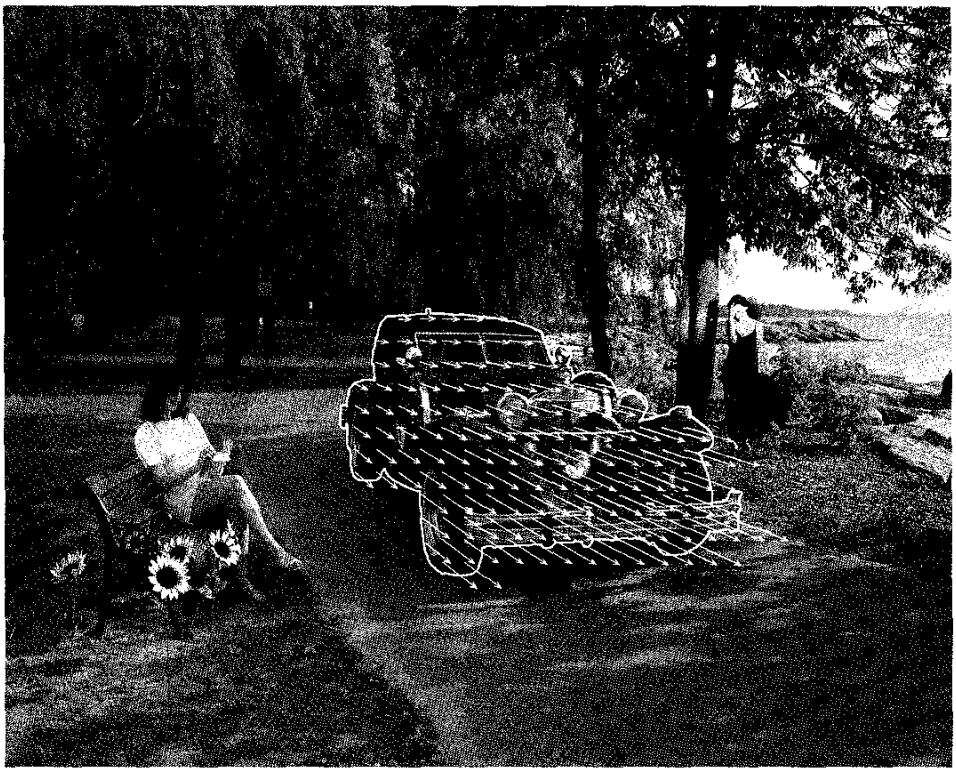


Estimating Motion in Image Sequences

A Tutorial on Modeling and Computation of 2D Motion

Christoph Stiller
and Janusz Konrad

Motion is a prominent source of temporal variations in image sequences. In order to model and compute motion, we need to understand how images (and, therefore, image motion) are formed. Motion in image sequences acquired by a video camera is induced by the movements of objects in a 3D scene and by camera motion. Thus, a camera's parameters, such as its 3D motion (rotation, translation) or focal length, play an important role in image motion modeling. If we know these parameters precisely, only object motion needs to be recovered. However, this scenario is rather rare, and both object and camera motion usually need to be computed. The 3D motion of objects and cameras induces 2D motion on the image plane via a suitable projection system. It is this 2D motion, also called *apparent motion* or *optical flow*[1], that needs to be recovered from intensity and color information of a video se-

1053-5888/98/$10.00©1998IEEE

quence. 2D motion finds diverse applications in video processing and compression as well as in computer vision, primarily because the temporal correlation of intensities (and color) in an image sequence is very high in the direction of motion.

In video compression, the knowledge of motion helps remove temporal data redundancy and therefore, attain high compression ratios. Motion estimation became a fundamental component of such standards as H.261, H.263, and the MPEG family [46], [45], [49]. Although motion models used by the older standards are very simple (one 2D vector per block), the new MPEG-4 standard[2] offers an alternative (region-based) model that allows increased efficiency and flexibility [47], [75]. In video processing, motion information is used for standards conversion (motion-compensated 3D sampling structure conversion), noise suppression (motion-compensated filtering) [11], or deblurring (motion-compensated restoration) [86]. In computer vision, 2D motion usually serves as an intermediary in the recovery of camera motion or scene structure [42].

To compute motion trajectories, three basic elements need to be specified. First, underlying models must be selected, e.g., the motion model (representation, region of support), motion and image data relationship model (observation model), motion boundary model, and occlusion model. The choice of models and their parameters is application-dependent. For example, the occlusion model may not be relevant for a block-based compression, whereas it would be essential in image analysis. Second, an estimation criterion must be identified. Such a criterion may take different forms, such as a simple mean-squared error over a block, a robust criterion (e.g., with saturation for large errors), or a complex rate-distortion or Bayesian criterion involving multiple terms. Third, a search strategy must be implemented to determine the motion parameters that optimize the selected criterion. In general, by a suitable selection of search strategy, one can trade, to a large extent, optimization performance against computational load. The strategy may be deterministic or stochastic in nature. Exhaustive and simplified search methods as well as deterministic relaxation belong to the most popular schemes and include, as special cases, block matching and gradient-based methods. Among the best-known deterministic relaxation methods are *Iterated Conditional Modes* and *Highest Confidence First*. Mean-field techniques stemming from statistical mechanics are important deterministic optimization techniques based on the approximation of a partition function. Stochastic relaxation techniques, including simulated annealing, are dominant among the stochastic approaches. An important element of the optimization strategy is its hierarchical implementation in order to avoid the violation of some underlying assumptions (e.g., local intensity linearity) and/or reduce the computational complexity of the algorithm.

Models

Motion Representation

Consider a point on an object moving in 3D space. Let its position at time t be

$$\boldsymbol{X} = \boldsymbol{X}(t) = (X(t), Y(t), Z(t))^T \in \mathbb{R}^3$$

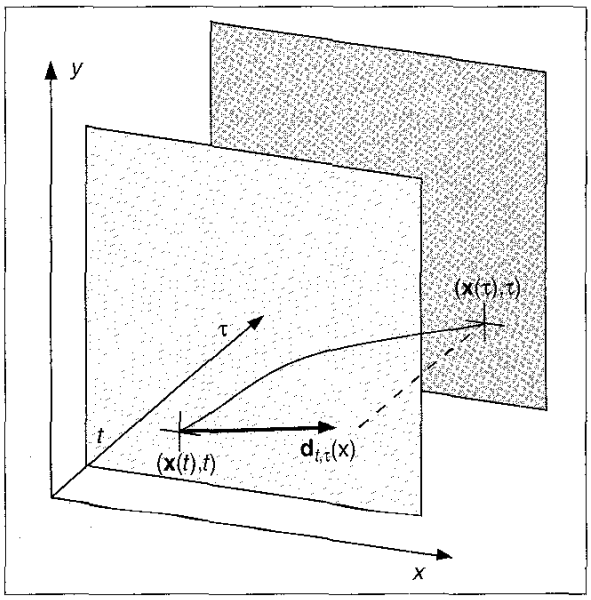
expressed in camera coordinates. $(\boldsymbol{X}(t), t)$ defines a curve in 3D space over time which we refer to as the *world motion trajectory*. For any two time instants t and τ, the world motion trajectory identifies a 3D displacement in position

$$\mathcal{D}_{t,\tau}(\boldsymbol{X}) = \boldsymbol{X}(\tau) - \boldsymbol{X}(t).$$

For a review of 3D motion and its relationship to the apparent 2D motion of interest here, the reader is referred to [1], [63].

An image acquisition system projects the 3D world onto a 2D image plane with image coordinates $\boldsymbol{x} = (x, y)^T \in \Lambda$, where Λ is a sampling grid, usually an orthogonal lattice. Upon this projection, world motion trajectories result in (2D) motion trajectories $(\boldsymbol{x}(t), t)$. We adopt the definition of a 2D motion trajectory proposed in [24]: a trajectory is defined only in the time interval in which the associated point is visible in the image. Thus, assuming that we are dealing with non-transparent objects, each spatio-temporal position $(\boldsymbol{x}, t)$ belongs to a motion trajectory of only one visible point. As depicted in Fig. 1, the 2D displacement can be expressed, similarly to the 3D displacement, as follows

$$\boldsymbol{d}_{t,\tau}(\boldsymbol{x}) = \boldsymbol{x}(\tau) - \boldsymbol{x}(t). \tag{1}$$



1. Motion trajectory $\boldsymbol{x}(t)$ and associated displacement vector $\boldsymbol{d}_{t,\tau}(\boldsymbol{x})$.

For simplicity of notation, either the second or both subscripts will be omitted whenever it is clear from the context.

In general, a *motion field* is a vector-valued function of continuous spatial coordinates. In practical applications, this function is often described in a parametric form using a finite, usually small, number of parameters.

Since 2D motion results from the projection of moving 3D objects onto the image plane, a model for 2D motion fields can be derived from models describing 3D motion, 3D surface function, and camera projection geometry. If these models are parametric, the resulting 2D motion model will be parametric as well. As a simple example, consider a 3D planar patch undergoing 3D affine motion under orthographic projection. The 3D affine motion can be written as follows

$$\mathcal{D}(\boldsymbol{X}) = (\boldsymbol{R} - \boldsymbol{I})\boldsymbol{X} + \boldsymbol{s}. \tag{2}$$

In general, the 3×3 matrix $\boldsymbol{R} = (r_{ij})$ has nine degrees of freedom, and the translational motion vector $\boldsymbol{s} = (s_1, s_2, s_3)^T$ has another three degrees of freedom. Equation (2) includes rigid motion as a special case. Then, $\boldsymbol{R}$ is a rotation matrix, i.e., its columns (rows) are orthonormal, thus allowing only three degrees of freedom corresponding to the three rotation axes.

Let the planar patch be specified by three parameters α, β, γ, as follows,

$$\alpha X + \beta Y + \gamma Z = 1. \tag{3}$$

The camera model provides two additional scalar equations mapping 3D world coordinates onto 2D coordinates of the image plane. For an orthographic projection, the following relationship holds:

$$x = cX, \quad y = cY; \quad c \in \mathbb{R}. \tag{4}$$

Substituting equations for the camera model (4) and for the 3D surface (3) into (2), we readily obtain a model for 2D motion which, for the given example, becomes the 2D affine model

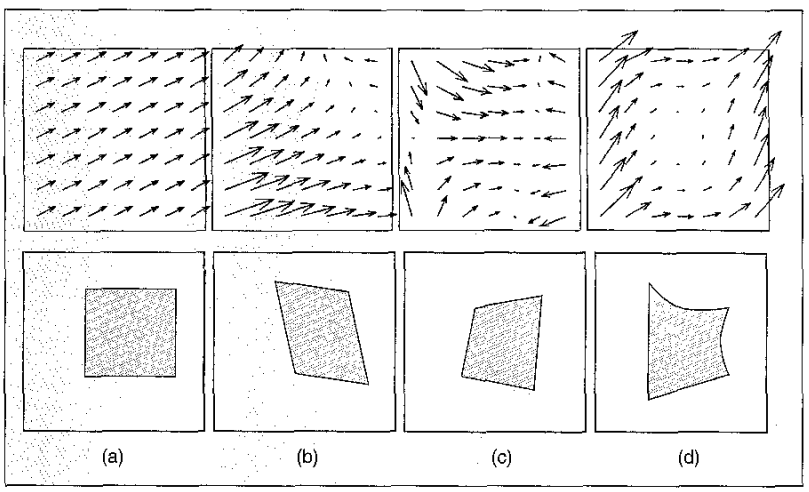
$$\boldsymbol{d}(\boldsymbol{x}) = (\boldsymbol{A} - \boldsymbol{I})\boldsymbol{x} + \boldsymbol{b},$$

with

$$\boldsymbol{A} = \begin{pmatrix} r_{11} - \frac{\alpha}{\gamma} r_{13} & r_{12} - \frac{\beta}{\gamma} r_{13} \\ r_{21} - \frac{\alpha}{\gamma} r_{23} & r_{22} - \frac{\beta}{\gamma} r_{23} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \frac{c}{\gamma} r_{13} + cs_1 \\ \frac{c}{\gamma} r_{23} + cs_2 \end{pmatrix}.$$

This model has been used extensively in the literature for 2D motion representation [4], [61]. Clearly, a 2D motion model does not uniquely correspond to one 3D model; identical 2D motion models may result from different assumptions about 3D motion, surface and camera projection models.

Table 1 summarizes some parametric models for 2D motion and provides possible underlying assumptions. The first four models are illustrated in Fig. 2; for each model, an example of a motion vector field is shown along with the corresponding motion-compensated square. The simplest (translational) model for 2D motion is used in the existing coding standards [45], [46], [49]. It accounts for a rigid translational 3D motion under orthographic projection, resulting in a spatially constant 2D motion. Clearly, motion compensation with such fields preserves any 2D shape. With affine motion, parallel lines remain parallel in the motion-compensated image. The 3D affine motion of planar patches under a perspective camera model leads to an eight-parameter model that is linear in projective coordinates [90]. We can easily see that this model includes the 2D affine model as a special case; lines remain lines after motion compensation. A quadratic model was proposed in [22] to describe 3D affine motion of parabolic surfaces under orthographic projection. It includes as special cases the 2D affine model and a close Taylor approximation of the eight-parameter model. As can be seen in Fig. 2, motion compensation with this model does not preserve lines.



▲ 2. *Examples of parametric motion vector fields (sampled) and corresponding motion-compensated predictions of a centered square: (a) translational, (b) affine, (c) projective linear, and (d) quadratic. See Table 1 for model descriptions.*

All models discussed so far are parametric and involve a fixed number of parameters. Such models can be used efficiently for the estimation, interpretation, and transmission of certain classes of motion fields. These models, however, are not capable of describing arbitrary 2D motion fields.

A different description of motion fields can be provided by vector fields represented on a rectangular lattice or a mesh. In this way, the number of parameters varies with the number of considered pixels. Off-lattice vectors of the motion field can be approxi-

mated by suitable interpolation of the sampled field [66]. In general, the interpolation kernel H (Table 1) has a small support, such that a motion vector is usually interpolated from, at most, four samples. The frequently used bilinear interpolation kernel is a tensor product of horizontal and vertical 1D triangular kernels. Recently, an interesting generalization of this model has been presented [65], where H is a complete multiresolution basis implemented using a perfect-reconstruction, non-separable subband scheme. When the motion sampling lattice involves at least one site per image pixel, the motion fields are called *dense*. Obviously, dense motion fields provide a fairly general description of motion, but estimation, interpretation, and transmission thereof involve large amounts of data.

