INTRODUCTION TO VIDEO PROCESSING

DEFINITION OF VIDEO SIGNAL

Video signal is basically any sequence of time varying images. A still image is a spatial distribution of intensities that remain constant with time, whereas a time varying image has a spatial intensity distribution that varies with time. Video signal is treated as a series of images called frames. An illusion of continuous video is obtained by changing the frames in a faster manner which is generally termed as frame rate.

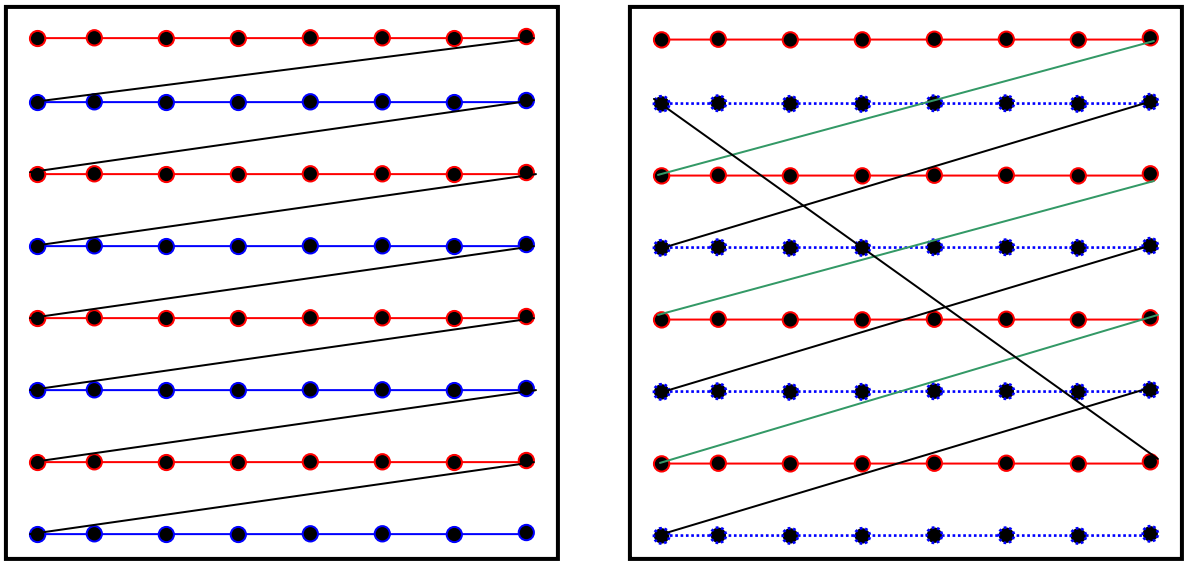
Analogue Video Signals

Despite the advance of digital video technology, the most common consumer display mechanism for video still uses analogue display devices such as CRT. Until all terrestrial and satellite broadcasts become digital, analogue video formats will remain significant. The three principal Analogue Video Signal formats are: NTSC (National Television Systems Committee), PAL (Phase Alternate Line) and SECAM (Sequential Color with Memory). All the three are television video formats in which the information in each picture is captured by CCD or CRT is scanned from left to right to create a sequential intensity signal. The formats take advantage of the persistence of human vision by using interlaced scanning pattern in which the odd and even

lines of each picture are read out in two separate scans of he odd and even fields respectively. This allows good reproduction of movement in the scene at the relatively low field rate of 50 fields/sec for PAL and SECAM and 60 fields/sec for NTSC.

Progressive and Interlaced Scan Pattern

Progressive scan patterns are used for high resolution displays like computer CRT monitors Digital cinema projections. In progressive scan, each frame of picture information is scanned completely to create the video signal. In interlaced scan pattern, the odd and even lines of each picture are read out in two separate scans of the odd and even fields respectively. This allows good reproduction of movement in the scene at relatively low field rate. The progressive and interlaced scan patterns are shown in figure 1.



(a) Progressive Scan Pattern **(b) Interlaced Scan Pattern**

Figure 1 Progressive and Interlaced Scan Pattern

Digital Video

In a digital video, the picture information is digitized both spatially and temporally and the resultant pixel intensities are quantized. The block diagram depicting the process of obtaining digital video from continuous natural scene is shown in figure 2.

Figure 2 Digital Video from natural scene

The demand for digital video is increasing in areas such as video teleconferencing, multimedia authoring systems, education, and video-on-demand systems.

Spatial Sampling

The sensitivity of Human Visual System (HVS) varies according to the spatial frequency of an image. In the digital representation of the image, the value of each pixel needs to be quantized using some finite precision. In practice, 8 bits are used per luminance sample.

Temporal sampling

A video consists of a sequence of images, displayed in rapid succession, to give an illusion of continuous motion. If the time gap between successive frames is too large, the viewer will observe jerky motion. The sensitivity of HVS drops off significantly at high frame rates. In practice, most video formats use temporal sampling rates of 24 frames per second and above.

Video formats

Digital video consists of video frames that are displayed at a prescribed frame rate. A frame rate of 30 frames/sec is used in NTSC video. The frame format specifies the size of individual frames in terms of pixels. The Common Intermediate Format (CIF) has 352 x 288 pixels, and the Quarter CIF (QCIF) format has 176 x 144 pixels. Some of the commonly used video formats are given in table 1. Each pixel is represented by three components: the luminance component Y, and the two chrominance components C_b and C_r.

Table 1 Video formats

Format	Luminance Pixel Resolution	Typical Applications
Sub-QCIF	128 X 96	Mobile Multimedia
QCIF	176 X 144	Video conferencing and Mobile Mulimedia
CIF	352 X 288	Video conferencing
4CIF	704 X 576	SDTV and DVD-Video
16CIF	1408 X 1152	HDTV and DVD-Video

Frame Type

Three types of video frames are I-frame, P-frame and B-frame. 'I' stands for Intra coded frame, 'P' stands for Predictive frame and 'B' stands for Bidirectional predictive frame. 'I' frames are encoded without any motion compensation and are used as a reference for future predicted 'P' and 'B' type frames. 'I' frames however require a relatively large number of bits for encoding. 'P' frames are encoded using motion compensated prediction from a reference frame which can be either 'I' or 'P' frame. 'P' frames are more efficient in terms of number of bits required compared to 'I' frames, but still require more bits than 'B' frames. 'B' frames require the lowest number of bits compared to both 'I' and 'P' frames but incur computational complexity.

