

## Chapter VI

# Video Processing

Our general approach to statistical image modeling and processing is extended to video in this chapter. Video enhancement, denoising, restoration and superresolution have been rapidly gaining popularity due to the proliferation of entry-level, low-resolution camcorders, webcams, cell phone cameras, and low-cost surveillance systems. Moreover a longstanding need exists for advanced processing methods in scientific applications such as medical imaging, study of high-speed physical phenomena, and forensics.

Video may be viewed as a function of three coordinates: two spatial and one temporal. However, video processing is not just an extension of 2-D image processing techniques to 3D. The characteristics of video in the temporal dimension are very different from the characteristics of images in the spatial dimensions. Additionally, applications often dictate that video be accessible at arbitrary points in time (“random access”), which places constraints on video processing algorithms. For example, it should be easy to edit the video and to perform operations such as fast forward on a videotape. Finally, video models should be understood and formulated in context of the physical world from which they are acquired and in which objects evolve in four dimensions: three spatial and one temporal.

### 1 Video Formation Models

We begin with the relationship of video to the physical world. As illustrated in Fig. 1, video formation can be viewed as a mapping, or projection, from 4-D space  $(X, Y, Z, t)$  to 3-D space  $(x, y, t)$ , where  $(X, Y, Z)$  are spatial coordinates in which the scene is described,  $t$  is time, and  $(x, y)$  are spatial coordinates in the sensor (image) plane. The physical scene is represented by a function  $I_S(X, Y, Z, t)$ , and the light intensity captured by the camera is represented by a function  $I(x, y, t)$ .

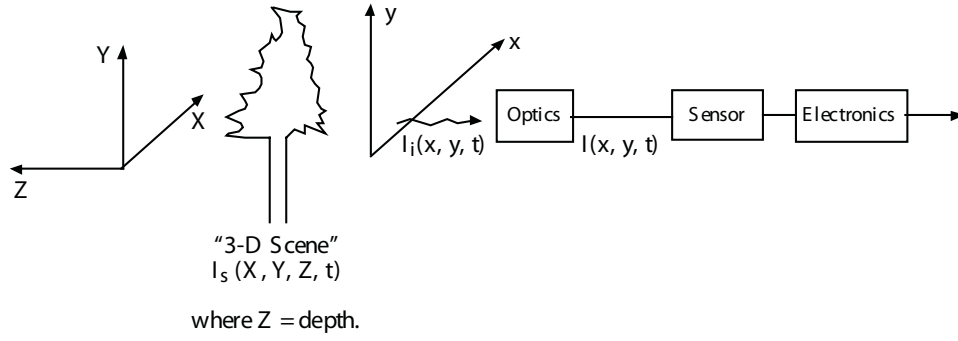


Figure 1: Video formation: from 4D to 3D

### 1.1 Geometric Models

Depending on the geometry of the image formation system, different types of projection may be considered. The most common ones are orthographic (or parallel) projection, and perspective (or central) projection.

**Orthographic projection.** If the scene is far from the sensors, it may be assumed that all the rays from the scene to the image plane are parallel to each other. In this case, the depth information in the scene is lost, and orthographic projection is a linear mapping from  $(X, Y, Z)$  coordinates to  $(x, y)$  coordinates:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

**Perspective projection.** As illustrated in Fig. 2, if the scene is close to the sensors and it is assumed that an ideal pinhole camera is used, then all the rays from the scene travel through the center of the lens. Let  $f$  denote the distance from the center of the lens to the image plane. Then the equations that describe the projection are given by identities between similar triangles,

$$\begin{aligned} \frac{x}{f} &= \frac{X}{Z + Z_0} \\ \frac{y}{f} &= -\frac{Y}{Z + Z_0}. \end{aligned}$$

This defines a nonlinear mapping. However, the mapping can be made linear in an augmented 4-D coordinate system called homogeneous coordinates [1].

### 1.2 Photometric Models

The image intensity  $I_S(X, Y, Z, t)$  is defined as the amount of light reflected by the objects in the scene. The surface properties of these objects determine the nature of the reflections. Two

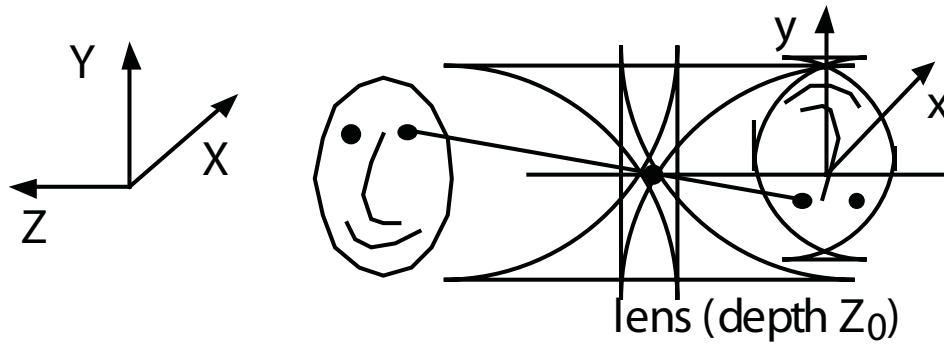


Figure 2: Perspective projection

basic types of interaction are represented in Fig. 3.

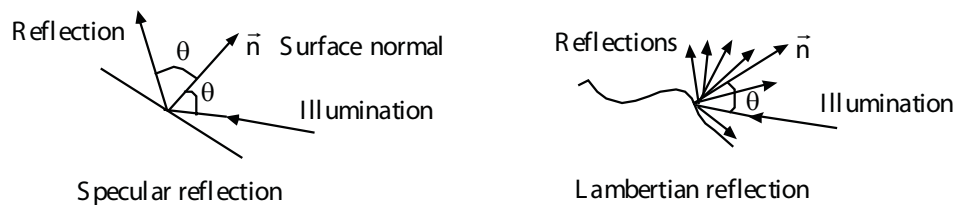


Figure 3: Photometric model

Mirror-like surfaces have reflection properties known as specular reflection: light is reflected in a specific direction in the plane formed by the normal to the surface and the direction of incoming light.