Another frequently used model employs a triangular mesh; motion vectors of a dense field are interpolated from three motion vectors at the corners of each triangle. When the motion is sampled on a predetermined mesh, the interpolation will be, in general, imprecise at discontinuities in the motion field. Therefore, *adaptive meshes* have been proposed [86], [92] that select the sampling points in such a way that the interpolated area of the motion field contains pixels from only one moving object. Typically, those points lie on intensity edges of the image.

As with image intensity patterns, motion fields are highly correlated spatially. Therefore, it can be expected that such fields can be efficiently represented using linear transforms followed by zeroing of high-frequency components. For example, the polynomial transform given in the last row of Table 1 includes most of the parametric models as special cases for relatively few coefficients. For $\mathcal{K} = \{(0,0),(0,1),(1,0)\}$, the polynomial description reduces to a 2D affine description, while the quadratic model is obtained for

$$\mathcal{K} = \{(0,0),(0,1),(1,0),(0,2),(1,1),(2,0)\}.$$

The number of coefficients can be adapted to the actual complexity of the scene, e.g., such that the error of motion-compensated prediction is sufficiently small [51], [82]. Clearly, for a sufficiently large set of parameters, the polynomial description allows representation of arbitrary motion fields.

By its definition, the displacement (1) can only capture the first-order dynamics of a moving point (constant-velocity motion). However, it has been shown that second-order temporal models capturing both velocity and acceleration can substantially improve the performance of motion-compensated predictive coding [29] and standards conversion [71]. To capture these second-order effects, each motion trajectory must be modeled explicitly. For example, it may be represented by two vectors: instantaneous velocity $\dot{\boldsymbol{x}}$ and acceleration $\ddot{\boldsymbol{x}}$ [13]:

$$\boldsymbol{x}(\tau) \approx \boldsymbol{x}(t) + \dot{\boldsymbol{x}}(t)(\tau - t) + \frac{\ddot{\boldsymbol{x}}(t)}{2}(\tau - t)^2 . \tag{5}$$

Motion estimation is a key technique in image sequence compression and processing, and in computer vision.

Such a temporal modeling can be applied in addition to the spatial modeling described thus far in Table 1. Although representation of motion trajectory fields rather than displacement fields is advantageous in certain applications, larger amounts of motion information must be processed and/or transmitted [13].

In the remainder of this article,

$$\boldsymbol{v}(\boldsymbol{x}) = \dot{\boldsymbol{x}}(t) = (u_t(\boldsymbol{x}), v_t(\boldsymbol{x}))^T$$

denotes the velocity of an image point at $(\boldsymbol{x}, t)$; u_t and v_t are horizontal and vertical velocity components, respectively. We shall omit the subscript or the argument whenever it is clear from the context.

Region of Support for Motion Representation

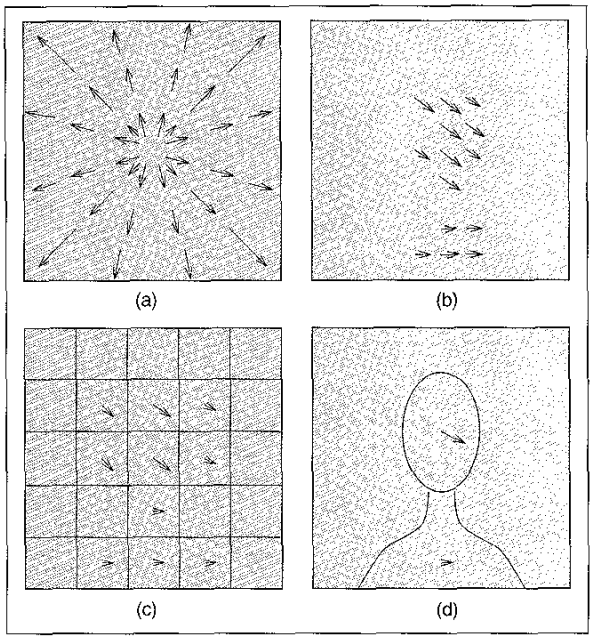
As discussed in the preceding section, 2D motion in an image can be described spatially by a model from Table 1. Models from this table differ in terms of the number of parameters and in terms of the functional dependence of $\boldsymbol{d}(\boldsymbol{x})$ on those parameters. In general, the higher the number of parameters, and thus the higher the function order, the more precise the description of the motion field. At the same time, however, an excessive number of parameters may result in motion "overmodeling" (excessive number of degrees of freedom—important in video processing and computer vision), and increased coding cost (important in video coding). In this case, the motion estimation accuracy may actually decrease. This is due to ill-posedness of motion estimation; for example, no unique solution may exist. The precision of the motion field also depends on the region of support $\mathcal{R} \subset \Lambda$ for the model, i.e., the set of image points to which the model applies. Since the *true* motion field $\tilde{\boldsymbol{d}}$ is rarely purely translational or divergent or exhibits other regularity, the smaller the region of support $\mathcal{R}$, the better the approximation. The quality of approximation for a given motion field $\boldsymbol{d}$ can be measured, for example, by the mean-squared error

$$\mathcal{E}_d^2 = \frac{1}{|\mathcal{R}|} \sum_{\boldsymbol{x} \in \mathcal{R}} \left\| \tilde{\boldsymbol{d}}(\boldsymbol{x}) - \boldsymbol{d}(\boldsymbol{x}) \right\|^2 . \tag{6}$$

Thus, for a given number of parameters the precision of a motion field can be adjusted by choosing a suitable region of support $\mathcal{R}$. Unfortunately, the error $\mathcal{E}_d^2$ can be measured only if $\tilde{\boldsymbol{d}}$ is known, i.e., for computer-generated (synthetic) images. In the following sections, we discuss different support regions proposed in the literature, with both fixed and variable sizes.

Global Motion

The most constrained, yet simplest case is *global motion*, i.e., motion such that all image points are displaced in a uniform manner. The region of support for such models consists of the whole image (Fig. 3a)



3. Various regions of support for a motion model: (a) global, (b) dense, (c) block-based, and (d) region-based. The implicit underlying scene is of "head-and-shoulders" type as captured by the region-based model in (d).

$$\mathcal{R} = \Lambda, \tag{7}$$

where it is assumed that the sampling grid Λ is an orthogonal lattice: $\Lambda = \{1,\ldots,K\} \times \{1,\ldots,L\}$, with K, L being the numbers of columns and lines in the image. The global motion is usually camera-induced, as is the case of a camera pan or zoom. It is the simplest case because the motion of all the image points can be described by a small set of parameters (e.g., affine; Table 1) related to camera parameters [30], [69], [93]. At the same time, this is the most constrained case, because very few parameters describe the motion of all image points; only simple motion fields can be represented in this manner. The global motion model has been extensively used in computer vision, but has only recently found applications in video coding. It has recently been adopted in phase II of the MPEG-4 standard [48], [60], [79]; in sequences with clear camera pan/zoom, substantial rate savings have been achieved compared to standard methods based solely on local block motion estimation.

Motion of Individual Image Points

At the other extreme of the spectrum, the region of support may consist of a single image point (Fig. 3b) [2], [43], [59], [68]:

$$\mathcal{R}_x = \{x\}, \quad x \in \Lambda.$$

Then, motion of each image point can be described by a set of parameters, such as displacement in the case of lin-

Table 1. Motion models.

	2D Model		3D Model		Camera Model
	Number of parameters	Motion field	3D surface function	3D motion	
Translational	2	$d(x) = (a_1, b_1)^T$	Arbitrary	Rigid 3D translation	Orthographic
Affine	6	$d(x) = \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix} x + \begin{pmatrix} a_3 \\ b_3 \end{pmatrix}$	Planar	3D affine	Orthographic
Projective linear	8	$d(x) = \begin{pmatrix} \frac{a_1 + a_2 x + a_3 y}{1 + a_4 x + b_4 y} \\ \frac{b_1 + b_2 x + b_3 y}{1 + a_4 x + b_4 y} \end{pmatrix} - x$	Planar	3D affine	Perspective
Quadratic	12	$d(x) = \begin{pmatrix} a_1 + a_2 x + a_3 y + a_6 x^2 + a_5 xy + a_4 y^2 \\ b_1 + b_2 x + b_3 y + b_6 x^2 + b_5 xy + b_4 y^2 \end{pmatrix}$	Parabolic	3D affine	Orthographic
Sampled	2 per Δ^2 pixels	$d(x) = \sum_{i,j} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix} H(x - \Delta \cdot i, y - \Delta \cdot j)$	"Smooth" as specified by interpolation kernel H		Arbitrary
Polynomial	$2\lvert K\rvert$ Motion-adaptive	$d(x) = \sum_{(i,j) \in K} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix} x^i y^j$	"Smooth" as specified by K		Arbitrary

ear motion, or velocity/acceleration in the case of quadratic trajectories in (5). This pixel-based or *dense* motion representation is the least constrained one since at least two parameters describe movement of each image point, and thus, at least $2 \times K \times L$ parameters are used to represent motion in an image. Consequently, a very large number of motion fields can be represented by all possible combinations of parameter values. At the same time, the method is the most complex due to the number of parameters involved. Although, from a purely computational point of view, it may not be the *most* demanding technique, pixel-based motion estimation is definitely *one* of the most demanding approaches. Dense motion representation has found applications in computer vision, e.g., for the recovery of 3D structure, and in video processing (standards conversion, deblurring, noise reduction). Its direct use in video compression has only shown reasonable success in the form of pel-recursive motion estimation. In this approach, first, a motion vector for each image point is causally predicted from previously estimated and transmitted motion vectors. Then, an update to this prediction is computed while minimizing the motion-compensated prediction error [73]. Clearly, there is no physical reason for causal spatial prediction of motion or for the choice of any particular direction in this prediction. However, noncausal estimation and transmission of pixel-based motion fields has not proved successful to date; the potential gains from a more precise motion description are usually outweighed by the need to transmit thousands of motion parameters. Current work continues in the direction of parametric approximations of dense motion, such as those given in Table 1.

Motion of Regions

Between the two extremes above, one can find methods that apply motion models from Table 1 to image regions. The motivation is to ensure a more accurate modeling [smaller approximation error (6)] of motion fields than in the global motion case and a reduced number of parameters in comparison with the dense motion. The simplest image partitioning is into nonoverlapping rectangular regions $\mathcal{R}_{mn}$ of fixed size $B_K \times B_L$, referred to as blocks, whose union covers the whole image (Fig. 3c):

$$\mathcal{R}_{mn} = \{\boldsymbol{x} = (i,j)^T \in \Lambda : (m-1)B_K < i \le mB_K, \; (n-1)B_L < j \le nB_L\};$$
$$m = 1, \ldots, K / B_K, n = 1, \ldots, L / B_L.$$

Block partitioning with simple translational motion is used today in all digital video compression standards, i.e., H.261, H.263, MPEG-1 and MPEG-2 [45], [46], [49]. Although very successful in hardware implementations due to its simplicity, this model is very imprecise if used on images with general motion, e.g., rotation, zoom, and deformation. To increase the number of degrees of freedom, affine motion models have been proposed in conjunction with the same rectangular partitioning [4], [61]. Such models permit a reduction of the approximation error (6) within each rectangular block, and, at the same time, assure a better match of intensities along motion trajectories.

In video compression, knowledge of motion helps remove temporal data redundancy and, therefore, attain high compression ratios.

Affine motion of rectangular blocks of image points is hardly a precise model for arbitrary motion in image sequences; objects in natural 3D scenes rarely result in rectangular projections onto the image plane. Thus, a more general image partitioning is necessary. The reasoning is that for objects with sufficiently smooth 3D surface and 3D motion, the induced 2D motion fields in the image plane can be suitably described by models from Table 1 if applied to the area of object projection. A natural image partitioning can be provided by the image acquisition process itself. Because several 3D objects typically move in front of a camera, it is straightforward to group all pixels arising from one surface of a 3D object into one region. Not all 3D objects, however, move independently (e.g., car and its driver). Therefore, it is more interesting to find image partitioning such that all image points in a region arise from objects that undergo *one* motion. Then, motion parameters can be estimated from all the image points in a moving region. In both cases, however, a region is described as follows (Fig. 3d)

$$\mathcal{R}_n = \xi_n \subset \Lambda,$$

where all arbitrarily shaped regions ξ_n are non-overlapping and their union covers the complete image. To find a motion-induced partition, certain knowledge about motion is necessary, and conversely, to find motion parameters a partition is needed. The problem is often solved by first applying a segmentation followed by motion estimation [94], or by first estimating motion parameters and then following with segmentation [25]. Since the two processes are not independent, a more appropriate solution is to carry out joint motion estimation and motion-based image segmentation. This can be done in an interleaved fashion, where estimation and segmentation steps alternate [22], or by a simultaneous estimation of segmentation labels and motion parameters at each location [15], [84]. Although the problem is quite difficult, some interesting results have been achieved to date.

It is important to realize that in the case of arbitrarily-shaped regions, motion representation for each region consists of a set of motion parameters and of a region boundary description. Compared to models based on rectangular blocks (block-based models), a region-based

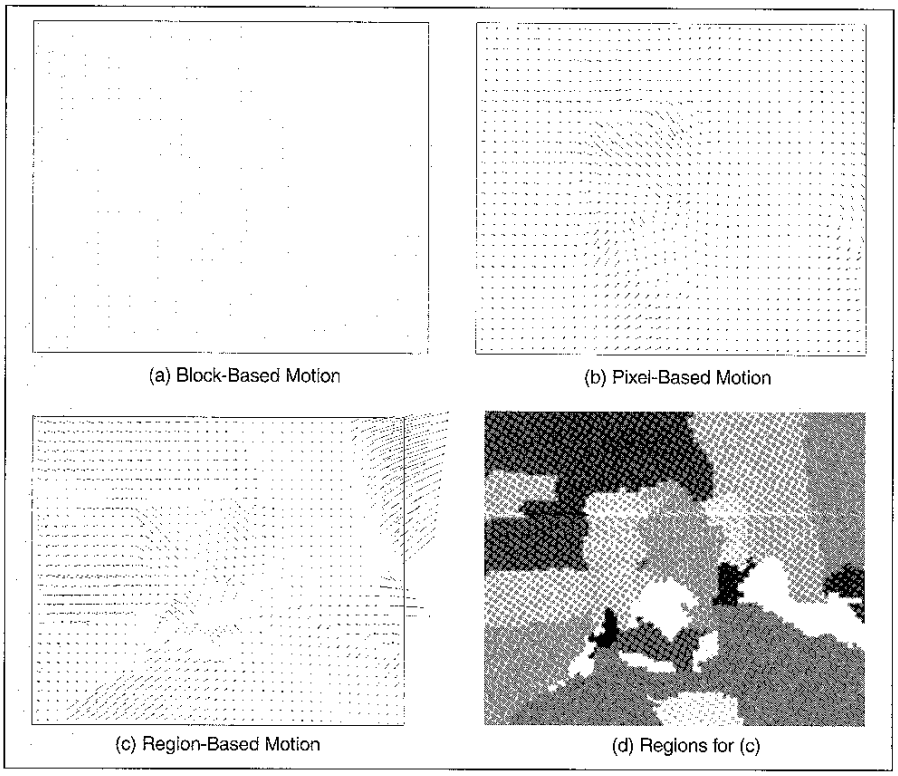
4. Typical videoconferencing images at QCIF (176×144) resolution: (a) "Carphone" (frame 171), and (b) "Miss America" (frame 6).

model has the capacity to perform better image matching at the cost of a more complex representation. This trade-off is well-known in video compression, since both intensity/color residual (image matching error) and motion information must be transmitted over a limited-capacity channel. The difficulty lies in finding a compromise between a precise, but rate-costly, motion representation that gives a small intensity/color residual and coarse motion information that results in an increased residual [25], [35], [85]. This is a very active area of research today. The resulting region-based image description is similar to the description language of computer graphics, thus somewhat merging worlds that, for a long time, have been considered unrelated. This enriches both worlds; for example, objects from natural imagery can be added to computer graphics, while functionalities, so far reserved for computer graphics, can be applied to natural imagery (object-based image manipulation).

