Chapter VII Video Coding for Mobile Communications

Ferdous Ahmed Sohel

Monash University, Australia

Gour C. Karmakar

Monash University, Australia

Laurence S. Dooley

Monash University, Australia

ABSTRACT

With the significant influence and increasing requirements of visual mobile communications in our everyday lives, low bit-rate video coding to handle the stringent bandwidth limitations of mobile networks has become a major research topic. With both processing power and battery resources being inherently constrained, and signals having to be transmitted over error-prone mobile channels, this has mandated the design requirement for coders to be both low complexity and robust error resilient. To support multilevel users, any encoded bit-stream should also be both scalable and embedded. This chapter presents a review of appropriate image and video coding techniques for mobile communication applications and aims to provide an appreciation of the rich and far-reaching advancements taking place in this exciting field, while concomitantly outlining both the physical significance of popular quality image and video coding metrics and some of the research challenges that remain to be resolved.

INTRODUCTION

While the old adage is that a picture is worth thousands of words, in the digital era a colour

image typically corresponds to more like a million words (double bytes). While an image is a twodimensional spatial representation of intensity that remains invariant with respect to time (Tekalp, 1995), video is a three-dimensional time-varying image sequence (Al-Mualla, Canagarajah, & Bull, 2002) and as a consequence represents far more information than a single image. Mobile technologies are becoming omnipresent in our lives with the common mantra to communicate with anybody, anytime, anywhere. This has fueled consumer demand for richer and more diverse mobile-based applications, products, and services, and given the human visual system (HVS) is the most powerful perceptual sensing mechanism, it has inevitably meant that image and latterly video technologies are the drivers for many of these new mobile solutions.

Second generation (2G) mobile communication systems, such as the Global System for Mobile (GSM) started by supporting a number of basic multimedia data services including voice, fax, short message services (SMS) and informationon-demand (news headlines, sports scores and weather). *General Packet Radio Service* (GPRS). which has often been referred to as 2.5G, extends GSM to provide packet switching services and afford the user facilities including e-mail, stillimage communication, and basic Internet access. By sharing the available bandwidth, GPRS offers efficiency gains in applications where data transfer is intermittent like Web-browsing, e-mail, and instant messaging. The popularity of GSM and GPRS led to the introduction of third generation (3G) mobile technologies which address live video applications, with real-time video telephony being advertised as the flagship application for this particular technology, offering a maximum theoretical data rate of 2Mbps, though in practice this is more likely to be 384Kbps. Multimedia communications along with bandwidth allocation for video and Web applications remains one of the primary focuses of 3G as well as the proposed fourth generation (4G) mobile technologies, which will provide such functionality as broadband wireless access and interactivity capability, though it is not due to be launched until 2010 at the earliest. Many technological challenges remain including

the need for greater coding efficiency, higher data rates, lower computational complexity, enhanced error resilience and superior bandwidth allocation, and reservation strategies to ensure maximal channel utilisation. When these are resolved, mobile users will benefit from a rich range of advanced services and enhanced applications including video-on-demand, interactive games, video telephony, video conferencing and tele-presence, tele-surveillance, and monitoring.

As video is a temporal sequence of still frames, coding in fact involves both single (intra) and multiple (inter) frame coding algorithms, with the former being merely still image compression. Since, for mobile applications only low bit-rate video sequences are suitable, this chapter analyses both high image and video compression techniques. Approaches to achieving high image compression are primarily based upon either the discrete cosine transform (DCT), as in the widely adopted Joint Picture Expert Group (JPEG) standard or the discrete wavelet transform (DWT) which affords scalable and embedded sub-band coding in the most recent interactive JPEG2000 standard. In contrast, a plethora of different inter-frame coding techniques have evolved within the generic blockbased coding framework, which is the kernel of most current video compression standards such as the Moving Picture Expert Group family of MPEG-1, MPEG-2, and MPEG-4, together with the symmetrical video-conferencing H.261 and H.263 coders and their variants. MPEG-4, which is the latest audio/video coding family member, offers object-based functionality and is primarily intended for Internet-based applications. It will be examined later in the chapter, together with the main features of the newest video coding standard, somewhat prosaically known as H.264 or advanced video coding (AVC), which is now formally incorporated into MPEG-4.

All these various compression algorithms remove information content from the original video sequence in order to gain compression efficiency, and without loss of generality the quality of the encoded video will be compromised to some extent. As a consequence, the issue of quality assessment arises, which can be subjective, objective, or both, and this chapter will explore both the definition and physical significance of some of the more popular quality metrics. In addition, the computational complexity of both the encoder and decoder directly impacts upon the limited power resources available for any mobile unit. Moreover, in this consumer driven age, the insatiable desire for choice means some people will pay more to get a higher quality of service product, while others will be more than happy with basic functionality and reasonable signal quality. In order to ensure the availability of different consumer levels within the same framework, it is essential to ensure that signal coding is both scalable and embedded.

PERFORMANCE AND QUALITY METRICS OF VIDEO CODING ALGORITHMS

The performance of all contemporary video coding systems is normally assessed using a series of well-accepted metrics mentioned by Bull, Canagarajah, and Nix (1999), including:

- Coding efficiency,
- Picture reconstruction quality,
- Scalable and embedded representations,
- Error resilience.
- Computational complexity, and
- Interactivity.

Coding Efficiency

This is one of the prime metrics for low bit-rate coding in mobile communications, with the inherent bandwidth limitations of mobile networks propelling research to explore highly efficient video coding algorithms. Compression is achieved by reducing the amount of data required to represent

the video signals by minimising inherent redundancies in both the spatial and temporal domains, as well as to some degree, dropping insignificant or imperceptible information at the pyrrhic cost of a loss in quality. In addition, higher coding gains can be achieved using lower spatial and temporal (frame rate) resolution video formats, such as the *common interchange format* (CIF) and sacrificing the colour depth of each pixel, though again this impacts on perceived quality. *Compression ratio* (CR) is the classical metric for measuring coding efficiency in terms of the information content of the video and can be evaluated in numerous ways (Al-Mualla et al., 2002). For example,

$$CR = \frac{\text{number of bits in original video}}{\text{number of bits in compressed video}}$$
 (1)

From a purely compression perspective, an encoder generating a higher CR is regarded as superior to one with a lower CR, as the clear advantage secured is that without loss of generality, a smaller bit-stream incurs a lower transmission time. An alternative representation is to use *compression* (C) which is quantified in *bits per pixel* (*bpp*), where the best encoder generates the lowest *bpp* value. This is formally defined as:

$$C = \frac{\text{size of the compressed video (bits)}}{\text{number of pels in original video}} \ bpp (2)$$

Picture Reconstruction Quality

Video coding for mobile communications is by its very nature lossy, so it is essential to be able to quantitatively represent the loss and reflect the compression achieved. To specify, evaluate, compare, and analyse video coding and communication systems, it is necessary to determine the level of picture quality of the decoded images displayed to the viewer. Visual quality is inherently *subjective* and is influenced by many factors that

make it difficult to obtain a completely accurate measure for perceived quality. For example, a viewer's opinion of visual quality can depend very much on their psycho-physical state or the task at hand such as passively watching a movie, keenly watching the last few overs of a tense cricket match, actively participating in a video conference session, or trying to identify a person in a video surveillance scene. Measuring visual quality using objective criteria can give both accurate and repeatable results, but as yet there is no unified quantitative measurement system that entirely reproduces the perceptual experience of a human observer (VQEG, 1998) or no single metric that consistently outperforms other (objective) techniques from a subjective viewpoint (Wu & Rao, 2006). In the following section, both subjective and objective quality measuring techniques are examined.

Subjective Quality Measurement

Human perception of a visual scene is formed by a complex interaction between the components of the HVS, particularly through the eye to the brain. Perceived visual quality is affected by many different factors, including:

- Spatial fidelity (how clear parts of a scene are to the viewer, whether there is obvious distortion, whether the objects retain their geometric or structural shape, what the objects look like, and other fine detail concerning colour, lighting, and shading effects);
- Temporal fidelity (whether the motion appears natural, continuous and smooth);
- Viewing conditions: Distance, lighting, and colour:
- Viewing environment: a comfortable, nondistracting environment usually leads to the perception of higher quality, regardless of the actual quality of the scene;

- Viewer's state of mind, domain knowledge, training and expertise, interest, and the extent to which the observer interacts;
- The *recency effect*: the psychological opinion upon a visual sequence is more heavily influenced by recently-viewed rather than older video material (Wade & Swanston, 2001);
- Viewer's psycho-physical condition.

All these factors combine to make it expensive, time consuming, and extremely difficult to accurately measure visual quality, though a number of subjective assessment methodologies do exist, such as the *double stimulus impairment scale* (DSIS), *double stimulus continuous quality scale* (DSCQS), and *single stimulus continuous quality scale* (SSCQS) as of Ghanbari (1999). Moreover, the *Video Quality Expert Group* (VQEG), which was established in 1997, is currently working on the establishment a unified quality measurement standard (Wu et al., 2006) to enable subjective testing of both image and video data. The current status of the VQEG will be discussed shortly.

Objective Quality Measurement

Since subjective quality measurement is so sensitive to a large number of factors and may not be repeatable, measuring the visual quality using objective criteria becomes of paramount importance. Objective measurements give accurate and repeatable results at low cost and so are widely employed in video compression systems. A number of objective quality measuring techniques have been adopted by researchers and these will now be briefly investigated:

PSNR: Among all the objective measurements the logarithmic *peak-signal-to-noise-ratio* (PSNR) metric is most widely used in the literature and is defined as:

$$PSNR_{DB} = 10\log_{10} \frac{(2^{n} - 1)^{2}}{MSE}$$
 (3)

where MSE is the mean squared error between the original and approximating video and $(2^n - 1)$ is the maximum possible signal value in an *n*-bit data representation. The MSE for each frame is given by:

$$MSE = \frac{1}{H \times V} \sum_{x=0}^{H} \sum_{y=0}^{V} \left[f(x, y) - \widetilde{f}(x, y) \right]^{2}$$
 (4)

where H and V are respectively the horizontal and vertical frame dimensions, while f(x, y) and f(x,y) are the original and approximated pixel values at location (x, y). Using this definition, a video with a higher PSNR is therefore rated better than one with a lower value. PSNR is commonly applied for three basic reasons as summarised by Topiwala (1998): (1) it is a first order analysis and treats data samples as independent events using a sum of squares of the error measure; (2) it is straightforward to compute and leads to easily tractable optimisation approaches; and (3) it has a reasonable correspondence with perceived image quality as interpreted by either humans or machine interpreters. It does, however, have some limitations (Richardson, 2003) most notably that it requires an unimpaired original video as a reference, though this may not be always available and also not easy to verify the original video had perfect fidelity and does not necessarily equate to an absolute subjective quality.

 L_p Metrics: In addition to the MSE metric, various other weightings derived from the L_p norms can be used as quality measures. While closed-form solutions are feasible for minimising MSE, they are virtually impossible to obtain for normalised L_p metrics, which are formally defined as:

$$\varepsilon_p = \frac{1}{H \times V} \sum_{x=0}^{H} \sum_{y=0}^{V} \left| f(x, y) - \widetilde{f}(x, y) \right|^p, p \neq 2 \quad (5)$$

Fast and efficient search algorithms make L_p norms ideally applicable, especially if they correlate more precisely with subjective quality. Two p-norms, namely $p = \infty$ (L_{∞}) and p = 1 (L_1), correspond to the peak absolute error and sum-of-error magnitudes respectively and are widely referred to as the *class one* and *class two* distortion metrics, in the literature (Katsaggelos et al., 1998; Kondi et al., 2004; Meier, Schuster, & Katsaggelos, 2000; Schuster & Katsaggelos, 1997; Sohel, Dooley, & Karmakar, 2006a).

In the vertex-based video-object shape coding algorithms (Katsaggelos, Kondi, Meier, Ostermann, & Schuster, 1998; Kondi, Melnikov, & Katsaggelos, 2004; Schuster et al., 1997), distortion is measured from a slightly different perspective. Instead of considering the entire object, only the geometric distortion at the object boundary points is considered. The shortest absolute distance (SAD) between the shape boundary points and the corresponding approximating shape is then applied as the measurement strategy, though this can lead to erroneous distortion measures, especially at shape corners and sharp edges. In fact, the SAD guarantees every point on the approximated shape is within the correct geometric distortion but does not ensure all points on the shape boundary produce the distortion accurately. To overcome this anomaly, a new distortion measurement strategy has been developed (Sohel, Dooley, & Karmakar, 2006a) that accurately measures the Euclidean distance between the reference shape and its approximation for generic shape coding.

In the MPEG-4 standard, the *relative area er*ror (RAE) D_n measure is used to represent shape distortion (Brady, 1999): $D_n = \frac{\text{number of mismatched pixels in the approximated shape}}{\text{number of pixels in the original shape}}$

(6)

Though it should be noted that since different shapes can have different ratios of boundary pels to interior pels, D_n only provides physical meaning when it is used to measure different approximations of the same shape (Katsaggelos et al., 1998).