In contrast, a Lambertian surface illuminated by the same illuminator would reflect light equally in all directions. The intensity of the reflected light is equal to  $\rho I_i \cos \theta$ , where  $I_i$  is the intensity of the light illuminating the object at a given point on the surface,  $\theta$  is the normal angle, and  $\rho$  is the surface albedo, a physical characteristic of the object.

### 1.3 Sensor Models

The physical models of Chapter IV apply to video sensors as well as to still image sensors. Of particular interest here is the issue of aperture time. Since video signals are typically undersampled in time (30 frames/sec or lower), a significant amount of temporal aliasing takes place. This aliasing could be reduced by increasing the aperture time (introducing temporal blurring) prior to sampling. This is what is done in television systems. For film, however, the aperture time is much shorter, and temporal aliasing is more visible, but pictures are crisper.

Film makers have long ago learned and mastered the art of turning temporal aliasing into artistic effects.

## 2 Motion

Most advanced video processing algorithms attempt to extract the motion of objects in the scene. Motion is a fundamental clue to scene understanding. Motion also helps to provide a sparse representation of the scene. This sparsity is highly useful in applications such as compression and restoration.

A distinction should be made between 3-D motion and 2-D motion. The former refers to relative motion between the camera and the objects in the scene and is conventionally classified as rigid or nonrigid. In the rigid case, the shape of the object is modeled as a nondeformable surface. Six parameters (three translation and three rotation parameters) suffice to describe the motion of a rigid object. In the nonrigid case, the surface of the object is modeled as deformable. Such models are used for instance to analyze human faces, gestures, etc. A large number of parameters is often required to describe nonrigid motion. For instance, a wireframe model would fit triangular patches to a 3-D surface, and would describe motion in terms of the motion of triangle vertices [2]. The motion of points within the triangles can then be computed using an interpolation method.

2-D motion (also called projected motion) refers to the projection of 3-D motion onto the image plane. In general, projection implies a loss of information, so it may not be possible to retrieve 3-D motion from 2-D motion. For instance, in orthographic (parallel) projection, the component of 3-D motion along the depth ( $Z$ ) axis does not contribute to 2-D motion.

In Fig. 4, the curve  $L$ , also called motion trajectory, describes the *apparent trajectory* of pixel  $A$  over time. Image intensity is constant along motion trajectories.

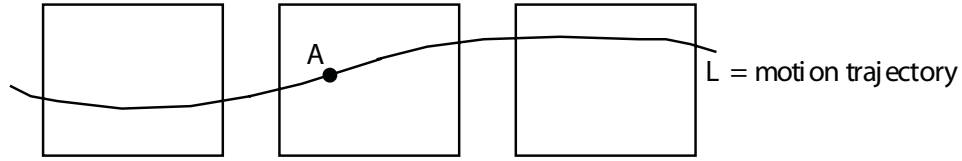


Figure 4: Motion trajectory

Another problem is that 2-D motion is not always observable. Consider for instance the case of a disk with uniform intensity spinning about its center. The rotation induces no visible changes in the image and is thus unobservable. At this point we shall thus make a clear distinction between (projected) 2-D motion and optical flow which is defined by assuming conservation of image intensity. Setting the total derivative of  $I(x, y, t)$  with respect to  $t$  to

zero, we write

$$\begin{aligned} 0 &= \frac{dI(x, y, t)}{dt} \\ &= \frac{\partial I(x, y, t)}{\partial x} v_x(x, y, t) + \frac{\partial I(x, y, t)}{\partial y} v_y(x, y, t) + \frac{\partial I(x, y, t)}{\partial t} \end{aligned} \quad (1)$$

where

$$\vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

is the so-called optical flow field, also known as “apparent 2-D motion.” The optical flow for the spinning disk example is zero, uniformly in the image plane. Note that apparent 2-D motion does not necessarily correspond to true motion. For instance, a still object whose illumination changes over time will give rise to apparent 2-D motion. The change in illumination may be caused by a variation in the intensity of the light source, or by a displacement of the light source with respect to the object.

In practice, the continuous-domain signal  $I(x, y, t)$  is not available, and only its samples on a spatio-temporal lattice are. For digital video it is convenient to introduce the notion of correspondence vectors, which describe the apparent displacement of pixels between two consecutive video frames.

Optical flow and correspondence vectors become identical only in the limit as the time interval between successive frames tends to zero.

In summary, it is generally not possible to reconstruct 3-D motion given 2-D image projections. However, it is often assumed, solely for convenience, that optical flow or correspondence vectors are the same as 2-D motion vectors.

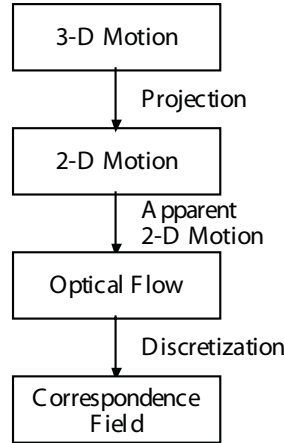


Figure 5: Relationship between different types of motion

### 3 2-D Motion Estimation

The problems of estimating the optical flow field (in continuous  $x, y, t$  coordinates) and the correspondence field (discrete  $x, y, t$ ) are closely related and can be stated as follows:

- Optical flow estimation: Given the continuous-domain signal  $I(x, y, t)$ , determine the optical flow field  $\vec{v}(x, y, t)$  for all  $x, y, t$ .
- Correspondence field estimation: If the motion vector  $\vec{v}$  is defined from time  $t$  to time  $t + \Delta t$ ,  $\vec{v}$  is called a forward motion vector. If  $\vec{v}$  is defined from time  $t$  to time  $t - \Delta t$ , it is called a backward motion vector. In either case, the motion vectors  $\vec{v}(x, y, t)$  are to be determined given the two reference frames  $t$  and  $t \pm \Delta t$ , for all  $(x, y)$ .