Frames between two successive 'I' frames, including the leading 'I' frame, are collectively called as group of pictures (GOP). The GOP is illustrated in figure 3. The illustrated figure has one 'I' frame, two 'P' frames and six 'B' frames. Typically, multiple 'B' frames are inserted between two consecutive 'P' or between 'I' and 'P' frames. The existence of GOPs facilitates the implementation of features such as random access, fast forward or fast and normal reverse playback

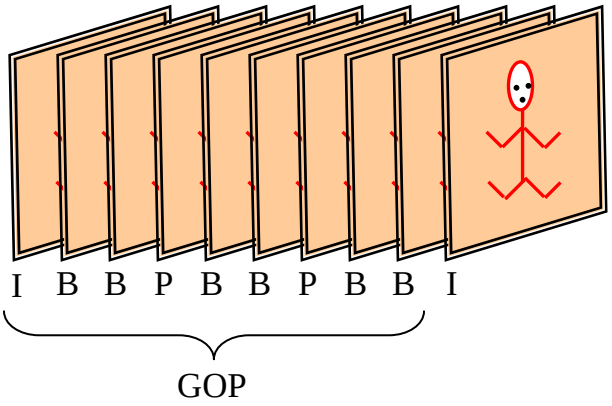


Figure 3 Group of Picture

Video Processing

Video processing technology has revolutionized the world of multimedia with products such as Digital Versatile Disk (DVD), the Digital Satellite System (DSS), high definition television (HDTV), digital still and video cameras. The different areas of video processing includes (i) Video Compression (ii) Video Indexing (iii) Video Segmentation (iv) Video tracking etc.

Video Indexing

Video indexing is necessary to facilitate efficient content-based retrieval and browsing of visual information stored in large multimedia databases. To create an efficient index, a set of representative key frames are selected which capture and encapsulate the entire video content.

Subsampling

The basic concept of subsampling is to reduce the dimension of the input video (horizontal dimension and / or vertical dimension) and thus the number of pels to be coded prior to encoding process. At the receiver the decoded images are interpolated for display. This technique may be considered as one of most elementary compression techniques which also makes use of specific physiological characteristics of the human eye and thus removes subjective redundancy contained in the video data. This concept is also used to explore subjective redundancies contained in chrominance data, i.e., human eye is more sensitive to changes in brightness than to chromaticity changes. RGB format is not preferred because R, G, B components are correlated and transmitting R,G,B components separately is redundant. To overcome this, the input image is divided into YUV components (one luminance and two chrominance components). Next, the chrominance components are subsampled relative to luminance component with a Y:U:V ratio specific to particular applications. Subsampling is denoted in the format X:X:X, where the first digits represent the number of luminance samples, used as a reference and typically “4”. The second and third digits are the number of chrominance samples, with respect to the number of Y samples. For example, 4:1:1 means that for every four Y samples, there are one U and one V samples. 4:4:4 chrominance format is shown in figure 4.

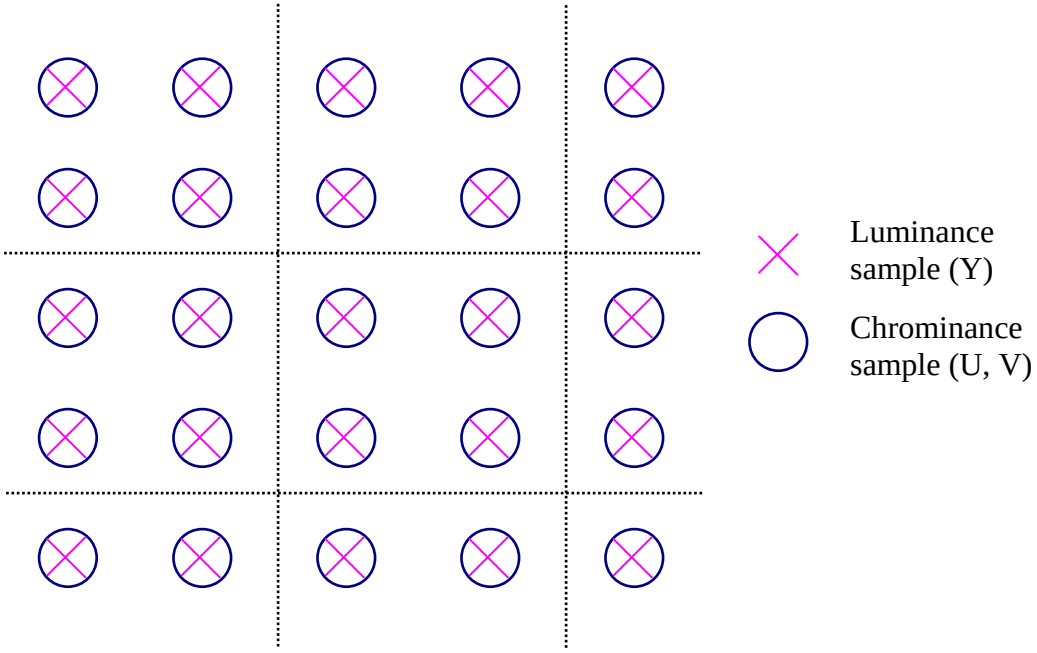


Figure 4 4:4:4 Chrominance format

The choice of the subsampling depends on application. Figure 5 illustrates the concept of 4:4:2 chrominance subsampling.

