IMAGE SEQUENCE PROCESSING

Sean Borman, Robert Stevenson
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA

October 14, 2002

Contents

List of Tables

List of Figures

Chapter 1

IMAGE SEQUENCE PROCESSING

Keywords: image sequence, video, motion estimation, optical flow, compression, coding, transform coding, hybrid coding, MPEG, restoration, super-resolution

1.1 INTRODUCTION

Image sequence processing refers to methods for the digital processing of a set of images with related content. Often the image sequence results from the observation of a time-varying scene (video). Since the range of techniques which fall into the category of image sequence processing is extremely broad, the topic is approached in three main areas:

1. Fundamental video concepts.

   The image formation process, three-dimensional motion, projected motion, analog and digital video, sampling and quantization are discussed.

2. Motion estimation.

   The problem of motion estimation is central to image sequence processing. The characteristics of, and difficulties associated with the motion estimation problem are discussed, followed by a detailed description of three popular motion estimation frameworks.

3. Applications.

   Digital video compression, an application of image sequence processing techniques, has made practical numerous new and emerging video and entertainment technologies. These compression technologies rely heavily on motion estimation and prediction, in addition to more traditional coding techniques.

   The discussion on compression is followed by an introduction to an emerging area in image sequence restoration – multi-frame super-resolution restoration. These techniques extend the classical single-image restoration theory by integrating information from multiple frames, taking advantage of the motion occurring in the image sequence.

   The chapter is concluded with a brief overview of other important applications.

1.2 VIDEO FUNDAMENTALS

1.2.1 Introduction

In this section an overview of the fundamentals of the image formation process is presented and the basic characteristics of analog and digital video are described. The typical imaging scenario where a camera observes an illuminated three-dimensional scene is assumed.

1.2.2 Image Formation

An *imaging system* records the time- and space-varying light intensity information reflected and emitted from objects in a three dimensional scene.

The imaging system

Recording of an image sequence is achieved using a *imaging system* which is composed of:

1. An optical system

   The purpose of the optical system is to form a two-dimensional image of the electromagnetic radiation emitted and reflected from objects in the three-dimensional scene. The optical system typically consists of a series of *lenses* which serve to focus the illumination on a two-dimensional surface, called the *focal plane*, where the characteristics of the incident radiation may be recorded.

2. A recording system

   The recording system is designed to measure and record the characteristics of the radiation incident at the focal plane. Usually the illumination incident at the focal plane varies as a function of location and time. Furthermore, the incident electromagnetic radiation typically consists of a range of wavelengths, so the incident energy may be measured in one or more spectral wavelength ranges or *bands*. hyperspectral

Image formation by geometric projection

Though the image formation properties of the optical system are typically complicated by the presence of various distortions or *aberrations*, it is nevertheless useful to model the idealized geometric properties of the optical system. The idealized geometric properties of the camera abstract away the complex process of optical image formation and replace it with purely geometric projection from locations in the 3-D scene to 2-D locations in the focal plane. Consider a point $\mathbf{X}(t) = [X(t), Y(t), Z(t)]^T$ in 3-D space. At time t, the optical system projects the 3-D point $\mathbf{X}(t)$ onto the focal plane at position $\mathbf{x}(t) = [x(t), y(t)]^T$. The most commonly used models of the image projection characteristics are the *perspective projection* and the simpler *orthographic projection*.

- Perspective projection

  Perspective projection models the geometric projection characteristics of an idealized pinhole camera as shown in Figure 1. Assuming that the origins of the world and image

plane coordinate systems are coincident, $x = fX/(f - Z)$ and $y = fY/(f - Z)$. In these expressions f is the focal length, which is the distance along the optical axis from the image plane to the center of projection. See [1, 2, 3] for discussions on projective geometry.

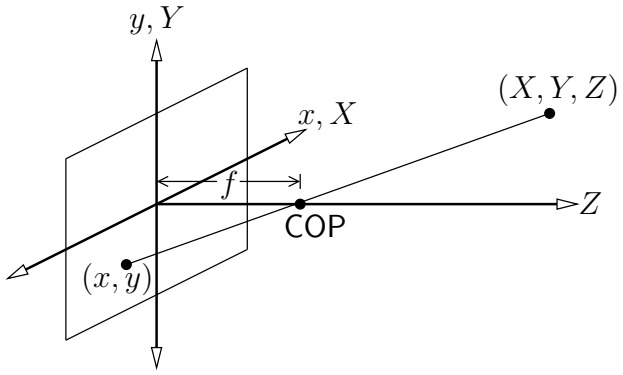

Figure 1: Perspective projection model with center of projection (COP) and focal length f.

- Orthographic projection

  Orthographic projection assumes parallel projection from the 3-D scene to the image plane as shown in Figure 2. Assuming coincident origins of the world and image planes implies that $x = X$ and $y = Y$.

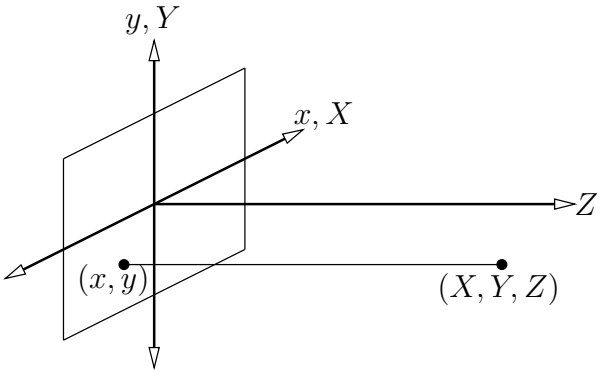

Figure 2: Orthographic projection model.

1.2.3 Sensing Radiation Incident at the Focal Plane

Existing approaches for sensing and recording the characteristics of the spatial, temporal and wavelength variation of the radiation incident at the focal plane may be categorized into two classes, chemical and electronic:

Motion picture film

Motion picture film represents the common chemical-based approach to recording the time-varying image projected at the focal plane. Motion picture film includes an emulsion which consists of a suspension of tiny, light-sensitive particles or *grains*. When exposed to light, the grains are altered so that a latent image is recorded in the material. A chemical development process fixes the latent image so that it may be viewed under normal illumination. A sequence of such photographic images, recorded at uniformly-spaced time instants (typically 24 frames per second) constitutes a motion picture sequence and forms an approximate representation of the time-varying light intensity pattern incident at the camera focal plane. Motion picture film is also capable of recording information in multiple wavelength bands for color reproduction.

Electronic sensors

Increasingly, electronic means are employed to record the time-varying focal plane image. Scanning video tubes and, more recently, focal plane arrays such as the charge-coupled device (CCD), charge-injection device (CID) and complementary metal oxide semiconductor (CMOS) sensors have been used for recording incident illumination. These devices utilize engineered semiconductor materials and devices tailored specifically for the purpose of generating and collecting charges produced by the interaction of incident photons and the material. Focal plane array sensors consist of a two-dimensional array of sensing elements called *pixels* which record the incident radiation. The spectral response of solid state sensors is determined by the engineered material characteristics. Color reproduction is typically achieved with the use of a color filter array which ensures that individual pixels measure wavelengths in specific spectral bands.

Noise

The recorded image intensity information is always affected by noise in the detection and recording system. In imaging systems utilizing focal plane sensor arrays, the electronic properties of these devices are a source of noise. For photochemical imaging processes, so-called film grain noise is present. In situations of extremely low illumination, or in certain medical imaging applications, the number of incident photons is so small that a Poisson model for the photon count is necessary. This results in what is referred to as *photon counting noise*. Other sources of noise include quantization, speckle and atmospheric effects. Reviews of image noise models may be found in [4, 5].

1.2.4 Representing the Time-Varying Focal Plane Image

The electromagnetic radiation incident at the focal plane is a function of four continuous variables – two spatial variables, a temporal variable, and wavelength. In order to represent the spatial, temporal and wavelength variation of the focal plane image, the techniques of *sampling* and *scanning* are utilized.

Sampling

In sampling the value of a function is recorded or *sampled* at *discrete* locations. When representing the properties of the incident radiation at the focal plane, sampling occurs in three domains:

1. Spatial sampling

   The spatial variation of light intensity is recorded at only a finite set of locations. This is true even for motion picture film where the size of the set of spatial sampling locations is ultimately limited by the grain size. In focal plane arrays the sampling density is determined by the number of discrete locations or *pixels* where photons are collected.

2. Temporal sampling

   The variation of the image as a function of time must be recorded. Typically this is achieved by sampling the focal plane image at regularly spaced time instants in a process known as temporal sampling.

3. Wavelength sampling

   The radiation incident at the focal plane usually contains a range of spectral wavelengths. Often the incident radiation is recorded in three wavelength bands which roughly correspond to the human visual system's perception of the colors red, green and blue. This is, in effect, a coarse sampling of the wavelength variation at a given spatio-temporal location in the image.

Scanning

Scanning is a hybrid continuous-discrete method for representing the spatio-temporal variation of the radiation incident at the focal plane. In scanning, the time-varying focal plane image is sampled in the vertical dimension to produce a discrete set of horizontal lines, while remaining continuous along each horizontal scan line through time. Usually, the image at the focal plane is scanned by a series of lines which follow a left-to-right, top-to-bottom *raster* as time proceeds.

Integration

The preceding discussion on sampling and scanning is incomplete in that sampling, as described, is an idealization. It is usually not possible, and often undesirable, to *impulse sample* a function, that is, sample at a point. In real systems, sampling invariably involves integration of the value of the function in a neighborhood surrounding the sampling location. For spatial sampling, this implies integration of the function over the spatial variable(s). In a CCD focal plane array sensor for example, each pixel accumulates the charge generated by photons which strike the light-sensitive area of the pixel. For functions of time, the integration is over the temporal variable. In a film camera for example, a mechanical shutter is opened for the duration of the exposure time during which the film integrates the incident

illumination. Similarly, in electronic sensors, temporal integration occurs over the so-called *aperture time*. Integration over wavelength is usually a side effect of the fact that sensing devices and materials (electronic or chemical) respond to photons in a *range* of wavelengths rather than at discrete wavelengths.

Analog and digital representation

The process of sampling the focal plane image yields measurements of the image intensity on a discrete sampling grid. The image intensity is, in most applications, assumed to be a *continuous-valued* quantity, taking values from the set of non-negative real numbers (this convenient approximation is violated in situations where the incident photon count is small).

The issue of actually *representing* the image intensity at a sample point has not, however, been addressed thus far. The image intensity is represented in one of two ways:

- *Analog*

  In an analog description the image intensity is represented by a continuous-valued physical variable which is proportional to the image intensity. Typically voltage is used since it is amenable to electronic signal processing and transmission.

- *Digital*

  In a digital representation the continuous-valued intensity is *quantized* to values drawn from a finite set of *reconstruction levels*. Quantization is a mapping from a continuous-valued variable to a discrete-valued variable. Since the set of reconstruction levels is finite, the reconstruction levels may be put in correspondence with a finite subset of the natural numbers. This makes the representation amenable to finite word-length representation and processing by digital computer.

1.2.5 Analog Video

Analog video systems represent video information using a 1-D, continuous-valued, continuous-time, (analog) signal. This is achieved by scanning the time-varying focal plane image. Two scanning patterns are in common use: 2:1 interlaced scanning and progressive scanning. In both approaches, images are scanned by a series of lines which follow a left-to-right, top-to-bottom *raster* (see Figure 3). In 2:1 *interlaced scanning* all odd numbered lines are scanned followed by all the even numbered lines. The odd and even *fields* which result constitute a *frame*. 2:1 interlaced scanning has the advantage of a field rate which is double that of the frame rate, thus reducing the perception of flicker without requiring additional bandwidth. This is, however, achieved at the expense of reduced vertical and temporal resolution. Interlaced scanning is used in all current analog TV systems as well as in new high-definition television systems.

In *progressive scanning* (see Figure 3) all the lines constituting a frame are scanned one after another. This has the advantage of higher temporal correlation between all adjacent lines of the frame, but the frame rate is half that of the field rate of an equivalent 2:1 interlaced system. Progressive scanning is sometimes referred to as *non-interlaced* scanning and is commonly used in computer monitors and high quality video systems.

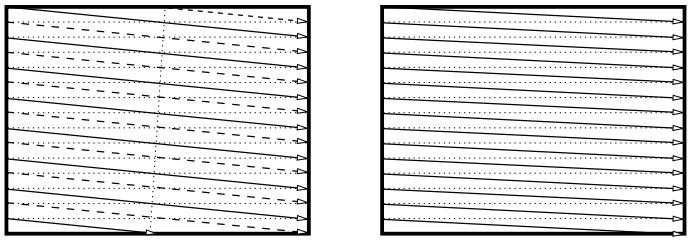

Figure 3: 2:1 interlaced scan (left) and progressive scan (right) rasters.

Several incompatible analog video systems exist. While most of the commonly used systems utilize 2:1 interlaced scanning, these systems differ significantly in their scanning characteristics. The number of scan lines per frame/field and the number of frames/fields represented per second differs among the major standards. These systems also differ in the way in which color information is encoded, as well as in the characteristics of the encoded electrical signal.

The most commonly used analog color video systems are NTSC, PAL and SECAM. *Standards conversion* techniques are used for converting material among these formats as well as to and from other digital and motion picture film formats. The immense existing consumer investment in analog video equipment means that these technologies and standards are likely to remain important for many years to come. See [6] for further information on analog television systems.

1.2.6 Digital Video

Digital video refers to representations of the time-varying image information which utilize *sampling* and *quantization*. The resulting digital representation may be conveniently stored, manipulated, copied and processed using digital computers with no loss of fidelity.

Digitizing time-varying imagery does, however, result in massive quantities of data generated at extremely high bit-rates. This raises important issues regarding the representation (coding and compression), processing, indexing, and transmission of image sequences, all of which are active research and application areas.

Uncoded or *raw* digital video is represented by spatio-temporal samples of the time-varying, continuous image $f(x, y, t)$ projected on the image plane. A digital image sequence (video) consists of a temporal sequence of frames (or less commonly fields) which in turn consist of a finite, regular 2-D lattice of picture elements (called *pels* or *pixels*) which are samples of the spatially-varying illumination intensity pattern incident at the focal plane. Individual pixels may have one or more components to represent multi-spectral information (e.g. three RGB color components). Pixel values are *quantized* and stored using a finite bit-length digital representation. Typically from 1 to 24 bits are used to represent each pixel value. As a result of the flexibility available in choosing the spatio-temporal sampling pattern as well as in the representation of pixel values, a dizzying variety of digital video

formats exist. A very approachable introduction to digital video may be found in [7].

Sampling for digital image sequences

The optical system projects a time-varying, continuous-valued image $f(x, y, t)$ of the scene onto a 2-D focal plane. The value of $f(x, y, t)$ is recorded at *discrete* locations to produce a sampled representation of the visual information. The sampled representation $f(n_1, n_2, k)$ is indexed by the discrete-valued variables $n_1 \in \{0, 1, \ldots, N_1\}$, $n_2 \in \{0, 1, \ldots, N_2\}$ and $k \in \{0, 1, \ldots, K\}$. The spatio-temporal sampling pattern is typically highly regular and constitutes a *sampling grid* [8]. The sampling points in the spatial and temporal dimensions must be carefully chosen so as to faithfully represent the spatio-temporal variation of $f(x, y, t)$ without requiring an excessive number of sampling points which would lead to redundancy. *Nyquist's sampling theorem* [5, 8] provides a lower bound on the sampling grid density required to enable accurate representation of a signal with given spatial and temporal bandwidth. Sampling in accordance with Nyquist's theorem ensures that it is possible to reconstruct the original signal exactly provided that appropriate reconstruction filters are implemented.

Since real-world scenes are usually not spatially band-limited, low-pass, so-called *anti-alias filters* are often used to ensure that the instantaneous image is spatially band-limited, thereby enabling spatial sampling without artifacts. The sampling density in the temporal dimension is typically driven by the application and little can be done about the problem of temporal aliasing.

In modern imaging systems, area scan sensors such as CCD or CMOS arrays have replaced older scanning devices like the Vidicon tube. Area scan devices typically sample the entire image area over a single temporal integration period. This differs substantially from older scanning technologies where samples along a scan line derived from temporally distinct instants. As a result, there is a tendency away from interlaced capture and display systems. Most computer display systems and many camera systems are now able to function in progressive scan mode, yielding high quality frames free from motion artifacts.

For in-depth discussions on sampling structures for spatio-temporal signals, see [8, 9, 10].

Quantization

The process of sampling produces a *continuous-valued* function on a discrete sampling grid. To enable digital storage, transmission or processing of the sampled data the continuous-valued signal must be discretized to yield a finite bit-length representation. This process is known as quantization. A quantizer maps a continuous variable to a discrete variable which takes on values from a finite set of reconstruction levels. The mapping commonly takes the form of a non-decreasing staircase function with a finite number of transitions with corresponding reconstruction levels. Quantization is therefore an inherently lossy process. Quantizer design often requires the minimization of some distortion measure, such as mean squared or mean absolute quantization error. The design of the quantizer has a direct impact on the storage and transmission requirements for samples. For this reason, the design of quantizers plays an important role in compression. See [11, 5, 12, 13] for further details on the characteristics and design of quantizers.

Sampling for color video

The sampling of multi-band image sequences (including color video) raises additional issues regarding the sampling pattern used for each spectral band. Though direct color component sampling such as RGB is common, often the sampling patterns used for color video take advantage of characteristics of the human visual system. By representing color information in a luma/chroma color space rather than in color component form, it is possible to utilize reduced spatial sampling rates for the chrominance components since the human visual system has lower sensitivity to chrominance spatial frequencies as compared with similar luminance spatial frequencies. For additional information on color spaces and sampling patterns for the luma/chroma representation of color images, see [7, 5, 14, 12, 13]. The most commonly used color sub-sampled digital image formats include 4:2:2 (the horizontal sampling rate is halved for chroma components) and 4:2:0 (the horizontal and vertical sampling rates are halved for chroma components). In the 4:4:4 format there is no sub-sampling of the chroma components.

Sampling analog video sources

Sampling of analog video signals to yield digital video sequences is also possible and is routinely accomplished using video capture cards which interface with a computer system. It is important to realize, however, that this approach is fundamentally limited by the scanning characteristics of the analog video signal. If, for example, the analog video signal is interlaced, the resulting sampled output will have the same interlaced structure. Similarly the resolution limitations of the analog signal are reflected in the sampled output. In order to overcome these limitations, techniques for video filtering, enhancement, restoration and deinterlacing have been developed.

1.3 MOTION ESTIMATION

1.3.1 Introduction

The estimation of motion in image sequences is often the first step required for such diverse applications as video compression, standards conversion, filtering, computer vision and restoration. Motion estimation is a vast field and an enormous variety of approaches to the problem may be found in the literature. As a result, there is little point in attempting to present every conceivable or less ambitiously, every prominent approach in even summary form.

Instead, it is hoped that the discussion presented here, which attempts to elucidate the fundamental character and formidable challenges of the motion estimation problem, and which concentrates attention on three important motion estimation frameworks, will leave the reader well prepared to delve deeper into the literature having gained a solid background in the field.

1.3.2 3-D Motion, Projected Motion and Optical Flow

The motion of objects in 3-D space (*3-D motion*) may be described by a 3-D velocity field. The two-dimensional projection on the image plane which corresponds to the 3-D motion is known as the *projected motion* or the *2-D motion field* (see Figure 4).

Often an estimate of the projected motion is desired – this is the *motion estimation* problem. The estimated motion is typically described using instantaneous velocity (flow) or displacement vector fields. Under the assumption of constant velocity motion between frames, the two descriptions are equivalent. More general descriptions of the motion field which take acceleration into account are also possible. The 2-D motion estimate may in turn be used to infer 3-D motion and structure using techniques from the computer vision literature [15].

In an imaging system, however, the only information available is the spatio-temporal variation of the light intensity incident at the focal plane. There is no direct access to the projected motion, let alone the 3-D motion. Instead the spatio-temporal variation at the focal plane results from the interaction of the scene illumination with the objects in the scene, motion of the objects in the scene, as well as changes of camera extrinsic parameters (position, orientation) or intrinsic parameters (focal length, focus setting, etc.) [16]. It is the resulting light intensity variation projected at the focal plane which is recorded. Though this spatio-temporal intensity variation carries information about the projected motion (and therefore also information about the 3-D motion) it does *not* directly correspond to the 2-D velocity field. In particular, not all changes in the image intensity correspond to scene motion, nor does all scene motion result in image intensity variation. For example, changes in scene lighting result in image changes which do not correspond to any 3-D motion, while a featureless, uniformly illuminated disk undergoing axial rotation (a definite 3-D motion) does not result in any observable change in image intensity at the focal plane.

Despite these difficulties, using the time-varying intensity information it is nevertheless possible to construct an *approximation* of the projected motion, known as the *optical flow*.

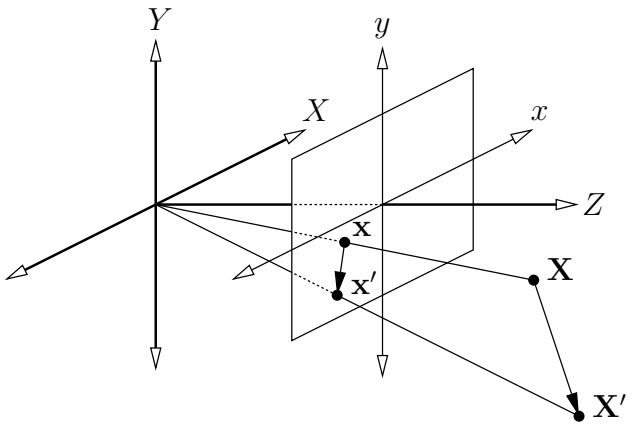

Figure 4: 3-D motion of a point in 3-D space and its projected 2-D motion at the image plane.

