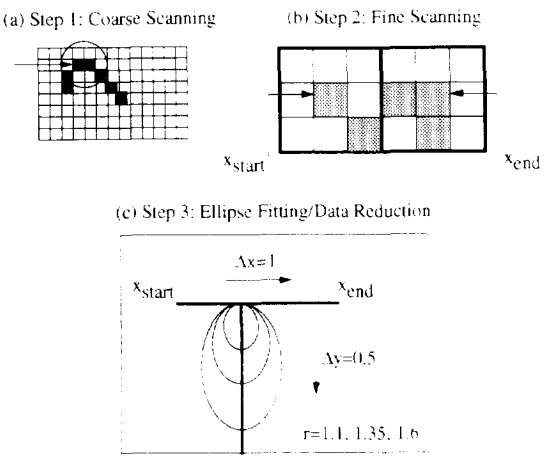

Fig. 11. Algorithm for automatic face detection and tracking in video sequences.

with coordinates (X_{start}, Y), (X_{end}, Y), define a *horizontal scanning region*.

The first two steps of the algorithm act as a horizontal edge-merging filter. The size of the block directly relates to the maximum allowable distance between merged edges. It also has a direct effect on the speed of the algorithm, which is favored by large block sizes. The purpose of these two steps is to identify candidate positions for the top of the head: due to the mechanics of human anatomy, head motion is performed under the limitations imposed by the neck joint. Consequently, and especially for sitting persons, the top of the head is usually subject to the fastest and most reliably detectable motion. At the end of the second step, the algorithm has identified a horizontal segment which potentially contains the top of the head.

Step 3: Ellipse fitting/data reduction. In this third step, illustrated in Fig. 11(c), the algorithm scans the line segment defined by (X_{start}, Y), (X_{end}, Y). At each point of the segment ellipses of various sizes and aspect ratios are tried-out for fitness, with the

top-most point of the ellipse always located on the horizontal scanning segment. Good matches are entered as entries in a list. After the search is completed on the segment, the algorithm continues at the point where it left off in Step 1. Only ellipses with 'zero tilt' ($\theta = 0$) were considered here. The primary reason for imposing this restriction is that we could trade-off an extra degree of freedom (and hence algorithm simplicity) by extending the search range for the aspect ratio.[12]

The fitness of any given ellipse to the data is determined by computing the normalized average intensities I_i and I_e of the binary pixel data on the ellipse *contour* and *border*, respectively. The fitness criterion has to be focused on the fringes of the face, since the interior region suffers from highly varying motion activity due to potentially moving lips and eyelids, or slight turns of the head. Although the contour of an ellipse is well-defined by its non-parametric form, the rasterization (spatial sampling) of image data necessitates the mapping of the continuous curve to actual image pixels. This is also true for the ellipse border. These discretized curves are defined as follows. Let $I_{\mathscr{E}}(i,j)$ be the index function for the set of points that are inside or on the ellipse $\mathscr{E}$. In other words,

$$I_{\mathscr{E}}(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ is inside or on } \mathscr{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

A pixel is classified as being on the ellipse *contour* if it is inside (or on) the ellipse, and at least one of the pixels in its $(2L+1)\times(2L+1)$ neighborhood is not, i.e.

$$(i,j) \in \mathscr{C}_i \Leftrightarrow I_{\mathscr{E}}(i,j) = 1 \text{ and } \sum_{k=i-L}^{i+L} \sum_{l=j-L}^{j+L} I_{\mathscr{E}}(k,l) < (2L+1)^2. \tag{6}$$

Similarly, a pixel is classified as being on the ellipse border if it is outside the ellipse, and at least one of the pixels in its $(2L+1)\times(2L+1)$ neighborhood is either inside or on the ellipse, i.e.:

$$(i,j) \in \mathscr{C}_e \Leftrightarrow I_{\mathscr{E}}(i,j) = 0 \text{ and } \sum_{k=i-L}^{i+L} \sum_{l=j-L}^{j+L} I_{\mathscr{E}}(k,l) > 0. \tag{7}$$

The parameter L defines the desired *thickness* of the ellipse contour and border, and is a tunable design parameter.