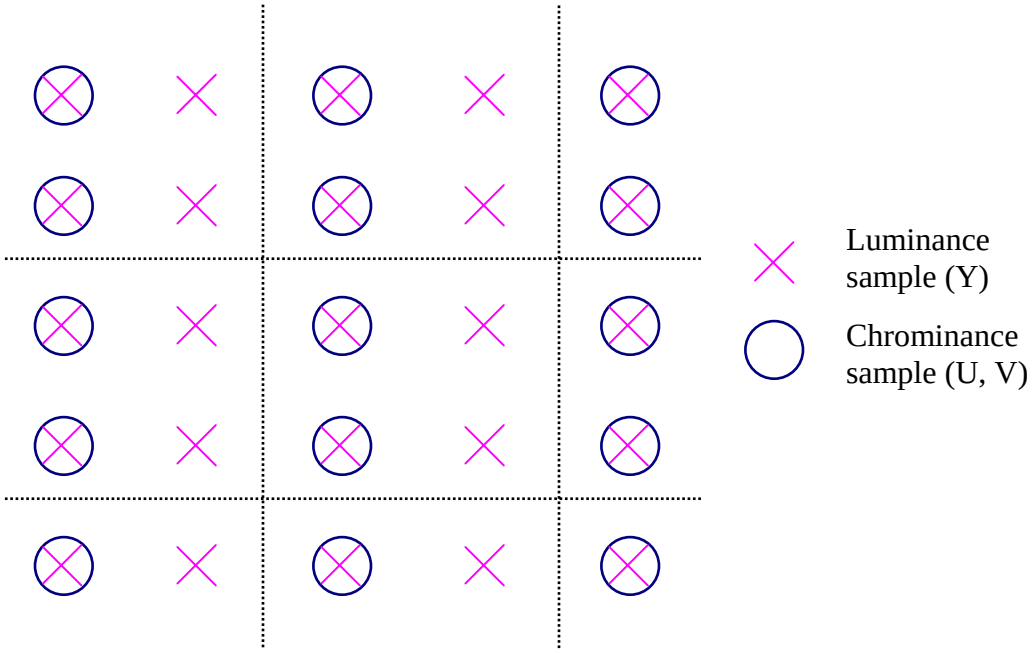


Figure 5 4:2:2 Chrominance subsampling

Video compression

Video compression plays an important role in many digital video applications such as digital libraries, video on demand, and high definition television. A video sequence with frame size of 176 X 144 pixels at 30 frames per second and 24 bits per pixel would require 18.25 Mbps, making it impractical to transmit the video sequence to transmit over standard telephone lines where data rates are typically restricted to 56,000 bits per second. This example illustrates the need for video compression. Effective video compression can be achieved by minimizing both spatial and temporal redundancy. A video sequence consists of a series of frames. In order to compress the video for efficient storage and transmission, the temporal redundancy among adjacent frames must be exploited. Temporal redundancy implies that adjacent frames are similar whereas spatial redundancy implies that neighboring pixels are similar. Video coding translates video sequences into an efficient bitstream. This translation involves the removal of redundant information from video sequence. Video sequence contains two kinds of redundancies spatial and temporal. Removal of spatial redundancy is generally termed as interframe coding and removal of temporal redundancy is termed as interframe coding. Video compression algorithms can be broadly classified into two types (i) Lossless video compression and (ii) Lossy video compression. Due to its importance in multimedia applications, most of the algorithms in video compression has centered on lossy video compression. Lossless video compression is important to applications in

which the video quality cannot tolerate any degradation such as archiving of a video, compression of medical and satellite videos etc.

Intraframe coding

Removing the spatial redundancy with a frame is generally termed as intraframe coding. The spatial redundancy within a frame is minimized by using transform. The commonly used transform is Discrete Cosine Transform.

Interframe coding

The temporal redundancy between successive frames is removed by interframe coding. Interframe coding exploits the interdependencies of video frames. Interframe coding relies on the fact that adjacent pictures in a video sequence have high temporal correlation. To minimize the temporal correlation, a frame is selected as a reference, and subsequent frames are predicted from the reference.

The general block diagram of a video encoder is shown in figure 6. The explanation of different blocks are given below