1.3.3 The Motion Estimation Problem

Consider a point $\mathbf{X}(t) = [X(t), Y(t), Z(t)]^T$ on a moving object in 3-D space. At time t, the camera system projects the 3-D point $\mathbf{X}(t)$ onto the camera focal plane at position $\mathbf{x}(t) = [x(t), y(t)]^T$. Given two time instants t, τ with $t < \tau$ and corresponding image intensities $f_t(\mathbf{x})$ and $f_\tau(\mathbf{x})$, the position of the projection of the 3-D point on the image plane at time t, given by $\mathbf{x}(t)$, may be related with its position at time τ, given by $\mathbf{x}(\tau)$, in two ways (see Figure 5):

1. The *forward motion*, $\mathbf{d}_{t,\tau}(\mathbf{x}) = \mathbf{x}(\tau) - \mathbf{x}(t)$, describes the displacement in the image plane of the projected 3-D point from time t to time τ. The forward motion estimate $\mathbf{d}_{t,\tau}(\mathbf{x})$ may be used for *backward prediction* where image values are predicted from a *future* reference frame using $f_t(\mathbf{x}) = f_\tau(\mathbf{x} + \mathbf{d}_{t,\tau}(\mathbf{x}))$.

2. The *backward motion*, $\mathbf{d}_{\tau,t}(\mathbf{x}) = \mathbf{x}(t) - \mathbf{x}(\tau)$, describes the displacement in the image plane of the projected 3-D point from time τ to time t. The backward motion estimate $\mathbf{d}_{\tau,t}(\mathbf{x})$ may be used for *forward prediction* where image values are predicted from a *past* reference frame using $f_\tau(\mathbf{x}) = f_t(\mathbf{x} + \mathbf{d}_{\tau,t}(\mathbf{x}))$.

Note that the definition of the forward and backward motion vectors presented here differs from that used in the video compression community. In that context, a forward motion vector is a motion vector that is used for forward prediction (motion compensation from a past reference frame) and a backward motion vector is a motion vector that is used for backward prediction (motion compensation from a future reference frame).

Assuming the velocity of the 2-D projection of the moving 3-D point is constant between frames, the 2-D instantaneous velocity (flow) $\mathbf{v}_t(\mathbf{x})$ and the displacement are related by $\mathbf{d}_{t,\tau}(\mathbf{x}) = \mathbf{v}_t(\mathbf{x})\Delta T$ where $\Delta T = \tau - t$ represents the time interval between frames. If, however, the apparent motion trajectory exhibits acceleration, the first order (constant velocity) representation will fail to accurately model the motion occurring between frames. This is

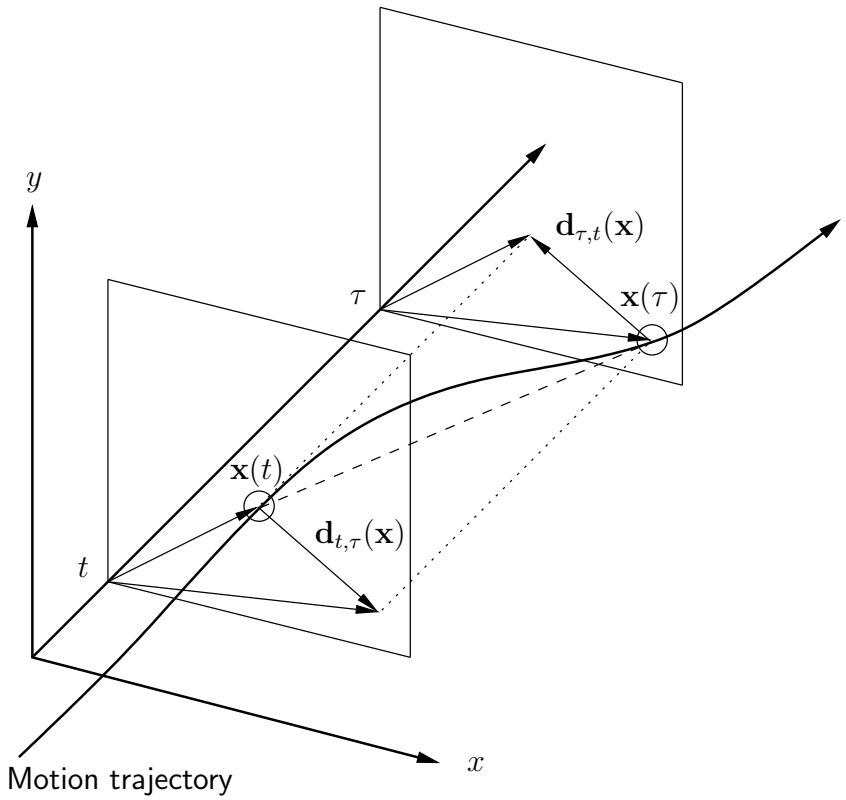

Figure 5: Motion trajectory showing forward displacement vector $\mathbf{d}_{t,\tau}(\mathbf{x})$ and backward displacement vector $\mathbf{d}_{\tau,t}(\mathbf{x})$.

illustrated in Figure 5 where the dashed line represents the first order approximation to the true motion trajectory. For certain applications, such as image sequence interpolation or standards conversion where it is necessary to reconstruct images between temporal sampling instants, performance is improved by utilizing higher order temporal motion models [17]. Where the notion of the displacement field and flow (velocity) field are interchangeable, the term *motion field* will be used.

Thus, the problem of motion estimation is that of determining the motion field (displacement, velocity, acceleration) associated with the observed image sequence.

1.3.4 Complications in Motion Estimation

Occlusion

Occlusion refers to covering of objects due to motion. Translating objects will uncover background regions behind the path of motion and will cover regions in the path of the motion. Similarly, object rotation as well as camera movement can result in occlusions. A consequence of occlusion is that it is not possible to find correspondences for presently visible points which become covered in future frames. Additionally, regions that become uncovered have no motion vectors that point into them. This is illustrated in Figure 6. Occlusion

complicates the process of motion estimation.

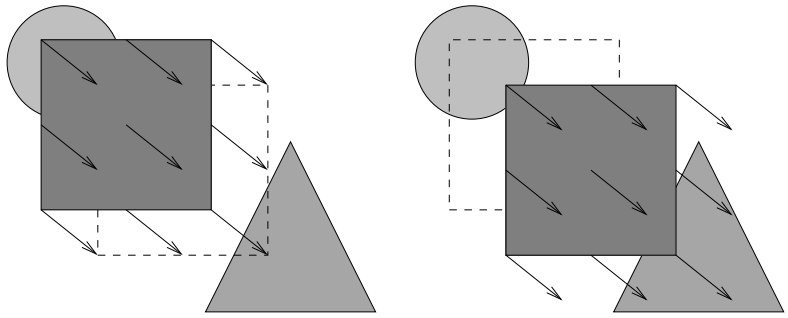

Figure 6: The occlusion problem.

The aperture problem

The *aperture problem* is a term used to describe a fundamental difficulty associated with the estimation of motion. The problem is illustrated on the left of Figure 7 where a uniformly shaded block undergoes translational motion in a north-easterly direction as indicated by the thin velocity vectors. The three smaller squares, labeled A, B and C, represent apertures through which the translating block might be viewed. Viewing the moving block through aperture A, one would only observe the uniform region moving upward, even though there is also a horizontal component of motion. Similarly at aperture B, only the horizontal motion component is perceived. At each of these apertures, only the motion component in the direction of the local image spatial gradient, the so-called *normal flow*, is discernible. At aperture C, however, there is sufficient local spatial variation to yield two distinct local image gradient components. As a result, correct recovery of the motion is possible from the observation at aperture C.

On the right half of Figure 7, the horizontal (u) and vertical (v) velocity components of the motion are represented on the u, v-plane of *velocity space*. The observations at apertures A and B do not uniquely determine the motion, but instead constrain the motion estimate to lie on a locus of points indicated by the dashed horizontal and vertical lines respectively. For aperture C, however, a unique location in the u, v-plane is determined.

The aperture problem is also apparent in the mathematical formulation of the *optical flow equation* which is discussed in Section 1.3.6. The aperture problem is not theoretical in nature – any estimate of local motion computed over a finite sized window of pixels is subject to the problem.

Aggregating constraints to remove ambiguities

It is instructive to note that if the motion component constraints for apertures A and B are combined, the correct motion can be determined. This suggests that the ambiguity

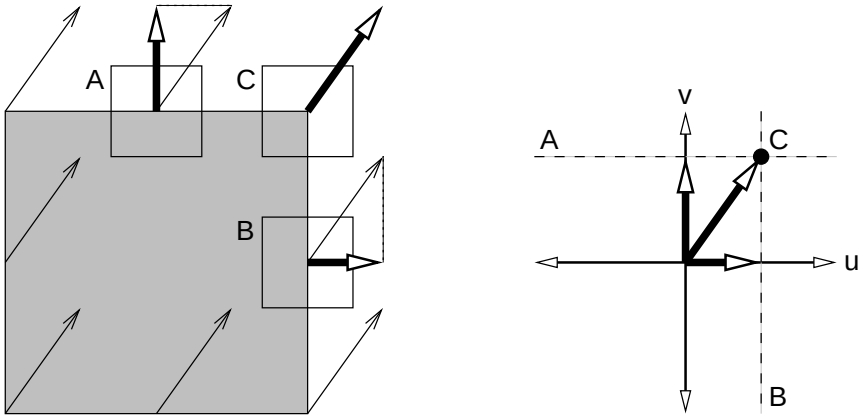

Figure 7: The aperture problem.

associated with the aperture problem can be ameliorated if motion is estimated using textured regions with sufficient spatial gradient information, or by aggregating local motion estimates. Both these approaches imply using information from larger spatial regions. Often this is achieved by imposing "smoothness" constraints to ensure that the motion field constraints are propagated to neighboring estimates to arrive at a smooth motion field which adequately fits the observed data.

Unfortunately, with the aggregation of data over larger spatial regions comes the increased likelihood of the region containing more than one motion which can lead to erroneous estimates. Thus there is a trade-off on region size: larger regions are required to address the aperture problem, but regions must not be so large that they will contain multiple moving objects.

Difficulties associated with multiple motions

As suggested in the previous section, the presence of multiple moving objects can lead to further difficulties in motion estimation. Consider the example (after Burt and Sperling [18] and Weiss [19]) in Figure 8 where two occluding objects undergo independent motion.

As before, local motion estimates from apertures A, B, D and E lead to ambiguous motion estimates in the form of constraint lines in velocity space. While estimators at apertures C and F lead to unambiguous estimates of the true motion of the objects, the local estimates from apertures G and H lead to unambiguous, but nevertheless spurious motion estimates as they do not correspond to the motion of either of the moving objects.

In this example, techniques which globally aggregate motion estimates from the local estimators in order to remove local ambiguities will be biased by these spurious estimates which result from the combined effect of the two independent motions. The global smoothness approach will lead to a motion representation which is something of a "compromise" between the two physical motions and will not correspond to the true motion. This may be seen as a fundamental limitation of the smoothness approach which implicitly assumes a single underlying motion model.

In order to adequately address this problem, knowledge of the structure of the scene is required. In particular, mechanisms are required to identify the independently moving objects so as to best utilize the local motion constraints which are consistent with the object motion. Thus there is an interrelationship between the motion estimates and the *motion segmentation* that is required. Knowledge of the motion is assumed in order to identify the moving objects, and knowledge of the moving objects is used to correctly estimate the motion. Motion estimation in the presence of multiple moving objects represents the current frontier for research.

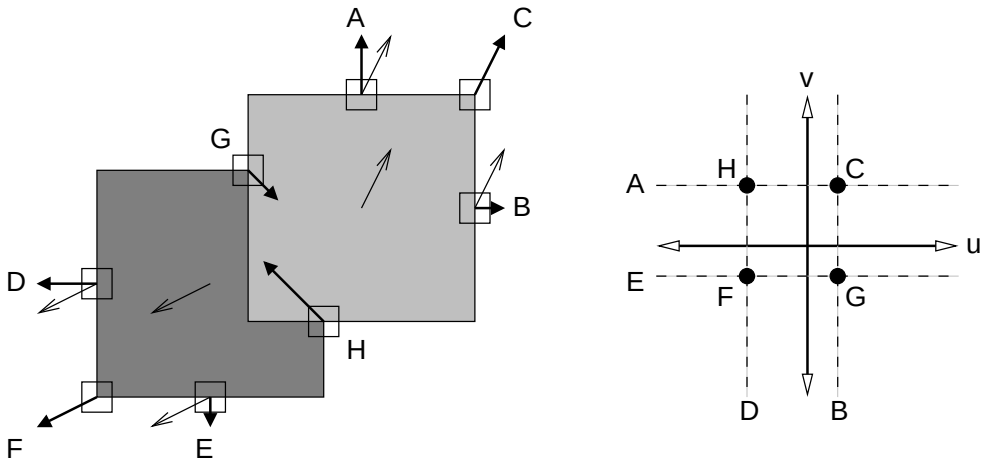

Figure 8: Multiple motions and occlusions leading to incorrect motion estimates.

Motion estimation as an ill-posed problem

If displacement vectors (each consisting of a horizontal and a vertical component) are to be estimated for each point in the spatio-temporal sampling lattice, *twice as many unknowns as there are observations* must be estimated. There are even more unknowns to estimate in the case of motion models which account for acceleration. Thus the problem of motion estimation is inherently under-constrained.

The problem of motion estimation is more accurately described as an *ill-posed* problem. Hadamard [20] defined a problem as *well-posed* if its solution (i) exists, (ii) is unique, and (iii) exhibits continuous dependence on the data. An *ill-posed* problem is one which fails to meet these requirements.

In the case of motion estimation, all three requirements may fail to be satisfied. Existence of the solution may not be guaranteed due to occlusions. Lacking sufficient observation constraints, uniqueness of the solution may fail to be satisfied as discussed earlier. Finally, the motion estimates tend to be highly sensitive to noise in the observed image sequence thereby failing to satisfy the requirement of continuous dependence on the data.

To begin to address these problems, additional constraints are required. One approach is to utilize more data when estimating the motion parameters. This approach typically implies utilizing more than two frames for estimating the motion, as is necessary when estimating

accelerated motion. Another approach is to include assumptions concerning the structure of the motion field. Naturally these approaches are not mutually exclusive. In the section that follows motion field representations which provide structural constraints are discussed.

1.3.5 Motion Field Representation

Modeling 3-D and projected motion

In Section 1.3.2 the concepts of 3-D motion, projected motion and optical flow were introduced. It was shown that the 3-D motion and the projected motion are related according to the geometry of the 3-D to 2-D projection determined by the imaging system. Two geometric projections, orthographic and perspective, were highlighted in Section 1.2.2. The choice of the projection model has a direct effect on the resulting 2-D projected motion of objects in the 3-D scene. By modeling the motion of objects in the 3-D scene it is possible, given the projective transformation, to determine the motion of projected points in the image plane [16].

The motion of points, planar patches and more general surfaces in the 3-D scene result in motion of their projections on the camera focal plane. Given that 3-D motion is typically highly structured (e.g. rigid translations and rotations), it is feasible to model the projected motion which results from a given 3-D motion. Typically, motion of planar patches (and less frequently, quadratic patches) undergoing rigid translation and/or rotation is assumed and the resulting projected motion determined. Knowledge of the relationship between 3-D and 2-D motion has considerable benefits: It provides appropriate models for 2-D motion which may be used in estimating 2-D motion from time-varying image intensity information (the motion estimation problem) [21]. Also, relating 3-D and 2-D motion is essential for computer vision and other applications where there is a need to infer 3-D structure from image sequences [15].

A summary of commonly used 2-D motion models and the 3-D surface and 3-D motion models from which they are derived may be found in [21].

Parametric and non-parametric motion models

Motion field representations may be divided into two broad categories, each having distinct advantages and disadvantages:

- Non-parametric motion field models

  In non-parametric motion field models, a representation of the motion field is typically sought on a finite set of points in the 2-D image plane indexed by $\mathbf{x}$. It is common to choose the set of points to correspond with the uniformly-spaced, discrete image sampling grid Λ.

  The primary advantage of this approach is that arbitrary motion fields may be represented (albeit on a finite sampling grid). The motion field may be interpolated to yield values between sampling points.

  The main disadvantage of the non-parametric representation is that it requires the estimation of a large number of motion parameters. Without additional assumptions

on the spatial variation of the motion field, this may be impossible due to a lack of available observation constraints, as discussed previously. A common assumption is that the motion field is, in some well defined sense, "smooth". The large number of motion vectors make non-parametric models poorly suited to image sequence compression applications as considerable overhead is required for the transmission of the motion vectors which compose the discrete motion field. Non-parametric motion field representations are sometimes referred to as *dense*.

- Parametric motion field models

  An alternate approach is to use a *parametric* motion model which represents the motion field over some region of the image plane. Parametric models are typically continuous parameterized functions of the spatial location $\mathbf{x}$. Common parametric motion models use from 2 to 12 parameters. Once the parameters and the region of support of the model are determined, the model may be evaluated at any location $\mathbf{x}$ within the region, thus there is no need for interpolation.

  Parametric models have the advantage of requiring relatively few model parameters to describe a potentially large region of the motion field. Data from larger regions of the image may be aggregated when estimating the model parameters. Since the number of model parameters is small, this tends to yield more reliable estimates. The parsimonious representation associated with parametric models makes them well suited for compression applications. Many of the parametric models used to describe the motion of regions are derived based on assumptions concerning the projection onto the 2-D image plane of moving planar or quadratic patches in 3-D space.

  Parametric models do have drawbacks. It is not possible to represent arbitrary motion fields using commonly used parametric models without increasing the number of model parameters to be comparable with non-parametric models. As parametric models represent a curve fitting approach to modeling the motion field, they are prone to local error due to over-smoothing. Finally, for general motion fields, estimation of the region of support of the parametric model can be very difficult. Since the region of support of the non-parametric model is a point, this problem is not encountered.

In summary: non-parametric models represent the motion field by samples. Parametric models represent the motion field by parameterized regions.

The parametric and non-parametric motion field models described above are, in some advanced applications, extended to include dependence on the temporal variable t as well as the spatial variable $\mathbf{x}$ to yield the value of the motion field at $\mathbf{x}, t$.

Common motion models

The three most commonly used motion models are the two-parameter translation model, the six-parameter affine model and the eight-parameter perspective model. These models describe the variation of the displacement/flow as a function of the image plane spatial coordinates. Recall that the 2-D displacement vector $\mathbf{d}_{t,\tau}(\mathbf{x})$ describes the motion of a point $\mathbf{x}(t)$ to $\mathbf{x}(\tau)$ where $\mathbf{x} = (x, y)^T$ is the spatial coordinate in the image plane. In describing

the three motion models, explicit dependence on t and τ is dropped since the models apply equally to forward and backward motion fields.

- The two-parameter translational motion model,

$$\mathbf{d}(\mathbf{x}) = \mathbf{b}, \quad \mathbf{b} \in \mathbb{R}^2, \tag{1.1}$$

  is appropriate for modeling the orthographic projection of surfaces undergoing 3-D translation. It finds widespread use in motion compensated video compression applications where the model applies to blocks in the image plane. The translational model is also commonly used at each sampling location in dense motion representations. In this way arbitrary motion fields may be modeled.

- The six-parameter affine motion model,

$$\mathbf{d}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{2\times 2}, \mathbf{b} \in \mathbb{R}^2, \tag{1.2}$$

  which includes the translational model as a special case, models the effects of orthographic projection of 3-D affine motion of planar patches in the 3-D scene. The affine model can describe rotation, but parallel lines in 3-D space are mapped to parallel lines under the projection assumptions.

- The eight-parameter perspective model,

$$\mathbf{d}(\mathbf{x}) = \frac{\mathbf{A}\mathbf{x} + \mathbf{b}}{\mathbf{c}^T\mathbf{x} + 1} - \mathbf{x}, \quad \mathbf{A} \in \mathbb{R}^{2\times 2}, \ \mathbf{b}, \mathbf{c} \in \mathbb{R}^2, \tag{1.3}$$

  is appropriate under the assumptions of perspective projection of 3-D affine motion of planar patches in the 3-D scene. The perspective model can accommodate more general quadrilateral deformations and includes the affine and translation models as special cases.

More complete reviews of parametric motion field models may be found in [21, 9, 22, 23].

Regions of support for motion models

In this section the *region of support* for the motion models presented above is discussed. In particular, various partitions of the image plane into regions on which these parametric models apply are described. Denoting the image plane by $\mathcal{R}$, a *partition* is a set of regions $\{\mathcal{R}_i\}_1^N$ such that $\bigcup_{i=1}^N \mathcal{R}_i = \mathcal{R}$ and $\mathcal{R}_i \bigcap \mathcal{R}_j = \emptyset \ \forall \ i \neq j$.