As there are numerous quality metrics for both subjective and objective evaluation, it has become essential to attempt to formally standardise them. As alluded earlier, the VQEG has the objective of unifying objective picture quality assessment methods to reflect the subjective perception of the HVS. Despite their best efforts and rigorous testing in two phases, Phase I (1997-1999) and Phase II (2001-2003), an overall decision has yet to be made, though Phase I concluded that no objective measurement system was able to replace subjective testing and no single objective model outperformed all others in all cases (VQEG, 1999; Wu et al., 2006). Details of the various quality measurement techniques and vision model-based digital impairment metrics, together with perceptual coding techniques are described and analysed by Wu et al. (2006).

Scalability and Embedded Representations

Scalable compression refers to the generation of a coded bit-stream that contains embedded subsets, each of which represents an efficient compression of the original signal. The one major advantage of scalable compression is that either the target bit-rate or reconstruction quality does not need to be known at the time of compression (Taubman, 2000). A related advantage of practical significance is that the video does not have to be compressed multiple times in order to achieve

a target bit-rate, so scalable encoding enables a decoder to only selectively decode portions of the bit-stream. It is very common in multicast/broadcast systems that there are different receivers having different capacities and different users supposed to be receiving different qualityof-service (QoS) levels. In scalable encoding, the bit-stream comprises one base layer and either one or more associated enhancement layers. The base layer can be independently decoded and the various enhancement layers conjointly decoded with the base layer to progressively increase the perceived picture quality. For mobile applications such as video-on-demand and TV access for mobile terminals, the server can transmit embedded bit-streams while the receiver processes the incoming data at its capacity and eligibility. In recent times, for example in Taubman (2000), Taubman and Marcellin (2002), Taubman and Zakhor (1994), and Atta and Ghanbari (2006), scalable video coding has assumed greater priority than other related issues including optimality and compression ratio.

As the example in Figure 1 shows, *Decoder* 1 only processes the base layer while Decoder N handles the base and all enhancement layers, thereby generating a range of possible picture qualities from basic through to the very highest possible quality, all from a single video bit-stream. An intermediate decoder, such as Decoder i then utilises the base layer and enhancement layers up to the ith inclusive layer to generate a commensurate picture quality. Video coding systems are typically required to support a range of scalable coding modes, with the following three being particularly important: (1) spatial scalability, which involves increasing or decreasing picture resolution, (2) temporal scalability which provides varying picture (frame) rates, and (3) quality (amplitude) scalability which varies the picture quality by changing the PSNR or L_n metric for instance.

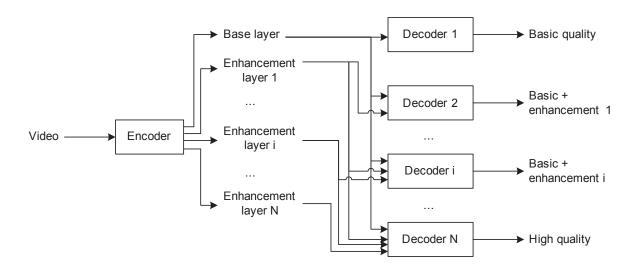


Figure 1. Generic concept of scalable and embedded encoding

Error Resilience

Mobile communication channels are notorious hostile environments with high error rates caused by many different loss mechanisms ranging from multi-path fading, carrier signal strengths, co-channel interference, network congestion, misrouting through to channel noise (Wu et al., 2006). For coded video, the impact of these errors is magnified due to the fact that the bit-stream is highly compressed. Indeed, the greater the compression, the more sensitive the bit-stream is to errors, since each bit represents a larger portion of the original video and crucially, the bit-stream synchronization may become disturbed. The effect of errors on video is also exacerbated by the use of predictive and variable-length coding (VLC), which can lead to both temporal and spatial error propagation, so it is clear that transmitting compressed video over mobile channels may be hazardous and prone to degradation. Error-resilience techniques are characterised by their ability to tolerate errors introduced into the compressed bit-stream while maintaining an acceptable video quality, with such strategies occurring at the encoder and/or decoder (Redmill, 1994; Salama, Shroff, Coyle, & Delp, 1995). The overall objective of error resilient coding is to reduce the effect of data loss by taking remedial action and displaying a quality video or image representation at the decoder, despite the fact the encoded signal may have been corrupted in transmission.

Computational Complexity

In mobile terminals, both processing power and battery life are scarce resources and given the high computational overheads necessitated to process video signals, employing computationally efficient algorithms is mandated. Moreover, for real-time video applications over mobile channels, the transmission delay of the signal must be kept as low as possible. Both symmetrical (video conferencing and telephony) and asymmetric (TV broadcast access and on-demand video) mobile applications mean that video coding-decoding algorithms should be designed so that mobile terminals incur minimal computational

overheads. There are various steps that can be adopted to reduce computational complexity. First by using fast algorithms, appropriate transformations, fast search procedures for motion compensation and efficient encoding techniques at every step. Second, minimise the amount of data to be processed—if the data size is small it requires a lower computational cost. The amount of data can be reduced for example by using lower spatial and/or temporal resolutions as well as by sacrificing pixel colour depth so attenuating the bandwidth requirements and power consumption in both processing and transmission.

Interactivity

In many mobile applications involving Web browsing, video downloading and enjoying on-demand video, playing online games, interactivity has become a key element and with it also the user's expectation over the degree of interactivity available. For instance, users normally expect to have control over the standard suite of video recorder functions like play, pause, stop, rewind/forward, and record, but may in addition, also wish to be able to select a portion of video and to edit it or insert into another application similar to a multimedia-authoring tool (Bull et al., 1999). MPEG-4 functionality includes object-based manipulation so providing much greater flexibility for interactive indexing, retrieval, and editing of video content to the mobile users. Moreover, the H.264 standard enables switching between multiple bit-rates while browsing and downloading, together with interactive bandwidth allocation and reservation

HIGH COMPRESSION INTRA-FRAME VIDEO CODING AND IMAGE CODING TECHNIQUES

As a single video frame is in fact a still image, *intra-frame video coding* and *image coding* have

exactly the same objective, namely to achieve the best compression by exploiting spatial correlations between neighbouring pels. High image compression techniques have attracted significant research interest in recent years, as they permit visible distortions to the original image in order to obtain a high CR. While numerous image compression techniques have been proposed, this chapter will specifically focus on waveform coding methods including transform and sub-band coding together with vector quantisation (VQ) since these are suitable for and commonly used in mobile communication applications. Second generation techniques that attempt to describe an image in terms of visually meaningful primitives including shape contour and texture will then be analysed.

Waveform-Based Coding

Waveform-based coding schemes typically comprise the following three principal steps:

- Decomposition/transformation of the image data,
- Quantisation of the transform co-efficients, and
- Rearrange and entropy coding of the quantised co-efficients.

Figure 2 shows the various constituent processing blocks of a waveform coder, each of which will now be considered.

Transform Coding

The first step of a waveform coder is *transformation* that maps the image data into an alternative representation so most of the energy is compacted into a limited number of transform co-efficients with the remainder either being very small or zero. This de-correlates the data so low energy co-efficients may be discarded with minimal impact upon the reconstruction image quality. In addition,

Figure 2. A generic waveform-based image coder

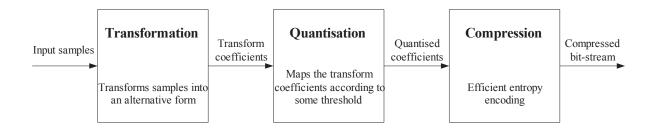
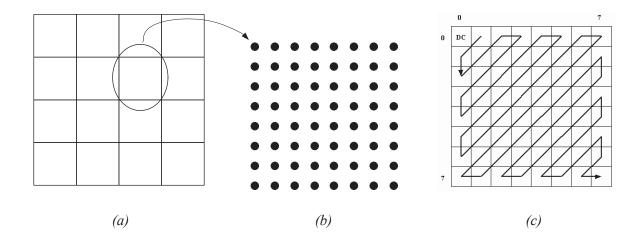


Figure 3. JPEG coding principles: (a) partition the image into 8×8 macroblocks, (b) the 64 pels in each block, (c) the conventional zigzag scan ordering for the quantised DCT co-efficients which are represented by the respective cells of the matrix



the HVS exhibits varying sensitivity to different frequencies with normally a greater sensitivity towards the lower than the high frequencies. There are many possible waveform transforms, though the most popular is the *discrete cosine transform* (DCT) which has as its basis the discrete fourier transform (DFT). Indeed, the DCT can be viewed as a special case of the DFT as it decomposes images using only cosines or even-symmetrical functions and for this reason, it is

the fundamental building block of both the JPEG image and MPEG video compression standards. With the DCT, the image is first subdivided into blocks known as *macroblocks* (MB), which are normally of fixed size and usually 8×8 pels, with the DCT applied to all pels in that block (see Figures 3(a) and 3(b). The next major step after applying the DCT is quantisation, which maps the resulting 64 DCT co-efficients into a much smaller number of output values. The *Q-Table*

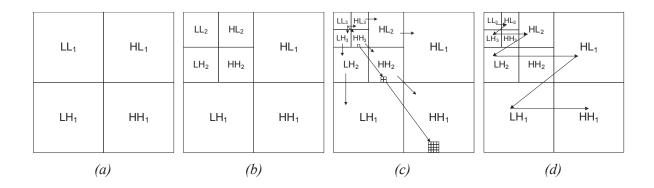
is the formal mechanism whereby the quantised DCT co-efficients are then adapted to reflect the subjective response of the HVS, with higher-order transform co-efficients being more coarsely quantised than lower frequencies. As the number of non-zero quantised co-efficients is usually much lower than the original number of DCT co-efficients, this is how image compression is achieved. The quantised 8×8 DCT co-efficients (many of which will be zero) are then converted into a single sequence, starting with the lowest DCT frequency and then progressively increasing the spatial frequency. As all horizontal, vertical, and diagonal components must be considered, rather than read the co-efficients either laterally or longitudinally from the matrix, the distinctive zigzag pattern shown in Figure 3(c) is used to scan the quantised co-efficients in ascending frequency which tends to cluster low-frequency, non-zero coefficients together. The first frequency component (top left of the DCT matrix) is the average (DC) value of all the DCT frequencies and so does not form part of the zigzag bit-stream, but is instead differentially coded using DPCM, with other DC components from adjacent MB. The resulting sequence of AC frequency co-efficients, will after quantisation, contain many zeros so the final step is to employ lossless entropy coding such as the VLC *Huffman Code* to minimise redundancy in the final encoded bit-stream. Arithmetic (Witten, Neal, & Cleary, 1987) and Lempel-Ziv-Welch (LZW) (Welch, 1984; Ziv & Lempel, 1977) coders can be used as alternatives since they are also entropy-based.

In terms of compression performance, at higher CR typically between 30 and 40, DCT-based strategies begin to generate visible *blocking artefacts* and as all block-based transforms suffer from this distortion to which the HVS is especially sensitive, DCT-based coders are generally considered inappropriate for very low bit-rate image and video coding applications.

