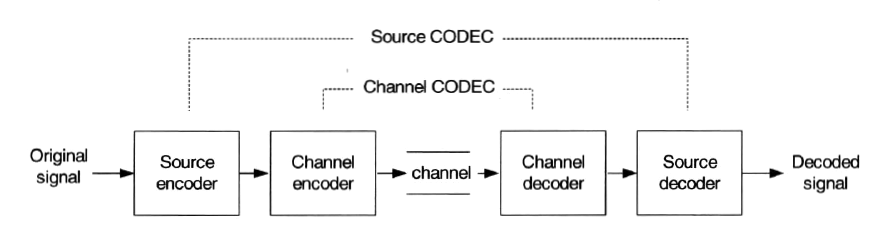
**Chapter 3 Image and Video**

**3.1 Introduction**

**3.2 Image & Video compression**

A device or program that compresses a signal is an *encoder* and a device or program that decompresses a singal is a *decoder*. An enCOder/DECoder pair is a *CODEC.*

Typical communication system



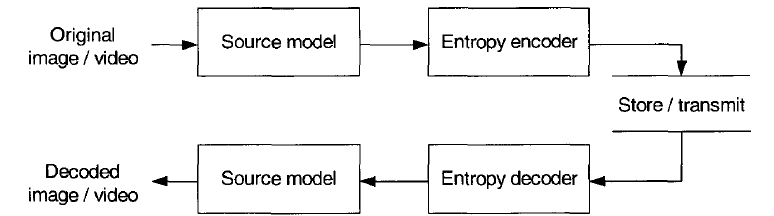
The source coded signal is then encoded further to add error protection (*channel coding*) prior to transmission over a channel.

General-purpose compression CODECs are available that are designed to encode and compress data containing *statistical redundancy*.

Compression is achieved by reducing the statistical redundancy in the text file. This type of general-purpose CODEC is known as an *entropy CODEC*(熵编解码). An entropy encoder performs best with data values that have a certain degree of independence(decorrelated data).

Photographic images and sequences of video frames are **NOT** amenable to compression using general-purpose CODECs.

Image or video CODEC



Neighbouring samples(pixels) within an image or a video frame tend to be highly correlated and so there is significant spatial redundancy(空间冗余).

Neighbouring regions within successive video frames also tend to be highly correlated (temporal redundancy) (时间冗余).

A source model may take advantage of subjective redundancy(主观冗余)， exploiting the sensitivity of the human visual system to various characteristics of images & video.

Examples of image and video source models includes:

DPCM(Differential Pulse Code Modulation)(差分脉冲编码调制)

Each sample or pixel is predicted from one or more previously

transmitted samples.

DPCM may be applied spatially(using adjacent pixels in the

same frame) and/or temporally(using adjacent pixels in a previous

frame to form the prediction) and gives modest compression with

low complexity.

Transform Coding(变换编码)

The image sample are transformed into another domain and

are represented by transform *coefficients*.

The aim of transform coding is to reduce this correlation,

ideally leaving a small number of visually significant transform

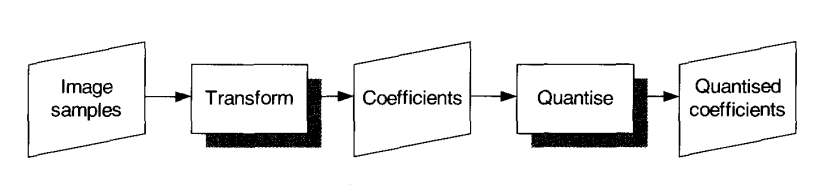
coefficients and a large number of insignificant coefficients.

The transform process itself doesn't achieve compression:

a lossy quantisation process in which the insignificants are

removed,leaving behind a small number of significant coefficients,

usually follow it.