- Global models

  For global models the partition consists of a single region $\mathcal{R}_1 \equiv \mathcal{R}$, that is, the region of support for global motion models is the entire image plane. Global models are particularly useful for describing camera translation, rotation or zooming on an unchanging scene. Applications which utilize global motion models include computer vision, image stabilization, camera calibration and video compression.

- Block-based models

  In block-based models the partition regions $\mathcal{R}_i$ are equal sized rectangular blocks. A parametric motion model applies for each block. Translational block motion models are used in all current video compression standards (MPEG and ITU H.26x). More general block motion models are also possible by applying the affine or perspective model to each block, though this is less common. Block-based models, though attractive for certain applications, are poorly suited to the task of accurately describing general motion fields. This is a consequence of the motion model being fixed for all locations within each block, and therefore unable to adequately represent motion field discontinuities or deviations which may be located within the block itself.

- Generalized block models (meshes)

  A generalization of the fixed size, regular block partition uses adaptive triangular or hierarchical block-based meshes. In generalized block models, the regions $\mathcal{R}_i$ are triangles or blocks of various sizes. The meshes are selected to model and adapt over time to the intensity and/or motion structures within the image sequence. Generalized block models have the advantage of a relatively small number of easily described regions while delivering considerably improved motion representation as compared with fixed block methods. Mesh-based models have been applied to video compression in the MPEG-4 standard.

- General regions

  Removing the restriction of regularly shaped regions leads to region-based motion models. The regions $\{\mathcal{R}_i\}_1^N$ may take on arbitrary shapes. This leads naturally to the question of how the regions are chosen. Since each region $\mathcal{R}_i \subseteq \mathcal{R}$ represents the region of support of a motion model, $\mathcal{R}_i$ is defined to cover those parts of $\mathcal{R}$ where the model is appropriate. Thus the partition is a *segmentation* of the motion field into regions which exhibit similar 2-D motion. This partition structure is allowed to adapt over time to track the apparent motion.

  Unfortunately, determining the regions $\mathcal{R}_i$ given the image sequence is a difficult undertaking. Determining the partition requires segmentation of the 2-D motion field, but in order to reliably estimate the motion field required for segmentation, some kind of region of support model is needed. This leads to approaches which use an initial motion estimation process followed by segmentation or, more recently, simultaneous motion estimation and segmentation.

  Region-based motion representations often provide accurate and efficient motion representations and as a result are favored for very low bit-rate video encoding. The MPEG-4 standard relies heavily on region-based object representation. As with generalized block models, region-based models require that the shape of the regions be represented in some form.

- Points

  Earlier a distinction was drawn between parametric and non-parametric (dense) motion models. Given the preceding discussion on regions of support for motion models, it is

instructive to think of non-parametric models as the extreme case of parametric models where the region of support for each motion model is a single point. Assuming that the image plane $\mathcal{R}$ is sampled on a finite lattice Λ consisting of N unique sampling locations, the partition of Λ consists of the set of N locations $\{\mathcal{R}_i\}_1^N$. Associated with each location $\mathcal{R}_i$ is a parametric motion model. Thus the total number of parameters used to describe the motion field on Λ is of the same order as the number of sample points. Typically a two-parameter motion model is used at each location which makes the tacit assumption of constant velocity along the motion trajectory between frames. As with any of the models discussed above, it is possible to include second order (acceleration) parameters if necessary.

1.3.6 Relating Motion and Image Intensities

In this section the relationship between the apparent motion and the observed image intensities is discussed. One of the most common assumptions made when attempting to determine optical flow is that the image intensity along a motion trajectory remains constant. This is referred to as the *brightness constancy assumption*. This assumption is implicit in a wide variety of motion estimation techniques even though the formulation of this constraint differs from technique to technique.

The optical flow equation

In their seminal paper, Horn and Schunck [24] formalize the brightness constancy assumption in the *optical flow equation* (OFE). Since the OFE is one of the most widely used models for the brightness constancy assumption, an in-depth treatment is provided. Denoting the time-varying intensity image as $I(x, y, t)$, let the variable s parameterize a motion trajectory. The assumption that the intensity remains constant along the motion trajectory may be described mathematically by

$$\frac{\mathrm{d}I}{\mathrm{d}s} = 0. \tag{1.4}$$

Applying the chain rule yields the *optical flow equation*

$$\frac{\partial I}{\partial x}\, u + \frac{\partial I}{\partial y}\, v + \frac{\partial I}{\partial t} = 0, \tag{1.5}$$

where $u = \partial x/\partial t$ and $v = \partial y/\partial t$ are the horizontal and vertical components of the optical flow respectively. It is common to write the OFE using the more compact vector notation as

$$\nabla I \cdot \mathbf{v} + \frac{\partial I}{\partial t} = 0, \tag{1.6}$$

where $\nabla I = [\partial I/\partial x\ ,\ \partial I/\partial y]^T$ and $\mathbf{v} = [u, v]^T$. It is essential to realize that the OFE (1.6) is a single equation with two unknown components u and v, thus it is not possible to determine the local motion without additional constraints. Only the component of the optical flow that is in the direction of the intensity gradient ∇I, called the *normal flow*, can be determined

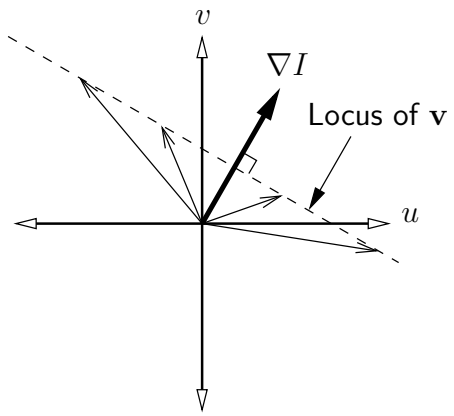

Figure 9: Normal flow constraint.

(see Figure 9). The normal flow is the projection of $\mathbf{v}$ onto ∇I given by $\mathbf{v} \cdot \nabla I / \|\nabla I\|$. By manipulating equation (1.6), it can be seen that the normal flow $\mathbf{v}_\perp$ is given by

$$\mathbf{v}_\perp \doteq \frac{\mathbf{v} \cdot \nabla I}{\|\nabla I\|} = -\frac{\frac{\partial I}{\partial t}}{\|\nabla I\|}. \tag{1.7}$$

Since the OFE constrains only the normal component of the optical flow to equal $-\frac{\partial I}{\partial t}/\|\nabla I\|$, there exists a locus of flow vectors satisfying this constraint. This is indicated by the dashed line in Figure 9. The distance from the origin along the intensity gradient vector ∇I to this constraint line is $\frac{\partial I}{\partial t}/\|\nabla I\|$.

From the above discussion, it is clear that given the intensity gradient and temporal partial derivative only the normal component of the optical flow $\mathbf{v}_\perp$ can be estimated without additional constraints. This limitation is often referred to as the *aperture problem* (see Section 1.3.4) and indicates that the problem is under-constrained. In order to overcome this limitation additional constraints must be formulated. These are typically obtained by aggregating observations and/or by imposing constraints on the structure of the flow field.

The gradient constancy assumption

Changes in scene illumination generally violate the brightness constancy assumption discussed above, resulting in optical flow which does not correspond to actual motion. To address this problem, the *gradient constancy assumption*

$$\frac{\mathrm{d}\nabla I}{\mathrm{d}s} = \mathbf{0} \tag{1.8}$$

has been formulated [25]. Unfortunately, this formulation, which assumes that the brightness gradient is locally constant, is violated under transformations such as scaling and rotation, thus limiting its applicability. Additionally, estimation of the second partial derivatives required in equation (1.8) tends to be error-prone, resulting in less reliable motion estimates.

1.3.7 Selected Motion Estimation Frameworks

In this section three motion estimation frameworks are discussed in detail:

1. Block-based motion estimation methods
2. The classic Horn-Schunck approach based on the optical flow equation
3. A modern Bayesian approach which accounts for multiple motions

Out of the broad selection of approaches to motion estimation that exist in the literature, these three frameworks have been identified due to their pivotal significance: block-based motion estimation and motion compensation techniques form the heart of motion compensated video coding schemes such as MPEG and ITU-T H.26x and as such represent the most tangible and widespread application of motion estimation. The classic presentation of Horn and Schunck [24] pioneered the use of the optical flow equation for motion estimation and revolutionized the way in which the motion estimation problem was addressed. The Horn-Schunck approach and its successors have found widespread use in computer vision and image sequence processing applications where accurate motion estimation is a requirement. Modern Bayesian methods represent the current state of the art for simultaneous motion estimation and segmentation required for reliable estimation of multiple independently moving objects. These methods offer significant performance improvements over earlier approaches and lend themselves well to applications where accurate motion estimation is critical. These methods provide a framework for object-based motion representation and will thus enable emerging object-based image coding schemes.

Block-based motion estimation

Block matching is the simplest motion estimation algorithm, but is nevertheless ubiquitous in video compression applications. The image is divided into N equal sized blocks $\{\mathcal{R}_i\}_1^N$ with dimensions K pixels by L pixels. Typical sizes for K and L used for compression applications are 8 or 16 pixels. The motion of each block is modeled by the two-parameter translational model in equation (1.1). Constant velocity motion is assumed between frames (linear temporal model).

In compression applications, the goal is to reduce the coding cost of the residual error between the current frame to be coded and the motion compensated prediction derived from a reference frame. The residual error is minimized on a block by block basis. Denoting the intensity values of the current and reference frames by $f_c(\mathbf{n})$ and $f_r(\mathbf{n})$ respectively, the objective is to determine the motion vector $\mathbf{d}_{c,r}^{\mathcal{R}_i}$ which minimizes the residual error for block $\mathcal{R}_i$. Formally this is expressed as,

$$\mathbf{d}_{c,r}^{\mathcal{R}_i} = \arg \min_{\mathbf{d} \in \mathcal{S}^{\mathcal{R}_i}} \sum_{\mathbf{n} \in \mathcal{R}_i} \rho\big(f_c(\mathbf{n}) - f_r(\mathbf{n}+\mathbf{d})\big). \tag{1.9}$$

The spatial coordinates $\mathbf{n}$ form a discrete lattice, which is commonly the pixel sampling lattice. Sub-pixel resolution block motion estimation can be achieved by interpolating between known pixel values. $\rho(x)$ is typically the function x^2 or $|x|$, yielding estimates that minimize

the mean squared error (MSE) and the mean absolute error (MAE) respectively. Other matching criteria involving non-linear operators such as the median or maximum matching pixel counts are also possible. The argument of the minimization $\mathbf{d}$ is allowed to range over a set $\mathcal{S}^{\mathcal{R}_i}$, the so-called *search area* (see Figure 10). The difference $f_c(\mathbf{n}) - f_r(\mathbf{n} + \mathbf{d})$ is known as the *displaced block difference* and the collection of such terms for all blocks $\{\mathcal{R}_i\}_1^N$ is called the *displaced frame difference* (DFD).

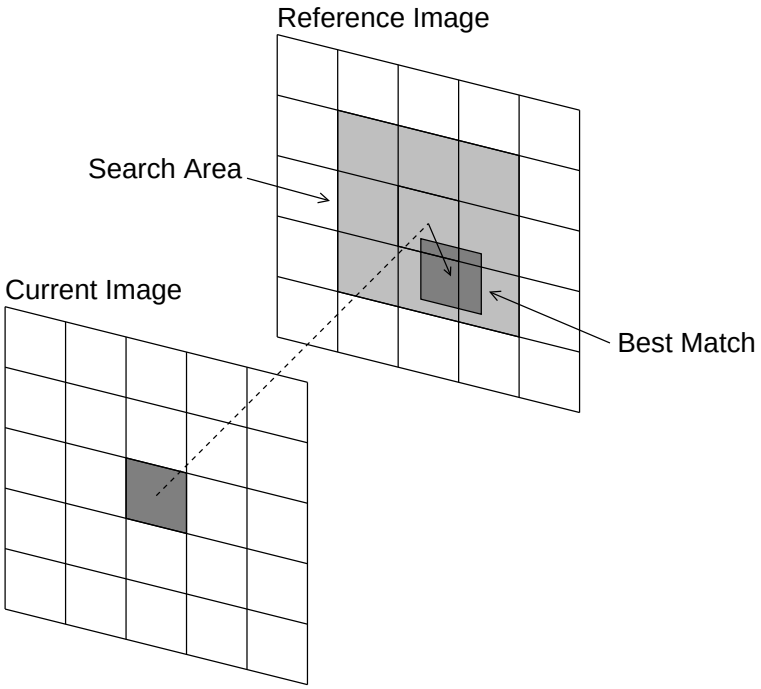

Figure 10: Block motion estimation.

Given the optimization implied by equation (1.9) a procedure must be determined for finding the value of $\mathbf{d}_{c,r}^{\mathcal{R}_i}$ which minimizes the residual error (the objective function). This can be accomplished by exhaustively evaluating the optimization criterion for every location in the search area $\mathcal{S}^{\mathcal{R}_i}$. This is known as *full search*. This approach, though guaranteed to find the optimal value for $\mathbf{d}_{c,r}^{\mathcal{R}_i}$, is computationally expensive so cheaper but sub-optimal alternatives have been investigated.

One such method, known commonly as the "three step search" is illustrated in Figure 11. The example presented is for a 15 pixel by 15 pixel search area. At the first level, the objective function is evaluated at the nine locations illustrated with large circles. The location with the lowest value of the objective function (shown as the large filled circle) is chosen as the starting location for the second iteration. The objective function is then evaluated at the eight locations (illustrated with squares) surrounding the optimal point from the first iteration. Once again, the optimal location is found and the process is repeated at the third level to find the final estimate.

The three step search is a special case of more general "n-step" or "logarithmic" searches. It is important to realize that these fast search algorithms are based on the assumptions of

a unique minimum of the objective function and that the objective function increases monotonically as the estimate moves away from the optimum. In reality the objective function may fail to meet these requirements and the estimate will be sub-optimal. This is usually not too serious for compression applications where the goal is not the estimation of the true motion, but rather a motion estimate which will result in a small prediction residual. The motion estimate which achieves this need not necessarily correspond to the true motion.

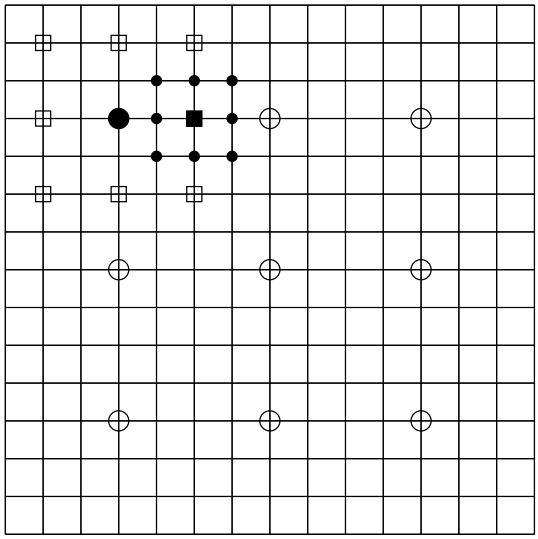

Figure 11: Three step search procedure.

Generalizations of the block matching algorithm to more complex parametric motion models are possible and include techniques used for mesh-based motion models. Hierarchical block motion estimation methods based on image pyramids have also been considered for improving the accuracy and computational performance of the basic block matching algorithm. For more details on these topics, see [22, 9, 21].

Motion estimation using the optical flow equation

In this section motion estimation approaches based on the optical flow equation (1.5) are discussed. The presentation begins with the now-classic formulation of Horn and Schunck [24] and continues with a brief discussion of some extensions to, and the significance of, this ground-breaking work.

As discussed in Section 1.3.6, the optical flow equation (1.5) constrains the motion estimate (u, v) to lie on a line perpendicular to the image brightness gradient ∇I. Thus the local flow cannot be computed without additional constraints. Horn and Schunck addressed this problem by introducing a global smoothness constraint on the local motion estimates. The assumption made is that neighboring points on a moving object are likely to have similar velocities. Horn and Schunck suggested imposing a smoothness constraint which seeks to

minimize the square of the magnitude of the optical flow gradients,

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 \quad \text{and} \quad \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2. \tag{1.10}$$

The motion estimation problem is thus posed as the minimization of the error associated with the optical flow equation

$$\mathcal{E}_b = I_x u + I_y v + I_t \tag{1.11}$$

and the smoothness constraint given by

$$\mathcal{E}_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2, \tag{1.12}$$

where $I_x = \partial I/\partial x$, $I_y = \partial I/\partial y$ and $I_t = \partial I/\partial t$. Introducing the parameter α which gives the relative weight of each term, the total error to be minimized is given by

$$\mathcal{E}^2 = \iint \left(\mathcal{E}_b^2 + \alpha^2 \mathcal{E}_c^2\right) \, \mathrm{d}x \, \mathrm{d}y. \tag{1.13}$$

In practice, the minimization is performed on sampled data, so the integrals are replaced by summations and the partial derivatives are approximated from the sampled data. By applying the calculus of variations the optimal solution to the minimization problem can be shown to be the solution to the equations

$$\begin{aligned} I_x^2 u + I_x I_y v &= \alpha^2 \nabla^2 u - I_x I_t \\ I_x I_y u + I_y^2 v &= \alpha^2 \nabla^2 v - I_y I_t. \end{aligned} \tag{1.14}$$

Horn and Schunck solved this problem using a Gauss-Seidel iteration of the form

$$\begin{aligned} u^{(n+1)} &= \bar{u}^{(n)} - I_x \left[I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t\right] / \left(\alpha^2 + I_x^2 + Iy^2\right) \\ v^{(n+1)} &= \bar{v}^{(n)} - I_y \left[I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t\right] / \left(\alpha^2 + I_x^2 + Iy^2\right), \end{aligned} \tag{1.15}$$

where $\bar{u}$ and $\bar{v}$ are respectively the weighted averages of the horizontal and vertical motion estimates of the eight nearest spatial neighbors of each site.

The iterative procedure in equation (1.15) has interesting characteristics. In regions of the image where the brightness gradient is zero, the velocity estimates are assigned to be the average of the surrounding estimates. Thus motion estimates propagate from areas of high spatial gradient to areas with small gradient, "filling in" uniform regions.

Estimation of the spatial and temporal gradient information required for application of the Horn-Schunck optical flow estimation algorithm is non-trivial. The original Horn-Schunck formulation used $2 \times 2 \times 2$-point spatio-temporal finite difference approximations for computing the necessary gradients. This simple approach is highly sensitive to noise and can lead to biases in the motion estimates. More sophisticated methods utilize smooth interpolation kernels or polynomial curve fitting methods for estimating the gradients.

The smoothness constraint in equation (1.12) is applied globally. Though the constraint is appropriate for regions with similar motion, it leads to erroneous smoothing across motion

boundaries. In an attempt to address this problem, directional-smoothness constraints have been formulated which suspend the smoothness constraint in the direction of the image gradient [25, 26]. Hildreth [27] proposed minimizing equation (1.13) along object contours. None of these techniques adequately addresses the problems of occlusions.

The Horn-Schunck formulation, which includes the use of a smoothness constraint, is an early example of a *regularized* solution to an *ill-posed* problem. Recall that the problem of motion estimation is inherently ill-posed. In addition to other difficulties, the motion estimation problem typically fails to have a unique solution due to the aperture problem. By imposing a smoothness constraint, a new regularized problem is formulated to closely resemble the characteristics of the original ill-posed problem, but with a unique and well-behaved solution.

A Bayesian method for multiple motions

Since the seminal work of Geman and Geman [28], Bayesian methods have become prominent in image and video processing. In this section, the Bayesian framework for simultaneous motion estimation and segmentation proposed by Chang, Tekalp and Sezan [29] is presented. Since the discussion is fully detailed, this section represents a complete example of the application of Bayesian techniques and thus may be considered as an in-depth tutorial. The approach taken here is representative of many modern Bayesian approaches for solving problems in imaging.

In the presentation, lexicographically ordered (row by row ordering of a 2-D image into a 1-D vector) vectors are represented in boldface, while elements of the vectors are in normal face. The motion vector field $\mathbf{d}$ for each frame is modeled as the sum of a parametric field $\mathbf{d}_p$ which is dependent on the segmentation labels $\mathbf{s}$, and a residual field $\mathbf{d}_r$ as

$$d(m,n) = d_p(m,n) + d_r(m,n). \tag{1.16}$$

It is assumed that there are N independently moving opaque objects in the scene, so the segmentation field $s(m,n)$ assumes values from the set $\{1, 2, \ldots, N\}$ and assigns each motion vector $d(m,n)$ to one of the N classes.

