P.S.R ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

SIVAKASI-626140

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE MATERIAL

NAME OF THE COURSE	**: Image and Video Analytics**
COURSE CODE	**: 191CS72**
REGULATION	**: 2019**
STAFF IN-CHARGE	**: Dr S Singaravelan**
BRANCH	**: CSE**
YEAR	**: IV**
SEMESTER	**: VII**

STAFF-IN-CHARGE **HoD/CSE**

		L	T	P	C
191CS72	**IMAGE AND VIDEO ANALYTICS**	**3**	**0**	**0**	**3**

Programme: B.E. Computer Science and Engineering **Sem: 7 Category: PC**

Prerequisites: 191CS62 – Deep Learning

Aim: The purpose of this course is to provide an understanding of the theory behind various video processing tasks. The course will extend the concepts from still images (spatial) to dynamic imagery (spatio-temporal).

Course Outcomes: The Students will be able to

CO1: Summarize the difference between analog and digital video, usage of digital videos, how digital videos are acquired, stored, different video file formats and spatio-temporal imagery.

CO2: Outline the concept of motion analysis such as motion detection, estimation and compensation.

CO3: Choose different video processing techniques such as enhancement, restoration and different filtering techniques.

CO4: List out the fundamentals of video segmentation techniques and their applications

CO5: Examine the real world to Identify as well as apply segmentation and tracking techniques to solve real-world video applications and propose solutions for the same following the ethics.

CO6: Construct different video compression techniques and unified framework.

REPRESENTATION OF DIGITAL VIDEO 9

Basics of Video – Analog Video – Digital Videos – Introduction to Digital Video Processing -Video Sampling and Interpolation.

MOTION DETECTION AND ESTIMATION 9

Notation and Preliminaries - Motion Detection- Motion Estimation - Practical Motion Estimation Algorithms – Perspectives.

VIDEO ENHANCEMENT AND RESTORATION 9

Spatiotemporal Noise Filtering - Coding Artifact Reduction - Blotch Detection and Removal - Vinegar Syndrome Removal - Kinescope Moire Removal - Scratch Removal.

VIDEO SEGMENTATION MOTIONAND TRACKING 9

Scene Change Detection- Spatiotemporal Change Detection- Motion Segmentation- Simultaneous Motion Estimation and Segmentation- Semantic Video Object Segmentation - Motion Tracking in Video- Rigid Object Tracking- Articulated Object Tracking.

VIDEO COMPRESSION AND UNIFIED FRAMEWORK 9

Introduction to Video Compression - Digital Video Signals and Formats -Video Compression Techniques - Transform Coding: Introduction to the Video Encoding Standards - MPEG-1 - MPEG-2 - H.261 - A Unified Framework for Video Indexing , Summarization , Browsing , and Retrieval.

Total Periods: 45

Text Book:

1. A. Murat Tekalp, “Digital Video Processing, Pearson Education”, Prentice Hall U.S., 2015, ISBN: 9780133991109
2. AlBovik, “Essential Guide to Video Processing”, Academic Press, 2009, ISBN 978-0-12-37445

References:

1. Yao. Wang, JomOstermann, and Ya-OinZhang, “Video Processing and Communications”, Prentice Hall, 2002, ISBN 0-13-017547-1
2. AlBovik, “Handbook of Image and Video Processing”, Academic Press, 2000, ISBN: 0121197905
3. Lain E.G. Richardson, “H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia”, Wiley, 2003, ISBN: 978--470-86960-4

Course Outcomes	**Program Outcomes (POs)**												**Program Specific Outcomes (PSOs)**			
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3				2							2	2	3		1
CO2	3	2	2									2	3	2	1	
CO3	3	2		2								2	3	2	1	1
CO4	3	2	2									2	2	2		1
CO5	3	2		2								2	3	2		
CO6	3	2	2									2	3	3		1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

191CS72 - Image and Video Analytics

Unit-I Notes:

Basics of Video:

Video refers to pictorial (visual) information, including still images and time-varying images. A still image is a spatial distribution of intensity that is constant with respect to time. A time-varying image is such that the spatial intensity pattern changes with time.

Analog Video:

Today most video recording, storage, and transmission is still in analog form. For example, images that we see on TV are recorded in the form of analog electrical signals, transmitted on the air by means of analog amplitude modulation, and stored on magnetic tape using video casette recorders as analog signals.

Analog Video Signal

The analog video signal refers to a one-dimensional (1-D) electrical signal f(t) of time that is obtained by sampling s,(x1, x2, f(t)) in the vertical ~2 and temporal coordinates. This periodic sampling process is called scanning. The signal f(t), then, captures the time-varying image intensity only along the scan lines, such as those shown in Figure 1.1. It also contains the timing information and the blanking signals needed to align the pictures correctly.

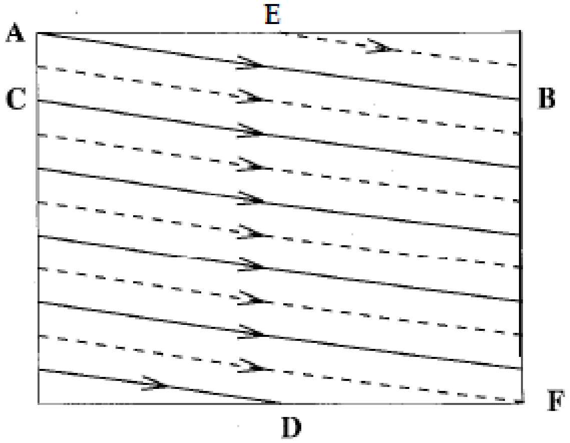


Figure 1.1: Scanning raster.

Analog Video Standards

In the previous section, we considered a monochromatic video signal. However, most video signals of interest are in color, which can be approximated by a superposition of three primary intensity distributions. The tri-stimulus theory of color states that almost any color can be reproduced by appropriately mixing the three additive primaries, red (R), green (G) and blue (B).

There exist several analog video signal standards, which have different image parameters (e.g., spatial and temporal resolution) and differ in the way they handle color. These can be grouped as:

- Component analog video
- Composite video
- S-video (Y/C video)

Composite video signal formats encode the chrominance components on top of the luminance signal for distribution as a single signal which has the same bandwidth as the luminance signal. There are different composite video formats, such as NTSC (National Television Systems Committee), PAL (Phase Alternation Line), and SECAM (Systeme Electronique Color Avec Memoire), being used in different countries around the world.

Analog Video Equipment

Analog video equipment can be classified as broadcast-quality, professional-quality, and consumer-quality. Broadcast-quality equipment has the best performance, but is the most expensive. For consumer-quality equipment, cost and ease of use are the highest priorities.

DIGITAL VIDEO

We have been experiencing a digital revolution in the last couple of decades. Digital data and voice communications have long been around. Recently, hi-fi digital audio with CD-quality sound has become readily available in almost any personal computer and workstation. Now, technology is ready for landing full-motion digital video on the desktop.

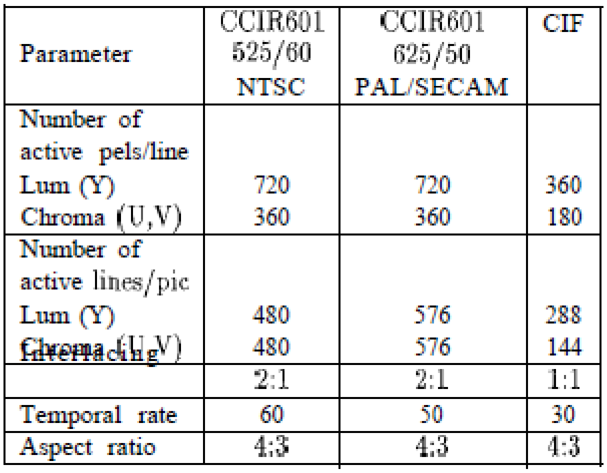
Digital Video Signal

Almost all digital video systems use component representation of the color signal. Most color video cameras provide RGB outputs which are individually digitized. Component representation avoids the artifacts that result from composite encoding, provided that the input RGB signal has not been composite-encoded before. In digital video, there is no need for blanking or sync pulses, since a computer knows exactly where a new line starts as long as it knows the number of pixels per line.

Digital Video Standards

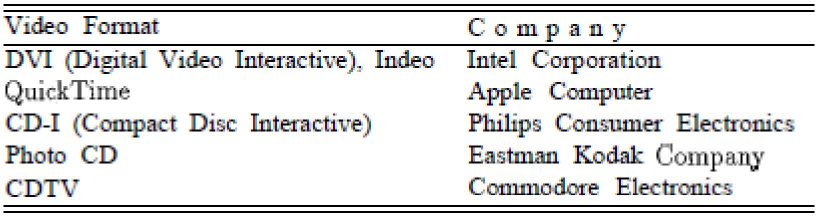
Exchange of digital video between different applications and products requires digital video format standards. Video data needs to be exchanged in compressed form, which leads to compression standards.

Table 1.1: Digital video studio standards



Parameter	CCIR601 525/60 NTSC	CCIR601 625/50 PAL/SECAM	CIF
Number of active pels/line Lum (Y) Chroma (U,V)	 720 360	 720 360	 360 180
Number of active lines/pic Lum (Y) Chroma (U,V) Interlacing	 480 480	 576 576	 288 144
	2:1	2:1	1:1
Temporal rate	60	50	30
Aspect ratio	4:3	4:3	4:3

Table 1.4: Examples of proprietary video format standards



Video Format	Company
DVI (Digital Video Interactive), Indeo	Intel Corporation
QuickTime	Apple Computer
CD-I (Compact Disc Interactive)	Philips Consumer Electronics
Photo CD	Eastman Kodak Company
CDTV	Commodore Electronics

Why Digital Video?

New developments in digital imaging technology and hardware are bringing together the TV, computer, and communications industries at an ever-increasing rate. The days when the local telephone company and the local cable TV company, as well as TV manufactures and computer manufacturers, will become fierce competitors are near [Sut 921. The emergence of better image compression algorithms, optical fiber networks, faster computers, dedicated video boards, and digital recording promise a variety of digital video and image communication products.

Digital Video Processing

Digital video processing refers to manipulation of the digital video bit stream. All known applications of digital video today require digital processing for data compression, In addition, some applications may benefit from additional processing for motion analysis, standards conversion, enhancement, and restoration in order to obtain better-quality images or extract some specific information.

Digital processing of still images has found use in military, commercial, and consumer applications since the early 1960s. Space missions, surveillance imaging, night vision, computed tomography, magnetic resonance imaging, and fax machines are just some examples. What makes digital video processing different from still image processing is that video imagery contains a significant amount of temporal correlation (redundancy) between the frames. One may attempt to process video imagery as a sequence of still images, where each frame is processed independently.

However, utilization of existing temporal redundancy by means of multi-frame processing techniques enables us to develop more effective algorithms, such as motion compensated filtering and motion-compensated prediction. In addition, some tasks, such as motion estimation or the analysis of a time-varying scene, obviously cannot be performed on the basis of a single image.

Introduction to Digital Video Processing

Digital video processing is the study of algorithms for processing moving images that are represented in digital format. Here, we distinguish *digital video* from *digital still images,* which are images that do not change with time – basically, digital photographs.

A digital video is a moving picture, or movie, that has been converted into a computer readable binary format consisting of logical 0s and 1s. Since video is dynamic, the visual content evolves with time and generally contains moving and/or changing objects.

Digital video is ordinarily a function of three dimensions – two in space and one in time, as depicted in
Fig. 1.1. Because of this, digital video processing is data intensive: significant bandwidth, computational, and storage resources are required to handle video streams in digital format.

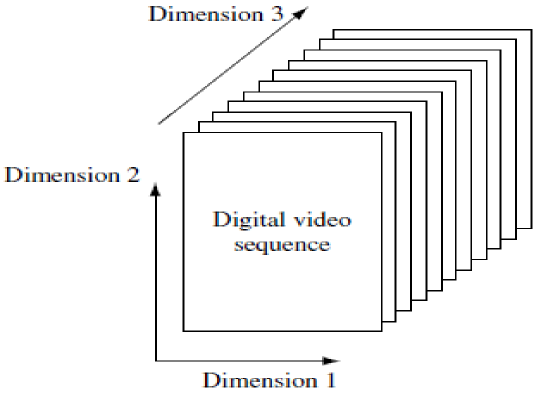


FIGURE 1.1

The dimensionality of video.

The Essential Guide to Image Processing, a tutorial is given on the processes of digitization of images, or *analog-to-digital conversion* (A/D conversion). This conversion process consists of two distinct sub processes: *sampling*, or conversion of a continuous-space/time video signal into a discrete-space/time video signal, and *quantization*, which is the process of converting a *continuous-valued video* that has a continuous range (set of values that it can take) of intensities and/or colors into a *discrete-valued video* that has a discrete range of intensities and/or colors.

Quantization is a necessary precursor to digital processing or display, since the image intensities and/or colors must be represented with a finite precision (limited by word length) in a digital video processor or
display device.

Video Sampling and Interpolation

We consider only scalar-valued images, such as the luminance or one of the RGB components of a color image. All the theory and results can be applied in a straightforward way to multicomponent (or vector-valued) signals, such as color video by treating the sampling and reconstruction of each scalar component independently.

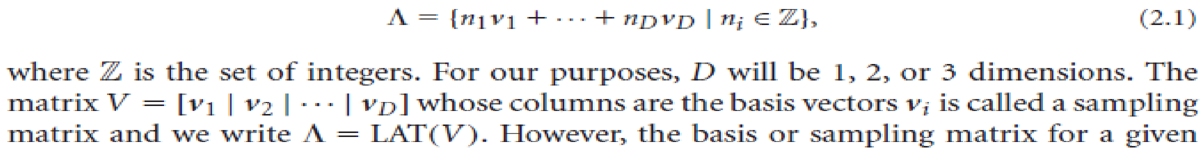
Three main sections. First, the sampling lattice, the basic tool in the analysis of spatiotemporal sampling, is introduced. The issues involved in the sampling and reconstruction of continuous time-varying imagery are then addressed.

SPATIOTEMPORAL SAMPLING STRUCTURES

A continuous time-varying image *fc (x, y, t)* is a scalar real-valued function of two spatial dimensions *x* and *y* and time *t* , usually observed in a rectangular spatial window *W* over some time interval *T* .

The spatiotemporal region *W _T* is denoted as *WT*. The spatial window is of dimension *pw _ph*, where *pw* is the picture width and *ph* is the picture height. Since the absolute physical size of an image depends on the display device used, and the sampling density for a particular video signal may be variable, we choose to adopt the picture height *ph* as the basic unit of spatial distance, as is common in the broadcast video industry.

The mathematical structure most useful in describing sampling of time-varying images is the *lattice*. A discussion of lattices from the point of view of video sampling can be found in [1] and [2]. Some of the main properties are summarized here.



$$\Lambda = \{n_1 \mathbf{v}_1 + \cdots + n_D \mathbf{v}_D \mid n_i \in \mathbb{Z}\}, \qquad (2.1)$$

where $\mathbb{Z}$ is the set of integers. For our purposes, D will be 1, 2, or 3 dimensions. The matrix $V = [\mathbf{v}_1 \mid \mathbf{v}_2 \mid \cdots \mid \mathbf{v}_D]$ whose columns are the basis vectors $\mathbf{v}_i$ is called a sampling matrix and we write $\Lambda = \mathrm{LAT}(V)$. However, the basis or sampling matrix for a given

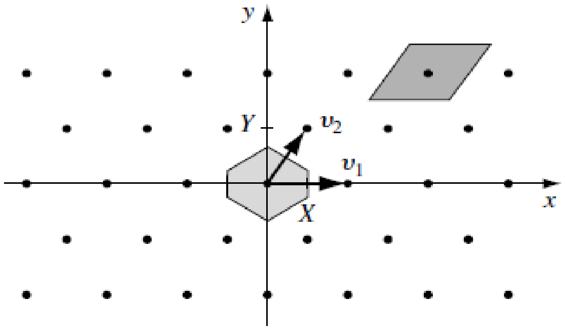


FIGURE 2.1

Example of a lattice in two dimensions with two possible unit cells.

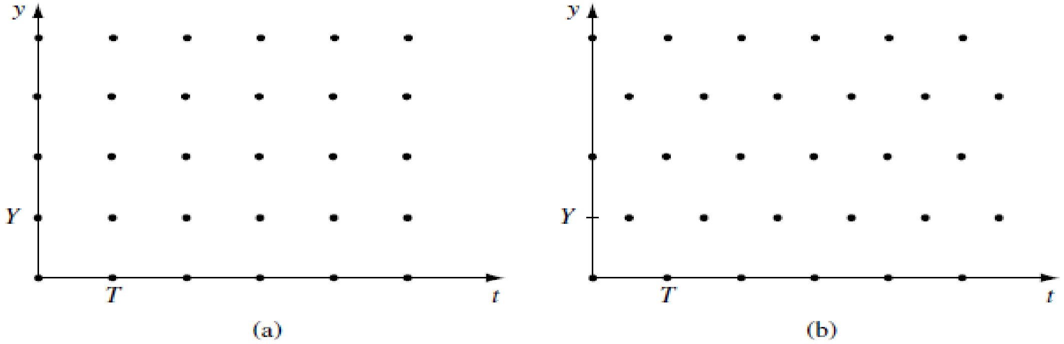


FIGURE 2.2

Two-dimensional vertical-temporal lattices. (a) Rectangular lattice Λ_R. (b) Hexagonal lattice Λ_H.

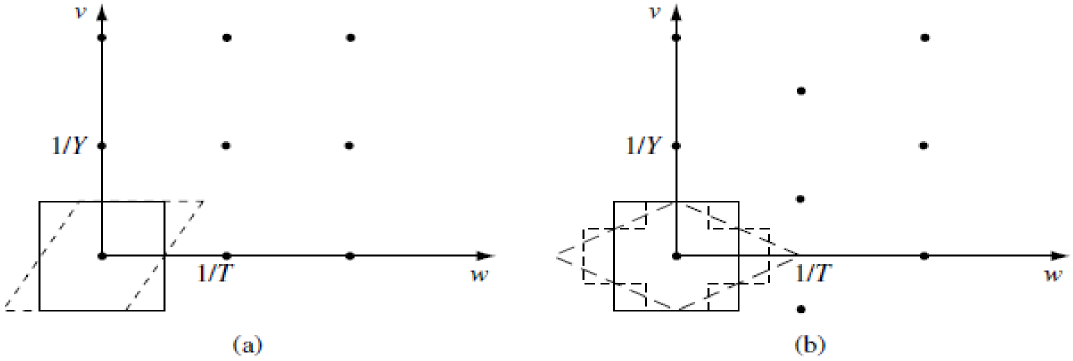


FIGURE 2.3

Reciprocal lattices of the two-dimensional vertical-temporal lattices of Fig. 2.2 with several possible unit cells. (a) Rectangular lattice Λ_R^*. (b) Hexagonal lattice Λ_H^*.

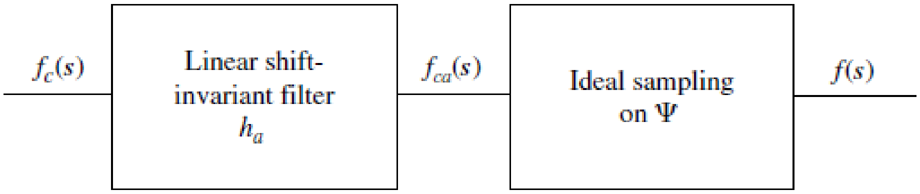


FIGURE 2.4

System for sampling a time-varying image.

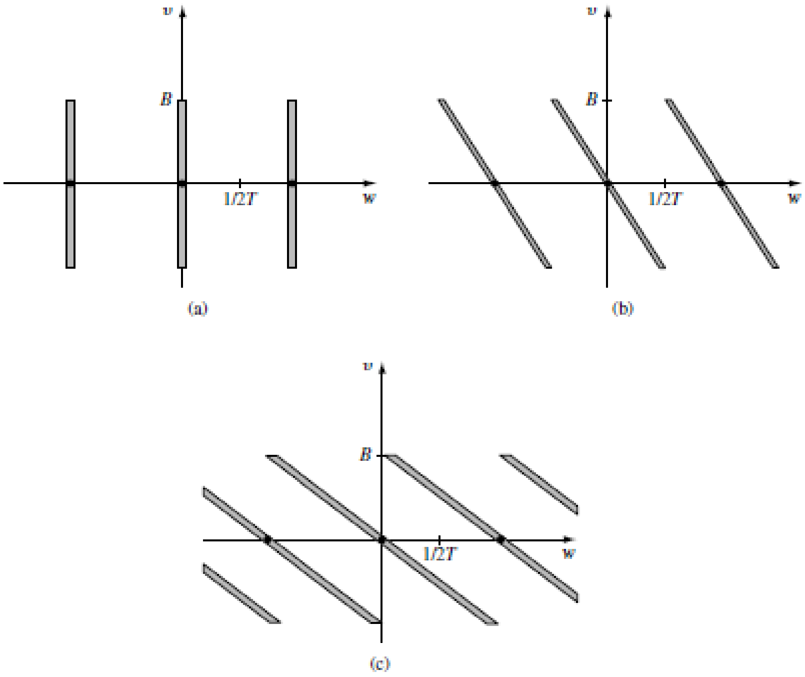


FIGURE 2.5

Vertical-temporal projection of the spectrum of temporally sampled time-varying image with vertical motion of velocity ν. (a) $\nu = 0$. (b) $\nu = 1/2TB$. (c) $\nu = 1/TB$.