Sub-Band Coding

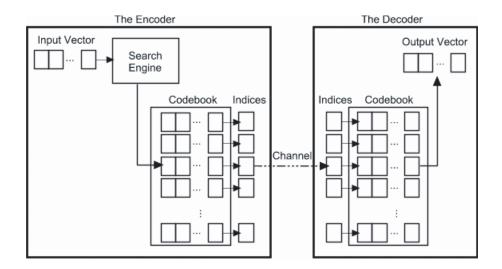
Sub-band image coding which does not produce the aforementioned blocking artefacts, has been the subject of intensive research in recent years (Crochiere, Webber, & Flanagan, 1976; Said & Pearlman, 1996; Shapiro, 1993; Taubman, 2000; Woods & O'Neil, 1986). It is observed from the mathematical form of the rate-distortion (RD) function that an efficient encoder splits the original signal into spectral components of infinitesimally

Figure 4. Sub-band decomposition: (a) first level DWT based on Daubechies (1990), (b) second level, (c) parent-child dependencies in the three level sub-band, and (d) the overall scanning order of the decomposed levels



small bandwidth and then independently encodes each component (Nanda & Pearlman, 1992). In sub-band coding, the input image is passed through a set of band-pass filters to decompose it into a set of sub-band images prior to critical sub-sampling (Johnston, 1980). For example, as shown in Figure 4(a), following the first decomposition level, the image is divided into 4 sub-bands where L and H respectively represent the low and high pass filtered outputs (for the horizontal and vertical directions), while the numerical subscript denotes the decomposition level. Subsequently, the lowest resolution sub-image (LL₁) is further decomposed at the 2nd level (see Figure 4(b)) because, as mentioned in the previous section, most signal energy tends to be concentrated in this sub-band. As each resulting sub-image has a lower spatial resolution (bandwidth) they are down-sampled before each is independently quantised and coded. It is worth noting that like the DCT, sub-band decomposition does not in itself lead to compression as the number of the sub-bands remains equal to the number of samples in the original image. However, the elegance of this approach is that each sub-band can be coded efficiently according to its statistics and visual prominence, leading to an inherent embeddedness and scalability in the sub-band coding process. As the example in Figure 4(c) illustrates, sub-bands at lower resolution levels contain more coarse information about their dependent levels in the hierarchy. For instance, LL₃ contains information about HL₃, LH₃, and HH₃ and these four sub-bands form LL, which contains information about HL, LH₂, and HH₂. LL₃ therefore contains the coarse and basic information about the image and the dependent levels contain some more hierarchical information so the sub-band process inherently affords both embedded and scalable coding. The discrete wavelet transform (DWT) (Daubechies, 1990) is most commonly used for decomposition as it has the capability to operate at various scales and resolution levels. As with the DCT, DWT coefficients are quantised before encoded and then a number of strategies can be used to code the resulting sub-bands using various scanning processes analogous to the zigzag pattern employed by JPEG. Note, due to the sub-band decomposition, scanning DWT co-efficients in ascending order of frequency is far more complex, though alternative techniques exist, with one of the most popular and efficient being Shapiro's embedded zero tree (EZW) wavelet compression (Shapiro, 1993), which introduced the concept of zero trees to derive a rate-efficient embedded coder. Essentially, the correlation and self-similarity across decomposed wavelet sub-bands is exploited to reorder the DWT co-efficient in terms of significance for embedded coding. Said and Pearlman (1996) presented a further advancement with the spatial partitioning of images into hierarchical trees (SPIHT) which at the time, was recognised as the best compression technique. Inspired by the EZW coder, they developed a set-theoretic data structure to achieve very efficient embedded coding that improved upon EZW in terms of both complexity and performance, though the major limitation of SPIHT is that it does not provide quality (SNR) scalability. Taubman subsequently introduced the embedded block coding with optimised truncation (EBCOT) (Taubman, 2000) method which affords higher performance as well as both SNR and spatial scalable image coding. EBCOT outperforms both EZW and SPIHT and is generally considered by the research community as the best DWT-based image compression technique to the extent that it has now been incorporated into the new JPEG2000 still image coding standard (Taubman, 2002). Each sub-band is partitioned into small blocks of samples called code-blocks so EBCOT generates a separate highly scalable bit-stream for each code-block which can be independently truncated to any of a collection of different lengths. The EBCOT block coding algorithm is built around the concept of fractional bit-planes (Li & Lei, 1997; Ordentlich, Weinberger, & Seroussi, 1998) which ensures efficient and finely embedded coding.

Figure 5. Encoder/decoder in a VQ arrangement. Given an input vector, the best matched codeword is found and its index in the codebook is transmitted. The decoder uses the index and outputs the codeword using the same VQ codebook.



From a mobile communication perspective, the scalability, high compression, and low complexity performance of JPEG2000 make it an increasingly attractive coding option for low bit-rate applications. The one drawback of sub-band image coding is however, that since higher frequency co-efficients are discarded from the encoded image, blurring effects occur at high CR levels, though perceptually this is preferable to the inherently blocky effects of transform coding.

Vector Quantisation (VQ)

In VQ, the input image data is first decomposed into a *k*-dimensional input *vector* and a corresponding *code-vector* which is stored in a predefined lookup table known as a *codebook*, which is searched so that it provides the best match for each input vector. The corresponding index of the code-vector in the codebook is then transmitted to

the decoder, where it is used to retrieve the relevant code-vector using exactly the same codebook, so enabling the image to be efficiently reconstructed. Figure 5 shows the schematic diagram of a typical VQ-based system. There are many VQ variants (Rabbani & Jones, 1991) including, for example, adaptive VQ, tree structured VQ, classified VQ, product VQ, pyramid VQ, while the challenging matter of how best to design the codebook is another well-researched area (Linde, Buzo, & Gary, 1980; Chou, Lookabaugh, & Gary, 1989).

Second Generation Techniques

Waveform-based image coding techniques operate either on individual pels or blocks of pel using a statistical model, which can lead to some disadvantages including: (1) Greater emphasis being given to a codeword assignment that statistically reduces the bit-requirement, rather than

the extraction of representative messages from the image; (2) The encoded entities are consequences of the technical constraints in transforming images into digital data, that is, from the spatial to frequency domains or RD constraints, rather than being real entities; (3) They do not fully exploit the properties of the HVS. This led to a new coding class collectively known as second generation methods (Kunt, Ikonomopoloulos, & Kocher, 1985) that decompose the image data into visual primitives such as contours and textures. There are many approaches for this type of coding such as, for example, dividing an image into directional primitives using segmentation-based techniques to extract regions from the image which are represented by their shape and texture content. Sketch-based coding also uses a similar segmentation based approach, and details on these and other second generation techniques may be found in Kunt et al. (1985).

Second generation methods provide higher compression than waveform-coding methods for the same reconstruction quality level (Al-Mualla et al., 2002) and do not possess the problems of blocking and blurring artefacts at very low bitrates. They are particularly suitable for encoding images and video sequences from regular domains known *a prior* such as, for example, animations. The extraction of real objects however, is both intractable and computationally expensive, and in addition these methods suffer from unnatural contouring effects like the loss of continuity and smoothness which can make image detail look artificial.

Other intra-frame coding techniques include *iterated function* systems (IFS) (Distasi, Nappi, & Riccio, 2006; Øien, 1993), fractal geometry-based coding (Barnsley, 1988; Jacquin, 1992), prediction coding, block-truncation coding, quad-tree coding, recursive coding, and multi-resolution coding. IFS expresses an image as the attractor of a *contractive function* system which can be retrieved simply by progressively iterating the set of functions starting from any initial

arbitrary shape. IFS-based compression affords good performance at high CR in the range of 70-80 (Barthel, Voye, & Noll, 1993; Jacobs, Fisher, & Boss, 1992), though this is counterbalanced by the fact that such techniques are computationally complex and hence time consuming. A comprehensive review of second generation techniques can be found in Clarke (1995).

INTER-FRAME VIDEO CODING

As video is a sequence of still frames, a naïve yet simple approach to video coding is to employ any of the still image (intra-frame) coding methods previously discussed on a frame-by-frame basis. Motion JPEG (M-JPEG) is one such approach that contiguously applies JPEG intra-frame coding (Wallace, 1991) to each individual frame, and while it has never been standardised, unlike the new M-JPEG2000 which is formally defined as part of the JPEG2000 compression standard, the drawback in both approaches is that they do not exploit the obvious temporal correlations that exist between many consecutive video frames, so limiting their coding efficiency. As the example in Figure 6 highlights, there is considerable similarity between the two frames of the popular Miss America test video sequence, so if the first frame is encoded in intra-mode and the difference between the current and the next frame is coded instead, a large bit-rate saving can be achieved. Inter-frame video coding refers to coding techniques that achieve compression by reducing the temporal redundancies within multiple frames. In addition, to reduce spatial redundancy, existing intra-frame coding techniques can serve as the basis for the development of inter-frame coding. This can be done either by generalising them for 3D signals, viewing the temporal as the third dimension, or by predicting the motion of the video in the current frame from some already encoded frame(s) as the reference to reduce the temporal redundancy. Inter-frame coding alone however, is

Figure 6. Temporal redundancy between successive frames: (a) and (b) are respectively the 29th and 30th frames of the Miss America video sequence, (c) pixel-wise difference between them







inappropriate for many video applications which for instance require random access within the frames, so all reference frames have to be intracoded. In practice a combination of intra- and inter-frame coding is usually applied whereby certain frames are intra-frame coded (so called *I-frames*) at specific intervals within the sequence and the remaining frames are inter-frame coded (*P-frames*) with reference to the I-frames. Some frames known as *B-frames*, may also have both forward and backward reference frames. There are also some video coding systems, such as the latest H.264 standard, which have a provision for switching between intra- and inter-frame coding modes within the same frame, and introduce new picture types known as Switching-P (SP) and Switching-I (SI) frames which enable drift-free switching between different bit-streams.

Three categories of inter-frame video coding suitable for mobile communications will now be discussed.

Waveform-Based Techniques

The easiest way to extend the 2D (spatial) image coding to inter-frame video coding is to consider 3D (spatial and temporal) waveform coding. The

basic framework will be similar to that in Figure 2, with the notable exception that 3D transformations are used followed by quantisation and entropy coding rather than the 2D transformation. The main advantage of this approach is that the computationally intensive process of motion compensation is not required, though it suffers from a number of major shortcomings including the requirement for a large frame memory which renders it inappropriate for real-time applications like video telephony, while blocking artefacts (as in the DCT) also make it unsuitable for low bit-rate coding. One other limitation, especially for the 3D sub-band based approaches is that the temporal filtering is not performed in the direction of the motion, and so temporal redundancies are not fully utilised to gain the highest compression efficiency. A solution of these problems is to combine the temporal components with motion compensation as proposed in Dufaux and Moscheni (1995).

Motion Compensation

The generic framework for motion compensated video coding techniques is given in Figure 7, with the primary difference with Figure 2 being the additional motion compensation block, where

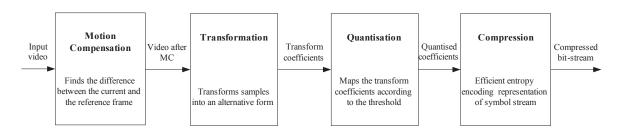


Figure 7. A generic waveform-based inter-frame video coder

the difference between the current and reference frame is predicted. To appreciate the development of motion compensation strategies, it is worth reviewing the conditional replenishment (Haskell, Mounts, & Candy, 1972) technique, which represents one of the earliest approaches to inter-frame coding, with the input frame separated into "changed" and "unchanged" regions with respect to a previously coded (reference) frame. Only the changed regions needed to be encoded while the unchanged regions were simply copied from the reference frame and for this purpose only the relative addresses of these regions were required to be transmitted. Coding of the changed regions can, in principle, be performed using any intra-frame coding technique, though improved performance can be achieved by predicting the changed regions using well established motion estimation (ME) and motion compensation (MC) processes. In fact, changes in a video are primarily due to the movement of objects in the sequence, so therefore by using an object motion model between frames, the encoder can estimate the motion that has occurred between the current and reference frames, in a process commonly referred to as ME. The encoder then uses this motion model and estimated motion information to move the content of the reference frame to provide a better prediction of the current frame, which is MC, and collectively the complete prediction process is known as motion compensated prediction. The reference frame used for ME may appear temporarily either before or after the current frame in the video sequence, with the two cases respectively being known as forward and backward prediction. Bidirectional prediction employs two frames (one each for forward and backward prediction) as the reference. As mentioned earlier, there are three different frame types used in the motion prediction process: I-frames which are intra-coded; P-frames which use either the previous or next I-frame as the reference frame; and B-frames which use the previous and next P-frames as the reference frames. The ME and MC-based coder is the most commonly used inter-frame coding method and is the bedrock for a range of popular video coding standards including MPEG-1 and MPEG-2 as well as the tele-conferencing coding H.261 and H.263 family.

In all these video coding standards, each frame is divided into regularly sized pixel blocks for ME (though the most recent H.264 standard also supports variable-sized MB), before block-by-block matching is performed. This block-matching motion estimation (BMME) strategy (Jain & Jain, 1981) is in fact the most commonly used ME algorithm, with the current frame first divided into blocks and then the motion of each block estimated by finding the best matching block in the reference frame. The motion of the

current block is then represented by a motion vector (MV) which is the linear displacement between this block and the best match in the reference frame. The computational complexity of MC mainly depends on the cost incurred by the searching technique for block matching, with various searching algorithms proposed in the literature, including the 2D logarithmic search (Jain & Jain, 1981), three-step search, diamond search (Tham, Ranganath, & Kassim, 1998), minimised maximum-error (Chen, Chen, Chiueh, & Lee, 1995), fast full search algorithms (Toivonen & Heikkilä, 2004), successive elimination algorithm (Li & Salari, 1995), and the simplex minimisation search (Al-Mualla, Canagarajah, & Bull, 2001). Following the MC step, all remaining steps are similar to those delineated for intraframe coding, that is, transformation/sub-band formation, quantisation, and compression using entropy coding.

Amongst the waveform inter-frame coding, block DCT-based methods are the most widely employed in the various standards, though the increasing requirement for scalability and higher compression ratios to enable very low bit-rate coding, has been the catalyst for wavelet-based image coders to be increasingly popular with considerable research being undertaken for instance into 3D sub-band coding (Ghanbari, 1991; Karlsson & Vetterli, 1988; Man, de Queiroz, & Smith, 2002; Ngan & Chooi, 1994; Podilchuk, Jayant, & Farvardin, 1995; Taubman & Zakhor, 1994) and motion compensated sub-band coding (Choi & Woods, 1999; Katto, Ohki, Nogaki, & Ohta, 1994).