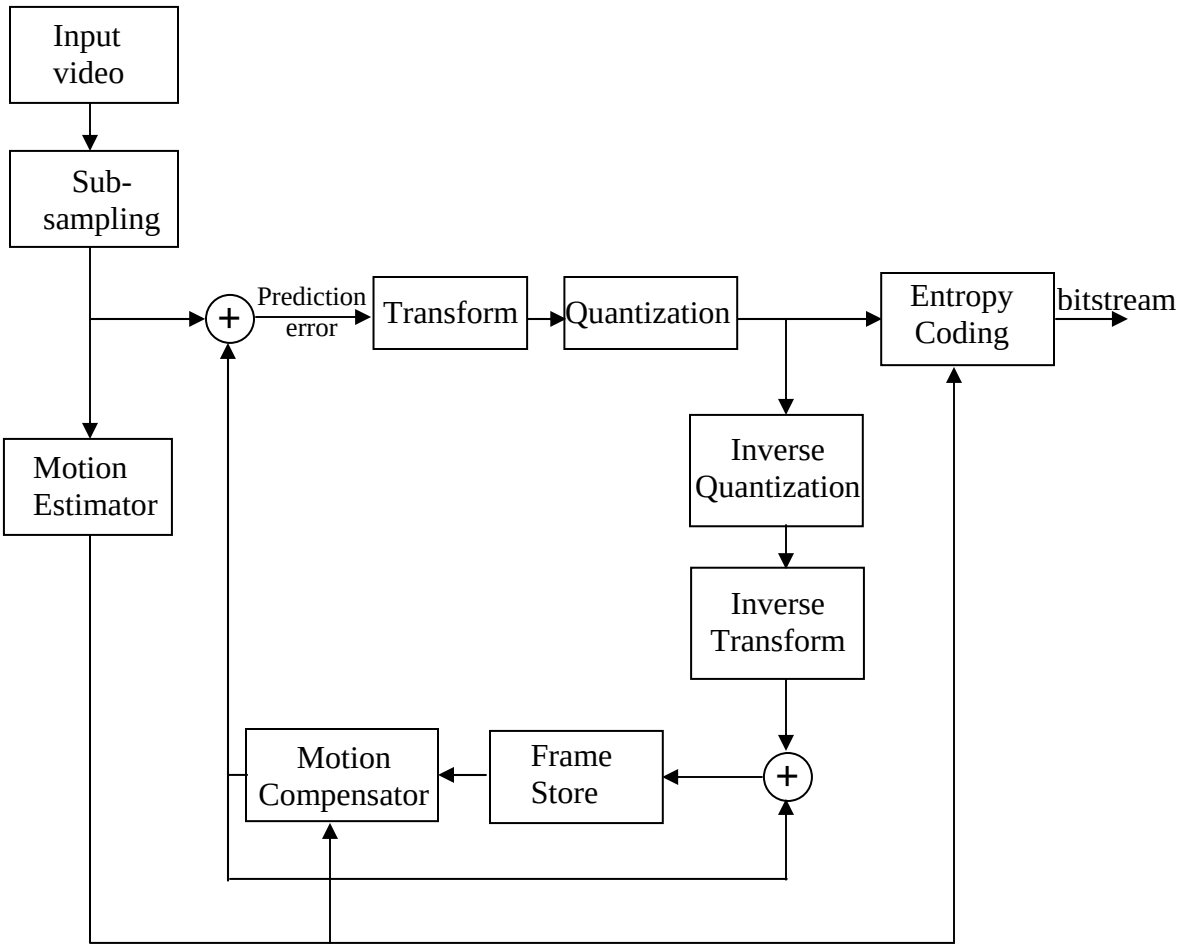


Figure 6 Block diagram of general video coding scheme

Subsampling

The basic concept of subsampling is to reduce the dimension of the input video (horizontal dimension and / or vertical dimension) and thus the number of pels to be coded prior to encoding process. At the receiver the decoded images are interpolated for display.

Motion estimation and compensation

Motion estimation describes the process of determining the motion between two or more frames in an image sequence. Motion compensation refers to the technique of predicting and reconstructing a frame using a given reference frame and a set of motion parameters. Motion compensation can be performed once an estimate of motion is available. Motion estimation/compensation is not only used in the field of video compression but also in the field of spatio-temporal segmentation, scene cut detection, frame rate conversion, de-interlacing, object tracking etc. Motion estimation and compensation have traditionally been performed using block-

based methods. They offer the advantage of being fast, easy to implement and fairly effective over a wide range of video content. Block-based motion estimation is the most practical approach to obtain motion compensated prediction frames. It divides frames into equally sized rectangular blocks and finds out the displacement of the best-matched block from previous frame as the motion vector to the block in the current frame within a search window. Based on block distortion measure or other matching criteria, the displacement of the best matched block will be described as the motion vector to the block in the current frame. The best match is evaluated by a cost function such as Mean Square Error (MSE), Mean Absolute Error (MAE), or Sum of Absolute Differences (SAD).

Transform Coding

Transform coding has been widely used to redundancy between data samples. In transform coding, a set of data samples is first linearly transformed into a set of transform coefficients. These coefficients are then quantized and entropy coded. A proper linear transform can de-correlate the input samples, and hence remove the redundancy. Another way to look at this is that a properly chosen transform can concentrate the energy of input samples into a small number of transform coefficients, so that the resulting coefficients are easier to encode than the original samples. The most commonly used transform for video coding is the discrete cosine transform. The DCT is a unitary transform, that is, the transformation preserves the energy of the signal. Unitary transforms pack a large portion of the energy of the image into relatively few components of the transform coefficients. When the transform is applied to a block of pixels that are highly correlated, as in the case in a block of an image, the transform coefficients tend to be uncorrelated. Block processing yields good results when he bits allocated to encoding the frame is enough to guarantee a good reconstruction in the decoder. However, if the bit budget is limited, as in low data rate applications, blocking artifacts may be evident in the reconstructed frame. This problem can be reduced by performing pre- and post-processing on the sequence. However, the visual quality can only be improved to a certain degree, and additional processing requires additional resources from the encoder and decoder. Another approach to solve this problem is to use non-block-based transform.

Predictive Coding

In interframe coding, the temporal redundancy of a video sequence is reduced by using motion estimation and motion compensation techniques. There are two types of frames used in interframe coding: predictive-coded (P) frames, which are coded relative to a temporally preceding 'I' or 'P' frame; and bidirectionally predictive-coded (B) frames, which are coded relative to the nearest previous / or future 'I' and 'P' frames. The forward motion-compensated prediction and bidirectional motion compensated prediction are illustrated in figure 7 and 8 respectively. In forward prediction, one motion vector per macroblock is obtained. For bidirectional prediction, two motion vectors are found. This motion vector specifies where to retrieve the macro-block from the reference frame.