SAMPLING STRUCTURE CONVERSION

There are numerous spatiotemporal sampling structures used for the digital representation of time-varying imagery. However, the vast majority of those in use fall into one of two categories corresponding to progressive or interlaced scanning with aligned horizontal sampling.

A three-dimensional view of these two sampling lattices is shown in Fig. 2.7. It can be observed how the odd numbered horizontal lines in each frame from the progressive lattice (y/Y odd) in Fig. 2.7(a) have been delayed temporally by $T/2$ for the interlaced lattice of Fig. 2.7(b).

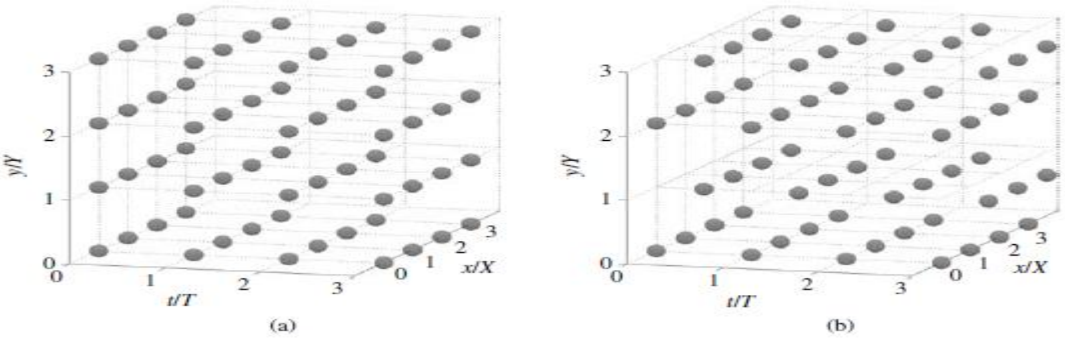
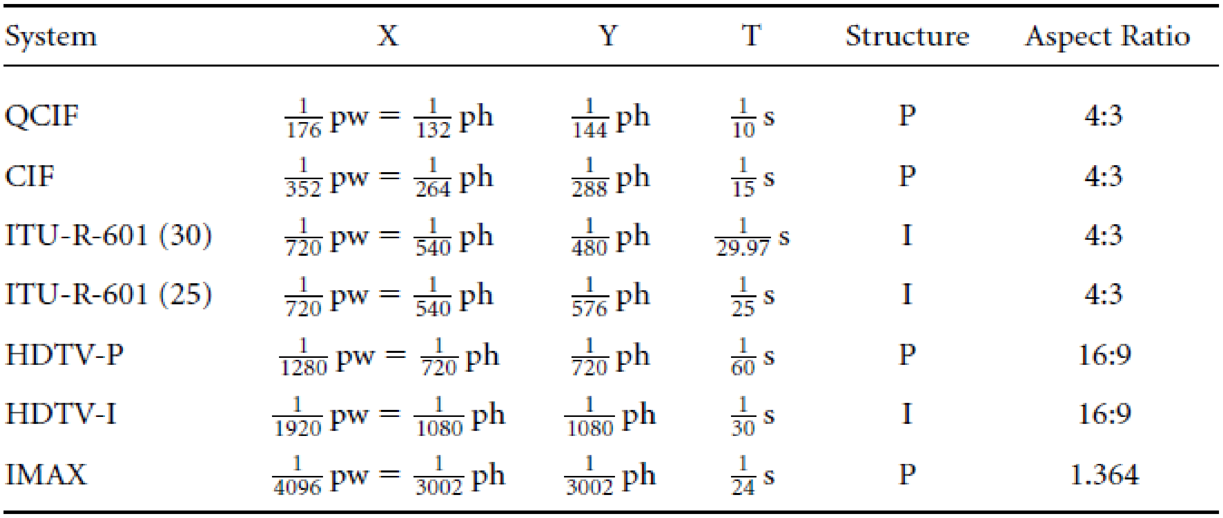


FIGURE 2.7
Three-dimensional view of spatiotemporal sampling lattices. (a) Progressive. (b) Interlaced.

TABLE 2.1 Parameters of several common spatiotemporal sampling structures.



System	X	Y	T	Structure	Aspect Ratio
QCIF	$\frac{1}{176}$ pw = $\frac{1}{132}$ ph	$\frac{1}{144}$ ph	$\frac{1}{10}$ s	P	4:3
CIF	$\frac{1}{352}$ pw = $\frac{1}{264}$ ph	$\frac{1}{288}$ ph	$\frac{1}{15}$ s	P	4:3
ITU-R-601 (30)	$\frac{1}{720}$ pw = $\frac{1}{540}$ ph	$\frac{1}{480}$ ph	$\frac{1}{29.97}$ s	I	4:3
ITU-R-601 (25)	$\frac{1}{720}$ pw = $\frac{1}{540}$ ph	$\frac{1}{576}$ ph	$\frac{1}{25}$ s	I	4:3
HDTV-P	$\frac{1}{1280}$ pw = $\frac{1}{720}$ ph	$\frac{1}{720}$ ph	$\frac{1}{60}$ s	P	16:9
HDTV-I	$\frac{1}{1920}$ pw = $\frac{1}{1080}$ ph	$\frac{1}{1080}$ ph	$\frac{1}{30}$ s	I	16:9
IMAX	$\frac{1}{4096}$ pw = $\frac{1}{3002}$ ph	$\frac{1}{3002}$ ph	$\frac{1}{24}$ s	P	1.364

P indicates progressive scanning and I indicates interlaced scanning.

This situation is illustrated in Fig. 2.8. The continuous signal *fc (**x**)* is acquired on the lattice _1 using a physical camera modeled as in Fig. 2.4 with impulse response *ha(**x**)* to yield *f (**x**)*. It is desired to estimate the signal *fo(**x**)* that would have been obtained if *fc (**x**)* was sampled on the lattice _2 with an ideal or theoretical camera having impulse response *hoa(**x**)*. Note that since this camera is *theoretical*, the impulse response *hoa(**x**)* does not have to be realizable with any particular technology. It can be optimized to give the best displayed image on _2 [3]. A system *H*, which can be linear or nonlinear, is then required to estimate *fo(**x**)* from *f (**x**)*.

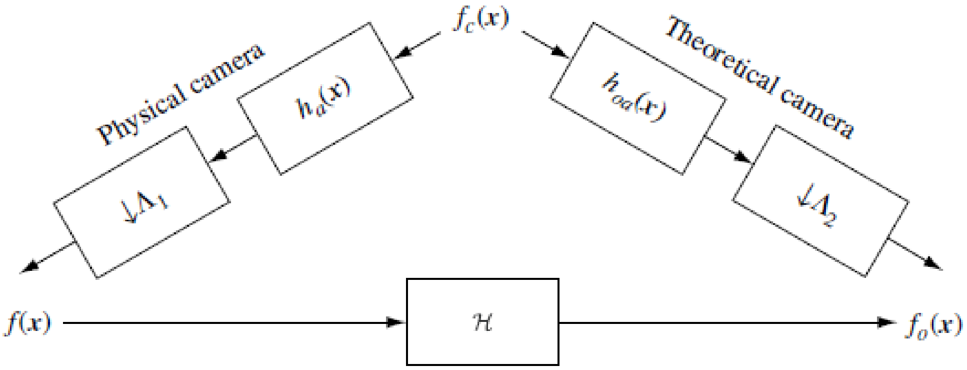


FIGURE 2.8

Acquisition models for the observed signal $f(\mathbf{x})$ on Λ_1 and the desired output signal $f_o(\mathbf{x})$ on Λ_2.

Frame-rate Conversion

Consider first the case of pure frame-rate conversion. This applies when both the input and the output sampling structures are separable in space and time with the same spatial sampling structure, and where spatial aliasing is assumed to be negligible. The temporal sampling period is to be changed from *T*1 to *T*2.

Pure Temporal Interpolation

The most straightforward approach is pure temporal interpolation, where a temporal resampling is performed independently at each spatial location ***x***. A typical application for this is increasing the frame rate in motion picture film from 24 frames/s to 48 or 60 frames/s, giving significantly better motion rendition.

Motion-compensated Interpolation

It is clear that to correctly deal with a situation, such as in Fig. 2.4(c), it is necessary to adapt the interpolation to the local orientation of the spectrum, and thus to the velocity, as suggested in Fig. 2.9(b). This is called motion-compensated interpolation.

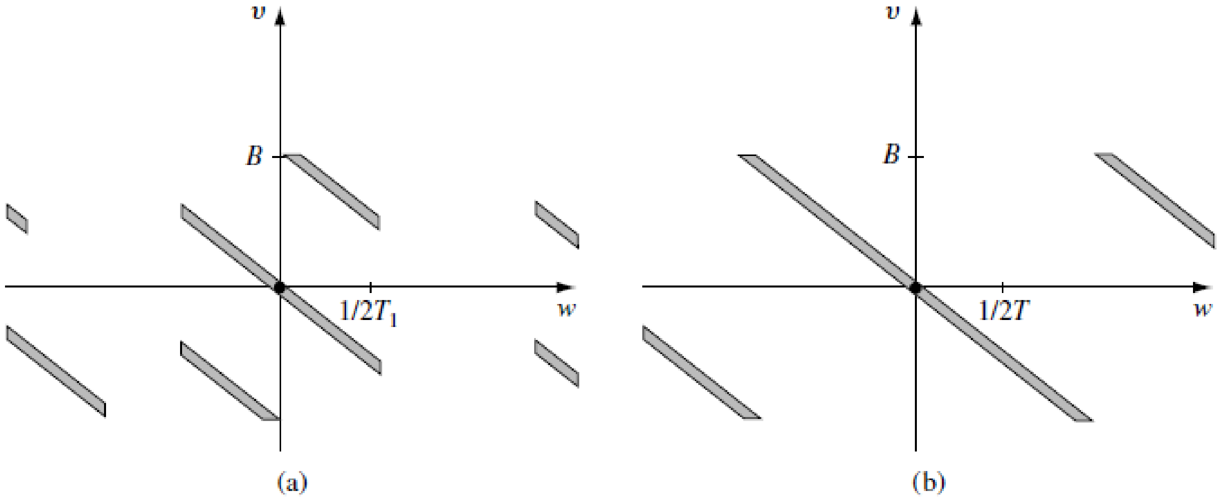


FIGURE 2.9

Frequency domain interpretation of 2:1 temporal interpolation of an image with vertical velocity $1/TB$. (a) Pure temporal interpolation. (b) Motion-compensated interpolation.

Spatiotemporal Sampling Structure Conversion

We consider the case where both the spatial and the temporal sampling structures are changed, and when one or both of the input and output sampling structures is not separable in space and time (usually because of interlace). If the input sampling structure _1 is separable in space and time (as in Eq. (2.18)) and spatial aliasing is minimal, then the methods of the previous section can be combined with pure spatial interpolation.

Deinterlacing

Deinterlacing generally refers to a 2:1 interpolation from an interlaced grid to a progressive grid with sampling lattices

Video acquisition:

Video capture is the process of converting an analog video signal—such as that produced by a video camera, DVD player, or television tuner—to digital video and sending it to local storage or to external circuitry. The resulting digital data are referred to as a digital video stream, or more often, simply video stream. Depending on the application, a video stream may be recorded as computer files, or sent to a video display, or both.

The video sensor in a camcorder (or digital camera) is what makes a digital camcorder "digital." Put simply, an image sensor transforms the light captured by your camcorder's lens and turns it into a digital signal.

That digitized light is processed and stored in your camcorder's memory as a digital video file you can later view on your computer or TV.

There are two main types of camcorder image sensors: charged coupled device (CCD) and complementary metal oxide semiconductor (CMOS).

Both types of image sensor technology contain hundreds of thousands or even millions of pixels.

In a CCD image sensor, pixels capture light and move it toward the edge of the chip, where it's converted into a digital signal. In a CMOS sensor, the light is converted at the pixel itself. No electrical conveyor belt required.

CCD Sensor:

A CCD camera is a video camera that contains a charged-coupled device (CCD), which is a transistorized light sensor on an integrated circuit. In plain English, CCD devices convert or manipulate an electrical signal into some kind of output, including digital values. In cameras, CCD enables them to take in visual information and convert it into an image or video. They are, in other words, digital cameras.

In terms of the working principle of CCD cameras, these video cameras capture an image and transfer it to the camera’s memory system to record it as electronic data. CCD cameras’ main accomplishment is the production of quality images without any distortion. Basically, the camera turns light into electricity. A CCD camera forms light sensitive elements called pixels which sit next to each other and form a particular image. CCD cameras have been in production for a long period of time and tend to have high quality pixels that produce a higher quality, low-noise image than any other camera.

CMOS Sensor:

A CMOS sensor is an electronic chip that converts photons to electrons for digital processing. CMOS (complementary metal oxide semiconductor) sensors are used to create images in digital cameras, digital video cameras and digital CCTV cameras.

The CMOS cells are surrounded by transistors, which amplify the charge of the electrons gathered by the cells, sending them across the chip by tiny wires in the chip’s circuitry. A digital-to-analog converter at one corner of the device reads the electrons and translates the differing charges of individual cells into pixels of various colors.

CMOS’ low manufacturing cost makes it possible to create low-cost consumer devices. Advances in CMOS technology have made it possible for them to approach their competitor in high-end digital cameras, charge-coupled devices (CCD). In contrast to CMOS, CCD cells are not surrounded by transistors and must actively use power to gather light. This makes them less power-efficient but also enables the benefits of a lower-noise image and greater light sensitivity.

Acronym	Charged Couple Devices	Complimentary Metal Oxide Semiconductor
Cost	More expensive	Cheaper
Shutter type	Global	Rolling
Skew	No	Yes
Wobble	No	Yes
Partial exposure	No	Yes
Vertical smear	Yes	No
Noise	Less	More
Power efficiency	Less efficient	More efficient

191CS72 - Image and Video Analytics

Unit-II Notes:

Motion Detection and Estimation

- **Introduction**
  - The success of video as a medium is primarily due to the capture of motion; a single image provides snapshot of a scene, whereas a sequence of images also records scene's dynamics.
  - Motion is equally important for video processing and compression for two reasons.
  - First, motion carries information about spatiotemporal relationships between objects in the field of view of a camera. This information can be used in applications such as traffic monitoring or security surveillance (Chapter 19), for example, to identify objects that move or those entering/leaving the scene.
  - . Second, image properties, such as intensity or color, have a very high correlation in the direction of motion, that is, they do not change significantly when tracked over time (the color of a car does not change as it moves across the camera field of view).
  - Motion can also be used in temporal filtering of video (Chapters 2 and 4); one-dimensional filter applied along a motion trajectory can reduce noise without spatially blurring a frame.

NOTATION AND PRELIMINARIES

- Let I : T → R+ be the intensity of image sequence defined over spatial domain and temporal domain T .
- Let x = (x1,x2)T ∈ and t ∈ T denote spatial and temporal positions of a point in this sequence, respectively.

Binary Hypothesis Testing – positive / Negative

- Hypothesis testing is **a formal procedure for investigating our ideas about the world using statistics**. It is used by scientists to test specific predictions, called hypotheses, by calculating how likely it is that a pattern or relationship between variables could have arisen by chance.
- Let y be an observation and let Y be the associated random variable. Suppose that there are two hypotheses H0 and H1 with corresponding probability distributions P(Y = y|H0) and P(Y =y|H1), respectively.
- The goal is to decide from which of the two distributions a given y is more likely to have been drawn
- Clearly, four possibilities exist (true hypothesis/decision): H0/H0, H0/H1, H1/H0, and H1/H1. Although H0/H0 and H1/H1 correspond to correct choices, H0/H1 and H1/H0 are erroneous

- Under the Bayes criterion, two a priori probabilities 0 and 1 1 0 are assigned to the two hypotheses H0 and H1, respectively, and a cost is assigned to each of the four scenarios listed above.

- The quantity on the left is called the likelihood ratio and is a constant dependent on the costs of the four scenarios. Since these costs are determined in advance, is a fixed constant. If 0 and 1 are predetermined as well, the above hypothesis test compares the likelihood ratio with a fixed threshold.

Markov Random Fields – Distribution Pixel Based

- A Markov Random Field (MRF) is a graphical model of a joint probability distribution. It consists of an undirected graph in which the nodes represent random variables. Let be the set of random variables associated with the set of nodes S.
- The equivalence between Markov random fields and Gibbs distributions is provided through the important Hammersley-Clifford theorem, which states that is a MRF on with respect to N if and only if its probability distribution is a Gibbs distribution with respect to and N.

MAP Estimation - an estimate of an unknown quantity, that equals the mode of the posterior distribution

- Let Y be a random field of observations, and let be a random field modeling the quantity we want to estimate based on Y .

- Let y, be their respective realizations. For example, y could be a difference between two images, and could be a field of motion detection labels. To compute based on y, a powerful tool is the (MAP) estimation.

MOTION DETECTION

- Motion detection is, arguably, the simplest of the three motion-related tasks, that is, detection, estimation and segmentation.

  Threshold

- Global Threshold

- Local / Fixed Threshold

  - The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than a fixed value called the threshold. , or a white pixel if the pixel intensity is greater than that threshold.

- Adaptive Threshold

  - Adaptive thresholding is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions.

Hypothesis Testing with Fixed Threshold

- Fixed-threshold hypothesis testing belongs to the simplest motion detection algorithms as it requires few arithmetic operations. Several early motion detection methods belong to this class, although originally they were not developed .
- The above pixel-based hypothesis test is not robust to noise in the image.

Experimental Comparison of Motion Detection Methods

- Figure 3.1 shows motion detection results on a typical urban surveillance video for the variational formulation (Fig. 3.1(a), bottom), frame-difference hypothesis test (Fig. 3.1(b)), stationary-only hypothesis test (Fig. 3.1(c)), and stationary/moving hypothesis test (Fig. 3.1(d)). The latter three results are shown without and with Markov model (bottom).
- In each case, the addition of Markov prior clearly improves the detection accuracy by reducing both false positives and misses. The object tunnels shown in Fig. 3.1(e) confirm the accuracy of detections and also illustrate the dynamic evolution of individual objects' masks.

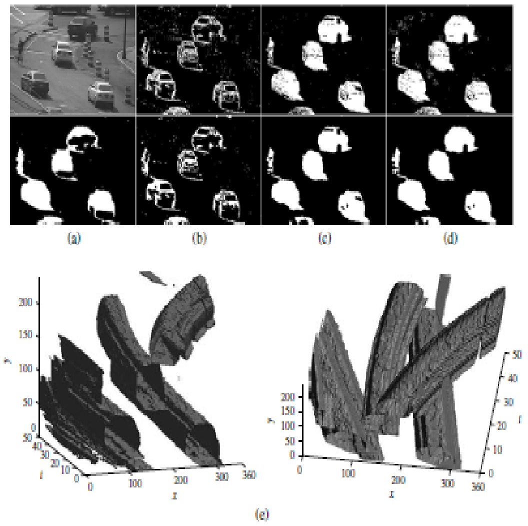


FIGURE 3.1

Motion detection results for a 360×240-pixel road traffic video: (a) original frame and active-surface detection result (bottom); (b) frame-difference result (3.7); (c) stationary-only hypothesis test result (3.9); and (d) stationary/moving hypothesis test result (3.11). Bottom results in (b–d) include MRF label model. (e) Two views of object tunnels, that is, surfaces "wrapped" around each moving object's mask, for the case of stationary/moving hypothesis test with Markov prior.

Hypothesis Testing with Adaptive Threshold

- The motion detection methods presented thus far assumed no knowledge about intensities of the moving object; *PM* was considered uniform. However, one may hope for a performance gain should amore accurate model of moving pixels be allowed.
- To build the local-in-space model at position *n*, ideally, one needs to identify all pixels of a moving object to which *n* belongs. Clearly, this labeling is not known in advance but since a

typical motion detection algorithm is both causal and iterative, one can use motion labels from previous frame or previous iteration. Below the latter approach is outlined.

