Chapter 2

VISUAL DATA FORMATS

1. Image and Video Data

Digital visual data is usually organised in rectangular arrays denoted as frames, the elements of these arrays are denoted as pixels (picture elements). Each pixel is a numerical value, the magnitude of the value specifies the intensity of this pixel. The magnitude of the pixels varies within a predefined range which is classically denoted as "bitdepth", i.e. if the bitdepth is 8 bit, the magnitude of the pixel varies between 0 and $2^8 - 1$ (8 bpp means 8 bits per pixel). Typical examples are binary images (i.e. black and white images) with 1 bpp only or grayvalue images with 8 bpp where the grayvalues vary between 0 and 255.

Colour is defined by using several frames, one for each colour channel. The most prominent example is the RGB representation, where a full resolution frame is devoted to each of the colours red, green, and blue. Colour representations closer to human perception differentiate among luminance and colour channels (e.g. the YUV model).

Video adds a temporal dimension to the purely spatially oriented image data. A video consists of single frames which are temporally ordered one after the other (see Fig. 2.1). A single video frame may again consist of several frames for different colour channels.

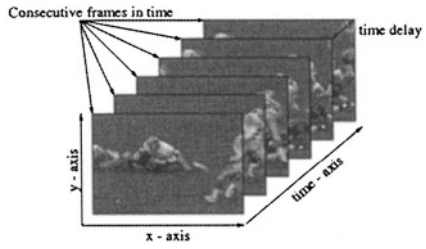

Figure 2.1. Frame-structure of video (football sequence)

Visual data constitutes enormous amounts of data to be stored,

transmitted, or processed. Therefore, visual data is mostly subjected to compression algorithms after capturing (or digitisation). Two big classes of compression algorithms exist:

- Lossless compression: after having decompressed the data, it is numerically identical to the original values.
- Lossy compression: the decompressed data is an approximation of the original values.

Lossy algorithms achieve much higher compression ratios (i.e. the fraction between original filesize and the size of the compressed file) as compared to the lossless case. However, due to restrictions imposed by some application areas, lossless algorithms are important as well (e.g. in the area of medical imaging lossless compression is mandatory in many countries due to legislative reasons). However, in the multimedia area lossy compression algorithms are more important, the most distinctive classification criterion is whether the underlying integral transform is the discrete cosine transform (DCT) or the wavelet transform.

2. DCT-based Systems

2.1 JPEG

The baseline system of the JPEG standard [169,110] operates on 8×8 pixels blocks onto which a DCT is applied. The resulting data are quantised using standardised quantisation matrices, subsequently the quantised coefficients are scanned following a zig-zag order (which orders the data in increasing frequency), the resulting vector is Huffman and runlength encoded (see right side of Fig. 2.4).

The JPEG standard also contains an extended system where several progressive modes are defined (see section 1.4.1 (chapter 5)) and a lossless codes which uses not DCT but is entirely DPCM (difference pulse coded modulation) based.

2.2 MPEG Video Coding (MPEG-1,2,4)

The main idea of MPEG motion compensated video coding [99,60] is to use the temporal **and** spatial correlation between frames in a video sequence [153] (Fig. 2.1) for predicting the current frame from previously (de)coded ones. Some frames are compressed in similar manner to JPEG compression, which are random access points to the sequence, these frames are called I-frames. All other frames are predicted from decoded I-frames – in case a bidirectional temporal prediction is done the corresponding frames are denoted B-frames, simple unidirectional prediction leads to P-frames. Since this prediction fails in some regions (e.g. due to occlusion), the residual between this prediction

and the current frame being processed is computed and additionally stored after lossy compression. This compression is again similar to JPEG compression but a different quantisation matrix is used.

Because of its simplicity and effectiveness block-matching algorithms are widely used to remove temporal correlation [53]. In block-matching motion compensation, the scene (i.e. video frame) is classically divided into non-overlapping "block" regions. For estimating the motion, each block in the current frame is compared against the blocks in the search area in the reference frame (i.e. previously encoded and decoded frame) and the motion vector (d_1, d_2) corresponding to the best match is returned (see Fig. 2.2). The "best" match of the blocks is identified to be that match giving the minimum mean square error (MSE) of all blocks in search area defined as

$$MSE(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{(n_1,n_2)\in\mathcal{B}} [s_k(n_1, n_2) - \hat{s}_{k-l}(n_1 + d_1, n_2 + d_2)]^2$$

where $\mathcal{B}$ denotes a $N_1 * N_2$ block for a set of candidate motion vectors (d_1, d_2), s is the current frame and $\hat{s}$ the reference frame.