Given the above definitions for contour and border pixels, the normalized average intensities I_e and I_i can be defined as follows:

$$I_i = \frac{1}{|\mathscr{C}_i|} \sum_{(m,n)\in\mathscr{C}_i} p(m,n), \tag{8}$$

where $p(i,j)$ are the (binary) image data, and $|\mathscr{C}_i|$ is the cardinality of $\mathscr{C}_i$. Similarly, we have

$$I_e = \frac{1}{|\mathscr{C}_e|} \sum_{(m,n)\in\mathscr{C}_e} p(m,n). \tag{9}$$

The normalization with respect to the 'length' of the ellipse contour and border is necessary, in order to accomodate ellipses of different sizes.

An ellipse will fit ellipse-shaped data well whenever the value of I_i is high (close to one), and that of I_e is low (close to zero). In order to translate this joint maximization-minimization problem to the maximization of a single quantity, we define a model-fitting ratio R as

$$R = \frac{1 + I_i}{1 + I_e}. \tag{10}$$

The higher the value of R, the better the fit of the candidate ellipse to the head outline.[13]

In order to filter out false candidates, only ellipses which satisfy

$$I_i > I_{i_{min}} \text{ and } I_e < I_{e_{max}}, \tag{11}$$

are considered, where $I_{i_{min}}$ and $I_{e_{max}}$ are tunable design parameters. Their use is necessitated by the

[12] Typical face outlines have been found to have aspect ratios in the range of (1.4, 1.6) [19]. Moreover, the face tilt has been found to be in the range $(-30°, +30°)$; a significant constraint due to the human anatomy. Within these ranges for θ and r, a tilted ellipse can be reasonably covered by a non-tilted one, albeit with a smaller aspect ratio (in the range (1.0, 1.4)). Although this approach will result in some bits being spent to code with high quality some of the non-facial area surrounding the head, a comparison of the results obtained with both manual and automatic detection shows that the differences are perceptually marginal.

[13] In the hypothetical situation of perfectly ellipse-shaped data, the best-fitting ellipse aligned with the data would correspond to $I_i = 1$, $I_e = 0$ and $R = 2$.

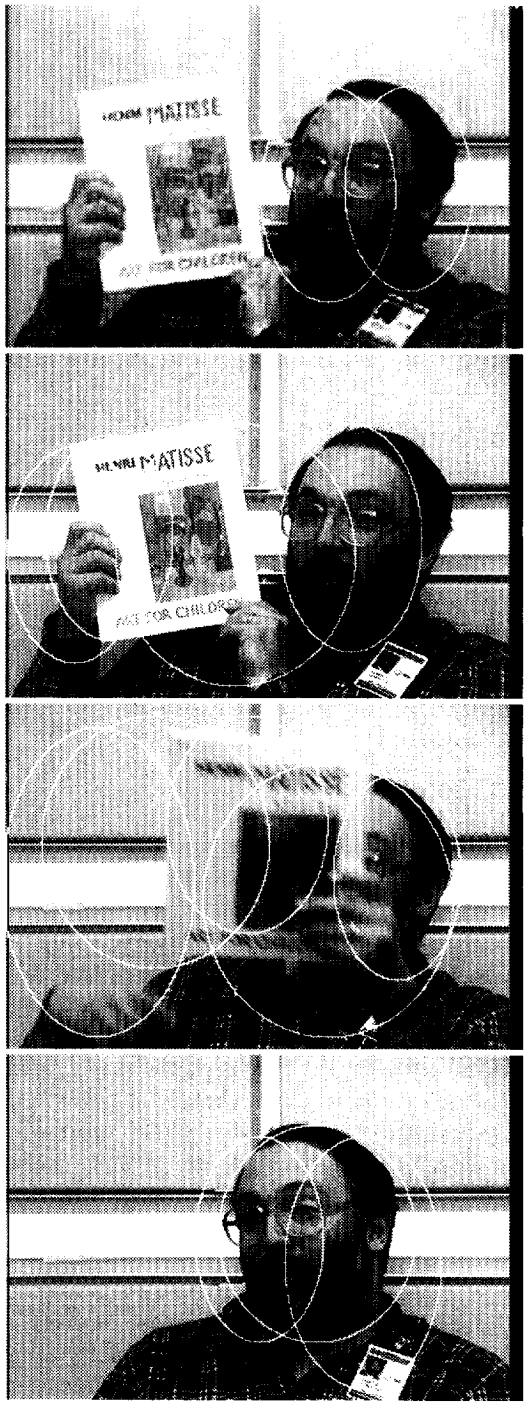

Fig. 12. Automatically detected candidate face locations in sequence 'jim' before multiple candidate elimination.

fact that R is mostly sensitive to the relative values of I_i and I_e, and much less to their absolute values.