MAP MRF Formulation

- Image segmentation has recently been studied in a framework of maximum a posteriori estimation for the Markov random field, where the cost function representing pixel-wise likelihood and inter-pixel smoothness should be minimized.

- The common drawback of these studies is the decrease in performance when a foreground object and the background have similar colors.

- The first term measures how well each label at ***n*** explains the observation *k* [***n***]. The other terms measure how contiguous the labels are in the image plane (*Vs*) and in time (*Vt*). Both *Vs* and *Vt* can be specified similarly to (3.15), thus favoring spatial and temporal similarity of the labels.

MAP Variational Formulation

- Variational methods and partial differential equations (PDE) based methods [6,7] have been introduced to explicitly account for intrinsic geometry in a variety of problems including image segmentation, mathematical morphology and image denoising.

- The motion detection problem has been formulated in discrete domain; pixels were explicitly labeled as moving or stationary. Alternatively, as mentioned before, moving areas can be defined implicitly by closed contours; the problem can be formulated and solved in continuous domain, and the final solution–discretized. One possible approach is through variational formulation

Motion Estimation

- The knowledge of motion is essential for both the **compression and processing** of image sequences.

- Methods explicitly **reducing the number of bits** needed to represent a video sequence will be classified as video compression techniques.

- In motion-compensated temporal interpolation (Fig. 3.2) **the task is to compute new images located between existing images of a video sequence** (e.g., video frame rate conversion between NTSC and PAL scanning standards).

- To develop a motion estimation algorithm, three important elements need to be considered: **models, estimation criteria, and search strategies.**

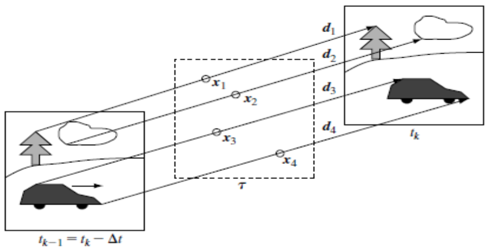


FIGURE 3.2

Motion-compensated interpolation between images at time $t_k - \Delta t$ and t_k. Motion compensation is essential for smooth rendition of moving objects. Shown are four motion vectors that map the corresponding image points at time $t_k - \Delta t$ and t_k onto image at time τ.

- There exist two fundamental models in motion estimation:

  1. A model relating **motion parameters to image intensities** called an observation model.

  2. The computation of motion is carried out **indirectly by examining intensity changes in time.**

Spatial Motion Models

- The goal is to estimate the motion of image points, that is, the ***2D* motion or apparent motion.** Such motion is a combination of projections of the motion of objects in a 3D scene and of 3D camera motion.

- This type of motion depends on the following:

  1. image formation model, for example, perspective, orthographic projection.

  2. motion model of 3D object, for example, rigid-body with 3D translation and rotation, 3D affine motion.

  3. surface model of 3D object, for example, planar, parabolic.

Temporal Motion Models

- The trajectories of individual image points drawn in the *(x*, *y*, *t)* space of an image sequence can be fairly arbitrary since they depend on object motion.

- The model is based on two velocity (linear) variables and two acceleration (quadratic) variables ***a*** *(a*1,*a*2*)T* , thus accounting for second-order effects.

- This model has been demonstrated to greatly benefit such motion-critical tasks as frame rate conversion [21] because of its improved handling of variable-speed motion present in typical videoconferencing images (e.g., hand gestures, facial expressions).

Region of Support

- The set of points ***x*** to which **spatial and temporal motion models apply is called the region of support,** denoted by *R*.

- The selection of a motion model and region of support is one of the major factors determining the precision of the resulting motion parameter estimates.

- Typically, the region of support for a motion model belongs to one of the four types listed below.

1. ***R* = the whole image** - A single motion model applies to all image points.(Single Object)

2. ***R* = one pixel** - This model applies to a single image point.(Multiple Objects)

***3. R* = rectangular block of pixels** - This motion model applies to a rectangular (or square) block of image points.(Digital Video Compression Models – Spatial / Temporal)

***4. R* = irregularly-shaped region** - This model applies to all pixels in region *R* of arbitrary shape.(3D surface / 3D motion objects)

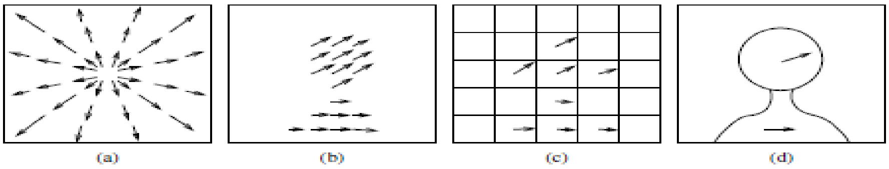


FIGURE 3.3

Schematic representation of motion for the four regions of support $\mathcal{R}$: (a) whole image, (b) pixel, (c) block, and (d) arbitrarily-shaped region. The implicit underlying scene is "head-and-shoulders" as captured by the region-based model (d).

Observation Models

- Since the goal is to **estimate motion based on intensity variations in time**, the relationship between **motion parameters and image intensities plays a very important role.**
- The usual, and reasonable, assumption made in this context is that **objects do not change their appearance as they move, that is, image intensity remains constant along motion trajectory.**

Estimation Criteria

- 1. ***Pixel-Domain Criteria -*** discrete version of the constant-intensity assumption.
- ***2. Frequency-Domain Criteria -*** (2D) Fourier transform of the intensity signal.
- ***3. Regularization -*** maintain motion resolution at the level of original images, the pixel-wise motion constraint.
- ***4. Bayesian Criteria -*** Bayesian criteria form a very powerful priori probability distribution.

Practical Motion Estimation Algorithms

Global Motion Estimation

- ➢ Global motion estimation (GME) is a technique that attempts to map one image onto another with a simple four-corner pin.

  - This differs from local motion estimation (LME), which attempts to find where each individual pixel in the image is in the other image.

  - GME is much cheaper to compute than LME, but gives you less information about the image.

  - Nevertheless, it is still very powerful for a variety of applications.

- **translation** - which allows the four corners to translate by the same amount,

- **rotation** - which allows the corners to rotate about their center,

- **scale** - which allows the size of the area defined by the corners to change,

- **perspective** - which allows the angles at the corners to change, so that the area defined by them is no longer a rectangle.

- Why do we want to estimate the global motion?

- – Panoramic stitching

- – Medical image registration

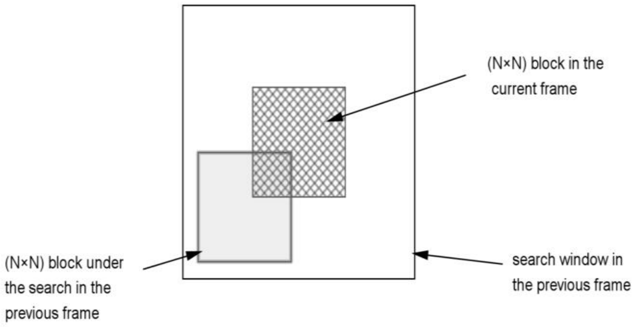
- – Change detection

- – Video stabilization

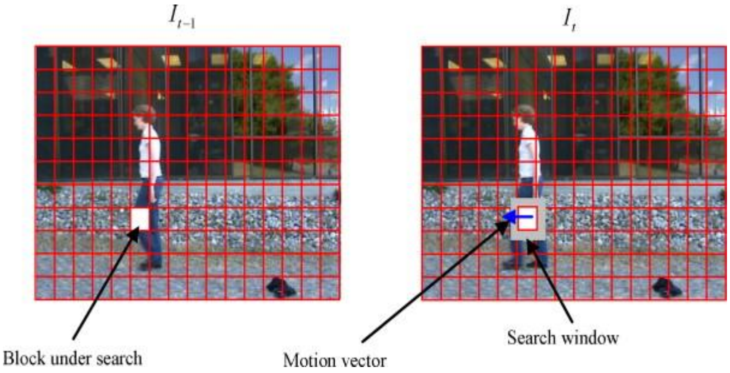
- – Moving object detection

Block Matching

A block matching algorithm involves **dividing the current frame of a video into macroblocks and comparing each of the macroblocks with a corresponding block and its adjacent neighbors in a nearby frame of the video** (sometimes just the previous one).

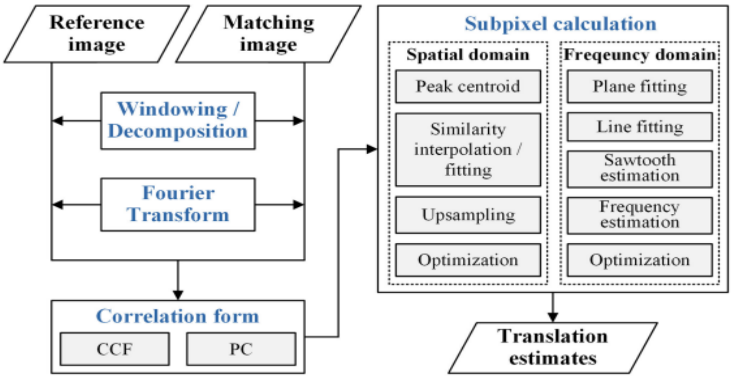

(N×N) block in the current frame
(N×N) block under the search in the previous frame
search window in the previous frame

The block-based motion estimation algorithm is mainly aimed to estimate the motion (motion vector) between macro block of current frame and perfectly matched candidate block of reference frame. The simple matching criterion is sum of absolute difference (SAD).



Phase Correlation

- Phase correlation is an approach to estimate the relative translative offset between two similar images (digital image correlation) or other data sets. It is commonly used in image registration and relies on a frequency-domain representation of the data, usually calculated by fast Fourier transforms.



- ➢ Divide
- ➢ Compute
- ➢ take the inverse FFT
- Use the dominant peak coordinates as the candidate vectors for 1616-pixel block matching.

Optical Flow via Regularization

- Optical flow or optic flow is **the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene**. Optical flow can also be defined as the distribution of apparent velocities of movement of brightness pattern in an image.

- Optical flow, or motion estimation, is **a fundamental method of calculating the motion of image intensities, which may be ascribed to the motion of objects in the scene**. Optical flow is an extremely fundamental concept that is utilized in one form or another in most video-processing algorithms.

MAP Estimation of Dense Motion

- In Bayesian statistics, a maximum a posteriori probability (MAP) estimate is **an estimate of an unknown quantity, that equals the mode of the posterior distribution**. The MAP can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data.

191CS72 - Image and Video Analytics

Unit-III Notes:

Video Enhancement and Restoration

Introduction:

Even with the advancing camera and digital recording technology, there are many situations in which recorded image sequences—or video for short—may suffer from severe degradations. The poor quality of recorded image sequences may be due to, for instance, the imperfect or uncontrollable recording conditions, such as one encounters in astronomy, forensic sciences, and medical imaging. Video enhancement and restoration has always been important in these application areas not only to improve the visual quality but also to increase the performance of subsequent tasks such as analysis and interpretation.

Another important application of video enhancement and restoration is preserving motion pictures and video tapes recorded over the last century. These unique records of historic, artistic, and cultural developments are deteriorating rapidly due to aging effects of the physical reels of film and magnetic tapes that carry the information. The preservation of these fragile archives is of interest not only to professional archivists but also to broadcasters as a cheap alternative to fill the many television channels that have come available with digital broadcasting.

An important difference between the enhancement and restoration of 2D images and video is the amount of data to be processed. While for the quality improvement of important images elaborate processing is still feasible, this is no longer true for the absolutely huge amounts of pictorial information encountered in medical sequences and film/video archives.

The most common artifact encountered in the above-mentioned applications is noise. Over the last two decades, an enormous amount of research has focused on the problem of enhancing and restoring 2D images. Clearly, the resulting spatial methods are also applicable to image sequences, but such an approach implicitly assumes that the individual pictures of the image sequence, or frames, are temporally independent. By ignoring the temporal correlation that exists, suboptimal results may be obtained, and the spatial intra frame filters tend to introduce temporal artifacts in the restored image sequences.

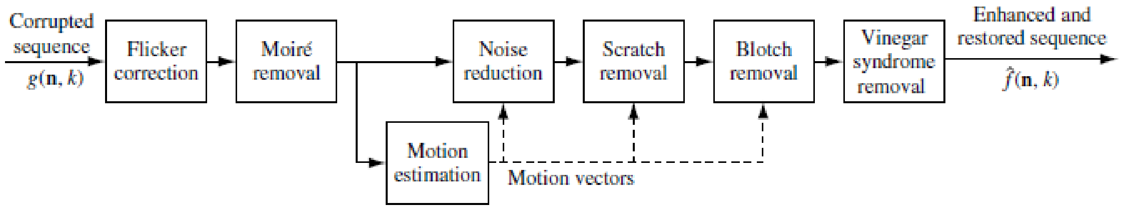


FIGURE 4.1

Some processing steps in the removal of various video artifacts.

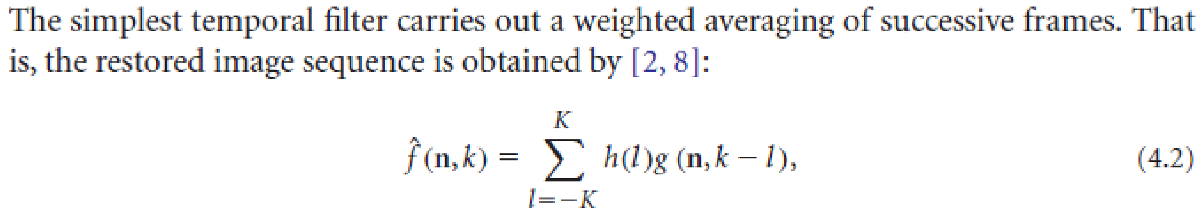
SPATIOTEMPORAL NOISE FILTERING

Any recorded signal is affected by noise, no matter how precise the recording equipment. The sources of noise that can corrupt an image sequence are numerous (see Chapter 7 of *The Essential Guide to Image Processing* [1]). Examples of the more prevalent ones include camera noise, shot noise originating in electronic hardware and the storage on magnetic tape, thermal noise, and granular noise on film.

The objective of noise reduction is to make an estimate ˆ *f (***n**,*k)* of the original image sequence given only the observed noisy image sequence *g (***n**,*k)*. Many different approaches toward noise reduction are known, including optimal linear filtering, nonlinear filtering, scale-space processing, and Bayesian techniques. In this section, we discuss successively the class of linear image sequence filters, OS filters, and multiresolution filters.

Linear Filters

Temporally Averaging Filters



The simplest temporal filter carries out a weighted averaging of successive frames. That is, the restored image sequence is obtained by [2, 8]:

$$\hat{f}(\mathbf{n},k) = \sum_{l=-K}^{K} h(l)g\,(\mathbf{n},k-l), \tag{4.2}$$

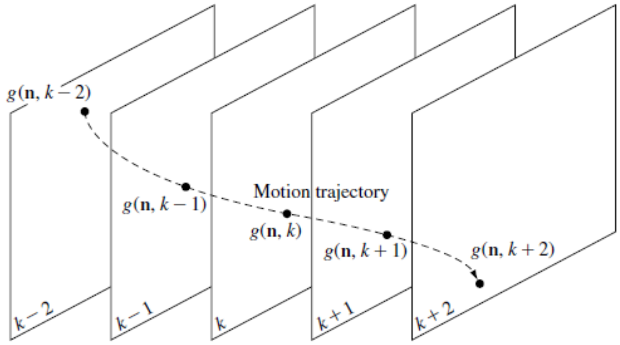


FIGURE 4.2

Noise filter operating along the motion trajectory of the picture element $(\mathbf{n},k)$.

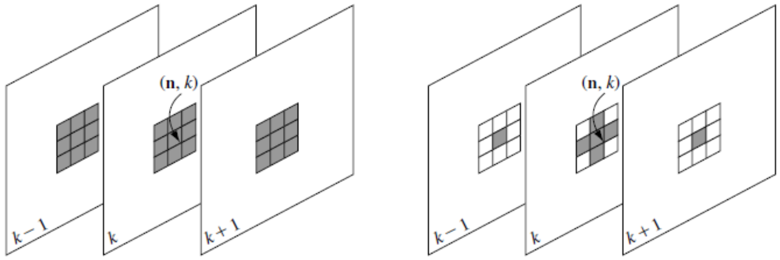


FIGURE 4.3

Examples of spatiotemporal windows to collect data for noise filtering of the picture element $(\mathbf{n},k)$.

Temporally Recursive Filters

A disadvantage of the temporal filter (4.2) and spatiotemporal filter (4.7) is that they need to buffer several frames of an image sequence. Alternatively, a recursive filter structure can be used that generally needs to buffer fewer (usually only one) frames.

More elaborate variations of (4.10) make use of a local estimate of the signal's mean within a spatiotemporal neighborhood. Furthermore, Eq. (4.9) can also be cast into a formal 3D motion-compensated Kalman estimator structure [12, 13]. In this case, the prediction ˆ $f_b(\mathbf{n},k)$ depends directly on the dynamic spatiotemporal state-space equations used for modeling the image sequence.

Order-Statistic Filters

Order-statistic filters are nonlinear variants of weighted-averaging filters. The distinction is that in OS filters the observed noisy data—usually taken from a small spatiotemporal window—is ordered before being used. Because of the ordering operation, correlation information is ignored in favor of magnitude information. Examples of simple OS filters are the minimum operator, maximum operator, and median operator. OS filters are often applied in directional filtering.

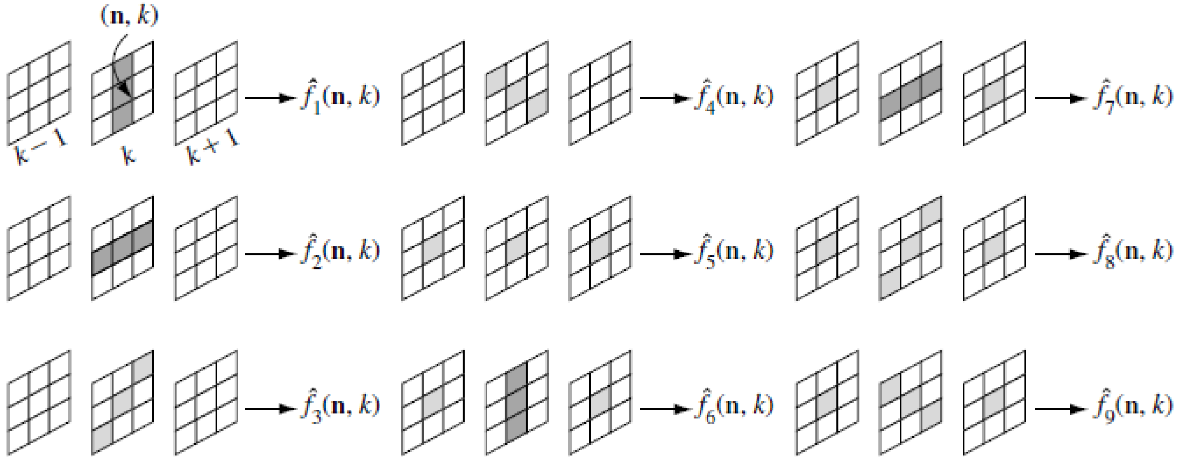

(n, k)
k − 1
k
k + 1
$\hat{f}_1(\mathbf{n}, k)$
$\hat{f}_2(\mathbf{n}, k)$
$\hat{f}_3(\mathbf{n}, k)$
$\hat{f}_4(\mathbf{n}, k)$
$\hat{f}_5(\mathbf{n}, k)$
$\hat{f}_6(\mathbf{n}, k)$
$\hat{f}_7(\mathbf{n}, k)$
$\hat{f}_8(\mathbf{n}, k)$
$\hat{f}_9(\mathbf{n}, k)$

FIGURE 4.4

Spatiotemporal windows used in the multistage median filter.

An additional advantage of ordering the noisy observation prior to filtering is that outliers can easily be detected. For instance, with a statistical test—such as the rank order test [16]—the observed noisy values within the spatiotemporal window *S* that are significantly different from the intensity *g* (**n**,*k)* can be detected.

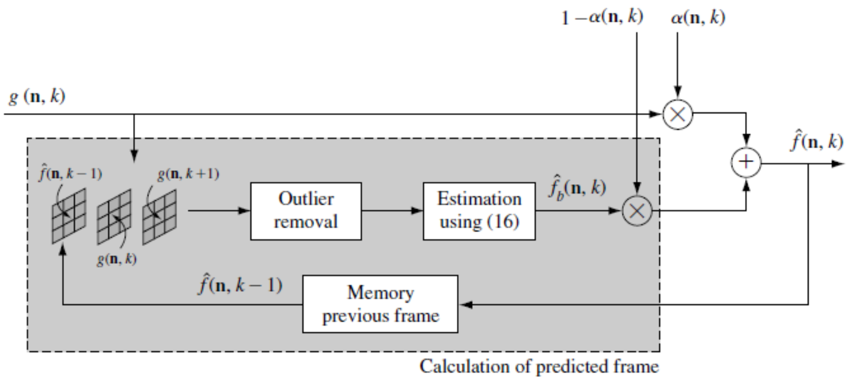


FIGURE 4.5

Overall filtering structure combining (4.9), (4.16), and an outlier removing rank order test.