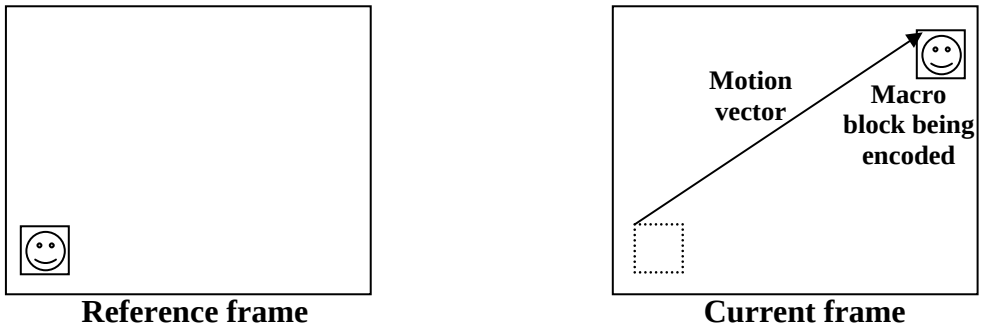


Figure 7 Forward motion-compensated prediction

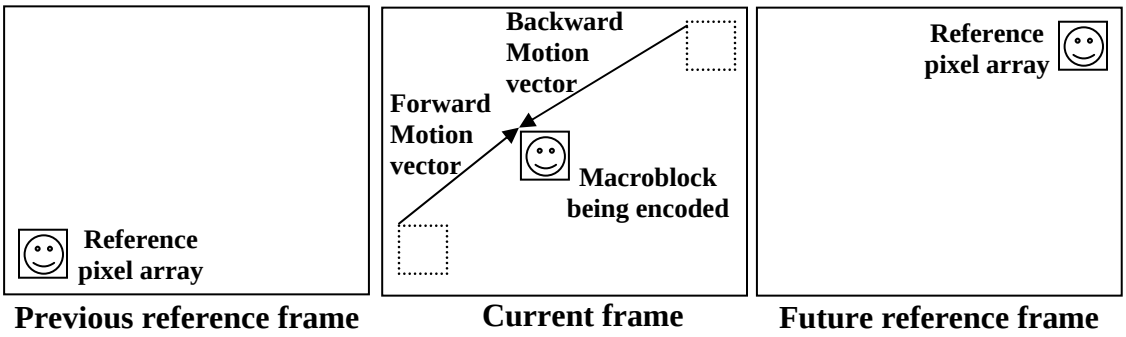


Figure 8 Bidirectional motion-compensated prediction

Motion Estimation algorithm

Compression techniques which reduce temporal redundancies are referred to as interframe techniques while those reducing spatial redundancies are referred to as intraframe techniques. Motion estimation (ME) algorithms have been applied for the reduction of temporal redundancies. ME algorithms are originally developed for applications such as computer vision, image sequence analysis and video coding. They can be categorized in the following main groups: gradient techniques, pel-recursive techniques, block matching techniques, and frequency-domain techniques. Gradient techniques have been developed in the framework of image sequence coding.

Gradient Techniques

Gradient techniques rely on the hypothesis that the image luminance is invariant along motion trajectories. The first assumption in gradient techniques is that image luminance is invariant during motions. Let $I(x, y, t)$ be the image intensity at time instant 't' at location $r = (x, y)$ and $d = (d_x, d_y)$ is displacement during time interval Δt. All techniques rely on the assumption that change in image intensity is only due to the displacement 'd', which is given by

$$I(r, t) = I(r - d, t - \Delta t) \text{ .....(1)}$$

Taylor's series expansion of right hand of equation (1) is given as

$$I(r - d, t - \Delta t) = I(r, t) - d \cdot \nabla I(r, t) - \Delta t \frac{\partial I(r, t)}{\partial t} + \text{higher order terms} \text{ .....(.2)}$$

where $\nabla = [(\partial / \partial x), (\partial / \partial y)]$ is the gradient operator and by assuming $\Delta t \to 0$, neglecting higher order terms, and defining the motion vectors as $v = (v_x, v_y) = d / \Delta t$, the change in intensity due to displacement is given by

$$v \cdot \nabla I(r, t) + \frac{\partial I(r, t)}{\partial t} = 0 \text{ .....(3)}$$

The above equation represents the spatio temporal constraint. Since the motion vector has two components, the motion field can be solved only by introducing an additional constraint. Additional constraint known as smoothing constraint is introduced to minimize the optical flow gradient magnitude. The motion field is obtained by minimizing the following error term which is given as

$$\iint \left\{ \left(v.\nabla I + \frac{\partial I}{\partial t} \right)^2 + \alpha^2 \left[\left(\frac{\partial v_x}{\partial x} \right)^2 + \left(\frac{\partial v_x}{\partial y} \right)^2 + \left(\frac{\partial v_y}{\partial x} \right)^2 + \left(\frac{\partial v_y}{\partial y} \right)^2 \right] \right\} \text{ .....(4)}$$

where α^2 is a minimization factor. This optimization problem can be solved by variational calculus. Many variations of the above algorithm are proposed in literature. From coding perspective, these motion estimation methods suffer from two main drawbacks. First, the prediction error has high energy due to smooth constraint and second the motion field requires high motion overhead.

Pel- Recursive Techniques

Pel-Recursive methods rely on recursive reduction of predictive error or displacement frame difference (DFD). The DFD or frame dissimilarity measure is denoted by

$$\mathrm{DFD}(r,t,d) = I(r,t) - I(r-d, t-\Delta t) \quad \text{....(5)}$$

These methods are among the very first algorithms designed for video coding with the goal of having low hardware complexity. The computational complexity is high in pel-recursive algorithm. Also, the error function to be minimized has generally many local minima. Pel-recursive algorithms are very sensitive to noise and have large displacements and discontinuities in the motion field which cannot be efficiently handled.

Block Matching Techniques

Block matching is widely used for stereo vision, vision tracking, and video compression. Video coding standards such as MPEG-1, MPEG-2, MPEG-4, H.261, H.263 and H.264 use block based motion estimation algorithms due to their effectiveness and simplicity for hardware implementation. The main idea behind block matching estimation is the partitioning of the target frame into square blocks of pixels and finding the best match for these blocks in a current frame. To find the best match, a search inside a previously coded frame is performed and the matching criterion is utilized on the candidate matching blocks. The displacement between the block in the predictor frame and the best match in the current frame defines a motion vector. In the encoder, it is only necessary to send the motion vector and a residue block, defined as the difference between the current block and the predictor block. The matching criterion is typically the mean of absolute errors (MAE) or the Mean of square error (MSE), given respectively in equation 6 and 7.

$$\mathrm{MAE} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \left| C_{mn} - R_{mn} \right| \quad \text{..... (6)}$$

$$\mathrm{MSE} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \left(C_{mn} - R_{mn} \right)^2 \quad \text{.....(7)}$$

where $N \times N$ is the size of each block, C_{mn} and R_{mn} are respectively the pixel values in the current block and the reference block.