Object-Based Video Coding

Object-based coding techniques can be viewed as an extension of second generation image coding techniques in the sense that a video object is defined in terms of visual primitives such as shape, colour, and texture. These techniques achieve very efficient compression by separating coherently moving objects from a stationary background, with each video object defined by its shape, texture, and motion. This enables content-based functionality such as the ability to selectively encode, decode, and manipulate specific objects in a video stream. MPEG-4 is the first object-based video coding standard to be developed and comprises of the following major steps:

- Moving object detection and segmentation,
- Shape coding,
- Texture coding,
- Motion estimation and compensation,
- Motion failure region detection, and
- Residual error encoding.

Figure 8 shows the overall block diagram for a generic object-based encoder. The first stage involves separating moving objects in the video sequence from the stationary background using fast and effective motion segmentation techniques, where the aforementioned ME and MC techniques are equally applicable. After the segmentation, shape coding is performed followed by ME and motion-compensated texture coding. The object segmentation, shape coding, motion estimation, and texture coding techniques are discussed in the MPEG-4 section.

The overall performance of the encoder is highly dependent on the performance of the segmentation approach employed. Following MC there may still be a significant amount of residual energy in certain areas of the image where MC alone was insufficient. The object segmentation is then re-applied to the compensated and original image to isolate these motion failure regions with high prediction errors (Bull et al., 1999). The residual information in the motion failure regions are then encoded using a block-based DCT scheme similar to the still image compression standard, for example, JPEG. The primary advantage of object-based coding is that it provides content-based flexibility and interactivity in video

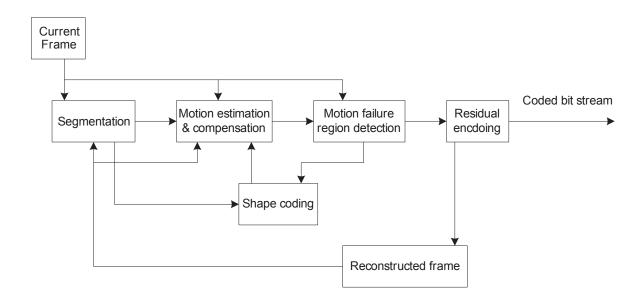


Figure 8. The overall block diagram for a generic object-based encoder

processing: encoding, decoding, manipulation, scalability, and also the interactive editing and error concealment.

Model-Based Video Coding

All compression techniques are based to some extent on an underlying model. The term model-based coding, however, refers specifically to an approach that seeks to represent the projected 2D image of a 3D scene using a semantic model. The aim is then to find an appropriate model together with the corresponding parameters, a task which can be divided into two main steps: *analysis* and *synthesis*. Model parameters are obtained by analysing the object's appearance and motion in the video scene, and these are then transmitted to a remote location, where a video display of the object is synthesised using pre-stored models at the receiver. In principle, only a small number of parameters are required to communicate the

changes in complex objects, thus enabling a very high CR. Analysis is by far the more challenging task due to the complexity of most natural scenes, such as a head and shoulder sequence (Aizawa, Harashima, & Saito, 1989; Li & Forchheimer, 1994) or face model (Kampmann, 2002). The synthesis block is easier to realise as it can build on techniques already developed for image synthesis in the field of computer graphics. For more detail about model-based video coding, the interested reader is referred to the comprehensive tutorial provided in Pearson (1995).

THE MPEG-4 VIDEO STANDARD

MPEG-4 is officially termed the *generic coding* of audio-visual objects, and the philosophy underpinning this latest video compression standard has shifted from the traditional perspective of considering a video sequence as simply being

Figure 9. Video object concepts, access, and manipulation (images from IMSI¹)



(a) Original video scene



(b) Segmented background object



(c) Segmented foreground object



(d) A different scene



(e) Edited scene, with the segmented object of (c) is inserted into scene (d)

a collection of rectangular video frames in the temporal dimension. Instead, MPEG-4 treats a video sequence as a collection of one or more video objects (VO) which are defined as a flexible entity that a user is allowed to access and manipulate. A VO may be arbitrarily shaped and exist for an arbitrary length of time, with a video scene made up of a background object and separate foreground objects. Consider the example in Figure 9, where the scene in Figure 9(a) is separated into two elements namely, the background (Figure 9(b)) and a single foreground object (Figure 9(c)), with both objects able to be independently used in other scene creations, so for instance, the VO in Figure 9(c) can be scaled and inserted into the new scene in Figure 9(d) giving the composite image in Figure 9(e). Clearly, for object manipulation, the object area is required to be defined and this leads to the challenging research area of object segmentation.

Object Segmentation

This has been the focus of considerable research and based upon the user interaction requirements, object segmentation methods usually fall into three distinct categories:

Manual Segmentation

This requires human intervention to manually identify the contour of each object in every source video frame, so it is very time-consuming and obviously only suitable for off-line video content. This approach however, can be appropriate for segmenting important visual objects that may be viewed by many users and/or re-used many times in differently composed sequences, such as cartoon animations.

Semi-Automatic Segmentation

Examples of this approach include ISO/IEC (2001) and Sun, Haynor, and Kim (2003), where a human

operator either inputs some rough initial objects that resemble the original objects or identifies the objects and even the objects' contour in a single frame. The segmentation algorithm then refines the object contours and tracks the objects through successive frames of the video sequence. Semiautomatic techniques are useful in applications where domain knowledge concerning the intended object is known as a priori. Semi-automatic segmentation algorithms may also require some different types of information, for instance, the number of objects that the user intends for as the input. One very good example of this type is the fuzzy clustering-based image segmentation algorithm (Ali, Dooley, & Karmakar, 2006), where the algorithm comes up with a number of segmented objects equal to the user input.

Fully-Automatic Segmentation

Paradoxically, semi-automatic segmentation has the potential to provide better results than the fully-automatic counterpart, since in the semi-automatic segmentation some relevant and domain specific information or an outline of the object is provided as an input, while in case of automatic segmentation, all of the information about an object is developed by the application itself which is both computationally expensive and sometimes can lead to erroneous results as the perceptual notion of what exactly is an object is not well defined. However, the main problem with semi-automatic segmentation is that it requires user inputs. Fully-automatic segmentation algorithms, such as those in Karmakar (2002) and Kim and Hwang (2002), attempt to perform a complete segmentation of the visual scene without any user intervention, based on for instance, spatial characteristics such as edges, colour, and distance, together with temporal characteristics such as the object motion between frames.

Again, as video is a sequence of still frames, a naïve approach, much similar to that of the

image coding, would be employing the image segmentation methods for video segmentation on a frame-by-frame basis. The image segmentation approaches can also be extended for video segmentation exploiting the temporal correlations. This can be done either by generalising the image segmentation algorithms for 3D signals, viewing the temporal as the third dimension or by utilising the motion of the objects in the consecutive frames. In recent times, detection and tracking of video objects has become an important and increasingly popular research area (Goldberger & Greenspan, 2006; Greenspan, Goldberger, & Mayer, 2004; Tao, Sawhney, & Kumar, 2002). As already mentioned a VO is primarily defined by its shape, texture, and motion; in MPEG-4 the VO shape and motion compensated texture are independently encoded. The following sections briefly discuss shape and texture coding paradigms for video objects.

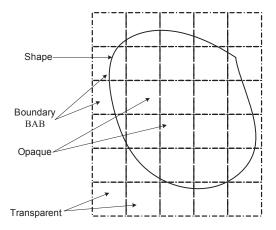
Shape Coding

In computer graphics, the shape of an object is defined by means of an α -map (plane) M_i of size $H \cdot V$ pels where $M_i = \{m_i(x, y) \mid 0 \le x \le H, 0 \le y\}$ $\langle V \rangle 0 \leq m_i \leq 255$ where H and V are respectively the horizontal and vertical frame dimensions. The grey scale shape M_i defines for each pixel whether it belongs to a particular video object or not, so if m(x, y) = 0, then pixel (x, y) does not belong to the shape. In the literature, for binary shapes $m_i(x)$, y) = 0 refers to the background, while $m_i(x, y)$ = 255 is a foreground object. Binary shape coders can be classified into two major classes: bitmapbased which encode every pixel as to whether it belongs to the object, and *contour*-based, which encodes the outline of the shape (Katsaggelos et al., 1998). The former are used in the fax standards G4 (Group 4) (CCITT, 1994) and JBIG (Joint Bilevel Image Experts Group) (ISO, 1992) and within the MPEG-4 coding standard, two bitmap-based shape coders have been developed: the non-adaptive context-based arithmetic encoder (CAE)

(Brady, Bossen, & Murphy, 1997), the adaptive modified modified-read (MMR) (Yamaguchi, Ida, & Watanabe, 1997) shape coder, and the newly developed digital straight line-based shape coding (DSLSC: digital straight line based shape coding) technique (Aghito & Forchhammer, 2006). Conversely, many different applications have fueled research into contour-based shape coding, including chain coders (Eden & Kocher, 1985; Freeman, 1961), parametric Bezier curve-based shape descriptors (Sohel, Karmakar, Dooley, & Arkinstall, 2005, 2007), polygon (H'otter, 1990; Katsaggelos et al., 1998; Kondi et al., 2004; Meier et al., 2000; O'Connell, 1997; Schuster et al., 1997; Sohel, Dooley, & Karmakar, 2006b) and B-spline based approximations (Jain, 1989; Katsaggelos et al., 1998; Meier et al., 2000; Kondi et al., 2004; Schuster et al., 1997; Schuster & Katsagellos, 1998). Within the MPEG-4 framework, two interesting contour-based shape coding strategies have been developed: (1) the vertex-based polynomial shape approximation based upon (Katsaggelos et al., 1998); and (2) the baseline-based shape coder (Lee et al., 1999). CAE is embedded in the MPEG-4 shape coder and so in the next section, both CAE and vertex-based operational rate-distortion optimal shape coding framework (Katsaggelos et al., 1998) will be outlined.

Context-based arithmetic coder: MPEG-4 has adopted a non-adaptive context-based arithmetic coder for shape information, since it allows regular memory access to the shape information and as consequence affords easier hardware implementation (Katsaggelos et al., 1998), and resourceful use of the existing block based motion compensation to exploit temporal redundancies. The binary α -planes are encoded by the CAE, while the grey scale α -planes are encoded by motion compensated DCT coding, which is similar to texture coding. For binary shape coding, a rectangular box enclosing the arbitrarily shaped Video Object Plane (VOP) is formed and the bounded box is divided into 16×16 macro-blocks, which are called binary-alpha-blocks (BAB). As

Figure 10. Binary α-block and its classification



illustrated in Figure 10, BAB are classified into three categories: transparent, opaque, and alpha or shape block. The transparent block does not contain any information about the object. The opaque block is located entirely inside an object, while the shape block is partially located in the object boundary, that is, part in the object and part background; thus these alpha-blocks are required to be processed by the encoder both for intra- and inter-coding modes.

CAE is a binary arithmetic encoder where the symbol probability is determined from the context of the neighbouring pixels based on templates, with Figure 11(a) and Figure 11(b) showing the templates for intra- and inter-modes respectively. In CAE, pixels are coded in scan-line order and in a three stage process:

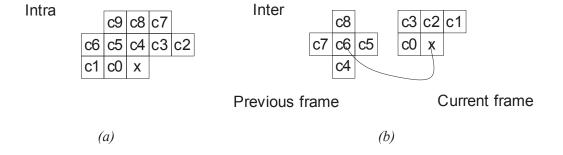
- Compute a context number based on the template and encoding mode.
- Index a probability table using the context number.
- Use the indexed probability to drive an arithmetic encoder.

In the case of the inter-mode, the alignment is performed after MC. For further details on CAE, the interested reader is referred to (Brady et al., 1997).

Vertex-Based Shape Coding

Vertex-based shape coding algorithms can be efficiently used in the high compression mobile communication applications and involve encoding the outline of an object's shape using either a polygon or B-spline based approximation for

Figure 11. Templates for defining those pels (x) to be encoded, c_i are pels in the neighbourhood of (x) within the templates: (a) Intra-mode, (b) Inter-mode (Note: Alignment is performed after MC)



lossy shape coding. The placement of vertices allows easy control of local variations in the shape approximation error. For lossless (zero geometric distortion) shape coding, the polygon approximation simply becomes that of a chain code (Lynn, Aram, Reddy, & Ostermann, 1997; Sikora, Bauer, & Makai, 1995). A series of vertex-based rate-distortion optimal shape coding algorithms has been proposed in Schuster et al. (1997), Katsaggelos et al. (1998), Meier et al. (2000), Kondi et al. (2004), and Sohel et al. (2006b), which employ weighted directed acyclic graph (DAG) based dynamic programming using polygons or parametric B-spline curves. The aim of these algorithms (Katsaggelos et al., 1998), is that for some prescribed admissible distortion, a shape contour is optimally encoded in terms of the number of bits, by selecting the set of control points that requires the lowest bit-rate and vice versa. These algorithms select the vertex on the shape-contour having the highest curvature as the starting vertex, and formulate the shape coding problem as finding the shortest path from the starting vertex to the last vertex of the shape-contour. The edge-weights are determined based on the admissible distortion and the bit requirement for the differential coding (Schuster et al., 1997) of the vertices. A number of performance enhancement techniques for these algorithms have been proposed in Sohel et al. (2006a, 2006b, 2007). The vertex-based *operational rate distortion* (ORD) optimal shape coding framework will now be briefly discussed.