Motion-compensated Prediction(运动补偿预测编码)

Similar principle to DPCM, the encoder forms a model of the

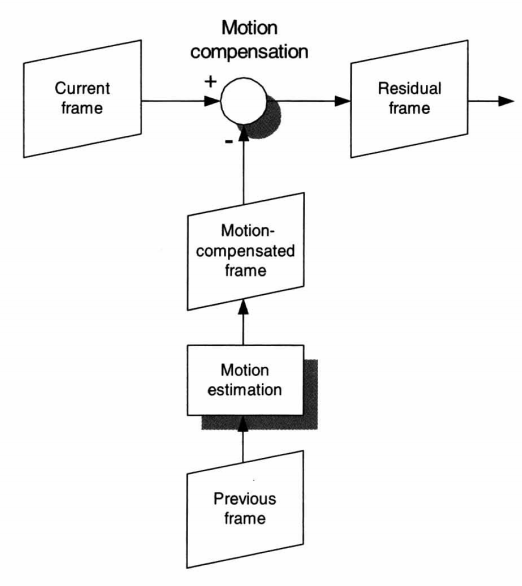
current frame based on the samples of a previously transmitted

frame.

The resulting motion-compensated predicted frame(the model

of the current frame) is substracted from the current frame to

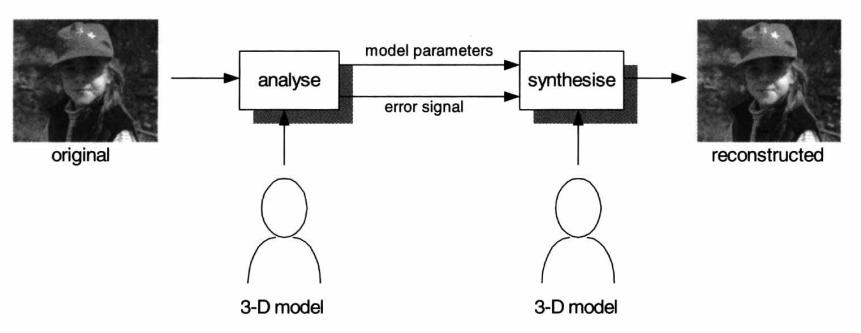
produce a residual 'error' frame.



Model-based Coding(模型基编码或知识基编码)

The encoder attempts to create a semantic model of the video

scene.



Model-based coding has the potential for far greater compression

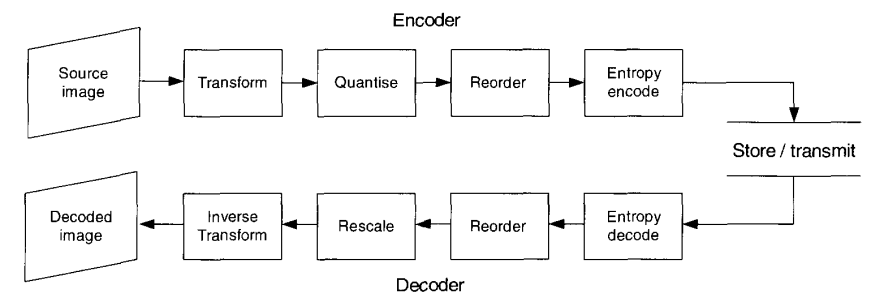
than the other source models described above.However, the

computational complexity required to analyse and synthesise 3-D

models of a video scene in real time is very high.

**3.3 Image CODEC**

An image CODEC encodes and decodes single images or individual frames from a video dequences and may consist of a *transform coding* stage followed by *quantisation* and *entropy coding*.



**3.3.1 Transform Coding**

The transform coding stage converts the image from the spatial domain into another domain in order to make it more amenable to compression.

Block Transform

The spatial image samples are prrocessed in discrete blocks, typically 8×8 or 16×16 samples.Each block is transformed using 2-D transform to produce a block of *transform coefficients*.The performance of a block-based transform for image compression depends on how well it can decorrelate the information in each block.

The Karhunen-Loeve transform(KLT) has the 'best' performance of any block-based image transform.

The KLT is very computationally inefficient.