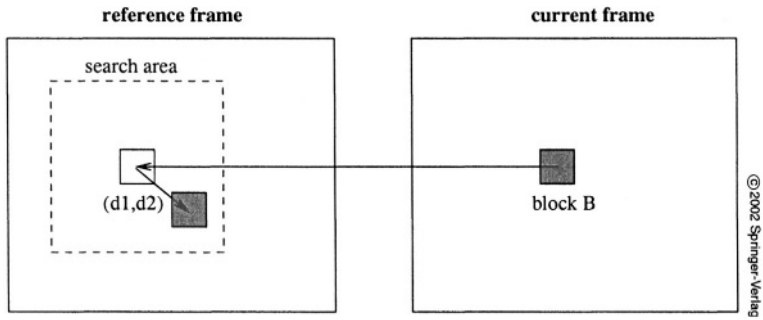

Figure 2.2. Block-Matching motion estimation

The algorithm which visits all blocks in the search area to compute the minimum is called full search. In order to speed up the search process, many techniques have been proposed to reduce the number of candidate blocks. The main idea is to introduce a specific search pattern which is recursively applied at the position of the minimal local error. The most popular algorithm of this type is called "Three Step Search" which reduces the computational amount significantly at the cost of a suboptimal solution (and therefore a residual with slightly more energy). The block giving the minimal error is stored describing the prediction in term of a motion vector which describes the displacement of

the block. The collection of all motion vectors of a frame is called motion vector field.

MPEG-1 has been originally defined for storing video on CD-ROM, therefore the data rate and consequently the video quality is rather low. MPEG,MPEG-2 [60] is very similar from the algorithmic viewpoint, however the scope is shifted to TV broadcasting and even HDTV. The quality is much higher as compared to MPEG-1, additionally methodologies have been standardised to enable scalable video streams and error resilience functionalities.

MPEG-4 [40, 124] extends the scope of the MPEG standards series to natural and synthetic (i.e. computer generated) video and provides technologies for interactive video (i.e. object-based video coding). The core compression engine is again similar to MPEG-2 to provide backward compatibility to some extent. Finally, MPEG-4 AVC (also denoted H.264 in the ITU standards series) increases compression efficiency significantly as compared to MPEG-4 video at an enormous computational cost [124].

2.3 ITU H.26X Video Conferencing

The ITU series of video conferencing standards is very similar to the MPEG standards, however, there is one fundamental difference: video conferencing has to meet real-time constraints. Therefore, the most expensive part of video coding (i.e. motion compensation) needs to be restricted. As a consequence, H.261 defines no B-frames in contrast to MPEG-1 and H.263 is also less complex as compared to MPEG-2. In particular, H.261 and H.263 offer better quality at low bitrates as compared to their MPEG counterparts. H.261 has been defined to support video conferencing over ISDN, H.263 over PSTN which implies the demand for even lower bitrates in H.263. The latest standard in this series is H.264 which has been designed by the JVT (joint video team) and is identical to MPEG-4 AVC. This algorithm uses a 4×4 pixels integer transform (which is similar to the DCT) and multi-frame motion compensation. Therefore, this algorithm is very demanding from a computational point of view.

3. Wavelet-based Systems

Image compression methods that use wavelet transforms [154] (which are based on multiresolution analysis – MRA) have been successful in providing high compression ratios while maintaining good image quality, and have proven to be serious competitors to DCT based compression schemes.

A wide variety of wavelet-based image compression schemes have been reported in the literature [62, 86], ranging from first generation systems which are similar to JPEG only replacing the DCT by wavelets to more complex techniques such as vector quantisation in the wavelet domain [7, 26,10], adap-

tive transforms [31, 160, 175], and edge-based coding [52]. Second generation wavelet compression schemes try to take advantage of inter subband correlation – the most prominent algorithms in this area are zerotree encoding [135, 81] and hybrid fractal wavelet codecs [142, 30]. In most of these schemes, compression is accomplished by applying a fast wavelet transform to decorrelate the image data, quantising the resulting transform coefficients (this is where the actual lossy compression takes place) and coding the quantised values taking into account the high inter-subband correlations.