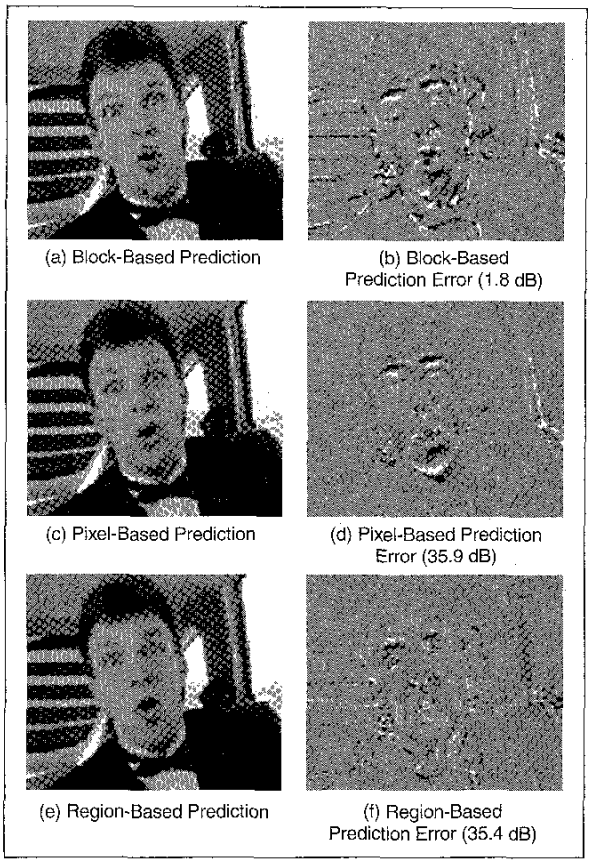
In general, image partitioning based on fixed rectangular blocks under translations is outperformed, in terms of intensity/color residual, by region-based affine motion models [20]. While the former is simple to implement in hardware, the latter requires fairly sophisticated image analysis. As a compromise, partitioning methods based on rectangular variable-size blocks have been investigated [14], [26], [70]. In such an approach, block size is reduced locally wherever a smaller block size improves motion compensation, i.e., reduces the intensity/color residual. Certainly, this increases the complexity of the image partition, but for simple tree-based schemes, such as quad-tree block splitting, the overhead is small [70]. Motion estimation with variable-size blocks has been shown to give substantial gains in practice, and is presently used in the H.263 video compression standard [49]; 16×16 blocks can be individually split into four 8×8 blocks.

Motion compensation using arbitrarily shaped regions has been adopted in the MPEG-4 standard [47]. Although only local translational motion of (partial) blocks is exploited, rather than higher-order parametric motion of complete regions, the approach results in substantial compression gains. Consequently, the visual quality of images is significantly improved around object boundaries (reduced "mosquito effects").

To demonstrate the impact of various regions of support on the motion estimates, Fig. 5 shows results for block-, pixel-, and region-based motion models computed for typical videoconferencing material (Fig. 4). Note the lack of detail due to the low resolution (16×16 blocks) of the block-based approach, but the approximately correct motion of objects. The pixel-based model results in a smooth estimate with more spatial detail, but at the cost of reduced precision, especially within the window of the car. As can be seen, the region-based model assures both accuracy and detail. Although the associated segmentation does not correspond exactly to the objects as perceived by humans, it nevertheless closely delineates object boundaries. The impact of various regions of support on motion-compensated prediction and prediction error is shown in Fig. 6. Note the blocking artifacts for the block-based motion model and the associated



5. Typical motion fields computed from sequence "Carphone" (Fig. 4a) for different regions of support: (a) block-based (16×16 blocks), (b) pixel-based [globally-smooth as in (17)], and (c,d) region-based with affine motion model (Table 1). For details of the region-based algorithm, see [20].



6. Typical motion-compensated prediction and prediction error (×2) for different regions of support (Fig. 5). The numerical measure shown is a peak prediction gain expressed in dB.

4-dB penalty in the peak prediction error as compared with the pixel-based model. The 0.5-dB penalty of the region-based model is small enough to make the region- and pixel-based prediction images virtually identical. This may be important in video processing, since the region-based model would allow object manipulation without significant quality penalty (assuming that semantically meaningful segmentations are available). The region-based model shows a 3.6-dB prediction gain compared to the block motion, however, this is offset to a large extent by the increased amount of data needed for model description (motion parameters and shape of regions). Presently, various approaches to joint motion segmentation and estimation are being developed worldwide. This seems to be a very promising framework both for video compression and video processing.

Hierarchical Motion Models

The practical concept of a variable-size block for motion models [14], [26], [70] can be regarded as a special case of *hierarchical* representation that has often been exploited in computer vision applications [28], [76]. In such a representation, the estimate (in this case motion) can be modeled at multiple levels of detail, making it possible to extract coarse characteristics first and add finer details later [37].

In Fig. 7, we show a multiresolution representation of a motion field in dual form. On the left, a motion field is represented at multiple resolutions and scales at the same time. Note that we follow the definitions of resolution and scale proposed in [91]. On the right, is shown an equivalent representation that can be obtained from the left representation by upsampling and interpolation. This representation is at multiple resolutions, but at a single scale.

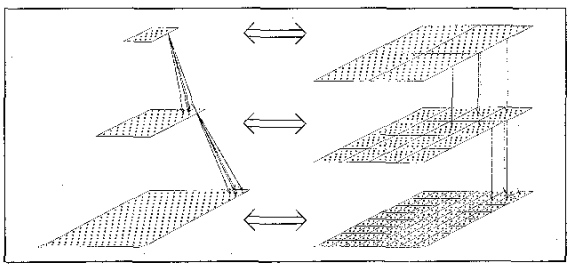
The multiresolution/multiscale representation from Fig. 7 (left) captures coarse motion properties at the higher levels of the pyramid while allowing for progressively more detail when descending in the pyramid. Due to scale change, motion vectors located two pixels apart at the lowest resolution are eight pixels apart at full resolution for a three-level pyramid with dyadic subsampling. Consequently, for models imposing spatial interaction between neighboring motion vectors (e.g., smoothness), longer-range interactions are enforced at lower resolution (higher scale) levels while shorter-range ones are recovered at higher resolution (lower scale) levels of the pyramid.

As mentioned above, the two representations in Fig. 7 are equivalent. The multiscale representation (left) is used in practical estimation algorithms due to its computational efficiency, whereas the single-scale representation (right) is more transparent for certain theoretical considerations, for example, to assure consistency of motion models between different resolutions [40]. This will be discussed further in "Hierarchical Optimization".

The single-scale representation can be also thought of as a motion model with an adjustable region of support. An early example of such a model, where motion parameters are confined to large blocks first and then fine-tuned using smaller blocks, is shown in [3]. This allows early capture of macroscopic motion properties and their subsequent refinement. This approach can be taken further by allowing a spatially nonuniform adjustment of the size of the region of support as is done, for example, in quad-tree splitting [26], [70].

Interdependence of Motion and Image Data

At the very essence of every motion-estimation algorithm lie assumptions about the relationship between motion parameters and image intensity. Let $s_t(x)$ be the image in-



7. Dual representation of a motion field at multiple resolutions: at multiple scales (left) and at a single scale (right). The representations are equivalent since one can be obtained from the other by filtering/downsampling or upsampling/interpolation operators [91].

In video processing, motion analysis is used for standards conversion, noise suppression, and deblurring.

tensity at position (x,t). The usual, and reasonable, assumption made is that image intensity remains constant along the motion trajectory. This assumption implies, among others, that any intensity change is due to motion, that scene illumination is constant, and that object surfaces are opaque (Lambertian surfaces). Although these constraints are almost never satisfied exactly, the constant-intensity assumption approximately describes the dominant properties of natural image sequences, and motion estimation methods based on it usually work well.

Let s be a variable along a motion trajectory. Then, the constant-intensity assumption translates into the following constraint equation

$$\frac{d\mathcal{g}}{ds} = 0. \tag{8}$$

By applying the chain rule, the above equation can be written as the well-known *motion constraint equation* [43]

$$\frac{\partial \mathcal{g}}{\partial x}u + \frac{\partial \mathcal{g}}{\partial y}v + \frac{\partial \mathcal{g}}{\partial t} = (\nabla \mathcal{g})^T \mathbf{v} + \frac{\partial \mathcal{g}}{\partial t} = 0, \tag{9}$$

where

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)^T$$

denotes the spatial gradient and $\mathbf{v} = (u, v)^T$ is the velocity. The above constraint equation, whether in the continuous form or as a discrete approximation, has recently served as the basis for many algorithms estimating linear motion [8], [39], [58]. The same assumption has been used to estimate nonlinear motion trajectories based on multiple images; in [17] the constraint (8) was expressed in the frequency domain, while in [13] it was applied directly to intensities. Note that equation (9) applied at one position (x,t) is underconstrained, since it only determines the component of velocity $\mathbf{v}$ in the direction of image gradient. Due to this so-called aperture problem, additional constraints must be used to uniquely solve for $\mathbf{v}$ [41], [43].

Since color is a very important attribute of images, a possible extension of the above model would be to include chromatic image components into the constraint equation. The motivation is that in areas of uniform intensity, but substantial color detail, the inclusion of a color-based constraint could prove beneficial. Let $\mathcal{g} = (\mathcal{g}_1, \ldots, \mathcal{g}_K)^T$ be a vector of attributes associated with an image; for example, its luminance and two chrominances as defined in the ITU-R 601 recommendation. Then, the constant-intensity and constant-color constraints can be written jointly in a vector form as follows:

$$\frac{\partial \mathcal{g}}{\partial x}u + \frac{\partial \mathcal{g}}{\partial y}v + \frac{\partial \mathcal{g}}{\partial t} = \vec{0}. \tag{10}$$

In general, estimates obtained using this constraint are more reliable than those calculated using (9) due to the additional information exploited. However, although (10) is a vector equation, different components of $\mathcal{g}$ may be closely related and therefore additional constraints may be needed. We will return to these constraints in the next section.

The assumption about intensity constancy is usually only approximately satisfied, but it is particularly violated when scene illumination changes. As an alternative, a constraint based on the spatial gradient's constancy in the direction of motion has been proposed [5], [88]

$$\frac{d\nabla \mathcal{g}}{ds} = \vec{0}. \tag{11}$$

This equation can be rewritten as follows:

$$\begin{bmatrix} \partial^2 \mathcal{g} / \partial x^2 & \partial^2 \mathcal{g} / \partial x \partial y \\ \partial^2 \mathcal{g} / \partial x \partial y & \partial^2 \mathcal{g} / \partial y^2 \end{bmatrix} v + \frac{\partial(\nabla \mathcal{g})}{\partial t} = \vec{0}. \tag{12}$$

It relaxes the constant-intensity assumption, but requires that the amount of dilation and rotation in the image be negligible, a limitation often satisfied in practice[3]. Note that although both (11) and (12) are linear vector equations with two unknowns (u and v), in practice they do not lend themselves to the direct computation of motion, but need to be supported by an appropriate motion model. The primary reason for this is that, in practice, the constraints are not satisfied exactly. Furthermore, the constraint (12) is based on second-order image derivatives. They are difficult to compute reliably due to the high-pass nature of the operator—usually data smoothing must be performed first [21], [88]. To assure smoothness of the resulting motion fields, post-filtering is often applied as well [21]. In order to alleviate problems associated with noise, vanishing gradients, etc., that may lead to ill-posedness, an alternative approach based on the minimization of a norm of $d\nabla \mathcal{g} / ds$ under a smoothness constraint has been developed [89]. The approach has been demonstrated to be very robust in the presence of time-varying illumination.

A different approach to handling nonconstant intensity in the direction of motion is through explicit modeling of the illumination [34], [64]. The approach is promising, although it requires complex minimization since, in addition to the motion field, illumination fields must also be estimated.

The constraints discussed above find different applications in practice. A discrete version of the constant-intensity constraint (9) is often applied in video compression since it results in small motion-compensated prediction error. Although motion can also be computed

based on color using the vector constraint (10), experience shows that the small gains achieved do not justify the substantial increase in complexity. However, motion estimation from color data is useful in video processing tasks (e.g., motion-compensated filtering, resampling), where any motion error may result in visible distortion. Moreover, the vector constraint is interesting for estimating motion from multiple data sources (e.g., range/intensity data). Finally, the gradient-based constraint (11) is often employed in computer vision to find the true motion despite varying illumination.

Estimation Criteria

Various motion representations as well as the relationship between motion and images discussed in the previous section can be used now to formulate an estimation criterion. There is no unique criterion for motion estimation, however. The difficulty in establishing a good criterion is primarily caused by the fact that motion in images is not directly observable[4] and that particular dynamics of intensity in an image sequence may be induced by more than one motion (nonuniqueness). Another problem is that most of the models discussed above are far from ideal. For example, the constant-intensity model expressed through the motion-constraint equation (9) is underconstrained and, at the same time, is often violated due to factors such as noise, nonopaque surface reflections, occlusions, or spatio-temporally varying illumination. Therefore, all attempts to establish suitable criteria for motion estimation require further implicit or explicit modeling of the image sequence.