This fitness criterion attempts to capitalize on specific properties observed on actual video data. In most cases, only an arc of the ellipse is clearly distinguishable, due to partial occlusion and to motion in the area surrounding the face (e.g. the shoulders). Using the above thresholds and the metric R, the algorithm is able to 'lock on' to such arcs, and hence yield very good results even in cases of severely occluded faces.

The above three-step procedure will in general yield more than one ellipse with a good fit, as is illustrated in Fig. 12 for the sequence 'jim'.[14] If there is a need to select a *single* final one (e.g. when it is known that the sequence only includes one person), then an elimination process has to be performed. This process uses two 'confidence thresholds' $\Delta R_{\min}$ and $\Delta I_{e_{\min}}$. If the value of R for the best-fitting ellipse is higher from the second best by more than $\Delta R_{\min}$, then the first ellipse is selected. If not, then if the border intensity difference between the two ellipses is higher than $\Delta I_{e_{\min}}$, the ellipse with the smallest I_e is selected. If the border intensity difference is smaller than that (which rarely occurs in practice), then the original best candidate (the one with the maximum R) is selected.

4.2. Experimental results

The output of sample test runs of the automatic face location detection algorithm is shown in Figs. 12 and 13. Fig. 12 shows an intermediate result, for the sequence 'jim', consisting of the output of the algorithm before the multiple candidate elimination step. The ellipses found at that stage

[14] In the case where no good fits are found, which occurs when the edge data is very sparse, the following strategy could be adopted. If this occurs at the very beginning of the video sequence to encode, the dynamic bit allocator could wait till the face tracking algorithm locks on a face location, i.e. as soon as the person starts moving. If it occurs during the course of the sequence, meaning that the person stops moving altogether, the previously found face location could be repeated; this latter case did not occur with any of the sequences used in our experiments.

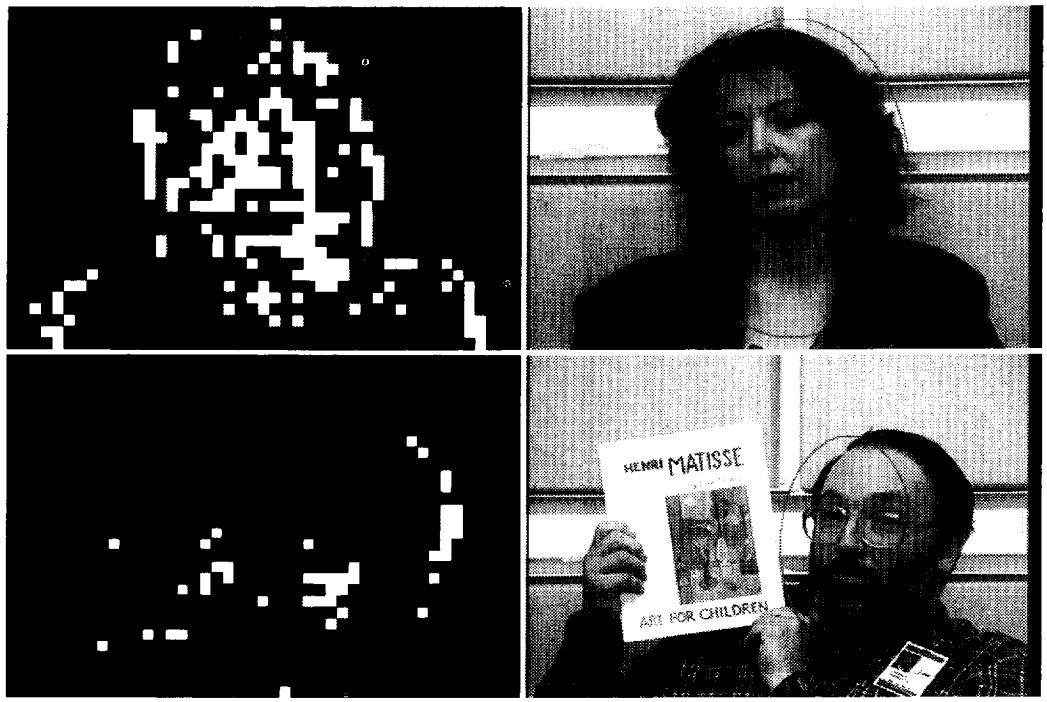

Fig. 13. Automatically detected face locations in sequences 'jelena' and 'jim'.

are 'candidate' face locations[15]. Fig. 13 shows two pairs of images. The images on the left show in white the binary edge data corresponding to subband $S_{1,1}$, with the best-fitting ellipse found by the automatic face location detection algorithm overlayed in gray. Note that these images are magnified by a factor eight in both the horizontal and vertical directions. The images on the right show the best fit magnified to the original image size of 360×240, and overlaid in gray on the originals.