In Fig. 6, the point  $A$  has the same spatial coordinates in all three frames, and the motion trajectory  $L$  describes the apparent trajectory of  $A$  over time.  $\vec{v}_f$  and  $\vec{v}_b$  are the forward and backward motion vectors, respectively. In the remainder of this chapter we will use the compact notation  $s = (x, y)$  for spatial coordinates and we will drop the arrows on the velocity vectors. The intensity of pixel  $A$  at time  $t$  is given by

$$I(s, t) = I(s + v_b(s, t), t - \Delta t) \quad (\text{backward motion})$$

or

$$I(s, t) = I(s + v_f(s, t), t + \Delta t) \quad (\text{forward motion}).$$

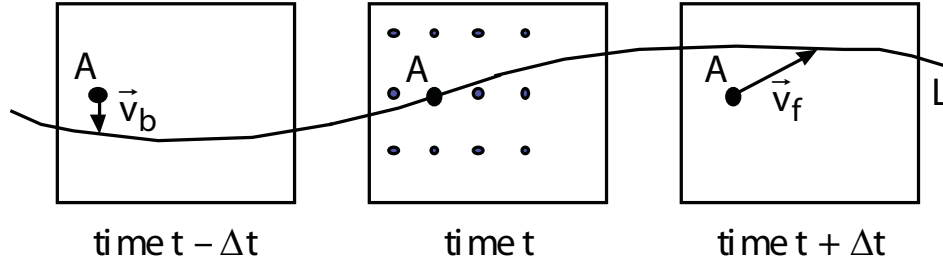


Figure 6: Backward and forward motion vectors

2-D motion estimation presents some fundamental difficulties.

**Occlusion problems.** In Fig. 7, an object is undergoing translation over a background. Some of the background is visible at time  $t$  but will be covered in the next frame, at time  $t + \Delta t$ . This is known as the covered background problem: no forward correspondence can be established for pixels in that part of the background. Conversely, some of the background visible at time  $t$  was covered at time  $t - \Delta t$ . Here no backward correspondence can be established

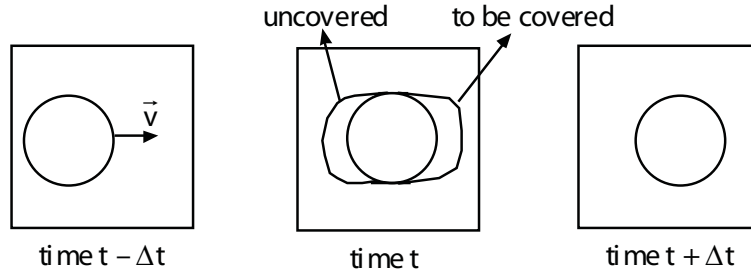


Figure 7: Occlusions

for pixels in that area. The presence of covered and uncovered background corresponds to the end and the beginning of a motion trajectory, respectively.

**Aperture problem.** The solution to the optical flow and correspondence problems is not uniquely determined. For instance, assume that the image sequence is a moving vertical edge, as shown in Fig. 8. The image intensity is constant along the vertical direction. Clearly, the vertical component of motion is unobservable; only the component of motion normal to the edge is observable. This is known as the aperture problem. The nonuniqueness of the solution is particularly clear for the (discrete) correspondence problem, where the number of unknowns  $v(s, t)$  is twice the number of data  $I(s, t)$ .

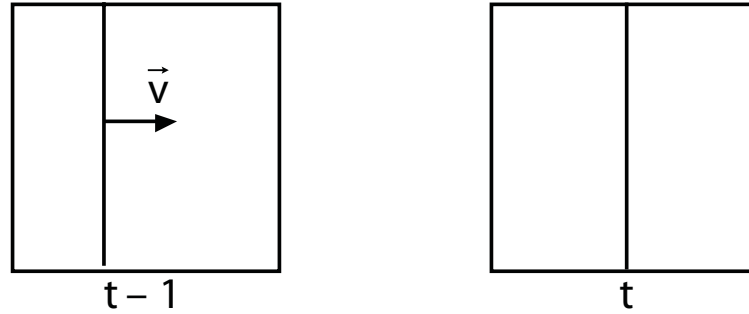


Figure 8: The aperture problem

Interestingly the 2-D motion estimation problem is closely related to the problem of image registration [3, 4], where two closely related images are to be “aligned”. Examples include matching left and right views in stereo images and matching an image to a template from a database, as occurs in pattern recognition and watermarking applications.

## 4 Correspondence Field Estimation

Assuming that image intensity along motion trajectories is constant, the intensity of a pixel at location  $s$  in frame  $t$  is given by

$$I(s, t) = I(s - d(s), t - 1), \quad s \in \Lambda, t \in \mathbb{Z}. \quad (2)$$

Here we have assumed that the time interval between consecutive video frames is unity, and that pixels are points on a lattice  $\Lambda$ . The horizontal and vertical components  $d_x(s)$  and  $d_y(s)$  of the correspondence field  $d(s)$ ,  $s \in \Lambda$  are in general real-valued, which implies that  $I(s - d(s), t - 1)$  is defined by interpolation from pixels of frame  $t - 1$ . The dependency of  $d(s)$  on  $t$  is implicit.

Seeking an exact correspondence between pixels of two consecutive frames is generally not meaningful because of discretization effects, quantization noise, and more fundamentally, the limited validity of the constant-intensity assumption along actual motion trajectories. As discussed by Del Bimbo *et al.* [5], the normal component of the projected 2-D motion field is equal to the normal optical flow only under idealized assumptions such as isotropic illumination of Lambertian surfaces and orthographic projection.

For these reasons, a more realistic model for  $I(s, t)$  is

$$I(s, t) = I(s - d(s), t - 1) + e(s, t), \quad s \in \Lambda, t \in \mathbb{Z} \quad (3)$$

where  $I(s - d(s), t - 1)$  is viewed as a prediction for  $I(s, t)$ , and  $e(s, t)$  is a prediction error image, often called Displaced Frame Difference (DFD).

If  $d(s)$ ,  $s \in \Lambda$  is representative of true motion, then  $d(s)$  should be expected to be piecewise smooth, with discontinuities corresponding to boundaries between objects moving at different velocities, and smoothness being consistent with the spatio-temporal homogeneity of any given object.

Hence, whereas the solution to the inverse problem (1) is highly noisy and is nonunique, the 2-D motion estimation problem can be made meaningful by seeking a piecewise-smooth solution to (2) that minimizes  $e(s, t)$  in some appropriate sense. Sections 5 and 6 describe two major techniques that attempt to achieve this goal.