The general aim of all these algorithms is that for some prescribed distortion, a shape contour is optimally encoded in terms of the number of bits, by selecting a set of *control points* (CP) that incurs the lowest bit-rate and vice versa. To select all CP that optimally approximate the boundary, a weighted DAG is formed and the minimum weight path is searched, with the start and end points of the boundary being respectively the source and destination vertices in the DAG. Both polygonal and quadratic B-spline based frameworks have been developed in Katsaggelos et al. (1998), with the admissible control points being considered as the vertices of DAG in the former case, and a trellis of admissible control points pairs are considered as the DAG vertices in the latter case. In this chapter only polygonal encoding will be discussed, while for B-spline based encoding the interested reader is referred to Katsaggelos et al. (1998), Kondi et al. (2004), and Sohel, Dooley, and Karmakar (2007).

Figure 12. DAG of five ordered admissible control points for polygonal encoding. There is a path in the DAG from a_i to a_i provided i < j.

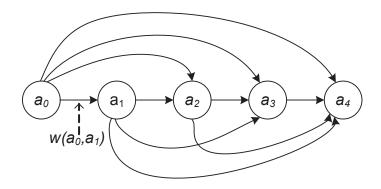
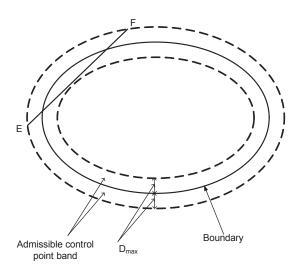


Figure 12 illustrates the DAG formation for polygonal encoding for five admissible CP namely a_0, a_1, a_2, a_3, a_4 with a_0 and a_4 being the start and end vertices respectively. Initially, admissible CP are restricted to be selected from only the boundary points, however this is subsequently relaxed by forming a fixed width band known as the admissible control point band (ACB) around the boundary, so points lying within this band can be admissible CP. This means that a point, though not on the boundary of an object, can still be selected as the CP and thereby further reduce the bit-rate. The framework presented in Katsaggelos et al. (1998) uses a single admissible distortion (D_{max}) , which is also used as the width of the ACB around the boundary as shown in Figure 13, so any point lying inside the ACB can be a CP for the shape approximation within the prescribed

Figure 13. Admissible control point band around a shape boundary



Algorithm 1. The polygonal ORD optimal shape coding algorithm

```
Inputs: B – the boundary; T_{\max} and T_{\min} – the admissible distortion bounds. Variables: MinRate(a_{i,m}) – current minimum bit-rate to encode up to vertex a_{i,m} from b_0; pred(a_{i,m}) – preceding CP of a_{i,m} (double subscripts is used to denote the ACB); N[i] – the number of vertices in A associated to b_i. Output: P – the ordered set of CP approximating B. Determine the admissible distortion T[i] for 0 < i < N_B - 1; Determine the sliding\ window\ width\ L[i] for 0 < i < N_B - 1 according to Sohel (2007);
```

```
Determine the sliding window width L[i] for 0 < i < N_B - 1 according to Sohel (2007); Form the ACB A using T[i] for 0 < i < N_B - 1 according to Sohel (2007); Initialise MinRate(a_{0,0}) with the total bits required to encode the first boundary point b_0; Set MinRate(a_{k,n}), 0 < k < N_B, 0 \le n < N[k] to infinity; FOR each vertex a_{i,m}, 0 \le i < N_B - 1, 0 \le m < N[i] FOR each vertex a_{j,n}, i < j \le \min\{(i + L[i])(N_B - 1)\}, 0 \le n < N[j] Check the edge-distortion dist(a_{i,m}, a_{j,n}); IF dist(a_{i,m}, a_{j,n}) maintains the admissible distortion THEN Determine bit-rate r(a_{i,m}, a_{j,n}) and edge-weight w(a_{i,m}, a_{j,n}); IF ((MinRate(a_{i,m}) + w(a_{i,m}, a_{j,n})) < MinRate(a_{j,n}) + w(a_{i,m}, a_{j,n}); pred(a_{i,n}) = a_{i,m};
```

Obtain P with properly indexed values from pred.

admissible distortion. The notion of fixed admissible distortion has been generalised by Kondi et al. (1998) and Kondi, Melnikov, and Katsaggelos (2001, 2004), where the admissible distortion T[i] for each individual boundary point is determined from the prescribed admissible distortion bounds T_{max} and T_{min} which are the maximum and minimum distortion respectively. To determine the admissible distortion for a boundary point, either the gradient of image intensity (Kondi, 1998; Kondi et al., 2001, 2004) or the curvature (Kondi et al., 2001) at that point is considered. The admissible distortions are determined such that boundary points with a high image intensity gradient or high curvature have lower smaller admissible distortion and vice versa. As a result, sharp features or high intensity gradient parts of a shape are better protected compared with low image gradient or flatter shape portions from an approximation perspective. Within the variable admissible distortion framework, the philosophy of ACB has also been generalised in Sohel, Dooley, and Karmakar (2006b, 2007) to support variable ACB so it can fully exploit the variable admissible distortion in reducing the bit-rate for a prescribed admissible distortion pair, with the width of the ACB for each boundary point being set equal to the admissible distortion. These works also defined the ACB width for individual boundary points for the B-spline based framework.

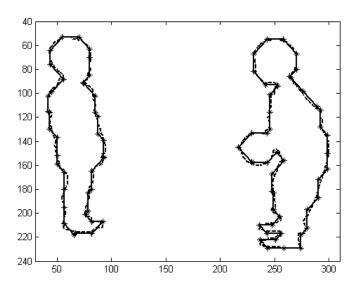
Each edge in the DAG is considered in the optimisation process for approximating the shape, though for a particular edge it is required to check whether all boundary points in between the end points of the edge maintain the admissible distortion, so in the example in Figure 13, edge *EF* does maintain the admissible distortion. If the admissible distortion is maintained, this edge is further considered in the rate-distortion optimisation process, so it becomes crucial to determine the level of distortion of each boundary point from the candidate DAG edge. The ORD framework in Katsaggelos et al. (1998) employs either the *shortest absolute distance* or alternatively the *distor-*

tion band approach; while in Kondi et al. (2004) the tolerance band which is the generalisation of the distortion band is used. The performance of these various distortion measurement techniques can be further enhanced by adopting the recently introduced accurate distortion metric in Sohel et al. (2006b) and the computationally efficient chord-length-parameterisation based approach in Sohel, Karmakar, and Dooley (2007b). Moreover, these algorithms use a *sliding window* (SW) which enforces the encoder to follow the shape boundary and also limits the search space for the next CP within the SW-width (Katsaggelos et al., 1998). The SW provides three fold benefit to the encoder: (1) avoid the trivial solution problems, (2) preserve the sharp features of the shape, and (3) computationally speed up the process. However, since the SW constricts the search space for the next CP within SW-width, the optimality of the algorithms is compromised in a bit-rate sense (Sohel, Karmakar, & Dooley, 2006). The techniques (Sohel, Dooley, & Karmakar, 2007; Sohel, Karmakar, & Dooley, 2006) formally define the most appropriate and suitable SW-width for the rate-distortion constrained algorithms.

After the distortion checking process, the edgeweight is determined which is *infinite* if the edge fails to maintain the admissible distortion for all the relevant boundary points. If the edge passes the distortion check, the edge weight is the bit required to encode the edge differentially, so for example, the edge weight $w(a_i, a_j)$ is equal to the edge bit-rate $r(a_i, a_j)$ which is the total number of bits required to differentially encode the vertex a_j given that vertex a_i is already encoded. For vertex encoding purposes, a combination of orientation dependent chain code and logarithmic run-length code are used in these algorithms (Schuster et al., 1997).

To summarise, the vertex-based ORD optimal shape coding algorithms seek to determine and encode a set of CP to represent a particular shape within prescribed RD constraints. Assume boundary $B = \{b_0, b_1, ..., b_{N_{p-1}}\}$ is an ordered set

Figure 14. Polygonal approximation results for the 1^{st} frame of the Kids sequence with $T_{max} = 3$ and $T_{min} = 1$ pel (Legends: Solid line—Approximated boundary; Dashed line—Original boundary; Asterisk—CP)



of shape points, where N_B is the total number of points and $b_0 = b_{N_B-1}$ for a closed boundary. $P = \{p_0, p_1, ..., p_{N_P-1}\}$ is an ordered set of CP used to approximate B, where N_P is the total number of CP and $P \subseteq A$, where A is the ordered set of vertices in ACB. For a representative example, the ORD polygonal shape coding algorithm for determining the optimal P for boundary B within RD constraints is formalised in Algorithm 1, with the detailed analysis provided in Schuster et al. (1997), Katsaggelos et al. (1998), Meier et al. (2000), Kondi et al. (2004), and Sohel, Dooley, and Karmakar (2007).

Some experimental results from this ORD shape coding framework are now presented.

Figure 14 shows the subjective results upon the 1st frame of the popular multiple-object *Kids* test video test sequence with L_{∞} distortion bounds of $T_{\max} = 3$ and $T_{\min} = 1pel$ respectively. In the experiments, the curvature-based approach of Kondi et al. (2001) was adopted from which it is visually apparent that those shape regions

having high curvature are well preserved in the approximation with lower admissible distortion, while in the smoother shape regions, the higher admissible distortion is fully utilised to ensure that the bit-rate requirement is minimised, while upholding the prescribed distortion bounds.

Figure 15 shows the corresponding rate-distortion (RD) results for the 1st frame Kids sequence. The bit-rate is plotted along the ordinate in bit units, while the MPEG-4 relative area error (D_n) is shown along the abscissa in percentiles. The curve reveals that according to the ORD theory as the distortion decreases the required bit-rate increases and vice versa, however as anticipated, a *diminishing rate of return* trend is observed at higher distortion values. At lower D_n values, a much higher bit-rate reduction is achieved for only a small increase in the distortion, while at higher distortion values, a change in distortion generates only a comparatively moderate improvement in the bit-rate.

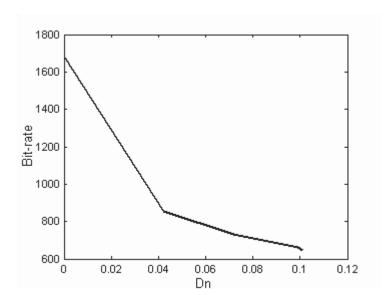


Figure 15. Rate-distortion results upon the 1st frame of Kids sequence

Motion Compensation

Motion estimation (ME) and compensation (MC) methods in MPEG-4 are very similar to those employed in the other standards, though the primary difference is that block-based ME and MC are adapted to the arbitrary-shape VOP structure. Since the size, shape, and location of a VOP can change from one instance to another, the absolute (frame) coordinate system is applied to reference each VOP. For opaque blocks, motion is estimated using the usual block matching method, however for the BAB, motion is estimated using a modified block matching algorithm, namely polygon matching where the distortion is measured using only those pixels in the current block. Padding techniques are used to define the values of pels where the ME and MC may require to access pels from outside the VOP. For the BAB in intramode, it is padded with horizontal and vertical repetition. For the inter-alpha blocks, not only are alpha blocks repeatedly padded, but also the region outside the VOP within the block is padded with zeros.

Texture Coding

Texture is an essential part of a video object and which is reflected by it being assigned more bits than the shape in the coded bit-stream (Bandyopadhyay & Kondi, 2005; Kaup, 1998; Kondi et al., 2001). Each intra VOP and MC inter VOP is coded using a 8×8 block DCT, with the DCT performed separately on each of the luminance and chrominance planes. The opaque alpha blocks are encoded with block-based DCT, with BAB padding techniques used as outlined in the previous section, while all transparent blocks are skipped and so not encoded.

Padding removes any abrupt transitions within a block and hence reduces the number of significant DCT co-efficients. Since the number of opaque pixels in the 8×8 blocks of some of the boundary alpha blocks is usually less than 64 pixels, it is more efficient if these opaque pixels are DCT coded without padding in a technique known as shape adaptive DCT (Sikora & Makai, 1995). In Kondi et al. (2004), a joint optimal texture and shape encoding strategy was proposed based on a combination of the shape adaptive DCT and vertex-based ORD optimal shape coding framework. While block transforms such as DCT are widely considered to be the best practical solution for MC video coding, the DWT is particularly effective in coding still images. Recent research findings including, the MPEG-4 visual (MPEG-4: Part 2) use the DWT (Daubechies, 1990) as the core basis texture compression tools, moreover shape adaptive DWT (Li & Li, 1995) has been employed in the texture coding algorithms as, for example, in the joint contour-based shape and texture coding strategy proposed in Bandyopadhyay et al. (2005).