The algorithm performs well, even in difficult situations such as partial occlusion of the face by a hand-held moving object. In the sequence 'jim' for example, the sweeping motion of the magazine in front of Jim's face does not 'confuse' the algorithm.[16] In other words, the elliptical model fits Jim's facial outline better (in terms of the model-fitting ratio of Eq. (10)) than the parallelogram defined by the outline of the magazine – as it should and even though the magazine severely occludes the face. In the case of more than one person in the scene, the algorithm tracks the location of the person's face for which the fit is best. Besides, the algorithm does not have the drawbacks of a feature tracking algorithm such as *snakes* [25] which has been reported to fail in situations where the motion of the features is relatively large [25, 22] – a commonplace situation in our framework where the input video is temporaly subsampled at 7.5 fps.

Fig. 14 shows stills from sequences coded at 96 kbps. The images on the left were obtained without the model-assisted concept. Those on the right show the improvement in rendition of facial features when model-assisted dynamic bit allocation is used – this time with the face outline models provided by the automatic face location detection algorithm described in Section 4.1, and with the DBA described in Section 3.2.3. A comparison of this

[15] For these stills from 'jim', the ellipses which remain after the (automatic) elimination procedure are shown in Fig. 15.

[16] Of course, a hand-held oval object of roughly the same size as Jim's face probably would.

Fig. 14. Stills from sequences 'jelena' and 'jim', coded with 3D SBC at 96 kbps, respectively without (left), and with (right), model-assisted DBA, with face locations models in white obtained by automatic detection algorithm.

figure with Fig. 10, where the face locations were obtained manually (corresponding to the benchmark), and are therefore more precisely drawn, shows that the perceptual improvement obtained in both cases is nearly identical – a result which also holds with the sequences viewed in motion.

The percentage of bits transferred to the coding of data blocks in the facial area in the high-pass spatial subbands $\{S_5, S_6\}$ varies from frame to frame.[17] The analysis of the behavior of the coder for the two sequences 'jelena' and 'jim' shows that the bit transfer rate varies between 0 and 30% of the total coding rate of 96 kbps, with an overall average over the two sequences of about 10%: a small but nevertheless significant amount. In cases where no face contours are found, the coder falls back to its non-model-assisted mode.

Fig. 15 also shows stills from sequences coded at 96 kbps, both with a coder making use of model-assisted dynamic bit allocation. In this figure, however, two different amounts of bits were transferred to the facial area. The images on the left correspond to an average bit-rate transfer of 10% of the total bit-rate to the facial area; the ones on the right to a 15% transfer.[18] Note that as the transfer rate becomes high, the discrepancy in terms of image quality between facial and surrounding areas becomes very pronounced (c.f. Jim's plaid shirt which becomes significantly blurred). A 10% average bit-rate transfer achieves a good compromise between

[17] The variation is a consequence of varying sequence content, especially in terms of motion. Bits can be devoted to the coding of subbands $\{S_5, S_6\}$ only when the motion content is not too high.

[18] This higher bit-rate transfer was achieved by zeroing blocks in the areas surrounding face locations in subbands $\{S_{5,2}, S_{5,3}, S_{5,4}\}$ and $\{S_2, S_3\}$.

Fig. 15. Stills from sequence 'jim' coded with 3D SBC at 96 kbps with model-assisted DBA, with different amounts of bit transfer to facial area. The average bit transfer is 10% and 15% of the total bit-rate for the stills on the left and right, respectively.

the two 'extreme' situations of no transfer at all and a higher (15%) transfer rate.

