3

Estimation of 2-D Motion Fields from Image Sequences with Application to Motion-Compensated Processing

E. Dubois and J. Konrad

INRS-Télécommunications, Verdun, Québec, Canada H3E 1H6

3.1 Introduction

In this chapter we are concerned with the estimation of 2-D motion from time-varying images and with the application of the computed motion to image sequence processing. Our goal for motion estimation is to propose a general formulation that incorporates object acceleration, nonlinear motion trajectories, occlusion effects and multichannel (vector) observations. To achieve this objective we use Gibbs-Markov models linked together by the Maximum *A Posteriori* Probability criterion which results in minimization of a multiple-term cost function. The specific applications of motion-compensated processing of image sequences are prediction, noise reduction and spatiotemporal interpolation.

Estimation of motion from dynamic images is a very difficult task due to its *ill-posedness* [4]. Despite this difficulty, however, many approaches to the problem have been proposed in the last dozen years [27],[24],[40]. This activity can certainly be attributed in large measure to the importance of motion in the processing and coding of image sequences. Below we explain why motion is important in these tasks.

Efficient encoding of time-varying images is essential to provide economical use of network or storage facilities in the provision of video services. Image sequences can be compressed by independent coding of each frame (intraframe coding) or by straightforward extension of spatial coding techniques to three dimensions (e.g., 3-D transform coding). However, such approaches ignore the fact that the majority of new information (innovations) in a time-varying image is carried by the motion. The correlation of image intensity or color is very high along the direction of motion. Thus, the knowledge of motion helps in removing significant interimage redundancy, as is the case in predictive or hybrid (predictive/transform or predictive/subband) coding compensated for motion. In fact, these techniques include most algorithms currently used in videoconferencing [41],[42] and proposed for High Definition Television (HDTV) [21],[60].

Sampling structure conversion and noise reduction are two examples of image sequence processing that can benefit greatly from the knowledge of motion. In the case of conversion, two scenarios are possible. In one situation a missing image must be recovered. Again, due to the high correlation along motion trajectories, motion-compensated interpolation is the most effective tool [53], [15]. In the other situation, only part of an image must be reconstructed. Since some spatial information is available and since motion estimates are occasionally unreliable, methods can be devised which combine "smart" spatial interpolation [50],[18] with motion-compensated interpolation [49]. Noise reduction also benefits from high correlation along motion trajectories. One-dimensional filtering applied along the direction of motion does not alter image features since they are highly correlated, but it attenuates noise since its samples are highly uncorrelated [16],[54].

One remark is in order at this point. Optimal motion fields, in the sense of final image quality, are not necessarily identical for motion-compensated predictive coding and for motion-compensated interpolation. This is due to the fact that the sole task of a predictor is to find the best match of a pixel value given previous images. To find this match any motion trajectory providing minimum prediction error is appropriate. This trajectory does not have to correspond to the real 2-D motion of objects in the image. On the contrary, the task of a temporal interpolator is to calculate a pixel value which is consistent with other pixel values along a motion trajectory so that continuity and smoothness of object motion are preserved. Thus, an estimate of the real 2-D motion is needed. It is a different situation, however, when motion fields need to be transmitted along with the prediction error. Optimal motion fields for prediction require relatively high bit rates since they lack correlation. To reduce the bit rate in this case, joint optimization of the prediction error and of motion correlation can be carried out [56]. Interestingly, since real 2-D motion is usually quite smooth (except for motion discontinuities), it is often estimated by combining a prediction-like error with a smoothness constraint imposed on the estimated motion field. Thus, methods used to estimate 2-D motion for the purpose of temporal interpolation are also applicable to predictive coding with simultaneous transmission of motion data.

The remainder of this chapter is divided into two parts. The first part is devoted to the problem of estimating 2-D motion from image sequences, while the second one treats the motion-compensated processing of dynamic images. The main focus of this contribution is the general formulation of the motion estimation and motion-compensated filtering problems. No experimental results are presented; many can be found in our previous publications cited in the text.

3.2 Estimation of 2-D Motion

In this section we define motion trajectories and other related terms, we formulate the estimation problem and we describe the models used. Then, we incorporate the proposed models into the formulation to obtain a cost function to be minimized. Finally, we discuss suitable solution methods, including the multiresolution approach.

3.2.1 Definitions

We assume that time-varying images from which motion is to be calculated are natural, i.e., obtained by a camera that projects a 3-D scene onto a rectangular portion $\mathcal{W}$ of the 2-D image plane. The special case of synthetic images, such as those generated by computer graphics, is not considered here. We make a further restricting assumption that every point in the image corresponds to a single point in the 3-D scene. Obviously, this assumption is violated for transparent or reflective surfaces. The relative motion of the scene and camera results in 2-D motion on the image plane of projections of scene points and a consequent time variation of the image. Let $\mathbf{x} = (x, y)$ denote the spatial coordinate of an image point [1]. Since coordinates x and y depend on time t, it is useful to consider the trajectory of an image point in a conceptual 3-D xyt space. An example of a 3-D trajectory $(x(t), y(t), t)$ of an image point drawn in such a space is shown in Figure 3.1.

[1] In this chapter, boldface characters indicate vector quantities, including 2-D spatial coordinates and parameter vectors.

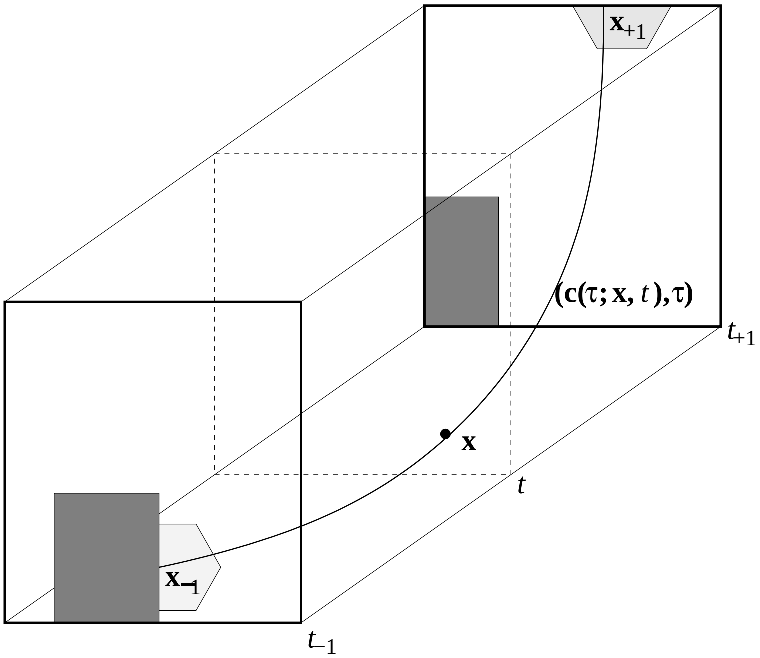

Figure 3.1: Trajectory of a projected scene point in 3-D xyt space; a point is uncovered at t_{-1} and leaves the image at t_{+1}. Note that $(\mathbf{c}(\tau;\mathbf{x},t),\tau) = (x(\tau),y(\tau),\tau)$, where τ is a time instant from the interval (t_{-1},t_{+1}).

Let the function $\mathbf{c}(\tau;\mathbf{x},t)$ mathematically describe a trajectory in the image plane, i.e., let $\mathbf{c}(\tau;\mathbf{x},t)$ be the spatial position at time τ of an image point which at time t was located at $\mathbf{x}$ [15]. $\mathbf{c}(\tau;\mathbf{x},t)$ describes a 2-D trajectory in the image plane, while $(\mathbf{c}(\tau;\mathbf{x},t),\tau) = (x(\tau),y(\tau),\tau)$ describes a 3-D trajectory in the xyt space. Clearly, there is a unique mapping between the two trajectories. For each $\mathbf{x}$ at t, the corresponding trajectory starts at time $t_i(\mathbf{x},t)$ and ends at time $t_f(\mathbf{x},t)$. The trajectory shown in Figure 3.1 corresponds to a point on the hexagon uncovered at time $t_i(\mathbf{x},t) = t_{-1}$ that leaves the image at $t_f(\mathbf{x},t) = t_{+1}$. For $\tau \neq t$, we can define a subset $\mathcal{V}(\tau;t)$ of the image at t consisting of pixels that are visible over the entire time interval between t and τ:

$$\mathcal{V}(\tau;t) = \{\mathbf{x} : t_i(\mathbf{x},t) \leq \tau \leq t_f(\mathbf{x},t)\}. \tag{3.1}$$

For $\tau > t$, $\mathcal{W} - \mathcal{V}(\tau;t)$ is the set of pixels occluded or leaving the image between t and τ, while for $\tau < t$, $\mathcal{W} - \mathcal{V}(\tau;t)$ is the set of pixels newly exposed or introduced into the image between τ and t. A more detailed treatment of motion trajectories in the xyt space can be found in [15].

The shape of the trajectories $\mathbf{c}(\tau;\mathbf{x},t)$ depends on the nature of object motion. Define the instantaneous velocity $\mathbf{v}$ of a pixel at $(\mathbf{x},t)$ as follows:

$$\mathbf{v}(\mathbf{x},t) = \left.\frac{d\mathbf{c}(\tau;\mathbf{x},t)}{d\tau}\right|_{\tau=t}. \tag{3.2}$$

If the velocity $\mathbf{v}$ is constant along the motion trajectory passing through $(\mathbf{x},t)$, then 2-D and 3-D trajectories are linear. In general, however, image points undergo acceleration. If an image point accelerates along a straight line, the 2-D trajectory in the image plane is linear. However, the same point traces out a nonlinear trajectory in the xyt space. In the most complex case an image point may accelerate along a nonlinear 2-D trajectory, thus tracing a nonlinear 3-D trajectory in the xyt space.

Trajectory $\mathbf{c}(\tau;\mathbf{x},t)$ mathematically describes motion of a point in the image plane. This motion may be very complex, thus needing a complex underlying model. Often, however, a simple model, such as the assumption of linear trajectories, is sufficient. For linear motion we use the concept of a *displacement*. Given an image at time t, the displacement field $\mathbf{d}$ is a collection of 2-D vectors describing pixel movements between times t and τ. We define $\mathbf{d}$ only for pixels visible between t and τ

$$\mathbf{d}(\tau;\mathbf{x},t) = \begin{cases} \mathbf{x} - \mathbf{c}(\tau;\mathbf{x},t), & \text{if } \tau < t; \\ \mathbf{c}(\tau;\mathbf{x},t) - \mathbf{x}, & \text{if } \tau > t; \end{cases} \qquad \mathbf{x} \in \mathcal{V}(\tau;t). \tag{3.3}$$

Note that for $\tau > t$, $\mathbf{d}(\tau;\mathbf{x},t)$ is a forward displacement field, while for $\tau < t$ it is a backward displacement field. The displacement field $\mathbf{d}(\tau;\mathbf{x},t)$ can be calculated from the velocity field by integration

$$\mathbf{d}(\tau;\mathbf{x},t) = \int_t^\tau \mathbf{v}(\mathbf{c}(s;\mathbf{x},t),s)ds, \qquad \mathbf{x} \in \mathcal{V}(\tau;t). \tag{3.4}$$

For motion with constant velocity $\mathbf{v}(\mathbf{c}(\tau;\mathbf{x},t),\tau) = \mathbf{v}(\mathbf{x},t)$, the displacement is simply $\mathbf{d}(\tau;\mathbf{x},t) = \mathbf{v}(\mathbf{x},t)\cdot(\tau-t)$. Thus, it follows from (3.3) that

$$\mathbf{c}(\tau;\mathbf{x},t) = \mathbf{x} + \mathbf{v}(\mathbf{x},t)\cdot(\tau-t), \qquad \mathbf{x} \in \mathcal{V}(\tau;t). \tag{3.5}$$

Consequently, for linear motion the task is to find, for each pixel $(\mathbf{x},t)$, the two components v_x and v_y of the velocity $\mathbf{v}(\mathbf{x},t)$.