DFD-Based Criteria

An important class of criteria arising from the constant-intensity assumption (8) aims at the minimization of the following error

$$\varepsilon_{t,\tau}(\boldsymbol{x}) = g_t(\boldsymbol{x}) - \hat{g}_{t,\tau}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{R} \tag{13}$$

where

$$\hat{g}_{t,\tau}(\boldsymbol{x}) = g_\tau(\boldsymbol{x} + \boldsymbol{d}_{t,\tau}(\boldsymbol{x}))$$

is called a motion-compensated prediction of $g_t(\boldsymbol{x})$. If $\mathcal{R}$ is a complete image ($\mathcal{R} = \Lambda$), this error is called a *displaced frame difference* (DFD). However, when $\mathcal{R}$ is a block or an arbitrarily shaped region, the corresponding error is called a *displaced block difference* or a *displaced region difference*. As before, subscripts may be omitted when the notation is clear from the context. Since, in general, $\boldsymbol{d}_{t,\tau}(\boldsymbol{x})$ is real-valued, intensities at positions outside of the sampling lattice Λ must be recovered by a suitable interpolation method. For estimation methods not requiring intensity gradients, C^0 interpolators that assure continuous interpolated intensity are usually sufficient. The case of methods employing intensity gradients will be discussed in the "Regularization" section.

While too small a number of motion parameters may lead to poor motion description, an excessive number may result in motion "overmodeling" and increased coding cost.

Motion fields calculated solely by minimization of the magnitude of the prediction error (13) are, in general, highly sensitive to noise if the number of pixels in the region of support $\mathcal{R}$ is not large compared to the number of motion parameters estimated, or if the region is poorly textured [38, Chapter 16]. However, such a minimization may yield good estimates for parametric motion models with few parameters and a reasonable region size.

To measure the magnitude of the prediction error ε (13), a common choice is an L_p norm. For the L_2 norm, this corresponds to the mean-squared motion-compensated prediction error:

$$J_1(\boldsymbol{d}) = \sum_{\boldsymbol{x} \in \mathcal{R}} (g_t(\boldsymbol{x}) - g_\tau(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})))^2 . \tag{14}$$

This criterion, although very often used, is unreliable in the presence of outliers; even for a single large error $\varepsilon(\boldsymbol{x})$, $\varepsilon^2(\boldsymbol{x})$ is very large and by overcontributing to J_1 it biases the estimate of $\boldsymbol{d}$. Therefore, a more robust mean absolute error criterion

$$J_2(\boldsymbol{d}) = \sum_{\boldsymbol{x} \in \mathcal{R}} |g_t(\boldsymbol{x}) - g_\tau(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x}))| \tag{15}$$

is the criterion of choice in practical video coders today. This criterion is less sensitive to bias due to the piecewise linear dependence of J_2 on ε, and, at the same time, is less involved computationally. Also, the median-squared error criterion

$$J_3(\boldsymbol{d}) = \text{med}_{\boldsymbol{x} \in \mathcal{R}} (g_t(\boldsymbol{x}) - g_\tau(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})))^2 ,$$

due to the use of a median operator, and a criterion based on the (differentiable) Lorentzian function

$$J_4(\boldsymbol{d}) = \sum_{\boldsymbol{x} \in \mathcal{R}} \log(1 + (g_t(\boldsymbol{x}) - g_\tau(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})))^2 / 2\sigma^2),$$

due to the saturation of the error function for outliers (σ is a scale parameter), perform well but require more computations. An interesting discussion of robust estimation criteria in the context of motion estimation can be found in [8].

Frequency-Domain Criteria

Another class of criteria for motion estimation uses transforms, such as the Fourier transform $\mathcal{F}$. For example, due to its shift property, the 2D Fourier transform of an im-

A region-based motion model reflects the physical nature of the world better than a model based on rectangular blocks, yielding a more accurate motion description at the cost of added complexity.

age undergoing spatially-constant motion, i.e., $g_t(\boldsymbol{x}) = g_\tau(\boldsymbol{x} + \boldsymbol{b})$, satisfies

$$\frac{\mathcal{F}\{g_t(\boldsymbol{x})\}}{\mathcal{F}\{g_\tau(\boldsymbol{x})\}} = \exp(j2\pi \boldsymbol{f}^T \boldsymbol{b}),$$

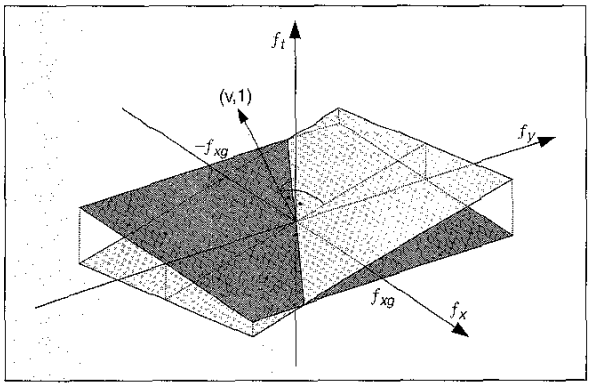
where $\boldsymbol{f} = (f_x, f_y)^T$ denotes 2D spatial frequency. Hence, $\boldsymbol{b}$ can be directly estimated from the phase, but all standard precautions need to be taken to remove phase ambiguity (phase wrapping with the period of 2π). This idea can be extended for constant-velocity motion with $\boldsymbol{b} = \boldsymbol{v}(\tau - t)$ $(\tau > t)$ by noting that for $\tau = 0$,

$$g_t(\boldsymbol{x}) = g_0(\boldsymbol{x}) * \delta(\boldsymbol{x} - \boldsymbol{v}t) \tag{16}$$

where $\delta(\boldsymbol{x})$ is the Dirac delta function and "*" denotes the convolution. Taking the 3D Fourier transform of (16) we can easily see [50] that

$$\frac{\mathcal{F}\{g_t(\boldsymbol{x})\}}{\mathcal{F}\{g_0(\boldsymbol{x})\}} = \delta(\boldsymbol{f}^T \boldsymbol{v} + f_t)$$

with f_t being the temporal frequency. Clearly, in the case of a uniformly translating image under the constant-intensity assumption, the Fourier spectrum is zero everywhere in the 3D spatio-temporal frequency space (f_x, f_y, f_t), except in a plane with an orientation uniquely defined by the velocity $\boldsymbol{v} = (u, v)^T$ (Fig. 8). Then, the estimation of $\boldsymbol{v}$ is reduced to the search for maximum-occupancy planes in the 3D spectrum. This can be done using, e.g., Wigner distribution [50]. Unfortunately, due to the lack of suitable theorems, the above spectral techniques cannot be applied to arbitrary motion models from Table 1.



▲ 8. Nonzero plane in the 3D Fourier spectrum of an image sequence without motion (darker plane) and with a spatially-constant motion v (brighter plane).

Regularization

Instead of dealing with the underconstrained nature of (9) by restricting the motion model to a few parameters, another approach is to explicitly model additional constraints. This can be done by a weak constraint on the estimate itself, reflecting the empirical observation that typical motion fields are spatially smooth. In a pivotal contribution, Horn and Schunck [43] have penalized the squared error resulting from the motion constraint (9) by a smoothness term, thus yielding for continuous $\boldsymbol{x}$

$$J(\boldsymbol{v}) = \iint \left(\nabla^T g(\boldsymbol{x}) \boldsymbol{v}(\boldsymbol{x}) + \frac{\partial g(\boldsymbol{x})}{\partial t} \right)^2 + \lambda \left(\|\nabla u(\boldsymbol{x})\|^2 + \|\nabla v(\boldsymbol{x})\|^2 \right) d\boldsymbol{x}. \tag{17}$$

In practice, when dealing with discrete (sampled) images, the integral is replaced by a summation while the derivatives are replaced by their discrete approximations. In [43], for example, an average of first-order differences computed over a $2 \times 2 \times 2$ cube was used. Since first-order differences are poor derivative approximations, they can severely bias solutions to (17). Using higher-order differences does not necessarily solve the problem. One solution is to use C^1 interpolators that produce continuous intensity and a continuous first derivative of the intensity [55]; the derivative can be found by convolving the intensity with the derivative of the interpolating kernel. In general, small-kernel operators are preferable because the operation must be applied to all pixels. Good results for a discrete variant of (17) [58] have been obtained with a bi-cubic C^1 interpolator developed in [53]. Another interesting solution (and discussion of the problem) can be found in [77], where a joint optimization of derivative and blurring filters in the frequency domain is described.

The smoothness term in (17) regularizes the ill-posed problem of motion estimation (aperture effect), thus turning it into a well-posed problem[5] [87]. Then, the scalar λ balancing the constant-intensity assumption against motion smoothness is termed a regularization constant. For practical reasons, (17) is often expressed in a discrete form, where the first term is replaced by $J_1(\boldsymbol{d})$ (13) and the second term becomes a discrete version of the Laplacian operator [59]. This formulation is often referred to as regularized, although formally it is not because the first term is no longer quadratic, but highly irregular in $\boldsymbol{d}$. This irregularity is due to the dependence of J_1 on $\boldsymbol{d}$ through the image data $g_\tau(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x}))$. Hence, the overall criterion may have multiple minima, keeping the problem ill-posed. This is unlike the formulation in

(17) where both terms are directly quadratic in $\boldsymbol{d}$, thus assuring a unique minimum.

Due to the smoothness term, (17) is often referred to as the *weak membrane model* [9]. In physics, J describes the energy of a membrane extended by v and reaching its minimum in the steady state. The data term accounts for external forces while the smoothness term accounts for elastic forces, with λ being the elasticity constant.

An undesired property of the smoothness term in (17) is that it enforces smooth motion across the whole image, while realistic motion fields exhibit discontinuities at object boundaries. In order to avoid smoothing across object boundaries, intensity edges may be extracted, and the smoothness term may only be applied along those edges [41]. This procedure is motivated by the observation that object boundaries often coincide with intensity edges. For the same reason, an *oriented smoothness constraint* has been proposed [67], [68] that applies smoothing only along the direction of a locally constant intensity. Investigations in [78] show that the oriented-smoothness constraint is the only plausible one among all separable constraints of the same order. It was also proposed to preserve boundaries in motion fields by nonstationary autoregressive modeling [27] or by a line process representing motion discontinuities (smoothness suspended across line elements switched on) [44], [57]. An example of such an adaptively smooth motion field and its associated line process is shown in Fig. 9. Note the improved motion discontinuities at object boundaries. However, since the line process model (discontinuity) is very local, a better object delineation is usually achieved by the region-based approach (Fig. 5c and 5d).

Bayesian Criteria

A general framework for motion-field estimation is provided by Bayesian methods [59]. Let motion field $\boldsymbol{d}$ be a realization of a random field $\boldsymbol{D}$ with given a posteriori probability distribution. An estimate is computed as a special realization of this a posteriori distribution, such as the mean or the mode. When a motion field is to be estimated given the image $\mathcal{G}_{t+1}$ (realization of G_{t+1}) and the previous image $\mathcal{G}_t$, the a posteriori probability distribution can be formally written, using the Bayes rule, as follows

$$P(\boldsymbol{D}=\boldsymbol{d}|G_{t+1}=\mathcal{G}_{t+1};\mathcal{G}_t) = \frac{P(G_{t+1}=\mathcal{G}_{t+1}|\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)\cdot P(\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)}{P(G_{t+1}=\mathcal{G}_{t+1};\mathcal{G}_t)}, \tag{18}$$

where P is a probability measure. In this notation, the semicolon indicates that subsequent variables are only deterministic parameters. For a given pair of images, the denominator is a normalizing constant. The two factors in the numerator are modeled separately based on the observation model and a priori model, respectively. To be more specific, let us consider the maximum a posteriori (MAP) estimate of D. Then, we have

$$\begin{aligned}\hat{\boldsymbol{d}} &= \arg\max_{\boldsymbol{d}} P(\boldsymbol{D}=\boldsymbol{d}|G_{t+1}=\mathcal{G}_{t+1};\mathcal{G}_t)\\ &= \arg\max_{\boldsymbol{d}} P(G_{t+1}=\mathcal{G}_{t+1}|\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)\cdot P(\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)\\ &= \arg\min_{\boldsymbol{d}}\{-\log[P(G_{t+1}=\mathcal{G}_{t+1}|\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)]\\ &\qquad -\log[P(\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)]\}.\end{aligned} \tag{19}$$

The first term denotes the likelihood of an image given a motion field and the previous image. With the given $\boldsymbol{d}$ and $\mathcal{G}_t$, one can compute the motion-compensated prediction of G_{t+1}. A common observation model is to assume that the likelihood is completely specified by a random field ε that models the displaced frame difference (13)

$$P(G_{t+1}=\mathcal{G}_{t+1}|\boldsymbol{D}=\boldsymbol{d};\mathcal{G}_t)=P(E=\varepsilon). \tag{20}$$

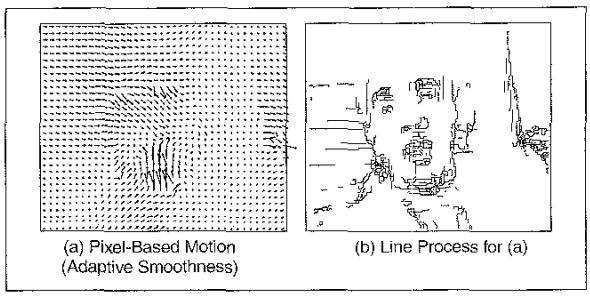
Various distributions have been proposed for $P(E=\varepsilon)$, e.g., zero-mean white Gaussian [56],

$$P(E=\varepsilon)=(2\pi\sigma^2)^{-|\mathcal{R}|/2}\exp\frac{-\sum_{x\in\mathcal{R}}\varepsilon(x)^2}{2\sigma^2}, \tag{21}$$

Laplacian, and segment-wise stationary generalized Gaussian [84]. Additionally, special consideration has been given to violations of the constant-intensity assumption; robust estimation via suppression of outliers [8], modeling of varying illumination [64], and of occlusions [23], [84] are just a few examples.

With the displaced frame difference model (20), the last formulation in (19) can be related to *minimum description length* (MDL) estimation [72]. It is well-known from the coding theory that an optimal encoder attains the code length of $-\log(P(y))$ for coding the sample y of a random variable Y. The code length is also referred to as the *description length* or *self-information* [19]. In hybrid video-coding schemes, motion is transmitted to the receiver along with the displaced frame difference signal.

There are numerous criteria for motion estimation.



▲ *9. Typical motion field computed from sequence "Carphone" (Fig. 4a) for a dense adaptively-smooth (line process) motion model (for example, a combination of (21), (26) and (27)).*

To achieve high compression, lossy transmission can be applied to both or to either one, however at the cost of reducing image quality. Hence, the first term under minimization in (19) denotes the description length for the displaced frame difference, while the second term denotes the description length for the motion field. Therefore, from a data-compression point of view, the MAP estimate $\hat{\boldsymbol{d}}$ is the motion field that minimizes the overall theoretical code length for lossless encoding of a video sequence. This relationship has been used in *rate-constrained* motion estimation [35], [85], where the coding gain resulting from the transmission of a motion vector is related to its cost (description length).