THE H.264 STANDARD

H.261 (ITU-T, 1993) was the first widely-used standard for videoconferencing and was primarily developed to support video telephony and conferencing applications over ISDN circuit-switched networks, hence the constraint that H.261 could only operate at multiples of 64Kbps, though it was specifically designed to offer computationally simple video coding at these bit-rates. H.261 employed a DCT model with integer-accuracy MC, while the next version known as H.263 (ITU-T, 1998) provides improved compression performance with half-pel MC accuracy and is able to provide high video quality at bit-rates lower than 30 kbps, as well as operating over both circuit- and packet-switched networks. The MPEG and the Video Coding Experts Group

(VCEG) subsequently developed the *advanced* video coding (AVC) standard H.264 (ISO/IEC, 2003) that aims to provide better video compression. H.264 does not explicitly define a CODEC as was the trend in the earlier standards, but rather defines the syntax of an encoded video bit-stream together with the method of decoding the bit-stream. The main features of H.264 as mentioned in Richardson (2003).

It supports multi-frame MC using previously-encoded frames as references in a more flexible way than other standards. H.264 permits up to 32 reference frames to be used in some cases, while in prior standards this limit was typically one or two only in the case of B-frames. This particular feature allows modest improvements in bit rate and quality in most video sequences, though for certain types of scenes, particularly rapidly repetitive flashing, back-and-forth scene cuts² and newly revealed background areas, significant bit-rate reductions are achievable. The computational cost of MC however, is increased with the increase in the search space for the best matched block.

It introduces the *tree-structured motion com*pensation. While using the same basic principle of block-based motion compensation that has been employed since the original H.261 standard was established, a major departure is the support for a range of different sized blocks from the usual fixed 8×8 DCT-based block size used in MPEG-1, MPEG-2, and H.263, through to the smaller 4×4 and larger 16×16 block sizes, with various intermediate combinations including 16×8 and 4×8. The tree structure comes from the actual method of partitioning the MB into motion compensated sub-blocks. Choosing a large block size, such as 16×16 or 8×16, means a smaller number of bits are required to represent the MV and partition choice, however the corresponding motion compensated residual signal may be large, especially in areas of high detail. Conversely, choosing a small block size, that is, 4×4 or 4×8, results in a much lower energy in the motion compensated

residual signal, but a larger number of bits will be required to represent the MV and partition choice. The variable block-size notion of H.264 has been illustrated in with the example in Figure 16. The choice of the MB partition size is therefore crucial to the efficiency of the compression. H.264 adopts what may be thought of as very much the intuitive approach, with a larger-sized partition being appropriate for predominantly smooth or homogeneous regions, while a smaller size is used in areas of high detail. The "best" partition size decision is made during encoding such that the residual energy and MV are minimised.

It also uses fractional (one-quarter) pel accuracy for MC and incorporates weighted prediction that allows an encoder to specify the use of scaling and offset when performing MC. This provides a significant benefit in performance in special cases, as for example, in fade-to-black, fade-in, and cross-fade transitions.

To reduce the blocking artefacts of DCT-based coding techniques, an in-loop de-blocking filters are employed. Moreover, the filtered MB is used in subsequent motion-compensated prediction of future frames, resulting in a lower residual error after prediction. Figure 17 presents an example of the effect of the de-blocking filter in the decoding loop in reducing the visual blocky artefacts.

It incorporates either a context-adaptive binary arithmetic coder (CABAC) or a context-adaptive variable-length coder (CAVLC). CABAC is an intelligent technique that compresses in a lossless manner the syntax elements in a video stream knowing their probabilities in a given context. CAVLC is a low-complexity alternative to CABAC for the coding of quantised transform co-efficient values. It is more elaborate and more efficient than the methods typically employed to code the quantised transform co-efficients in previous designs.

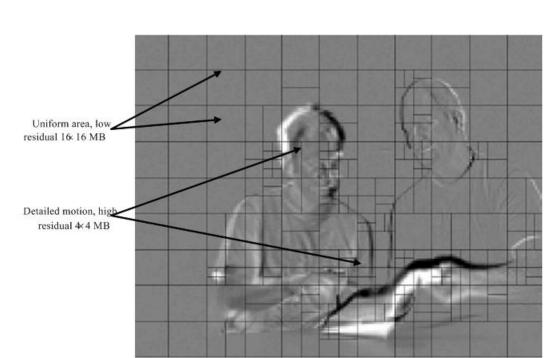
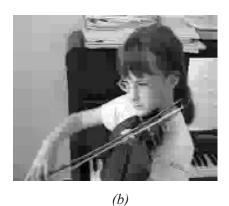


Figure 16. Tree structured variable block motion compensation in H.264

Figure 17. Effect of de-blocking filter: (a) reference frame, (b) reconstructed frame without a filter, (c) reconstructed frame with filter





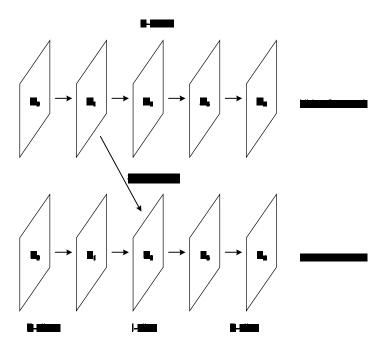


A network abstraction layer (NAL) is defined that allows the same video syntax to be used in many network environments, including features such as sequence parameter sets (SPS) and picture parameter sets (PPS) providing greater robustness and flexibility than previous standards.

Switching slices (known as SP and SI slices) features facilitate an encoder with efficient switching between different video bit-streams. Consider for instance a video decoder receiving multiple bit-rate streams across the Internet. The decoder attempts to decode the highest rate stream, but

may need, if data throughput falls, to switch automatically to decoding a lower bit-rate stream. The example in Figure 18 explains the switching using I-slice. Having already decoded frames A_0 and A_1 the decoder wishes to switch across to the other bit-stream at B_2 . This however is a P-frame (synonymously P-slice) with no reference at all to the previous P-frames in video stream A. This means a solution is to code B_2 as an I-frame so it does not involve prediction and can therefore be coding independently. However, this results in an increase in the overall bit-rate, since the coding

Figure 18. Switching between video streams using SI-slices



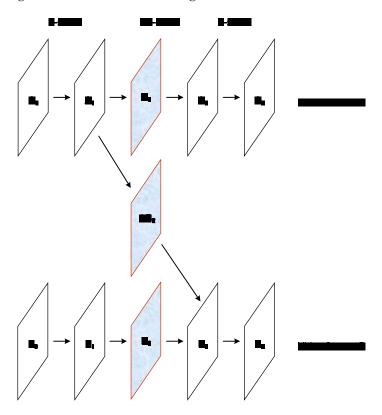
efficiency of an I-frame is much lower than that of a P-frame. H.264 provides an elegant solution to this problem through the use of SP-slices. Figure 19 illustrates an example of switching using SP-slices. At the switching points (frame 2 in both Streams A and B) which would be at regular intervals in the coded sequence, there are now three SP-slices involved (highlighted), which are all encoded using motion compensation prediction, so they will be more efficient than I-frame coding. SP-slice A₂ is decoded using reference frame A₁ and SP-slice B₂ is decoded using reference frame B₁; however the key to this technique is SP-slice AB₂—the switching slice. This is generated in such a manner that it is able to decode using motion-compensated prediction frame A₁ to produce slice B₂. This means the decoder output frame B, is the same regardless of

whether it is directly decoding B₁ or A₁ followed by AB₂. A reciprocal arrangement means an extra SP-slice BA₂ will also be included to facilitate switching from bit-stream B to A, though this is not shown in the diagram. While an extra SP-slice will be required at every switching point, the additional overhead this incurs is more than offset by not requiring the decoding of I-frames at these switching points.

All H.264 frames are numbered which allows the creation of *sub-sequences* (enabling temporal scalability by the optional inclusion of extra pictures between other pictures), and the detection and concealment of losses of even entire pictures (which can occur due to network packet losses or channel errors).

It ranks picture count which keeps the ordering of the pictures and the values of samples in the

Figure 19. Switching between video streams using SP-slices



decoded pictures isolated from timing information, allowing timing information to be carried and controlled/changed separately by a system without affecting decoded picture content.

ERROR RESILIENT VIDEO CODING

There are many causes in mobile communication systems whereby the encoded data may experience error, which readily become magnified because in the compressed data, one single bit usually represents much more information than it did in the original video. Error resilient techniques therefore have become very important and have

attracted the attention of researchers. This section provides the reader with a lucid insight into some of the requirements, challenges, and various methods that exist for error resilient video coding in mobile communications.

Requirements of an Error Resilient CODEC

An error-resilient video coding system should be able to provide the following functionality:

• Error Detection: This is the most fundamental requirement. The system decoder can encounter a syntactic error, that is, an illegal code word of variable or fixed length, or a semantic error such as an MPEG-4 decoder generating a shape that is not enclosed. However, the error may not be detected until some point after it actually occurs, so error localisation is an important prerequisite.

- Error Localisation: When an error has been detected, the decoder has to resynchronise with the bit-stream without skipping too many bits, for example, via the use of additional resynchronisation markers (Ebrahimi, 1997).
- Data Recovery: After error localisation, data recovery attempts to recover some information from the bit-stream between the location of the detected error and the determined resynchronisation point, thus minimising information loss. Reversible variable length coding (RVLC) (VLC that can be decoded both in a forward and backward directions, a double-ended decodable code) (Wen & Villasenor, 1997) can be explicitly used for this purpose.
- Error Concealment: Finally, error concealment tries to hide the effects of the erroneous bit-stream by replacing lost information by meaningful data, that is, copying data from the previous frame into the current frame. The smaller the spatial and temporal extent of the error, the more accurate the concealment strategies can be.

Error Resilience at the Encoder

There are both pre- and post-processing techniques available for error correction. In the former, the encoder plays a pivotal role by introducing a controlled level of redundancy in the video bit-stream to enhance the error resilience, by sacrificing some coding efficiency. Resynchronisation of the code-words inserts unique markers into the encoded bit-stream so enabling the decoder to localise the detected error in the received bit-stream. When an error is detected at

the decoder, it can skip the remaining bits until it locates the next resynchronisation marker. The more recent H.263+video coding standard adopts this particular strategy. An alternative approach is the error resilience entropy encoder (Redmill, 1994; Redmill & Kingsbury, 1996), which takes variable length blocks of data and rearranges them into fixed-length slots. It has the advantage that when an error is detected, the decoder simply jumps into the start of the next block so there is no need for resynchronisation keywords, though the drawback is that the decoder discards all data until the next resynchronisation code or starting point of the next block is reached, even though much of the discarded data may have been correctly received. Reversible-VLC (Wen et al., 1997) coding also known as double-ended, decodes the received bits in reverse order instead of blindly discarding them when a resynchronisation code or start of the next block has been received, so that the decoder can attempt to recover and utilise those bits which were simply discarded with other coding schemes. It is noteworthy to mention that RVLC also keeps on proceeding with the incoming bit-streams. There are some other common forward techniques such as layered coding with prioritisation (Ghanbari, 1989), multiple-description coding (Kondi, 2005; Vaishampayan, 1993), and interleaved coding (Zhu, Wang, & Shaw, 1993) which are all designed for very low bit-rate video coding and so are well suited for video communications over the mobile networks.

Error Resilience at the Decoder

For either the post-processing or concealment techniques, the decoder plays the primary role in attempting to mask the effects of errors by providing a subjectively acceptable approximation of the original data using the received data. Error concealment is an ill-posed problem since there is no unique solution for a particular problem. Depending on the information used for concealment, these are divided into three major categories:

spatial, temporal, and hybrid techniques. Spatial approaches use the inherently high spatial correlation of video signals to conceal erroneous pels in a frame, using information from correctly received and/or previously concealed neighbouring pels within the same frame (Ghanbari & Seferides, 1993; Salama et al., 1995). Temporal methods exploit the latent inter-frame correlation of video signals and conceal damaged pels in a frame again using the information from correctly received and/or previously concealed pels within the reference frame (Narula & Lim, 1993; Wang & Zhu, 1998), while hybrid techniques seek to concomitantly exploit both spatial and temporal correlations in the concealment strategy (Shirani, Kossentini, & Ward, 2000).

There also exist some *interactive* approaches to error concealment, including the automatic repeat request (ARQ), sliding window, refreshment based on feedback, and selective repeat. A comprehensive review of these techniques can be found in Girod and Farber (1999). In all these cases, the encoder and decoder cooperate to minimise the effects of transmission errors, with the decoder using a feedback channel to inform the encoder about the erroneous data. Based on this information the encoder adjusts its operation to combat the effects of the errors. For shape coding techniques, there are number of efficient error concealment techniques (Schuster & Katsaggelos, 2006; Schuster, Katsaggelos, & Xiaohuan, 2004; Soares & Pereira, 2004, 2006), some of them use the parametric curves, such as Bezier curves (Soares et al., 2004) and Hermite splines (Schuster et al., 2004). While the techniques in Schuster et al. (2004) and Soares et al. (2004) are designed to conceal the errors by exploiting only the spatial information within intra-mode, the techniques by Schuster et al. (2006) and Soares et al. (2006) work in inter-mode and also utilises the temporal information.