SEARCH ALGORITHM FOR BLOCK MATCHING IN MOTION ESTIMATION

Motion estimation deals with the process of finding the movement of objects within a video sequence. The basic operation of a Block Matching Algorithm (BMA) is to pick up the best candidate image block in the reference image frame by calculating and comparing the matching functions between the current image block and all candidate blocks inside a confined area in the reference frame. The sizes of image block and the search area have a strong impact on the performance of the motion estimation result. A small size block offers a good approximation to the moving object, but it also produces a large amount of redundant motion information data. Small size blocks are easily interfered by random noise. On the other hand, large size blocks may contain two or more objects moving at different speed and directions. Block sizes of 8X 8 or 16 X 16 are generally considered as adequate.

(a) Full-search algorithm

The full-search algorithm is the most straight forward brute-force block matching algorithm, which provides the optimal result by matching all possible candidates within the search window. But it is computationally expensive.

(b) Three Step Search Algorithm

In the three-step search procedure, nine candidate macroblocks are selected in the first step, one centering at the center pixel and the other eight centering at eight coarsely spaced pixels around the center pixel. The matching function is then calculated. In the second step, eight more shifts are tested around the position of minimum distortion as found in the first step, but this time, the spacing of the pixels is tuned finer than before. The above procedure is repeated until the step size is smaller than one and the final motion vector is found. This process is illustrated in figure 9.

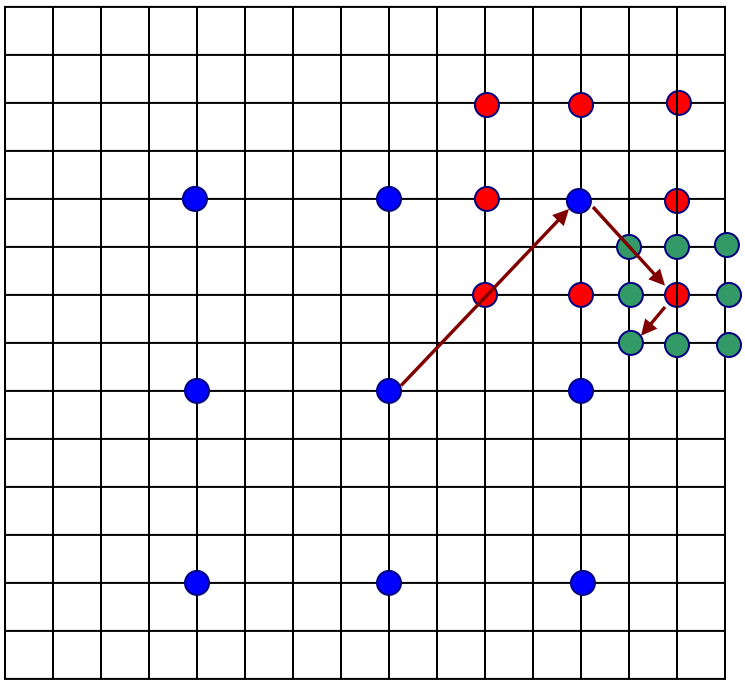


Figure 9 Three step search

(c) New three step search algorithm

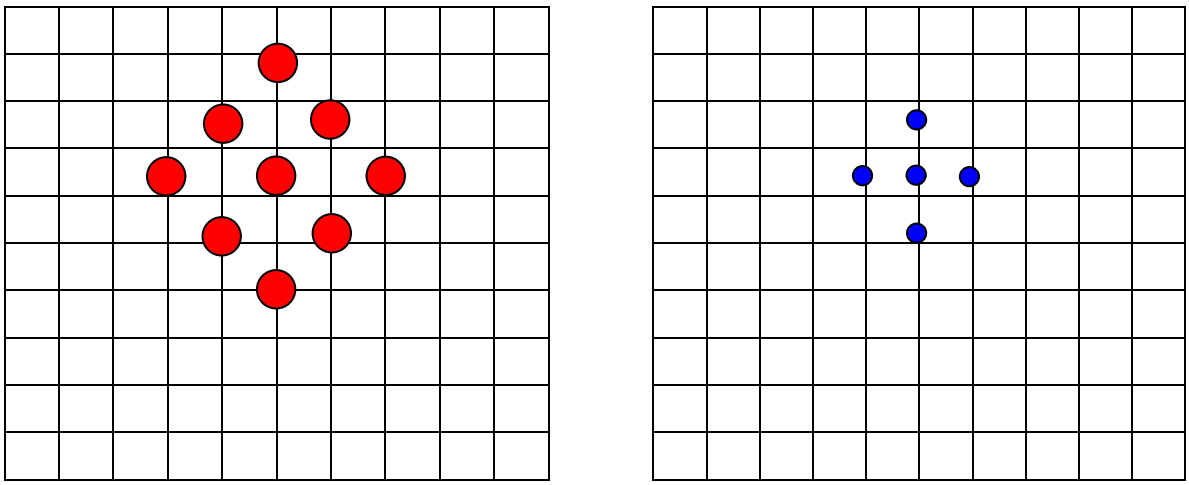
The new three step search (NTSS) algorithm utilizes additional checking points and two half stop conditions to improve the performance of three step search algorithm. In the first step, additional eight neighbors of the center are checked. If the best match is found on this small window, then additional three or five points are checked and the algorithm will stop. The main difference between three step search and new three step search is that three step search algorithm utilizes uniformly distributed search points in its first step on the other hand, new three step search algorithm employs center based checking pattern in first step and half way stop technique is applied to reduce computational cost.