Multiresolution Filters

The multiresolution representation of 2D images has become quite popular for analysis and compression purposes [17]. This signal representation is also useful for image sequence restoration. The fundamental idea is that if an appropriate decomposition into bands of different spatial and temporal resolutions and orientations is carried out, the energy of the structured signal will locally be concentrated in selected bands, whereas the noise is spread out over all bands.

The discrete wavelet transform has been widely used for decomposing one dimensional and multidimensional signal into bands.

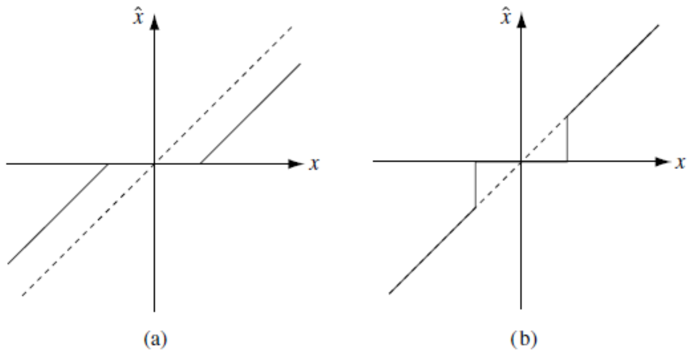
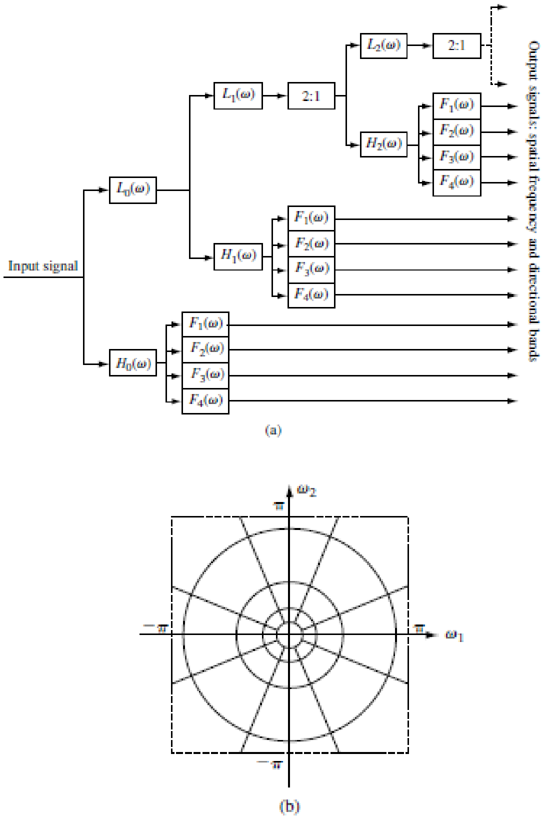


FIGURE 4.6

Coring functions: (a) Soft-thresholding. (b) Hard-thresholding. Here x is an signal amplitude taken from one of the spatiotemporal bands (which carry different resolution and orientation information), and $\hat{x}$ is the resulting signal amplitude after coring.

The Simoncelli pyramid gives a spatial decomposition of each frame into bands of different resolution and orientation. The extension to temporal dimension is obtained by temporally decomposing each of the spatial resolution and orientation bands using a regular wavelet transform. The low-pass and high-pass filters are operated along the motion trajectory to avoid blurring of moving objects. The resulting motion compensated spatiotemporal wavelet coefficients are filtered by one of the coring functions, followed by the reconstruction of the video frame by an inverse wavelet transformation and Simoncelli pyramid reconstruction.



(a)

(b)

FIGURE 4.7

(a) Simoncelli pyramid decomposition scheme. (b) Resulting spectral decomposition, illustrating the spectral contents carried by the different resolution and directional bands.

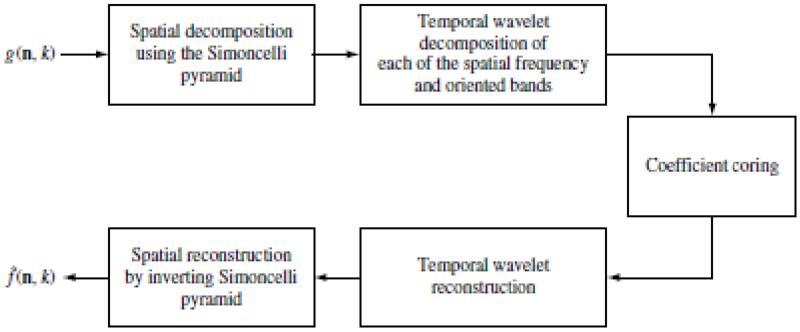

$g(\mathbf{n}, k)$
Spatial decomposition using the Simoncelli pyramid
Temporal wavelet decomposition of each of the spatial frequency and oriented bands
Coefficient coring
Temporal wavelet reconstruction
Spatial reconstruction by inverting Simoncelli pyramid
$\hat{f}(\mathbf{n}, k)$

FIGURE 4.8

Overall spatiotemporal multiresolution filtering using coring.

CODING ARTIFACT REDUCTION

The call for coding artifact reduction started with the introduction of digital transmission and storage. Theoretically, digital transmission can be lossless, but in practice, the available channel bandwidth makes lossy compression necessary, which results in visible artifacts.

The block-based DCT is among the most popular transform techniques and has been widely used in image and video compression. It is the core of several industry standards of image and video compression, such as JPEG, H.26x, and MPEG-x. But these block based codecs suffer from annoying blocking artifacts when they are applied in low bit rate coding to obtain a reasonable compression ratio.

The quantization is the lossy stage because the original values of the DCT coefficients are permanently lost. Because each block is encoded separately, there may emerge visible discontinuities along block boundaries of the decoded image frame, commonly defined as blocking artifacts. Also ringing, blurring, and mosquito noise can be introduced as a consequence of omitting the high frequencies by truncating the DCT coefficients [21]. Generally speaking, increasing coding bit rate can improve the quality of the reconstructed image/video, but it is limited by channel bandwidth or storage capacity.

Artifact Reduction in the Spatial Domain

Due to horizontal and vertical “edges” or lines demarking the block grid, there are additional high frequencies in the spectrum of the decoded video signal. The simplest remedy would be the application of low-pass filtering to suppress those frequencies. Initially, Gaussian low-pass filtering with a high-frequency emphasis has been proposed, or Gaussian filtering only applied to pixels near the block boundaries. The general drawback of these early methods is the loss of high frequencies and excessive blurring of true edges.

To overcome these drawbacks, adaptive deblocking algorithms have been proposed.In general, adaptive filtering requires a classification step to determine whether a block is amonotone one or contains an edge, followed by a (non) linear filtering step. If a block is monotone, a 2D filtering is applied; otherwise 1D directional filtering is applied [23, 24].

Finally, we mention the so-called projection onto convex sets (POCS)-based methods for blocking artifacts reduction, originating from image restoration. One tries to use a priori information about the original image data to iteratively restore a degraded (sub)image. The convergence behavior can be explained by the theory of POCS. In blockiness reduction, the general assumption is that that the input image is highly correlated so that similar frequency characteristics are maintained between adjacent blocks. If we are able to detect the high-frequency components that are not present within blocks, we can consider them the result of the blocking artifact. Some of the popular constraints such as the intensity and the smoothness constraint as well as other information about the POCS technique can be found in [26]. The proposed method [27] shows that blocking artifacts can be significantly alleviated, while the original high-frequency components, such as edges, are faithfully preserved. Although quite effective for still pictures, these POCS based methods are less practical for real-time video post processing implementations due to their computationally demanding iterative nature.

Artifact Reduction in the Frequency Domain

We have seen that spatial domain methods apply low-pass filtering and as such result in blurring of the picture. To alleviate this problem, we have discussed a few advanced methods as well. Relatively speaking, only a few approaches in the literature have tackled the problem of blockiness reduction in the frequency domain. These methods can either use coefficients available in the bit stream or re-compute DCT coefficients in the postprocessing.

An early method for removing blocking effects is proposed in [28]. It exploits the correlation between intensity values of boundary pixels of two neighboring blocks. Specifically, it is based on the empirical observation that quantization of theDCT coefficients of two neighboring blocks increases the mean squared difference of slope (MSDS) between the neighboring pixels on their boundaries. Therefore, among all permissible inverse quantized coefficients, the set that minimizes this MSDS is most likely to decrease the blocking effect. This minimization problem can be formulated as a computationally intensive quadratic programming (QP) problem.

BLOTCH DETECTION AND REMOVAL

Blotches are artifacts that are typically related to film. Dirt particles covering the film introduce bright or dark spots on the frames, and the mishandling or aging of film causes loss of gelatin covering the film. Figure 4.9(a) shows a film frame containing dark and bright spots:

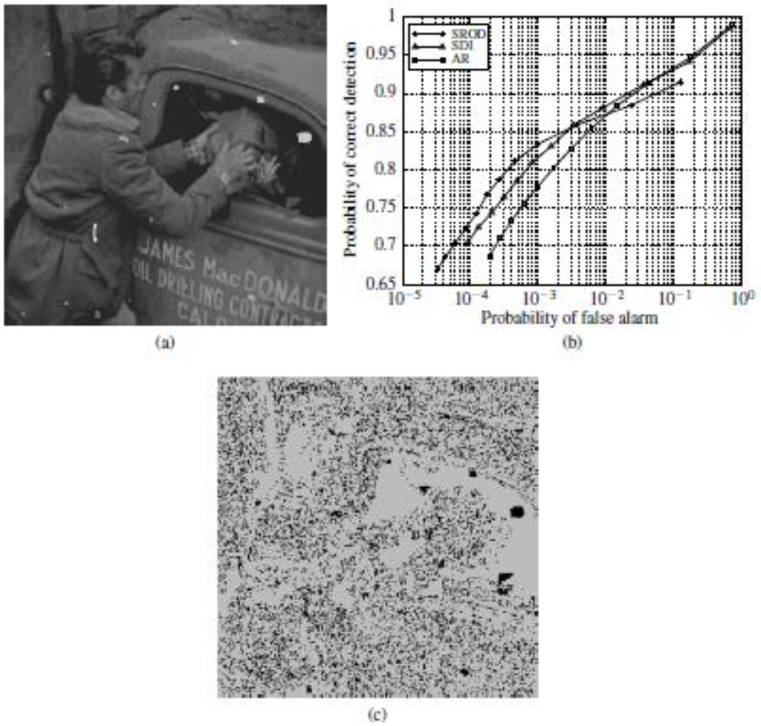


FIGURE 4.9

(a) Video frame with blotches. (b) Correct detection versus false detection for three different blotch detectors. (c) Blotch detection mask using the sROD ($T = 0$).

Blotch Detection

Blotches have three characteristic properties that are exploited by blotch detection algorithms. First, blotches are temporally independent and therefore hardly ever occur at the same spatial location in successive frames. Second, the intensity of a blotch is significantly different from its neighboring uncorrupted intensities. Finally, blotches form coherent regions in a frame, as opposed to, for instance, spatiotemporal shot noise.

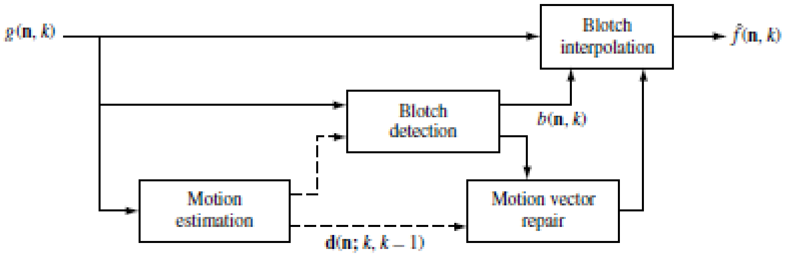


FIGURE 4.10

Blotch detection and removal system.

There are various blotch detectors that exploit these characteristics. The first is a pixel-based blotch detector known as the spike-detector index (SDI). This method detects temporal discontinuities by comparing pixel intensities in the current frame with motion compensated reference intensities in the previous and following frames:

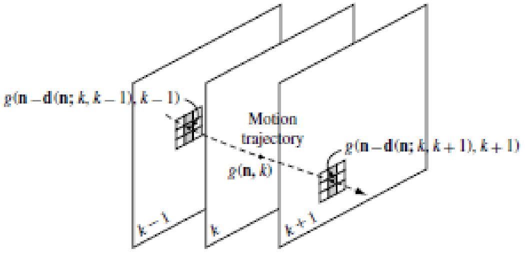

$g(\mathbf{n}-\mathbf{d}(\mathbf{n}; k, k-1), k-1)$
Motion trajectory
$g(\mathbf{n}-\mathbf{d}(\mathbf{n}; k, k+1), k+1)$
$g(\mathbf{n}, k)$
$k-1$
k
$k+1$

FIGURE 4.11

Example of motion-compensated spatiotemporal window for obtaining reference intensities in the ROD detector.

Motion Vector Repair and Interpolating Corrupted Intensities

Block-based motion estimators will generally find the correct motion vectors even in the presence of blotches, provided that the blotches are small enough. The disturbing effect of blotches is usually confined to small areas of the frames. Hierarchical motion estimators will experience little influence of the blotches at the lower resolution levels. At higher resolution levels, blotches covering larger parts of (at those levels) small blocks will significantly influence the motion estimation result.

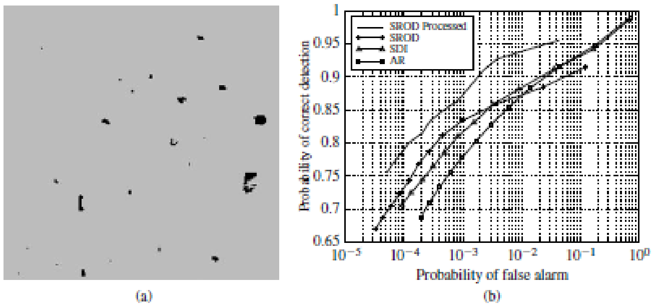


FIGURE 4.12

(a) Blotch detection mask after postprocessing. (b) Correct detection versus false detections obtained for sROD with postprocessing (top curve), compared to results from Fig. 4.9(b).

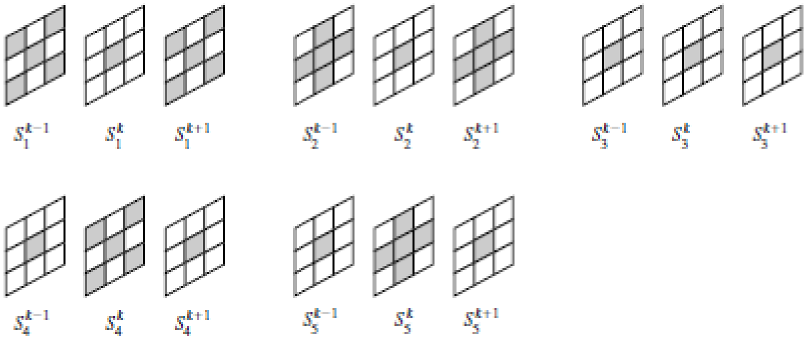


FIGURE 4.13

Five spatiotemporal windows used to compute the partial results in Eq. (4.25).

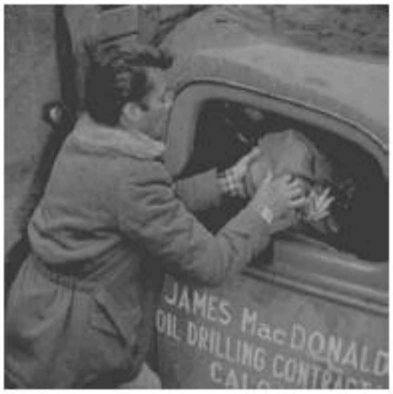


FIGURE 4.14

Blotch-corrected frame resulting from Figure 4.9a.

Video Inpainting

In the last decade, several types of blotch removal methods were proposed that are able to fill in large missing areas in still images [34–38] Initially, these spatial restoration algorithms could be classified mainly in two categories: smoothness-preserving and texture synthesis. The first category comprises various inpainting approaches which usually propagate isophotes (i.e., level lines), gradients, or curvatures inside the artifact by means of variational models.

As their naming suggests, texture synthesis methods do a better reconstruction of the textural content. However, they do not preserve object edges properly. This is done better with smoothness-preserving methods, which, on their turn, do not reproduce textural content. To combine the advantages of both methods, a third category of algorithms has appeared comprising hybrid methods that combine structure and texture reconstruction

Restoration in Conditions of Difficult Object Motion

Due to various types of complicated object movements [54], wrong motion vectors are sometimes extracted from the sequence. As a result, the spatiotemporal restoration process that follows may introduce unnecessary errors that are visually more disturbing than the blotches themselves. The extracted temporal information becomes unreliable, and a source of errors by itself. This triggered research on situations in which the motion vectors cannot be used.

An initial solution to the problem was to detect areas of “pathological” motion and protect them against any restoration [54]. This solution, however, preserves the artifacts which happen to lie in those areas. To restore these artifacts, too, several solutions have been proposed, which discard the temporal information and use only spatial information coming from the same frame.

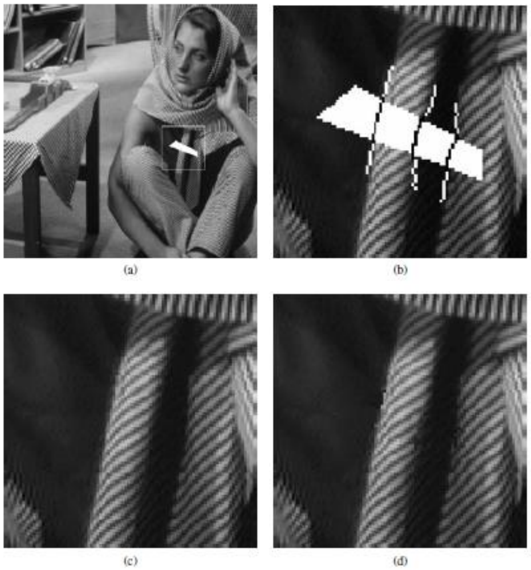


FIGURE 4.15

Constrained texture synthesis. (a) Original image, with artificial artifact surrounded by a white box. (b) Structure. (c) Original content of the artifact. (d) Restoration result.

VINEGAR SYNDROME REMOVAL

Content stored on acetate-based film rolls faces deterioration at a progressive pace. These rolls can be affected by dozens of types of film artifacts, each of them having its own properties and showing up in particular circumstances. One of the major problems of all film archives is the vinegar syndrome [38]. This syndrome appears when, in the course of their chemical breakdown, the acetate-based film bases start to release acetic-acid, giving a characteristic vinegar smell. It is an irreversible process, and from a certain moment on, it becomes auto-catalytic, progressively fuelling itself in the course of time.

The vinegar syndrome has various appearances. It may show up as a partial loss of color, bright or dark tree-like branches, nonuniformly blurred images.Here, we only focus on one type of vinegar syndrome, namely the partial loss of color.

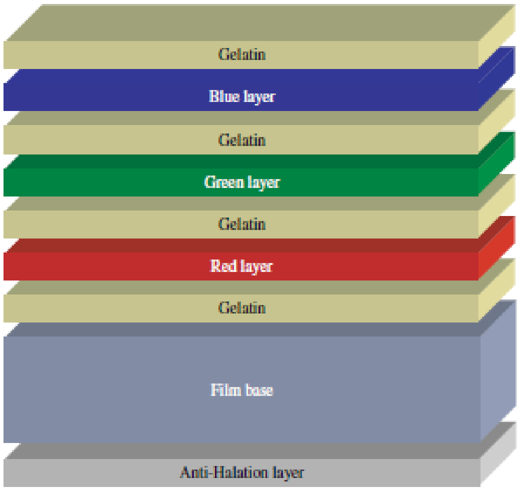


FIGURE 4.16

The layered structure of a film.

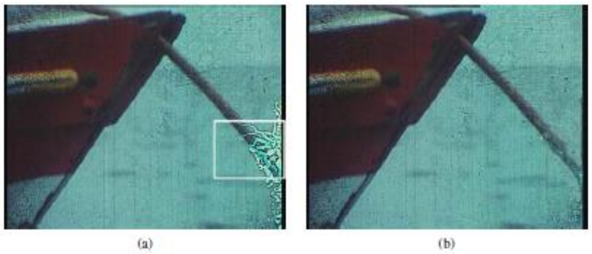


FIGURE 4.17

Restoration example (sequence courtesy of RTP-Radiotelevisão Portuguesa). (a) Original frame, with artifact surrounded by a white box. (b) Restored frame.