It is worthwhile to note that for a motion field with a uniform a priori distribution, the a posteriori distribution (18) depends on the displaced frame difference only. In other words, from a statistical point of view, methods that minimize the displaced frame difference only perform *maximum likelihood* estimation.

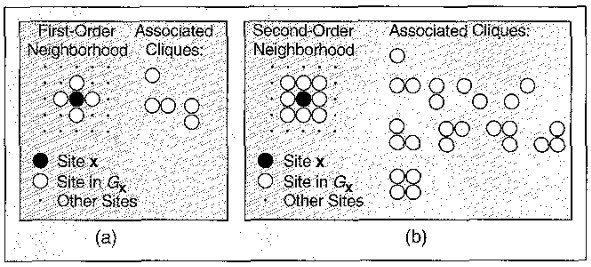
To incorporate prior knowledge into the estimate $\hat{\boldsymbol{d}}$ (19), an a priori distribution $P(\boldsymbol{D}=\boldsymbol{d};\mathcal{J}_t)$ for displacements must be selected. Numerous forms for this distribution have been proposed in the literature. In order to exploit correlation of displacements at adjacent sites, the a priori distribution may favor displacements close to some expected displacement $\bar{\boldsymbol{d}}(x)$ (deterministic, but unknown):

$$P(\boldsymbol{D}(x)=\boldsymbol{d}(x);\mathcal{J}_t)=\frac{1}{2\pi\sigma_d^2}\exp-\frac{\left\|\boldsymbol{d}(x)-\bar{\boldsymbol{d}}(x)\right\|^2}{2\sigma_d^2}$$

The expected displacement $\bar{\boldsymbol{d}}(x)$ may be computed via causal prediction from displacement estimates at adjacent sites and from previous frames. The scalar $\lambda=1/2\sigma_d^2$ may be viewed as a regularization constant balancing a small displaced frame difference and high correlation of motion fields.

Although a spatially causal model is advantageous computationally, spatial causality cannot be justified in displacement fields, unlike for time-dependent signals such as speech. Elegant noncausal models capturing properties such as "smoothness" are provided by Gibbs/Markov random fields [33]. Those random fields require specification of a neighborhood system $\mathcal{G}$, i.e., neighborhood $\mathcal{G}_x \subset \Lambda$ for each site $x \in \Lambda$. Neighborhood systems satisfy the following conditions:

▲ No site is its own neighbor; $x \notin \mathcal{G}_x$, $\forall x \in \Lambda$,



▲ *10. Neighborhoods and cliques for (a) first-order and (b) second-order neighborhood systems.*

A general framework for motion-field estimation is provided by Bayesian methods.

▲ Neighborhood membership is symmetric; $x \in \mathcal{G}_y \Leftrightarrow y \in \mathcal{G}_x$, $\forall x, y \in \Lambda$.

Fig. 10 depicts first- and second-order neighborhood structures, often used in image processing, that consist of the four and eight nearest sites, respectively. Another important element of Gibbs random field definition is a *clique*. A clique c is a subset of Λ, such that any two different elements from c are neighbors (Fig. 10). The set of all cliques will be denoted by $\mathcal{C}$, and $\boldsymbol{d}_C$ will denote a vector of elements of $\boldsymbol{d}$ associated with the sites in c.

A (discrete-valued) Gibbs/Markov random field $\boldsymbol{D}$, with respect to a neighborhood system $\mathcal{G}$, can be defined by the Gibbs distribution

$$P(\boldsymbol{D}=\boldsymbol{d})=\frac{1}{Z}\exp(-H(\boldsymbol{d})), \tag{22}$$

where the *Hamiltonian* H and the *partition sum* Z are defined as follows

$$H(\boldsymbol{d})=\sum_{c\in C}V_c(\boldsymbol{d}_c), \tag{23}$$

$$Z=\sum_{\boldsymbol{d}}\exp(-H(\boldsymbol{d})). \tag{24}$$

In these definitions, V_c may be any real function of variables $\boldsymbol{d}_c$, that is the variables at sites within the clique c. The only, although nontrivial, condition on P to be a well-defined distribution is that $Z \in \mathbb{R}$ must be finite. Continuous-valued Gibbs distributions are defined in the same way except for the partition sum, which is replaced by an integral called the *partition function*.

An important feature of Gibbs/Markov fields is the following Markov property:

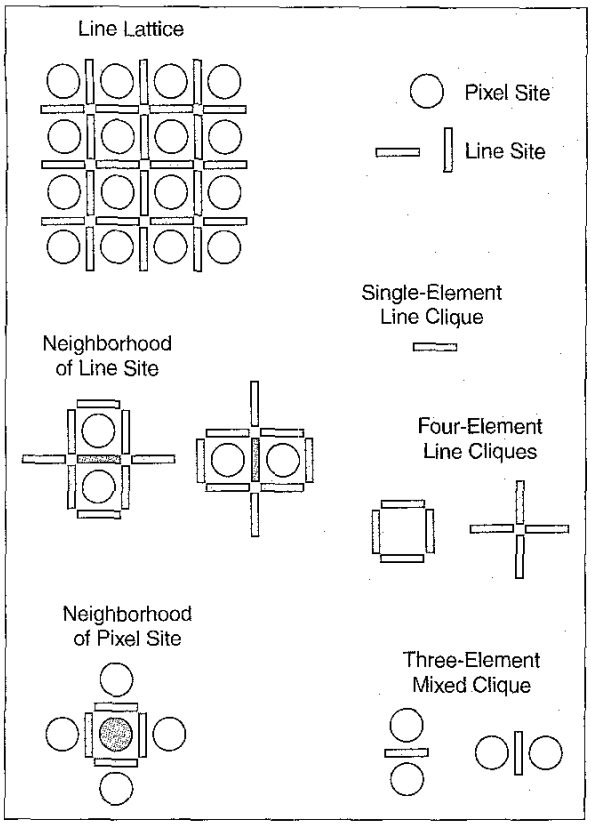
$$P(\boldsymbol{D}(x)=\boldsymbol{d}(x)\,|\,\boldsymbol{D}(y)=\boldsymbol{d}(y),\forall y\neq x)=P(\boldsymbol{D}(x)=\boldsymbol{d}(x)\,|\,\boldsymbol{D}(y)=\boldsymbol{d}(y),\forall y\in G_x).$$

The conditional distribution of a single variable $\boldsymbol{D}(x)$ is completely specified by the variables $\boldsymbol{D}(y)$ within the neighborhood of x. For this reason, the conditional distribution is often referred to as the local characteristic.

As mentioned above, smoothness constraints can be easily modeled by Gibbs/Markov random fields, for example by a first- or second-order Gibbs/Markov random field with the following pair potential

$$V_{\{x,y\}}((\boldsymbol{d}(x),\boldsymbol{d}(y))=\lambda\left\|\boldsymbol{d}(x)-\boldsymbol{d}(y)\right\|^n,\quad\forall\{x,y\}\in C \tag{25}$$

and a vanishing potential for all cliques comprising other number of elements than two. This model yields a discretized version of the weak membrane model (17) for the Euclidean norm and $n=2$.



11. Line field lattice with neighborhood system and some associated cliques.

By constructing the a priori distribution $P(\boldsymbol{D}=\boldsymbol{d};\vartheta_t) = P(\boldsymbol{D}=\boldsymbol{d})$ from (22) and (25), and by combining it with any of the discussed observation models, the MAP criterion (19) becomes well-defined.

Gibbs/Markov random fields also allow explicit modeling of discontinuities. A straightforward way is to model the discontinuities by a binary-valued line field on a dual lattice (Fig. 11). A line field b can be incorporated into the Bayesian formulation by replacing $\boldsymbol{D}$ and $\boldsymbol{d}$ in (19) by $(\boldsymbol{D}, B)$ and $(\boldsymbol{d}, b)$, respectively. The line process does not influence the observation model, while the a priori model for $(\boldsymbol{D}, B)$ is now defined by the following Gibbs/Markov random field:

$$V_{\{x,y,z\}}(\boldsymbol{d}(\boldsymbol{x}),\boldsymbol{d}(\boldsymbol{y}),b(\boldsymbol{z})) = \lambda_d \left\|\boldsymbol{d}(\boldsymbol{x})-\boldsymbol{d}(\boldsymbol{y})\right\|^2 (1-b(\boldsymbol{z})),\ \forall\{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\}\in\mathcal{C}, \tag{26}$$

$$V_{\{z\}}(b(\boldsymbol{z})) = \lambda_b b(\boldsymbol{z}),\quad \forall\{\boldsymbol{z}\}\in\mathcal{C}. \tag{27}$$

The motion smoothness constraint in $V_{\{x,y,z\}}$ is suspended whenever the line process is switched "on" ($b(\boldsymbol{z})=1$). At the same time, $V_{\{z\}}$, defined for single-element cliques, penalizes the introduction of discontinuities. This model can be improved by extending the neighborhood of line elements and considering cliques as depicted in Fig. 11 in order to favor continuity and smoothness of the line process [32], [55]. An extended line process that models discontinuities between motion vectors within two-element motion vector cliques of the second-order neighborhood model has been proposed in [10]. The resulting line process includes four line elements per pixel.

Another way to model discontinuities explicitly is by the introduction of segmentation [16], [82], [83]. The segmentation can be represented by a generic label field $s(\boldsymbol{x})$, where all pixels of the same region possess the same label. Then, the region of support for the smoothness constraint can be limited to the same-label sites and the a priori distribution of the label field can be modeled by a first- or second-order Gibbs/Markov random field:

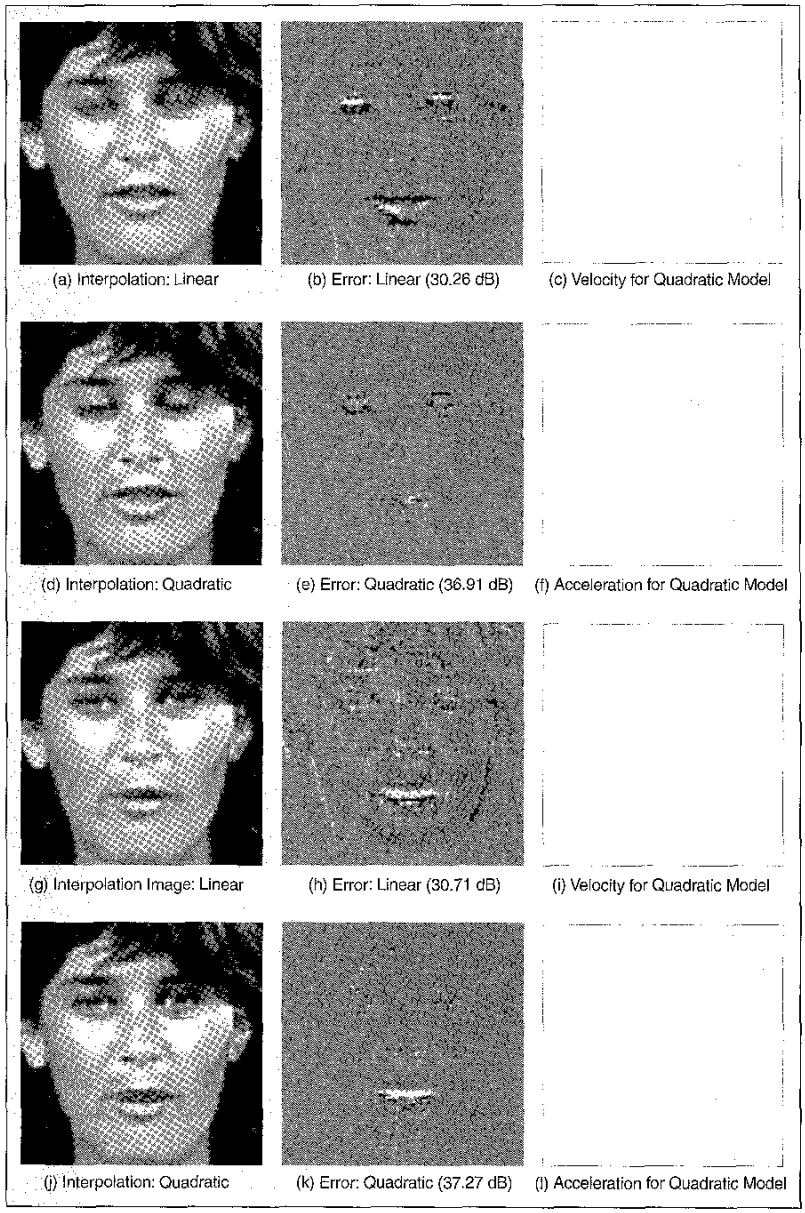
$$V_{\{x,y\}}(\boldsymbol{d}(\boldsymbol{x}),\boldsymbol{d}(\boldsymbol{y}),s(\boldsymbol{x}),s(\boldsymbol{y})) = \lambda_d \left\|\boldsymbol{d}(\boldsymbol{x})-\boldsymbol{d}(\boldsymbol{y})\right\|^2 \delta(s(\boldsymbol{x})-s(\boldsymbol{y})) + \lambda_s (1-\delta(s(\boldsymbol{x})-s(\boldsymbol{y}))),\quad \forall\{\boldsymbol{x},\boldsymbol{y}\}\in\mathcal{C}.$$

The delta function in the first term suspends the smoothness constraint across region boundaries. The second term favors compact regions with short boundaries. Although it is formally similar to the line process, the segmentation offers the advantage that the images are partitioned meaningfully and individual segments tend to undergo continuous motion. Because segments correspond to continuous surfaces of objects in the real world, the segmentation may be considered not only as a tool to estimate discontinuous motion, but also as a valuable information in itself.

It is worthwhile to note that the motion smoothness constraint can be also extended in the temporal direction [7], [8], [74], or in the direction of motion trajectories [83]. An even further extension of this idea is explicit modeling of motion trajectories and estimation of the associated parameters. For example, in [13] motion trajectories are modeled by second-order curves (5) and their parameters (velocity and acceleration) are estimated from several frames using deterministic relaxation. As shown in Fig. 12, such a model can give unquestionable gains in motion-compensated video sequence interpolation; reduction of the reconstruction error due to the inclusion of acceleration is evident both visually (around mouth and eyes) and numerically. Possible applications for this approach are in video standards conversion and in very-lowbit-rate video coding in order to reconstruct missing frames at the receiver (transmission is usually at lower temporal rates).