We are interested here in the estimation of segments of motion trajectories $\mathbf{c}(\tau;\mathbf{x},t)$ for τ over some time interval containing t, where $(\mathbf{x},t)$ is defined on a sampling lattice $\Lambda_c \subset R^3$ [14]. These trajectories, which we refer to as *2-D motion*, correspond to the term *optical flow* often used in computer vision [27]. Usually, they are estimated from intensity or luminance images. However, there is no particular reason to use only luminance for motion estimation, especially if the resulting motion fields are applied to full-color images at the coding or processing stage. Thus, we consider a more general case where motion is estimated from color images. We hope to improve the estimated motion quality in this way, and also to reduce the residual error to be transmitted (for example, in motion-compensated DPCM). Consequently, we develop an approach to motion estimation that is based on vector data. Let $\mathbf{u}$ be the *true* underlying K-component image that is continuous in amplitude and in coordinates, and let $\mathbf{g}$ be the observed K-component image, i.e., $\mathbf{g} = [g_1, g_2, ..., g_K]$. Let each g_k be sampled on lattice Λ_{g_k} $(k = 1, .., K)$. This representation is very general, since individual g_k's $(k = 1, ..., K)$ can be components of a color image such as RGB, $YC1C2$, YIQ, can be derived from a spectral decomposition of an image, e.g., in the form of sub-bands, or even can come from a completely new set of measurements, such as infrared data. For simplicity, we consider only orthorhombic lattices Λ_c and Λ_{g_k} with sampling periods (T_c^h, T_c^v, T_c) and $(T_{g_k}^h, T_{g_k}^v, T_g)$, respectively [14]. Note that we assume identical temporal sampling for all components of $\mathbf{g}$, as is the case in the color representations mentioned above.

To make our formulation complete, we take into account occlusion effects present in dynamic images. We do so by defining an *occlusion field* $o(\mathbf{x},t)$ with samples on the lattice Λ_c. Every occlusion tag o can take one of several possible occlusion states, e.g., moving/stationary (visible), occluded, newly exposed. The number of such states is finite and depends on the number of images used in the estimation. To estimate the trajectories $\mathbf{c}$ in practice, we model them by parametric functions over the time interval of interest. Since parameters of these functions may change rapidly at object boundaries, we permit such a variation by using the concept of motion discontinuity. We define a motion discontinuity field $l(\mathbf{x},t)$ over a union of shifted lattices $\Psi_l = \psi_h \cup \psi_v$, where $\psi_h = \Lambda_c + [0, T_c^v/2, 0]^T$ and $\psi_v = \Lambda_c + [T_c^h/2, 0, 0]^T$ are orthorhombic cosets [14] specifying positions of horizontal and vertical discontinuities, respectively. l is often called a *line field* or a *line process* [19], while a single sample is called a *line element*.

3.2.2 Formulation

Motion present in images is not directly observable; we can only see the effect of motion and not the motion itself. Consequently, it cannot be measured but must be estimated instead. The process of estimating motion is difficult because the problem is *ill-posed* [4], i.e., the solution to the problem may not exist, may be non-unique or may be discontinuous with respect to the data. The most common and thus cumbersome problem is non-uniqueness of the solution. Even in the case of a perfect acquisition system, i.e., noise-free, continuous image obtained with an ideal pinhole camera, many different motion fields may be consistent with the observed data. In practice, however, the imaging system is not ideal, and the obtained data are subject to filtering, nonlinearity, sampling error and noise of a real image acquisition system. This introduces uncertainty and further complicates the estimation process. Finally, in some cases the motion estimation problem may have no solution or it may be discontinuous with respect to the data [33] as in the presence of transparent components and reflective surfaces. Thus, we cannot simply calculate motion. The most we can do is to estimate a motion field that is "best" in a certain sense or "most likely". This usually involves some prior assumptions about motion properties. We address this issue later by specifying appropriate *motion models*.

The uncertainty involved in image acquisition suggests a possible statistical approach to motion estimation. For example, image acquisition and motion attributes can be modeled by stochastic processes and used in a statistical criterion [40]. For certain types of criteria, such as the Minimum Expected Cost (MEC) [36], only a statistical formulation can be considered. For other criteria, such as the Maximum *A Posteriori* Probability (MAP) [40], a cost function results that can also be proposed directly without any recourse to statistical formulation [27],[26].

In the following sections we employ *Markov random field* (MRF) models to both observations $\mathbf{g}$ along motion trajectories and motion descriptors $(\mathbf{c}, o, l)$. Although the usual characterization of a MRF through initial and transitional probabilities is complex and cumbersome, thanks to the *Hammersley-Clifford theorem* [5] it is known that a random field has Markovian properties *if and only if* it is governed by a *Gibbs distribution*. This distribution is uniquely specified by a *neighborhood system*, *cliques* and a *potential function*, as well as by two constants: *partition function* and *natural temperature*[2]. In the case of MAP estimation [35], Gibbs distributions lead directly to a cost function to be optimized that is a weighted sum of certain "energies". In this chapter, we use this probabilistic formulation to arrive at our cost functions; however, similar cost functions could be determined without the application of the statistical framework by the direct use of these energies.

Let subscript t denote the restriction of a field to time instant t, e.g., $\mathbf{g}_t$. Also, let $(\Lambda.)_t$ be a restriction of the lattice $\Lambda.$ to time t and to the window $\mathcal{W}$ (image window in which estimates are sought) simultaneously. Let $\mathcal{I}_t$ denote a finite set of time instants of images $\mathbf{g}$ used to estimate trajectory $\mathbf{c}_t$; $\mathcal{I}_t = \{\tau : \mathbf{g}_\tau$ is used in estimation of $\mathbf{c}_t\}$. Two examples of the set $\mathcal{I}_t$ are shown in Figure 3.2. In order to carry out estimation of trajectories $\mathbf{c}_t$ in practice, they need to belong to a finite-dimensional space. This can be achieved by expressing each trajectory over a given interval of time as a function of a finite number of parameters[3]. Then, we can assume that trajectories $\mathbf{c}_t$ are samples from a vector random field $\mathbf{C}_t$. Similarly, we assume that images $\mathbf{g}_t$ are samples from vector random field $\mathbf{G}_t$, and that occlusion fields o_t and motion discontinuity fields l_t are samples from scalar random fields O_t and L_t, respectively.

Our goal is to determine the *most likely* triplet $(\mathbf{c}_t, o_t, l_t)$ corresponding to the true underlying image $\mathbf{u}$ based on observations $\mathcal{G}_t = \{\mathbf{g}_\tau : \tau \in \mathcal{I}_t\}$. Thus, according to the MAP criterion, we seek the triplet $(\mathbf{c}_t, o_t, l_t)$ that maximizes the conditional probability distribution $p(\mathbf{c}_t, O_t = o_t, L_t = l_t | \mathcal{G}_t)$. In general, this function is a mixed probability density/probability mass function, corresponding to a continuous-valued $\mathbf{c}_t$ and discrete-valued o_t and l_t. We cannot obtain an exact expression for $p(\mathbf{c}_t, O_t = o_t, L_t = l_t | \mathcal{G}_t)$, but based on certain models we can get a good approximation whose maximization will

[2] In order to facilitate understanding of the models, the Appendix at the end of this chapter contains a brief review of Markov random fields, Gibbs distributions and the relationship between them.

[3] Details of this parametrization are provided in Section 3.2.5; until then, explicit dependence of $\mathbf{c}_t$ and $\mathbf{C}_t$ on parameters will not be used.

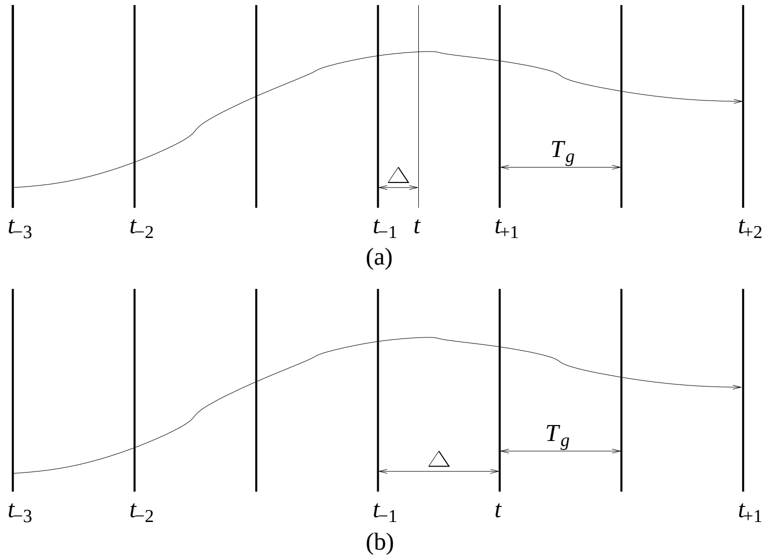

Figure 3.2: Relative temporal position t of motion trajectories with respect to positions of images: (a) $0 < \Delta < T_g$, $\mathcal{I}_t = \{t_{-3}, t_{-2}, t_{-1}, t_{+1}, t_{+2}\}$; (b) $\Delta = T_g$, $\mathcal{I}_t = \{t_{-3}, t_{-2}, t_{-1}, t, t_{+1}\}$. Note that not all images need to be used in the process of motion estimation.

yield excellent motion field estimates.

Our approach is based on the following factorization of the conditional probability distribution using Bayes' rule:

$$\begin{aligned} p(\mathbf{c}_t, O_t = o_t, L_t = l_t | \mathcal{G}_t) \quad = \quad & p(\mathcal{G}_t^n | \mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n}) \cdot \\ & p(\mathbf{c}_t | o_t, l_t, \mathbf{g}_{t_n}) \cdot p(O_t = o_t | l_t, \mathbf{g}_{t_n}) \cdot \\ & p(L_t = l_t | \mathbf{g}_{t_n}) / p(\mathcal{G}_t^n | \mathbf{g}_{t_n}), \end{aligned} \tag{3.6}$$

where $t_n \in \mathcal{I}_t$ is an arbitrarily chosen time instant and $\mathcal{G}_t^n = \{\mathbf{g}_\tau : \tau \in \mathcal{I}_t - \{t_n\}\}$. The MAP estimate is then given by

$$\begin{aligned} (\widehat{\mathbf{c}}_t, \widehat{o}_t, \widehat{l}_t) = \arg \max_{(\mathbf{c}_t, o_t, l_t)} \quad & p(\mathcal{G}_t^n | \mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n}) \cdot p(\mathbf{c}_t | o_t, l_t, \mathbf{g}_{t_n}) \cdot \\ & p(O_t = o_t | l_t, \mathbf{g}_{t_n}) \cdot p(L_t = l_t | \mathbf{g}_{t_n}). \end{aligned} \tag{3.7}$$

Note that since the denominator in (3.6) is not a function of $(\mathbf{c}_t, o_t, l_t)$, it can be ignored in the maximization (3.7).