## 5 Block Matching

Here frame  $t$  is partitioned into elementary square blocks whose pixels are assumed to undergo purely translational motion from frame  $t - 1$  to frame  $t$ . A typical block size is  $16 \times 16$  pixels. The motion field  $d(s)$  that results from this model is blockwise constant. Let  $d$  be the displacement assigned to pixels  $s$  in block  $\mathcal{B}$ . To produce an estimate  $\hat{d}$  of  $d$ , one could select the motion vector that minimizes the mean-square value of  $e(s, t)$  in (3),

$$\mathcal{E}_{MSE}(d) = \sum_{s \in \mathcal{B}} |I(s, t) - I(s - d, t - 1)|^2. \quad (4)$$



This is simply called the mean-square error (MSE) criterion. Note that the solution to this problem is also the maximum-likelihood solution to (3), when  $e(s, t)$  is modeled as iid Gaussian with zero mean and variance  $\sigma^2$ . Indeed, the log likelihood function for  $d$  given  $I$  is given by

$$\ln p(I|d) = -\frac{|\mathcal{B}|}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \mathcal{E}_{MSE}(d).$$

Another common estimation criterion is the mean absolute error (MAE) criterion

$$\mathcal{E}_{MAE}(d) = \sum_{s \in \mathcal{B}} |I(s, t) - I(s - d, t - 1)| \quad (5)$$

which is often preferred to the MSE criterion in hardware implementations because no squaring operation is required. It may be verified that the MAE solution is also the maximum-likelihood solution to (3) under an iid Laplacian model for  $e(s, t)$ .

Yet another criterion is the matching pixel count (MPC) criterion which minimizes the number of pixels that cannot be matched within some specified tolerance level  $\tau$ :

$$\mathcal{E}_{MPC}(d) = \sum_{s \in \mathcal{B}} \mathbb{1}\{|I(s, t) - I(s - d, t - 1)| > \tau\} \quad (6)$$

where  $\mathbb{1}\{\cdot\}$  denotes the indicator function of a set.

In this case, a statistical interpretation for the MPC criterion is less straightforward because  $\hat{d}$  is, strictly speaking, not a maximum-likelihood estimator. However  $\hat{d}$  still takes the form of a maximum-likelihood estimator under the following improper distribution for  $e$ : the pixels  $e(s, t)$  are iid with pdf equal to  $A$  for  $|e| \leq \tau$  and equal to  $B$  for  $|e| > \tau$ , where  $A > B > 0$  are arbitrary positive numbers.

In practice, the performance of the MSE criterion is similar to that of the MAE criterion and superior to that of the MPC criterion. Examination of  $e(s, t)$  for each block does not suggest an obvious superiority of the iid Gaussian model over the iid Laplacian model, or vice versa. The independence assumption on  $e(s, t)$  within each block is questionable, as discussed in Sec. 7.

In general, the main advantage of block-matching methods is simplicity, hence their adoption in video compression standards. Their major disadvantage is they produce motion fields with discontinuities at artificial locations (block boundaries). The choice of the size of the blocks is also an issue. For large blocks, the blockwise-constant model for the motion field is inappropriate, as the blocks may contain parts of objects moving at different velocities. The presence of groups of pixels moving at different velocities within each block is a flagrant violation of the uniform translation assumption. Such artifacts are less likely to occur for small blocks. In this case however, motion vector estimates become too noisy because they are based on an insufficient number of data.

The effects of motion model errors can be quantified under some simplifying assumptions. Assume for instance that  $e(s, t)$  within a block  $\mathcal{B}$  are iid  $\mathcal{N}(0, \sigma^2)$ . We can then compute the

Cramer-Rao lower bound on the variance of any unbiased estimator of  $d$ . The gradient of the loglikelihood function for  $d$  is given by

$$\nabla_d \ln p(I|d) = -\frac{1}{\sigma^2} \sum_{s \in \mathcal{B}} e(s, t) \nabla I(s - d, t - 1)$$

where  $\nabla_d \triangleq (\frac{\partial}{\partial d_x}, \frac{\partial}{\partial d_y})$ , and  $\nabla I$  is the spatial gradient of  $I$ . The Fisher information for  $d$  is the  $2 \times 2$  matrix

$$\begin{aligned} J(d) &= \mathbb{E}[\nabla_d \ln p(I|d) \nabla_d^T \ln p(I|d)] \\ &= \frac{1}{\sigma^4} \sum_{s \in \mathcal{B}} \mathbb{E}[e^2(s, t)] \nabla I(s - d, t - 1) \nabla^T I(s - d, t - 1) \\ &= \frac{1}{\sigma^2} \sum_{s \in \mathcal{B}} \nabla I(s - d, t - 1) \nabla^T I(s - d, t - 1) \end{aligned} \quad (7)$$

where the second equality uses the facts that  $e(s, t)$  are iid and that  $\nabla I(s - d, t - 1)$  is a deterministic quantity. For any unbiased estimator  $\hat{d}$  of  $d$ , we have the information inequality:

$$\text{Cov}(\hat{d}) \geq J^{-1}(d) = \sigma^2 \left( \sum_{s \in \mathcal{B}} \nabla I(s - d, t - 1) \nabla^T I(s - d, t - 1) \right)^{-1}$$

(See Chapter V.4.) Confidence bounds for motion estimation and image registration have also been explored in [4, 6, 7, 8].

The information inequality reveals some interesting facts. First, the accuracy of estimates is of the order of  $\sigma$ . Second, accuracy improves with block size, because the summand  $(\nabla I)(\nabla I)^T$  is nonnegative definite. Third, accuracy improves in the presence of large image gradients. One should expect poor motion accuracy in flat areas of the image. Interestingly, if the gradients are large in one direction but small or even zero in the orthogonal direction (as is the case for blocks that contain two homogeneous regions separated by a straight edge), the Fisher information matrix in (7) has rank one. The eigenvector corresponding to the nonzero eigenvalue in the example above is  $\frac{1}{\|\nabla I\|} \nabla I$ . In this case, motion estimates are accurate in the direction of  $\nabla I$ , but not at all in the orthogonal direction. This is a manifestation of the aperture problem discussed in Section 3.