5. Conclusion

In this paper, we described a way to selectively encode different areas in head-and-shoulders video sequences typical of teleconferencing situations, thereby ensuring that facial features are sharp in image sequences coded at a low bit-rate. The approach, referred to as model-assisted coding, relies on the automatic detection and tracking of face locations in video sequences. The face location information is used by a 3D subband-based low bit-rate video coding system in two modules: a motion compensation module, and a model-assisted dynamic bit allocator which uses pairs of quantizers for the subband signals. In effect, the coder is assigned to transfer a small but nevertheless perceptually significant fraction of the available bit-rate from the coding of the non-facial area (area surrounding the face location model) to that of the facial area, thereby providing images with sharper facial features.

The results of coding experiments using either manually or automatically generated face location information consistently showed significant improvement in the region of interest. The improvement was found to be mostly due to the use of model-assisted dynamic bit allocation, with model-assisted motion compensation having a smaller impact. For the coding system described in this paper, at the total coding rate of 128 kbps, the perceptually optimal bit transfer rate seems to be close to 10% on average over several sequences, even though more remains to be done in this area as this number may be sensitive to the nature of the sequence (i.e. face shown in a close-up, amount of motion, etc.).

Even though a specific coding system is described, the concept is very general and could be used in the context of other video coders. The detection and tracking algorithm could also be tailored to different applications, i.e. to track any object with a simple geometric outline known a priori to be present in the scene, and also be extended to operate on multiple simultaneous objects.

References

[1] K. Aizawa, C.S. Choi, H. Harashima and T.S. Huang, "Human facial motion analysis and synthesis with applications to model-based coding", in: *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, Dordrecht, 1993, Chapter 11.

[2] K. Aizawa, H. Harashima and T. Saito, "Model-based analysis synthesis image coding (MBASIC) system for a person's face", *Signal Processing: Image Communication*, Vol. 1, No. 2, October 1989, pp. 139–152.

[3] R. Aravind, G. L. Cash, D. L. Duttweiler, H-M. Hang, B. G. Haskell and A. Puri, "Image and video coding standards", *AT&T Tech. J.*, Vol. 72, No. 1, January/February 1993.

[4] E. Badiqué, "Knowledge-based facial area recognition and improved coding in a CCITT-compatible low-bit-rate video-codec", *Proc. Picture Coding Symposium*, 1990.

[5] R. C. Beach, *An Introduction to the Curves and Surfaces of Computer-Aided Design*, Van Nostrand Reinhold, New York, 1991.

[6] C. Braccini, S. Curinga and F. Lavagetto, "Visual-acoustic modeling of videophone sources for very low bit-rate coding", *Italian National Council of Research Workshop*, Rome, Italy, January 1994.

[7] M. Buck and N. Diehl, "Model-based image sequence coding", in: *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, Dordrecht, 1993, Chapter 10.

[8] C.S. Choi, H. Harashima and T. Takebe, "Analysis and synthesis of facial expressions in knowledge-based coding of facial image sequences", *Proc. Internal. Conf. Acoust, Speech Signal Process. '91*, Toronto, Canada, 1991.

[9] I. Craw, H. Ellis and J.R. Lishman, "Automatic extraction of face features", *Pattern Recognition Lett.*, Vol. 5, No. 2, February 1987.

[10] K. Dachiku, K. Takahashi, S. Yamaguchi, T. Kuratate and K. Ohzeki, "Motion compensation subband extra/interpolative prediction coding at very low bit-rate", *Proc. VLBV 94*, University of Essex, UK, April 1994.

[11] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences", *Signal Processing: Image Communication*, Vol. 3, No. 1, February 1991, pp. 23–56.

[12] "Draft revision of recommendation H.261: Video codec for audiovisual services at $p \times 64$ kbit/s", *Signal Processing: Image Communication*, Vol. 2, No. 2, August 1990, pp. 221–239.

[13] B. Falcidieno and T.L. Kunii, Eds. *Modeling in Computer Graphics*, Springer, Berlin, 1993.

[14] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design*, Academic Press, New York, 1993.

[15] M.A. Fischler and R.A. Elschlager, "The representation and matching of pictorial structures", *IEEE Trans. on Computers*, January 1973.

[16] R. Forchheimer and T. Kronander, "Image coding – From waveforms to animation", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 37, No. 12, December 1989, pp. 2008–2023.

[17] K.H. Goh, J.J. Soraghan and T.S. Durrani, "Multi-resolution motion estimation compensation for very low bit-rate video transmission", *Proc. VLBV'94*, University of Essex, UK, April 1994.