(d) Four step search algorithm

Four step search algorithm employs the center biased property of the motion vectors. First the search center is located at (0,0) and the search step size is set to two. Nine points are checked in the search window. If the best match occurs at the center of the window, the neighbor search window with step size reduced to one with eight checking points on the sides will be checked and the best match is the best predicted motion vector. If the best match in the first step occurs on the edges or corners of the search window, additional three or five points will be checked in the second step. If the current minimum occurs on the center of the search window, the step size will reduce to one.

(e) Diamond Search Algorithm

The Diamond Search algorithm employs two search pattern namely large diamond search pattern (LDSP), comprises nine checking points of which eight points surround the center one to compose a diamond shape, while small diamond search pattern (SDSP) comprises of five checking points. The LDSP and SDSP search patterns are illustrated in figure 10.



(a) LDSP search pattern **(b) SDSP search pattern**

Figure 10 Search patterns in Diamond Search Algorithm

In the searching procedure of the diamond search algorithm, LDSP is repeatedly used until the step in which the minimum block distortion occurs at the center point. The search pattern is then switched from LDSP to SDSP to reach final search stage.

(f) Hierarchical block matching algorithm

The block matching algorithms can be broadly classified into two classes namely fixed block size block matching algorithm and hierarchical block matching algorithm. In the fixed block size BMA, a fixed size image block termed as motion block is compared with the candidate block in the fixed size search are on the previous frame and the best match is found using matching criteria. In the hierarchical BMA, the search is carried out in hierarchical fashion; where the sizes of the blocks and the search are vary at different levels of hierarchy. The hierarchical structure is formed by pyramid construction, multirate sampling, wavelet transformation etc. Once the hierarchical structure is constructed, the candidate motion vector having the smallest matching error is selected as the coarse motion vector. The motion vector detected at this level is propagated to the next lower level in order to guide the refinement step at that level. This process is continued to the lowest level.

Drawbacks of block-based motion estimation

The two main drawbacks of block-based motion estimation are:

1. Block based approaches are used to estimate only the translational motion, they cannot accurately model more complex types of motion such as rotation.
2. Block based approaches uses fixed blocks and is independent of scene content. A block covering two regions of different motion will not be able to accurately model the motion within both regions simultaneously.

To overcome these drawbacks, affine transform based motion estimation and compensation can be performed.

Video Compression Standards

Video coding standards define the bitstream syntax, the language that the encoder and the decoder used to communicate. Besides defining the bitstream syntax, video coding standards are also required to be efficient, in that they should support good compression algorithms as well as allow the efficient implementation of the encoder and decoder. Standardization of video compression standards has become a high priority because only a standard can reduce the high cost of video compression codecs and resolve the critical problem of interoperability of equipments from different vendors. Standardization of compression algorithms for video was first initiated by CCITT for teleconferencing and videotelephony.

H.120: H.120 is the first international digital video coding standard. H.120 was developed by ITU-T organization. ITU-T stands for the International Telecommunications Union – Telecommunications Standardization Sector. H.120 got its approval in 1984. In 1988, a second version of H.120 added motion compensation and background prediction.

H.261: H.261 was approved by ITU-T in early 1991. It was later revised in 1993 to include backward-compatible high resolution graphics transfer mode. It is a coding standard targeted to video conference and video telephone applications operating at bit rates between 64 Kbit/s and 2 M bit/s. This bit rate was chosen because of the availability of ISDN (Integrated Services Digital Network) transmission lines that could be allocated in multiples of 64 Kbit/s. The colour space used by H.261 is YC_bC_r with 4:2:0 chrominance subsampling.

H.263: H.263 standard is intended for video telecommunication. It was approved in early 1996. The key features of H.263 standard were variable block size compensation, overlapped block motion compensation. H.263 can achieve better video at 18-24 Kbps than H.261 at 64 Kbps and enable video phone over regular phone lines or wireless modem. H.263 standard supports five resolutions: QCIF, CIF, SQCIF, 4CIF, and 16 CIF.

H.263+: H.263+ standard offers a high degree of error resilience for wireless or packet-based transport networks. It was approved by ITU-T in 1998.

MPEG-1: The main objective of the MPEG-1 standard was to compress 4:1:1 CIF digital video sequences to a target bit rate of 1.5 Mbits/s. The standard defined a generic decoder but left the implementation of the encoder open to the individual design. MPEG-1 was designed for non-interlaced video sources, common in displays. Although it can be used with interlaced video streams such as television signals, its compression efficiency is smaller than other techniques due to its non-interlaced frame-based processing.

MPEG-2: MPEG-2 forms the heart of broadcast quality digital television for both standard definition and high definition television. MPEG-2 incorporates various features from H.261 and MPEG-1. MPEG-2 can be seen as a superset of MPEG-1 and it was designed to be backward compatible to MPEG-1. MPEG-2 supports various modes of scalability, including spatial, temporal, and SNR scalability.

MPEG-4: MPEG-4 became an international standard in 1998. MPEG-4 is designed to address the requirement of the interactive multimedia applications, while simultaneously supporting traditional applications. Bit rates targeted for MPEG-4 video standard range between 5-64 Kbits/s for mobile or PSTN (Public Switched Telephone Network) video applications and up to 2 Mbit/s for TV/Film applications so that this standard supersedes MPEG-1 and MPEG-2 for most applications. Video object coding is one of the most important features introduced by MPEG-4. By compressing an arbitrarily shaped video object rather than a rectangular frame, MPEG-4 enables the possibility to manipulate and interact with the objects after they are created and compressed. The compression of an arbitrarily shaped video object includes the compression of its shape, motion and texture.

The features of different video coding standards are given in table 2.

Table 2 Features of Video Coding Standards

Standards Organization	Video Coding Standard	Bit rate range	Applications
ITU-T	H.261	P x 64 Kbits/s	ISDN Video phone
ISO	MPEG-1	1.2 Mbits/s	CD-ROM
ISO	MPEG-2	4-80 Mbits/s	SDTV, HDTV
ISO	MPEG-4	24-1024 kbits/s	Wide range of applications
ITU-T	H.263		PSTN Video Phone

Scalable Video Coding

Scalable video coding schemes are intended to encode the signal once at highest resolution, but enable decoding from partial streams depending on the specific rate and resolution required by a certain application. Scalable video coding enables a simple and flexible solution for transmission over heterogeneous networks, additionally providing adaptability for bandwidth variations and error conditions. Video compression standards like ITU-T H.261 and ISO/IEC MPEG-1 did not provide any scalability mechanism.