The fast wavelet transform (which is used in signal and image processing) can be efficiently implemented by a pair of appropriately designed Quadrature Mirror Filters (QMF). Therefore, wavelet-based image compression can be viewed as a form of subband coding. A 1-D wavelet transform of a signal s is performed by convolving s with both QMF's and downsampling by 2; since s is finite, one must make some choice about what values to pad the extensions with [150]. This operation decomposes the original signal into two frequency-bands (called subbands), which are often denoted as coarse scale approximation (lowpass subband) and detail signal (highpass subband). Then, the same procedure is applied recursively to the coarse scale approximations several times (see Figure 2.3.a).

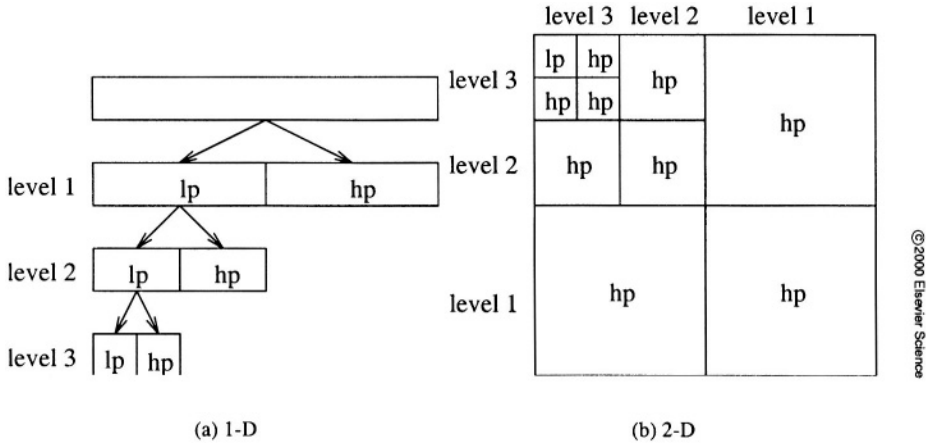

Figure 2.3. 1-D and 2-D wavelet decomposition: lowpass (lp) and highpass (hp) subbands, decomposition levels (level 1 – level 3)

The classical 2-D transform is performed by two separate 1-D transforms along the rows and the columns of the image data, resulting at each decomposition step in a low pass image (the coarse scale approximation) and three detail images (see Figure 2.3.b); for more details see [91].