The discrete cosine transform(**DCT**) performs nearly as well as the KLT and is much more computationally efficient.

Image Transform

A DCT is usually applied to small, discrete blocks of an image.

An image transform may be applied to a complete video image (or to large 'tile' within the image).

The most popular transform of this type is the *discrete wavelet transform*.(离散小波变换)

Another image transform that has received much attention is the so-called *fractal transform.*

**3.3.2 Quantisation**

The block and image transforms described above do not themselves achieve any compression.They represent the image in a different domain in which the image data is separated into components of varying 'importance' to the appearance of the image.

The purpose of quantisation is to remove the components of the transformed data that are unimportant to the appearance of the image and to retain the visually important components.

The quantiser scale factor or step size is often the main parameter used to control image quality and compression in an image or video CODEC.

**3.3.3 Entropy Coding**

A typical image block will contain a few significant non-zero coefficients and a large number of zero coefficients after block transform coding and quantisation. The remaining non-zero data can be efficiently compressed using a statistical compression method('entropy coding')

1. *Recording the quantised coefficients*.

2.*Run-level coding.*

3.*Entropy coding*.A statistical coding algorithm is applied to the data.

The purpose of the entropy coding algorithm is to represent frequently

occurring pairs with a short code and infrequently occurring pairs with a

longer code.

*Huffman coding* and arithmetic coding are widely used for entropy

coding of image and video data.

*Arithmetic coding* maps a series of symbols to a fractional number that

is then converted into a binary number and transmitted, has the potential

for higher compression than Huffman coding.

**3.3.4 Decoding**

The output of the entropy encoder is a sequence of binary codes representing the original image in compressed form.

Because of the data loss during quantisation, this image will not be identical to the original image: the amount of dirrerence depends partly on the 'coarseness' of quantisation.

**3.4 Video CODEC**

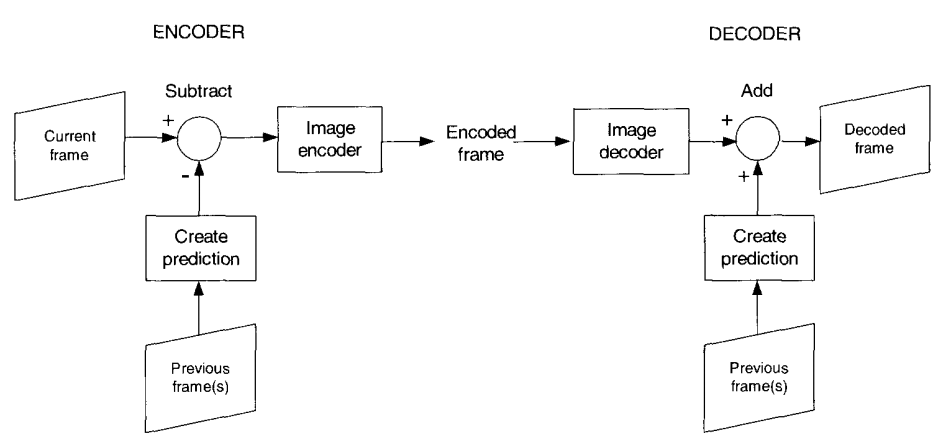
Intra-frame coding: A video signal consists of a sequence of individual frames. Each frame may be compressed individually using an image CODEC. Each frame is 'intra' coded without any reference to other frames.

Exploiting the temporal redundancy in a video sequence() may be achieved better compression performance.

1. *Prediction*: create a prediction of the current frame based on one or more previously transmitted frames.