Scalability includes temporal scalability, data rate scalability, spatial resolution scalability and temporal resolution scalability. Temporal scalability is defined as the representation of the same video in varying temporal resolutions or frame rates. Temporal resolution scalability empowers a consumer with the flexibility to choose different video frame rates for playback from a common compressed video source. A higher frame rate will smooth motion rendition, while a lower frame rate causes perception of jerkiness. Data rate scalability implies that from a single compressed bit stream, any target data rate can be achieved. The visual quality of the decoded sequence is related to the data rate. Spatial scalability is defined as the representation of the same video in varying spatial resolutions or sizes. Higher spatial resolution displays clear pictures; on the other hand, lower spatial resolution destroys fine details. For example, personal digital assistant has only 160 X 160 pixel resolution, while high-end monitor can support a display resolution of 1600 X 1200 pixels per inch. The capability of scalable video to simultaneously support broad range of display resolutions is the key in heterogeneous multiparty environment, which can span the range of high resolution device such as high-definition television (HDTV) to very low resolution gadgets such as mobile phones. A layered bit stream achieved by a multiresolution decomposition of the original image can be used to realize spatial scalability.

Review Questions

1. Consider the following two raster scan formats: progressive scan using 20 frames/second, 500 lines/frame, and interlaced scan using 40 fields/second, 250 lines/field. For each scan format, determine (i) The overall line rate (ii) the maximum temporal frequency the system can handle and (iii) the maximum vertical frequency the system can handle.

Solution

(i) The overall line rate is $20 \times 500 = 10000$ lines/sec.

(ii) The maximum temporal frequency is half of the temporal frame/field rate

(a) For progressive scan, the maximum temporal frequency is $\frac{20}{2} = 10\,Hz$

(b) For interlaced scan, the maximum temporal frequency is $\frac{40}{2} = 20\,Hz$

(iii) The maximum vertical frequency is half of the line numbers per frame

(a) For progressive scan, the maximum vertical frequency is 250 cycles/frame-height.

(b) For interlaced scan, the maximum vertical frequency is 125 cycles/frame-height.

2. Distinguish between interframe and intraframe coding?

Intraframe coding removes only the spatial redundancy within a picture, whereas interframe coding removes the temporal redundancy between pictures. In intraframe coding, the picutre is coded without referring to other pictures in the video sequence. Effective video compression can be achieved by minimizing both spatial and temporal redundancy.

3. What is the meaning of the term “scalabe coding”?

Scalabe coding allows partial decoding at a variety of resolution and quality levels from a single compressed codestream. That is, a single codestream can be applied to diverse application environments by selectively transmitting and decoding related sub-bitstreams. Scalabe coding allows for efficient signal representation and transmission in a heterogeneous environment.

4. List two valid differences between H.261 and H.263 video coding standard.

H.261	H.263
H.261 standard uses integer pel motion search	H.263 standard uses half-pel motion search
In H.261 explicit loop filtering is employed	In H.263 half-pel motion compensation accomplishes filtering

5. Explain the objective of motion estimation algorithm? Also mention the classification of motion estimation algorithms

The main objective of motion estimation algorithm is to exploit the strong interframe correlation along the temporal dimension. The key idea is to estimate the set of motion vectors that map the previous frame to the current frame. It is sufficient to transmit the motion vectors along with the error frame associated with the difference between the motion-predicted and the current frames. The error frame has much lower zero-order entropy than the current frame, hence fewer bits are need to code the information. The motion estimation algorithms can be broadly classified into (i) Gradient techniques (ii) Pel-recursive techniques, (iii) Block matching techniques, and (iv) Frequency-domain techniques.

References

(a) Books

[1] A. Murat Tekalp, "Digital Video Processing," Prentice Hall Signal Processing Series, Upper Saddle River, 1995.

[2] Iain E. G. Richardson, "Video Codec Design," John Wiley and Sons, 2002.

[3] Iain E.G. Richardson, "H.264 and MPEG-4 Video Compression," John Wiley and Sons, 2003.

[4] Al Bovic, "Handbook of Image and Video Processing," Elsevier, 2005.

[5] K.R. Rao and J. J. Hwang ,"Techniques and Standards for Image, Video and Audio Coding," Prentice Hall, Upper Saddle River, New Jersey.

[6] Yun Q. Shi and Huifang Sun, "Image and Video Compression for Multimedia Engineering," CRC Press, 2000.

[7] Barry G. Haskell, Atul Puri and Arun N. Netravali, "Digital Video, an introduction to MPEG-2," Kluwer Academic Publishers, 1996.

(b) Journal Papers

[1] D. LeGall, "MPEG: a Video Compression Standard for Multimedia Applications," Communications of the ACM, vol. 34, no. 4, pp. 46-58, April 1991.

[2] Gary J. Sullivan and Thomas Wiegand, "Rate-Distortion Optimization for Video Compression," IEEE Signal Processing Magazine, pp. 74-90, November 1998.

(c) Websites

[1] Professor Bernd Girod's course page on Image Communication II provides rich information about video compression: http://www.stanford.edu/class/ee398b/

[2] Professor Edward J. Delp of Purdue University class home page gives lot of information related to video processing http://cobweb.ecn.purdue.edu/~ace/courses/ee695-vid/

[3] Professor Yao's Video processing course page http://eeweb.poly.edu/~yao/EL6123

[4] Moving Picture Expert group home page: www.mpeg.org

[5] Advanced System Television committee home page: www.atsc.org

[6] Digital Video Broadcasting home page: www.dvb.org