From the discussion above, it is clear that motion vectors can be quantized to a certain accuracy without impacting the quality of motion estimates. In compression applications,  $16 \times 16$  blocks are often used, with motion vectors quantized to half-pixel accuracy. In some cases quarter-pixel accuracy is used locally.

If a  $16 \times 16$  block contains groups of pixels moving with different velocities, it is often advantageous to split the block into four  $8 \times 8$  blocks and to perform motion estimation for each of the smaller blocks. The motion field can then be represented as a quadtree as shown below. The decision of whether to split a  $16 \times 16$  block is often made by comparing the values

of the estimation criterion before and after splitting. The splitting criterion is often simple but heuristic. The motion estimation methods introduced in the following section provide a more fundamental tradeoff between spatial homogeneity of the motion field and size of the motion-compensated prediction errors.

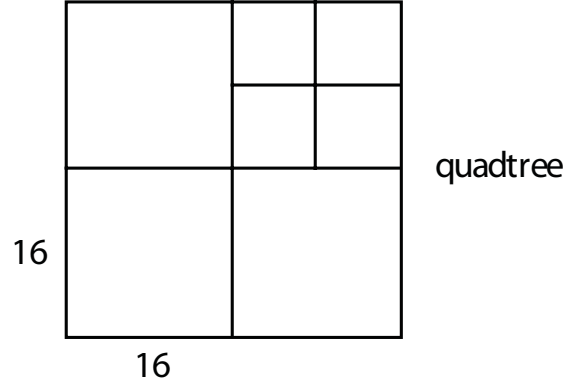


Figure 9: Quadtree model for motion field

## 6 Regularization

We now investigate motion estimation techniques that present three significant advantages over block-based techniques:

- The motion field estimates do not present artificial discontinuities at block boundaries.
- A regularization parameter  $\lambda$  provides a systematic tradeoff between spatial homogeneity of motion field estimates and energy of the motion-compensated prediction errors.
- Occlusions are explicitly recognized.

These techniques are obtained by application of the regularization principles covered in Chapter V. Consider for instance the following regularized MSE criterion for estimating the motion field  $d(s)$ ,  $s \in \Lambda$ :

$$\mathcal{E}(d) = \sum_{s \in \Lambda} |I(s, t) - I(s - d(s), t - 1)|^2 + \lambda \phi(d). \quad (8)$$

The motion field estimate  $\hat{d}(s)$ ,  $s \in \Lambda$  is the minimizer of this expression. Here  $\phi(d)$  is a regularization functional which favors piecewise smooth motion fields, and  $\lambda$  is the regularization parameter, which trades off fidelity to the data (as captured by the first term of  $\mathcal{E}(d)$ ) and spatial coherence of the motion field (second term).

As discussed in Chapter V, regularization criteria such as  $\mathcal{E}(d)$  above are equivalent to MAP estimation assuming that  $e(s, t) = I(s, t) - I(s - d(s), t - 1)$  are iid  $\mathcal{N}(0, \sigma^2)$ , and  $d$  is random with pdf  $p(d) \propto \exp\{-\frac{\lambda}{2\sigma^2}\phi(d)\}$ . One difference with the applications studied in Chapter V should be pointed out: the field  $d$  being modeled and estimated here is never directly observable and is used for video modeling purposes only.

Serious computational difficulties arise due to the highly nonlinear nature of  $\mathcal{E}(d)$  in (8). For convenience, the regularization functional is often chosen to be quadratic:

$$\phi(d) = \|\nabla d\|^2 = \|\nabla d_x\|^2 + \|\nabla d_y\|^2.$$

Moreover, the DFD  $e(s, t)$  often linearized in  $d$  by expanding  $I(s - d(s), t - 1)$  in a first-order Taylor series expansion around  $I(s, t)$ :

$$\begin{aligned} I(s - d(s), t - 1) &\approx I(s, t) - \nabla^T I(s, t) \cdot d(s) - \frac{\partial I(s, t)}{\partial t} \\ \Rightarrow e(s, t) &\approx \nabla^T I(s, t) \cdot d(s) + \frac{\partial I(s, t)}{\partial t}. \end{aligned}$$

The resulting estimation criterion is quadratic,

$$\mathcal{E}_{HS}(d) = \sum_{s \in \Lambda} \left| \nabla^T I(s, t) \cdot d(s) + \frac{\partial I(s, t)}{\partial t} \right|^2 + \lambda \|\nabla d\|^2.$$

This method was introduced by Horn and Schunck [9] to estimate optical flows. The first-order Taylor series approximation is valid if  $I(s, t)$  is twice differentiable and if the time interval between successive frames is small enough. Unfortunately this approximation typically breaks down in standard digital video sequences. The linearization idea can be salvaged, but more robust computational methods are needed, as described in Barron *et al.* [10], Luetthgen [11], Krishnamurthy *et al.* [12, 13], and Bruhn *et al.* [14]. Fig. 10 compares typical motion fields estimated from a videoconferencing sequence, using block matching and a regularized multiresolution optical flow estimation method. The estimated motion field in Fig. 10(c) is closer to “true motion” which in this case is head motion.

As discussed in Chapter V, quadratic regularization penalizes edges too much, so motion field discontinuities get blurred. An attractive alternative to quadratic regularization, proposed by Dubois and Konrad [15], is to use a Markov random field model for  $d(s)$ , coupled with a line process  $l$ . The quadratic penalty  $\|\nabla d\|^2$  is replaced with a potential function of the type

$$U(d, l) = \sum_{s \sim s'} (1 - l_{ss'}) \|d(s) - d(s')\| + \mu l_{ss'} \quad (9)$$

where  $\mu$  is a positive constant, and the summation is over neighbors  $s \sim s'$ .