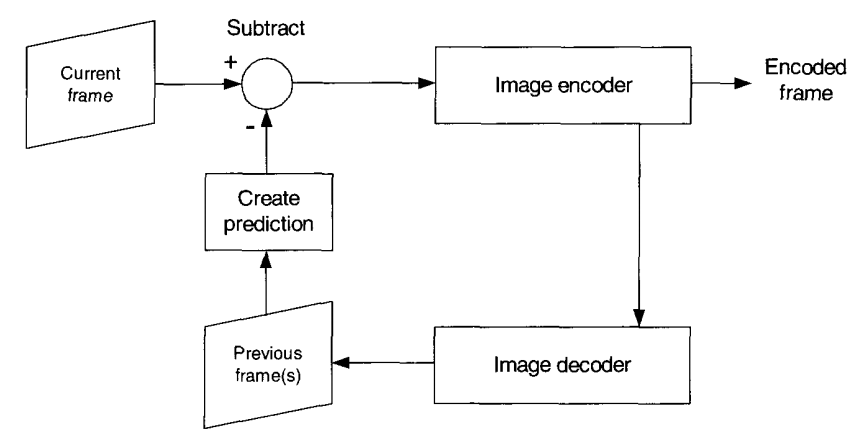
2.*Compensation*: subtract the prediction from the current frame to produce a 'residual frame'.

The residual frame is then processed using an 'image CODEC'. The key to this approach is the prediction function: if the prediction is accurate ,the residual frame will contain little data and will hence be compressed to a very small size by the image CODEC. In order to decode the frame, the decoder must 'reverse' the compensation process, adding the prediction to the decoded residual frame(*reconstruction*). This is *inter-frame coding*: frames are coded based on some relationship with other vider frames.



**3.4.1 Frame Differenceing**

Encoder use a decoded frame to form the prediction.The encoder and decoder use the same prediction and drift should be reduced or removed.



Frame differencing gives better compression performance than intra-frame coding when successive frames are very similar, but does not perform well when there is a significant change between the previous and current frames.

**3.4.2 Motion-compensated Prediction（运动补偿预测）**

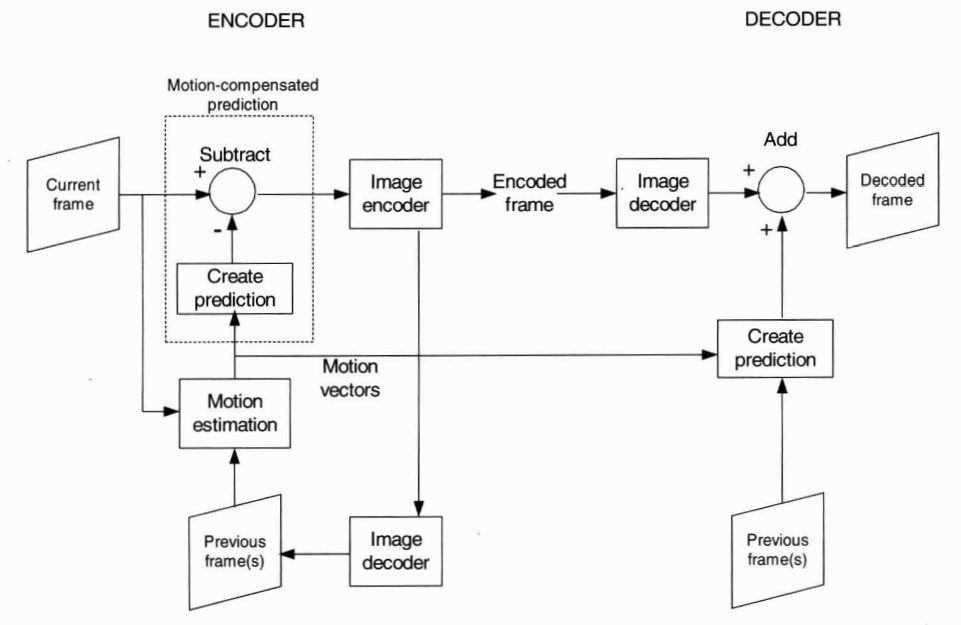
Movement in the video scene and a significantly better prediction can b e achieved by *estimating* this movement and *compensating* for it.

Two new steps are required in the encoder:

1. *Motion estimation*: a region of the current frame(ofter a rectangular block of luminance samples) is compared with neighbouring rregions of the previous reconstructed frame.