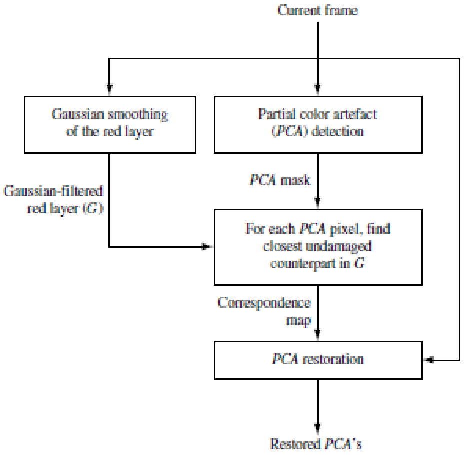

Current frame
Gaussian smoothing of the red layer
Partial color artefact (PCA) detection
Gaussian-filtered red layer (G)
PCA mask
For each PCA pixel, find closest undamaged counterpart in G
Correspondence map
PCA restoration
Restored PCA's

FIGURE 4.18

Restoration scheme for the vinegar syndrome.

191CS72 - Image and Video Analytics

Unit-IV Notes:

VIDEO SEGMENTATION AND TRACKING

Introduction:

Video segmentation refers to partitioning video into spatial, temporal, or spatiotemporal regions that are homogeneous in some feature space

It is an integral part of many video analysis and coding problems, including (i) video summarization, indexing, and retrieval, (ii) advanced video coding, (iii) video authoring and editing, (iv) improved motion (optical flow) estimation, (v) 3D motion and structure estimation with multiple moving objects [1–3], and (vi) video surveillance/understanding.

The first three applications concern multimedia services, which require temporal segmentation of video intoshots or groups of pictures (GoP). The latter three are computer vision applications, where spatiotemporal segmentation helps to identify foreground and background objects, as well as optical flow boundaries (motion edges) and occlusion regions.

Factors that affect the choice of a specific segmentation method include the following:

1) ■ *Real-time performance*
2) *Precision of segmentation*
3) *Scene complexity*

SCENE CHANGE DETECTION

Scene change or shot boundary detection is a relatively easy segmentation problem since it is one dimensional, along the temporal dimension. Shot boundary detection methods locate temporal discontinuities, that is, frames across which large differences are observed in some feature space, usually a combination of color and motion.

Temporal discontinuities may be abrupt (cuts) or gradual (special effects, such as wipes and fades). It is easier to detect cuts than special effects. The simplest approach for detecting temporal discontinuities is to quantify frame differences in the pixel intensity domain.

Videos are almost always stored and transmitted in compressed form. Detection of scene changes in real time poses a challenge in many applications since decompressing and processing video data sequentially requires significant computational resources. Hence, the need for scene segmentation algorithms in the compressed domain (without completely decoding the bit stream).

SPATIOTEMPORAL CHANGE DETECTION

Change detection methods segment each frame into two regions, namely changed and unchanged regions in the case of a static camera or global and local motion regions in the case of a moving camera [22]. This section deals only with the former case, where unchanged regions correspond to the background (null hypothesis) and changed regions to the foreground object(s) or uncovered (occlusion) areas.

Various change detection methods in the literature differ according to (i) what features and background model are used, (ii) what distance metrics are used, and (iii) what kind of threshold and background model adaptation rules are used.

The simplest method to detect changes between two registered frames would be to analyze the frame difference (FD) image, which is given by FDk,r(x) s(x,k)

s(x,r), (6.1) where x (x1,x2) denotes pixel location and s(x,k) stands for the intensity value at pixel x in frame k. FD image shows the pixel-by-pixel difference between the current image k and the reference image r.

Temporal Integration:

An important consideration is to add memory to the motion detection process to ensure both spatial and temporal continuity of the changed regions at each frame. This can be achieved in a number of different ways, including temporal filtering (integration) of the intensity values across multiple frames before thresholding and postprocessing of labels after thresholding.

An alternative procedure that was adopted by MPEG-4 as a non-normative tool considers postprocessing of labels [25]. First, scene changes are detected. Within each scene (shot), an initial change detection mask is estimated between successive pairs of frames by global thresholding of the FD function.

Combination with Spatial Segmentation

Another consideration is to enforce consistency of the boundaries of the changed regions with spatial edge locations at each frame. This may be accomplished by first segmenting each frame into uniform color and/or texture regions. Next, each region resulting from the spatial segmentation is labeled as changed or unchanged as a whole as opposed to labeling each pixel independently.

- **MOTION SEGMENTATION**

Motion segmentation (also known as optical flow segmentation) methods label pixels (or optical flow vectors) at each frame that are associated with independently moving part of a part of a scene. The region boundaries may or may not be pixel-accurate or semantically meaningful. For example, a single object with articulated motion may be segmented into multiple regions.

Motion segmentation is closely related to two other problems, motion (change) detection and motion estimation. Change detection is a special case of motion segmentation with only two regions, namely changed and unchanged regions (in the case of a static camera) or global and local motion regions (in the case of a moving camera). An important distinction between change detection and motion segmentation is that the former can be achieved without motion estimation if the scene is recorded with a static camera. Change detection in the case of a moving camera and general motion segmentation, on the other hand, requires some sort of global and/or local motion estimation either explicitly or implicitly.

In general, application of standard image segmentation methods directly to estimated optical flow vectors may not yield meaningful results, since an object moving in 3D usually generates a spatially varying optical flow field [28]. For example, in the case of a rotating object, there is no flow at the center of the rotation, and the magnitude of the flow vectors grows as we move away from the center of rotation. Therefore, a parametric model-based approach, where we assume that the motion field can be described by a set of K parametric models, is usually adopted. In parametric motion segmentation, the model parameters are the motion features. Then, motion segmentation algorithms aim to determine the number of motion models that can adequately describe a scene, type/complexity of these motion models, and the spatial support of each motion model. Most commonly used types of parametric models are affine, perspective, and quadratic mappings, which assume a 3D planar surface in motion. In the case of a nonplanar object, the resulting optical flow can be modeled by a piecewise affine, perspective, or quadratic flow field if we approximate the object surface by a union of a small number of planar patches. Since each independently moving object and/or planar patch will best fit a different parametric model, the parametric approach may lead to over segmentation of motion in the case of nonplanar objects.

- **Dominant Motion Segmentation**

Segmentation by dominant motion analysis refers to extracting one object (with the dominant motion) from the scene at a time [24, 29, 30, 38]. Dominant motion segmentation can be considered a hierarchically structured top-down approach, which starts by fitting a single parametric motion model to the entire frame and then partitions the frame into two regions, those pixels which are well represented by this dominant motion model and those that are not. The process converges to the dominant motion model in a few iterations, each time fitting a new model to only those pixels that are well represented by the motion model in the previous iteration. The dominant motion may correspond to the camera (background) motion or a foreground object motion, whichever occupies a larger area in the frame.

Segmentation Using Two Frames

Motion estimation in the presence of more than one moving objects with unknown supports is a difficult problem. It was Burt et al. [29] who first showed that the motion of a 2D translating object can be accurately estimated using a multiresolution iterative approach even in the presence of other independently moving objects without prior knowledge of their supports. This is, however, not always possible with more sophisticated motion models (e.g., affine and perspective), which are more sensitive to presence of other moving objects in the region of analysis.

To this effect, Irani et al. [24] proposed multistage parametric modeling of dominant motion. In this approach, first a translational motion model is employed over the whole image to obtain a rough estimate of the support of the dominant motion. The complexity of the model is then gradually increased to affine and projective models with refinement of the support of the object in between. The parameters of each model are estimated only over the support of the object based on the previously used model.

Temporal Integration

Temporal continuity of the estimated dominant objects can be facilitated by extending the temporal integration scheme introduced in Section 6.3.2. To this effect, we define an internal representation image

Multiple Motions

Multiple object segmentation can be achieved by repeating the same procedure on the residual image after each object is extracted. Once the first dominant object is segmented and tracked, the procedure can be repeated recursively to segment and track the next dominant object after excluding all pixels belonging to the first object from the region of analysis. Hence, the method is capable of segmenting multiple moving objects in a top-down fashion if a dominant motion exists at each stage.

Multiple Motion Segmentation

Multiple motion segmentation methods let multiple motion models compete against each other at each decision site. They consist of three basic steps, which are strongly interrelated: estimation of the number K of independent motions, estimation of model parameters for each motion, and determination of support of each model (segmentation labels). If we assume that we know the number K of motions and the K sets of motion parameters, then we can determine the support of each model. The segmentation procedure then assigns the label of the parametric motion vector that is closest to the estimated flow vector at each site. Alternatively, if we assume that we know the value of K and a segmentation map consisting of K regions, the parameters for each model can be computed in the least squares sense (either from estimated flow vectors or from spatiotemporal intensity values) over the support of the respective region. But since both the parameters and supports are unknown in reality, we have a chicken-egg problem; that is, we need to know the motion model parameters to find the segmentation labels, and the segmentation labels are needed to find the motion model parameters.

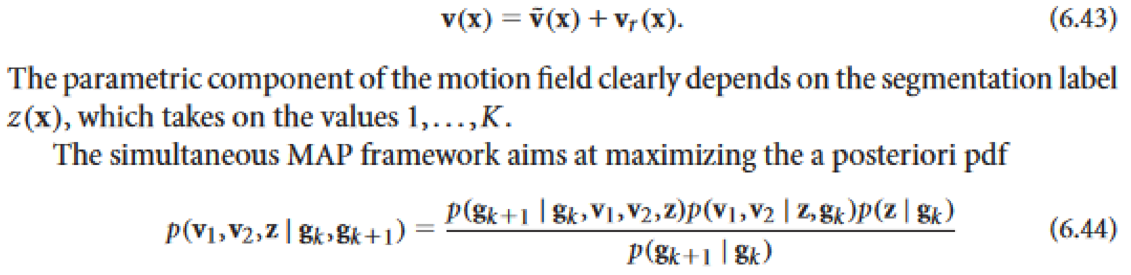
Simultaneous Motion Estimation and Segmentation

Up to now, we discussed methods to compute the segmentation labels from either precomputed optical flow or directly from intensity values but did not address how to compute an improved dense motion field along with the segmentation map. It is clear that the success of optical flow segmentation is closely related to the accuracy of the estimated optical flow field (in the case of using precomputed flow values) and vice versa.

It follows that optical flow estimation and segmentation should be addressed simulta-neously for best results. Here, we present a simultaneous Bayesian approach based on are presentation of the motion field as the sum of a parametric field and a residual field. The interdependence of optical flow and segmentation fields is expressed in terms of a Gibbs distribution within the MAP framework. The resulting optimization problem, to find estimates of a dense set of motion vectors, a set of segmentation labels, and a set of mapping parameters, is solved using the highest confidence first (HCF) and iterated conditional mode (ICM) algorithms.

- **Motion-Field Model and MAP Framework**

We model the optical flow field v(x) as the sum of a parametric flow field v˜(x) and a nonparametric residual field vr(x), which accounts for local motion and other modeling errors; that is,



$$\mathbf{v}(\mathbf{x}) = \tilde{\mathbf{v}}(\mathbf{x}) + \mathbf{v}_r(\mathbf{x}). \tag{6.43}$$

The parametric component of the motion field clearly depends on the segmentation label $z(\mathbf{x})$, which takes on the values $1, \ldots, K$.

The simultaneous MAP framework aims at maximizing the a posteriori pdf

$$p(\mathbf{v}_1, \mathbf{v}_2, \mathbf{z} \mid \mathbf{g}_k, \mathbf{g}_{k+1}) = \frac{p(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) p(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}, \mathbf{g}_k) p(\mathbf{z} \mid \mathbf{g}_k)}{p(\mathbf{g}_{k+1} \mid \mathbf{g}_k)} \tag{6.44}$$

with respect to the optical flow v1, v2, and the segmentation labels z, where v1 andv2 denote the lexicographic ordering of the first and second components of the flow vectors v(x) [v1(x) v2(x)] T at each pixel x. Through careful modeling of these pdfs, we can express an interrelated set of constraints that help improve both optical flow and segmentation estimates.

The first conditional pdf p(gk1 | gk ,v1,v2, z) provides a measure of how well the present displacement and segmentation estimates conform with the observed frame k 1 given frame k. It is modeled by a Gibbs distribution as

$$p(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) = \frac{1}{Q_1} \exp\{-U_1(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z})\}, \tag{6.45}$$

where Q_1 is the partition function (normalizing constant) and

$$U_1(\mathbf{g}_{k+1} \mid \mathbf{g}_k, \mathbf{v}_1, \mathbf{v}_2, \mathbf{z}) = \sum [g_k(\mathbf{x}) - g_{k+1}(\mathbf{x} + \mathbf{v}(\mathbf{x})\Delta t)]^2 \tag{6.46}$$

is called the Gibbs potential. Here, the Gibbs potential corresponds to the norm square of the displaced FD (DFD) between the frames gk and gk1. Thus, maximization of imposes the constraint that v(x) minimizes the DFD.

The second term in the numerator in is the conditional pdf of the displacement field given the motion segmentation and the search image. It is also modeled by a Gibbs distribution,

$$p(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}, \mathbf{g}_k) = p(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}) = \frac{1}{Q_2} \exp\{-U_2(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z})\}, \tag{6.47}$$

where Q_2 is a constant and

$$\begin{aligned} U_2(\mathbf{v}_1, \mathbf{v}_2 \mid \mathbf{z}) = {} & \alpha \sum_{\mathbf{x}} ||\mathbf{v}(\mathbf{x}) - \tilde{\mathbf{v}}(\mathbf{x})||^2 \\ & + \beta \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} ||\mathbf{v}(\mathbf{x}_i) - \mathbf{v}(\mathbf{x}_j)||^2 \, \delta(z(\mathbf{x}_i) - z(\mathbf{x}_j)) \end{aligned} \tag{6.48}$$

is the corresponding Gibbs potential, || · || denotes the Euclidian distance, and Nx is the set of neighbors of site x. The first term in enforces a minimum norm estimate of the residual motion field vr(x); that is, it aims to minimize the deviation of the motion field v(x) from the parametric motion field v˜(x) while minimizing the DFD.

Note that the parametric motion field v˜(x) is calculated from the set of model para-meters ai, i 1,...,K, which in turn is a function of v(x) and z(x). The second term in (6.48) imposes a piecewise local smoothness constraint on the optical flow estimates without introducing any extra variables such as line fields. Observe that this term is active only for those pixels in the neighborhood Nx, which share the same segmentation label with the site x. Thus, spatial smoothness is enforced only on the flow vectorsgenerated by a single object. The parameters and allow for relative scaling of the two terms.

The third term in (6.44) models the a priori probability of the segmentation field in a manner similar to that in MAP segmentation. It is given by

$$p(\mathbf{z} \mid \mathbf{g}_k) = p(\mathbf{z}) = \frac{1}{Q_3} \sum_{\omega \in \Omega} \exp\{-U_3(\mathbf{z})\}\delta(\mathbf{z} - \omega), \qquad (6.49)$$

where denotes the sample space of the discrete-valued random vector z, and Q3 and U3(z) are as defined in and, respectively. The dependence of the labels on the image intensity is usually neglected, although region boundaries generally coincide with intensity edges.

- **Two-Step Iteration Algorithm**

Maximizing the a posteriori pdf is equivalent to minimizing the cost function,

$$E= U1(gk+1 \mid gk ,v1,v2, z)= U2(v1,v2 \mid z) + U3(z)$$

that is composed of the potential functions .Direct minimization of with respect to all unknowns is an exceedingly difficult prob-lem, because the motion and segmentation fields constitute a large set of unknowns.To this effect, we perform the minimization of through the following two-step iterations [48]:

1) Given the best available estimates of the parameters ai, i 1,...,K, and z, update the optical flow field v1, v2. This step involves the minimization of a modified cost function

which is composed of all terms in that contain v(x). Although the first term indicates how well v(x) explains our observations, the second and third terms impose prior constraints on the motion estimates that they should conform with the parametric flow model and that they should vary smoothly within each region. To minimize this energy function, we employ the HCF method recently proposed by Chou and Brown [49]. HCF is a deterministic method designed to efficiently handle the optimization of multivariable problems with neighborhood interactions.

2) Update the segmentation field z, assuming that the optical flow field v(x)is known. This step involves the minimization of all the terms in (6.50), which contain z as well as v˜(x)

The first term in (6.52) quantifies the consistency of v˜(x) and v(x). The second termis related to the a priori probability of the present configuration of the segmentation labels. We use an ICM procedure to optimize E2 [48]. The mapping parameters ai are updated by least squares estimation within each region.

An initial estimate of the optical flow field can be found by using the Bayesian approach with a global smoothness constraint. Given this estimate, the segmentation labels can be initialized by a procedure similar to Wang and Adelson's [37]. The determi- nation of the free parameters , and is a design problem. One strategy is to choose them to provide a dynamic range correction so that each term in the cost function has equal emphasis. However, because the optimization is implemented in two steps, theratio /also becomes consequent. We recommend to select 1/5, depending on how well the motion field can be represented by a piecewise-parametric model andwhether we have a sufficient number of classes.

A hierarchical implementation of this algorithm is also possible by forming successive low-pass filtered versions of the images gk and gk1. Thus, the quantities v1, v2, and z can be estimated at different resolutions. The results of each hierarchy are used to initialize the next lower level. Note that the Gibbsian model for the segmentation labels has been extended to include neighbors in scale by Kato et al. [53].Several other motion analysis approaches can be formulated as special cases of this framework. If we retain only the first and the third terms in (6.50) and assume that all sites possess the same segmentation label, then we have Bayesian motion estimation with a global smoothness constraint. The motion estimation algorithm proposed by Iu [47] uses the same two terms but replaces the (·) function by a local outlier rejection function. The motion estimation and region labeling algorithm proposed by Stiller [52] involve all terms in (6.50) except the first term in (6.48). Furthermore, the segmentation labels in Stiller's algorithm are used merely as tokens to allow for a piecewise smoothness constraint on the flow field and do not attempt to enforce consistency of the flow vectors with a parametric component. We also note that the motion estimation method of Konrad and Dubois [51], which use line fields, are fundamentally different in that they model discontinuities in the motion field rather than modeling regions that correspond to different parametric motions. On the other hand, the motion segmentation algorithm of Murray and Buxton [34] (Section 6.4.2.3) employs only the second term in and third term in (6.50) to model the conditional and prior pdf, respectively. Wang and Adelson [37] relies on the first term in (6.48) to compute the motion segmentation (Section 6.4.2.2). However, they also take the DFD of the parametric motion vectors into consideration when the closest match between the estimated and parametric motion vectors, represented by the second term, exceeds a threshold.

SEMANTIC VIDEO OBJECT SEGMENTATION

So far, we discussed methods for automatic motion segmentation. However, it is difficult to achieve semantically meaningful object segmentation using fully automatic methods based on low-level features such as motion, color, and texture. This is because a semantic object may contain multiple motions, colors, textures, and so on, and definition of seman- tic objects may depend on the context, which may not be possible to capture by low-level features. Thus, in this section, we present two approaches that can extract semantically meaningful objects using capture-specific information or user interaction.

Chroma-Keying

Chroma-keying is an object-based video capture technology where each video object is recorded individually in a special studio against a key color. The key color is selected such that it does not appear on the object to be captured. Then, the problem of extracting the object from each frame of video becomes one of color segmentation. Chroma-keyed video capture requires special attention to avoid shadows and other nonuniformity in the key color within a frame; otherwise, segmentation of key color may become a nontrivial problem**.**

Semiautomatic Segmentation

Because chroma-keying requires special studios and/or equipment to capture video objects, an alternative approach is interactive segmentation using automated tools to aid a human operator. To this effect, we assume that the contour of the first occurrence of the semantic object of interest is marked interactively by a human operator. Although detection of moving regions (by change detection methods) may result in semantically meaningful objects in well-constrained settings, in an unconstrained environment, user interaction is indeed the only way to define a semantically meaningful object unambigu-ously because only the user can know what is semantically meaningful in the context of an application. For example, if we have the video clip of a person carrying a ball, whether the ball and the person are two separate objects or a single object may depend on the application. Once the boundary of the object of interest is interactively determined in one or more key frames, its boundary in all other frames can be automatically computed by 2D motion tracking until the object exits the field of view.