Moreover, Bezier curve theory has been extended, by incorporating the localised control point information so that it reduces the gap between the curve and the control polygon, as the half-way shifting Bezier curve, the dynamic Bezier curve, and the enhanced Bezier curves respectively in Sohel, Dooley, and Karmakar (2005a, 2005b), Sohel et al. (2005), and Sohel, Karmakar, and Dooley (2007). To improve the respective performance, these new curves can be seamlessly embedded into algorithms, for instance, the ORD optimal shape coding frameworks and the shape error concealment techniques, where currently the B-splines and the Bezier curves are respectively used. While these techniques conceal the shape error independent of the underlying image information, an image dependent shape error concealment technique has been proposed by Sohel, Karmakar, and Dooley (2007b), which utilises the underlying image information in the concealment process and obtains a more robust performance.

DISTRIBUTED VIDEO CODING

This is a novel paradigm in the video coding applications where instead of doing the compression at the encoder, it is either partially or wholly performed at the decoder. It is not, however, a new concept as the origins of this interesting idea can be traced back to the 1970s and the information-theoretic bounds established by Slepian and Wolf (1973) for distributed lossless coding, and also by Wyner and Ziv (1976) for lossy coding with decoder side information.

Distributed coding exploits the source statistics in the decoder, so the encoder can be very simple, at the expense of a more complex decoder, so the traditional balance of a complex encoder and simple decoder is essentially reversed. A highlevel schematic visualization of distributed video coding is provided in Figure 20, where Figure 20(a) and (b) respectively contrast the conventional and distributed video coding paradigms. In a conventional video coder, all compression is undertaken at the encoder which must therefore

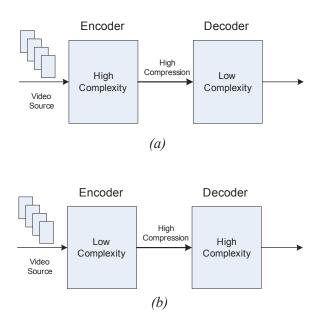


Figure 20. High-level view of: (a) conventional and (b) distributed video coding

be sufficiently powerful to cope with this requirement. Many applications however, may require a dual system, that is, lower-complexity encoder at the possible expense of a higher-complexity decoder. Examples of such systems include: wireless video sensors for surveillance, wireless personal-computer cameras, mobile cameraphones, disposable video cameras, and networked camcorders. In all these cases, with conventional coding the compression must be implemented at the camera where memory and computation are scarce, so a framework is mandated where the decoder performs all the high complexity tasks, which is the essence of distributed video coding, namely to distribute the computational workload incurred in the compression between the encoder and decoder. This approach is actually more robust than conventional coding techniques in the sense of handling packet loss or frame dropping which are fairly common events in hostile mobile channels. Girod, Aaron, Rane, and Rebollo-Monendero (2005) and Puri and Ramchandran (2002) have pioneered this research field and provide a good starting point for the interested reader on this contemporary topic.

FUTURE TREND

It was mentioned early that the VQEG have been striving for sometime to establish a single quality metric that truly represents the best of what is perceived by the HVS within the image and video coding domains. When eventually this is formally devised, it will command considerable attention from the research community as they rapidly endeavour to ensure that new findings and outcomes are fully compliant with this quality metric and that the performance of algorithm and system is superior from this new perspective (Wu et al., 2006).

Bandwidth allocation and reservation will inevitably, as it has in recent times, remain a very challenging research topic especially for mobile technologies (Bandyopadhyay et al., 2005; Kamaci, Altunbasak, & Mersereau, 2005; Sun, Ahmad, Li, & Zhang, 2006; Tang, Chen, Yu, & Tsai, 2006; Wang, Schuster, & Katsaggelos, 2005), and this can only be expected to burgeon in the future as the next series of mobile generation technologies, namely 3G and 4G mature.

Finally, as discussed previously, distributed video coding has gained increasing popularity among researchers as it affords a number of potential advantages for mobile operation over traditional and well-established compression strategies. Much work, however, remains in both revisiting and innovating new compression techniques for this distributed coding framework (Girod et al., 2005).

CONCLUSION

This chapter has presented an overview of video coding techniques for mobile communications where low bit-rate and computationally efficient coding are mandated in order to cope with the stringent bandwidth and processing power limitations. It has provided the reader with a comprehensive review of contemporary research work and developments in this rapidly burgeoning field. The evolution of high compression, intra-frame coding strategies from JPEG to JPEG2000 (version 2) and very low bit-rate inter-frame coding from block-based motion compensated MPEG-2 to the flexible object-based MPEG-4 coding have been outlined. Moreover, the main features of the AVC/H.264 have also been outlined together with a discussion on emerging distributed video coding techniques. It also has provided a functional discussion on the physical significance of the various video coding quality metrics that are considered essential for mobile communications

in conjunction with the aims and interests of the Video Quality Expert Group.

REFERENCES

Aghito, S. M., & Forchhammer, S. (2006). Context based coding of bi-level images enhanced by digital straight line analysis. *IEEE Transactions on Image Processing*, *15*(8), 2120-2130.

Aizawa, K., Harashima, H., & Saito, T. (1989). Model-based analysis synthesis image coding (MBASIC) system for a person's face. *Signal Processing: Image Communication*, *1*(2), 139-152.

Al-Mualla, M. E., Canagarajah, C. N., & Bull, D. R. (2001). Simplex minimization for single-and multiple-reference motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(12), 1209-1220.

Al-Mualla, M. E., Canagarajah, C. N., & Bull, D. R. (2002). *Video coding for mobile communications: Efficiency, complexity, and resilience*. Amsterdam: Academic Press.

Ali, M. A., Dooley, L. S., & Karmakar, G. C. (2006). *Object based segmentation using fuzzy clustering*. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, May 15-19.

Atta, R., & Ghanbari, M. (2006). Spatio-temporal scalability-based motion-compensated 3-D subband/DCT video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(1), 43-55.

Bandyopadhyay, S. K., & Kondi, L. P. (2005). Optimal bit allocation for joint contour-based shape coding and shape adaptive texture coding. *International Conference on Image Processing (ICIP), I,* Genoa, Italy, September 11-14 (pp. 589-592).

Barnsley, M. F. (1988). *Fractals everywhere*. Boston: Academic Press.

Barthel, K. U., Voye, T., & Noll, P. (1993). *Improved fractal image coding*. Picture Coding Symposium, Lausanne, Switzerland, March 17-19.

Brady, N. (1999). MPEG-4 standardized methods for the compression of arbitrarily shaped video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, *9*(8), 1170-1189.

Brady, N., Bossen, F., & Murphy, N. (1997). Context-based arithmetic encoding of 2D shape sequences. *International Conference on Image Processing (ICIP), I,* Washington, DC, October 26-29 (pp. 29-32).

Bull, D. R., Canagarajah, N. C., & Nix, A. (1999). *Insights into mobile multimedia communications: Signal processing and its applications.* San Diego, CA: Academic Press.

CCITT. (1994). Facsimile coding schemes and coding functions for group 4 facsimile apparatus. CCITT Recommendation T.6.

Chen, M. J., Chen, L. G., Chiueh, T. D., & Lee, Y. P. (1995). A new block-matching criterion for motion estimation and its implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, *5*(3), 231-236.

Choi, S. J., & Woods, J. W. (1999). Motion-compensated 3-D subband coding of video. *IEEE Transactions on Image Processing*, 8(2), 155-167.

Chou, P. A., Lookabaugh, T., & Gary, R. M. (1989). Entropy constrained vector quantisation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(1), 31-42.

Clarke, R. J. (1995). *Digital compression of still images and video*. London: Academic Press.

Crochiere, R. E., Webber, S. A., & Flanagan, F. L. (1976). Digital coding of speech in sub-bands. *IEEE International Conference on Acoustics*,

Speech, and Signal Processing, 1, April (pp. 233-236).

Daubechies, I. (1990). The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, *36*(5), 961-1005.

Distasi, R., Nappi, M., & Riccio, D. (2006). A range/domain approximation error-based approach for fractal image compression. *IEEE Transactions on Image Processing*, 15(1), 89-97.

Dufaux, F., & Moscheni, F. (1995). Motion estimation techniques for digital TV: A review and a new contribution. *Proceedings of the IEEE*, 83(6), 858-876.

Ebrahimi, T. (1997). *MPEG-4 video verification model version 8.0*. International Standards Organization, ISO/IEC JTC1/SC29/WG11 MPEG97/N1796.

Eden, M., & Kocher, M. (1985). On the performance of a contour coding algorithm in the context of image coding. Part i: Contour segment coding. *Signal Processing*, 8, 381-386.

Freeman, H. (1961). On the encoding of arbitrary geometric configurations. *IRE Trans. Electronic Computers, EC-10,* 260-268.

Ghanbari, M. (1989). Two-layer coding of video signals for VBR networks. *IEEE Journal on Selected Areas in Communications*, 7(5), 771-781.

Ghanbari, M. (1991). Subband coding algorithms for video applications: Videophone to HDTV-conferencing. *IEEE Transactions on Circuits and Systems for Video Technology, 1*(2), 174-183.

Ghanbari, M. (1999). Videocoding: An introduction to standard codecs. *IEE Telecommunications Series*, 42.

Ghanbari, M., & Seferides, V. (1993). Cellloss concealment in ATM video codecs. *IEEE*

Transactions on Circuits and Systems for Video Technology, 3(3), 238-247.

Girod, B., Aaron, A., Rane, S., & Rebollo-Monendero, D. (2005). Distributed video coding. *Proceedings of the IEEE*, *93*(1), 71-83.

Girod, B., & Farber, N. (1999). Feedback-based error control for mobile video transmission. *Proceedings of the IEEE: Special Issue on Video for Mobile Multi-media*, *97*(10), 1707-1723.

Goldberger, J., & Greenspan, H. (2006). Context-based segmentation of image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3), 463-468.

Greenspan, H., Goldberger, J., & Mayer, A. (2004). Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3), 384-396.

Haskell, B. G., Mounts, F. W., & Candy, J. C. (1972). Interframe coding of videotelephone pictures. *Proceedings of the IEEE*, 60(7), 792-800.

H'otter, M. (1990). Object-oriented analysis-synthesis coding based on moving two-dimensional objects. *Signal Processing*, 2, 409-428.

ISO. (1992). Coded representation of picture and audio information—Progressive bi-level image compression. ISO Draft International Standard 11544.

ISO/IEC 14496-2. (2001). Coding of audio-visual objects – Part 2: Visual. Annex F.

ISO/IEC 14496-10 & ITU-T Rec. (2003). *H.264, Advanced video coding.*

ITU-T Recommendation H.261. (1993). Video CODEC for audiovisual services at px64 kbit/s.

ITU-T Recommendation H.263. (1998). *Video coding for low bit rate communication, Version* 2.

Jacobs, E. W., Fisher, Y., & Boss, R. D. (1992). Image compression: A study of the iterated transform method. *Signal Processing*, 29(3), 251-263.

Jacquin, A. E. (1992). Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, *I*(1), 18-30.

Jain, A. K. (1989). Fundamentals of digital image processing. Englewood Cliffs, NJ: Prentice-Hall.

Jain, J., & Jain, A. (1981). Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communication, COMM-29*(12), 1799-1808.

Johnston, J. D. (1980). A filter family designed for use in quadratic mirror filter banks. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 291-294).

Kamaci, N., Altunbasak, Y., & Mersereau, R. M. (2005). Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models. *IEEE Transactions on Circuits and Systems for Video Technology, 15*(8), 994-1006.

Kampmann, M. (2002). Automatic 3-D face model adaptation for model-based coding of videophone sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, *12*(3), 172-182.

Karlsson, G., & Vetterli, M. (1988). Three-dimensional subband coding of video. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New York, April (pp. 1100-1103).

Karmakar, G. C. (2002). *An integrated fuzzy rule-based image segmentation framework*. PhD Thesis. Gippsland School of Computing and Information Technology. Monash University: Australia.