2. *Motion compensation*: the 'matching' region or block from the reference frame (identified by the motion estimator) is subtracted from the current region or block.

The decoder carries out the same motion compensation operation to reconstruct the current frame. This means that the encoder has to transmit the location of the 'best' matching blocks to the decoder(typically in the form of a set of *motion vectors*)



The design of a motion estimation algorithm can have a dramatic effect on the compression performance and computational complexity of a video CODEC.

**3.4.3 Transform,Quantisation and Entropy Encoding**

**3.4.4 Decoding**

**3.5 Summary**

**Chapter 4 Video Coding Standards: JPEG and MPEG**

**4.1 Introduction**

There are two standards bodies(ISO,ITU).

International Standards Organisation(**ISO**):

**JPEG**: image storage;

**MPEG-2**:Digital television, DVD-video system;

International Telecommunications Union(**ITU**):

**H.261**: video conferencing over ISDN;

**H.261** and **H.263**：real-time video communications over a range of

networks;

**4.2 The International Standards Bodies**

Each standard describes a *syntax* or method of representation for compressed images and video.

**4.2.1 The Expert Groups**

**4.2.2 The Standardisation Process**

**4.2.3 Understanding and Using the Standards**

**4.3 JPEG(Joint Photographic Experts Group)**

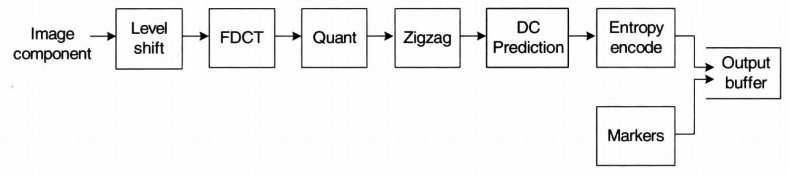
**4.3.1 JPEG**

Provide a method and syntax for compressing continuous-tone still images (such as photographs).

The JPEG standard defines a syntax and decoding process for a *baseline* CODEC and this includeds a set of features that are designed to suit a wide range of applications.

The baseline CODEC

Image data is processed one 8 × 8 block at a time.Each block is coded using the following steps.



Level shift

Forward DCT

Quantiser

Zigzag reordering

DC differential prediction

Entropy encoding

Marker insertion

Lossless JPEG

Optional modes

**4.3.2 Motion JPEG**

A 'Motion JPEG' or MJPEG CODEC codes a video sequence as a series of JPEG images, each corresponding to one frame of video( a series of intra-coded frames).

**4.3.3 JPEG-2000**

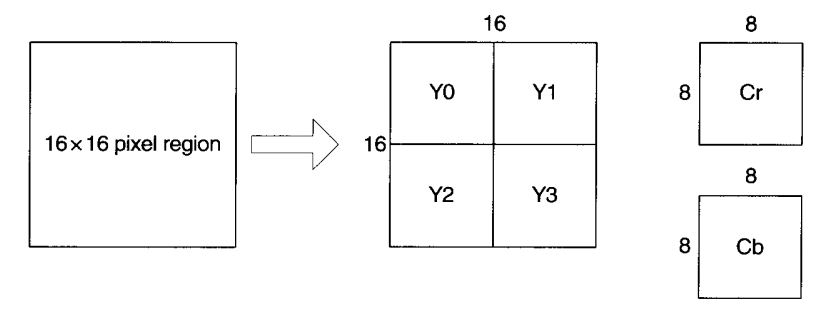
**4.4 MPEG(Moving Picture Experts Group)**

**4.4.1 MPEG-1**

Target market was the 'video CD', a standard CD containing up to 70 minutes of stored video and audio. A transfer rate of 1.4Mbps.MPEG-1 decoding is considerably simpler than encoding (JPEG have similar levels of complexity between encoder and decoder)

**MPEG-1 features**

The input video signal to an MPEG-1 video encoder is 4:2:0 Y:Cr:Cb with a typical spatial resolution of 352 × 288 or 352 × 240 pixels.Each frame of video is processed in units of a *macroblock*, corresponding to a 16 × 16 pixel area in the displayed frame. This area is made up of 16 × 16 luminance samples, 8 × 8 Cr samples and 8 × 8 Cb samples. A macroblock consists of six 8 ×8 blocks: four luminance(Y) blocks, one Cr block and one Cb block.