2D object tracking is closely related to the problem of spatiotemporal segmentation in the sense it provides temporally linked spatial segmentation maps. The general approach can be summarized as projecting the current segmentation map into the next frame using 2D motion information. The projected region can be updated by morphological or other operators using the color and edge information in the next frame. This update step allows fine tuning of the segmentation map to alleviate motion estimation errors as well as including newly uncovered regions in the segmentation map. Object tracking methods can be classified as feature-point-based, contour-based, and region-based track-ing methods[4]. Feature points are points on the object (current segmentation map) that can be used as markers, such as corner points. Motion of these points can be found by gradient-based (e.g., Lukas–Kanade) or matching-based (e.g., block matching) methods. Goodness of tracking results at each feature point can be evaluated at each frame and some feature points can be removed and others may be added [54]. In contour-based methods, the tracking step defines a polygonal or spline approximation of the boundary of the video object, which may be further refined automatically or interactively using appropriate software tools [55, 56]. In region-based methods, the object region is repar-titioned into color- and/or motion-homogeneous subregions, and each subpartition isprojected into the next frame individually with or without using subregion connectivity constraints [57, 58].

Motion Tracking in Video

INTRODUCTION

Motion tracking in digital video aims at deriving the trajectory over time of moving objects or, in certain cases, the trajectory of the camera. Tracking should be distinguished from object detection, that aims at estimating an object's position and/or orientation in a certain image. However, detection and tracking are not totally unrelated. As a matter of fact, detection is involved in one of the two major approaches that one can adopt to devise a tracking algorithm [1]. According to this approach, object detection is performed on each frame of a video sequence and, subsequently, correspondences between objects detected in successive frames are sought. Thus, the trajectory of each object is established. According to the second approach, that essentially combines the detection and correspondence finding steps, the objects positions, and/or orientations in the next frame(s) are predicted, rather than detected, using information derived from the current (or previous) frames. The output of an object tracking algorithm depends on the application and the rep- resentation used to describe the object that is being tracked over time. Thus, this output can be, for example, the contour (silhouette) of the object, the 2D image coordinates of its center of mass, its 3D position in world coordinates, the posture of an articulated object (i.e., the set of joint angles that define the configuration in space of an articulated structure), and so forth.Object tracking has received considerable attention in the past few years mainly due to the wide range of its potential applications. One important application domainis advanced human–machine interfaces, where tracking, along with human motionanalysis and behavior understanding, plays a role complementary to speech recogni-tion and natural language understanding [2]. In this context, gesture recognition, body and face pose estimation, facial expression analysis and recognition are employed in an effort to enable machines to interact more cleverly with their users and also with their environment. In all these tasks, tracking constitutes an essential part of the overall process.

RIGID OBJECT TRACKING

➢ 2D Rigid Object Tracking

2D rigid object tracking tries to determine the motion of the projection of one or more rigid objects on the image plane. This motion is induced by the relative motion between the camera and the observed scene. A basic assumption behind 2D rigid motion tracking is that there is only one, rigid, relative motion between the camera and the observed scene [29]. This is the case of for example a moving car. This assumption rules out articulated objects, like a moving human body, or deformable objects like a piece of cloth. Methods for 2D rigid object tracking can be classified in different categories according to the tools that are used in tracking:

- region-based methods,
- contour-based methods,
- feature point-based methods,
- template-based methods.

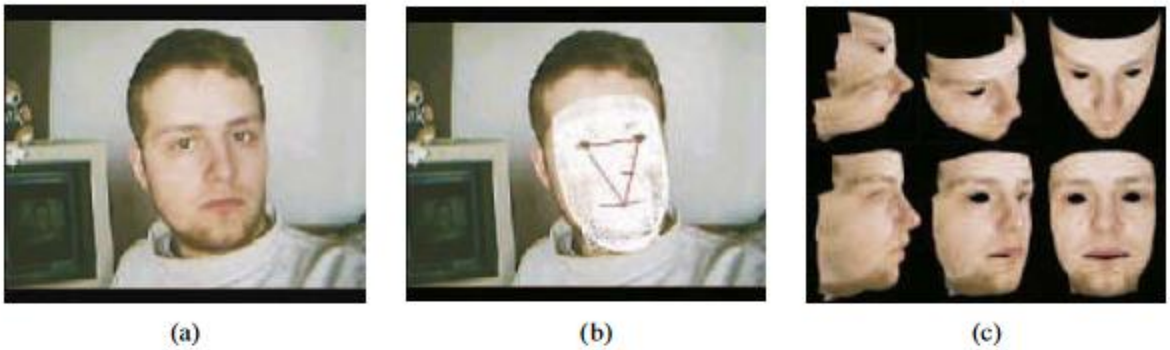
2D rigid object tracking methods sometimes constitute the basic building blocks for other categories of tracking algorithms. For example, an articulated object tracking algorithm may include a rigid object tracking module in order to track the rigid parts that make up the articulated structure. In the following, we will review the basic principles of the main categories of 2D rigid object tracking algorithms, describe the Bayesian framework frequently employed in object tracking algorithms and discuss the crucialtopic of occlusion handling.

- **3D Rigid Object Tracking**

3D rigid object tracking can be defined as the estimation of the position and orientation of a rigid object in 3D space from video data obtained from one or more video cameras. The location of a rigid object in the 3D space is determined by the position of its center of mass with respect to a world coordinate system, as well as the relative orientation of a coordinate system attached to its center of mass with respect to the world coordinate system. Thus, 3D rigid object tracking has to estimate a total of six parameters although in certain applications determining only the position (i.e., considering the object as a point mass) or only the orientation parameters might suffice. One of the most important applications of 3D rigid object tracking is 3D head tracking (often referred to as head pose estimation) which is being used as a preprocessing step or a building block in face recognition and verification, facial expression analysis, avatar animation, human- computer interaction, and model-based coding systems. Although the human head is not a rigid object, considering (and tracking) it as such (i.e., considering only the global head motion) is sufficient in a number of cases. Alternatively, head deformations can be taken into account in the tracking algorithm [142]. Other methods focus on tracking the facial features and expressions in two or three dimensions [143–148] but these fall outside the scope of this Section. 3D vehicle tracking is another application of 3D rigid object tracking [149]. Apart from its obvious importance as a stand-alone task, 3D rigid object tracking often constitutes a basic building block of 3D articulated object tracking methods, where one needs to estimate the position of rigid objects (links) that make up the articulated structure.

Certain methods (e.g., [150]) use 2D tracking techniques to derive the 2D image- plane motion of the object of interest and then a Kalman filter for deriving the 3D motion parameters. Another approach for head pose estimation [90] employs tracking in 2D of salient facial features (eye corners and nose). Projective invariance of the cross- ratios of the eye corners and anthropometric statistics are subsequently used to compute orientation relative to the camera plane. A similar approach in [86] locates robust facial features (eyebrows, eyes, nostrils, mouth, cheeks, and chin) and uses the symmetric properties of certain facial features and rules of projective geometry to determine the direction of gaze.

An approach for estimating 3D head orientation (i.e., the pan, tilt, and roll angles) in single-view video sequences is presented in [151]. Following initialization by a face detector, a variant of the tracking technique [103] (see Section 7.2.1.3) is used to track the face in the video sequence. The difference with [103] is that, for face tracking, the deformable surface model is used to approximate the image intensity surface of the entire face area. The generalized displacement vector (which, as already mentioned in Section 7.2.1.3, is involved in the equations that govern the deformation of the surface model) is used for face tracking. This vector is also used for head pose estimation. In more detail, the generalized displacement vector of a certain frame is provided, along with the head angles estimates from the previous frame, as input to three RBF interpolation networks that are trained off-line to estimate the pan, tilt, and roll angles. Training of the RBF networks (i.e., estimation of their parameters) is performed using example image sequences for which the head pose angles of each frame have been evaluated through the use of a magnetic head tracker.



In [153], the automatic adaptation of the CANDIDE face model [156] to video data is presented. Matching is performed by finding the main facial features (eyes, mouth) using deformable templates. Then, the model is fit using global and local adaptation. The global adaptation step estimates 3D eye and mouth center positions of the face model and uses them to perform scaling, rotation, and translation of the face model in the 3D world. The system presented in [152] also involves finding the main facial features and performing global and local adaptation. Geometrical considerations are used to perform the adap-tation. The method involves a geometric model that is more detailed than CANDIDE and does not restrict the initialization images to depict the subject in a certain facial expression (e.g., with mouth closed). The two methods presented earlier are applied to head and shoulders image sequences as they aim at the initialization of coding algorithms for videophone sequences.

ARTICULATED OBJECT TRACKING

A number of physical entities in real-world environments can only be represented using articulated structures, that is, structures composed of rigid parts (links) connected by joints, typically described through a tree-like kinematic hierarchy (Fig. 7.7). Living beings, such as humans or animals, exhibit such attributes. To be able to extract higher level information about the behavior of such entities (e.g., gesture recognition, understanding animal behavior, etc.), precise tracking of the corresponding articulated structures in 3D is necessary. Therefore, most of the articulated tracking algorithms attempt object tracking in a 3D space. Furthermore, even if the goal is to track an articulated object in 2D (i.e., in the frames of a video sequence), the methodology is similar to that of 3D articulated tracking algorithms. Moreover, many 2D articulated object tracking algorithms employ a 3D model of the articulated object. For all these reasons, most of this section covers 3D articulated object tracking, followed by a brief discussion on 2D articulated object tracking.

3D Articulated Object Tracking

3D articulated object tracking approaches can be model-free or model-based. In the former case, no model of the articulated structure is used. Instead, a bottom-up approach is used to combine image information extracted locally (edges, corners, etc.), to create coherent structures, such as the limbs of the human body. Obviously, this approach requires that the structures reconstructed are constantly visible in the images.

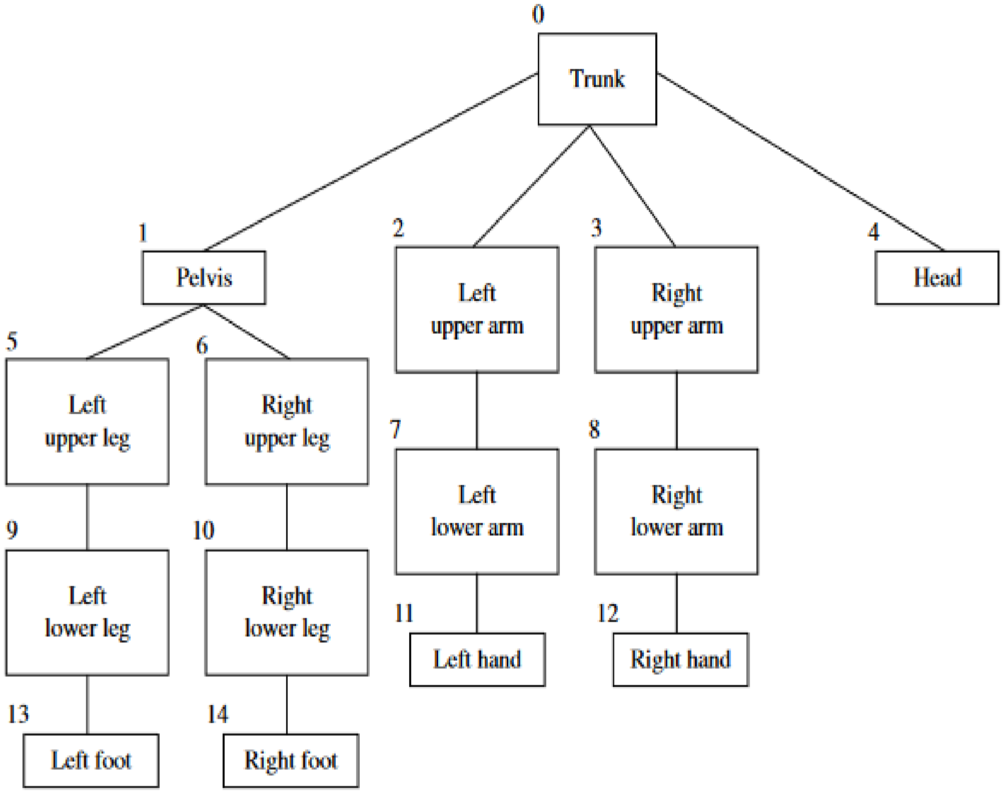


FIGURE 7.7 A tree-like hierarchical representation of the human body.

subset of model-free approaches for 3D human body (or human body parts) pose estimation and tracking are based on learning [159]. These methods often utilize the fact that the set of typical human poses is considerably smaller than the set of kinematically possible ones. Thus, these approaches try to devise methods for recovering the 3D body pose directly from image observations. A subcategory of learning-based approaches are the so-called example-based methods [160–163]. These methods store in an explicit way a large number of training images for which the corresponding 3D poses of the human body (or parts of it) are known. For given frame of an image sequence, stored images that are similar to this frame are retrieved and the pose in this frame is estimated by interpolating between the poses corresponding to the retrieved images. Some of these methods, for example [162, 163], are actually 3D pose estimation methods applicable to single images. However, one can easily incorporate them in a tracking framework, for example by applying them in each frame of a video sequence. In model-based approaches, a model of the articulated object is employed. The com-plexity of the model depends on the accuracy required in a specific application. The human body, for instance, is represented by rigid parts (resembling limbs) connected to each other at joints. Even such a minimal representation has around 30 degrees of freedom (DOF).

191CS72 - Image and Video Analytics

Unit - V Notes:

VIDEO COMPRESSION AND UNIFIED FRAMEWORK

Introduction to Video Compression

Video or visual communications require significant amounts of information transmission. Video compression as considered here involves the bit rate reduction of the digital video signal carrying visual information. Traditional video-based compression, like other information compression techniques, focuses on eliminating redundancy and unimportant elements of the source. The degree to which the encoder reduces the bit rate is called its coding efficiency, or equivalently its inverse is termed the compression ratio, that is,

coding efficiency (compression ratio)$^{-1}$ encoded bit rate/decoded bit rate.

Compression can be a lossless or lossy operation. Due to the immense volume of video information, lossy operations are a key element used in video compression algorithms. The loss of information or distortion measure is usually evaluated using the mean square error (MSE), mean absolute error (MAE) criteria, or peak signal-to-reconstruction noise (PSNR),

Video compression application requirements

A wide variety of digital video applications currently exist. They range from simple low-resolution and bandwidth applications (multimedia, Picture Phone) to very high-resolution and bandwidth (HDTV) demands. This section will present requirements of current and future digital video applications and the demands they place on the video compression system. To demonstrate the importance of video compression, the transmission of digital video television signals is presented. The bandwidth required by a digital television signal is approximately one-half the number of picture elements (pixels) displayed per second. The analog video monitor pixel size in the vertical dimension is the distance between scanning lines, and the horizontal dimension is the distance the scanning spot moves during half cycle of the highest video signal transmission frequency. The video signal bandwidth is given by Eq. 8.3,

$$\begin{aligned} BW &= (\text{cycles/frame})(FR) \\ &= (\text{cycles/line})(NL)(FR) \quad (8.3) \\ &= (0.5)(\text{aspect ratio})(FR)(NL)(RH)/0.84, \\ &= (0.8)(FR)(NL)(RH) \end{aligned}$$

where BW video signal system bandwidth, FR number of frames transmitted per second (fps), NL number of scanning lines per frame, and RH horizontal resolution (lines), proportional to pixel resolution.

The National Television Systems Committee (NTSC) picture aspect ratio is 4/3, the constant 0.5 is the ratio of the number of cycles to the number of lines, and the factor 0.84 is the fraction of the horizontal scanning interval that is devoted to signal transmission.

The NTSC transmission standard used for television broadcasts in the United States has the following parameter values: FR 29.97 fps, NL 525 lines, and RH 340 lines.

This yields an analog video system bandwidth BW of 4.2 MHz for the NTSC broadcast system. To transmit a color digital video signal, the digital pixel format must be defined. The digital color pixel is made of three components: one luminance (Y) component occupying 8 bits and two color difference components (U and V) each requiring 8 bits. The NTSC picture frame has 720 480 2 total luminance and color pixels. To transmit this information for an NTSC broadcast system at 29.97 fps, the following bandwidth is required:

Digital BW≅ 1/2 bit rate = 1/2(29.97 fps)* (24 bits/pixel)* (720 *480 *2 pixels/frame) =249MHz.

This represents an increase of approximately 59 times the required NTSC system bandwidth and about 41 times the full transmission channel bandwidth (6 MHz) for current NTSC signals. HDTV picture resolution requires up to three times more raw bandwidth than this example! (Two transmission channels totaling 12 MHz are allocated for terrestrial HDTV transmissions.) It is clear from this example that terrestrial television broadcast systems have to use digital transmission and digital video compression to achieve the overall bit rate reduction and image quality required for HDTV signals.

The example not only points out the significant system bandwidth requirements fordigital video information but also indirectly brings up the issue of digital video quality requirements. The trade-off between bit rate and quality or distortion is a fundamental issue facing the design of video compression systems. To this end, it is important to fully characterize an application's video communications requirements before designing or selecting an appropriate video compression system. Factors that should be considered in the design and selection of a video compression system include the following items:

■ Video characteristics—Video parameters such as the dynamic range, source statistics, pixel resolution, and noise content can affect the performance of the compression system.

■ Transmission requirements—Transmission bit rate requirements determine the power of the compression system. Very high-transmission bandwidth, storage capacity, or quality requirements may necessitate lossless compression. Conversely,extremely low bit rate requirements may dictate compression systems that trade-off image quality for a large compression ratio. Progressive transmission is a key issue for selection of the compression system. It is generally used when the transmission bandwidth exceeds the compressed video bandwidth. Progressive coding refers to a multiresolution, hierarchical, or subband encoding of the video information. It allows for transmission and reconstruction of each resolution independently from low to high resolution.

Channel errors affect system performance and the quality of the reconstructedvideo. Channel errors can affect the bit stream randomly or in burst fashion. The channel error characteristics can have different effects on different encoders and can range from local to global anomalies. In general, transmission error correction codes (ECCs) are used to mitigate the effect of channel errors, but awareness and knowledge of this issue is important.

■ Compression system characteristics and performance.The nature of video applications makes many demands on the video compression system. Interactive video applications such as videoconferencing demand that the video compression systems have symmetric capabilities. That is, each participant in the interactive video session must have the same video encoding and decoding capabilities, and that the system performance requirements must be met by both the encoder and decoder. On the other hand, television broadcast video has significantly greater performance requirements at the transmitter because it has the responsibility of providing real-time high-quality compressed video that meets the transmission channel capacity.

Digital video system implementation requirements can vary significantly. Desktop televideo conferencing can be implemented using software encoding and decoding or may require specialized hardware and transmission capabilities to provide high-quality performance. The characteristics of the application will dictate the suitability of the video compression algorithm for particular system implementations. The importance of the encoder and system implementation decision cannot be overstated; system architectures and performance capabilities are changing at a rapid pace, and the choice of the best solution requires careful analysis of the all possible system and encoder alternatives.

■ Rate-distortion requirements—The rate-distortion requirement is a basic consideration in the selection of the video encoder. The video encoder must be able to provide the bit rate(s) and video fidelity (or range of video fidelity) required by the application. Otherwise, any aspect of the system may not meet specifications. For example, if the bit rate specification is exceeded to support a lower MSE, a larger than expected transmission error rate may cause a catastrophic system failure.

■ Standards requirements—Video encoder compatibility with existing and future standards is an important consideration if the digital video system is required to inter-operate with existing and/or future systems. A good example is that of a desktop videoconferencing application supporting a number of legacy video compression standards. This requires support of the older video encoding standards on new equipment designed for a newer incompatible standard. Videoconferencing equipment not supporting the old standards would not be capable or as capable to work in environments supporting older standards. These factors are shown in Table 8.1 to demonstrate video compression system requirements for some common video communications applications. The video

Application	**Bit rate requirement**	**Distortion requirements**	**Transmission requirements**	**Computationl requirements**	**Standards requirements**
Network video on demand	1.5 Mbps, 10 Mbps	High, medium	Internet, 100 Mbps LAN	MPEG-1, MPEG-2/4	MPEG-1, MPEG-2/4, MPEG-7
Video phone	64 kbps	High distortion	ISDN p 64	H.261 encoder, H.261 decoder	H.261
Desktop multimedia video CDROM	1.5 Mbps	High Distortion to Medium	PC channel	MPEG-1 decoder	MPEG-1, MPEG-2, MPEG-7
Desktop LAN videoconference	10 Mbps	Medium distortion	Fast Ethernet 100 Mbps	Hardware encoders decoders	MPEG-2/4, H.261
Desktop WAN videoconference	1.5 Mbps	High distortion	Ethernet	Hardware encoders decoders	MPEG-1, MPEG-4, H.263

Desktop dial-up videoconference	64 kbps	Very High distortion	POTS and Internet	Software encoders decoders	MPEG-4, H.263
Digital satellite television	10 Mbps	Low distortion	Fixed service satellites FSS	MPEG-2 decoder	MPEG-2
HDTV	20 Mbps	Low distortion	12 MHz terrestrial link	MPEG-2/4 encoder decoder	MPEG-2, MPEG-4
HD DVD, DVD	36.5 Mbps 20 Mbps	Low distortion	PC channel	H.264, MPEG-2 decoder	H.264, MPEG-2

compression system designer as a minimum should consider these factors in making a determination about the choice of video encoding algorithms and technology to implement.