Search Strategies

With models expressing our knowledge about motion and images specified, and an estimation criterion selected, what remains is to identify an estimation procedure. This procedure involves an optimization of the selected criterion with respect to the parameters of the chosen model. For dense motion fields, both the number of unknowns and the state space for each of them may be large as their



▲ *12. Interpolated and error images as well as velocity and acceleration fields for motion-compensated interpolation of sequence "Miss America" (Fig. 4b) using linear and quadratic trajectories [equation (5)] under global smoothness constraint (17): (a-f) frame 6; and (g-l) frame 14. In each case, five images were used in the estimation; for details of the algorithm, see [13]. The numerical measure shown is an interpolation error expressed in decibels.*

state spaces; an exhaustive search over the complete state space is, with rare exception, computationally prohibitive. Below, we discuss faster search strategies.

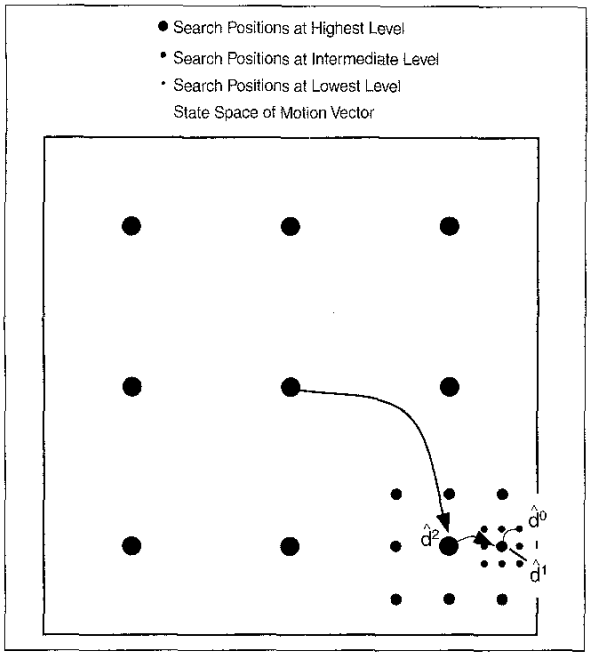
Matching

For a small number of motion parameters and a small state space, the most common search strategy, when minimizing a prediction error, is matching. In this approach, motion-compensated predictions for various motion candidates are compared with the original image within the region of support of the motion model. The candidate yielding the best match for a given criterion becomes the optimal estimate. For small state spaces, as is the case in block-constant motion models used in today's video coding standards, the full state space of each motion vector can be examined. This leads to exhaustive-search block matching.

Assuming that the estimation criterion $J(\boldsymbol{d})$ varies slowly within the state space near the motion estimate sought, *hierarchical search* strategies can be applied to reduce the computational complexity. These strategies aim at successive improvement of the estimate over subsequent levels of the hierarchy. At each level, only a small number of motion candidates is examined. Fig. 13 illustrates a hierarchical search in the case of three-step block matching—the higher the level of hierarchy, the lower the search resolution. A coarse estimate is computed at the highest level as the best match among all motion vector candidates. The state space at this level can be considered as a subsampled version of the motion vector's state space at full resolution. At lower levels, the estimate is successively refined by testing a set of nearby vector candidates. Clearly, hierarchical techniques do not guarantee finding the global optimum. They may be trapped in a local optimum of the estimation criterion, i.e., the reduction in the computational load compromises the quality of motion estimates. Note that in the example of Fig. 13, the motion model is not hierarchical—neither multiscale nor multiresolution (as discussed in "Hierarchical Motion Models"); it is the search strategy that is hierarchical. Other hierarchical search strategies will be discussed in the "Hierarchical Optimization" section.

Relaxation

For dense motion fields based on a noncausal model, simultaneous optimization of all parameters (often hundreds of thousands) may be computationally prohibitive[6]. To alleviate the problem, relaxation techniques construct a sequence of estimates such that consecutive estimates differ in one variable at most. Let's consider the estimation of a dense motion field $\boldsymbol{d}$. A series of motion fields $\boldsymbol{d}^{(0)}, \boldsymbol{d}^{(1)}, \ldots$ is constructed such that any two consecutive estimates $\boldsymbol{d}^{(k-1)}, \boldsymbol{d}^{(k)}$ differ at most at a single site $\boldsymbol{x}_k$, which is either predetermined by some



▲ *13. Three-step search strategy for block matching.*

site-visiting order, e.g., line scanning, or selected randomly. Hence, at each step of the relaxation procedure only the motion vector of a single site needs to be computed while vectors at all other sites remain unchanged.

In a deterministic relaxation, each motion vector is selected from its respective state space with 100% certainty. For example, a new local estimate is computed by minimizing the given criterion. Variables are updated one after another and the criterion is monotonically improved step by step. A well-known deterministic relaxation technique is the method of *iterated conditional modes* (ICM) [6]. For the Bayesian estimation criterion (19), the optimal motion vector is selected from its full state space as follows:

$$\boldsymbol{d}^{(k)}(\boldsymbol{x}_k) = \arg\max_{\boldsymbol{d}(\boldsymbol{x}_k)} P(\boldsymbol{D}(\boldsymbol{x}_k) = \boldsymbol{d}(\boldsymbol{x}_k),$$
$$\boldsymbol{D}(\boldsymbol{x}) = \boldsymbol{d}^{(k-1)}(\boldsymbol{x}), \forall \boldsymbol{x} \neq \boldsymbol{x}_k | G_{t+1} = \mathcal{G}_{t+1}; \mathcal{G}_t).$$

For a Gibbs/Markov field, the above expression is significantly simplified since the conditional probability distribution of a single variable $\boldsymbol{D}(\boldsymbol{x}_k)$ is completely specified by its neighborhood. Then,

$$\boldsymbol{d}^{(k)}(\boldsymbol{x}_k) = \arg\max_{\boldsymbol{d}(\boldsymbol{x}_k)} \pi(\boldsymbol{d}(\boldsymbol{x}_k))$$

where

$$\pi(\boldsymbol{d}(\boldsymbol{x}_k)) = P(\boldsymbol{D}(\boldsymbol{x}_k) = \boldsymbol{d}(\boldsymbol{x}_k) | D(\boldsymbol{x}) = \boldsymbol{d}^{(k-1)}(\boldsymbol{x}),$$
$$\forall \boldsymbol{x} \in \mathcal{G}_{\boldsymbol{x}_k}, G_{t+1} = \mathcal{G}_{t+1}; \mathcal{G}_t).$$

This makes relaxation techniques particularly suitable for the estimation of Gibbs/Markov fields. Computational complexity can be further reduced by selecting $\boldsymbol{d}^{(k)}(\boldsymbol{x}_k)$ from a limited set of motion vector candidates, as proposed in Bayesian block matching [80]. Deterministic relaxation techniques are capable of correcting spurious motion vectors in the initial state $\boldsymbol{d}^{(0)}$. Their major drawback, however, is that they often get trapped in a local optimum near the initial state. Therefore, the availability of a good initial state that may include large-scale information about the optimum is crucial.

The dependence on a good initial state is reduced in stochastic relaxation. In contrast to deterministic techniques, the motion vector under consideration is selected randomly, thus allowing (with a small probability) a momentary deterioration of the criterion. One of the earliest stochastic relaxation techniques was the Metropolis algorithm [62]. In its adaptation to the estimation of motion vectors [55] only two candidate vectors are considered during each relaxation step: the vector from the previous iteration $\boldsymbol{d}^{(k-1)}(\boldsymbol{x}_k)$ and a new candidate vector $\boldsymbol{d}(\boldsymbol{x}_k)$ randomly selected from a single-site state space. Moreover, the site $\boldsymbol{x}_k$ is selected randomly as well. If the new candidate has a larger probability than the previous one, this new vector is accepted; otherwise, the new candidate is accepted with probability

$$Q^{(k)} = \frac{\pi(\boldsymbol{d}(\boldsymbol{x}_k))}{\pi(\boldsymbol{d}^{(k-1)}(\boldsymbol{x}_k))}$$

and the previous estimate $\boldsymbol{d}^{(k-1)}(\boldsymbol{x}_k)$ is kept with probability $1 - Q^{(k)}$. Clearly, the lower the probability of the new candidate $\boldsymbol{d}(\boldsymbol{x}_k)$, the lower the likelihood of its acceptance.

Another important stochastic relaxation technique for Gibbs/Markov random fields is the Gibbs sampler [33] that selects $\boldsymbol{d}^{(k)}(\boldsymbol{x}_k)$ randomly with probability $\pi(\boldsymbol{d}(\boldsymbol{x}_k))$. It can be shown that the estimates $\boldsymbol{d}^{(k)}$ of both the Metropolis algorithm and the Gibbs sampler become independent of the initial state $\boldsymbol{d}^{(0)}$ and maximize the *a posteriori* distribution when k approaches infinity.

In order to find the MAP estimate, these algorithms can be combined with simulated annealing [33], [54]. This optimization technique simulates physical systems of a large number of particles. In equilibrium, such systems follow a Boltzmann distribution

$$P(\boldsymbol{D} = \boldsymbol{d}) = \frac{1}{Z(T)} \exp\left(-\frac{H(\boldsymbol{d})}{k_B T}\right), \tag{28}$$

where k_B denotes the Boltzmann constant, T is the absolute temperature, and H is the Hamiltonian of the system. By writing the a posteriori distribution for $\boldsymbol{D}$ in the form of equation (28) and selecting realizations of $\boldsymbol{D}$ for a monotonically decreasing annealing schedule T_k $(k = 0,1,\ldots)$, annealing of a physical system is simulated. Clearly, for T_k approaching zero, $P(\cdot)$ converges to a Dirac impulse at the MAP estimate. For a sufficiently large T_0 and a sufficiently slow annealing schedule, simulated annealing with either the Metropolis algorithm or the Gibbs sampler can be shown to converge to the MAP estimate. However, the required annealing schedule is extremely

Elegant noncausal models capturing properties such as "smoothness" are provided by Gibbs/Markov random fields.

slow. In practice, simulated annealing is applied with a faster annealing schedule, thus yielding suboptimal results.

HCF Method

Another deterministic optimization technique for Markov random fields that update a single site in each step is the *highest confidence first* (HCF) algorithm [18]. In contrast to relaxation schemes, its site visiting schedule is not fixed, but is driven by the input data. Initially, all the sites are marked as "uncommitted". A new a priori probability is defined based on the original one by modifying the clique potentials V_c of equation (23) as follows:

$$V_c' = \begin{cases} V_c & \text{if all sites in } c \text{ are committed} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, a site does not influence its neighbors until it is committed. The site-visiting order is controlled by the non-positive stability field

$$S_{\boldsymbol{d}}(\boldsymbol{x}) = \begin{cases} H(\boldsymbol{d}_{\text{opt1}}) - H(\boldsymbol{d}_{\text{opt2}}) & \text{if } \boldsymbol{x} \text{ is uncommitted} \\ H(\boldsymbol{d}_{\text{opt1}}) - H(\boldsymbol{d}) & \text{otherwise,} \end{cases}$$

where

$$\boldsymbol{d}_{\text{opt1}} = \arg\min_{\hat{\boldsymbol{d}}} H(\hat{\boldsymbol{d}}) \text{ with } \hat{\boldsymbol{d}}(\boldsymbol{y}) = \boldsymbol{d}(\boldsymbol{y}) \forall \boldsymbol{y} \neq \boldsymbol{x}$$

and

$$\boldsymbol{d}_{\text{opt2}} = \arg\min_{\hat{\boldsymbol{d}} \neq \boldsymbol{d}_{\text{opt1}}} H(\hat{\boldsymbol{d}}) \text{ with } \hat{\boldsymbol{d}}(\boldsymbol{y}) = \boldsymbol{d}(\boldsymbol{y}) \forall \boldsymbol{y} \neq \boldsymbol{x}$$

denote two fields with the best and second-best vectors, respectively, at site $\boldsymbol{x}$ given the vectors at all other sites. At each step, a site with the minimum stability $S_{\boldsymbol{d}}(\boldsymbol{x})$ is updated and marked as committed. The procedure stops when the complete field is committed and the complete stability field is zero. At the beginning, the HCF algorithm selects sites with a "peaked" likelihood function, which is typically the case for highly structured regions. Later, the algorithm includes more and more sites that may not possess such an ideal likelihood function, and thus builds on the neighborhood information of already estimated sites. Since only variables at committed sites influence the optimization, and initially all the sites are uncommitted, the estimated field is independent of the initial state.

Gradient-Based Optimization

Gradient-based techniques require an estimation criterion $J(\boldsymbol{d})$ that is differentiable. Because this criterion depends on motion parameters via the image function g, such as in (14), it is usually approximated by a Taylor expansion with respect to motion parameters. Then, the differentiation of the Taylor-approximated criterion involves differentiation and interpolation of image intensities, already discussed in "Regularization." Due to the Taylor approximation, the model is applicable only in a small vicinity of the desired motion estimate. Therefore, it comes as no surprise that gradient-based estimation is reported to yield accurate estimates only in regions of small motion; the approach fails if motion is large. This can be partially compensated for by low-pass filtering of image sequences. Due to the loss of image detail, however, the accuracy of the estimates suffers. A solution to this problem is to use this less-accurate estimate as an initial state for estimation based on nonfiltered images. This approach is discussed in "Hierarchical Optimization."

Mean-Field Techniques

Much work on the theoretical analysis of Gibbs/Markov random fields has been performed in equilibrium statistical mechanics. Mean-field approaches have proven a powerful tool for the approximation of the mean of such fields.

As outlined in "Relaxation," the MAP estimate of a field governed by a distribution, such as in (28), can be found as its mean for $T \to 0$. A fundamental difference between mean-field annealing and stochastic annealing is that the former is a deterministic procedure and has been demonstrated in practice to converge quickly. Moreover, mean-field optimization does not necessitate annealing, but can be performed at zero or any other temperature right from the start. In many experiments, however, it was found that higher temperatures prove beneficial during the beginning of optimization due to the improved smoothness of the objective function.

The motivation for mean-field techniques is based on the important result from statistical mechanics stating that mean values of a Gibbs/Markov random field can be obtained from its partition function. For this purpose, the partition function Z is considered to be a function of the data. Therefore, mean-field approaches first formulate the desired mean field through the partition function and then approximate the partition function by assuming that this sum is governed by realizations near the equilibrium state. Then, one can benefit from the property that typical optimization criteria exhibit fewer local optima at higher temperatures. Hence, one can design deterministic optimization procedures that find initial estimates at high temperatures, and improve them by decreasing the temperature (annealing).