We now discuss the models that allow us to specify the constituent probabilities in (3.7). The first probability $p(\mathcal{G}_t^n | \mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n})$ is determined by the *structural model* relating motion to the observed image, and the remaining probabilities are determined by the *motion model*.

3.2.3 Observation process

The data from which motion is estimated is usually obtained by an image acquisition system. Thus, the observed image $\mathbf{g}$ is related to the true underlying image $\mathbf{u}$ by a fairly complex random operator called the *observation process*. A model of the observation process is needed to specify the *structural model* with respect to the underlying image $\mathbf{u}$. Typically, the observation process is considered to consist of a nonlinear shift-variant filtering, a random perturbation, and a spatiotemporal sampling. It can be expressed as follows:

$$g_k(\mathbf{x}, t) = \mathcal{H}_k[u_k(\mathbf{x}, t)] \circ n_k(\mathbf{x}, t), \qquad \mathbf{x} \in (\Lambda_{g_k})_t, \; k = 1, ..., K \tag{3.8}$$

where $\mathcal{H}_k$ denotes a nonlinear, shift-variant filter, n_k is a random noise term and $\circ$ is an operator combining the filtered image with noise. Note that in equation (3.8) absence of channel cross-talk is assumed. Usually, the above model is simplified to a linear, shift-invariant filter and additive noise:

$$g_k(\mathbf{x}, t) = h_k(\mathbf{x}, t) * u_k(\mathbf{x}, t) + n_k(\mathbf{x}, t), \qquad \mathbf{x} \in (\Lambda_{g_k})_t,\ k = 1, ..., K \tag{3.9}$$

which is much easier to analyze. In the above equation, h_k's form a bank of K filters used to model the point-spread function (PSF) of the imaging system and $*$ denotes convolution. Despite the above simplification, impulse responses h_k $(k = 1, ..., K)$ are still needed. Unless precise parameters of the imaging system are known or can be estimated, the use of mismatched filters in the model can do more harm than good. To simplify subsequent developments we constrain ourselves, without loss of generality, to the case of no filtering, i.e., $h_k(\mathbf{x}, t) = \delta(\mathbf{x}, t)$ for $k = 1, ..., K$.

The degree of sophistication used in modeling the observation process undoubtedly has an impact on the estimated motion fields. It is not clear, however, whether it is more beneficial to increase the complexity of the observation process above or of the *motion model* discussed below, especially in the view of the usual lack of knowledge of imaging system parameters.

3.2.4 Structural model and matching error

Since motion is not observable and cannot be measured directly, it is imperative that a relationship between motion and image sequence be established. This relationship, called the *structural model*, expresses assumptions about the properties of objects undergoing motion. On one hand, such complex descriptors as size, shape or structure of an object can be used. Alternatively, simple characteristics such as brightness, color or their derivatives can be employed. While the former are usually more stable (constant) under motion than the latter, their evaluation is a problem in itself. They are mostly used in long-range correspondence tasks where stability of features over extended periods of time is essential. The second type of characteristics, insufficiently stable over long intervals, is usually appropriate over short time.

We are interested in resolving the correspondence problem over relatively short time intervals. Under this assumption, the most frequent hypothesis made is that image brightness along motion trajectories is constant [48], [27], which can be expressed as follows:

$$u(\mathbf{x}, t) = u(\mathbf{c}(\tau; \mathbf{x}, t), \tau), \qquad \mathbf{x} \in \mathcal{V}(\tau; t). \tag{3.10}$$

Since chromatic properties of moving objects seem to be at least as stable as their brightness, it has been suggested to use additional color constraints [59],[44]. Recently, a structural model for multiple cues, where representation using luminance and chrominances is a special case, has been proposed in the case of linear motion [38],[34]. This model can be generalized to arbitrary trajectories as follows:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{c}(\tau; \mathbf{x}, t), \tau), \qquad \mathbf{x} \in \mathcal{V}(\tau; t), \tag{3.11}$$

where $\mathbf{u}$ is the true underlying image with K components.

To deal with the departure from brightness or color constancy due to effects such as change of illumination, two approaches have been proposed. In the first approach [22],[45] the difference between both sides of equation (3.10) is modeled by a smooth function representing a slowly varying illumination error. In the second approach [58], [4] it is the spatial variation of intensity rather than its value that is assumed constant along motion trajectories.

Equations (3.10) and (3.11) express the structural model for the underlying image $\mathbf{u}$ over a continuum of spatiotemporal locations. The observed images $\mathbf{g}$, however, are corrupted and sampled versions of $\mathbf{u}$. To take these effects into account assume that we first find an estimate $\widetilde{\mathbf{g}}$ of $\mathbf{u}$ by a suitable operation on the observed data $\mathbf{g}$, i.e., $\widetilde{\mathbf{g}} = \mathcal{P}(\mathbf{g})$. This operation usually involves spatial interpolation, such as proposed in [31],[50], and possibly a noise reduction (linear or non-linear filtering). The effect of the operator $\mathcal{P}$ can be expressed as $\widetilde{\mathbf{g}} = \mathbf{u} + \mathbf{e}$, where $\mathbf{e}$ is a K-component estimation error. Exploiting equation (3.11) we can write:

$$\begin{aligned} \widetilde{\mathbf{g}}(\mathbf{x},t) - \widetilde{\mathbf{g}}(\mathbf{c}(\tau;\mathbf{x},t),\tau) &= \\ \mathbf{e}(\mathbf{x},t) - \mathbf{e}(\mathbf{c}(\tau;\mathbf{x},t),\tau) &= \chi(\mathbf{x},t), \qquad \mathbf{x} \in \mathcal{L}_\tau, \end{aligned} \tag{3.12}$$

where $\mathcal{L}_\tau = (\Lambda_c)_t \cap \mathcal{V}(\tau;t)$ is the set of all pixels on lattice Λ_c at time t visible in the window $\mathcal{W}$ between t and τ. When no occlusion effects are present, then $\mathcal{L}_\tau = (\Lambda_c)_t \cap \mathcal{W} = (\Lambda_c)_t$. $\chi(\mathbf{x},t)$ is a noiselike K-component term governed by a probability distribution depending on the statistics of the estimation error $\mathbf{e}$. We assume that $\chi(\mathbf{x},t)$ is independent of $\widetilde{\mathbf{g}}$.

The fact that pairwise differences of the interpolated image along motion trajectories have the properties of random noise is used to model the conditional distribution $p(\mathcal{G}_t^n|\mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n})$ from equation (3.7). Specifically, we assume that this conditional probability depends only on the variability of $\widetilde{\mathbf{g}}$ along motion trajectories, and that this variability is independent for each distinct trajectory, i.e., for each $\mathbf{x} \in (\Lambda_c)_t$. Thus, we assume that conditional distribution along each trajectory is Gibbsian:

$$p_{\mathbf{x}}(\mathcal{G}_t^n|\mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n}) = \frac{1}{Z_g} e^{-U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t)/\beta_g}, \tag{3.13}$$

where $\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t} = \{\widetilde{\mathbf{g}}(\mathbf{c}(\tau;\mathbf{x},t),\tau) : \tau \in \mathcal{I}_t\}$ is the set of interpolated observations along the trajectory through $(\mathbf{x},t)$, $U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t)$ is an energy function that measures departure of these interpolated observations from the structural model, and Z_g, β_g are constants called partition function and natural temperature, respectively (see the Appendix for a discussion of Gibbs distributions).

The total conditional distribution is hence a product of distributions[4] (3.13)

$$p(\mathcal{G}_t^n|\mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n}) = \prod_{\mathbf{x}\in(\Lambda_c)_t} p_{\mathbf{x}}(\mathcal{G}_t^n|\mathbf{c}_t, o_t, l_t, \mathbf{g}_{t_n}) = \frac{1}{Z'_g} e^{-U'_g(\mathcal{G}_t, \mathbf{c}_t, o_t)/\beta_g}, \tag{3.14}$$

where Z'_g is a product of Z_g's for all $\mathbf{x} \in (\Lambda_c)_t$, and the total energy function is

$$U'_g(\mathcal{G}_t, \mathbf{c}_t, o_t) = \sum_{\mathbf{x}\in(\Lambda_c)_t} U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t). \tag{3.15}$$

In the above formulation, we assume that the trajectory $\mathbf{c}(\tau;\mathbf{x},t)$ extends through the whole set $\mathcal{I}_t$, but that the occlusion function $o(\mathbf{x},t)$ identifies which pixels $(\mathbf{c}(\tau;\mathbf{x},t),\tau)$ are visible at $(\mathbf{x},t)$; only these will contribute to the energy $U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t)$.

The occlusion function $o(\mathbf{x},t)$ is a discrete function with a finite number of states depending on the cardinality of the set $\mathcal{I}_t$. For each spatiotemporal position $(\mathbf{x},t)$ the following set

$$\mathcal{I}_t^{\mathbf{x}} = \{\tau \in \mathcal{I}_t : \mathbf{x} \in \mathcal{L}_\tau\} \tag{3.16}$$

can be defined. This set, called a visibility set, contains time instants from $\mathcal{I}_t$ at which pixel $(\mathbf{x},t)$ is visible. Two examples of occlusion states and visibility sets for 3- and 5-image estimation are given in Table 3.1. In the case of estimation from 5 images, not all possible visibility/non-visibility combinations are defined, but rather only the most likely ones.

[4] Although the representation (3.14) is intuitively plausible, we have given here only qualitative arguments justifying it. A more formal approach can be followed using a spatiotemporal neighborhood system, with potentials that are non-zero only on spatiotemporal cliques oriented along the trajectories, and where the potentials involve spatial interpolation to get $\widetilde{\mathbf{g}}$ values.