DIGITAL VIDEO SIGNALS AND FORMATS

Video compression techniques make use of signal models to be able to use the body of digital signal analysis/processing theory and techniques that have been developed over the past 50 or so years. The design of a video compression system as represented by the generalized model introduced in Section 8.1 requires knowledge of the signal characteristics and the digital processes that are used to create the digital video signal. It is also highly desirable to understand video display systems and the behavior of the HVS.

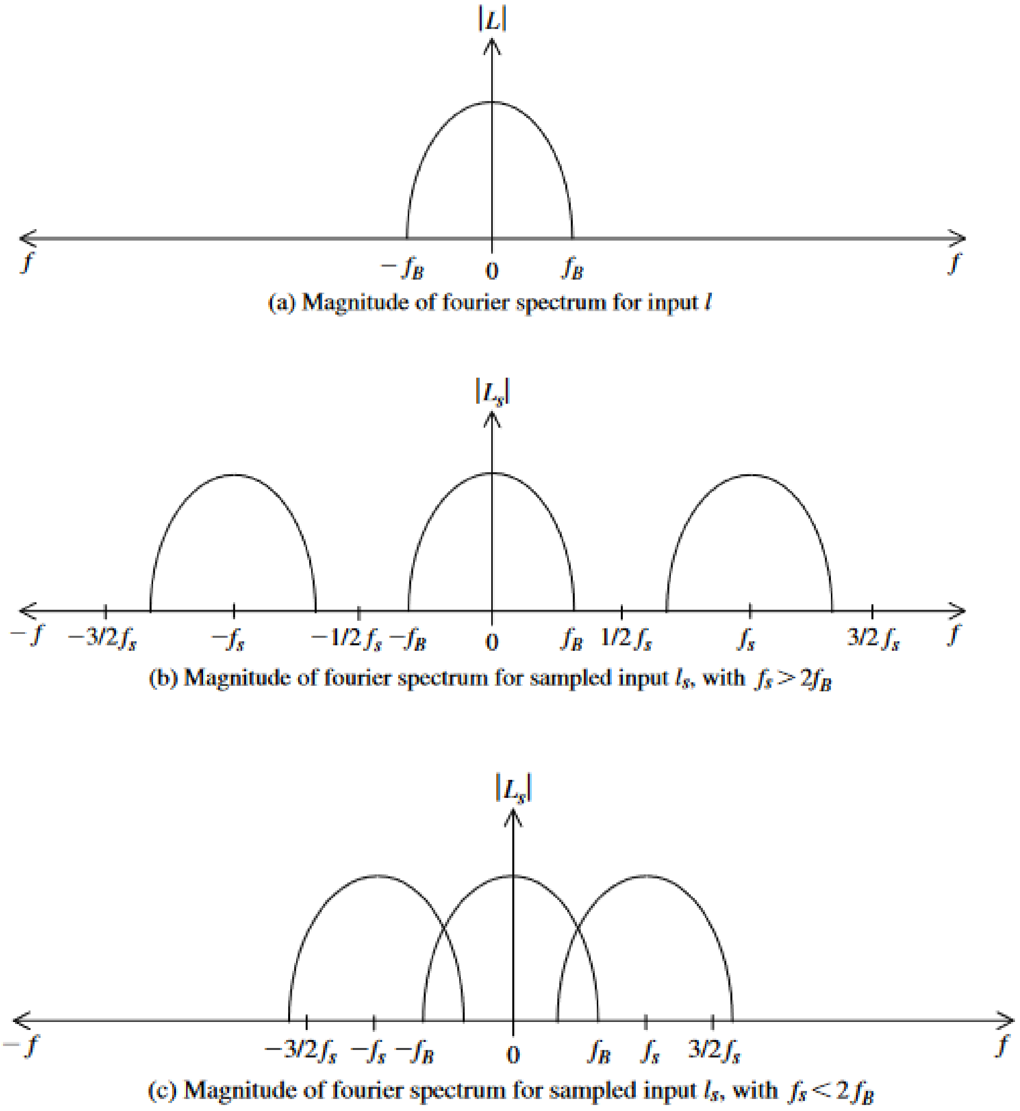
Sampling of Analog Video Signals

Digital video information is generated by sampling the intensity of the original continuous analog video signal I(x, y,t) in three dimensions. The spatial component of the video signal is sampled in the horizontal and vertical dimensions (x, y), and the temporal component is sampled in the time dimension (t). This generates a series of digital images or image sequence I(i,j,k). Video signals that contain colorized information are usually decomposed into three parameters (YCrCb,YUV, RGB, etc.) whose intensities are likewise sampled in three dimensions. The sampling process inherently quantizes the video signal due to the digital word precision used to represent the intensity values. Therefore, the original analog signal can never be reproduced exactly, but for all intents and purposes, a high-quality digital video representation can be reproduced with arbitrary closeness to the original analog video signal. The topic of video sampling and interpolation is discussed in Chapter 2.

An important result of sampling theory is the Nyquist Sampling Theorem. This theorem defines the conditions under which sampled analog signals can be “perfectly” reconstructed. If these conditions are not met, the resulting digital signal will contain aliased components, which introduce artifacts into the reconstruction. The Nyquist conditions are depicted graphically for the one-dimensional (1D) case in Fig. 8.2.

The 1D signal l is sampled at rate fs. It is band limited (as are all real world signals) in the frequency domain with an upper frequency bound of fB. According to the Nyquist Sampling Theorem, if a band-limited signal is sampled, the resulting Fourier spectrum is made up of the original signal spectrum |L| plus replicates of the original spectrum spaced at integer multiples of the sampling frequency fs. Figure 8.2(a) depicts the magnitude |L| of the Fourier spectrum forl. The magnitude of the Fourier spectrum |Ls| for the sampled signal ls is shown for two cases. Figure 8.2(b) presents the case where the original signal l can be reconstructed by recovering the central spectral island. Figure 8.2(c) shows the case where the Nyquist sampling criteria has not been met and spectral overlap occurs.

The spectral overlap is termed aliasing and occurs when $fs < 2fB$. When $fs > 2fB$, the original signal can be reconstructed by using a low-pass digital filter whose pass band is designed to recover |L|. These relationships provide a basic framework for the analysis and design of digital signal processing systems. 2D or spatial sampling is a simple extension of the 1D case. The Nyquist criteria have to be obeyed in both dimensions, that is, the sampling rate in the horizontal direction must be two times greater than the upper frequency bound in the horizontal direction, and the sampling rate in the vertical direction must be two times greater than the upper frequency bound in the vertical direction. In practice, spatial sampling grids are square so that an equal number of samples per unit length in each direction are collected. Charge coupled devices (CCDs) are typically used to spatially sample analog imagery and video 8.3 Digital Video Signals and Formats 243 The sampling grid spacing of these devices is more than sufficient to meet the Nyquist criteria for most



(a) Magnitude of fourier spectrum for input l

(b) Magnitude of fourier spectrum for sampled input l_s, with $f_s > 2f_B$

(c) Magnitude of fourier spectrum for sampled input l_s, with $f_s < 2f_B$

resolution and application requirements. The electrical characteristics of CCDs have a greater effect on the image or video quality than its sampling grid size.

Digital Video Formats

Sampling is the process used to create the image sequences used for video and digitalvideo applications. Spatial sampling and quantization of a natural video signal digitizes the image plane into a 2D set of digital pixels that define a digital image. Temporal sampling of a natural video signal creates a sequence image frames typically used for motion pictures and television. The combination of spatial and temporal sampling creates a sequence of digital images termed digital video. As described earlier, the digital video signal intensity is defined as I (i, j, k), where 0

i

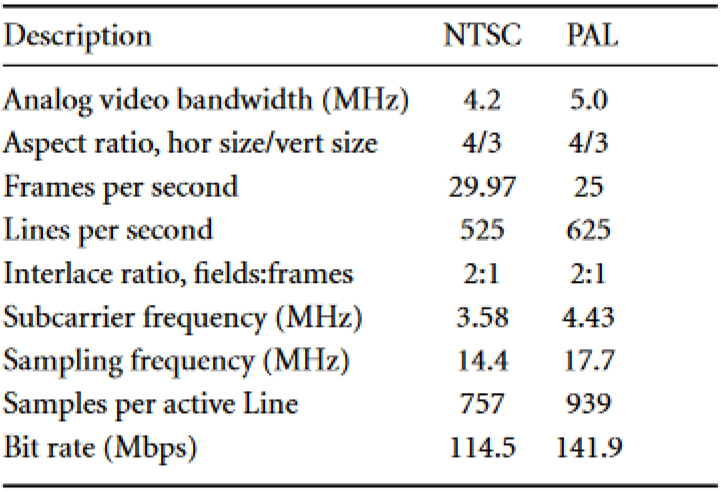
M , 0

j

N are the horizontal and vertical spatial coordinates, and 0

k is the temporal coordinate. The standard digital video formats introduced here are used in the broadcast for both analog and digital television, as well as computer video applications. Composite television signal digital broadcasting formats are included here due to their use in video compression standards, digital broadcasting, and standards format conversion applications. Knowledge of these digital video formats provides background for understanding the international video compression standards developed by the ITU and the ISO. These standards contain specific recommendations for use of the digital video formats described here.

Digital composite television parameters



Description	NTSC	PAL
Analog video bandwidth (MHz)	4.2	5.0
Aspect ratio, hor size/vert size	4/3	4/3
Frames per second	29.97	25
Lines per second	525	625
Interlace ratio, fields:frames	2:1	2:1
Subcarrier frequency (MHz)	3.58	4.43
Sampling frequency (MHz)	14.4	17.7
Samples per active Line	757	939
Bit rate (Mbps)	114.5	141.9

VIDEO COMPRESSION TECHNIQUES

Video compression systems generally comprise two modes that reduce information redundancy in the spatial and the temporal domains. Spatial compression and quantization operates on a single image block, making use of the local image characteristics to reduce the bit rate. The spatial encoder also includes a VLC inserted after the quantization stage. The VLC stage generates a lossless encoding of the quantized image block. Temporal domain compression makes use of optical flow models (generally in the form of block-matching motion estimation methods) to identify and mitigate temporal redundancy.

This section presents an overview of some widely accepted encoding techniques used in video compression systems. Entropy Encoders are lossless encoders that are used in the VLC stage of a video compression system. They are best used for information sources that are memoryless (sources in which each value is independently generated), and try to minimize the bit rate by assigning variable length codes for the input values according to the input probability density function (pdf). Predictive Coders are suited to information sources that have memory, that is, a source in which each value has a statistical dependency on some number of previous and/or adjacent values. Predictive coders can produce a new source pdf with significantly less statistical variation and entropy than the original. The transformed source can then be fed to a VLC to reduce the bit rate. Entropy and predictive coding are good examples for presenting the basic concepts of statistical coding theory. Block transformations are the major technique for representing spatial information in a format that is highly conducive to quantization and VLC encoding. Block transforms can provide a coding gain by packing most of the block energy into a fewer number of coefficients. The quantization stage of the video encoder is the central factor in determining the rate-distortion characteristics of a video compression system. It quantizes the block TCOEFF according to the bit rate and distortion specifications. MC takes advantage of the significant information redundancy in the temporal domain by creating current frame predictions based on block matching motion estimates between the current and previous image frames. MC generally achieves a significant increase in the video coding efficiency over pure spatial encoding.

Entropy and Predictive Coding

Entropy coding is an excellent starting point in the discussion of coding techniques because it makes use of many of the basic concepts introduced in the discipline of Information Theory or Statistical Communications Theory [11]. The discussion of VLC and predictive coders requires the use of information source models to lay the statistical foundation for the development of this class of encoder. An information source can be viewed as a process that generates a sequence of symbols from a finite alphabet. Video sources are generated from a sequence of image blocks that are generated from a “pixel” alphabet. The number of possible pixels that can be generated is 2n, when n is the number of bits per pixel. The order in which the image symbols are generated depends on how the image block is arranged or scanned into a sequence of symbols. Spatial encoders transform the statistical nature of the original image so that the resulting coefficient matrix can be scanned in a manner such that the resulting source or sequence of symbols contains significantly less information content. Two useful information sources are used in modeling video encoders: the Discrete Memoryless Source (DMS) and Markov sources. VLC coding is based on the DMS model, and the predictive coders are based on the Markov source models. The DMS is simply a source in which each symbol is generated independently. The symbols are statistically independent and the source is completely defined by its symbols/events and the set of probabilities for the occurrence for each symbol, that is, E {e1,e2,...,en} and the set {p(e1),p(e2),...,p(en)}, where n is the number of symbols in the alphabet. It is useful to introduce the concept of entropy at this point. Entropy is defined as the average information content of the information source. The information content of a single event or symbol is defined as

$$I(e_i) = \log \frac{1}{p(e_i)}.$$

The base of the logarithm is determined by the number of states used to represent the information source. Digital information sources use base 2 to define the information content using the number of bits per symbol or bit rate. The entropy of a digital source is further defined as the average information content of the source, that is,

$$H(E) = \sum_{i=1}^{n} p(e_i) \log_2 \frac{1}{p(e_i)} = -\sum_{i=1}^{n} p(e_i) \log_2 p(e_i) \text{bits/symbol.}$$

This relationship suggests that the average number of bits per symbol required to represent the information content of the source is the entropy. The Noiseless Source Coding Theorem states that a source can be encoded with an average number of bits per source symbol that is arbitrarily close to the source entropy. So-called entropy encoders seek to find codes that perform close to the entropy of the source. Huffman and Arithmetic encoders are examples of entropy encoders.

Block Transform Coding—The DCT

Block transform coding is widely used in image and video compression systems. The transforms used in video encoders are unitary, which means that the transform operation has an inverse operation that uniquely reconstructs the original input. The DCT successively operates on 8 8 image blocks and is used in the H.261, H.262, H.263, MPEG-1, MPEG-2, and MPEG-4 standards. Block transforms make use of the high degree of correlation between adjacent image pixels to provide energy compaction or coding gain in the transformed domain. The block transform coding gain G TC is defined as the logarithmic ratio of the arithmetic and geometric means of the transformed block variances (VAR), that is,

$$G_{\mathrm{TC}} = 10 \log_{10} \left[\frac{\frac{1}{N} \sum_{i=0}^{N-1} \sigma_i^2}{\left(\prod_{i=0}^{N-1} \sigma_i^2 \right)^{1/N}} \right],$$

Where the transformed image block is divided into N subbands, and o^{-2}i is the variance of each subband i, for 0 ≤ i ≤ N - 1. GTC also measures the gain of block transform coding over pulse code modulation coding. The coding gain generated by a block transform is realized by packing most the original signal energy content into a small number of TCOEFF. This results in a lossless representation of the original signal that is more suitable for quantization. That is, there may be many TCOEFF containing little or no energy that can be completely eliminated. Spatial transforms should also be orthonormal, that is, generate uncorrelated coefficients, so that simple scalar quantization can be used to quantize the coefficients independently.

The DCT is the most widely used block transform for digital image and video encoding. It is an orthonormal transform and has been found to perform close to the KLT [14] for first-order Markov sources. The DCT is defined on an 8 * 8 array of pixels,

$$F(u,v) = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right)$$

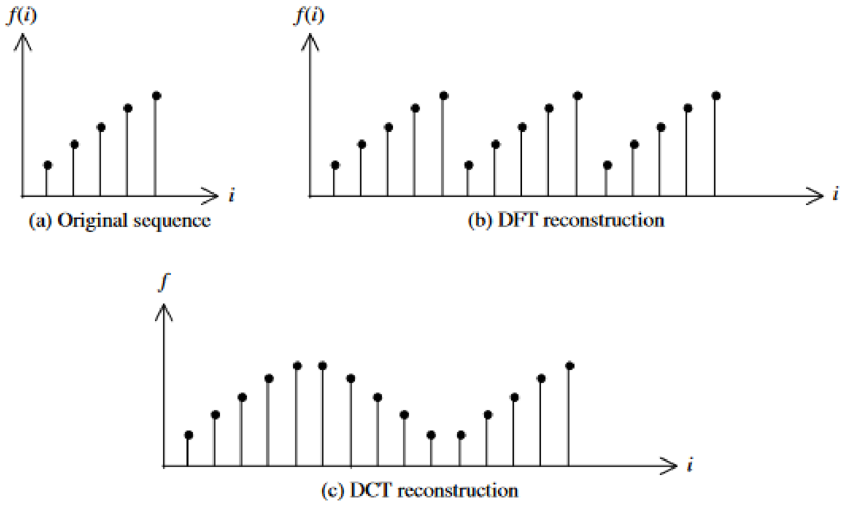
and the inverse IDCT is defined as,

$$f(i,j) = C_u C_v \sum_{u=0}^{7} \sum_{v=0}^{7} F(u,v) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right),$$

where

$$C_u = \frac{1}{\sqrt{2}} \text{ for } u = 0, \; C_u = 1 \text{ otherwise}$$
$$C_v = \frac{1}{\sqrt{2}} \text{ for } v = 0, \; C_v = 1 \text{ otherwise}$$

where i and j are the horizontal and vertical indices of the 8*8 spatial array, and u and v are the horizontal and vertical indices of the 8 8 coefficient array. The DCT is the chosen method for image transforms for a couple of important reasons. The DCT has fast O(n log n) implementations using real calculations. It is even simpler to compute than the DFT because it does not require the use of complex numbers.



(a) Original sequence

(b) DFT reconstruction

(c) DCT reconstruction

Quantization

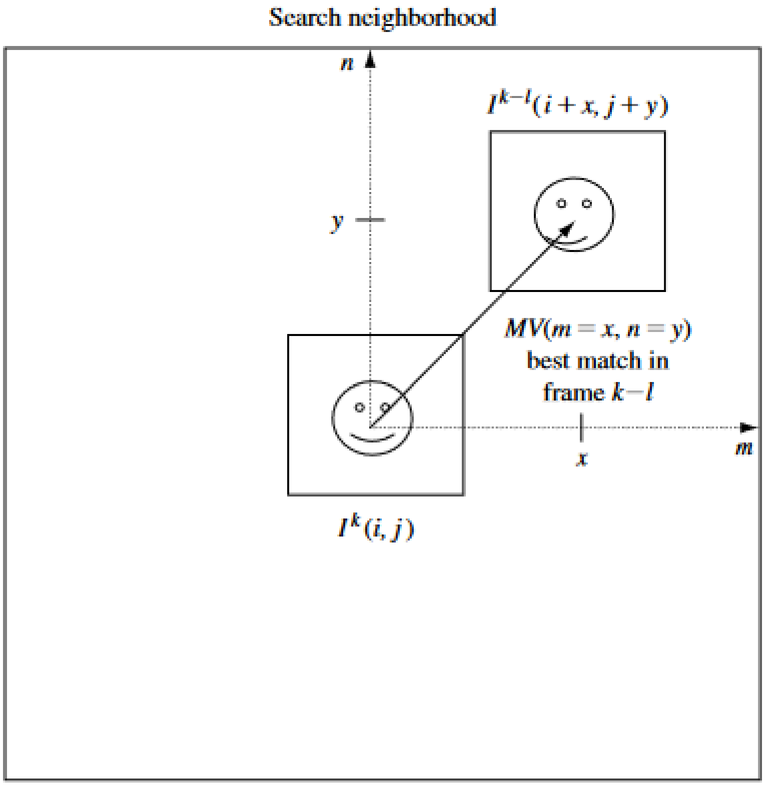
The quantization stage of the video encoder creates a lossy representation of the input. The input as discussed earlier should be conditioned with a particular method of quantization in mind. And vice versa, the quantizer should be well matched to the characteristicsof the input to meet or exceed the rate-distortion performance requirements. As always is the case, the quantizer has an effect on system performance that must be taken under consideration. Simple scalar versus VQ implementations can have significant system performance implications.