Given the current frame $\mathbf{f}_\mathrm{c}$ and a reference frame $\mathbf{f}_\mathrm{r}$, the authors propose computing the *maximum a-posteriori probability* (MAP) estimate of the horizontal ($\mathbf{u}$) and vertical ($\mathbf{v}$) components of the motion field $\mathbf{d}$ and the segmentation field $\mathbf{s}$. The posterior probability is given by the expression, $\mathcal{P}(\mathbf{u}, \mathbf{v}, \mathbf{s} \,|\, \mathbf{f}_\mathrm{c}, \mathbf{f}_\mathrm{r})$. By applying Bayes' rule, the posterior probability can be expressed as

$$\mathcal{P}(\mathbf{u}, \mathbf{v}, \mathbf{s} \,|\, \mathbf{f}_\mathrm{c}, \mathbf{f}_\mathrm{r}) = \frac{\mathcal{P}(\mathbf{f}_\mathrm{c} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_\mathrm{r}) \mathcal{P}(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_\mathrm{r}) \mathcal{P}(\mathbf{s} \,|\, \mathbf{f}_\mathrm{r})}{\mathcal{P}(\mathbf{f}_\mathrm{c} \,|\, \mathbf{f}_\mathrm{r})}. \tag{1.17}$$

The MAP estimate for $(\mathbf{u}, \mathbf{v}, \mathbf{s})$ corresponds to the values $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{s}$ which maximize the posterior probability. Noting that the denominator in equation (1.17) is constant with respect to $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{s}$, the MAP estimate $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{s}})$ may be expressed as

$$(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{s}}) = \arg \max_{(\mathbf{u}, \mathbf{v}, \mathbf{s})} \mathcal{P}(\mathbf{f}_\mathrm{c} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_\mathrm{r}) \mathcal{P}(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_\mathrm{r}) \mathcal{P}(\mathbf{s} \,|\, \mathbf{f}_\mathrm{r}). \tag{1.18}$$

Equation (1.18) consists of three probability expressions. The significance of each expression is interpreted in turn:

1. $\mathcal{P}(\mathbf{f}_\mathrm{c} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_\mathrm{r})$

   The conditional probability density function (pdf) $\mathcal{P}(\mathbf{f}_\mathrm{c} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_\mathrm{r})$ describes how well the current frame $\mathbf{f}_\mathrm{c}$ is modeled given the motion vector field components $\mathbf{u}$ and $\mathbf{v}$, the motion segmentation $\mathbf{s}$ and the reference frame $\mathbf{f}_\mathrm{r}$. More technically, this term describes the *likelihood* of observing the current image given the motion field, the segmentation map and the reference image. A Gibbs distribution with the energy function described by

   $$\mathcal{E}_1(\mathbf{f}_\mathrm{c} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_\mathrm{r}) = \sum_{(m,n)} \big\|\epsilon(m,n)\big\|_2^2, \tag{1.19}$$

   where

   $$\epsilon(m,n) = f_\mathrm{c}(m,n) - f_\mathrm{r}\left(m + u(m,n), n + v(m,n)\right) \tag{1.20}$$

   is used to model this dependence. The term $\epsilon(m,n)$ is the residual error in the motion compensated prediction of the pixel $f_\mathrm{c}(m,n)$ from the reference frame $\mathbf{f}_\mathrm{r}$. Thus the Gibbs energy function $\mathcal{E}_1$ computes the energy of the DFD and represents the degree to which the brightness constancy constraint is satisfied for given $\mathbf{d}$, $\mathbf{s}$, $\mathbf{f}_\mathrm{c}$ and $\mathbf{f}_\mathrm{r}$. Since maximizing the posterior probability of a Gibbs distribution corresponds to minimizing the Gibbs energy function, the MAP solution ensures that the DFD is minimized. It is interesting to note that this formulation of the Gibbs energy for the likelihood function is equivalent to assuming that the pixel errors in the motion compensated prediction are independent, identically distributed (IID) zero mean Gaussian random variables with variance $\sigma^2 = 0.5$. Other models for the error include the longer-tailed Laplacian pdf (L^1 norm), generalized Gaussian pdf, as well as robust measures with outlier rejection.

2. $\mathcal{P}(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_\mathrm{r})$

   The second term, $\mathcal{P}(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_\mathrm{r})$ represents the conditional pdf of the motion vector field given the segmentation field and the reference frame. A Gibbs distribution is again used to model this dependence, this time with the Gibbs energy described by

   $$\begin{aligned}\mathcal{E}_2(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_\mathrm{r}) = \alpha \sum_{(m,n)} \big\|d(m,n) - d_p(m,n)\big\|_2^2 \\ + \beta \sum_{(m,n)} \sum_{(i,j)\in\mathcal{N}_{(m,n)}} \big\|d(m,n) - d(i,j)\big\|_2^2 \, \delta\big(s(m,n) - s(i,j)\big).\end{aligned} \tag{1.21}$$

   Recall that $d(m,n) = d_p(m,n) + d_r(m,n)$, so $\big\|d(m,n) - d_p(m,n)\big\|_2^2 = \big\|d_r(m,n)\big\|_2^2$. Thus the first term measures the energy of the residual motion vector field $\mathbf{d}_r$. Since this term is to be minimized, this corresponds to a minimum norm residual field $\mathbf{d}_r$ while the parametric field $\mathbf{d}_p$ minimizes the DFD. The second term describes the energy associated with the motion vector field, modeled as a Markov random field (MRF). Smoothness is imposed between neighbors if and only if they are within the same labeled region (this is controlled by the multiplicative Kronecker delta term). α and β describe the relative weight given to each term.

3. $\mathcal{P}(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}})$

   The third conditional probability term, $\mathcal{P}(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}})$ models the dependence of the region labels on the reference image $\mathbf{f}_{\mathrm{r}}$. The authors choose not to model dependence on $\mathbf{f}_{\mathrm{r}}$, but instead use this term to model the structure of the segmentation field independent of $\mathbf{f}_{\mathrm{r}}$. In particular they chose a Gibbs distribution with a energy function given by

$$\mathcal{E}_3(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}}) = \gamma \sum_{(m,n)} \sum_{(i,j)\in\mathcal{N}_{(m,n)}} V_2\big(s(m,n), s(i,j)\big) \tag{1.22}$$

   where

$$V_2\big(s(m,n), s(i,j)\big) = \begin{cases} +1 & \text{if } \; s(m,n) = s(i,j) \\ -1 & \text{otherwise} \end{cases} . \tag{1.23}$$

   This is, once again, a MRF model with equation (1.23) describing the two-site clique potential.

It is interesting to note that the second and third conditional probability terms, $\mathcal{P}(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_{\mathrm{r}})$ and $\mathcal{P}(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}})$ are used to model *prior knowledge* of the properties of the solution. In particular, these terms model piecewise smoothness of the motion field and global smoothness of the segmentation field. When these pdf terms do not involve the observations, they are termed "priors" as they represent the a-priori knowledge of the distribution of the solution. The use of prior information to constrain the solution places Bayesian methods in the class of stochastic regularization techniques.

Combining the Gibbs energy functions in equations (1.19), (1.21) and (1.22) yields the Gibbs posterior probability density

$$\mathcal{P}(\mathbf{u}, \mathbf{v}, \mathbf{s} \,|\, \mathbf{f}_{\mathrm{c}}, \mathbf{f}_{\mathrm{r}}) = \frac{1}{Z} \exp\left\{-\mathcal{E}_1(\mathbf{f}_{\mathrm{c}} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_{\mathrm{r}}) - \mathcal{E}_2(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_{\mathrm{r}}) - \mathcal{E}_3(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}})\right\} \tag{1.24}$$

where Z is a scaling constant called the *partition function*. Since the logarithm is an increasing function, the argument which maximizes equation (1.24) also maximizes the logarithm of equation (1.24), thus it is equivalent to maximize

$$\log \mathcal{P}(\mathbf{u}, \mathbf{v}, \mathbf{s} \,|\, \mathbf{f}_{\mathrm{c}}, \mathbf{f}_{\mathrm{r}}) = -\log Z - \mathcal{E}_1(\mathbf{f}_{\mathrm{c}} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_{\mathrm{r}}) - \mathcal{E}_2(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_{\mathrm{r}}) - \mathcal{E}_3(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}}), \tag{1.25}$$

or minimize

$$-\log \mathcal{P}(\mathbf{u}, \mathbf{v}, \mathbf{s} \,|\, \mathbf{f}_{\mathrm{c}}, \mathbf{f}_{\mathrm{r}}) = \mathcal{E}_1(\mathbf{f}_{\mathrm{c}} \,|\, \mathbf{u}, \mathbf{v}, \mathbf{s}, \mathbf{f}_{\mathrm{r}}) + \mathcal{E}_2(\mathbf{u}, \mathbf{v} \,|\, \mathbf{s}, \mathbf{f}_{\mathrm{r}}) + \mathcal{E}_3(\mathbf{s} \,|\, \mathbf{f}_{\mathrm{r}}), \tag{1.26}$$

having dropped the constant term $\log Z$ which does not alter the argument resulting in the minimum. Finally, expanding the terms involving the energy functions $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$, results in the problem of minimizing the energy

$$\begin{aligned} \mathcal{E}_{\mathbf{u},\mathbf{v},\mathbf{s}} = \sum_{(m,n)} \Bigg\{ & \big\|\epsilon(m,n)\big\|_2^2 \; + \; \alpha \, \big\|d(m,n) - d_p(m,n)\big\|_2^2 \\ & + \; \beta \sum_{(i,j)\in\mathcal{N}_{(m,n)}} \big\|d(m,n) - d(i,j)\big\|_2^2 \, \delta\big(s(m,n) - s(i,j)\big) \\ & + \; \gamma \sum_{(i,j)\in\mathcal{N}_{(m,n)}} V_2\big(s(m,n), s(i,j)\big) \Bigg\}. \end{aligned} \tag{1.27}$$

Finding the MAP estimate for the motion vector field $\mathbf{d}$ (with components $\mathbf{u}$ and $\mathbf{v}$) and the segmentation field $\mathbf{s}$ requires the minimization of the energy expression in equation (1.27). The Gibbs energy in equation (1.27) describes all the interacting constraints formulated through the conditional probability terms. To review, the term $\big\|\epsilon(m,n)\big\|_2^2$ represents the energy of the DFD (the energy of the error in the motion compensated prediction) and ensures that the motion estimates are compatible with the observed data. The term scaled by α ensures that the parametric motion field $\mathbf{d}_p$ is responsible for modeling the majority of the variation of the total motion vector field $\mathbf{d}$, with the residual field $\mathbf{d}_r$ being of minimum energy. The term scaled by β ensures that the motion field $\mathbf{d}$ within each labeled region is smooth and the γ term ensures that the segmentation field $\mathbf{s}$ is smooth.

Being highly non-linear, direct minimization of equation (1.27) is a challenging task, so the authors propose a two step, iterative procedure:

1. Update the motion field $\mathbf{d}$ given the current estimate of the segmentation field $\mathbf{s}$. This requires minimizing

$$\begin{aligned}\mathcal{E}_{\mathbf{u},\mathbf{v}} = &\sum_{(m,n)} \big\|\epsilon(m,n)\big\|_2^2 \;+\; \alpha \sum_{(m,n)} \big\|d(m,n) - d_p(m,n)\big\|_2^2 \\ &+\; \beta \sum_{(m,n)} \sum_{(i,j)\in\mathcal{N}_{(m,n)}} \big\|d(m,n) - d(i,j)\big\|_2^2 \,\delta\big(s(m,n) - s(i,j)\big) \end{aligned} \tag{1.28}$$

   with respect to $\mathbf{d}$. The highest confidence first (HCF) method [30] is used to perform this minimization.

2. Update the segmentation field $\mathbf{s}$ using the current estimate of the motion field $\mathbf{d}$. This involves minimizing

$$\begin{aligned}\mathcal{E}_{\mathbf{s}} = &\;\alpha \sum_{(m,n)} \big\|d(m,n) - d_p(m,n)\big\|_2^2 \\ &+\; \beta \sum_{(i,j)\in\mathcal{N}_{(m,n)}} \big\|d(m,n) - d(i,j)\big\|_2^2 \,\delta\big(s(m,n) - s(i,j)\big) \\ &+\; \gamma \sum_{(m,n)} \sum_{(i,j)\in\mathcal{N}_{(m,n)}} V_2\big(s(m,n), s(i,j)\big) \end{aligned} \tag{1.29}$$

   with respect to the segmentation field $\mathbf{s}$. Terms involving the motion vector field are included as $\mathbf{d}$ and $\mathbf{s}$ are interdependent. The minimization is performed using the HCF approach. Once $\mathbf{s}$ is updated, the parametric motion field parameters are updated using a least squares estimate with outlier rejection based on the new segmentation field estimate.

The weight parameters α, β and γ are determined experimentally. As mentioned previously, the number of objects N is assumed to be known a-priori.

Some comments are in order regarding the MAP motion estimation framework. This detailed presentation demonstrates the typical structure of the Bayesian approach. First

the problem is formulated in a probabilistic framework with modeling resulting in an a-posteriori probability expression which relates the unknowns to be estimated to the observations. Bayes' rule is invoked to break the posterior probability expression into likelihood and prior terms. Each of these terms is modeled according to knowledge of the problem domain and a-priori knowledge of the solution. Maximizing the a-posteriori distribution implies an optimization problem. Often this optimization cannot be solved in closed form, so iterative methods are used.

In this section a Bayesian motion estimation approach that simultaneously performs motion estimation and segmentation was examined. This combined approach directly addresses the problem of optimal constraint aggregation to resolve the motion estimation ambiguity of the aperture problem. Local motion estimates are aggregated only if they are supported under a single motion hypothesis (a labeled region) and regions are labeled according to the structure of the motion field.

Recently, techniques based on robust estimation have come to the fore. Some extend the Bayesian framework by using robust, but non-convex Gibbs energy functions [31, 32], while others have used layered representations and clustering techniques [33], and mixture models and Bayesian inference [19]. Most of these approaches are capable of estimating the number of moving objects N. Motion estimation remains a challenging and active research area.

1.3.8 Motion Models for Applications

It should be clear from the preceding discussions that the type of motion information that is appropriate for a given application is highly dependent on the specifics of that application. Or, put more succinctly, there is no single motion representation that is appropriate for all applications.

This is especially clear when considering compression applications, where the objective is not accurate motion representation per sé, but motion compensation to reduce video transmission bit-rate for a given video coder/decoder (codec) implementation complexity. Motion information is used in compression applications to remove temporal redundancy by predicting image values using knowledge of the motion of regions in previously transmitted frames. The motion information (and typically a residual error) is transmitted to enable the receiver to reconstruct the motion compensated frame. For compression, where the objective is reduced bit-rate transmission, accurate motion compensation is traded off against the cost of encoding both the motion information and the residual error. An additional requirement is that the motion compensation scheme be amenable to real-time and VLSI implementation. Though it may be possible to utilize a dense motion representation to achieve highly accurate motion compensation thereby reducing the residual error, the cost of transmitting a dense representation outweighs the gain in reduced residual error. For this reason, and for reduced computational complexity, video compression standards utilize parametric models, especially block-based and more recently, region-based and global motion methods. The motion models used are invariably translation only so as to minimize the cost of coding the motion information as well as to minimize decoder complexity. Despite the simplicity of the motion representations used for compression applications, this presently represents the complexity/performance optimum.

For many video processing applications, however, the situation is markedly different. In computer vision, video restoration, video tracking and similar applications, motion estimation is often the performance limiting factor. For this reason, emphasis is placed on the choice of appropriate motion models and the performance of the motion estimator applied. In restoration applications computational requirements often take second place to performance issues so sophisticated, often computationally intensive, motion estimation techniques are common. Dense motion representations as well as high-order parametric region-based methods are common for such applications.

1.4 VIDEO COMPRESSION

1.4.1 Introduction

Digital video compression is a key enabling technology for applications such as digital television, high-definition television, DVD video, video conferencing, multimedia databases, video over the Internet and many more. A strong argument can be made that there is no other image sequence processing research area that has made so many important new technologies viable. Video compression is a vast and complex field and remains fertile ground for researchers.

The presentation in the sections that follow begins with a discussion of the fundamentals of compression, describing those principles which are common to all compression techniques. This is followed with a discussion of some widely used techniques in compression applications, before delving into the core technologies which are common to all the current compression standards. The material is concluded with a brief mention of emerging techniques which have not yet found wide application in standards.

1.4.2 Compression Fundamentals

What is compression?

Compression refers to methods which reduce the storage or transmission bit-rate requirements for an information source. Fundamentally, compression is achieved by *removing redundancy* from the information stream. In this section emphasis is placed on compression techniques for digital information sources which convey visual information. Applying compression to visual information sources allows several specializations made possible by the spatial and temporal characteristics of visual information.

In order to describe the efficiency of compression techniques, the notion of *coding efficiency* is introduced and is defined as the ratio of the encoded (compressed) bit-rate to the decoded bit-rate.

Compression technologies may be broadly divided into two categories – lossless and lossy. In lossless compression the original data are coded so that *exact* (lossless) reconstruction is possible. In lossy compression, information is discarded in the compression process with the result that decompression yields only an *approximation* to the original data. Lossy compression techniques achieve far higher compression efficiency as compared with lossless techniques.

In applications where some loss of fidelity is acceptable (such as video compression for entertainment) lossy compression is favored due to the high coding efficiency achievable, whereas in applications such as medical image databases, lossless techniques are preferred at the expense of greater storage requirements.

Lossy and lossless compression indicate a more general principle. There is a direct tradeoff between the degree of distortion resulting from the compression process and the coding efficiency. That is, coding efficiency may be achieved at the expense of fidelity. Typically the balance between coding efficiency and fidelity is determined by application constraints. In

digital video compression for example, compression schemes are designed to meet bandwidth requirements while providing acceptable image quality for human viewers.

Why is compression necessary?

The following simple exercise illustrates the data rates involved in the transmission of a modest quality, color digital video stream. The result provides more than ample motivation for compression.

Assume that each image in the sequence contains 288 rows of 352 pixels – a very modest image resolution, well below the quality of standard television. Each image is assumed to be stored as a single full-resolution luminance channel along with two chrominance channels which are sub-sampled in each spatial dimension (4:2:0 format) yielding two chrominance components for every four luminance values. Luminance and chrominance components are assumed to be quantized to $256 = 2^8$ levels and represented as 8 bit binary values. Thus in a *single* image there are a total of $288 \times 352 \times \frac{3}{2} \times 8 = 1.22$ million bits (Mbits) of data. To avoid the perception of flicker, 30 such frames are transmitted per second, yielding a data rate of 36.5 million bits per second (Mbps).

Comparing the required throughput for the digital video signal with that of a high speed telephone modem (56 kbps) or even a dedicated T1 trunk line (1.544 Mbps) it is clear that transmission of raw digital video over these channels is impossible. The implication for storage devices is equally severe. At the video data rate computed above, a DVD which stores around 17 billion bytes of information would contain fewer than 8 minutes of video – a far cry from the feature-length TV-resolution movies currently delivered on this medium.

Video compression thus exists to maximize the efficiency of utilization of band-limited transmission channels and finite storage devices. In order to deliver video content to a remote user, information must be transmitted (over some channel) or delivered on some storage medium. The transmission channel is characterized by finite bandwidth, thus limiting the maximum information transmission rate, while the storage medium will have finite storage capacity. If the delivery and storage of digital video content is sought, some form of data compression is required to reduce the data rate while still maintaining acceptable quality.

Why is video amenable to compression?

There are three factors which not only make video compression possible, but highly practical:

- Spatial redundancy

  Pixel values within typical images exhibit a very high degree of spatial correlation. A consequence of this fact is that it is possible to *predict* with high confidence the value of a pixel given nearby pixels. The "predictability" implied by spatial correlation allows the image to be represented in a compressed manner where the intra-frame correlation characteristics are used to represent the image information in a more compact way. The most prominent techniques for removing spatial redundancy are differential coding and transform coding.

- Temporal redundancy

Temporal redundancy refers to the correlation of pixels over time. Similar to the spatial correlation discussed above, pixels tend to be correlated with pixels in past and future frames. The vast majority of image sequences exhibit motion, where objects appearing in a given frame typically also appear in temporally neighboring frames with reasonably little change. Techniques have been developed to compensate for the motion occurring from frame to frame. Motion compensation enables accurate prediction of pixel values from temporally neighboring frames and may be used to remove temporal redundancy in a manner entirely analogous to the removal of spatial redundancy. Typically the major gain in coding efficiency for video data is achieved by removing temporal redundancy.

- Characteristics of the human visual system

  Most image sequences are created for human consumption – be it for entertainment or otherwise. This seemingly trite observation actually plays a fundamental role in the design of video compression schemes which take advantage of characteristics of the human visual system (HVS) to further reduce the amount of information transmitted or stored. Psycho-visual studies have shown that human visual perception is not uniformly sensitive across all spatial frequency ranges, luminance levels, color, texture or temporal effects. These observations suggest that spatial and temporal information which will not be perceived by the viewer need not be coded, thereby allowing a reduction of the video data rate with minimal loss of perceived quality. In the major video coding standards, HVS characteristics are typically taken into account in the design of quantizers. Information which is perceptually less importance is removed in the process of quantization.

In summary, video compression techniques remove spatial and temporal redundancy and code information specifically for a human observer. All video compression standards utilize some combination of these approaches.

1.4.3 Statistical Background

Random variables and correlation

A formal definition of image pixel correlation has not yet been provided. To do this, consider the value of each image pixel to be the realization of a random variable $\{X_i\}_1^N$. Let $\bar{X}_i = \mathcal{E}(X_i)$ denote the expected value of the random variable X_i. The *covariance* of two random variables X_i and X_j is defined as $K(X_i, X_j) = \mathcal{E}\{(X_i - \bar{X}_i)(X_j - \bar{X}_j)\}$. Expanding this expression gives $K(X_i, X_j) = \mathcal{E}\{X_i X_j\} - \mathcal{E}\{X_i\}\mathcal{E}\{X_j\}$. Two random variables are said to be *uncorrelated* if their covariance is zero. This is equivalent to requiring that $\mathcal{E}\{X_i X_j\} = \mathcal{E}\{X_i\}E\{X_j\}$.