State	Description	Visibility set $\mathcal{I}_t^{\mathbf{x}}$ for pixel at $(\mathbf{x}, t)$
$\mathcal{M}$	moving/stationary	$\{t_{-1}, t, t_{+1}\}$
$\mathcal{E}$	exposed	$\{t, t_{+1}\}$
$\mathcal{C}$	covered	$\{t_{-1}, t\}$

(a)

State	Description	Visibility set $\mathcal{I}_t^{\mathbf{x}}$ for pixel at $(\mathbf{x}, t)$
$\mathcal{M}$	moving/stationary	$\{t_{-2}, t_{-1}, t, t_{+1}, t_{+2}\}$
$\mathcal{E}$	exposed in (t_{-1}, t)	$\{t, t_{+1}, t_{+2}\}$
$\mathcal{E}_{-1}$	exposed in (t_{-2}, t_{-1})	$\{t_{-1}, t, t_{+1}, t_{+2}\}$
$\mathcal{C}$	covered in (t, t_{+1})	$\{t_{-2}, t_{-1}, t\}$
$\mathcal{C}_{+1}$	covered in (t_{+1}, t_{+2})	$\{t_{-2}, t_{-1}, t, t_{+1}\}$

(b)

Table 3.1: Tables of occlusion states and visibility sets for the case of $\Delta = T_g$: (a) estimation from 3 images, $\mathcal{I}_t = \{t_{-1}, t, t_{+1}\}$, (b) estimation from 5 images, $\mathcal{I}_t = \{t_{-2}, t_{-1}, t, t_{+1}, t_{+2}\}$. The single-element set $\{t\}$ is not included in (a) because it means that a pixel is visible only at t. Similarly, 11 combinations are excluded from (b) because of trajectory discontinuities.

Note that in $U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t)$, dependence on motion discontinuities l_t has been omitted since the information about such discontinuities is already conveyed through motion field $\mathbf{c}_t$. The line field l_t is not a descriptor of motion, but rather an artificial concept that permits introduction of abrupt changes in trajectory parameters.

There is considerable flexibility in the choice of the form of $U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t)$, based on the structural model. We do this by defining a one-dimensional neighborhood system on $\mathcal{I}_t$, and choosing appropriate cliques and clique potentials. The energy takes the form

$$U_g(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = \sum_{\theta_g \in \Theta_g} V_g(\theta_g, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) \tag{3.17}$$

where θ_g is a clique and Θ_g is the set of all cliques determined by the neighborhood system, and V_g is the potential function. Since the energy function expresses variability along a trajectory, we do not use single-element cliques in any of our models. For two-element cliques of the form $\theta_g = \{\tau_1, \tau_2\}$, a suitable potential function is

$$V_g^{(2)}(\{\tau_1, \tau_2\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = \|\widetilde{\mathbf{g}}(\mathbf{c}(\tau_1; \mathbf{x}, t), \tau_1) - \widetilde{\mathbf{g}}(\mathbf{c}(\tau_2; \mathbf{x}, t), \tau_2)\|^2, \tag{3.18}$$

for some norm $\|\cdot\|$ on the K-dimensional observation space. This potential penalizes deviation from constancy along the trajectory. For the norm, we can use any quadratic form $\mathbf{g}^T\mathbf{M}\mathbf{g}$ where $\mathbf{M}$ is a positive-definite matrix, although a diagonal matrix $\mathbf{M} = diag(m_1, ..., m_K)$, allowing a different relative weighting for each component is probably sufficient for most applications. There are many possibilities for potentials on cliques of three or more elements; some examples are given below.

The flexibility of this model can be illustrated with the following examples that generalize or are equivalent to existing approaches.

A. $\Lambda_c \subset \Lambda_g$, $\mathcal{I}_t^{\mathbf{x}} = \mathcal{I}_t = \{t - T_g, t\}$. This corresponds to the case of motion-compensated prediction of the image field at time t based on the image at $t - T_g$, with no treatment of newly exposed areas. In this case, there is only one two-element clique, $\{t, t-T_g\}$, and by choosing the potential function from (3.18), the energy function is given by

$$U_g^A(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = V_g^{(2)}(\{t-T_g, t\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), \quad \mathbf{x} \in (\Lambda_c)_t. \tag{3.19}$$

Since the above model uses two-element cliques, only linear trajectories need to be used. Then, for velocity $\mathbf{v}(\mathbf{x}, t)$ with trajectory given by (3.5) and for scalar data g, the above energy simplifies to the very well-known squared displaced pixel difference

$$U_g = (g(\mathbf{x}, t) - g(\mathbf{x} - \mathbf{v}(\mathbf{x}, t)T_g, t - T_g))^2. \tag{3.20}$$

This energy has been used extensively in various motion estimation algorithms, of which those presented in [10],[48],[30] are just few examples.

B. $\Lambda_c \cap \Lambda_g = \emptyset$, $t_{-1} < t < t_{+1}$, $\mathcal{I}_t^{\mathbf{x}} = \mathcal{I}_t = \{t_{-1}, t_{+1}\}$. This corresponds to the estimation of motion at a time between two existing observed image fields based on these two fields only, and is applicable to frame rate conversion (temporal interpolation). Occlusions are not considered. Again, there is only one possible two-element clique, $\{t_{-1}, t_{+1}\}$, and a suitable energy function is

$$U_g^B(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = V_g^{(2)}(\{t_{-1}, t_{+1}\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), \quad \mathbf{x} \in (\Lambda_c)_t. \tag{3.21}$$

For linear trajectories this energy has been used in [39] for scalar data and in [38],[34] for vector data.

C. $\Lambda_c \subset \Lambda_g$, $\mathcal{I}_t^{\mathbf{x}} = \mathcal{I}_t = \{t_{-1}, t, t_{+1}\}$. Here, we want to estimate a motion field at the position of an existing image field using three image fields, without consideration of occlusions. This could be applied in interpolative coding, as in MPEG [41]. If we consider only a first-order neighborhood system (i.e., with neighborhoods $\eta_g(t_{-1}) = \{t\}$, $\eta_g(t) = \{t_{-1}, t_{+1}\}$, $\eta_g(t_{+1}) = \{t\}$), then there are two two-element cliques, $\{t_{-1}, t\}$ and $\{t, t_{+1}\}$. A suitable energy function for these cliques is given by

$$\begin{aligned} U_g^{C1}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = \alpha_1 V_g^{(2)}(\{t_{-1}, t\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) \quad &+ \quad \alpha_2 V_g^{(2)}(\{t, t_{+1}\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), \\ &\mathbf{x} \in (\Lambda_c)_t. \end{aligned} \tag{3.22}$$

where α_i's are weights.

If we expand the neighborhood system to second order, then every time position is a neighbor of every other, and we have the additional two-element clique $\{t_{-1}, t_{+1}\}$. We can thus augment the energy (3.22) with the additional potential to give

$$U_g^{C2}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = U_g^{C1}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) + \alpha_3 V_g^{(2)}(\{t_{-1}, t_{+1}\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), \quad \mathbf{x} \in (\Lambda_c)_t. \tag{3.23}$$

The resulting energy is proportional to the sample variance of the three intensities along the trajectory if the weights α_i are equal.

The second-order neighborhood system also includes a three-element clique $\{t_{-1}, t, t_{+1}\}$, for which the potential can be chosen rather arbitrarily. If our structural model permits linear intensity variation along a trajectory, for example, we can consider a potential of the form

$$\begin{aligned} V_g^{(3)}(\{t_{-1}, t, t_{+1}\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) \quad &= \quad \|\widetilde{\mathbf{g}}(\mathbf{x}, t) - ((1-\Delta_t)\widetilde{\mathbf{g}}(\mathbf{c}(t_{-1}; \mathbf{x}, t), t_{-1}) + \\ &\qquad \Delta_t \widetilde{\mathbf{g}}(\mathbf{c}(t_{+1}; \mathbf{x}, t), t_{+1}))\|^2, \end{aligned} \tag{3.24}$$

where $\Delta_t = (t - t_{-1})/(t_{+1} - t_{-1})$. This can be interpreted as the norm of the difference between the observation at time t and the linearly interpolated value based on the observations at times t_{-1} and t_{+1}. The overall energy function can be a linear combination of (3.23) and (3.24):

$$U_g^{C3}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = U_g^{C2}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) + \gamma V_g^{(3)}(\{t_{-1}, t, t_{+1}\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), \ \mathbf{x} \in (\Lambda_c)_t, \tag{3.25}$$

where γ must be determined according to the structural model. Previous work has used $\gamma = 1$ and $\alpha_i = 0, i = 1, ..., K$ [13].

D. $\Lambda_c \subset \Lambda_g$, $\mathcal{I}_t^{\mathbf{x}} \subset \mathcal{I}_t = \{t_{-1}, t, t_{+1}\}$. This example follows Example C, but with consideration of the occlusion states, as given in Table 3.1(a). In this case, if $t_i \notin \mathcal{I}_t^{\mathbf{x}}$, the potential for any two-element clique involving t_i is set to zero. Similarly, if $\mathcal{I}_t^{\mathbf{x}} \neq \mathcal{I}_t$, the potential $V_g^{(3)}$ can be set to zero, since the two-element clique remaining is sufficient. Thus, a general expression for the energy function is [13]

$$U_g^{D}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t) = \begin{cases} U_g^{C3}(\widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), & \text{if } o(\mathbf{x}, t) = \mathcal{M}; \\ V_g^{(2)}(\{t, t_{+1}\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), & \text{if } o(\mathbf{x}, t) = \mathcal{E}; \\ V_g^{(2)}(\{t_{-1}, t\}, \widetilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}_t}, o_t), & \text{if } o(\mathbf{x}, t) = \mathcal{C}; \end{cases} \quad \mathbf{x} \in (\Lambda_c)_t. \tag{3.26}$$

3.2.5 Motion models

So far trajectories $\mathbf{c}_t$ have been written as general functions of $\mathbf{x}$ and t, which could potentially belong to an infinite-dimensional space. To make the estimation problem tractable, we assume that each motion trajectory $\mathbf{c}(\tau; \mathbf{x}, t)$ over a certain time interval containing t can be described by a parametric function $\mathbf{c}^p(\tau; \mathbf{x}, t)$ uniquely identified by the parameter vector $\mathbf{p}$. With this assumption $\mathbf{c}_t$'s belong to a finite-dimensional space, as assumed in previous sections.

In typical images, motion fields are usually smooth functions of spatial position $\mathbf{x}$, except at motion boundaries. Thus, we model trajectories $\mathbf{c}_t^p$ by continuous-valued vector MRFs $\mathbf{C}_t^p$ [46],[35], and also we model occlusions o_t and motion discontinuities l_t by multi-level and binary MRFs O_t [13] and L_t [28],[37], respectively. We ensure the smoothness of $\mathbf{c}_t^p$ and o_t, and continuity of l_t by appropriate choice of Gibbs distribution parameters.

Motion trajectory model

As stated before, in order to uniquely characterize a MRF it is sufficient to specify parameters of its Gibbs distribution. These parameters must be compatible with some expected properties of the random field. Let the *a priori* distribution $p(\mathbf{c}_t^p | o_t, l_t, \mathbf{g}_{t_n})$ for the *motion trajectory model* be defined as follows:

$$p(\mathbf{c}_t^p | o_t, l_t, \mathbf{g}_{t_n}) = \frac{1}{Z_c} e^{-U_c(\mathbf{c}_t^p, l_t, \mathbf{g}_{t_n})/\beta_c}, \tag{3.27}$$

where Z_c and β_c are constants. The energy function U_c is defined as follows:

$$U_c(\mathbf{c}_t^p, l_t, \mathbf{g}_{t_n}) = \sum_{\theta_c \in \Theta_c} V_c(\theta_c, \mathbf{c}_t^p, l_t, \mathbf{g}_{t_n}). \tag{3.28}$$

θ_c is a clique for trajectories $\mathbf{c}_t$ while Θ_c is a set of all such cliques derived from the neighborhood system $\mathcal{N}_c$ defined over lattice Λ_c. V_c is a potential function essential to characterization of the properties of motion field $\mathbf{c}_t$.