Each frame of video is encoded to produce a coded *picture*. There are three main types: **I-pictures, P-pictures and B-pictures**.(Also a D-picture,but seldom used.)

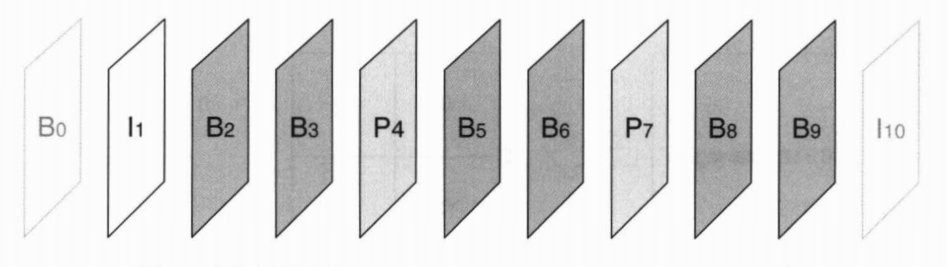
***I-pictures***: Intra-coded without any motion-compensatied prediction (in a similar way to a baseline JPEG image). An I-picture is used as a reference for further predicted pictures (P- and B - pictures).

***P-pictures***: Inter-coded using motion-compensated prediction from a *reference picture*(the P-picture or I-picture preceding the current P-picture). Hence a P-picture is predicted using *forward prediction* and a P-picture may itself be used as a reference for further predicted pictures(P- and B- pictures).

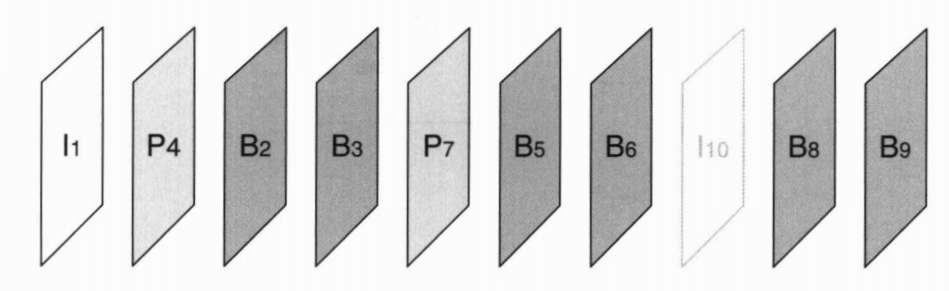
***B-pictures***: Inter-coded using motion-compensated prediction from *two* reference pictures, the P- and/or I-pictures before and after the current B-picture. Two motion vectors are generated for each macroblock in a B-picture: one pointing to a matching area in the previous reference picture(*a forward vector*) and one pointing to a matching area in the future picture(*a backward vector*). An encoder chooses the prediction mode(forward, backward or bidirectional) that gives the lowest energy in the difference macroblock.

B-pictures are not themselves used as prediction references for any further predicted frames.