A similar idea can be used to take occlusions into account. Define a binary-valued occlusion field  $o(s)$ ,  $s \in \Lambda$ , where  $o(s) = 1$  indicates the presence of an occlusion, and  $o(s) = 0$  means no

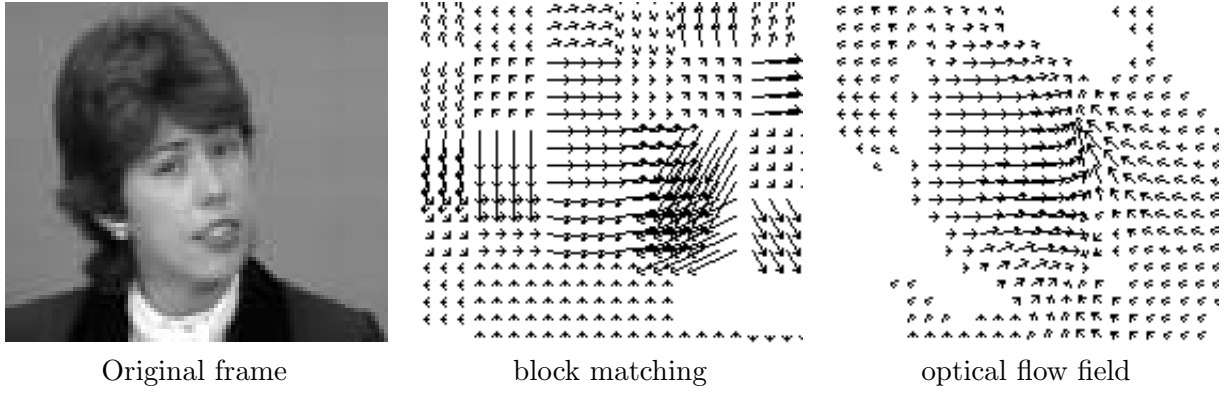


Figure 10: Block matching *vs* optical flow for the *Claire* videoconferencing sequence [12].

occlusion. In the presence of an occlusion, the motion-compensated prediction model breaks down. The data fidelity term  $\sum_{s \in \Lambda} e(s, t)^2$  is then be replaced by

$$U(e, o) = \sum_{s \in \Lambda} (1 - o(s)) e^2(s, t) + \nu o(s) \quad (10)$$

where  $\nu$  is a positive constant. Note that  $o$  can be eliminated, since

$$\min_o U(e, o) = \sum_{s \in \Lambda} \min\{e^2(s, t), \nu\}.$$

The truncated quadratic function  $\min\{e^2(s, t), \nu\}$  avoids excessive penalization of outliers, which are presumably due to motion model failure.

The particular choice of  $U(e, o)$  in (10) does not take into account the spatial coherence of the occlusion field. Neither does it take into account dependencies between  $o(s)$  and the line process  $l_{ss'}$  which indicates the location of motion field discontinuities. One may thus generalize (10) to a model of the form [15]

$$U(e, o) = \sum_{s \in \Lambda} (1 - o(s)) e^2(s, t) + \nu U(o|l). \quad (11)$$

The cost function to be minimized is  $U(e, o) + U(d, l)$ , where  $e$  is an explicit function of  $d$ . This is a high-dimensional nonlinear optimization problem involving a vector-valued field  $d$  and two binary-valued fields  $o$  and  $l$ . An alternating minimization algorithm for solving this problem is described by Dubois and Konrad [15]. The algorithm successively minimizes the cost function over  $d$ ,  $o$ , and  $l$  before revisiting  $d$ , etc. Practical experiments have shown that the resulting motion and occlusion fields exhibit the expected spatial coherence properties.

## 7 DFD Models

The choice of the MSE criterion for block matching in Section 5 and the choice of the potential functions  $U(e, o)$  in Section 6 implicitly assume that the DFD pixels  $e(s, t)$  are iid Gaussian over each block (Section 5) or over the entire frame (Section 6). This assumption is clearly simplistic. There are in fact significant dependencies between DFD pixels, as the observation of typical DFDs does suggest. While there has been little systematic analysis of statistical properties of DFDs, it is generally well understood that DFDs have mostly high-frequency components. To see this, assume that the motion of some image region is uniform but is measured with an error  $\Delta d$ . Consider the effect of this error on two sinusoids along the direction of  $\Delta d$ . Referring to Fig. 11, we have

$$\begin{aligned}
 I(s, t) &= \sin(\omega s - t) \\
 \hat{I}(s, t) &= \sin(\omega(s - \Delta d) - t) \\
 \Rightarrow \Delta(s, t) &= I(s, t) - \hat{I}(s, t) \\
 &= 2 \sin\left(\frac{1}{2}\omega\Delta d\right) \cos\left(\omega s - t - \frac{1}{2}\omega\Delta d\right) \\
 &\approx \omega\Delta d \cos(\omega s - t) \quad \text{for } \omega\Delta d \ll 1.
 \end{aligned}$$

The effect of the motion measurement error is much less significant on the low-frequency sinusoid than on the high-frequency sinusoid.

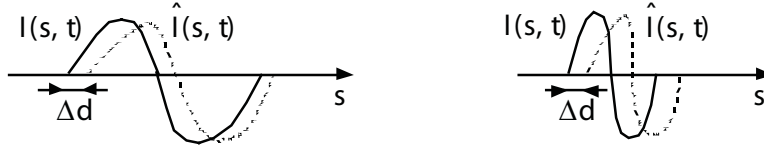


Figure 11: Prediction error due to motion estimation error

## 8 Motion-Compensated Frame Interpolation

In many applications, digital video is subsampled temporally to reduce storage or transmission requirements. For instance, rates of 60, 30, 15, 10 and 7.5 frames/sec are often encountered. It is often desirable to interpolate between available frames to enhance the quality of the display. In some applications, the interpolation technique can be crude but still effective. For instance, motion pictures are shot at a rate of 24 pictures/sec, but each picture is projected twice (or even thrice) in movie theaters, resulting in a display rate of 48 or 72 pictures/sec.

In other applications, such a crude temporal interpolation technique would cause objectionable visual artifacts. Note that standard linear interpolation techniques based on bandlimit- edness assumptions are usually ineffective due to the presence of significant temporal aliasing.

The best frame interpolation techniques are based on motion models: if motion trajectories can be properly estimated, then the value of pixels such as  $A$  in Fig. 6 can be estimated from pixels in the previous and next frames, at times  $t \pm \Delta t$ .

In order for motion-compensated frame interpolation to be successful, good motion estimates should be available. This usually precludes the use of block-matching algorithms, as illustrated in Figs 12 and 13 [16, 12, 13].