Now construct a vector $\mathbf{X} = [X_1, X_2, \ldots, X_N]^T$. The expectation of the random vector $\mathbf{X}$ is the vector given by $\bar{\mathbf{X}} = \mathcal{E}(\mathbf{X})$. For a random vector $\mathbf{X}$, the associated *covariance matrix* $\mathbf{K}$ is defined as the expectation of the outer product of centered random vectors as, $\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathcal{E}\{(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T\}$. Since $\mathbf{K}_{ij} = \mathcal{E}\{(X_i - \bar{X}_i)(X_j - \bar{X}_j)\} = \mathcal{E}\{(X_j - \bar{X}_j)(X_i - \bar{X}_i)\} = \mathbf{K}_{ji}$, $\mathbf{K}$ is symmetric. Also, $\mathbf{K}_{ii} = \mathcal{E}\{(X_i - \bar{X}_i)(X_i - \bar{X}_i)\}$, which is simply the variance σ_i^2 of X_i.

From this discussion it should be clear that a set of random variables $\{X_i\}_1^N$ is uncorrelated if its covariance matrix is diagonal (with the variances of the individual random variables appearing on the diagonal).

Decorrelation by the Hotelling transformation

Decorrelating a set of random variables may be achieved with the application of the Hotelling transformation, also known as the discrete Karhunen-Loève transformation (KLT). For conciseness, the abbreviation KLT shall be used to refer to the discrete form of the Karhunen-Loève transformation. Application of the KLT *diagonalizes* the covariance matrix $\mathbf{K}$, that is, it decorrelates the random variables $\{X_i\}_1^N$.

The KLT relies on the fact that a set of N orthonormal eigenvectors $\{e_i\}_1^N$ with corresponding eigenvalues $\{\lambda_i\}_1^N$ can be found for the covariance matrix $\mathbf{K}$. In particular, constructing $\mathbf{\Phi}$, the rows of which contain the eigenvectors of $\mathbf{K}$ as $\mathbf{\Phi} = [e_1, e_2, \ldots, e_N]^T$, implies that $\mathbf{K}\mathbf{\Phi}^T = [\lambda_1 e_1, \lambda_2 e_2, \ldots, \lambda_N e_N] = \mathbf{\Phi}^T\mathbf{\Lambda}$ where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$. Since $\{e_i\}_1^N$ are orthonormal, $\mathbf{\Phi}$ is unitary and $\mathbf{\Phi}^T = \mathbf{\Phi}^{-1}$ yields, $\mathbf{\Phi}\mathbf{K}\mathbf{\Phi}^T = \mathbf{\Lambda}$. Forming $\mathbf{Y} = \mathbf{\Phi}(\mathbf{X} - \bar{\mathbf{X}})$ yields a random vector $\mathbf{Y}$ which has zero mean and uncorrelated components. The covariance matrix associated with the random vector $\mathbf{Y}$ is diagonal, with the diagonal components being the eigenvalues of $\mathbf{K}$ which are variances σ_i^2 of X_i.

In the case of multi-spectral images X_i may be considered to be a vector random variable and the results above remain valid. In the multi-spectral case, however, there is the additional issue of inter-band correlation to consider.

Basis vector interpretation of the discrete KLT

The KLT applied to a zero mean random vector $\mathbf{X}$, $\mathbf{Y} = \mathbf{\Phi}\mathbf{X}$ is a decomposition in terms of an orthonormal basis consisting of the eigenvectors of the covariance matrix $\mathbf{K}$ associated with $\mathbf{X}$. Thus the KLT basis is *image specific*. The basis is chosen according to the image covariance statistics and is optimal by virtue of this fact. This differs from fixed transforms such as the discrete Fourier transform (DFT) or discrete cosine transform (DCT) the bases of which are fixed, independent of the data to which the transform is applied. Nevertheless, the interpretation of the KLT in terms of a basis decomposition enables comparison between the KLT and other transforms.

The significance of the KLT

The KLT is optimal in the sense that it decorrelates the data. This may be equivalently interpreted in terms of maximal energy compaction (most of the transform coefficients are zero) or minimum variance (due to the representation with respect to the principal component basis). The KLT thus provides a lower bound on the variance for transform coefficients.

A suboptimal basis – the discrete cosine transform

Unfortunately, the KLT is signal dependent and requires knowledge of both the mean vector and covariance matrix associated with the random vector $\mathbf{X}$. These must often be estimated

from the data. For general image coding purposes the fact that the decomposition basis is image dependent is problematic as it requires transmission of the basis along with the transformation coefficients to enable reconstruction.

For these reasons, the performance of fixed basis decompositions such as the discrete Fourier transform, Haar transform, discrete cosine transform, discrete Walsh transform and others have been investigated. In performance comparisons [34, 12] it has been shown that the DCT provides performance which very closely approximates that of the KLT in terms of energy compaction, mean square reconstruction error and rate/distortion. This is the reason why the DCT enjoys widespread use in image and video transform coding.

1.4.4 Basic Tools for Compression

Quantization

A scalar quantizer is a mapping of a continuous-valued variable x to a discrete-valued variable d which assumes values from a finite set of *reconstruction levels*. The mapping is generally a staircase function similar to the one shown in Figure 12. The staircase quantizer mapping is fully described by the set of $N-1$ *decision* or *transition levels* $\{t_1, t_2, \ldots, t_{N-1}\}$ and the corresponding *reconstruction levels*, $\{r_1, r_2, \ldots, r_N\}$.

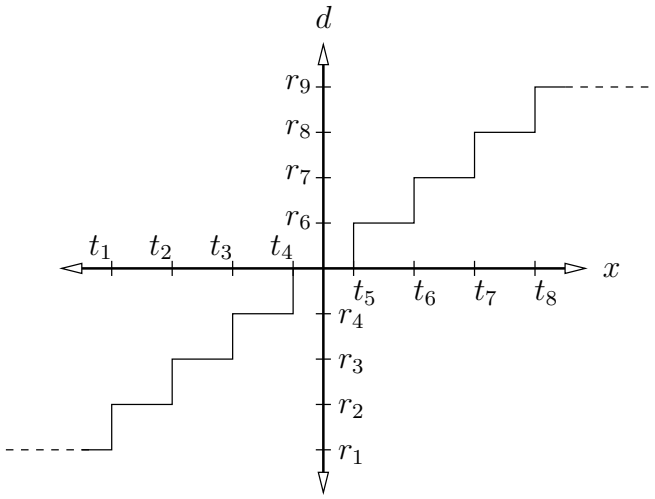

Figure 12: Scalar quantizer mapping function.

Application of the quantizer mapping is a lossy operation. Once the function is applied it is no longer possible to exactly reconstruct the values of the original continuous-valued variable (the map is many-to-one). Apart from being necessary for obtaining a digital (finite bit-length) representation of an analog value, quantization plays an essential part in compression applications. Since the quantizer transition and reconstruction levels are a design variable, quantizers may be constructed so as to minimize the mean squared error between the quantized output and the original input signal. This approach was pioneered by Lloyd and Max [11, 12, 5] and requires knowledge of the probability density function of the vari-

able x. Design of quantizers for video coding applications may also be done on the basis of psycho-visual testing, coupled with existing statistical design methods [14].

Quantization is a first step toward digital coding. Since the quantizer is a mapping to a total of N reconstruction levels, the signal is thus represented using only N values. These values may be assigned codewords according to their relative frequency of occurrence. The freedom available in quantizer design is put to good use in compression applications, especially for the quantization of the coefficients resulting from transform coding.

A generalization of scalar quantization to vector-valued variables is *vector quantization* (VQ). In vector quantization, the problem is more complex in that the decision levels become decision boundaries in higher dimensional space, and the reconstruction levels constitute a *codebook*. Though there exist compression approaches based on vector quantization, these are not considered here. The interested reader is referred to [35].

Differential coding

Instead of coding a signal directly (by coding the value of the signal at each time instant), the approach taken in *differential coding* is to code the *difference* between the signal and a prediction of the signal. As a result, differential coding also goes by the name *predictive coding*. The basic premise of differential coding is simple – code only the information that cannot be predicted at the receiver. Thus differential coding is a framework for removing redundancy from the signal.

The idea is demonstrated using the well known technique of differential pulse code modulation (DPCM). The structure of the DPCM coder/decoder (codec) is shown in Figure 13, where quantized values are distinguished with an overbar, and predictions with a hat.

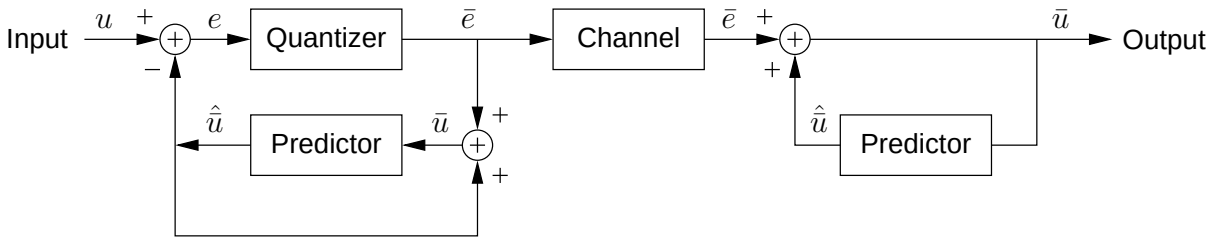

Figure 13: Differential pulse code modulation (DPCM) coder/decoder (codec).

The operation of the encoder is analyzed first. The *prediction error* e is the difference between the unquantized input u and the prediction of u given by $\hat{\bar{u}}$ as $e = u - \hat{\bar{u}}$. The prediction error e is quantized to yield $\bar{e}$. The reconstructed quantized approximation of u is found at the input to the predictor and is given by $\bar{u}$ which is related to the quantized prediction error and the predictor output as $\bar{u} = \bar{e} + \hat{\bar{u}}$. Adding the expressions $e = u - \hat{\bar{u}}$ and $\bar{u} = \bar{e} + \hat{\bar{u}}$ and rearranging yields $u - \bar{u} = e - \bar{e}$. Thus the difference between the input u and the reconstruction $\bar{u}$ is exactly the quantization error resulting from quantizing the prediction error e.

The operation of the decoder side is obvious once it is noticed that the decoder loop is identical to the prediction loop at the encoder. Thus the decoder reconstructs $\bar{u}$ as its output. Recall from above that $\bar{u} = u - (e - \bar{e})$, thus the reconstruction $\bar{u}$ is equal to the input signal u but for the quantization noise of the error signal given by $e - \bar{e}$.

With a good signal predictor, the variance of the error signal will be much smaller than the variance of the input. The result is that for a given mean squared quantization error, fewer bits are required to code the error signal than the original input signal. Furthermore, the pdf of the error signal tends to deviate significantly from the uniform pdf which enables efficient coding.

The design of the predictor is critical to the success of the DPCM scheme. Simple approaches in image coding use previous pixels in the scan-line and/or on previous lines to predict the value of the current pixel. More sophisticated approaches also utilize pixel values from past fields or frames. See [14] for more details on common DPCM predictors for video coding.

Transform coding

Transform coding techniques, applied to the problem of image and video compression are highly effective for removing redundancy. In Section 1.4.3 the statistical background required to understand decorrelation techniques based on representations of the pixel intensities in a *transform domain* with respect to a decomposition basis were discussed. It was shown how the choice of a suitable transformation led to a representation in a transform domain using a small number of coefficients which have minimum or near minimum variance. The covariance matrix associated with the transformed random vector could be made diagonal or close to diagonal by choosing the appropriate basis, indicating uncorrelated or nearly uncorrelated random variables.

An appropriate transform domain representation compacts most of the signal energy into a small number of non-zero coefficients. In order to reconstruct the signal, only these non-zero coefficients are needed. The remaining coefficients need not be stored or transmitted. Thus the transform domain coefficients serve as an efficient representation of the original signal. At the receiver, the original signal may be reconstructed from the coefficients given knowledge of the transform basis. Fixed basis transforms, though suboptimal in comparison with the KLT, are therefore appropriate (see Section 1.4.3).

Further representational savings are possible by quantizing the transform coefficients, however these savings come at the expense of a lossy signal representation. Exact reconstruction of the original signal is no longer possible due to the loss of information inherent in the process of quantizing the transform domain coefficients. For image and video compression applications the quantization levels are chosen according to the characteristics of the human visual system. Coefficients which represent visual information of low perceptual importance are represented with a coarsely quantized approximation, while perceptually important coefficients are represented with a fine quantization.

When coding the coefficients for transmission, the fact that the transform domain coefficient values tend to be statistically non-uniformly distributed allows for efficient bit allocation by an entropy coder.

The discussion of transforms in Section 1.4.3 centered on random vectors. The region of support of the spatio-temporal sampling lattice which is represented by the random vector was not, however, addressed. Though it is possible to utilize transform coding techniques where the observation vector is an entire image, or even an entire image sequence, by far the most commonly used approach is 2-D *block transform coding* for the removal of spatial redundancy.

In block transform coding, an image is typically partitioned into equal-sized square or rectangular regions called *blocks*. The most commonly used block size is 8×8 pixels. The transform is applied to each block and the resulting coefficients are quantized and coded for transmission. This, in essense, describes the basic operation of the JPEG standard for still image compression which utilizes the discrete cosine transform. In video coding, this approach is called *intra-frame* transform coding. Frames coded in this manner are referred to as intra-frames or *I-frames* for short.

In Section 1.4.3 it was mentioned that the DCT is the most popular transform used for image and video applications as its performance, in terms of coefficient variance, energy compaction and rate/distortion characteristics is close to that of the optimal KLT. A discussion of the DCT along with details on its efficient implementation may be found in [14]. Figure 14 shows the DCT basis functions for the 8×8 pixel 2-D DCT that is most commonly used in video compression standards.

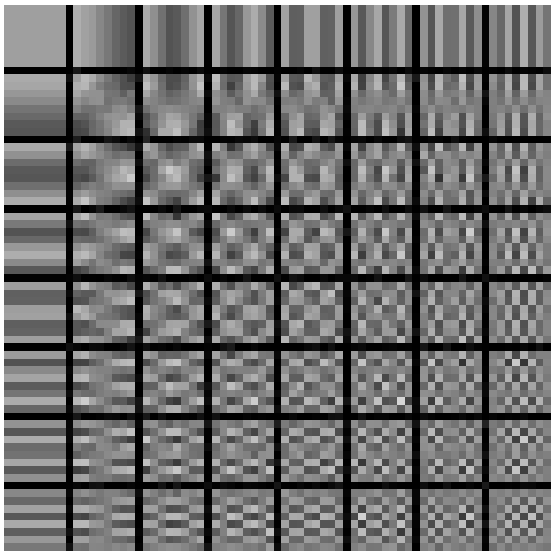

Figure 14: Basis functions of the 8×8 pixel 2-D discrete cosine transform.

Motion compensated prediction

Motion compensated prediction is the tool of choice for removing temporal redundancy. Typical image sequences exhibit a high degree of temporal correlation. In order to take full advantage of this it is necessary to account for the motion that occurs from frame to frame. The basic principle of motion compensated prediction is to predict the value of

pixels in the current frame from one or more reference frames. As with differential coding techniques the prediction error is transmitted but in addition, the motion information must also be transmitted (as so-called *side information*) to ensure that the decoder can correctly reconstruct the frame.

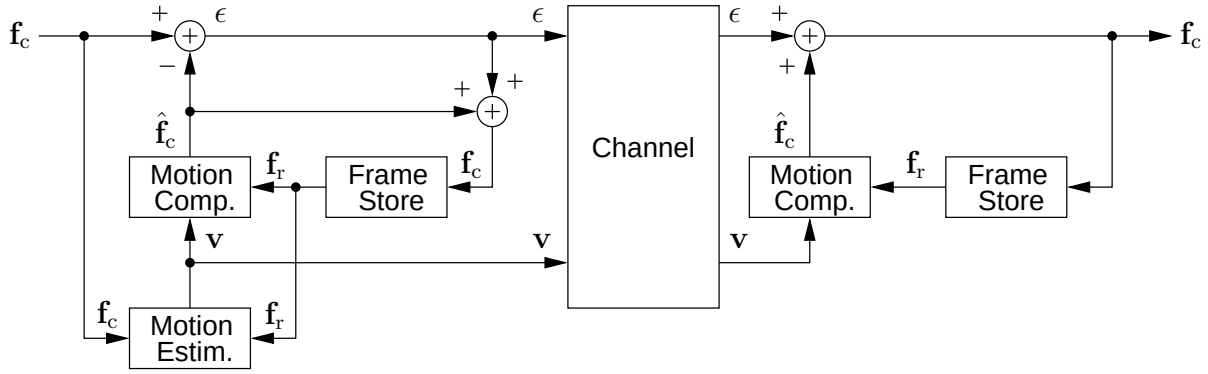

Figure 15: Simplified motion compensated prediction codec.

A block diagram of a highly simplified motion compensated prediction codec is shown in Figure 15. The similarity to the DPCM codec in Figure 13 should be apparent. Both the coder and the decoder store a complete reference frame $\mathbf{f}_r$ in their frame stores. The operation of the motion compensated prediction codec is illustrated with a simplified example.

Assume for simplicity that the reference frame is the last transmitted frame (in general, this need not be the case as a reference frame may be used for predicting several frames before being replaced). Additionally, assume, as is the case in all major video compression standards, that the input frame $\mathbf{f}_c$ is partitioned into equal sized blocks of pixels called *macroblocks*. Macroblocks are typically 16×16 pixels in dimension.

The process of encoding a macroblock begins with the motion estimator searching for a region in the stored reference frame $\mathbf{f}_r$ which closely matches the macroblock to be coded. This is achieved using any of the motion estimation techniques discussed in Section 1.3.7. The most common approach is to use translational block motion estimation (see Figure 10). The result of the motion estimation process is a motion vector $\mathbf{v}$ which is used to form a prediction $\hat{\mathbf{f}}_c$ for the current frame macroblock using the closest matching region in the reference frame. A residual prediction error (the displaced block difference) is formed and transmitted along with the motion information to the decoder. In practice, transform coding, quantization and entropy coding of the prediction residual as well as differential and entropy coding of the motion information occurs.

This encoding process is repeated for each macroblock composing the frame. Once the entire frame is encoded and transmitted, the process is repeated with the next input frame. In this example, the reference frame is replaced with the last encoded frame. Frames coded with respect to a single reference frame are referred to as *predicted-frames* or *P-frames*.

The decoder includes a frame store which contains the last decoded frame. This stored frame is identical to the reference frame $\mathbf{f}_r$ used at the encoder. Using the received motion

information for each macroblock, the decoder reconstructs the predictions computed at the encoder using the stored reference frame $\mathbf{f}_r$. These predictions are then corrected using the prediction residuals received for the macroblock. The result is a reconstructed macroblock. This process is repeated for each macroblock and the completely reconstructed frame is stored in the frame buffer as the next reference frame.

Bidirectional coding is also possible. Figure 16 illustrates how the macroblock to be coded can be predicted on the basis of a past reference frame (*forward prediction*), a future reference frame (*backward prediction*), or a combination of both past and future frames (*interpolation*).

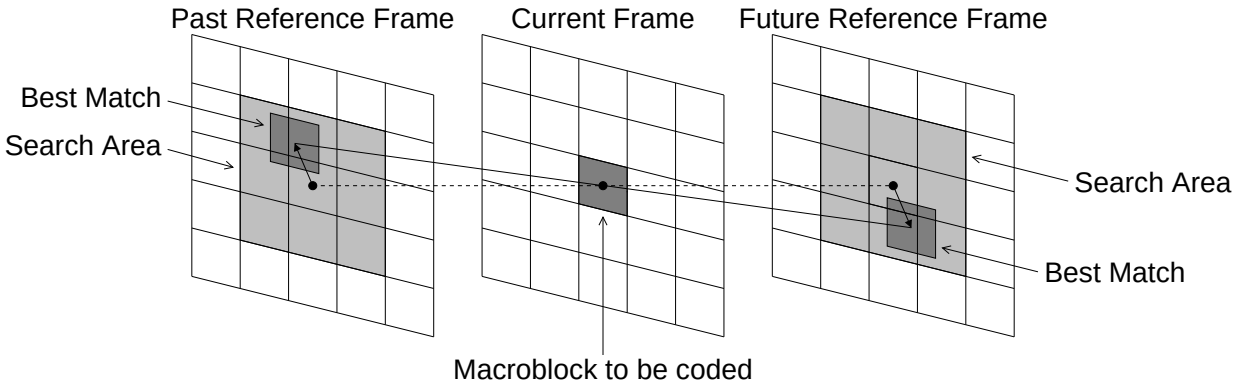

Figure 16: Forward, backward and bidirectional (interpolated) macroblock prediction.

Prediction based on a future reference frame is possible provided that the reference frame is encoded and sent to the receiver before it is used as a reference. This requires the encoding order to be different from the order of the input frames. Figure 17 illustrates a group of pictures (GOP) with a typical structure of intra-coded (I) frames, bidirectionally-coded (B) frames and predicted (P) frames. The encoder reorders the input sequence as necessary to achieve the desired encoding structure. Due to the encoder side reordering, all bidirectional predictions become causal so the decoder can reconstruct the frames. The decoder reorders the reconstructed frames for display.