To specify the *a priori* motion model, $\mathcal{N}_c$, θ_c and V_c have to be specified. For example, the first-order neighborhood system $\mathcal{N}_c^1$ (Figure 3.3(a)) and two-element horizontal and vertical cliques (Figures 3.3(b) and 3.3(c)) [40] can be chosen. To model the smoothness of trajectories $\mathbf{c}_t^p$, the potential V_c should be such that adjacent similar trajectories belonging to the same object give a small value of V_c (high probability), while dissimilar ones give a large value of V_c. Also, dissimilar adjacent trajectories located across an occlusion boundary should give a small V_c, since they belong to two differently moving objects. We define the potential function V_c as follows:

$$V_c(\theta_c, \mathbf{c}_t^p, l_t, \mathbf{g}_{t_n}) = (\mathbf{p}_i - \mathbf{p}_j)^T \Gamma(\mathbf{g}_{t_n})(\mathbf{p}_i - \mathbf{p}_j) \cdot [1 - l(<\mathbf{x}_i, \mathbf{x}_j>, t)],$$
$$\theta_c = \{\mathbf{x}_i, \mathbf{x}_j\} \in \Theta_c, \tag{3.29}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are parameter vectors for trajectories at $(\mathbf{x}_i, t)$ and $(\mathbf{x}_j, t)$, respectively, and $\Gamma(\mathbf{g}_{t_n})$ is a positive-definite weight matrix depending on the observations. $(<\mathbf{x}_i, \mathbf{x}_j>, t) \in \Psi_l$ denotes the site of a motion discontinuity located between trajectory sites $(\mathbf{x}_i, t)$ and $(\mathbf{x}_j, t)$ which both belong to Λ_c. This potential captures smoothness of the random field $\mathbf{C}_t^p$; for $\mathbf{p}_i = \mathbf{p}_j$, $V_c = 0$.

According to (3.27) the probability of having a particular trajectory at location $(\mathbf{x}, t)$ depends on the occlusion and motion discontinuity fields as well as on the observations. The dependence on local discontinuities is expressed through the multiplicative term $1 - l(<\mathbf{x}_i, \mathbf{x}_j>, t)$ in (3.29) [28],[37]. For a line element "on" $(l(<\mathbf{x}_i, \mathbf{x}_j>, t) = 1)$ no contribution (penalty) is added to the energy U_c. Since the potential V_c is non-negative, such a contribution would lower the probability of $\mathbf{c}(\tau; \mathbf{x}_i, t)$. Thus, a jump in trajectory parameters is not penalized if a motion discontinuity had been detected. Note that in equation (3.27) the energy U_c does not depend on the occlusion field o_t. We assume that sufficient information about occlusion boundaries is conveyed by the field l_t, and thus that o_t is not necessary here. To ensure that this assumption is valid, below we make the *occlusion model* dependent on motion discontinuities l_t.

The dependence of probability (3.27) on the observations is expressed through the weight matrix $\Gamma(\mathbf{g}_{t_n})$ in (3.29). This matrix permits different weighting of horizontal and vertical parameters of $\mathbf{c}_t^p$ as well as of lower and higher order parameters. If Γ is the identity matrix, the Euclidean norm results. In general, $\Gamma(\mathbf{g}_{t_n})$ does not have to be diagonal, and may include off-diagonal entries, thus causing cross-terms to appear in the potential function. Also, it may depend on the observations $\mathbf{g}_{t_n}$ to allow suitable adaptation of motion properties to local image structure. This kind of adaptive smoothness constraint for the case of linear trajectories (3.5) has been proposed in [47].

At this point we must specify the functional form of $\mathbf{c}_t^p$ in order to compute the potential (3.29). In the case of linear trajectory model (equation (3.5)) the parameter vector $\mathbf{p}_i$ takes the following form:

$$\mathbf{p}_i = \begin{bmatrix} v_x(\mathbf{x}_i, t) \\ v_y(\mathbf{x}_i, t) \end{bmatrix}, \tag{3.30}$$

where v_x and v_y are horizontal and vertical components of velocity $\mathbf{v}$, respectively [35]. Thus, Γ becomes a 2×2 matrix.

A natural extension of the linear model is a quadratic trajectory model accounting for acceleration of image points, which can be described by the equation

$$\mathbf{c}(\tau; \mathbf{x}, t) = \mathbf{x} + \mathbf{v}(\mathbf{x}, t) \cdot (\tau - t) + \mathbf{a}(\mathbf{x}, t) \cdot (\tau - t)^2, \quad \mathbf{x} \in \mathcal{V}(\tau; t). \tag{3.31}$$

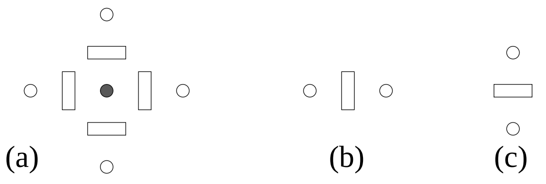

Figure 3.3: (a) First-order neighborhood system $\mathcal{N}_c^1$ for motion field $\mathbf{c}_t$ defined over Λ_c with motion discontinuities l_t defined over Ψ_l ; (b) horizontal clique; (c) vertical clique (empty circles: trajectory positions in $\mathcal{N}_c^1$; filled circle: central position not in $\mathcal{N}_c^1$; rectangles: positions of motion discontinuities).

This new model is based on two velocity (linear) variables $\mathbf{v} = [v_x, v_y]^T$ and two acceleration (quadratic) variables $\mathbf{a} = [a_x, a_y]^T$. Thus, each quadratic trajectory $\mathbf{c}_t^p$ is described by the following parameter vector:

$$\mathbf{p}_i = \begin{bmatrix} v_x(\mathbf{x}_i, t) \\ v_y(\mathbf{x}_i, t) \\ a_x(\mathbf{x}_i, t) \\ a_y(\mathbf{x}_i, t) \end{bmatrix}. \tag{3.32}$$

Γ is a 4×4 matrix. Higher-dimensional approximations to trajectories $\mathbf{c}_t^p$ can be obtained similarly.

Occlusion model

The state space for each occlusion label $o(\mathbf{x}, t)$ consists of a number of states depending on the cardinality of the set $\mathcal{I}_t$ and on the configurations chosen; in Table 3.1 two examples with 3 and 5 states, respectively, are given. Thus, as the *occlusion model* for o_t we use a discrete-valued scalar MRF model described by the following Gibbs distribution

$$P(O_t = o_t | l_t, \mathbf{g}_{t_n}) = \frac{1}{Z_o} e^{-U_o(o_t, l_t, \mathbf{g}_{t_n})/\beta_o}, \tag{3.33}$$

with Z_o and β_o being constants. The energy function U_o is defined as follows:

$$U_o(o_t, l_t, \mathbf{g}_{t_n}) = \sum_{\theta_o \in \Theta_o} V_o(\theta_o, o_t, l_t, \mathbf{g}_{t_n}), \tag{3.34}$$

where θ_o is an occlusion clique and Θ_o is the set of all occlusion cliques derived from the neighborhood system $\mathcal{N}_o$ defined over the lattice Λ_c of motion trajectories. Thus, it is natural to choose the first-order neighborhood system $\mathcal{N}_c^1$ and two-element horizontal and vertical cliques (Figure 3.3). Additionally, we use single-element cliques. Other configurations are possible as well.

The potential function V_o provides a penalty associated with an occlusion state; otherwise energy (3.17) could be reduced freely by a suitable choice of occlusion states. V_o is usually expressed in a tabulated form. It can be expected that a typical occlusion field consists mostly of patches of pixels labeled as visible, and some smaller clusters of pixels labeled as exposed or covered. To penalize the introduction of a label, in addition to the horizontal and vertical cliques from Figure 3.3, single-element cliques are used as well. For the 3-state occlusion field presented in Table 3.1(a), the choice of potential values is similar to the one used in [13] and is shown in Figure 3.4. To ensure that occlusion states get clustered, $V_o = 0$ (high probability) for adjacent identical labels, and a high value of V_o (low probability) for different labels should be specified (Figure 3.4(b)). The boundaries between different patches are expected to be occlusion boundaries, which as mentioned above should coincide with motion discontinuities. To achieve this goal, the dependence of the potential function V_o on field l_t is utilized. V_o is set to 0 whenever two different occlusion states are separated by a motion discontinuity ($l(<\mathbf{x}_i, \mathbf{x}_j>, t) = 1$). Finally, discontinuities in areas of identical occlusion labels are discouraged by assigning a high value to V_o (Figure 3.4(b)). The total potential for a given occlusion label is defined as a sum of potentials for single-element and two-element cliques $V_o = V_A + V_B$ (Figure 3.4).

Motion discontinuity model

The *motion discontinuity model* is based on binary MRF L_t, and is described by the Gibbs distribution

$$P(L_t = l_t | \mathbf{g}_{t_n}) = \frac{1}{Z_l} e^{-U_l(l_t, \mathbf{g}_{t_n})/\beta_l}, \tag{3.35}$$

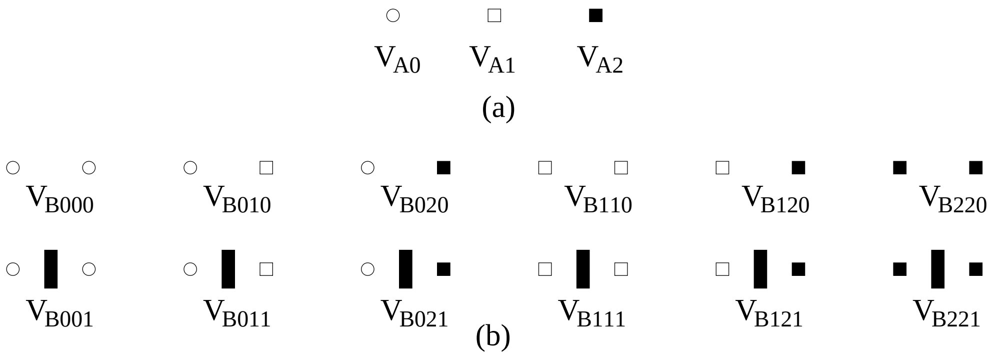

Figure 3.4: Costs associated with various configurations (up to rotation and permutation) of occlusion cliques for the 3-state occlusion field (Table 3.1(a)); (a) single-element cliques; (b) two-element cliques. Typically, $V_{A0} = 0 < V_{A1} = V_{A2}$, $V_{B000} = V_{B110} = V_{B220} = 0 < V_{B010} = V_{B020} \ll V_{B120}$ and $V_{B011} = V_{B021} = 0 < V_{B001} < V_{B111} = V_{B221} \ll V_{B121}$ (circle: moving/stationary pixel; open square: exposed pixel; filled square: covered pixel; filled rectangle: line element "on").