Let us concisely illustrate the above ideas for the following example [see (17) and (25)]

$$H(\boldsymbol{d}) = \sum_{\boldsymbol{x}} \frac{(g_x(\boldsymbol{x})d_x(\boldsymbol{x}) + g_y(\boldsymbol{x})d_y(\boldsymbol{x}) + \Delta g(\boldsymbol{x}))^2}{2\sigma^2} + \lambda \sum_{\{\boldsymbol{x},\boldsymbol{y}\} \in C} \|\boldsymbol{d}(\boldsymbol{x}) - \boldsymbol{d}(\boldsymbol{y})\|^2,$$

where $(g_x, g_y)^T \approx \nabla g$ denotes the spatial image derivatives, $\Delta g = g_t - g_{t-1}$ is the frame difference and $\boldsymbol{d}(\boldsymbol{x}) = (d_x(\boldsymbol{x}), d_y(\boldsymbol{x}))^T$. Based on the prior distribution (28), the mean field of the horizontal displacement at temperature T is defined by

$$\bar{d}_x(\boldsymbol{x}) = \sum_{\boldsymbol{d}} d_x(\boldsymbol{x}) \frac{1}{Z} \exp\left(-\frac{H(\boldsymbol{d})}{k_B T}\right),$$

where $\bar{d}_x$ denotes the expectation of d_x. This can be rewritten in terms of the partition function as

$$\begin{aligned}\bar{d}_x(\boldsymbol{x}) &= -\frac{\sigma^2 k_B T}{Z} \sum_{\boldsymbol{d}} \frac{\partial^2}{\partial \Delta g(\boldsymbol{x}) \partial g_x(\boldsymbol{x})} \exp\left(-\frac{H(\boldsymbol{d})}{k_B T}\right) \\ &= -\frac{\sigma^2 k_B T}{Z} \frac{\partial^2 Z}{\partial \Delta g(\boldsymbol{x}) \partial g_x(\boldsymbol{x})}. \end{aligned} \tag{29}$$

The mean field of the vertical component $\bar{d}_y$ can be found in the same way. Expression of the mean field through the partition function Z above does not directly provide an optimization procedure, since exact computation of the partition function is, in general, a prohibitive task. Instead, mean-field optimization is based upon the approximation of the partition function, e.g., by saddle-point approximation [31] or mean-field approximation [95]. The latter is based upon the assumption that the influence of the neighboring motion vectors $\boldsymbol{d}(\boldsymbol{y}), \boldsymbol{y} \in \mathcal{G}_{\boldsymbol{x}}$ on a single motion vector $\boldsymbol{d}(\boldsymbol{x})$ can be approximated by the influence of the mean of the neighbors $\bar{\boldsymbol{d}}(\boldsymbol{y})$. Then, we can approximate the Hamiltonian H by H_{mf} as follows:

$$\begin{aligned} H(\boldsymbol{d}) &= \sum_{\boldsymbol{x}} \frac{(\nabla^T g(\boldsymbol{x}) \boldsymbol{d}(\boldsymbol{x}) + \Delta g(\boldsymbol{x}))^2}{2\sigma^2} \\ &\quad + \frac{\lambda}{2} \sum_{\boldsymbol{y} \in \mathcal{G}_{\boldsymbol{x}}} \|\boldsymbol{d}(\boldsymbol{x}) - \boldsymbol{d}(\boldsymbol{y})\|^2 \end{aligned}$$

$$\begin{aligned} H_{\text{mf}}(\boldsymbol{d}) &= \sum_{\boldsymbol{x}} \frac{(\nabla^T g(\boldsymbol{x}) \boldsymbol{d}(\boldsymbol{x}) + \Delta g(\boldsymbol{x}))^2}{2\sigma^2} \\ &\quad + \frac{\lambda}{2} \sum_{\boldsymbol{y} \in \mathcal{G}_{\boldsymbol{x}}} \|\boldsymbol{d}(\boldsymbol{x}) - \bar{\boldsymbol{d}}(\boldsymbol{y})\|^2 \\ &= \sum_{\boldsymbol{x}} H_{\text{mfe}}(\boldsymbol{d}(\boldsymbol{x})). \end{aligned}$$

The above approximation is separable with respect to $\boldsymbol{x}$, so that $H_{\text{mfe}}(\boldsymbol{d}(\boldsymbol{x}))$ depends on the motion vector at site $\boldsymbol{x}$ only. Hence, the partition function Z_{mf} defined by H_{mf} through (24) is also separable in $\boldsymbol{x}$ and can be used in the computation of the mean field via (29):

$$\begin{aligned} Z \approx Z_{\text{mf}} &= \prod_{\boldsymbol{x}} \sum_{\boldsymbol{d}(\boldsymbol{x})} \frac{(\nabla^T g(\boldsymbol{x}) \boldsymbol{d}(\boldsymbol{x}) + \Delta g(\boldsymbol{x}))^2}{2\sigma^2} \\ &\quad + \frac{\lambda}{2} \sum_{\boldsymbol{y} \in \mathcal{G}_{\boldsymbol{x}}} \|\boldsymbol{d}(\boldsymbol{x}) - \bar{\boldsymbol{d}}(\boldsymbol{y})\|^2. \end{aligned}$$

It is worth noting that the mean-field calculation at site $\boldsymbol{x}$ depends on the mean-field values at the neighboring sites. Hence, iterative schemes, often similar to relaxation procedures, are used in optimization. Because mean-field annealing starts at higher temperatures, where local optima are less distinctive, it tends to avoid some of them. However, in contrast to simulated annealing, it cannot guarantee being able to reach the global optimum.

Hierarchical Optimization

The search strategies presented in the preceding sections are often computationally expensive. To lower this computational burden, the hierarchical motion representations discussed in "Hierarchical Motion Models" are often exploited as shown below.

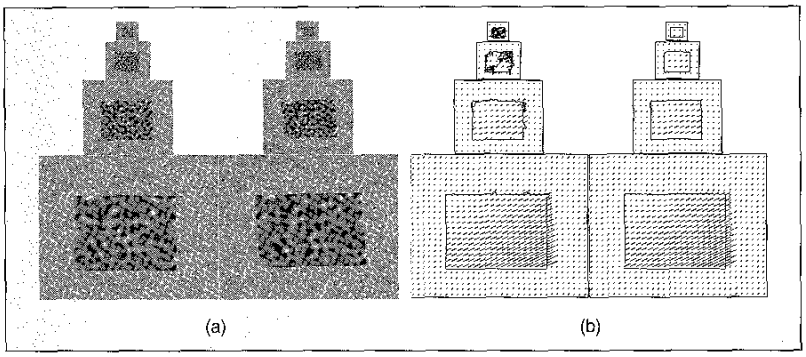
In the multiresolution/multiscale approach (Fig. 7, left), the motion field is represented over a multiresolution pyramid. Usually a dyadic structure is employed. Assuming, for simplicity, that Λ is an orthogonal sampling grid, the grid at level ν can be defined as follows:

$$\Lambda^{(\nu)} = \{\boldsymbol{x} | \boldsymbol{x} \in \Lambda, 2^{-\nu}\boldsymbol{x} \in \mathbb{N}^2\},$$

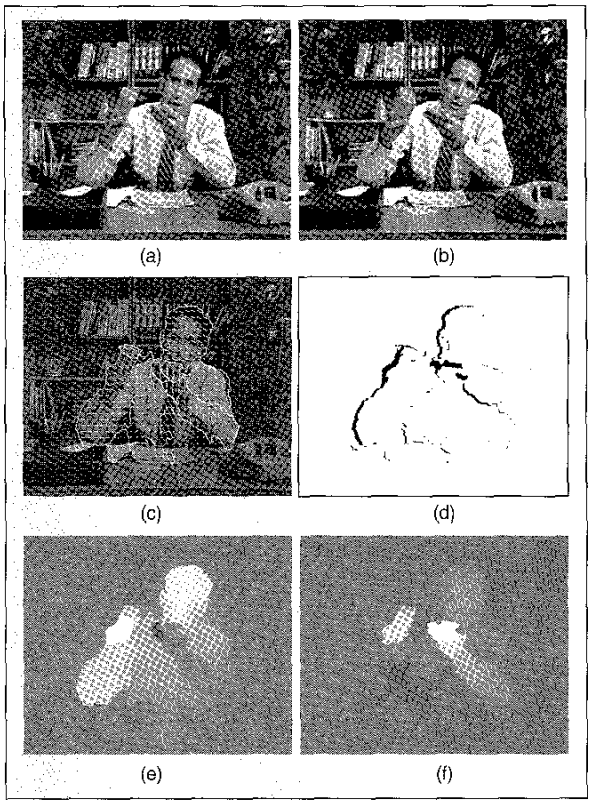
where $\mathbb{N}$ is the set of all integer numbers. The motion field represented on grid $\Lambda^{(\nu)}$ is denoted by $\boldsymbol{d}^{(\nu)}$. Clearly, the grid at the lowest level ($\nu = 0$) is the original image grid, $\Lambda^{(0)} = \Lambda$, and the motion field at that level $\boldsymbol{d}^{(0)}$ is the desired estimate. Likewise, the image sequence may be represented at multiple resolutions by successive low-pass filtering and subsampling.

In a multiresolution/multiscale motion estimation, motion parameters are computed at the lowest resolution first [36]. The computational load of this task is low as compared to the estimation at full resolution because the dimension of the state space of motion vector fields is reduced by $2^{2\nu}$ and the amplitude of motion is reduced by 2^{ν}. Also, due to the scale change between levels of the motion pyramid, as discussed in "Hierarchical Motion Models," methods based on a spatial smoothness constraint [e.g., (17)] converge much faster than their nonhierarchical counterparts. Consequently, a coarse estimate is found very rapidly at the highest level, especially by fast schemes such as the deterministic relaxation. By a suitable projection, this estimate is decreased in scale to serve as an initial state for the motion estimate at the next lower level of the pyramid. More detailed information is added at this level by the same or another optimization scheme. This procedure is repeated until an estimate at the lowest level of the pyramid is found.

While the estimation criterion $H(\boldsymbol{d}) = H^{(0)}(\boldsymbol{d}^{(0)})$ has been formulated for motion fields on the original image grid $\Lambda^{(0)}$, appropriate criteria $H^{(\nu)}(\boldsymbol{d}^{(\nu)})$ for motion at all other levels need to be defined for multiresolution optimization. This has often been performed through heuristic modifications of $H^{(0)}$. However, a more consistent way can be derived by recalling the equivalence between multiresolution representations at multiple scales and at a



▲ *14. (a) Synthetic image pair at multiple resolutions, and (b) multiresolution/multiscale motion estimate (left) and the underlying true motion (right). The true motion field of the inner rectangle is an example of affine motion similar to that shown in Fig. 2b. For more details see [83].*



▲ *15. Results for region-based multiresolution motion estimation applied to the sequence "Salesman": (a) original frame 121, (b) original frame 125, (c) subsampled motion field estimate and segmentation boundaries superimposed over frame 121, (d) occlusion areas, (e) horizontal and (f) vertical motion shown as intensity.*

single scale as illustrated in Fig. 7. A field at any level ν of the left pyramid in Fig. 7 can be reduced in scale by a suitable projection $\phi^{(\nu)}$, and thus be transferred into the equivalent field in the right pyramid. Since all fields are now represented at a single scale, the estimation criterion is naturally formulated at all levels.

$$H^{(\nu)}(\boldsymbol{d}^{(\nu)}) = H(\phi^{(\nu)}(\boldsymbol{d}^{(\nu)}))$$

In the above equation, the same estimation criterion H is applied to all fields $\phi^{(\nu)}(\boldsymbol{d}^{(\nu)})$ in the single scale pyramid [40]. For several forms of the Hamiltonian H, reformulations have been derived for $H^{(\nu)}$ using single scale [40] or multiscale representations of the image sequence [81]. These reformulations do not require explicit scale reduction, and hence, further improve computational efficiency.

Fig. 14 illustrates the results of a multiresolution/multiscale motion estimation for a synthetic image pair. Note that the smooth increase in vector amplitude (affine motion) in the true field is progressively recovered throughout the estimated motion pyramid. On the other hand, Fig. 15 shows results of region-based multiresolution/multiscale motion estimation for a natural image. The underlying estimation criterion is based on Gibbs/Markov random fields; details can be found in [83]. The estimated fields are, in general, consistent with human perception. Nevertheless, the estimates reveal some phenomena frequently observed in motion estimation. Local problems persist in areas of nonunique motion, such as the moving reflections on the table. Furthermore, the segmentation shows a certain degree of inaccuracy in low-texture areas and between segments of similar motion. The latter effect is due to low "motion contrast," a phenomenon similar to low image contrast in intensity-based segmentation. Note that the resulting estimate of uncovered background regions is close to reality. Instead of transferring motion information strictly from the top to the bottom in the estimation pyramid, methods have also been developed that transfer the information in both directions within the pyramid [28], [52]. This approach, unlike the top-to-bottom approach, implements a feedback from higher-resolution estimates to lower-resolution levels, thus facilitating recovery of motion errors at lower resolutions. However, the control strategy in such a bi-directional flow algorithm is not trivial.

Summary and Conclusions

We have reviewed the estimation of 2D motion from time-varying images, paying particular attention to the underlying models, estimation criteria, and optimization strategies. Several parametric and nonparametric models for the representation of motion vector fields and motion trajectory fields have been discussed. For a given region of support, these models determine the dimensionality of the estimation problem as well as the amount of data that has to be interpreted or transmitted thereafter. Also, the interdependence of motion and image data has been addressed. We have shown that even ideal constraints may not provide a well-defined estimation criterion. There-

fore, the data term of an estimation criterion is usually supplemented with a smoothness term that may be expressed explicitly or implicitly via a constraining motion model. We have paid a particular attention to the statistical criteria based on Markov random fields. Because the optimization of an estimation criterion typically involves a large number of unknowns, we have presented several fast search strategies.

We did not cover all possible aspects of 2D motion estimation, but we believe that this article should be helpful to researchers and practitioners working in the fields of video compression and processing, as well as in computer vision. Although the understanding of issues involved in the computation of motion has significantly increased over the last decade, we are still far from generic, robust, real-time motion-estimation algorithms. The selection of the best motion estimator is still highly dependent on the application. Nevertheless, a broad variety of estimation models, criteria, and optimization schemes can be treated in a unified framework presented here, thus allowing a direct comparison and leading to a deeper understanding of the properties of the resulting estimators.

Acknowledgments

We would like to thank Prof. Eric Dubois for reading and commenting upon an early version of this manuscript, and an anonymous reviewer for his diligence in helping us improve the manuscript. In addition, we would like to thank Imax Corp. for providing us with some of the images used in this research.

Christoph Stiller is working in the area of computer vision at the Corporate Research and Advanced Development of Robert Bosch GmbH, Hildesheim, Germany (christoph.stiller@fr.bosch.de). *Janusz Konrad* is an Associate Professor at INRS-Telecommunications, Montreal, Canada (konrad@inrs-telecom.uquebec.ca).

References

[1] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—a review," *Proc. IEEE*, vol. 76, pp. 917-935, Aug. 1988.