Figure 12: Frames 46, 47 and 48 of *Susie* sequence

## 9 Video Denoising

Digital video is often corrupted by noise such as film-grain noise, electronic amplifier noise and photon noise in sensing devices, and quantization noise. Speckle noise is present in coherent imaging modalities such as ultrasound or microwave imaging.

Various techniques can be employed to reduce the noise in digital video [17]. As in still image denoising, the key is to properly model image and noise characteristics. For simplicity, we consider a simple additive white Gaussian noise model:

$$y(s, t) = I(s, t) + w(s, t), \quad s \in \Lambda, t = 1, 2, \dots, T \quad (12)$$

where  $I(s, t)$  and  $y(s, t)$  are respectively the clean and noisy video sequences, and  $w(s, t)$  is iid  $\mathcal{N}(0, \sigma^2)$ .

While each frame could be denoised independently of the other ones (using the methods of Chapter V), this would fail to exploit temporal redundancies. Currently the best denoising techniques are those based on motion models. Consider the following four scenarios which differ in the degree of complexity of the motion field:

**Case I.** Still image sequence:  $I(s, t) = I(s, t - 1) = \tilde{I}(s)$  for  $t = 1, 2, \dots, T$ . The model (12) takes the form

$$y(s, t) = \tilde{I}(s) + w(s, t), \quad s \in \Lambda, t = 1, 2, \dots, T.$$



Figure 13: Reconstruction of Frame 47 from Frames 46 and 48 [13]

Hence temporal averaging reduces noise without any degradation of the images:

$$\hat{I}(s) = \frac{1}{T} \sum_{t=1}^T y(s, t).$$

Further improvements in performance are possible because spatial redundancies can be exploited using a spatio-temporal filter. Note that linear filters are simple but also suboptimal owing to the non-Gaussian nature of image statistics.



**Case II.** Ideal motion model:  $I(s, t) = I(s - d(s, t), t - 1)$  for  $t = 1, 2, \dots, T$ , known  $d(s, t)$ . The image intensity is constant along known motion trajectories. In this case, one can simply perform temporal averaging along motion trajectories. Specifically, view the first image in the original sequence as the template to be estimated:  $\tilde{I}(s) \triangleq I(s, 1)$ . The subsequent images are obtained recursively from the first one. Denote by  $\psi(s, t) \in \mathbb{R}^2$  a motion trajectory starting at location  $s \in \Lambda$  in frame 1, i.e.,  $\psi(s, 1) = s$ . By our assumption on the conservation of image intensity along motion trajectories, we have  $I(\psi(s, t), t) = \tilde{I}(s)$ . For simplicity we also assume that all motion trajectories terminate in frame  $T$  and not before. Now realign the observed image sequence along the above motion trajectories. The realigned sequence is defined as

$$\tilde{y}(s, t) = y(\psi(s, t), t).$$

We may similarly define the realigned noise sequence  $\tilde{w}(s, t) = w(\psi(s, t), t)$ . These realigned image sequences are related by

$$\tilde{y}(s, t) = \tilde{I}(s) + \tilde{w}(s, t), \quad s \in \Lambda, t = 1, 2, \dots, T$$

which reduces the problem to Case I.

**Case III.** Motion-compensated predictive model:  $I(s, t) = I(s - d(s, t), t - 1) + e(s, t)$ , where  $d(s, t)$  is known and  $e(s, t)$  is additive white Gaussian noise. In this case again, motion-compensated spatio-temporal filtering is appropriate.

**Case IV.** Same as in Case III, but now  $d(s, t)$  is unknown. The denoising problem can be addressed in two steps: first estimate  $d(s, t)$ , second apply methods from Case III assuming that the true  $d(s, t)$  is available. Of course, noise affects the estimate of  $d(s, t)$  and this reduces the efficiency of the second step.

An alternative method is to estimate  $d(s, t)$  and  $I(s, t)$  jointly. See Burl [18] for a method based on Kalman filtering, and illustrated by an example involving uniform motion of a toy automobile. A more general method based on alternating minimization is presented in the next section.

## 10 Video Restoration

Digital video sequences are often blurred due to out-of-focus cameras and to motion during the aperture time of the camera. It is often assumed that consecutive frames are not affected by temporal blurring, so that only spatial blurring should be taken into account. A classical model is the following one:

The use of motion models raises some technical difficulties because the blurred video  $h \star I$  does in general not satisfy the same motion model as the original video  $I$ . A possible approach is to alternate between estimation steps for motion and for the image sequence itself, as described below.

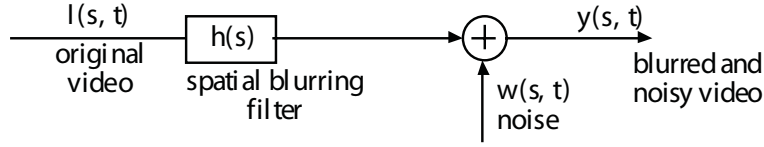


Figure 14: Video restoration model

The observational model is

$$y(s, t) = \mathcal{H} I(s, t) + w(s, t) \quad s \in \Lambda, t = 1, 2, \dots, T \quad (13)$$

where  $\mathcal{H}$  is the blur operator,  $I(s, t)$  and  $y(s, t)$  are respectively the clean and noisy video sequences, and  $w(s, t)$  is iid  $\mathcal{N}(0, \sigma_w^2)$ . The motion-based model for  $I(s, t)$  is given by

$$I(s, t) = I(s - d(s, t), t - 1) + e(s, t) \quad (14)$$

where  $e(s, t)$  is modeled as iid  $\mathcal{N}(0, \sigma^2)$ .