Bidirectional coding has the major advantage of being able to efficiently code regions which become uncovered. Regions that are uncovered cannot be predicted from past frames (since they are covered in the past) but they can be predicted from future frames. Additionally, interpolating predictions from both past and future reference frames can reduce the prediction error. As the number of consecutive B frames increases, however, the correlation between the reference frames is typically reduced resulting in increased errors in interpolation.

Codeword assignment

Information that must be transmitted to the receiver for decoding, such as quantized prediction errors in the case of DPCM, quantized DCT coefficients in the case of transform coding, or motion vector information, must be represented as a coded bitstream for transmission. In

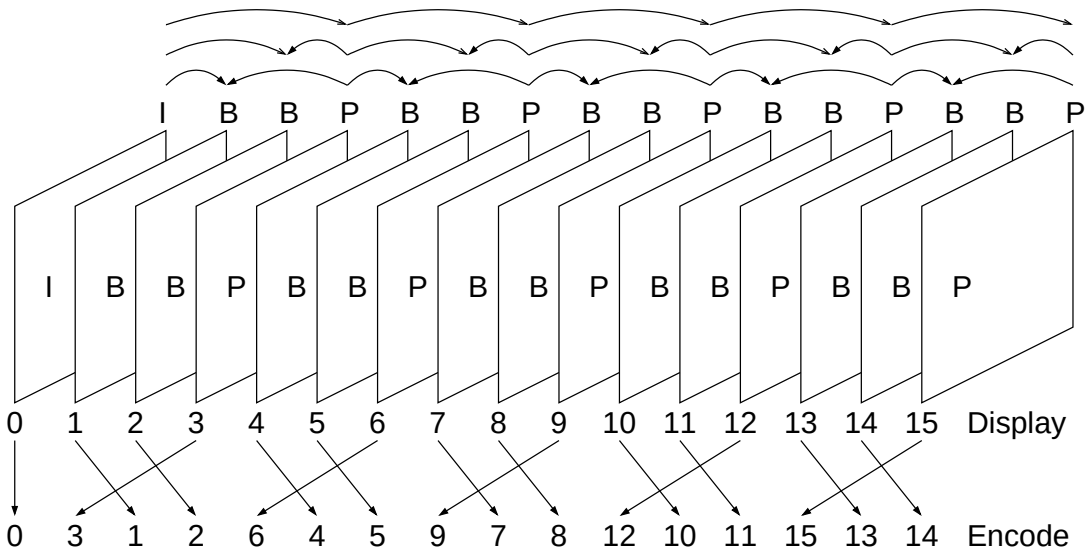

Figure 17: Group of pictures (GOP) structure consisting of I, B and P frames showing encoding and display order.

this section the code assignment techniques that are most widely used in video compression standards are briefly discussed. For further details on coding and information theory results for image and video coding see [12, 9].

Recall that a scalar quantizer is a mapping from a continuous-valued variable x to a discrete-valued variable d which assumes values from a set of N reconstruction levels. It is helpful to think of each of the N reconstruction levels as a *symbol* in the *source alphabet*. The problem of codeword assignment is that of assigning to each symbol a unique codeword from a *code alphabet*. The codewords are sequences of (typically) binary digits.

The probability distributions of the source symbols resulting from quantization in intra-frame and inter-frame predictive coders tend to be non-uniform. The problem of codeword assignment under these circumstances is effectively addressed using the technique of Huffman coding which assigns a variable-length code (VLC) to each of the source symbols. Source symbols which have high frequency of occurrence (and thus must be transmitted more frequently) are assigned short codewords, while symbols which occur rarely are assigned longer codewords. Huffman codes are called *prefix* codes as no codeword in the code alphabet is a prefix of any other. This allows codes to be concatenated producing a stream of bits which is easy to parse prior to decoding. Huffman codes approach the optimal coding efficiency given by the *entropy* (a measure of information content) of the source [36].

Variable-length codes see widespread use in image and video compression applications. One drawback is that VLC coding leads to a bit-rate that fluctuates according to image content (referred to as *variable bit-rate* or VBR). This can lead to buffer overflow and underflow problems in architectures where buffers are used to facilitate data transmission at a constant bit-rate. Another, more serious problem with variable-length codes is the lack of robustness to bit errors that may occur during transmission. Fixed-length codes (FLC) also see application in image and video coding [37], but to a lesser extent and mostly under

special circumstances.

1.4.5 Hybrid Coding

Introduction

In the previous sections the most important tools used in compression were introduced and it was shown how these tools work individually. In this section these techniques are combined to form what are known as *hybrid codecs*, which combine motion compensated prediction and transform coding techniques to achieve highly efficient video coding. What is presented here is the common core of the existing ITU-T and MPEG standards. Each of the major standards differs in relatively minor ways from the basic codec structure that is presented. The differences that do exist are to address the specific design objectives of each standard.

Hybrid coding

The term *hybrid* is used to describe video coders which utilize transform coding *as well as* motion compensated predictive coding techniques. Reviewing briefly:

Transform coding is especially efficient for removing spatial redundancy in images and is applied at the block level, typically 8×8 pixels. A transformation (typically the 2-D DCT) is applied to the block of pixels, yielding an efficient transform domain representation. The transform domain coefficients are then quantized and coded for transmission. The block values may be (approximately) reconstructed by inverse quantization followed by inverse transformation.

Coding based on motion compensated prediction is very effective for removing temporal redundancy by predicting the value of pixels in the frame to be encoded on the basis of past and/or future reference frames. Motion compensated prediction is performed at the macroblock level, typically 16×16 pixels. The prediction residual for the macroblock is computed, quantized for coding, and transmitted along with the motion information necessary for the decoder to reconstruct the macroblock.

Hybrid coding brings these two techniques together, gaining the best of both worlds. The hybrid coding approach allows coding of intra, predicted and bidirectionally predicted frames in a single coder structure.

Macroblocks and blocks

Frames are partitioned into macroblocks of 16×16 pixels which in turn consist of blocks of 8×8 pixels. For example, a 4:2:0 color sub-sampled macroblock has size 16×16 pixels and consists of four 8×8 pixel luma blocks and two 8×8 chroma blocks. The relationship between a macroblock and the color component blocks which constitute the macroblock is shown in Figure 18 for the three commonly used color sub-sampling schemes.

Macroblock coding

A macroblock may be either transform coded or predictively coded depending on whether the frame is to be intra-coded (I) or inter-coded (P or B).

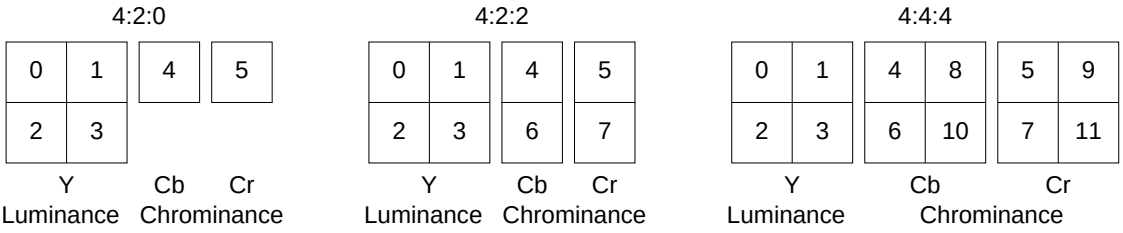

Figure 18: Macroblock/block structure for 4:2:0, 4:2:2 and 4:4:4 color sub-sampling.

In the case of I frames, only intra-coding is allowed, so each block which comprises the macroblock is DCT transformed, quantized and entropy coded for transmission. The block values entering the transform are pixel luma and chroma values.

In inter-coded (P or B) frames, macroblocks can be encoded using motion compensated prediction or they may be intra-coded. For example, it may be cheaper (from a bit-rate standpoint) to simply intra-code a macroblock than to code the motion compensated prediction residual and motion information for the macroblock.

When a macroblock is to be predictively coded, motion estimation is used to find a matching region (possibly two regions in the case of interpolated B frames) in the past and/or future reference frames which may be used to predict the current macroblock intensity values. The prediction residual (the displaced macroblock difference) is the error between the macroblock values and the prediction. In motion compensated predictive coding this prediction residual is quantized and coded for transmission. In a hybrid coder, however, rather than transmitting the prediction residual directly, the DCT is applied to the prediction residual, followed by quantization and entropy coding.

Quantization and coding of the transform coefficients

The DCT coefficient distributions resulting from intra-coded and inter-coded macroblocks are markedly dissimilar. As a result, different quantizer thresholds are used for each case.

In the case of intra-coded blocks, the result of DCT transformation and quantization is the concentration of energy in a small number of low frequency coefficients. A zig-zag scan of the 8×8 block of coefficients is applied as illustrated in Figure 19. The zig-zag scan is designed to scan the important low frequencies first so as to increase the chances of long runs of zeros at higher spatial frequencies. Since only the non-zero coefficients need to be coded, runs of zeros between non-zero values may be efficiently signaled using a run-length approach [38].

Since the local spatial average in an image varies slowly, the DC coefficient (the transform coefficient corresponding to zero horizontal and vertical spatial frequency) for intra-coded blocks is usually differentially coded with respect to an earlier intra-coded block. The AC coefficients (transform coefficients of non-zero horizontal or vertical spatial frequencies) are quantized and entropy coded as usual. In some video compression standards, selected AC coefficients may also be differentially coded.

For inter-coded macroblocks, scanning patterns appropriate for the statistical distribution of the inter-coded macroblock transform coefficients are used [38, 14, 12, 13].

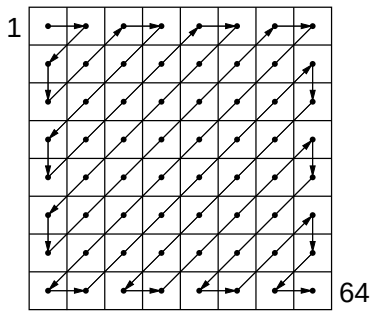

Figure 19: Zig-zag scanning pattern for DCT coefficients.

Hybrid codec architecture

In Figure 20 a typical hybrid motion compensated prediction/transform encoder architecture is shown. The input to the encoder consists of an image sequence which has been reordered so that I-frames and P-frames used for predicting B-frames may be encoded and transmitted first (see Figure 17 for an example). The frame type (I/B/P) for each frame is assumed to be predetermined and known to the encoder.

The structure of the encoder builds on what was presented previously for motion compensated predictive coding, and transform coding. Notice that the encoding loop includes a discrete cosine transform followed by quantization. When the encoder operates in intra-frame mode, the DCT is applied to actual pixel values, whereas in inter-frame prediction modes (predicted or bidirectionally predicted macroblocks) the DCT is performed on the prediction residual. The resulting DCT coefficient probability distributions are markedly different, so the quantizer characteristics are controlled according to whether the macroblock is intra-coded or inter-coded.

Since the operations of the DCT and quantization occur in the forward path of the encoder, the inverse operations are required to reconstruct the encoder output as it would appear after decoding. This is the reason for including the inverse quantization and inverse DCT operations in the feedback path of the encoding loop.

For bidirectional coding, both past and future reference frame stores are required. Additionally, for I-frames and P-frames, the reconstructed frame is written to the future frame store, and the contents of the future frame store are moved to the past frame store.

Also of note is the presence of quantization adaptation which is used to control the output bit-rate. By modifying the coarseness of the quantizer, the output bit-rate of the encoder can be controlled. Overflow and underflow of the transmit buffer may also be avoided in this way. The quantizer characteristics are usually controlled via a coding statistics processor (not shown) and are influenced by the content of the input video sequence. The design of buffer and bit-rate control mechanisms is an active research area [39, 40, 41, 42, 43, 44, 38].

The VLC coder is responsible for multiplexing and assigning VLC codewords to the symbols representing the quantized DCT coefficients, motion vector information and various other side information. The output from the VLC coder is then buffered to allow constant bit-rate transmission.

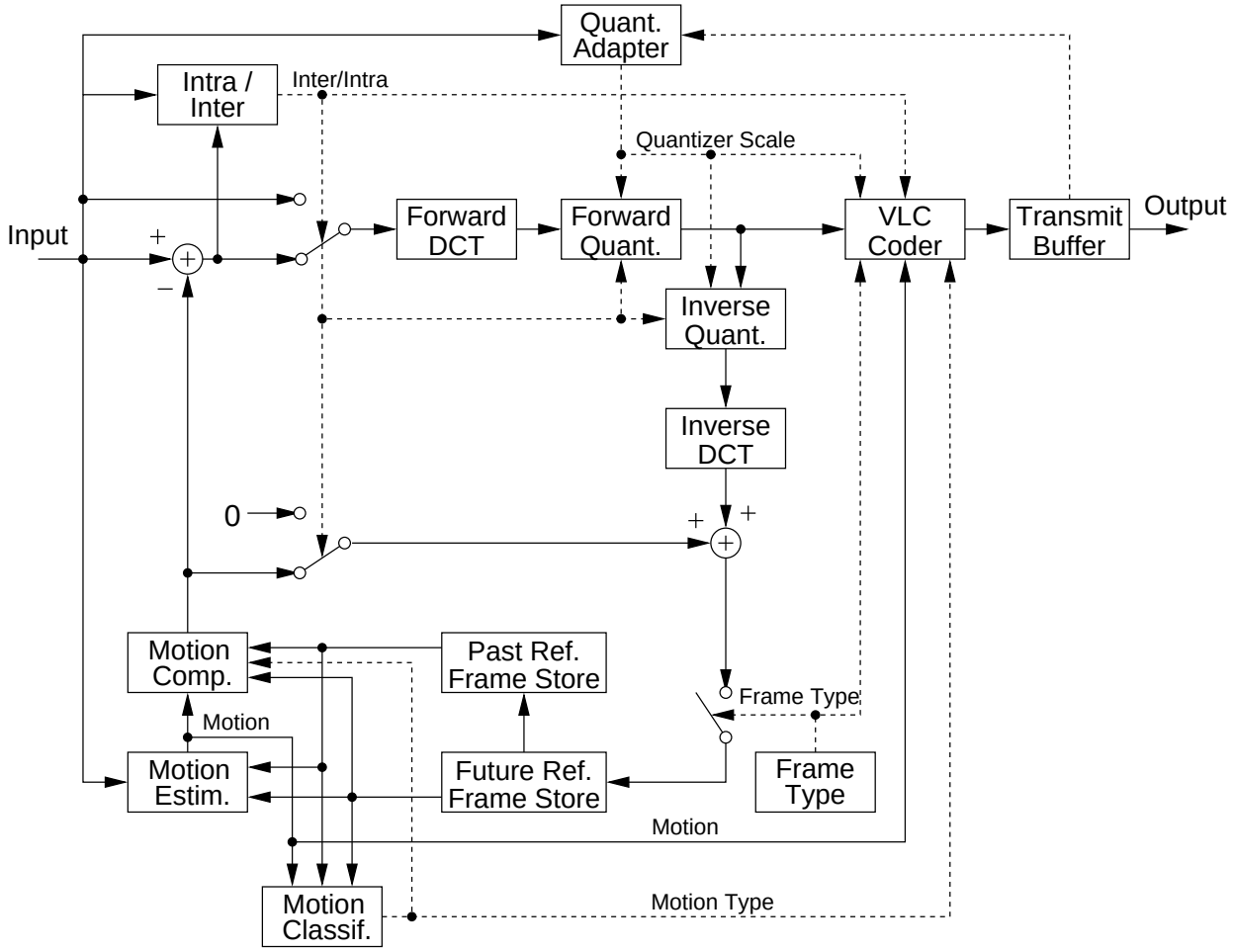

Figure 20: Hybrid motion compensated prediction / transform encoder.

The decoder structure is shown in Figure 21. At first glance it is clear that the decoder is far less complex than the encoder. Computationally the decoder has much less work to do than the encoder as no expensive motion estimation is necessary. The most expensive tasks at the decoder are motion compensated prediction based on the two stored reference frames and inverse DCT computation.

1.4.6 Video Compression Standards

Several popular international standards for video compression exist, including the ITU-T H.26x sequence and the MPEG series. These standards are all based on hybrid coding schemes similar to the one described in the previous section. All utilize macroblocks for motion compensation and block-based DCTs for coding. They tend to differ in terms of their design objectives, for example, target bit-rate and application, as well as the feature set they provide. For example the MPEG-1 standard was primarily designed for coding at around 1.5 Mbps for CDROM applications and does not support encoding of interlaced

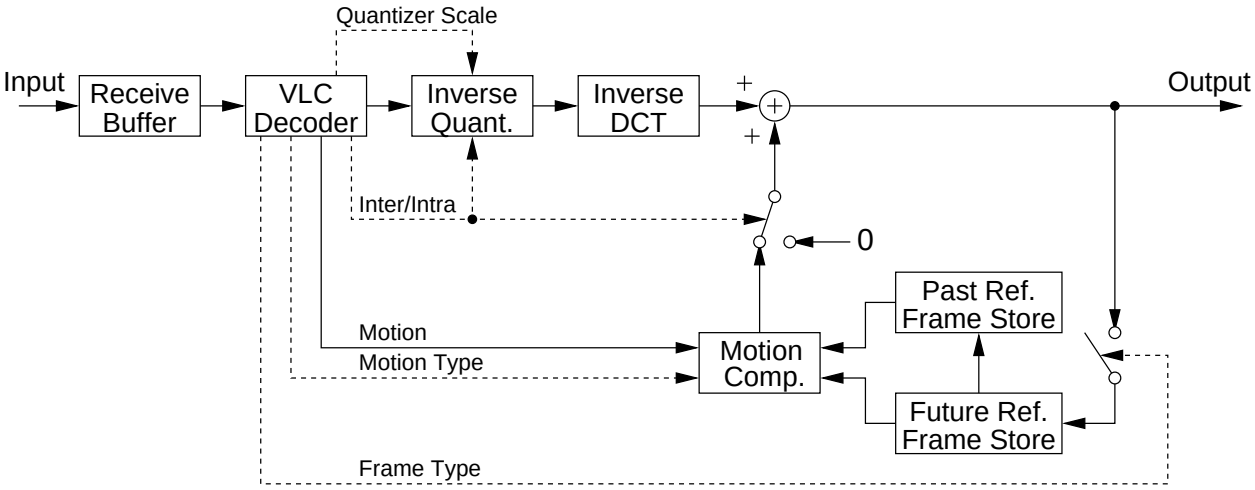

Figure 21: Hybrid motion compensated prediction / transform decoder.

source material. MPEG-2 is better suited to higher bit-rate applications (e.g. for digital cable and satellite television, high-definition television and DVD video) and has numerous extensions for efficient coding of interlaced video. The new MPEG-4 standard is optimized for very low bit-rate coding and includes many extensions to enable efficient coding using, for example, mesh-based representations, face coding, and object-based coding as well as the more traditional block-based coding. The ITU-T H.261 standard is primarily designed for low bit-rate coding for teleconferencing applications at $p \times 64$ kbps rates. The H.262 standard is virtually identical to MPEG-2, while H.263 is designed for low bit-rate communications. The emerging H.26L standard provides a number of improvements over previous standards, including significant changes to the way in which data are encoded, increased flexibility with respect to partitioning data into coded blocks, and a wider range of permissible reference frames.

Full documentation for these standards is available from standards bodies such as the ISO, ANSI or ITU. Full discussions on these standards may be found in [38, 14, 12, 13].

1.4.7 Other Compression Approaches

In this final section brief mention is made of alternate strategies which have been applied to image and video coding.

Object-based video coding

In object-based video coding, visual information is represented in terms of textured regions which are chosen to better approximate objects in the scene. This is a considerable departure from the techniques described earlier which partition images into fixed sized blocks, making little or no assumption about the presence of objects within the scene. Object-based approaches are therefore concerned with object representation as well as coding efficiency which tends to follow as a natural consequence.

Object-based methods need to represent both the shape and the texture of objects. In order to ascertain shape, segmentation is necessary. As for the representation of shape, adaptive mesh models have found widespread use [45, 46]. Efficient texture representation for triangular mesh models has been achieved using modifications to the basic block transform coding technique which allow for shape adaptive coding [47].

Object-based coding has been found to be highly effective in very low bit-rate applications and plays a central role in the MPEG-4 standard. Additionally, an object-based representation provides enormous flexibility regarding presentation and manipulation of video objects thus providing the potential for interactivity and customization of the presentation by the viewer. For a full treatment of object-based coding methods see [48].

Object-based coding methods must still balance the cost of coding the prediction residual against the cost of coding the motion vector and shape information for each object. The gains in coding efficiency achieved using an object-based representation can be overwhelmed by the cost of coding the object's texture, shape and motion. This is especially true in very low bit-rate applications.

Wavelets and sub-band coding

The wavelet transform is an efficient method for finding a multi-scale or multi-resolution representation of signals or images. Since the wavelet transform can be implemented using an analysis filter bank which decomposes the original image into sub-bands, wavelet and sub-band decomposition techniques are often treated as one. These sub-bands may be down-sampled to yield a complete transform domain representation that has the same dimensions as the original data. The sub-bands form a hierarchy beginning with a low-pass version of the signal or image, with each additional sub-band containing successively more "detail" information. This representation is very well suited to compression applications where the coefficients of each sub-band, which represent information at a specific scale, may be quantized and coded according to the available bit-rate allocation.