[2] J. Aisbett, "Optical flow with intensity-weighted smoothing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 512-522, May 1989.

[3] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *Intern. J. Comput. Vis.*, vol. 2, pp. 283-310, 1989.

[4] C. Bergeron and E. Dubois, "Gradient-based algorithms for block-oriented MAP estimation of motion and application to motion-compensated temporal interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 72-85, Mar. 1991.

[5] M. Bertero, T. Poggio, and V. Torre, "Ill-posed problems in early vision," *Proc. IEEE*, vol. 76, pp. 869-889, Aug. 1988.

[6] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Statistical Society, Ser.B*, vol. 48, pp. 259-302, Aug. 1986.

[7] M. Black, "Combining intensity and motion for incremental segmentation and tracking over long image sequences," in *Proc. European Conf. Computer Vision*, pp. 485-493, May 1992.

[8] M. Black, "Robust incremental optical flow," Ph.D. thesis, Yale University, Department of Computer Science, Sept. 1992.

[9] A. Blake and A. Zisserman, *Visual Reconstruction*,. MIT Press, 1987.

[10] J. Brailean and A. Katsaggelos, "A recursive nonstationary MAP displacement vector field estimation algorithm," *IEEE Trans. Image Processing*, vol. 4, pp. 416-429, Apr. 1995.

[11] J.C. Brailean, R.P. Kleihorst, S.N. Efstratiadis, A.K. Katsaggelos, R.L. Lagendijk, "Noise -reduction filters for dynamic image sequences: a review," *Proc. IEEE*, vol. 83, no. 9, pp. 1272-1292, Sept. 1995.

[12] P. Burt, "Smart sensing within a pyramid vision machine," *Proc. IEEE*, vol. 76, pp. 1006-1015, Aug. 1988.

[13] M. Chahine and J. Konrad, "Estimation and compensation of accelerated motion for temporal sequence interpolation," *Signal Process., Image Commun.*, vol. 7, pp. 503-527, Nov. 1995.

[14] M. Chan, Y. Yu, and A. Constantinides, "Variable size block matching motion compensation with applications to video coding," *IEE Proc. I, Commun. Speech Vis.*, vol. 137, pp. 205-212, Aug. 1990.

[15] M. Chang, M. Sezan, and A. Tekalp, "An algorithm for simultaneous motion estimation and scene segmentation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. V, pp. 221-224, Apr. 1994.

[16] M. Chang, A. Tekalp, and M. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Trans. Image Processing*, vol. 6, pp. 1326-1333, Sept. 1997.

[17] W.-G. Chen, G.B. Giannakis, and N. Nandhakumar, "Spatio-temporal approach for time-varying global image motion estimation," *IEEE Trans. Image Processing*, vol. 10, pp. 1448-1461, Oct. 1996.

[18] P. Chou and C. Brown, "The theory and practice of Bayesian image labelling," *Intern. J. Comput. Vis.*, vol. 4, pp. 185-210, 1990.

[19] T. Cover and J. Thomas, *Elements of information theory*. No. 2 in Wiley Series in Telecommunications, John Wiley & Sons, Inc., 1991.

[20] V.-N. Dang, A.-R. Mansouri, and J. Konrad, "Motion estimation for region-based video coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, pp. 189-192, Oct. 1995.

[21] E. De Micheli, V. Torre, and S. Uras, "The accuracy of the computation of optical flow and of the recovery of motion parameters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 434-447, May 1993.

[22] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Process., Image Commun.*, vol. 3, pp. 23-56, Feb. 1991.

[23] J. Driessen and J. Biemond, "Motion field estimation for complex scenes," in *Proc. SPIE Visual Communications and Image Processing*, vol. 1605, pp. 511-521, Nov. 1991.

[24] E. Dubois and J. Konrad, "Estimation of 2D motion fields from image sequences with application to motion-compensated processing," in *Motion Analysis and Image Sequence Processing*, M. Sezan and R. Lagendijk, Eds., ch. 3, pp. 53-87, Kluwer Academic Publishers, 1993.

[25] F. Dufaux, I. Moccagatta, F. Moscheni, and H. Nicolas, "Vector quantization-based motion field segmentation under the entropy criterion," *J. Vis. Commun. Image Represent.*, vol. 5, pp. 356-369, Dec. 1994.

[26] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, pp. 858-876, June 1995.

[27] S. Efstratiadis and A. Katsaggelos, "Nonstationary AR modeling and constrained recursive estimation of the displacement field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 334-346, Dec. 1992.

[28] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Comput. Vis. Graph. Image Process.*, vol. 43, pp. 150-177, 1988.

[29] M. Foodeci and E. Dubois, "Coding image sequence intensities along motion trajectories using EC-CELP quantization," in *Proc. IEEE Int. Conf. Image Processing*, pp. I.720-I.724, Nov. 1994.

[30] C.-S. Fuh and P. Maragos, "Motion displacement estimation using an affine model for image matching," *Opt. Eng.*, vol. 30, pp. 881-887, July 1991.

[31] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRF's: Surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 401-412, May 1991.

[32] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 609-628, July 1990.

[33] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721-741, Nov. 1984.

[34] M. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," Tech. Rep. 975, MIT Artificial Intelligence Laboratory, June 1987.

[35] B. Girod, "Rate-constrained motion estimation," in *Proc. SPIE Visual Communications and Image Processing*, vol. 2308, pp. 1026-1034, Sept. 1994.

[36] F. Glazer, "Multilevel relaxation in low-level computer vision," in *Multiresolution Image Processing and Analysis*, A. Rosenfeld, Ed., pp. 312-330, Berlin Heidelberg: Springer-Verlag, 1984.

[37] F. Glazer, *Hierarchical motion detection*, Ph.D. thesis, Univ. of Massachusetts, Dept. Comp. Inform. Sci., Feb. 1987.

[38] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1993.

[39] F. Heitz and P. Bouthemy, "Multimodal estimation of discontinuous optical flow using Markov random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1217-1232, Dec. 1993.

[40] F. Heitz, P. Perez, and P. Bouthemy, "Multiscale minimization of global energy functions in some visual recovery problems," *CVGIP: Image Underst.*, vol. 59, pp. 125-134, Jan. 1994.

[41] E. Hildreth, "Computations underlying the measurement of visual motion," *Artif. Intell.*, vol. 23, pp. 309-354, 1984.

[42] B. Horn, *Robot Vision*, Cambridge, MA: MIT Press, 1986.

[43] B. Horn and B. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185-203, 1981.

[44] J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing motion using analog and binary resistive networks," *Computer*, vol. 21, pp. 52-63, Mar. 1988.

[45] ISO/IEC JTC1 IS 11172-2 (MPEG-1), "Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," 1993.

[46] ISO/IEC JTC1 IS 13818-2 (MPEG-2), "Information technology - generic coding of moving pictures and associated audio information," 1996.

[47] ISO/IEC JTC1/SC29/WG11, "MPEG-4 video verification model version 8.0," N1796, July 1997.

[48] ISO/IEC JTC1/SC29/WG11, "MPEG-4 version 2 visual working draft revision 2.0," N1993, Feb. 1998.

[49] ITU-T Recommendation H.263, "Video coding of narrow telecommunication channels at < 64 kbit/s," 1995.

[50] L. Jacobson and H. Wechsler, "Derivation of optical flow using a spatiotemporal-frequency approach," *Comput. Vis. Graph. Image Process.*, vol. 38, pp. 29-65, 1987.

[51] M. Karczewicz, J. Niewegłowski, and P. Haavisto, "Video coding using motion compensation with polynomial motion vector fields," *Signal Process., Image Commun.*, vol. 10, pp. 63-91, July 1997.

[52] Z. Kato, M. Berthod, and J. Zerubia, "Parallel image classification using multiscale Markov random fields," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. V, pp. 137-140, Apr. 1993.

[53] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, pp. 1153-1160, Dec. 1981.

[54] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, May 1983.

[55] J. Konrad, *Bayesian estimation of motion fields from image sequences*, Ph.D. thesis, McGill University, Dept. Elec. Eng., June 1989.

[56] J. Konrad and E. Dubois, "Estimation of image motion fields: Bayesian formulation and stochastic solution," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 1072-1075, Apr. 1988.

[57] J. Konrad and E. Dubois, "Bayesian estimation of discontinuous motion in images using simulated annealing," in *Proc. Conf. Vision Interface VI'89*, pp. 51-60, June 1989.

[58] J. Konrad and E. Dubois, "Comparison of stochastic and deterministic solution methods in Bayesian estimation of 2D motion," *Image Vis. Comput.*, vol. 9, pp. 215-228, Aug. 1991.

[59] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 910-927, Sept. 1992.

[60] J. Konrad and F. Dufaux, "Improved global motion estimation for N3." ISO/IEC JTC1/SC29/WG11, MPEG97/-M3096, Feb. 1998.

[61] C. Labit and H. Nicolas, "Compact motion representation based on global features for semantic image sequence coding," in *Proc. SPIE Visual Communications and Image Processing*, pp. 697-708, 1991.

[62] N. Metropolis, A. Rosenbluth, M. Rosenbluth, H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, June 1953.

[63] A. Mitiche, *Computational analysis of visual motion*, New York: Plenum Press, 1994.

[64] C. Moloney and E. Dubois, "Estimation of motion fields from image sequences with illumination variation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 2425-2428, May 1991.

[65] P. Moulin, R. Krishnamurthy, and J. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Trans. Image Processing*, vol. 6, pp. 1606-1620, Dec. 1997.

[66] H. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Process., Image Commun.*, vol. 1, pp. 117-138, Oct. 1989.

[67] H.-H. Nagel, "On the estimation of optical flow: relations between different approaches and some new results," *Artif. Intell.*, vol. 33, pp. 299-324, 1987.

[68] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 565-593, Sept. 1986.

[69] H. Nicolas and C. Labit, "Global motion identification for image sequence analysis and coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 2825-2828, May 1991.

[70] H. Nicolas and C. Labit, "Region-based motion estimation using deterministic relaxation schemes for image sequence coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 1605, pp. 265-268, Apr. 1992.

[71] A.J. Patti, M.I. Sezan, and A.M. Tekalp, "Digital video standards conversion in the presence of accelerated motion," *Signal Process., Image Commun.*, vol. 6, pp. 213-227, June 1994.

[72] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, no. 2, pp. 416-431, 1983.

[73] J. Robbins and A. Netravali, "Recursive motion compensation: a review," in *Image Sequence Processing and Dynamic Scene Analysis*, E.T.S. Huang, Ed., pp. 76-103, Springer-Verlag, 1983.

[74] H. Shariat and K. Price, "Motion estimation with more than two frames," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 417-434, May 1990.

[75] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Process. Magazine*, vol. 14, pp. 82-100, Sept. 1997.

[76] T. Simchony, R. Chellappa, and Z. Lichtenstein, "Pyramid implementation of optimal-step conjugate-search algorithms for some low-level vision problems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1408-1425, Nov. 1989.

[77] E. Simoncelli, *Distributed representation and analysis of visual motion*, Ph.D. thesis, Massachusetts Institute of Technology, Dept. Electr. Eng. Comp. Sci., Jan. 1993.

[78] M. Snyder, "On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 1105-1114, Nov. 1991.

[79] Special issue on MPEG-4, *IEEE Trans. Circuits Syst. Video Technol.*, Feb. 1997.

[80] C. Stiller, "Motion-estimation for coding of moving video at 8 kbit/s with Gibbs modeled vector field smoothing," in *Proc. SPIE Visual Communications and Image Processing*, vol. 1360, pp. 468-476, Oct. 1990.

[81] C. Stiller, *Modellbasierte Bewegungsschätzung in Bildfolgen*, Ph.D. thesis, Aachen Univ. of Techn., Fortschr.-Ber. VDI, Ser. 10, No. 320, Düsseldorf: VDI-Verlag, 1994 (in German).

[82] C. Stiller, "Object-oriented video coding employing dense motion fields," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. V, pp. 273-276, Apr. 1994.

[83] C. Stiller, "Object-based estimation of dense motion fields," *IEEE Trans. Image Processing*, vol. 6, pp. 234-250, Feb. 1997.

[84] C. Stiller and B. Hürtgen, "Combined displacement estimation and segmentation in image sequences," in *Proc. SPIE/EUROPTO Video Communications and PACS for Medical Applications*, vol. 1977, pp. 276-287, Apr. 1993.

[85] C. Stiller and D. Lappe, "Gain/cost controled displacement-estimation for image sequence coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. IV, pp. 2729-2732, May 1991.

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

[87] A. Tikhonov and A. Goncharsky, eds., *Ill-posed problems in the natural sciences*, Moscow: MIR Publishers, 1987.

[88] O. Tretiak and L. Pastor, "Velocity estimation from image sequences with second order differential operators," in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 16-19, July 1984.

[89] P. Treves and J. Konrad, "Motion estimation and compensation under varying illumination," in *Proc. IEEE Int. Conf. Image Processing*, vol. I, pp. 373-377, Nov. 1994.

[90] R. Tsai and T. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, pp. 1147-1152, Dec. 1981.

[91] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Signal Processing, Prentice Hall, 1995.

[92] Y. Wang and O. Lee, "Active mesh: A feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Processing*, vol. 3, pp. 610-624, Sept. 1994.

[93] S. Wu and J. Kittler, "A differential method for simultaneous estimation of rotation, change of scale and translation," *Signal Process., Image Commun.*, vol. 1, pp. 69-80, May 1990.

[94] S. Wu and J. Kittler, "A gradient-based method for general motion estimation and segmentation," *J. Vis. Commun. Image Represent.*, vol. 4, pp. 25-38, Mar. 1993.

[95] J. Zhang, "The mean-field theory in EM procedures for Markov random fields," *IEEE Transactions on Signal Processing*, vol. 40, pp. 2570-2583, Oct. 1992.

Endnotes

[1]Although in computer vision literature a distinction is often made between 2D motion and optical flow [42], here we will use the term 2D motion to denote either apparent motion or optical flow. This is consistent with video compression terminology where the description of variations in an image is of direct interest regardless of its compliance with the physical cause of that variation.

[2]Detailed information about MPEG standards can be obtained from MPEG home page at www.cselt.it/mpeg.

[3]It is worthwhile noting that even when the constant-intensity assumption is valid, the intensity gradient changes its amplitude under dilation and its direction under rotation.

[4]We can only see the result of motion, not the motion itself; we cannot measure motion directly, but have to use indirect measurements, such as an intensity change.

[5]According to Hadamard's definition, a problem is called *well-posed* if it has a unique solution that continuously depends on the data.

[6]Although pel-recursive methods that are based on a causal model for dense motion are computationally inexpensive, their accuracy is usually lower than that of methods based on noncausal motion models.