MC and Estimation

MC [17] is a technique created in the 1960s, which is used to increase the efficiency of video encoders. Motion compensated video encoders are implemented in three stages.The first stage estimates objective motion (motion estimation) between the previously reconstructed frame and the current frame. The second stage creates the current frame prediction (MC) using the motion estimates and the previously reconstructed frame. The final stage differentially encodes the prediction and the actual current frame as the prediction error. Therefore, the receiver reconstructs the current image only using the VLC encoded motion estimates and the spatially and VLC encoded prediction error.

$$\text{Best Match}_{\text{MSE}} = \min_{m,n} \frac{1}{N^2} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[I^k(i,j) - I^{k-1}(i+m, j+n) \right]^2, \qquad (8.12)$$

where k is the frame index, l is the temporal displacement in frames, M is the number of pixels in the horizontal direction, N is the number of pixels in the vertical direction of the image block, i and j are the pixel indices within the image block, and m and n are the indices of the search neighborhood in the horizontal and vertical directions. Therefore, the best match motion vector estimate MV (m = x, n = y) is the pixel displacement between the block I^k (i, j) in frame k and the best matched block I $^{k-1}$ (i +x, j + y) in the displaced frame k – l.



TRANSFORM CODING: INTRODUCTION TO THE VIDEO ENCODING STANDARDS

The major internationally recognized video compression standards have been developed by the ISO, the International Electrotechnical Commission (IEC), and the ITU standards organizations. The Moving Pictures Experts Group (MPEG) is a working group operating within ISO and

IEC. Since starting its activity in 1988, MPEG has produced ISO/IEC 11172 (MPEG-1, 1992), ISO/IEC 13818 (MPEG-2, 1994), ISO/IEC 14496 (MPEG-4,1999), ISO/IEC 15938 (MPEG-7, 2001), and ISO/IEC 21000 (MPEG-21, 2002). The ITU adopted the original CCITT Recommendation H.261: “Video Codec for Audio Visual Services at p 64 kbps,” in 1990, followed by the ITU-T SG 15 WP 15/1 Draft Recommendation H.262 (Infrastructure of audiovisual services—Coding of moving video) 1995, ITU-T SG 15 WP 15/1 Draft Recommendation H.263 (Video coding for low bit rate communications) 1995, and lastly the latest ITU Recommendation H.264: “Advanced Video Coding (AVC),” in 2002. H.264 and MPEG-4 Part 10 are equivalent video coding specifications as are H.262 and MPEG-2.

The MPEG-1 specification was motivated by T1 network transmission speeds, the CD-ROM, and the early multimedia capabilities of the desktop computer. It is intended for video coding up to the rate of 1.5 Mbps and is composed of five sections: System Configurations, Video Coding, Audio Coding, Compliance Testing, and Software for MPEG-1 Coding. The standard does not specify the actual video coding process, but only the syntax and semantics of the bit stream, and the video generation at the receiver. It does not accommodate interlaced video and only supports CIF quality format at 25 or 30 fps.

Transform Coding Standard Example: The H.261 Video Encoder

A brief description of the H261 video coding standard is offered in this section as an introduction to the techniques used in the transform-based video coding standards. The H.261 recommendation [4] is targeted at the videophone and videoconferencing application market running on connection-based ISDN at p 64 kbps, p 1,...,30. It explicitly defines the encoded bit-stream syntax and decoder, while leaving the encoder design to be compatible with the decoder specification. The video encoder is required to carry a delay of less than 150 ms so that it can operate in real-time bidirectional video-conferencing applications. H.261 is part of a group of related ITU Recommendations that define Visual Telephony Systems. This group includes as follows:

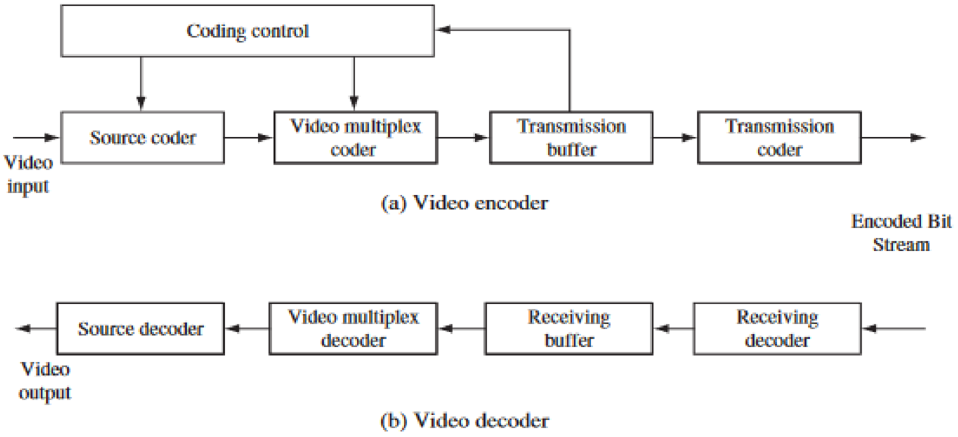
H.221 – Defines the frame structure for an audiovisual channel supporting 64–1920 kbps.

H.230 – Defines frame control signals for audiovisual systems.

H.242 – Defines audiovisual communications protocol for channels supporting up to 2 Mbps.

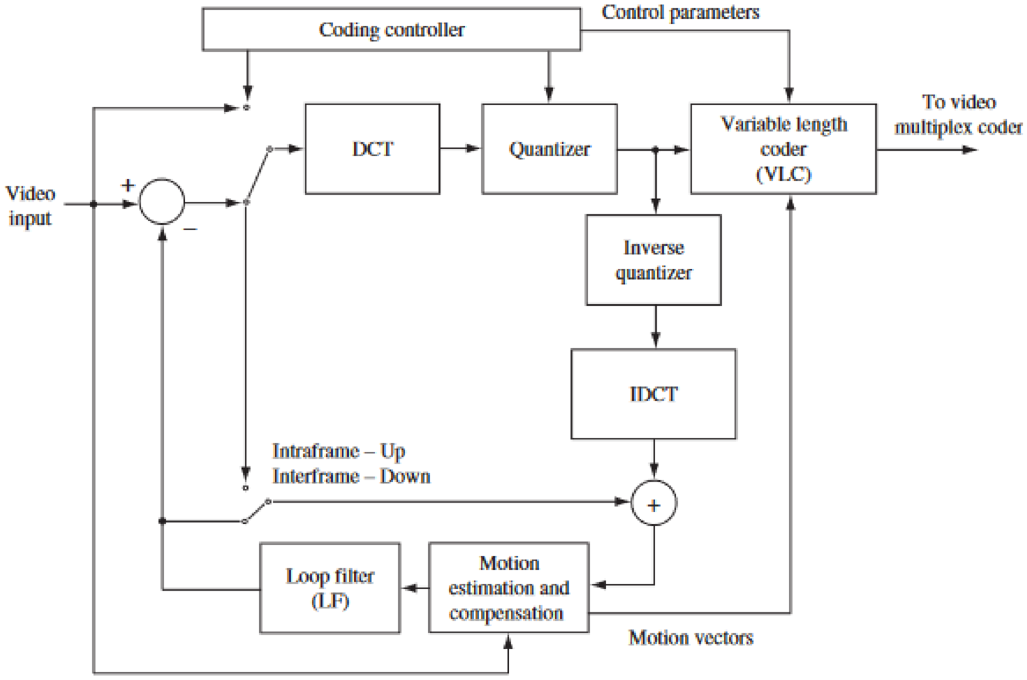
H.261 – Defines the video encoder/decoder for audiovisual services at p 64 kbps.

H.320 – Defines narrow-band audiovisual terminal equipment for p 64 kbps transmission.



(a) Video encoder

(b) Video decoder



(c) H.261 Source encoder implementation

MPEG-1 Target Applications and Requirements

The MPEG standard is a generic standard, which means that it is not limited to a particular application. A variety of digital storage media applications of MPEG-1 have been proposed based on the assumptions that acceptable video and audio quality can be obtained for a total bandwidth of about 1.5 Mbps. Typical storage media for these applications include VCD, digital audio tape (DAT), Winchester-type computer disks, and writable optical disks. The target applications are asymmetric applications where the compression process is performed once and the decompression process is required often. Examples of the asymmetric applications include VCD, video on demand (VOD), and video games.In these asymmetric applications, the encoding delay is not a concern. The encoders are needed only in small quantities while the decoders are needed in large volumes. Thus, the encoder complexity is not a concern while the decoder complexity needs to be low to result in low-cost decoders.

The requirements for compressed video in digital storage media mandate several important features of the MPEG-1 compression algorithm. The important features include normal playback, frame-based random access and editing of video, reverse playback, fast forward/reverse play, encoding high-resolution still frames, robustness to uncorrectable errors, etc. The applications also require MPEG-1 to support flexible picture sizes and frame rates. Another requirement is that the encoding process can be performed in reasonable speed using existing hardware technologies and the decoder can be implemented in low cost.

Since the MPEG-1 video coding algorithm was developed based on H.261, in the following sections, we will focus only on those parts that are different from H.261.

Motion-Compensated Prediction with Half-Pixel Accuracy

The motion estimation in H.261 is restricted to only integer-pixel accuracy. However, a moving object often moves to a position that is not on the pixel grid but between the pixels. MPEG-1 allows half-pixel accuracy motion vectors. By estimating the displacement at a finer resolution, we can expect improved prediction and, thus, better performance than motion estimation with integer-pixel accuracy. As shown in Fig. 9.2, since there is no pixel value at the half-pixel locations, interpolation is required to produce the pixel values at the half-pixel positions. Bilinear interpolation is used in MPEG-1 for its simplicity. As in H.261, the motion estimation is performed only on luminance blocks. The resulting motion vector is scaled by 2 and applied to the chrominance blocks. Motion vectors are differentially encoded with respect to the motion vector in the preceding adjacent macroblock. The reason is that the motion vectors of adjacent regions are highly correlated, as it is quite common to have relatively uniform motion over areas of the picture.

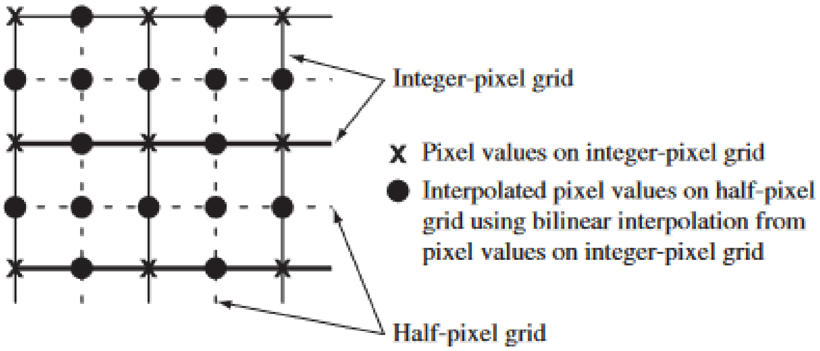

Integer-pixel grid
X Pixel values on integer-pixel grid
● Interpolated pixel values on half-pixel grid using bilinear interpolation from pixel values on integer-pixel grid
Half-pixel grid

FIGURE 9.2

Half-pixel motion estimation.

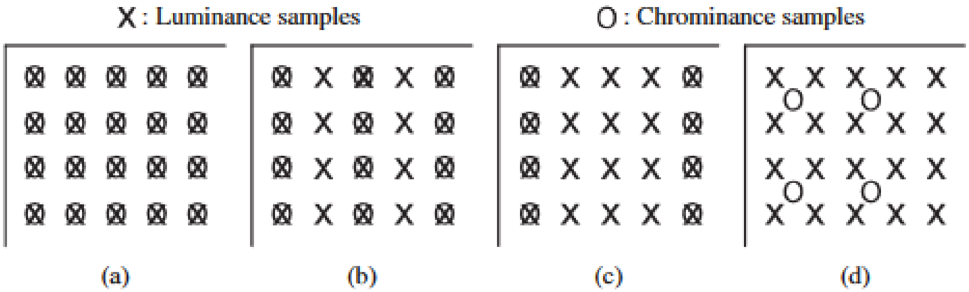


FIGURE 9.3

Luminance and chrominance samples in (a) 4:4:4 format (b) 4:2:2 format (c) 4:1:1 format (d) 4:2:0 format.

MPEG-2 VIDEO CODING Standard

Target Applications and Requirements

MPEG-2 is primarily targeted at coding high-quality video at 4–15 Mbps for VOD, standard definition (SD) and high-definition (HD) digital television broadcasting, and digital storage media such as digital versatile disc (DVD). The requirements from MPEG-2 applications mandate several important features of the compression algorithm. Regarding picture quality, MPEG-2 needs to be able to provide good NTSC quality video at a bit rate of about 4–6 Mbps and transparent NTSC quality video at a bit rate of about 8–10 Mbps. It also needs to provide the capability of random access and quick channel switching by means of I-pictures in GOPs. Low-delay mode is specified for delay-sensitive visual communications applications. MPEG-2 has scalable coding modes to support multiple grades of video quality, spatial resolutions, and frame rates for various applications. Error resilience options include intramotion vector, data partitioning, and scalable coding. Compatibility with the existing MPEG-1 video standard is another prominent feature provided by MPEG-2. For example, MPEG-2 decoders should be able to decode MPEG-1 bit streams. If scalable coding is used, the base layer of MPEG-2 signals can be decoded by a MPEG-1 decoder. Finally, it should allow reasonable complexity encoders and low-cost decoders be built with mature technology. Since MPEG-2 video is based heavily on MPEG-1, in the following sections, we will focus only on those features which are different from MPEG-1 video.

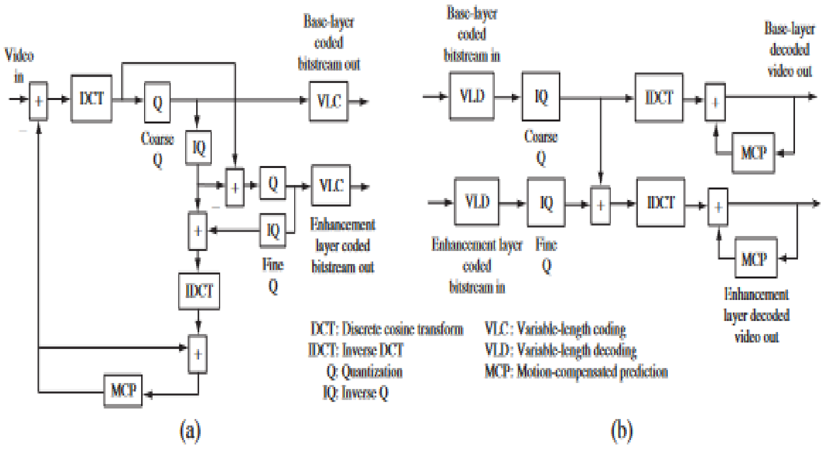
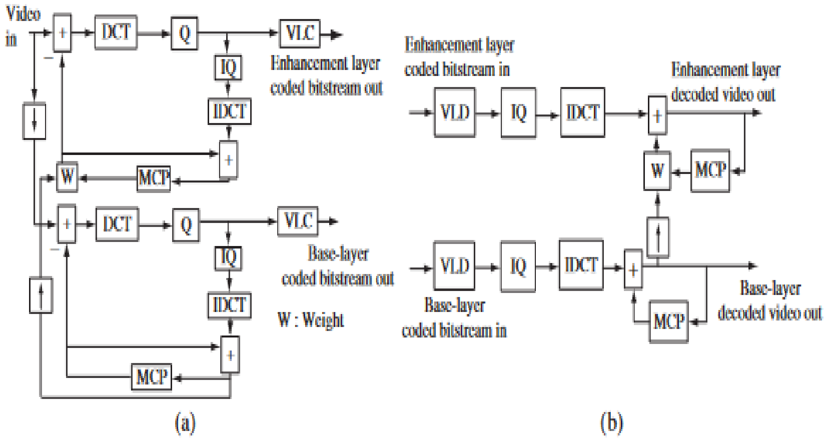

Video in
DCT
Q
Coarse Q
IQ
Base-layer coded bitstream out
VLC
Q
VLC
IQ
Fine Q
Enhancement layer coded bitstream out
IDCT
MCP
(a)
Base-layer coded bitstream in
VLD
IQ
Coarse Q
IDCT
MCP
Base-layer decoded video out
VLD
IQ
Enhancement layer coded bitstream in
Fine Q
IDCT
MCP
Enhancement layer decoded video out
DCT: Discrete cosine transform
IDCT: Inverse DCT
Q: Quantization
IQ: Inverse Q
VLC: Variable-length coding
VLD: Variable-length decoding
MCP: Motion-compensated prediction
(b)

FIGURE 9.16

(a) SNR scalable encoder. (b) SNR scalable decoder.


Video in
DCT
Q
VLC
IQ
Enhancement layer coded bitstream out
IDCT
W
MCP
DCT
Q
VLC
IQ
Base-layer coded bitstream out
IDCT
W : Weight
MCP
(a)
Enhancement layer coded bitstream in
VLD
IQ
IDCT
Enhancement layer decoded video out
W
MCP
VLD
IQ
IDCT
Base-layer coded bitstream in
MCP
Base-layer decoded video out
(b)

A UNIFIED FRAMEWORK FOR INDEXING, SUMMARIZATION, BROWSING, AND RETRIEVAL

The above two subsections described video browsing (using ToC generation and highlights extraction) and retrieval techniques separately. In this section, we integrate them into a unified framework to enable a user to go "back and forth" between browsing and retrieval. Going from the Index to the ToC or the Highlights, a user can get the context where the indexed entity is located. Going from the ToC or the Highlights to the Index, a user can pinpoint specific queries. Figure 15.12 illustrates the unified framework.

An essential part of the unified framework is composed of the weighted links. The links can be established between Index entities and scenes, groups, shots, and key frames

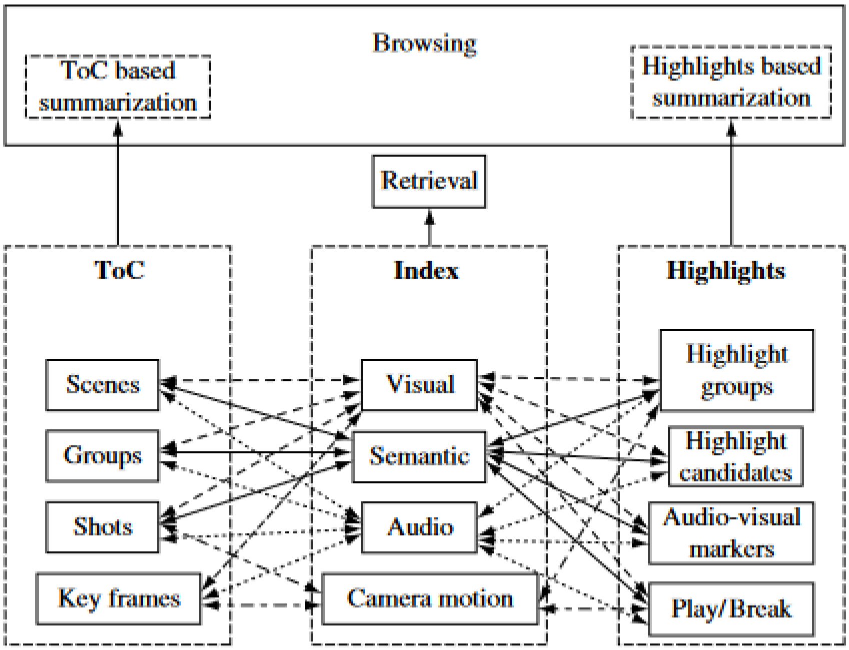


FIGURE 15.12

A unified framework.

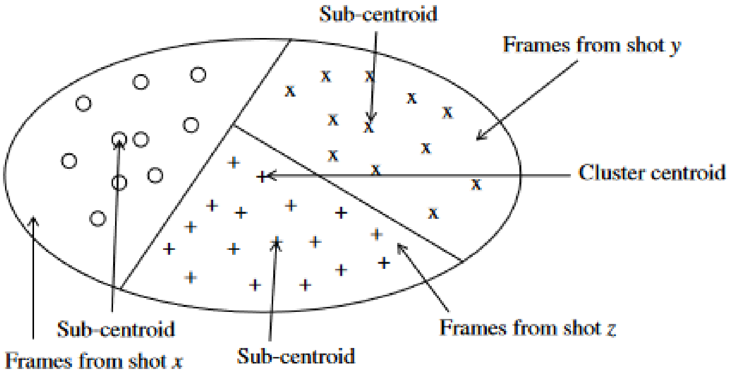


FIGURE 15.13

Subclusters.

in the ToC structure for scripted content and between Index entities and finer-resolution highlights, highlight candidates, audiovisual markers and plays/breaks. For scripted content, as a first step, in this article, we focus our attention on the links between Index entities and shots. Shots are the building blocks of the ToC. Other links are generalizable from the shot link. To link shots and the visual Index, we propose the following techniques. As we mentioned before, a cluster may contain frames from multiple shots. The frames from a particular shot form a subcluster. This subcluster's centroid is denoted as "csub" and the centroid of the whole cluster is denoted as "c." This is illustrated in Fig. 15.13. Here, c is a representative of the whole cluster (and thus the visual Index) and csub is a representative of the frames from a given shot in this cluster.