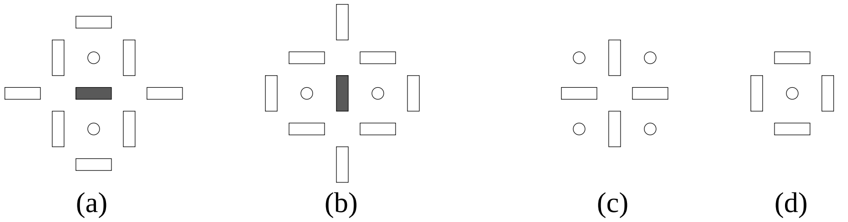

Figure 3.5: Neighborhood system $\mathcal{N}_l$ for motion discontinuity field l_t defined over Ψ_l ; (a) centered on a horizontal discontinuity; (b) centered on a vertical discontinuity; (c) four-element cross-shaped clique; (d) four-element square-shaped clique (empty rectangles: positions in $\mathcal{N}_l^2$; filled rectangle: central position not in $\mathcal{N}_l^2$; circles: positions of trajectories.)

with Z_l and β_l being the usual constants. U_l is the energy function defined as follows:

$$U_l(l_t, \mathbf{g}_{t_n}) = \sum_{\theta_l \in \Theta_l} V_l(l_t, \theta_l, \mathbf{g}_{t_n}), \tag{3.36}$$

where θ_l is a motion discontinuity clique and Θ_l is the set of all such cliques derived from neighborhood system $\mathcal{N}_l$ defined over Ψ_l. The potential function V_l provides a penalty associated with introduction of a discontinuity.

In order to model continuity of motion boundaries, a sufficiently large neighborhood system $\mathcal{N}_l$ for the "dual" sampling structure Ψ_l should be used [40]. Example of such a neighborhood system is shown in Figure 3.5. In Figure 3.5(a) it is centered on a horizontal discontinuity, while in Figure 3.5(b) it is centered on a vertical discontinuity. To exercise control over straight lines, corners and intersections, typically four-element cliques are used (Figure 3.5(c) and 3.5(d)). The cross-shaped cliques (Figure 3.6(a)) with potential V_C discourage creation of unended and intersecting segments. The square-shaped cliques (Figure 3.6(b)) with potential V_D discourage formation of double lines and also inhibit generation of isolated trajectories ($V_{D4} = \infty$).

Note that the *a priori* probability of the line process (3.35) is conditioned on the observations $\mathbf{g}_{t_n}$. It means that image information should be considered when computing l_t. In general, a 3D scene giving rise to a motion discontinuity also contributes to an intensity edge. Only under specific circumstances does a motion discontinuity not correspond to an edge of intensity. Hence, we assume that the introduction of a motion discontinuity should coincide with an intensity edge. To enforce

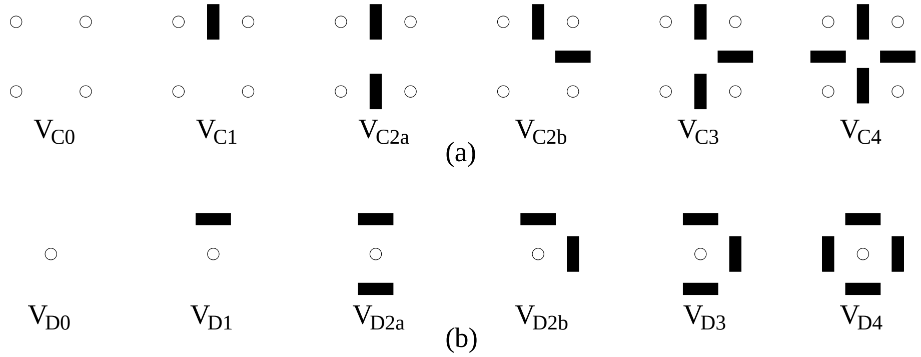

Figure 3.6: Costs associated with various configurations (up to rotation) of line cliques; (a) four-element cross-shaped cliques; (b) four-element square-shaped cliques. Usually, $V_{C0} = 0 < V_{C2a} < V_{C2b} < V_{C1}, V_{C3}, V_{C4}$ and $V_{D0} = 0 < V_{D1} < V_{D2b} < V_{D3} < V_{D2a} < V_{D4} = \infty$ (filled rectangle: line element "on", circle: position of a motion trajectory).

such a coincidence, we use single-element cliques with potential V_E assuming a high value whenever a motion discontinuity does not match an intensity edge. This can be done in two ways. We can detect intensity edges first, and then set potential V_E to a large value whenever motion discontinuity is attempted at locations where no intensity edge is present [28]. The other approach is to make V_E inversely proportional to a norm of the local image gradient [37]. The total potential for a given motion discontinuity is defined as a sum of potentials for single-element and four-element cliques $V_l = V_C + V_D + V_E$ (Figure 3.6).

3.2.6 A posteriori probability and cost function

In preceding sections we have defined the correspondence, trajectory, occlusion and motion discontinuity models through Gibbs distributions (3.14), (3.27), (3.33) and (3.35), respectively. We combine these probabilities using the Bayesian decomposition (3.6) to obtain the following Gibbs distribution:

$$p(\mathbf{c}_t^p, O_t = o_t, L_t = l_t | \mathcal{G}_t) = \frac{1}{Z} e^{-U(\mathbf{c}_t^p, o_t, l_t, \mathcal{G}_t)}, \tag{3.37}$$

where Z is a new normalizing constant incorporating the probability $p(\mathcal{G}_t^n | \mathbf{g}_{t_n})$ from (3.6), as well as Z_g, Z_c, Z_o, Z_l. The new energy function $U(\mathbf{c}_t^p, o_t, l_t, \mathcal{G}_t)$ is defined as follows

$$\begin{aligned} U(\mathbf{c}_t^p, o_t, l_t, \mathcal{G}_t) &= U_g'(\mathcal{G}_t, \mathbf{c}_t, o_t) + \lambda_c U_c(\mathbf{c}_t^p, l_t, \mathbf{g}_{t_n}) + \\ &\quad \lambda_o U_o(o_t, l_t, \mathbf{g}_{t_n}) + \lambda_l U_l(l_t, \mathbf{g}_{t_n}). \end{aligned} \tag{3.38}$$

The constituent energies are defined in (3.17), (3.28), (3.34) and (3.36) respectively, and $\lambda_c = 1/\beta_c$, $\lambda_o = 1/\beta_o$, $\lambda_l = 1/\beta_l$.

The *a posteriori* probability (3.37) may lead to different Bayesian solutions of the motion estimation problem. For example, it may be used in the Maximization of *A Posteriori* Probability [35] or in the Minimization of Expected Cost [36]. We will use the first criterion since for Gibbs-Markov models it leads directly to minimization of the following multiple-term cost function[5]:

$$\begin{aligned} (\mathbf{c}_t^{\widehat{p}}, \widehat{o}_t, \widehat{l}_t) = \min_{\{\mathbf{c}_t^p, o_t, l_t\}} \quad & U_g'(\mathcal{G}_t, \mathbf{c}_t, o_t) + \lambda_c U_c(\mathbf{c}_t^p, l_t, \mathbf{g}_{t_n}) + \\ & \lambda_o U_o(o_t, l_t, \mathbf{g}_{t_n}) + \lambda_l U_l(l_t, \mathbf{g}_{t_n}). \end{aligned} \tag{3.39}$$

[5] Due to parametrization of motion trajectories the estimate $\widehat{\mathbf{c}}_t$ is equal to the field of trajectories determined by the parameter vector $\widehat{\mathbf{p}}$: $\mathbf{c}_t^{\widehat{p}}$.

The functional to be minimized consists of four terms: U'_g is a matching energy for all K components and describes the ill-posed problem of matching the data $\mathcal{G}_t$ by the motion field $\mathbf{c}_t^{\widehat{p}}$; U_c is responsible for conforming to the properties of the *a priori* trajectory model; U_o models occlusion areas; and U_l allows for occasional motion discontinuities. The four-term formulation of the energy function (3.38) can be viewed as regularization of the original correspondence problem.

In the formulation (3.39) the ratios of λ's play an important role weighting the confidence in the data and in the *a priori* models. A modification of any λ has an effect on the estimate; however the magnitude of this effect is highly dependent on the data itself. The parameters β_g, β_c, β_o and β_l, which characterize individual models, are difficult to compute. When MRFs are used in estimation of such observables as images or textures, β's can be estimated by analyzing a number of samples (training process), and then used to perform estimation on some other data. The success of the estimation is highly dependent on the similarity between the real data and the model (the training data). In the case of estimating an unobservable such as motion, it is not clear how to compute β's and thus they are usually chosen *ad hoc*.

3.2.7 Solution method

Minimization (3.39) is a very difficult task for two reasons. First, there are hundreds of thousands of unknowns to be found; for a typical 512×512 image, there are 262,144 vectors of trajectory parameters (minimum 2-D), 262,144 occlusion labels and 523,264 motion discontinuities. Secondly, the cost function under minimization is not convex. Thus, it is very likely that multiple minima exist. Additionally, the cost function is not differentiable since some energies are not expressed analytically but are obtained from look-up tables of potential values.

Consequently, a minimization method is sought which can handle hundreds of thousands of variables, which can find the global optimum and which can deal with non-differentiable functions. A method satisfying all three requirements is *simulated annealing* [32],[19] which is a stochastic search algorithm. This method is a computer simulation of the process of *annealing* of solids; the behavior of a solid is simulated by generating sample configurations from a Gibbs distribution with a suitable energy function divided by a "temperature" parameter T. The sample configurations are produced using *stochastic relaxation* such as the *Metropolis algorithm* [43] or the *Gibbs sampler* [19]. Initially T is chosen to be sufficiently high to generate a full range of configurations, even the unlikely ones. This assures avoidance of local minima and is equivalent to melting the solid. As the process evolves, T is very slowly reduced. It has been shown [19], that if the reduction of T is sufficiently slow, then the system attains (in a limit) the state of minimal energy.

Simulated annealing has been proposed as the solution method for MAP estimation of 2-D motion [40]. The method has been applied to the case of linear motion (equation (3.5)) without occlusions (λ_o=0) for scalar [40] and vector (K=3) [38] observations. In fact, two types of simulated annealing algorithms for the estimation of 2-D motion have been proposed: a discrete state space algorithm and a continuous state space one. In the first case, *a priori* precision (quantization step) is chosen and only vectors satisfying the precision constraint are generated. In the second case, a local approximation of the Gibbs distribution by a Gaussian leads to a Gauss-Newton iterative update with an additional T-dependent stochastic update. This equation is a variant of the diffusion equation proposed in [20] for global optimization. In the limit with $T \rightarrow 0$, the stochastic term disappears resulting in a deterministic Gauss-Newton algorithm.

By instantaneously "freezing" the system ($T = 0$), equivalent deterministic methods can be obtained. For example, the discrete state space simulated annealing results in pixel matching with smoothness constraints, while the continuous state space one leads to equations similar to those proposed in [27],[28].