- Katsaggelos, A. K., Kondi, L. P., Meier, F. W., Ostermann, J., & Schuster, G. M. (1998). MPEG-4 and rate-distortion-based shape-coding techniques. *Proceedings of the IEEE*, 86(6), 1126-1154.
- Katto, J., Ohki, J., Nogaki, S., & Ohta, M. (1994). A wavelet codec with overlapped motion compensation for very low bit-rate environment. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3), 328-338.
- Kaup, A. (1998). Object-based texture coding of moving video in MPEG-4. *IEEE Transactions on Circuits and Systems for Video Technology*, *9*(1), 5-15.
- Kim, C., & Hwang, J.-N. (2002). Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2), 122-129.
- Kondi, L. P. (2005). Transactions letters. A rate-distortion optimal hybrid scalable/multiple description video codec. *IEEE Transactions on Circuits and Systems for Video Technology, 15*(7), 921-927.
- Kondi, L.P., Meier, F. W., Schuster, G. M., & Katsaggelos, A. K. (1998) Joint optimal object shape estimation and encoding. *SPIE Visual Communication and Image Processing*, San Jose, California, USA, January (pp. 14-25).
- Kondi, L. P., Melnikov, G., & Katsaggelos, A. K. (2001). Jointly optimal coding of texture and shape. *International Conference on Image Processing (ICIP)*, *3*, Thessaloniki, Greece, October 7-10 (pp. 94-97).
- Kondi, L. P., Melnikov, G., & Katsaggelos, A. K. (2004). Joint optimal object shape estimation and encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, *14*(4), 528-533.
- Kunt, M., Ikonomopoloulos, A., & Kocher, R. (1985). Second generation image coding

- techniques. *Proceedings of the IEEE*, 73(4), 549-574.
- Lee, S., Cho, D., Cho, Y., Son, S., Jang, E., Shin, J., & Seo, Y. (1999). Binary shape coding using baseline-based method. *IEEE Transactions on Circuits and Systems for Video Technology*, *9*(1), 44-58.
- Li, H., & Forchheimer, R. (1994). Two-view facial movement estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3), 276-287.
- Li, J., & Lei, S. (1997). Rate-distortion optimized embedding. In *Proc. Picture Coding Symposium*, Berlin, Germany, September (pp. 201-206).
- Li, S., & Li, W. (1995). Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(5), 725-743.
- Li, W., & Salari, W. (1995). Successive elimination algorithm for motion estimation. *IEEE Transactions on Image Processing*, 4(1), 105-107.
- Linde, Y., Buzo, A., & Gary, R. M. (1980). An algorithm for vector quantization. *IEEE Transactions on Communication*, 28(1), 84-95.
- Lynn, L. H., Aram, J. D., Reddy, N. M., & Ostermann, J. (1997). Methodologies used for evaluation of video tools and algorithms in MPEG-4. *Signal Processing: Image Communication*, *9*(4), 343-365.
- Man, H., de Queiroz, R., & Smith, M. (2002). Three-dimensional subband coding techniques for wireless video communications. *IEEE Transactions on Circuits and Systems for Video Technology*, *12*(3), 386-397.
- Meier, F. W., Schuster, G. M., & Katsaggelos, A. K. (2000). A mathematical model for shape coding with B-splines. *Signal Processing: Image Communications*, 15(7-8), 685-701.

- Nanda, S., & Pearlman, W. S. (1992). Tree coding of image subbands. *IEEE Transactions on Signal Processing*, 1(2), 133-147.
- Narula, A., & Lim, J. S. (1993). Error concealment techniques for an all-digital high-definition television system. In *Proc. SPIE Conf. Visual Commun. and Image Proc.*, Chicago, IL, May (pp. 304-315).
- Ngan, K. N., & Chooi, W. L. (1994). Very low bit rate video coding using 3D subband approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3), 309-316.
- Øien, G. E. (1993). L_2 -optimal attractor image coding with fast decoder convergence. PhD thesis. Trondheim, Norway.
- O'Connell, K. J. (1997). Object-adaptive vertex-based shape coding method. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1), 251-255.
- Ordentlich, E., Weinberger, M., & Seroussi, G. (1998). A low-complexity modeling approach for embedded coding of wavelet coefficients. *IEEE Data Compression Conference (DCC)*, Snowbird, Utah, March 30-April 1 (pp. 408-417).
- Pearson, D. E. (1995). Developments in model-based video coding. *Proceedings of the IEEE*, 83(6), 892-906.
- Podilchuk, C., Jayant, N., & Farvardin, N. (1995). Three dimensional subband coding of video. *IEEE Transactions on Image Processing*, 4(2), 125-39.
- Puri, R., & Ramchandran, K. (2002). PRISM: A new robust video coding architecture based on distributed compression principles. *Allerton Conference on Communication, Control, and Computing*, Allerton, IL, October.
- Rabbani, M., & Jones, P. W. (1991). *Digital image compression techniques*. Bellingham, Washington: SPIE Optical Engineering Press.

- Redmill, D. W. (1994). *Image and video coding for noisy channels*. PhD thesis. University of Cambridge. Signal Processing and Communications Laboratory.
- Redmill, D. W., & Kingsbury, N. G. (1996). The EREC: An error resilient technique for coding variable-length blocks of data. *IEEE Transactions on Image Processing*, *5*(4), 565-574.
- Richardson, I. E. (2003). *H.264 and MPEG-4 video compression*. Chichester: John Wiley & Sons.
- Said, A., & Pearlman, W. (1996). A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3), 243-250.
- Salama, P., Shroff, N. B., Coyle, E. J., & Delp, E. J. (1995). Error concealment techniques for encoded video streams. *IEEE International Conference on Image Processing (ICIP)*, Washington, DC, October 23-26 (pp. 9-12).
- Schuster, G. M., & Katsaggelos, A. K. (1997). Rate-distortion based video compression: Optimal video frame compression and object boundary encoding. Boston: Kluwer Academic Publishers.
- Schuster, G. M., & Katsaggelos, A. K. (1998). An optimal boundary encoding scheme in the rate distortion sense. *IEEE Transactions on Image Processing*, 7(1), 13-26.
- Schuster, G. M., & Katsaggelos, A. K. (2006). Motion compensated shape error concealment. *IEEE Transactions on Image Processing*, *15*(2), 501-510.
- Schuster, G. M., Katsaggelos, A. K., & Xiaohuan, L. (2004). Shape error concealment using Hermite splines. *IEEE Transactions on Image Processing*, 13(6), 808-820.
- Shapiro, J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE*

- *Transactions on Signal Processing, 41*(12), 3445-3462.
- Shirani, S., Kossentini, F., & Ward, R. (2000). A concealment method for video communications in an error-prone environment. *IEEE Journal on Selected Areas in Communications, 18*(6), 1122-1128.
- Sikora, T., Bauer, S., & Makai, B. (1995). Efficiency of shape adaptive transforms for coding of arbitrarily shaped image segments. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(3), 254-258.
- Sikora, T., & Makai, B. (1995). Shape-adaptive DCT for generic coding of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(3), 59-62.
- Slepian, J. D., & Wolf, J. K. (1973). Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, *IT-19*, 471-480.
- Soares, L. D., & Pereira, F. (2004). Spatial shape error concealment for object-based image and video coding. *IEEE Transactions on Image Processing*, 13(4), 586-599.
- Soares, L. D., & Pereira, F. (2006). Temporal shape error concealment by global motion compensation with local refinement. *IEEE Transactions on Image Processing*, 15(6), 1331-1348.
- Sohel, F. A., Dooley, L. S., & Karmakar, G. C. (2005a). A dynamic Bezier curve model. *International Conference on Image Processing (ICIP), II*, Genoa, Italy, September (pp. 474-477).
- Sohel, F. A., Dooley, L. S., & Karmakar, G. C. (2005b). *A novel half-way shifting Bezier curve model*. IEEE Region 10 Conference (Tencon), Melbourne, Australia, November.
- Sohel, F. A., Dooley, L. S., & Karmakar, G. C. (2006a). Accurate distortion measurement for

- generic shape coding. *Pattern Recognition Letters*, 27(2), 133-142.
- Sohel, F. A., Dooley, L. S., & Karmakar, G. C. (2006b). *Variable width admissible control point band for vertex based operational-rate-distortion optimal shape coding algorithms*. International Conference on Image Processing (ICIP), Atlanta, GA, October.
- Sohel, F. A., Dooley, L. S., & Karmakar, G. C. (2007). New dynamic enhancements to the vertex-based rate-distortion optimal shape coding framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(10).
- Sohel, F. A., Karmakar, G. C., & Dooley, L. S. (2005). An improved shape descriptor using Bezier curves. First International Conference on Pattern Recognition and Machine Intelligence (PReMI). Lecture Notes in Computer Science, 3776, Kolkata, India, December (pp. 401-406).
- Sohel, F. A., Karmakar, G. C., & Dooley, L. S. (2006). *Dynamic sliding window width selection strategies for rate-distortion optimal vertex-based shape coding algorithms*. International Conference on Signal Processing (ICSP), Guilin, China, November 16-20.
- Sohel, F. A., Karmakar, G. C., & Dooley, L. S. (2007a). Fast distortion measurement using chord-length parameterisation within the vertex-based rate-distortion optimal shape coding framework. *IEEE Signal Processing Letters*, *14*(2), 121-124.
- Sohel, F. A., Karmakar, G. C., & Dooley, L. S. (2007b). Spatial shape error concealment utilising image-texture. *IEEE Transactions on Image Processing* (revision submitted).
- Sohel, F. A., Karmakar, G. C., & Dooley, L. S. (2007c). *Bezier curve-based character descriptor considering shape information*. IEEE/ACIS International Conference on Computer and Information Science (ICIS), Melbourne, Australia, July.

- Sohel, F. A., Karmakar, G. C., Dooley, L. S., & Arkinstall, J. (2005). Enhanced Bezier curve models incorporating local information. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IV,* Philadelphia, PA, March 18-23 (pp. 253-256).
- Sohel, F. A., Karmakar, G. C., Dooley, L. S. & Arkinstall, J. (2007). Quasi Bezier curves integrating localised information. *Pattern Recognition* (in press).
- Sun, Y., Ahmad, I., Li, D., & Zhang, Y.-Q. (2006). Region-based rate control and bit allocation for wireless video transmission. *IEEE Transactions on Multimedia*, 8(1), 1-10.
- Sun, S., Haynor, D., & Kim, Y. (2003). Semiautomatic video object segmentation using v-snakes. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1), 75-82.
- Tang, C.-W., Chen, C.-H., Yu, Y.-H., & Tsai, C.-J. (2006). Visual sensitivity guided bit allocation for video coding. *IEEE Transactions on Multimedia*, 8(1), 11-18.
- Tao, H., Sawhney, H. S., & Kumar, R. (2002). Object tracking with Bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 75-89.
- Taubman, D. (2000). High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, *9*(7), 1158-1170.
- Taubman, D. S., & Marcellin, M. W. (2002). *JPEG2000: Image compression fundamentals, standards and practice*. Boston: Kluwer Academic Publishers.
- Taubman, D. S., & Zakhor, A. (1994). Multirate 3-D subband coding of video. *IEEE Transactions on Image Processing*, *3*(4), 572-88.

- Tekalp, A. M. (1995). *Digital video processing*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs: NJ.
- Tham, J. Y., Ranganath, S., Ranganath, M., & Kassim, A. A. (1998). A novel unrestricted center-biased diamond search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(4), 369-377.
- Toivonen, T., & Heikkilä, J. (2004). Fast full search block motion estimation for H.264/avc with multilevel successive elimination algorithm. In *Proc. International Conference on Image Processing (ICIP)*, *3*, Singapore, October (pp. 1485-1488).
- Topiwala, P. N. (1998). Wavelet image and video compression. Boston: Kluwer Academic Publishers.
- Vaishampayan, V. A. (1993). Design of multiple description scalar quantizers. *IEEE Transaction on Information Theory*, *39*(3), 821-834.
- VQEG. (1998). Final report from the Video Quality Expert Group on the validation of objective models of video quality assessment.
- Wade, N., & Swanston, M. (2001). *Visual perception: An introduction (2nd ed.)*. London: Psychology Press.
- Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM*, 34(4), 30-44.
- Wang, H., Schuster, G. M., & Katsaggelos, A. K. (2005). Rate-distortion optimal bit allocation for object-based video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(9), 113-1123.
- Wang, Y., & Zhu, Q. (1998). Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5), 974-997.

Welch, T. A. (1984). A technique for high performance data compression. *IEEE Computer*, 17(6), 8-19.

Wen, J., & Villasenor, J. D. (1997). A class of reversible variable length codes for robust image and video coding. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 2, Washington, DC, October (pp. 65-68).

Witten, I., Neal, R., & Cleary, J. (1987). Arithmetic coding for data compression. *Communication of the ACM*, 30(6), 520-540.

Woods, J., & O'Neil, S. (1986). Subband coding of images. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 34(5), 1278-1288.

Wu, H. R., & Rao, K. R. (2006). *Digital video image quality and perceptual coding.* Boca Raton, FL: CRC Press: Taylor and Francis.

Wyner, A. D., & Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory, IT-22*(1), 1-10.

Yamaguchi, N., Ida, T., & Watanabe, T. (1997). A binary shape coding method using modified MMR. In *Proc. Special Session on Shape Coding (ICIP97), I,* Washington, DC, October (pp. 504-508).

Zhu, Q.-F., Wang, Y., & Shaw, L. (1993). Coding and cell-loss recovery in DCT-based packet video. *IEEE Transactions on Circuits and Systems for Video Technology*, *3*(3), 238-247.

Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, *IT-23*(3), 337-343.

ENDNOTES

- IMSI's Master Photo Collection, 1895 Francisco Blvd. East, San Rafael, CA 94901-5506, USA.
- Cut is defined as a visual transition created in editing in which one shot is instantaneously replaced on screen by another.