In the JPEG2000 standard, the wavelet decomposition is applied to image "tiles" which have dimensions which are typically powers of two, except for those tiles on the image boundaries. Tiles are usually much larger than the 8×8 pixel blocks typical for DCT block transform coding. The coarse quantization of the wavelet coefficients required for high compression tends to result in blurry images rather than the blocky images associated with block transform techniques. Though wavelet-based representations are extremely well suited to still image compression applications, the extension to video coding is less natural. One approach is to use the wavelet transformation in the traditional hybrid coder as a spatial transformation for energy compaction. Another is to use the wavelet as a 3-D transformation on the spatio-temporal data. For further details on the theory and application of wavelets to image coding, see the seminal papers [49, 50] and for video compression see [51, 52]. A classic introduction to wavelet theory is found in [53].

Vector quantization

Techniques based on vector quantization (VQ) have found application in signal and image compression. VQ is a generalization of scalar quantization to higher dimensions, based on

decision regions and a codebook as compared with decision levels and reconstruction levels in the case of scalar quantization. With VQ, since a single codeword represents a vector of values, the potential for compression is great. The challenge is in the design of efficient codebooks given the data to be represented. There are several ways in which VQ is applied in image coding.

VQ may be directly applied to image compression by partitioning the image into blocks, constructing an appropriate codebook for the image and then coding accordingly. This direct approach is complicated by the fact that a large codebook is often needed, thus limiting performance. Residual VQ addresses this by normalizing blocks to have zero mean and unit variance before the VQ step. Related to this approach is the idea of block truncation coding [54]. VQ may also be applied in the transform domain for efficient coding of the transform domain coefficients. Several other approaches have also been proposed. For an introduction to the application of VQ to image compression, see [12] and for detailed theory on VQ including algorithms for codebook selection see [35].

Fractal image compression

Fractal-based compression methods rely on the presence of self-similar structure in images. Self similar figures may be very compactly represented using an iterated function system (IFS) which is essentially a system consisting of repeated applications of a geometric transformation on the data. A suitably defined IFS will have a stationary point or *attractor*. The basic notion of fractal-based coding is to represent image spatial variation by determining iterated function systems which have that spatial variation as the attractor of the IFS. For more details on fractal compression see [55].

1.5 MULTI-FRAME RESTORATION

1.5.1 Introduction

In this section modern multi-frame image restoration algorithms which may be applied to video applications are discussed.[1] In an earlier chapter, classical image restoration algorithms were introduced. The material presented here builds on these results, but takes advantage of the additional information available from a sequence of images. In recent years, growing interest in *super-resolution* restoration of video sequences and the closely related problem of constructing super-resolution still images from image sequences has led to the emergence of several competing methodologies. The state of the art of super-resolution restoration techniques is reviewed, using a taxonomy of existing approaches. For an overview of other multi-frame restoration techniques, see [57].

1.5.2 Super-Resolution Restoration

The problem of spatial resolution enhancement of video sequences has been an area of active research since the seminal work by Tsai and Huang [58] which considers the problem of resolution enhanced stills from a sequence of low-resolution images of a translated scene. Whereas in the traditional single image restoration problem only a single input image is available, the task of obtaining a super-resolved image from an under-sampled and degraded image sequence can take advantage of the additional spatio-temporal data available in the image sequence. In particular, camera and scene motion lead to frames in the video sequence containing similar, but not identical information. This additional information content, as well as the inclusion of a-priori constraints, enables restoration of a super-resolved image with wider bandwidth than that of any of the individual low-resolution frames.

Much of the super-resolution literature addresses the problem of producing super-resolution still images from a video sequence – several low-resolution frames are combined to produce a single super-resolution frame. These techniques may be applied to video restoration by computing successive super-resolution frames from a "sliding window" of low-resolution frames.

Super-resolution restoration is an example of an *ill-posed* inverse problem. Such problems may be tackled by constraining the solution space according to a-priori knowledge of the form of the solution (smoothness, positivity etc.). Inclusion of such constraints is essential for achieving high quality restoration.

Super-resolution restoration methods may be categorized into two main divisions – frequency domain and spatial domain techniques.

1.5.3 Frequency Domain Restoration Methods

One broad class of restoration methods utilizes a frequency domain formulation of the super-resolution restoration problem. These frequency domain methods are based on three fundamental principles: (i) the shifting property of the Fourier transform (FT), (ii) the aliasing

[1] Part of the material presented in this section is based on, "Super-Resolution from Image Sequences – A Review" by Sean Borman and Robert L. Stevenson which appeared in the Proceedings of the 1998 Midwest Symposium on Circuits and Systems, Notre Dame, IN, USA [56].

relationship between the continuous Fourier transform (CFT) and the discrete Fourier transform (DFT), (iii) the original scene is band-limited. These properties allow the formulation of a system of equations relating the aliased DFT coefficients of the observed images to samples of the CFT of the unknown scene. These equations are solved yielding the frequency domain coefficients of the original scene, which may then be recovered by applying the inverse DFT. Formulation of the system of equations requires knowledge of the translational motion between frames to sub-pixel accuracy. Each observed image must contribute *independent* constraint equations, which places restrictions on the inter-frame motion that contributes useful data.

Observation model and solution

Denote the continuous scene by $f(x,y)$. Global translations yield R shifted images, $f_r(x,y)=f(x+\Delta x_r, y+\Delta y_r),\ r=1,2,\ldots,R$. The CFT of the scene is given by $\mathcal{F}(u,v)$ and that of the translations by $\mathcal{F}_r(u,v)$. The shifted images are impulse sampled to yield observed images $y_r[m,n]=f(mT_x+\Delta x_r, nT_y+\Delta y_r)$ with $m=0,1,\ldots,M-1$ and $n=0,1,\ldots,N-1$. The R corresponding 2D DFT's are denoted $\mathcal{Y}_r[k,l]$. The CFT of the scene and the DFT's of the shifted and sampled images are related via aliasing,

$$\mathcal{Y}_r[k,l]=\frac{1}{T_xT_y}\sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty}\mathcal{F}_r\left(\frac{k}{MT_x}+pf_{s_x},\frac{l}{NT_y}+qf_{s_y}\right) \tag{1.30}$$

where $f_{s_x}=1/T_x$ and $f_{s_y}=1/T_y$ are the sampling rates in the x and y dimensions respectively. The shifting property of the CFT relates spatial domain translation to frequency domain phase shifting as,

$$\mathcal{F}_r(u,v)=e^{j2\pi(\Delta x_r u+\Delta y_r v)}\mathcal{F}(u,v). \tag{1.31}$$

If $f(x,y)$ is band-limited, $\exists\ L_u, L_v$ s.t. $\mathcal{F}(u,v)=0$ for $|u|\geq L_u f_{s_x}$ and $|v|\geq L_v f_{s_y}$. Assuming $f(x,y)$ is band-limited, equation (1.31) may be used to rewrite the alias relationship in equation (1.30) in matrix form as,

$$\mathbf{Y}=\mathbf{\Phi}\mathbf{F}. \tag{1.32}$$

$\mathbf{Y}$ is a R-dimensional vector with the r^{th} element being the DFT coefficients $\mathcal{Y}_r[k,l]$ of the observed image $y_r[m,n]$. $\mathbf{\Phi}$ is a matrix which relates the DFT of the observation data to samples of the unknown CFT of $f(x,y)$ contained in the $4L_uL_v\times 1$ vector $\mathbf{F}$. Super-resolution restoration thus requires finding the DFT's of the R observed images, determining $\mathbf{\Phi}$ (motion estimation), solving the system of equations (1.32) for $\mathbf{F}$ and applying the inverse DFT to obtain the reconstructed image.

Extensions of the frequency-domain method

Several extensions to the basic Tsai-Huang method have been proposed. A linear, shift-invariant (LSI) blur point spread function (PSF) is included in [59] and the equivalent of equation (1.32) is solved using a least squares approach to mitigate the effects of observation noise and insufficient observation data. A computationally efficient recursive least squares (RLS) solution for equation (1.32) is proposed in [60] and extended with a Tikhonov

regularization solution method in [61] in an attempt to address the ill-posedness of the super-resolution restoration inverse problem. Robustness to errors in observations as well as $\mathbf{\Phi}$ motivated the use of a total least squares (TLS) approach in [62] which is implemented using a recursive algorithm for computational efficiency.

Methods utilizing the multi-channel sampling theorem

Techniques based on the multi-channel sampling theorem [63] have also been considered [64]. Though implemented in the spatial domain, the technique is fundamentally a frequency domain method relying on the shift property of the Fourier transform to model the translation of the source imagery.

Summary

Frequency domain super-resolution methods provide the advantages of theoretical simplicity and low computational complexity, are highly amenable to parallel implementation due to decoupling of the frequency domain equation (1.32) and exhibit an intuitive dealiasing super-resolution mechanism. Disadvantages include the limitation to global translational motion and space invariant degradation models (necessitated by the requirement for a Fourier domain analogue of the spatial domain motion and degradation model) and limited ability for inclusion of spatial domain a-priori knowledge for regularization.

1.5.4 Spatial Domain Restoration Methods

In this class of super-resolution restoration methods, the observation model is formulated, and restoration is effected in the *spatial domain*. The linear spatial domain observation model can accommodate global and non-global motion, optical blur, motion blur, spatially-varying point spread functions, non-ideal sampling, compression artifacts and other degradations. The spatial domain restoration formulation is inherently amenable to the inclusion of (possibly nonlinear) spatial domain a-priori constraints (e.g. Markov random fields or convex sets) which may be used to encourage bandwidth extrapolation in restoration.

Observation model

Consider estimating a super-resolution image $\mathbf{z}$ from multiple low-resolution images $\mathbf{y_r},\ r \in \{1, 2, \ldots, R\}$. Images are written as lexicographically ordered vectors. $\mathbf{y_r}$ and $\mathbf{z}$ are related as $\mathbf{y_r} = \mathbf{H_r z}$. The matrix $\mathbf{H_r}$, which must be estimated, incorporates motion compensation, degradation effects and sub-sampling. The observation equation may be generalized to $\mathbf{Y} = \mathbf{Hz} + \mathbf{N}$ where $\mathbf{Y} = \left[\mathbf{y_1}^T \cdots \mathbf{y_R}^T\right]^T$ and $\mathbf{H} = \left[\mathbf{H_1}^T \cdots \mathbf{H_R}^T\right]^T$ with $\mathbf{N}$ representing observation noise.

Interpolation of non-uniformly spaced samples

Registering a set of low-resolution images using motion compensation results in a single, dense composite image of non-uniformly spaced samples. A restored image may be derived from this composite image using techniques for reconstruction from non-uniformly

spaced samples. Restoration techniques are sometimes applied to compensate for degradations [59]. Iterative restoration techniques, based on the Landweber iteration, have also been applied [65]. Such interpolation methods are unfortunately overly simplistic. Since the observed data result from severely under-sampled, spatially averaged areas, the restoration step (which typically assumes impulse sampling) is incapable of reconstructing significantly more frequency content than is present in a single low-resolution frame. Degradation models are limited, and no a-priori constraints are used. There is also some question as to the optimality of separate merging and restoration steps.

Iterated backprojection

Given a super-resolution estimate $\hat{\mathbf{z}}$ and the imaging model $\mathbf{H}$, it is possible to *simulate* the low-resolution images $\hat{\mathbf{Y}}$ as $\hat{\mathbf{Y}} = \mathbf{H}\hat{\mathbf{z}}$. Iterated back-projection (IBP) procedures update the estimate of the super-resolution restoration by *back-projecting* the error between the j^{th} iteration of the simulated low-resolution images $\hat{\mathbf{Y}}^{(j)}$ and the observed low-resolution images $\mathbf{Y}$ via the back-projection operator $\mathbf{H}^{BP}$ which apportions "blame" to pixels in the super-resolution estimate $\hat{\mathbf{z}}^{(j)}$. Typically $\mathbf{H}^{BP}$ approximates $\mathbf{H}^{-1}$. Algebraically,

$$\begin{aligned} \hat{\mathbf{z}}^{(j+1)} &= \hat{\mathbf{z}}^{(j)} + \mathbf{H}^{BP}\left(\mathbf{Y} - \hat{\mathbf{Y}}^{(j)}\right) \\ &= \hat{\mathbf{z}}^{(j)} + \mathbf{H}^{BP}\left(\mathbf{Y} - \mathbf{H}\hat{\mathbf{z}}^{(j)}\right). \end{aligned} \tag{1.33}$$

Equation (1.33) is iterated until some error criterion dependent on $\mathbf{Y}$ and $\hat{\mathbf{Y}}^{(j)}$ is minimized. An application of the IBP method may be found in [66]. IBP enforces the constraint that the super-resolution restoration be consistent with the observed data (via the observation equation). Unfortunately the solution is often not unique since super-resolution is usually an ill-posed inverse problem. Unfortunately the inclusion of a-priori constraints is cumbersome in the IBP framework.

Stochastic restoration methods

Stochastic methods (Bayesian methods in particular) which treat super-resolution restoration as a statistical inference problem have rapidly gained prominence since these methods provide a powerful theoretical framework for the inclusion of a-priori constraints necessary for satisfactory solution of the ill-posed super-resolution restoration inverse problem. The observed data $\mathbf{Y}$, noise $\mathbf{N}$ and super-resolution image $\mathbf{z}$ are assumed stochastic. Consider now the *stochastic* observation equation $\mathbf{Y} = \mathbf{H}\mathbf{z} + \mathbf{N}$. The *maximum a-posteriori probability* (MAP) approach to estimating $\mathbf{z}$ seeks the estimate $\hat{\mathbf{z}}_{\text{MAP}}$ for which the a-posteriori probability, $\Pr\{\mathbf{z}|\mathbf{Y}\}$ is a maximum. Formally, the objective is to find $\hat{\mathbf{z}}_{\text{MAP}}$ such that,

$$\begin{aligned} \hat{\mathbf{z}}_{\text{MAP}} &= \arg\max_{\mathbf{z}} \left[\Pr\{\mathbf{z}|\mathbf{Y}\}\right] \\ &= \arg\max_{\mathbf{z}} \left[\log \Pr\{\mathbf{Y}|\mathbf{z}\} + \log \Pr\{\mathbf{z}\}\right]. \end{aligned} \tag{1.34}$$

The second line is found by applying Bayes' rule, recognizing that $\hat{\mathbf{z}}_{\text{MAP}}$ is independent of $\Pr\{\mathbf{Y}\}$ and taking logarithms. The term $\log \Pr\{\mathbf{Y}|\mathbf{z}\}$ is the *log-likelihood function* and $\Pr\{\mathbf{z}\}$ is the *a-priori density* of $\mathbf{z}$. Since $\mathbf{Y} = \mathbf{H}\mathbf{z} + \mathbf{N}$, the likelihood function is determined

by the pdf of the noise as $\Pr\{\mathbf{Y}|\mathbf{z}\} = f_{\mathbf{N}}(\mathbf{Y} - \mathbf{H}\mathbf{z})$. It is common to utilize Markov random field (MRF) image models as the prior term $\Pr\{\mathbf{z}\}$. Under typical assumptions of Gaussian noise the prior may be chosen to ensure a convex optimization in equation (1.34) enabling the use of gradient descent optimization procedures. Examples of the application of Bayesian methods to super-resolution restoration may be found in [67] using a Huber MRF and [68, 69] with a Gaussian MRF.

Maximum likelihood (ML) estimation has also been applied to super-resolution restoration [70]. ML estimation is a special case of MAP estimation with a uniform prior or no prior term at all. Since the inclusion of a-priori information is essential for the solution of ill-posed inverse problems, MAP estimation should be used in preference to ML.

A major advantage of the Bayesian framework is the direct inclusion of a-priori constraints on the solution, often as MRF priors which provide a powerful method for image modeling using (possibly non-linear) local neighbor interaction. MAP estimation with convex priors implies a globally convex optimization, ensuring solution existence and uniqueness allowing the application of efficient descent optimization methods. Simultaneous motion estimation and restoration is also possible [69]. The rich area of statistical estimation theory is directly applicable to stochastic super-resolution restoration methods.

Set-theoretic restoration methods

Set theoretic methods, especially the method of projection onto convex sets (POCS), are popular as they are simple, utilize the powerful spatial domain observation model, and allow convenient inclusion of *a priori* information. In set theoretic methods, the space of super-resolution solution images is intersected with a set of (typically convex) constraint sets representing desirable super-resolution image characteristics such as positivity, bounded energy, fidelity to data, smoothness etc., to yield a reduced solution space. POCS refers to an iterative procedure which, given any point in the space of super-resolution images, locates a point which satisfies all the convex constraint sets.

Convex sets which represent constraints on the solution space of $\mathbf{z}$ are defined. Data consistency is typically represented by a set $\{\mathbf{z} : |\mathbf{Y} - \mathbf{H}\mathbf{z}| < \delta_0\}$, positivity by $\{\mathbf{z} : z_i > 0 \ \forall i\}$, bounded energy by $\{\mathbf{z} : \|\mathbf{z}\| \leq E\}$, compact support $\{\mathbf{z} : z_i = 0, i \in \mathcal{A}\}$ and so on. For each convex constraint set so defined, a *projection operator* is determined. The projection operator $\mathcal{P}_\alpha$ associated with the constraint set $\mathcal{C}_\alpha$ projects a point in the space of $\mathbf{z}$ onto the closest point on the surface of $\mathcal{C}_\alpha$. It can be shown that repeated application of the iteration, $\mathbf{z}^{(n+1)} = \mathcal{P}_1\mathcal{P}_2\mathcal{P}_3\cdots\mathcal{P}_K\mathbf{z}^{(n)}$ will result in convergence to a solution on the surface of the intersection of the K convex constraints sets. Note that this point is in general non-unique and is dependent on the initial guess. POCS restoration methods have been successfully applied to sophisticated observation and degradation models [71, 72].

An alternate set theoretic super-resolution restoration method [73] uses an ellipsoid to bound the constraint sets. The centroid of this ellipsoid is taken as the super-resolution estimate. Since direct computation of this point is infeasible, an iterative solution method is used.

The advantages of set theoretic super-resolution restoration techniques were discussed at the beginning of this section. These methods have the disadvantages of non-uniqueness of

solution, dependence of the solution on the initial guess, slow convergence and high computational cost. Though the bounding ellipsoid method ensures a unique solution, this solution is has no claim to optimality.

Hybrid methods

Work has been undertaken on combined ML/MAP/POCS-based approaches to super-resolution restoration [67, 74]. The desirable characteristics of stochastic estimation and POCS are combined in a hybrid optimization method. The a-posteriori density or likelihood function is maximized subject to containment of the solution in the intersection of the convex constraint sets.

Optimal and adaptive filtering methods

Inverse filtering approaches to super-resolution restoration have been proposed, but these techniques are limited in terms of inclusion of a-priori constraints as compared with POCS or Bayesian methods and are mentioned only for completeness. Techniques based on adaptive filtering, have also seen application in super-resolution restoration [75, 76]. These methods are in effect LMMSE estimators and do not include non-linear a-priori constraints.

Regularization-based methods

Due the the ill-posedness of super-resolution restoration, Tikhonov regularized super-resolution restoration methods have been examined [77]. The regularizing functionals characteristic of this approach are typically special cases of MRF priors in the Bayesian framework.

1.5.5 Summary and Comparisons

A general comparison of frequency and spatial domain super-resolution restoration methods is presented in Table 1.1.

Spatial domain super-resolution restoration methods, though computationally more expensive, and more complex than their frequency domain counterparts, offer important advantages in terms of flexibility. Two powerful classes of spatial domain methods; the Bayesian (MAP) approach and the set theoretic POCS methods are compared in Table 1.2.

Super-resolution restoration is critically dependent on accurate, sub-pixel motion estimation. Restoration is sensitively dependent on the observation model which takes into account the motion occurring in the video sequence. Sub-pixel relative motion in multiple frames contributes novel information which constrains the super-resolution inverse problem solution space. The problem of motion estimation is complicated by the fact the motion must be estimated from the observed *undersampled* data. General motion models can only be accommodated by spatial domain restoration methods. Model-based, multiple independent motion estimation and tracking allows super-resolution of objects subject to partial occlusion, transparency, motion, etc.

Accurate degradation modeling typically results in improved restorations. This includes, for example, modeling of color correlations [78] and degradation models for lossy compression

	Frequency Domain	**Spatial Domain**
Observation model	Frequency domain	Spatial domain
Motion models	Global translation	Almost unlimited
Degradation model	Limited, LSI	LSI or LSV
Noise model	Limited, SI	Very flexible
SR mechanism	Dealiasing	Dealiasing, a-priori info
Computation requirement	Low	High
A-priori info	Limited	Almost unlimited
Regularization	Limited	Excellent
Extensibility	Poor	Excellent
Applicability	Limited	Wide
Application performance	Good	Good

Table 1.1: Frequency vs. spatial domain super-resolution.

schemes [79]. Similarly, modeling of degradations inherent in magnetic media recording and playback is expected to improve super-resolution restoration from low cost camcorder data. The response of typical commercial CCD arrays departs considerably from the simple integrate and sample model prevalent in much of the literature. Better modeling of these devices promises performance improvements.

1.5.6 Examples

In this section images restored using a Bayesian super-resolution video restoration approach [80] are presented. A low resolution image sequence was captured with a consumer grade camcorder which pans across an architectural scene. The original images are 160×120 pixels in size. The output video frames have dimension 640×480 pixels – a scaling factor of 4 in each spatial dimension. Thus each output frame contains 16 times the number of pixels (unknowns to be estimated) as the number of pixels in each of the original images. A small selection of the original low-resolution frames is shown in Figure 22. The motion occurring is the result of a camera pan upward and to the right.