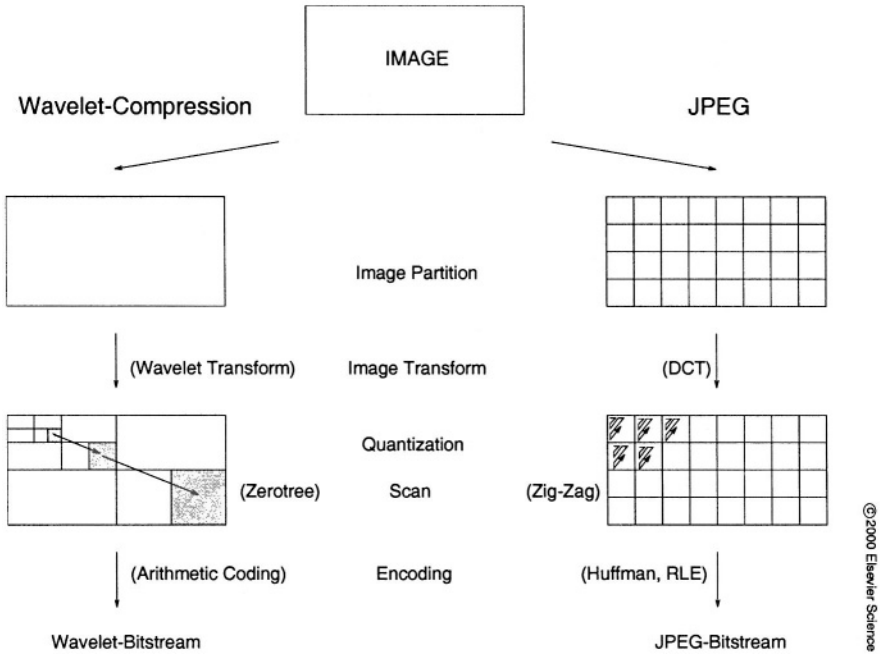

Figure 2.4. Comparison of DCT-based and wavelet-based compression schemes

Fig. 2.4 shows the differences between DCT and wavelet based schemes – whereas the differences are restricted to the transform stage for first generation schemes, also the scan order and entropy encoding is different for second generation systems.

3.1 SPIHT

It can be observed that the coefficients calculated by a wavelet decomposition contain a high degree of spatial self similarity across all subbands. By considering this similarity, a more efficient coefficient representation can be obtained which is exploited by all second generation wavelet coding schemes. SPIHT [126] uses a spatial orientation tree which is shown in Figure 2.5. This data structure is very similar to the zerotree structure used by the EZW zerotree algorithm [135], each value in the wavelet multiresolution pyramid is assigned to a node of the tree.

Three lists are used to represent the image information: The LIS (list of insignificant sets), the LIP (list of insignificant pixels), and the LSP (list of significant pixels). The latter list contains the sorted coefficients which are stored. The following algorithm iteratively operates on these lists thereby adding and

deleting coefficients $c_{i,j}$ to/from the lists (where μ_n denotes the number of coefficients which have their most significant bit within bitplane n):

1 output $n = \lfloor \log_2 \left(max_{(i,j)} \{|c_{i,j}|\} \right) \rfloor$ to the decoder.

2 output μ_n, followed by the pixel coordinates and sign of each of the μ_n coefficients such that $2^n \leq |c_{i,j}| < 2^{n+1}$ **(sorting pass);**

3 output the n-th most significant bit of all the coefficients $|c_{i,j}| \geq 2^{n+1}$ (i.e., those that had their coordinates transmitted in previous sorting passes), in the same order used to send the coordinates **(refinement pass);**

4 decrement n by 1, and go to step 2.

The SPIHT codec generates an embedded bitstream and is optimised for encoding speed. SPIHT is not a standard but a proprietary commercial product which has been the state of the art codec each new image compression system was compared to for several years. The SMAWZ codec [78] used is some sections of this book is a variant of SPIHT which uses bitplanes instead of lists to ease processing and to save memory accesses. Additionally, SMAWZ generalises SPIHT to wavelet packet subband structures and anisotropic wavelet decomposition schemes.

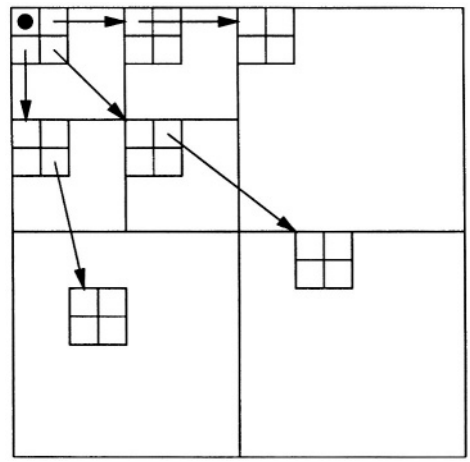

Figure 2.5. Spatial Orientation Tree

3.2 JPEG 2000

The JPEG 2000 image coding standard [152] is based on a scheme originally proposed by Taubman and known as EBCOT ("Embedded Block Coding with Optimised Truncation" [151]). The major difference between previously proposed wavelet-based image compression algorithms such as EZW or SPIHT (see [154]) is that EBCOT as well as JPEG 2000 operate on independent, non-overlapping blocks which are coded in several bit layers to create an embedded, scalable bitstream. Instead of zerotrees, the JPEG 2000 scheme depends on a per-block quad-tree structure since the strictly independent block coding strategy precludes structures across subbands or even code-blocks. These independent code-blocks are passed down the "coding pipeline" shown in Fig.