If the point of departure is a four-term cost function like in (3.39), then a different strategy must be used. Note that trajectories $\mathbf{c}^p$ are described by continuous-valued parameters, while the occlusion labels and line elements are discrete-valued with a finite number of states. This suggests that different methods must be used in order to estimate $\mathbf{c}^p$ and o, l. This can be done in an interleaved fashion, i.e., while one is iteratively updated, the others are kept constant. For given estimates $\widehat{o}_t$ and $\widehat{l}_t$, in order to improve the estimate of $\mathbf{c}_t^p$, one iteration (full scan of a field) of minimization

$$\min_{\{\mathbf{c}_t^p\}} U_g'(\mathcal{G}_t, \mathbf{c}_t, \widehat{o}_t) + \lambda_c U_c(\mathbf{c}_t^p, \widehat{l}_t, \mathbf{g}_{t_n}), \tag{3.40}$$

is carried out. U_g' is still non-quadratic, but U_c is quadratic because $\widehat{l}_t$ is given and the potential (3.29) is quadratic (as explained in Section 3.2.5 dependence on o_t has been omitted). To solve (3.40), for example Gauss-Newton optimization can be used, as was done in [3],[39]. In this method, a sequence of local quadratic approximations of U_g' is carried out. Once $\mathbf{c}_t^p$ has been updated, it becomes a known estimate $\mathbf{c}_t^{\widehat{p}}$ and o_t is improved by executing one iteration of

$$\min_{\{o_t\}} U_g'(\mathcal{G}_t, \mathbf{c}_t^{\widehat{p}}, o_t) + \lambda_o U_o(o_t, \widehat{l}_t, \mathbf{g}_{t_n}). \tag{3.41}$$

Since the number of possible states for each occlusion label is small, usually an exhaustive search is used. Also, such methods as Besag's *Iterated Conditional Modes* [6] can be used. Finally, l_t can be re-iterated by executing

$$\min_{\{l_t\}} \lambda_c U_c(\mathbf{c}_t^{\widehat{p}}, l_t, \mathbf{g}_{t_n}) + \lambda_o U_o(\widehat{o}_t, l_t, \mathbf{g}_{t_n}) + \lambda_l U_l(l_t, \mathbf{g}_{t_n}). \tag{3.42}$$

Once all three fields have been updated, the process is repeated until no reduction of energies can be observed. This approach usually leads to good results.

The stochastic algorithms for linear motion with piecewise smoothness have been shown to perform very well on synthetic and natural images [40]. These algorithms are, however, very demanding computationally and require from one (for the continuous state space) to two (for the discrete state space) orders of magnitude more time than deterministic algorithms. In a direct comparison [39], the stochastic algorithms perform better than equivalent ("frozen") deterministic ones both in terms of the cost function value and subjective evaluation. The improvement is particularly pronounced for some cases of ambiguous data with synthetic motion [39]. For typical video images, however, the improvement is small.

Other methods have been proposed to minimize variants of energy (3.38) for the case of linear trajectories and scalar data. For example, the simplest cost function, using only matching and motion energies, proposed in [35], has been optimized using mean field annealing [1]. Also, method called Highest Confidence First (HCF) [11] has been used to optimize the three-term energy (with additional motion discontinuity energy) [9].

3.2.8 Estimation over a hierarchy of resolutions

Both stochastic estimation methods summarized in the preceding section as well their deterministic counterparts are inappropriate for the estimation of fast motion (large displacements). The computational complexity of discrete state space algorithms becomes prohibitively large because the state space for each trajectory grows very rapidly with the maximum allowed displacement. On the other hand, the continuous state space methods fail when certain underlying assumptions are violated. For example, to simplify U_g' (Section 3.2.7) small displacements are assumed so that local intensity linearity holds; this linearity does not hold for large displacements.

To deal with the above problems various multiresolution methods have been proposed for motion estimation. One class of such methods uses coarse-to-fine strategy to compute motion. First, a pyramid of observations is constructed by low-pass or band-pass filtering and subsampling (usually by 2 in both directions) [8]. Then, a pyramid of fields to be estimated is built by similar subsampling. Estimation starts at the top of the pyramid. Since only a fraction of the field must be computed and since distances between field elements (e.g., trajectories) are relatively large, the convergence is very rapid. Once an estimate is obtained, it is propagated to a lower level by an appropriate operator, such as interpolation. The process is repeated until an estimate is obtained at full resolution. At each level any of the methods discussed in the preceding section can be used. For example, a matching algorithm for the estimation of discrete displacements has been proposed in [2] (see Chapter 1 in this book as well). Also, gradient-based methods for the continuous state space have been suggested in [23], and [17].

A similar multiresolution approach has been proposed for Gibbs-Markov models of motion [36]. Markov models at different levels of hierarchy have been linked by experimentally established parameters. Stochastic algorithms for discrete and continuous state space, and piecewise-smooth motion model (including motion discontinuities) have been given in [40]. Since subsampling of the observations aims only at efficient storage, no subsampling has been applied to images, thus resulting in a constant-width observations pyramid. The filtering of the observations has been used, however, as it is essential for gradient-based methods (reduction of degree of violation of local intensity linearity assumption). The same filtering, although not essential, is helpful in discrete matching methods because it reduces ambiguity due to local detail. Unfortunately, the filtering may also introduce unwanted artifacts, such as confusion of objects with background, which usually leads to locally unreliable estimates.

The above arguments suggest that for discrete state space methods it may be possible to use an estimation pyramid and raw data with no filtering or subsampling. Such a method, called *multiscale* estimation has been proposed for globally smooth linear motion (displacements) based on Markov hierarchical model [25]. In this approach Markov models at higher levels of the hierarchy have been rigorously derived from the full resolution model. This method belongs to another class of coarse-to-fine algorithms where instead of subsampling, the size of a block in which all estimates are kept constant varies from level to level. In this way a hierarchy of detail scales of an estimated field is obtained. The method has been reported to give results very similar to those obtained by stochastic monoresolution techniques [35] and superior to results obtained by deterministic multiresolution methods using filtering and subsampling of observations [39].

3.3 Motion-Compensated Processing

This section presents a number of applications of 2-D motion estimates to motion-compensated processing of video signals for coding, enhancement or standards conversion. The first application is predictive coding, in which we encode information along motion trajectories, where redundancy is highest. The second application is in noise reduction, where low-pass filtering along motion trajectories can be used to reduce the level of additive noise with little effect on picture content. We then discuss motion-compensated sampling structure conversion, which is a problem of spatiotemporal interpolation. Examples are frame rate increase, interlace-to-progressive conversion, and general standards conversion (e.g., NTSC $\leftrightarrow$ PAL, or NTSC $\leftrightarrow$ HDTV).

3.3.1 Motion-compensated prediction

Predictive coding is the most widely used technique for temporal redundancy reduction, and is incorporated in some way in most existing or proposed image coding standards, from low resolution teleconferencing [42] to HDTV [21],[60]. Motion trajectory estimates are used to predict an image field from previous fields using samples along the same estimated trajectory. The prediction error is then encoded and transmitted. Motion-compensated predictive coding is normally accomplished using the DPCM structure with a configuration similar to that shown in Figure 3.7. In the DPCM encoder, the input g_r to the predictor is equal to the image reconstructed at the receiver (in the absence of transmission errors). Both the input g and the reconstructed signal g_r can potentially be used as input to the motion estimation process. If only g_r is used, the receiver can duplicate the same operations, and motion information need not be transmitted. However, that may compromise the accuracy of the motion trajectory estimates and reduce coding efficiency. Since it is known that motion fields can be coded efficiently [51], we can use the original image to get the best possible motion field estimate and transmit it as side information.

The goal of prediction is to determine an estimate $\widehat{g}(\mathbf{x},t)$ of the image $g(\mathbf{x},t)$ on the image sampling lattice. Thus, we must estimate a field of motion trajectories terminating at $(\mathbf{x},t)$, $\mathbf{x} \in (\Lambda_g)_t$ and passing through M previous fields using methods of the previous sections. The prediction is then given by

$$\widehat{g}(\mathbf{x},t) = \sum_{m=1}^{M} b_m \widetilde{g}_r(\mathbf{c}^{\widehat{p}}(t - mT_g; \mathbf{x}, t), t - mT_g), \tag{3.43}$$

where M is the predictor order and $\mathbf{c}^{\widehat{p}}(\tau; \mathbf{x}, t)$ is the trajectory terminating at $(\mathbf{x},t)$ determined by the estimated parameter vector $\widehat{\mathbf{p}}(\mathbf{x},t)$. In the case of first order prediction,

$$\widehat{g}(\mathbf{x},t) = b\widetilde{g}_r(\mathbf{x} - \widehat{\mathbf{d}}(\mathbf{x},t), t - T_g), \tag{3.44}$$

where b is usually taken to be unity due to the high correlation along motion trajectories and the fact that this gives perfect prediction in flat areas, where the viewer is most sensitive to distortions. For higher order prediction, the coefficients can be determined by estimating the correlation function along motion trajectories and solving the normal equations to get minimum mean square prediction error. Specifically, the coefficients b_m are chosen to minimize

$$\mathrm{E}\left[g(\mathbf{x},t) - \sum_{m=1}^{M} b_m \widetilde{g}_r(\mathbf{c}^{\widehat{p}}(t - mT_g; \mathbf{x}, t), t - mT_g) \right]^2 . \tag{3.45}$$

By the orthogonality principle, the optimal coefficients satisfy the normal equations

$$\mathbf{R}\mathbf{b} = \mathbf{r} \tag{3.46}$$

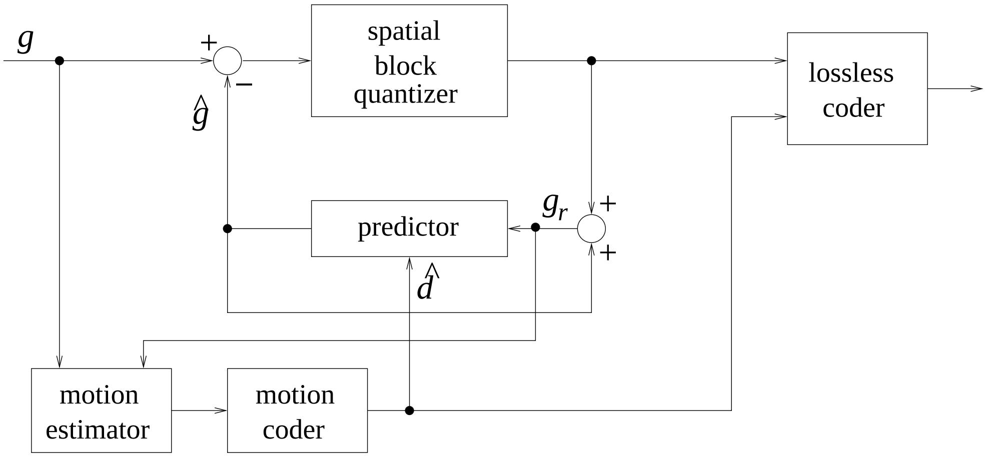

Figure 3.7: Block diagram of motion-compensated DPCM coder

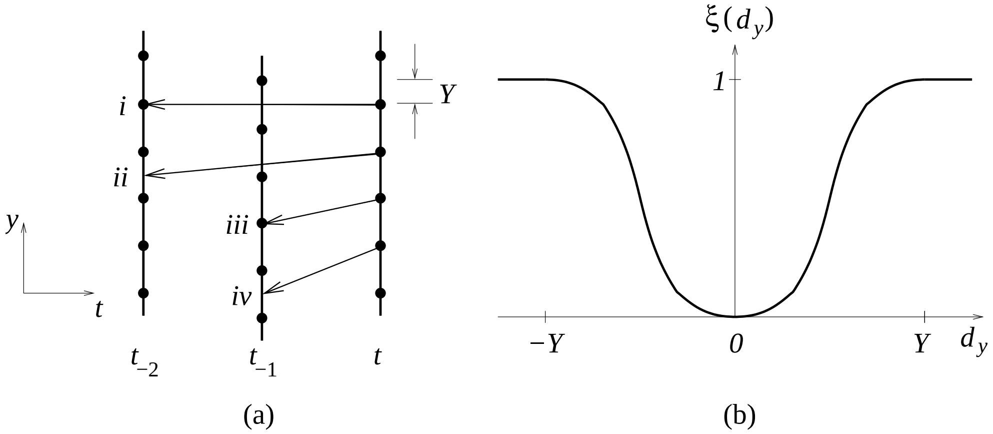

Figure 3.8: (a) Three fields used for prediction in an interlaced sequence. The field t is to be predicted from the two fields t_{-1} and t_{-2}. (b) The weighting function $\xi(d_y)$.