[18] V. Govindaraju, D.B. Sher and S.N. Srihari, "Locating human faces in newspaper photographs", *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 1989.

[19] V. Govindaraju, S.N. Srihari and D.B. Sher, "A computational model for face location", *Proc. Third Internat. Conf. on Computer Vision*, Osaka, Japan, December 1990, pp. 718–721.

[20] M. Hötter, "Object-oriented analysis synthesis coding based on moving two-dimensional objects", *Signal Processing: Image Communications*, Vol. 2, No. 4, December 1990, pp. 409–428.

[21] M. Hötter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation", *Signal Processing*, Vol. 15, No. 3, October 1988, pp. 315–334.

[22] T.S. Huang, S.C. Reddy and K. Aizawa, "Human facial motion modeling, analysis and synthesis for video compression", *Proc. SPIE VCIP*, Vol. 1605, Boston, MA, 1991, pp. 234–241.

[23] *International workshop on Coding Techniques for Very Low Bit-rate Video*, University of Essex, UK, April 1994.

[24] A. Jacquin and C. Podilchuk, "Very low bit-rate 3D subband based video coding with a dynamic bit allocation", *Proc. Internat. Symp. on Fiber Optic Networks and Video Communications*, Vol. 1977, Berlin, 1993, pp. 156–167.

[25] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active contour models", *Proc. Internat. Conf. on Computer Vision*, London, UK, 1987, pp. 259–268.

[26] M. Kunt, A. Ikonomopoulos and M. Kocher, "Second-generation image coding techniques", *Proc. IEEE*, Vol. 73, No. 4, April 1985, pp. 549–574.

[27] F. Lavagetto and S. Curinga, "Object-oriented scene modeling for interpersonal video communication at very low bit-rate", *Signal Processing: Image Communication*, Vol. 6, No. 5, October 1994, pp. 379–395.

[28] C. Lettera and L. Masera, "Foreground/background segmentation in videotelephony", *Signal Processing: Image Communication*, Vol. 1, No. 2, October 1989, pp. 181–189.

[29] M. Liou, "Overview of the $p \times 64$ kbit/s Video Coding Standard", *Comm. ACM*, Vol. 34, No. 4, April 1991.

[30] M. Menezes de Sequeira and F. Pereira, "Knowledge-based videotelephone sequence segmentation", *VCIP '93*, Vol. 2094, Part 2, Cambridge, MA, November 1993.

[31] *MPEG-4 Seminar organized by Dimitri Anastasiou*, Columbia University, New York, NY, July 1993.

[32] H.G. Musmann, M. Hötter and J. Ostermann, "Object-oriented analysis–synthesis coding of moving images", *Signal Processing: Image Communication*, Vol. 1, No. 2, October 1989, pp. 117–138.

[33] Y. Nakaya, Y.C. Chuah and H. Harashima, "Model-based/waveform hybrid coding for videotelephone images", *Proc. Internat. Conf. Acoust. Speech Signal Process. '91*, Toronto, Canada, 1991.

[34] Y. Nakaya and H. Harashima, "Model-based/waveform hybrid coding for low-rate transmission of facial images", *IEICE Trans. on Communications*, Vol. E75-B, No. 5, May 1992.

[35] H. Nasr, ed., *Automatic Object Recognition*, SPIE Milestone Series, Vol. MS 41, 1991.

[36] T. Pavlidis, *Structural pattern recognition*, Springer, Berlin, 1977.

[37] C. Podilchuk and A. Jacquin, "Subband video coding with a dynamic bit allocation and geometric vector quantization", *Proc. SPIE IS&T Symp. on Electronic Imaging and Tech.*, Vol. 1666, San Jose, CA, 1992, pp. 241–252.

[38] D. Qian, "Block-based motion compensated subband coding at very low bit-rates using a psychovisual model", *Proc. VLBV94*, April 1994.

[39] H. Ueno, K. Dachiku, K. Ohzeki and F. Sugiyama, "A study on facial region detection in the standard video coding method", *Proc. 3rd Internat. Workshop on 64 kbit/s Coding of Moving Video*, 1990.

[40] Y. Wang and O. Lee, "Active mesh – a video representation scheme for feature seeking and tracking", *VCIP '93*, Vol. 2094, Part 3, Cambridge, MA, November 1993.

[41] *Workshop on Very Low Bit-rate Video Compression*, University of Illinois at Urbana-Champaign, May 1993.