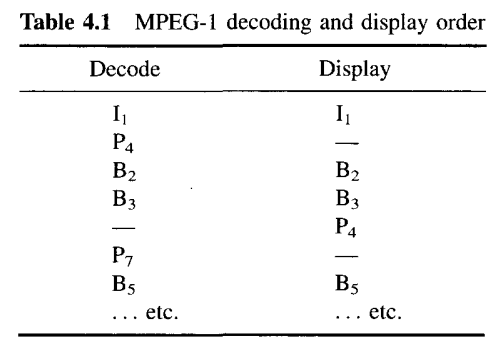
A typical series of I-,B- and P-pictures. In order to encode a B-picture, two neigthbouring I- or P-pictures( 'anchor' picture or 'key' picture) must be processed and stored in the prediction memory, introducing a delay of several frames into the encoding procedure. Before frame B2 can be encoded,its two 'anchor' frames I1 and P4 must be processed and stored, frames 1-4 must be processed before frames 2 and 3 can be coded. In this example, there is a delay of at least three frames during encoding(frames 2,3,4 must be stored before B2 can be coded) and this delay will be larger if more B pictures are used.



In order to limit the delay at the decoder, encoded pictures are *reordered* before transmission, such that all the anchor pictures required to decode a B-picture are placed before the B-picture. The frames reordered prior to transmission.



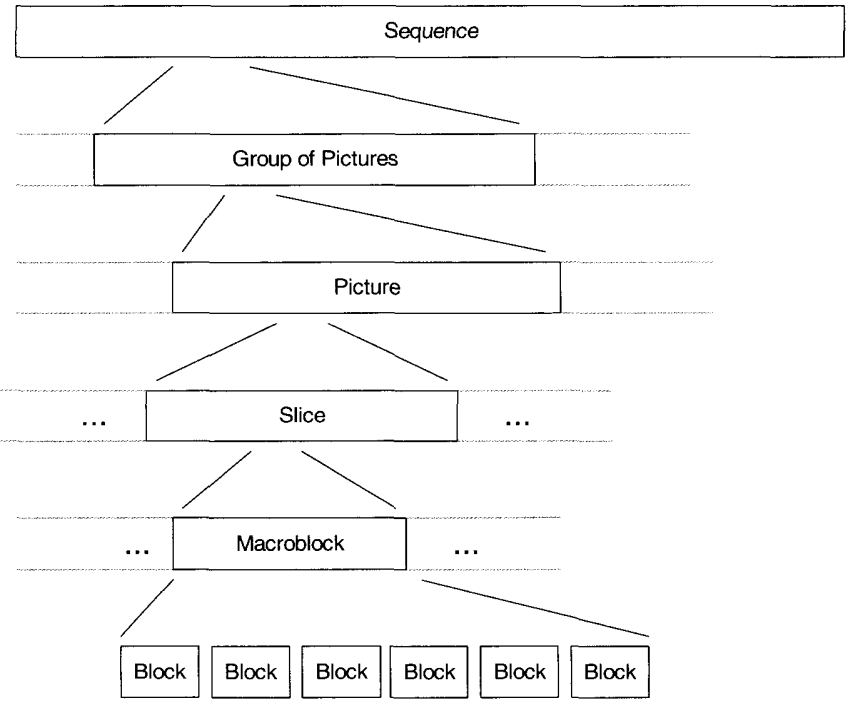
This is a totally new decode processed.P4 is decoded immediately after I1 and is stored by the decoder.Then B2 and B3 can be decoded and displayed. There is at most one frame delay between decoding and display and the decoder only needs to store two decoded frames. This is one example of 'asymmetry' between encoder and decoder: the delay and storage in the decoder are significantly lower than in the encoder.



I-picture are useful resynchronisation points in the coded bit stream: because it is coded without prediction, an I-picture may be decoded independently of any other coded pictures. An I-picture has poor compression efficiency because no temporal prediction is used. P-pictures provide better compression efficiency due to motion-compensated prediction and can be used as prediction references. B-pictures have the highest compression efficiency of each of the three picture types.