Figure 22: Original low resolution frames.

	Bayesian (MAP)	**POCS**
Applicable theory	Vast	Limited
A-priori info	Prior pdf Easy to incorporate No hard constraints	Convex sets Easy to incorporate Powerful yet simple
SR solution	Unique MAP estimate	Non-unique $\cap$ of constraint sets
Optimization	Iterative	Iterative
Convergence	Good	Possibly slow
Computational requirement	High	High
Complications	Optimization under non-convex priors	Definition of projection operators

Table 1.2: MAP vs. POCS super-resolution.

Figure 23 illustrates the result of cubic spline interpolation of one of the original images (second from the left, bottom row, in Figure 22) to yield an expanded image with dimension 640×480 pixels. Cubic spline interpolation is a commonly used technique for image expansion even though it tends to result in over-smoothed images at high magnification. Severe smoothing is evident in Figure 23.

The result of super-resolution restoration of the same original low-resolution frame is shown in Figure 24. Details are much sharper as compared with the cubic spline interpolated image. To aid in comparison, a region of interest from the cubic spline interpolated image and the same region in the super-resolution restoration are presented side by side in Figure 25.

Figure 23: Cubic spline interpolated image.

Figure 24: Super-resolution restoration.

Figure 25: Region of interest comparison between the cubic spline interpolated image (left) and super-resolution restoration (right).

1.6 FURTHER TECHNIQUES AND APPLICATIONS

The problem of motion estimation, which lies at the heart of image sequence processing, has been discussed in some detail. This was followed by a presentation on video compression – from a practical standpoint, perhaps the single most important application of image sequence processing techniques. Modern video restoration methods which enhance spatial resolution and which build on results from classical single image restoration were examined. In this section, other applications which fall into the category of image sequence processing are briefly introduced without going into detail.

1.6.1 Image Sequence Interpolation

Given a sequence of video frames, the problem of *image sequence interpolation* involves the estimation of the image intensity at time instants *between* the observed frames. Since motion compensation techniques are typically applied, the problem is also known as *motion compensated interpolation*.

An example of the application of image sequence interpolation is in videophones where tight bandwidth restrictions typically require that at most 15 compressed frames can be coded per second, leading to jerky image motion at the receiver. If images can be estimated at the time instants between the received images, a higher frame rate can be produced, thereby mitigating the effects of jerky motion. Since the interpolated frames are computed solely on the basis of the received frames, it is not necessary to increase the amount of information sent to the receiver. This makes image interpolation an attractive method for improving the perceived video quality.

Motion compensated interpolation is challenging due to the problems of accelerated motion and object occlusion/disocclusion. See [81] for an introduction to image interpolation and especially the occlusion/disocclusion problem, and [17] for details associated with accelerated motion.

1.6.2 Standards Conversion and Deinterlacing

A problem related to image sequence interpolation discussed above is *standards conversion*. The term is used to refer to methods used to convert between differing spatio-temporal sampling lattices. One of the major applications is the conversion of material between incompatible television standards, for example PAL and NTSC which differ both in spatial as well as temporal sampling resolutions. The conversion of motion picture source material to video broadcast standards is another important application area. Though film material may be easily sampled spatially, the original frame rate is typically incompatible with the major broadcast standards.

An overview of the traditional standards conversion methodologies which are in wide commercial use may be found in [9, 14]. These include field/frame conversion methods for interlaced/non-interlaced standards conversion. For the most part these techniques are designed to be simple due to computational and implementation constraints. For a review of traditional deinterlacing techniques see [82].

More sophisticated approaches to standards conversion utilize full motion information as well as more elaborate observation models and reconstruction techniques. A discussion of these topics may be found in [83]. The deinterlacing problem too has seen the application of a broad range of methods. In [84], for example, a Bayesian approach for simultaneously deinterlacing and enhancing spatial resolution was proposed, while in [85] an approach based on projection onto convex sets (POCS) was presented.

1.6.3 Image Mosaicking

Related to the problem of super-resolution restoration discussed earlier, is that of image mosaicking. The objective in mosaicking is the creation of a large composite image which combines information from multiple frames. The composite image consists of images "stitched together" to form a single image with a field of view much larger than that of the individual frames. Examples of mosaicking approaches may be found in [86, 87]. Closely related is the problem of scene stabilization.

1.6.4 Post Processing of Compressed Video

A niche application which is useful for practical systems is post-processing of compressed image sequences. At low bit-rates the popular video compression standards all tend to exhibit visible blocking artifacts resulting from the coarse quantization of block transform coefficients required to achieve high compression. Post-processing techniques may be applied to the decoded video output with positive effects. These techniques typically use prior image models to smooth the blocking effects while still preserving image edge detail. A description of one such system may be found in [88].

1.6.5 Object Identification and Tracking

Motion detection, estimation and segmentation are often first steps toward the identification and tracking of objects. For computer vision applications where the goal is often automated interaction with objects in the scene, identification and tracking are essential. Achieving this typically requires high level representations of objects in the scene. In general these techniques fall into the broad category of computer vision. The related issue of automatic target recognition is discussed elsewhere in this volume.

1.6.6 Image and Motion Segmentation

In Section 1.3 the important inter-relationship of motion segmentation and motion estimation was discussed and an algorithm for simultaneous motion estimation and segmentation was presented. A wide range of techniques have been proposed for motion segmentation many of which are reviewed in [89]. Image segmentation techniques are also discussed in this encyclopedia. Motion and image segmentation are essential tools for obtaining object-based motion representations required for emerging video compression standards, as well as for video restoration and computer vision applications.

1.6.7 Structure from Motion

An interesting problem based on the tools of image sequence processing, especially on motion estimation is the problem of determining 3-D motion and structure from 2-D projected motion. This problem is often referred to as *structure from motion*. An introduction to this problem may be found in [15].

1.6.8 Indexing for Content Retrieval

Recently, effort has been expended in research and development of systems to enable fast indexing and content retrieval of images as well as video. This work is motivated by the expectation that vast image and video databases will become commonplace, perhaps even in the home. Access to specific content, be it text, image, audio or video becomes a problem when the size of multimedia databases is measured in terabytes. In these instances intelligent automated indexing and retrieval systems become essential. Common to all these systems are methods for classifying the content of the video information. These methods include broad statistical features of the data, image histograms, detection of scene and shot changes, motion content, object recognition (e.g. faces or captions) as well as audio cues. These features are used for classification and content searching and retrieval.

The emerging MPEG-7 standard plays a central role in media content description and retrieval. The standard is fundamentally concerned with "data about the data," so-called *metadata*. MPEG-7 is designed to provide a standardized method for describing multimedia content. As such it will provide a basis for the development of content retrieval systems. An introduction to these topics may be found in [90].

1.6.9 Video Watermarking

With digital video content fast becoming ubiquitous, concerns over authenticity and copyright protection have come to the fore. The ease with which digital content can be copied has led to intense work in the area of digital watermarking. The objective is the encoding of information within the data which may be used to confirm authenticity or signal copyright ownership. In general it is desirable that the encoded information be invisible to the user, but be robust against malicious attack. For further information on this fast growing field, see [91, 92, 93].

1.6.10 Motion Compensated Filtering

Image sequences provide the opportunity for the generalization of standard filtering techniques into both the spatial and the temporal dimensions. Improved performance is possible by designing filters which take advantage of the spatio-temporal Fourier domain structure of the video signal.

Under assumptions of global motion, it is possible to derive a spatio-temporal Fourier domain representation of the video signal [9]. This description may be used for the design of spatio-temporal filters which outperform simple intra-frame techniques. In the case of more general motion, motion compensated filtering is possible. In particular, filtering is

effected along motion trajectories. For further details on motion compensated filtering, see [9, 94, 17, 95]. Non-linear, 3-D median filtering techniques have also been investigated in the context of image sequences [22, 95].

1.7 ACKNOWLEDGMENTS

The authors wish to thank Kyle Erickson for producing the image of the discrete cosine transform basis functions in Figure 14 as well as for his editorial assistance and endless patience in reviewing drafts of this work.

Part of the material presented in Section 1.5 is based on, "Super-Resolution from Image Sequences – A Review" by Sean Borman and Robert L. Stevenson which appeared in the proceedings of the 1998 Midwest Symposium on Circuits and Systems, Notre Dame, IN, U.S.A. [56]. This material is © 1998 IEEE and is used with the kind permission of the IEEE.

Bibliography

[1] R. Haralick and L. Shapiro, *Computer and Robot Vision*, vol. II, ch. 13. Addison-Wesley, 1993.

[2] R. Mohr and B. Triggs, "Projective geometry for image analysis," in *International Symposium of Photogrammetry and Remote Sensing*, (Vienna), July 1996.

[3] T. Moons, "A guided tour through multiview relations," in *Proceedings SMILE Workshop*, vol. 1506 of *Lecture Notes in Computer Science*, pp. 304–346, Springer-Verlag, 1998.

[4] C. Boncelet, "Image noise models," in *Handbook of Image and Video Processing* (A. Bovik, ed.), ch. 4.5, pp. 325–335, Academic Press, 2000.

[5] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[6] D. Pritchard, "U.S. color television fundamentals – a review," *IEEE Transactions on Consumer Electronics*, vol. CE-23, pp. 467–478, Nov. 1977.

[7] C. Poynton, *A Technical Introduction to Digital Video*. John Wiley & Sons, 1996.

[8] D. Dudgeon and R. Mersereau, *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[9] A. Tekalp, *Digital Video Processing*. Prentice Hall, 1995.

[10] E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *IEEE Proceedings*, vol. 73, pp. 502–522, Apr. 1985.

[11] R. Gray and D. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, pp. 2325–2383, Oct. 1998.

[12] Y. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*. Boca Raton: CRC Press, 2000.

[13] K. Rao and J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*. Prentice Hall, 1996.

[14] B. Haskell, A. Puri, and A. Netravali, *Digital Video: An Introduction to MPEG-2*. Kluwer Academic Publishers, Nov. 1996.

[15] T. Jebara, A. Azarbayejani, and A. Pentland, "3D structure from 2D motion," *IEEE Signal Processing Magazine*, pp. 66–84, May 1999.

[16] A. Mitiche, *Computational Analysis of Visual Motion*. New York: Plenum Press, 1994.

[17] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "Digital video standards conversion in the presence of accelerated motion," *Signal Processing: Image Communication*, vol. 6, pp. 213–227, June 1994.

[18] P. Burt and G. Sperling, "Time, distance and feature trade-offs in visual apparent motion," *Psychological Review*, vol. 88, no. 2, pp. 171–195, 1981.

[19] Y. Weiss, *Bayesian motion estimation and segmentation*. PhD thesis, MIT, May 1998.

[20] J. Hadamard, *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. New Haven, CT: Yale University Press, 1923.

[21] C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Processing Magazine*, pp. 70–91, July 1999.

[22] M. Sezan and R. Lagendijk, eds., *Motion analysis and image sequence processing*. Kluwer Academic Publishers, 1993.

[23] S. Mann and R. Picard, "Video orbits of the projective group: A simple approach to featureless estimation of parameters," *IEEE Transactions on Image Processing*, vol. 6, pp. 1281–1295, Sept. 1997.

[24] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[25] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 565–593, Sept. 1986.

[26] H.-H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results," *Artificial Intelligence*, vol. 33, pp. 299–324, Nov. 1987.

[27] E. C. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intelligence*, vol. 23, pp. 309–354, Aug. 1984.

[28] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, June 1984.

[29] M. Chang, A. Tekalp, and M. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Transactions on Image Processing*, vol. 6, pp. 1326–1333, Sept. 1997.

[30] P. B. Chou and C. M. Brown, "The theory and practice of Bayesian image labeling," *International Journal of Computer Vision*, vol. 4, pp. 185–210, 1990.

[31] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, pp. 75–104, Jan. 1996.

[32] M. J. Black and A. D. Jepson, "Estimating optical flow in segmented images using variable-order parametric models with local deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 972–986, Oct. 1996.

[33] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Transactions on Image Processing*, vol. 3, pp. 625–638, Sept. 1994.

[34] P. Wintz, "Transform picture coding," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 809–820, 1972.

[35] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Jan. 1992.

[36] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

[37] R. Llados-Bernaus and R. Stevenson, "Fixed-length entropy coding for robust video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 745–755, Oct. 1998.

[38] A. Netravali and B. Haskell, *Digital Pictures: Representation and Compression and Standards*. Applications of Communications Theory, Plenum, 2 ed., 1995.

[39] A. Puri and R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 351–361, Dec. 1991.

[40] P. Westerink, R. Rajagopalan, and C. Gonzales, "Two-pass MPEG-2 variable-bit-rate encoding," *IBM Journal of Research and Development*, vol. 43, pp. 471–488, July 1999.

[41] J. I. Ronda, M. Eckert, F. Jaureguizar, and N. García, "Rate control and bit allocation for MPEG-4," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 1243–1258, Dec. 1999.

[42] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 172–185, Feb. 1999.

[43] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 186–199, Feb. 1999.

[44] D. Hoang and J. Vitter, *Efficient Algorithms for MPEG Video Compression*. John Wiley & Sons, 2001.

[45] Y. Wang and O. Lee, "Use of two-dimensional deformable mesh structures for video coding, part I — the synthesis problem: mesh-based function approximation and mapping," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 636–646, Dec. 1996.

[46] Y. Wang, O. Lee, and A. Vetro, "Use of two-dimensional deformable mesh structures for video coding, part II - the analysis problem and a region-based coder employing an active mesh representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 647–659, Dec. 1996.

[47] O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, "High-performance compression of visual information – a tutorial review– part I : Still pictures," *Proceedings of the IEEE*, vol. 87, p. 1999, June 1999.

[48] L. Torres and M. Kunt, eds., *Video Coding : The Second Generation Approach*. Kluwer Academic Publishers, 1996.

[49] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[50] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[51] J. W. Woods and T. Naveen, "Subband encoding of video sequences," in *Visual Communications and Image Processing IV*, vol. 1199 of *Proceedings of SPIE*, (Philadelphia, PA), pp. 724–732, Nov. 1989.

[52] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 8, pp. 155–167, Feb. 1999.

[53] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Philadelphia: Society for Industrial and Applied Mathematics, 1992.

[54] E. Delp and O. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, vol. 27, pp. 1335–1341, 1979.

[55] Y. Fisher, *Fractal Image Compression — Theory and Application*. New York: Springer-Verlag, 1994.

[56] S. Borman and R. Stevenson, "Super-resolution from image sequences – a review," in *Proceedings of the 1998 Midwest Symposium on Circuits and Systems*, (Notre Dame, IN, USA), pp. 374–378, IEEE, Aug. 1998.

[57] T. Schulz, "Multiframe image restoration," in *Handbook of Image and Video Processing* (A. Bovik, ed.), ch. 3.8, pp. 175–189, Academic Press, 2000.

[58] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," in *Advances in Computer Vision and Image Processing* (R. Y. Tsai and T. S. Huang, eds.), vol. 1, pp. 317–339, JAI Press Inc., 1984.

[59] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. III, (San Francisco), pp. 169–172, 1992.

[60] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 6, pp. 1013–1027, 1990.

[61] S. P. Kim and W.-Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images," *IEEE Transactions on Image Processing*, vol. 2, pp. 534–539, Oct. 1993.

[62] N. K. Bose, H. C. Kim, and H. M. Valenzuela, "Recursive total least squares algorithm for image reconstruction from noisy, undersampled multiframes," *Multidimensional Systems and Signal Processing*, vol. 4, pp. 253–268, July 1993.

[63] J. L. Brown, "Multichannel sampling of low-pass signals," *IEEE Transactions on Circuits and Systems*, vol. 28, no. 2, pp. 101–106, 1981.

[64] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graphical Models and Image Processing*, vol. 54, pp. 181–186, Mar. 1992.

[65] T. Komatsu, T. Igarashi, K. Aizawa, and T. Saito, "Very high resolution imaging scheme with multiple different aperture cameras," *Signal Processing: Image Communication*, vol. 5, pp. 511–526, Dec. 1993.

[66] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion and transparency," *Journal of Visual Communications and Image Representation*, vol. 4, pp. 324–335, Dec. 1993.

[67] R. Schultz and R. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol. 5, pp. 996–1011, June 1996.

[68] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," in *Maximum Entropy and Bayesian Methods*, pp. 293–308, Santa Barbara, CA: Kluwer, 1996.

[69] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint MAP registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1621–1633, Dec. 1997.

[70] B. C. Tom and A. K. Katsaggelos, "Reconstruction of a high resolution image from multiple degraded mis-registered low resolution images," in *SPIE Visual Communications and Image Processing*, vol. 2308, (Chicago), pp. 971–981, Sept. 1994.

[71] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Transactions on Image Processing*, vol. 6, pp. 1064–1076, Aug. 1997.

[72] P. E. Eren, M. I. Sezan, and A. Tekalp, "Robust, object-based high-resolution image reconstruction from low-resolution video," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1446–1451, 1997.

[73] B. C. Tom and A. K. Katsaggelos, "An iterative algorithm for improving the resolution of video sequences," in *SPIE Visual Communications and Image Processing*, vol. 2727, (Orlando), pp. 1430–1438, Mar. 1996.

[74] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1646–1658, Dec. 1997.

[75] A. Patti, A. Tekalp, and M. Sezan, "A new motion compensated reduced order model Kalman filter for space-varying restoration of progressive and interlaced video," *IEEE Transactions on Image Processing*, vol. 7, pp. 543–554, Apr. 1998.

[76] M. Elad and A. Feuer, "Superresolution restoration of an image sequence: Adaptive filtering approach," *IEEE Transactions on Image Processing*, vol. 8, pp. 387–395, Mar. 1999.

[77] M.-C. Hong, M. G. Kang, and A. K. Katsaggelos, "A regularized multichannel restoration approach for globally optimal high resolution video sequence," in *SPIE Visual Communications and Image Processing*, vol. 3024, (San Jose), pp. 1306–1316, Feb. 1997.

[78] N. Shah and A. Zakhor, "Multiframe spatial resolution enhancement of color video," in *Proceedings of the IEEE International Conference on Image Processing*, vol. I, (Lausanne, Switzerland), pp. 985–988, Sept. 1996.

[79] D. Chen and R. Schultz, "Extraction of high-resolution video stills from MPEG image sequences," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 465–469, 1998.

[80] S. Borman and R. Stevenson, "Simultaneous multi-frame MAP super-resolution video enhancement using spatio-temporal priors," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, (Kobe, Japan), pp. 469–473, Oct. 1999.

[81] S. Tubaro and F. Rocca, "Motion field estimators and their application to image interpolation," in *Motion Analysis and Image Sequence Processing* (M. Sezan and R. Lagendijk, eds.), ch. 6, Kluwer Academic Publishers, 1993.

[82] G. De Haan and E. B. Bellers, "Deinterlacing - an overview," *Proceedings of the IEEE*, vol. 86, pp. 1839–1857, Sept. 1998.

[83] E. Dubois, G. de Haan, and T. Kurita, "Special issue on motion estimation and compensation technologies for standards conversion," *Signal Processing: Image Communication*, vol. 6, no. 3, pp. 189–280, 1994.

[84] R. R. Schultz and R. L. Stevenson, "Motion-compensated scan conversion of interlaced video sequences," in *Image and Video Processing IV*, vol. 2666 of *Proceedings of the SPIE*, (San Jose, CA), pp. 107–118, Feb. 1996.

[85] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "High resolution standards conversion of low resolution video," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, (Detroit, MI), pp. 2197–2200, 1995.

[86] S. Mann and R. W. Picard, "Virtual bellows: constructing high quality stills from video," in *Proceedings of the IEEE International Conference on Image Processing*, (Austin, TX), pp. 363–367, 1994.

[87] L. Teodosio and W. Bender, "Salient video stills: content and context preserved," in *Proceedings of the ACM International Conference on Multimedia*, pp. 39–46, 1993.

[88] T. O'Rourke and R. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 490–499, Dec. 1995.

[89] A. M. Tekalp, "Video segmentation," in *Handbook of Image and Video Processing* (A. Bovik, ed.), ch. 4.9, pp. 383–399, Academic Press, 2000.

[90] M. A. Smith and T. Chen, "Image and video indexing and retrieval," in *Handbook of Image and Video Processing* (A. Bovik, ed.), ch. 9.1, pp. 687–704, Academic Press, 2000.

[91] A. H. Tewfik and M. Swanson, "Data hiding for multimedia personalization, interaction, and protection," *IEEE Signal Processing Magazine*, vol. 14, pp. 41–44, July 1997.

[92] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, pp. 283–301, May 1998.

[93] G. Voyatzis and I. Pitas, "Image watermarking for copyright protection and authentication," in *Handbook of Image and Video Processing* (A. Bovik, ed.), ch. 9.4, pp. 733–745, Academic Press, 2000.

[94] E. Dubois, "Motion-compensated filtering of time-varying images," *Multidimensional Systems and Signal Processing*, vol. 3, pp. 211–239, May 1992.

[95] R. L. Lagendijk, P. van Roosmalen, and J. Biemond, "Video enhancement and restoration," in *Handbook of Image and Video Processing* (A. Bovik, ed.), ch. 3.11, pp. 227–241, Academic Press, 2000.