The joint MAP estimation problem for  $(I, d)$  takes the form

$$\max_{I, d} p(y|I) p(I|d) p(d).$$

Under our Gaussian model for  $w$  and  $e$  in (13) and (14), this criterion takes the form

$$\min_{I, d} \left\{ \frac{1}{2\sigma_w^2} \sum_{t=1}^T \sum_{s \in \Lambda} |y(s, t) - \mathcal{H} I(s, t)|^2 + \frac{1}{2\sigma^2} \sum_{t=1}^T \sum_{s \in \Lambda} |I(s, t) - I(s - d(s, t), t - 1)|^2 + \lambda \phi(d) \right\}. \quad (15)$$

The alternating minimization approach is as follows:

- Pick an initial estimate  $\hat{I}^{(0)}$  of  $I$ , e.g., the degraded data  $y$  themselves.
- For  $i = 1, 2, \dots$  do

- Estimate  $\hat{d}^{(i)}$  from  $\hat{I}^{(i-1)}$  by solving

$$\min_d \left\{ \frac{1}{2\sigma^2} \sum_{t=1}^T \sum_{s \in \Lambda} |\hat{I}^{(i)}(s, t) - \hat{I}^{(i)}(s - d(s, t), t - 1)|^2 + \lambda \phi(d) \right\}$$

using one of the methods of Sec. 6.

- Estimate  $\hat{I}^{(i)}$  from  $\hat{d}^{(i)}$  by solving

$$\min_I \left\{ \frac{1}{2\sigma_w^2} \sum_{t=1}^T \sum_{s \in \Lambda} |y(s, t) - \mathcal{H} I(s, t)|^2 + \frac{1}{2\sigma^2} \sum_{t=1}^T \sum_{s \in \Lambda} |I(s, t) - I(s - \hat{d}^{(i)}(s, t), t - 1)|^2 \right\}$$

which is a quadratic optimization problem.

The algorithm is guaranteed to reduce the cost function at each step but is greedy and does generally not converge to a global minimum (and perhaps not even to a local minimum unless some differentiability assumptions are met.)

## 11 Video Superresolution

As mentioned in the introduction to this chapter, video is often acquired at low spatial and/or temporal resolution, and it is often desired to improve that resolution. One problem that has received considerable attention is the reconstruction of a high-resolution single frame from a sequence of low-resolution images.

This problem can be cast in the framework of the previous section, with proper definition of the operator  $\mathcal{H}$ . For the reconstruction problem given above,  $\mathcal{H}$  is the cascade of a lowpass filter and a downsampling operator. This is a linear shift-variant system. Excellent results have been obtained using the alternating minimization algorithm of Sec. 10 [19]. Applications are given to face recognition, text reconstruction, hyperspectral image reconstruction.

A related problem is post-processing of compressed video sequence. In this case the observational model is of the form

$$z(s, t) = (\mathcal{C}I)(s, t)$$

where  $\mathcal{C}$  is a *compression algorithm*, i.e., the compressed data  $z$  determine a feasible set  $\mathcal{C}^{-1}(z)$  for the original image sequence. While  $\mathcal{C}$  is a nonlinear operator, the alternating minimization algorithm of Sec. 10 can be used, with  $\mathcal{C}$  in place of the operator  $\mathcal{H}$ .

## References

- [1] A. M. Tekalp, *Digital Video Processing*, Prentice-Hall, Upper Saddle River, NJ, 1995.
- [2] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced coding techniques for very low bit rate applications," *Proceedings of the IEEE*, Vol.83, No.2, pp.259—271, Feb. 1995.
- [3] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pp. 674—679, 1981.
- [4] D. Robinson and P. Milanfar, "Fundamental Performance Limits in Image Registration," *IEEE Trans. Im. Proc.*, Vol. 13, No. 9, pp. 1185-1199, Sep. 2004.
- [5] A. Del Bimbo, P. Nesi and J. L. C. Sanz, "Analysis of Optical Flow Constraints," *IEEE Trans. Im. Proc.*, Vol. 4, No. 4, pp. 460—469, Apr. 1995.
- [6] G.-S. Young and R. Chellappa, "Statistical Analysis of Inherent Ambiguities in Recovering 3-D Motion from a Noisy Flow Field," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 10 pp. 995—1013, Oct. 1992.
- [7] Simoncelli, E.P., Adelson, E.H., and Heeger, D.J. "Probability distributions of optical flow," *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society Press: Maui, HI, pp. 310—315, 1991.
- [8] B. Girod, "Motion Compensation: Visual Aspects, Accuracy, and Fundamental Limits," Chapter 5 in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds., Kluwer, 1993.
- [9] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, Vol. 17, pp. 185—203, 1981.
- [10] J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Performance of Optical Flow Techniques," *Int. J. Comp. Vision*, 12:1, pp. 43—77, 1994.
- [11] M. R. Luetttgen, W. C. Karl and A. S. Willsky, "Efficient Multiscale Regularization with Applications to the Computation of Optical Flow," *IEEE Trans. Im. Proc.*, Vol. 3, No. 1, pp. 41—64, Jan. 1994.
- [12] P. Moulin, R. Krishnamurthy and J. W. Woods, "Multiscale Modeling and Estimation of Motion Fields for Video Coding," *IEEE Trans. Im. Proc.*, Vol. 6, No. 12, pp. 1606—1620, Dec. 1997.
- [13] R. Krishnamurthy, P. Moulin and J. W. Woods, "Frame Interpolation and Bidirectional Prediction of Video Using Compactly Encoded Optical-Flow Fields and Label Fields," *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 9, No. 5, pp. 713—726, 1999.

- 
- [14] A. Bruhn, J. Weichert, C. Schnörr, “Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods,” *Int. J. Computer Vision*, Vol. 61, pp. 211—231, 2005.
  - [15] E. Dubois and J. Konrad, “Estimation of 2-D Motion Fields from Image Sequences with Application to Motion-Compensated Processing,” Chapter 3 in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds., Kluwer, 1993.
  - [16] E. Dubois, “Motion-Compensated Filtering of Time-Varying Images,” *Multidimensional Systems and Signal Processing*, Vol. 3, pp. 211—239, 1992.
  - [17] J. C. Brailean et al., “Noise Reduction Filters for Dynamic Image Sequences: A Review,” *Proc. IEEE*, Vol. 83, No. 9, pp. 1270—1292, Sep. 1995.
  - [18] J. B. Burl, “A Reduced Order Extended Kalman Filter for Sequential Images Containing a Moving Object,” *IEEE Trans. Im. Proc.*, Vol. 2, No. 3, pp. 285—295, July 1993.
  - [19] A. Katsaggelos, R. Molina, and J. Mateos, *Super Resolution of Images and Video*, Morgan & Claypool, 2007.