**MPEG-1 syntax**

The syntax of an EMPEG-1 coded video sequence forms a hierarchy shown below:



Layers of the hierarchy:

**Sequence layer**: This may correspond to a complete encoded video programme. This sequence starts with a *sequence header* that describes certain key information about the coded sequence including picture resolution and frame rate. The sequence consists of a series of ***groups fo pictures*** (GOPs)

**GOP layer**: A GOP is one I-picture followed by a series of P- and B- pictures,but may other GOP structures are possible,for example:

(a) All GOPs contain just on I-picture;

(b) GOPs contain only I- and P-pictures. No bidirectional prediction is used: compression efficiency is relatively poor but complexity is low;

(c) Large GOPs: the proportion of I-pictures in the coded stream is low and hence compression efficiency is high. However, there are few synchronisation points which may not be ideal for random access and for error resilience.

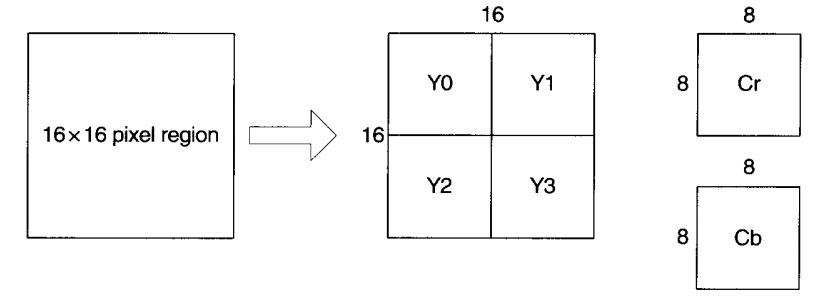
(d) Small GOPs: there is a high proportion of I- pictures and so compression efficiency is how, however there are frequent opportunities for resunchronisation.

**Picture layer**: A picture defines a single coded frame. The pircture header describes the type of coded picture(I,P,B) and a temporal reference that defines when the picture should be displayed in relation to the other pictures in the sequence.

**Slice layer**: A picture is made up of a number of slices, each of which contains an integral number of macroblocks. In MPEG-1 there is no restriction on the size or arrangement of slices in a picture, except that slices should cover the picture in raster order.

A slice starts with a slice header that defines its position. Each slice may be decoded independently of other slices within the picture and this helps the decoder to recover from transmission errors: if an error occurs within a slice, the decoder can always restart decoding from the next slice header.

**Macroblock layer**: A slice is made up of an integral number of macroblocks, each of which consists of six blocks.



The macroblock header describes the type of macroblock, motion vector and defines which 8×8 blocks actually contain coded transform data. The picture type(I,P or B) defines the 'default' prediction mode for each macroblock, but individual macroblocks within P- or B picture may be intar-coded if required. This can be useful if no good match can be found within the search area in the reference frames since it may be more efficient to code the macroblock without any prediction.

**Block layer**: A block contains variable-length code that represent the quantised transform coefficients in an 8×8 block. Each DC coefficient is coded differentially from the DC coefficient of the previous coded block, to exploit the fact that neighbouring blocks tend to have very similar DC values. AC coefficients are coded as a pair, where 'run' indicates the number of preceding zero coefficients and 'level' the value of a non-zero coefficient.

**4.4.2 MPEG-2**

The next important entertainment application for coded video was digital television. MPEG-2 consists of three main sections: Video, Audio and Systems. MPEG-2 Video is a superset of MPEG-1. The main enhancements added by the MPEG-2 standard are as follow:

*Efficient coding of television-quality video*:

The most important application of MPEG-2 is broadcast digital television. The 'core' functions of MEPG-2 are optimised for efficient coding of television resolutions at a bit rate of around 3-5 Mbps.

*Support for coding of interlaced video*:

MPEG-2 support flexible coding of interlaced video. The two fields that make up a complete interlaced frame can be encoded as separate pictures, each of which is coded as an I-, P- or B-picture.

*Scalability*

The progressive modes of JPEG described earlier are forms of scalable coding. A scalable coded bit stream consists of a number of layers, a base layer and one or more enhancement layers.The base layer can be decoded to provided a recognisable video sequence that has a limited visual quality, and a higher-quality sequence may be produced by decoding the base layer plus enhancement layer, with each extra enhancement layer improving the quality of the decoded sequence.