2.6 and generate separate bitstreams (Tier-1 coding). Transmitting each bit layer corresponds to a certain distortion level. The partitioning of the available bit budget between the code-blocks and layers ("truncation points") is determined using a sophisticated optimisation strategy for optimal rate/distortion performance (Tier-2 coding).

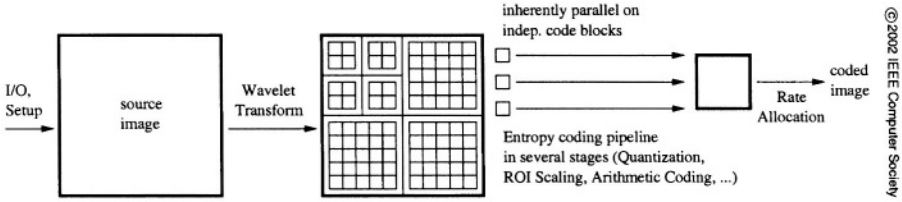

Figure 2.6. JPEG 2000 coding pipeline

The main design goals behind EBCOT and JPEG 2000 are versatility and flexibility which are achieved to a large extent by the independent processing and coding of image blocks [23], and of course to provide a codec with a better rate-distortion performance than the widely used JPEG, especially at lower bitrates. The default for JPEG 2000 is to perform a five-level wavelet decomposition with 7/9-biorthogonal filters and then segment the transformed image into non-overlapping code-blocks of no more than 4096 coefficients which are passed down the coding pipeline.

Two JPEG 2000 reference implementations are available online: the JJ2000 codec (see `http://jj2000.epfl.ch`) implemented in JAVA and the Jasper C codec (see `http://www.ece.ubc.ca/~madams`).

4. Further Techniques

4.1 Quadtrees

Quadtree compression partitions the visual data into a structural part (the quadtree structure) and colour information (the leave values). The quadtree structure shows the location and size of each homogeneous region, the colour information represents the intensity of the corresponding region. The generation of the quadtree follows the splitting strategy well known from the area of image segmentation. Quadtree image compression comes in lossless as well in lossy flavour, the lossy variant is obtained in case the homogeneity criterion is less stringent. This technique is not competitive from the rate distortion efficiency viewpoint, but it is much faster than any transform based compression technique.

4.2 Fractal Coding

Fractal image compression [47,11] exploits similarities within images. These similarities are described by a contractive transformation of the image whose fixed point is close to the image itself. The image transformation consists of block transformations which approximate smaller parts of the image by larger ones. The smaller parts are called ranges and the larger ones domains. All ranges together (range-pool) form a partition of the image. Often an adaptive quadtree partition is applied to the image. The domains can be selected freely within the image and may overlap (domain-pool). For each range an appropriate domain must be found. If no appropriate domain can be found (according to a certain error measure and a tolerance) the range blocks are split which reduces the compression efficiency.

Although fractal compression exhibits promising properties (like e.g. fractal interpolation and resolution independent decoding) the encoding complexity turned out to be prohibitive for successful employment of the technique. Additionally, fractal coding has never reached the rate distortion performance of second generation wavelet codecs.

4.3 Vector Quantisation

Vector quantisation [3] exploits similarities between image blocks and an external codebook. The image to be encoded is tiled into smaller image blocks which are compared against equally sized blocks in an external codebook. For each image block the most similar codebook block is identified and the corresponding index is recorded. From the algorithmic viewpoint, the process is similar to fractal coding, therefore fractal coding is sometimes referred to as vector quantisation with internal codebook. Similar to fractal coding, the encoding process involves a search for an optimal block match and is rather costly, whereas the decoding process in the case of vector quantisation is even faster since it is a simple lookup table operation.

4.4 Lossless Formats: JBIG, GIF, PNG

Whereas most lossy compression techniques combine several algorithms (e.g., transformation, quantisation, coding), lossless techniques often employ a single compression algorithm in rather pure form. Lossless JPEG as described before employs a DPCM codec. GIF and PNG both use dictionary coding as the underlying technique – LZW coding in the case of GIF and LZSS coding in the case of PNG. JBIG uses context-based binary arithmetic coding for compressing bitplanes. For some details on these lossless compression techniques see [61].