where for $m = 1, 2, \ldots, M$

$$r_m = \mathrm{E}[g(\mathbf{x}, t)\widetilde{g}_r(\mathbf{c}^{\widehat{p}}(t - mT_g; \mathbf{x}, t), t - mT_g)], \tag{3.47}$$

and for for $k, m = 1, 2, \ldots, M$

$$[R]_{km} = \mathrm{E}[\widetilde{g}_r(\mathbf{c}^{\widehat{p}}(t - kT_g; \mathbf{x}, t), t - kT_g)\widetilde{g}_r(\mathbf{c}^{\widehat{p}}(t - mT_g; \mathbf{x}, t), t - mT_g)]. \tag{3.48}$$

To a first approximation, the correlations can be estimated using the input signal along estimated motion trajectories (i.e., ignoring the spatial block quantizer in the DPCM loop) over a representative training set containing a wide variety of images and motion trajectories.

In many cases, it may be advantageous to use coefficients that depend on the motion field, since the correlation function along trajectories may depend on the motion (i.e., on $\widehat{\mathbf{p}}$). Interlace plays an important role in the form of the correlation function along a trajectory, since intrafield spatial aliasing may be present. With reference to Figure 3.8(a), for trajectory *i* the correlation with field t_{-2} may be greater than with field t_{-1}, while for trajectory *iii* the opposite is true. A prediction with motion-dependent coefficients takes the form

$$\widehat{g}(\mathbf{x}, t) = \sum_{m=1}^{M} b_m(\widehat{\mathbf{p}})\widetilde{g}_r(\mathbf{c}^{\widehat{p}}(t - mT_g; \mathbf{x}, t), t - mT_g). \tag{3.49}$$

In this case, we must solve the equation

$$\mathbf{R}(\widehat{\mathbf{p}})\mathbf{b}(\widehat{\mathbf{p}}) = \mathbf{r}(\widehat{\mathbf{p}}) \tag{3.50}$$

for each $\widehat{\mathbf{p}}$ of interest. In order to determine the correlations that specify $\mathbf{r}(\mathbf{p})$ and $\mathbf{R}(\mathbf{p})$, the parameter vectors can be clustered into classes within which the correlation function is similar, and the normal equations would then need to be solved for each of these classes.

Another approach to motion-adaptive prediction, that is based on *linear* trajectories and is applicable to second order prediction for interlaced sequences, is [12]

$$\widehat{g}(\mathbf{x}, t) = \xi(\widehat{d_y})\widetilde{g}_r(\mathbf{x} - \widehat{\mathbf{d}}(\mathbf{x}, t), t - T_g) + (1 - \xi(\widehat{d_y}))\widetilde{g}_r(\mathbf{x} - 2\widehat{\mathbf{d}}(\mathbf{x}, t), t - 2T_g). \tag{3.51}$$

The coefficients depend only on the vertical component of $\widehat{\mathbf{d}}$, and enforce perfect prediction in flat areas. When the vertical component of $\widehat{\mathbf{d}}$ is zero, we choose $\xi = 0$ so that the prediction is based on the second previous field only, where correlation is highest. As $|d_y|$ increases, ξ increases until the point where $|d_y| = Y$ (Figure 3.8(a)), where we use $\xi = 1$, i.e., previous field prediction. A suitable function is (Figure 3.8(b))

$$\xi(d_y) = \begin{cases} \frac{1}{2}(1 - \cos(\pi d_y/Y)) & 0 \leq |d_y| \leq Y \\ 1 & |d_y| > Y \end{cases} \tag{3.52}$$

This adaptive prediction can also be used to form the matching energy for the objective function used in the motion estimation phase

$$U_g = \|g(\mathbf{x}, t) - \widehat{g}(\mathbf{x}, t)\|^2. \tag{3.53}$$

Since ξ is a continuous function of $\mathbf{d}$, it can be used without problems in the gradient descent algorithms described previously.

This method was shown to perform better than either fixed previous field or fixed second previous field prediction on a variety of image sequences [12].

3.3.2 Motion-compensated noise reduction

Random noise can be a significant impairment in video signals. Many techniques for image filtering to reduce perceived noise level have been studied, such as Wiener filtering or non-linear filtering (e.g., median filtering) [29], [52]. The major difficulty is to distinguish between noise and image information. If spatiotemporal motion trajectories are known, most random temporal variation along the trajectory can be attributed to noise. Thus, one-dimensional processing along motion trajectories is an attractive approach for noise reduction. We assume in the following that motion trajectory estimates are available for the noisy sequence. They may either be estimated from the noisy sequence by a robust motion estimation algorithm, or from the original uncorrupted sequence and assumed to be available as side information that was transmitted in a noiseless fashion (e.g., digitally, in a hybrid analog/digital transmission scheme [57]).

Let $g(\mathbf{x}, t)$ be the noisy input image. The idea is to perform a one-dimensional filtering for each $(\mathbf{x}, t)$ of the 1-D signal $s(\tau; \mathbf{x}, t) = g(\mathbf{c}(\tau; \mathbf{x}, t), \tau)$. If FIR filtering is used, this can be expressed as

$$g_r(\mathbf{x}, t) = \sum_{m=-M_1}^{M_2} b_m \widetilde{g}(\mathbf{c}(t - mT_g; \mathbf{x}, t), t - mT_g). \tag{3.54}$$

As in the previous section, the b_m could be determined from knowledge of the correlation function of the image along the trajectory by solution of the normal equations. However, a fairly high filter order is required to obtain adequate noise reduction, which is likely to be too costly in terms of frame memories. Furthermore, it may be difficult to accurately specify motion trajectories over long time periods. A very attractive alternative is recursive filtering which uses only a few frames of delay.

An all pole motion-compensated recursive filter has the form

$$g_r(\mathbf{x}, t) = \gamma g(\mathbf{x}, t) + (1 - \gamma) \sum_{m=1}^{M} b_m \widetilde{g}_r(\mathbf{c}(t - mT_g; \mathbf{x}, t), t - mT_g). \tag{3.55}$$

The transfer function of the 1-D filter operating along the motion trajectory is

$$H(z) = \frac{\gamma}{1 - (1 - \gamma) \sum_{m=1}^{M} b_m z^{-m}}. \tag{3.56}$$

This can be represented by the block diagram shown in Figure 3.9, where $P(z) = \sum_{m=1}^{M} b_m z^{-m}$ has the form of a predictor. Thus, this component can be designed using the methods of the previous section. For progressively scanned image sequences, a first order filter of the form

$$g_r(\mathbf{x}, t) = \gamma g(\mathbf{x}, t) + (1 - \gamma) \widetilde{g}_r(\mathbf{x} - \widehat{\mathbf{d}}(\mathbf{x}, t), t - T_g) \tag{3.57}$$

is suitable. This low-pass filter increasingly attenuates noise as the parameter γ approaches 0; it can be shown that the increase in SNR when the true image intensity is constant on the motion trajectory is given by $10\log_{10}(2-\gamma)/\gamma$ dB [16]. For $\gamma = 1$, there is no filtering. For interlaced sequences, a second order temporal filter is advantageous for the same reasons cited for the case of prediction. In this case, the noise reduction filter takes the form

$$\begin{aligned} g_r(\mathbf{x},t) &= \gamma g(\mathbf{x},t) + (1-\gamma)[\xi(d_y)\widetilde{g}_r(\mathbf{x}-\widehat{\mathbf{d}}(\mathbf{x},t), t-T_g) + \\ & \quad (1-\xi(d_y))\widetilde{g}_r(\mathbf{x}-2\widehat{\mathbf{d}}(\mathbf{x},t), t-2T_g)]. \end{aligned} \tag{3.58}$$

The motion-compensated filters described above perform well when motion is uniform and correctly estimated. In practice, due to effects such as occlusion or complex motion, the image intensity is not constant along the estimated motion trajectory. The filtering will then significantly impair the image. The value of the "prediction error" $e(\mathbf{x},t)$ can be used as an indicator of whether filtering should be performed. If $e(\mathbf{x},t)$ is relatively small (relative to the noise level), it can be assumed that the motion trajectory estimate is accurate enough and filtering should be performed. However, if $e(\mathbf{x},t)$ is large, we assume that filtering should be reduced or disabled. This can be achieved by replacing the multiplier γ with a memoryless nonlinearity having the desired effect: the output of the nonlinearity is $\gamma(e)\cdot e$ where $\gamma(e)$ has a form such as shown in Figure 3.10. The overall noise reduction system is then as shown in Figure 3.11.

A study of this system showed that on a variety of image sequences, a good choice of parameters was $P_1 = 10$, $P_2 = 20$, $\gamma_1 = 0.3$ and $\gamma_2 = 1.0$ when the noise standard deviation was 5.0. A clear subjective improvement in image quality was noted for all sequences [7].

3.3.3 Motion-compensated interpolation

Interpolation refers to the process of filling in missing samples in a signal based on neighboring samples. If the signal has been sampled respecting the multidimensional Nyquist criterion, the interpolation can be carried out by fixed interpolation filters with good results. However, video signals are often sampled in space-time in a way that the Nyquist criterion is violated, so that aliasing is present and fixed interpolation filters do not give good results. This is particularly true with respect to temporal sampling, as described in [15]. Motion-compensated interpolation along motion trajectories can overcome the limitations of fixed filtering.

Two cases arise in spatiotemporal interpolation. The first case is when no input data is available at the time instant t for which interpolation is being carried out. In other words, an image field is being generated at a time for which no input image field exists, as in Figure 3.2(a). This is the case for applications such as field rate standards conversion (e.g., between 50 Hz and 60 Hz), up-conversion

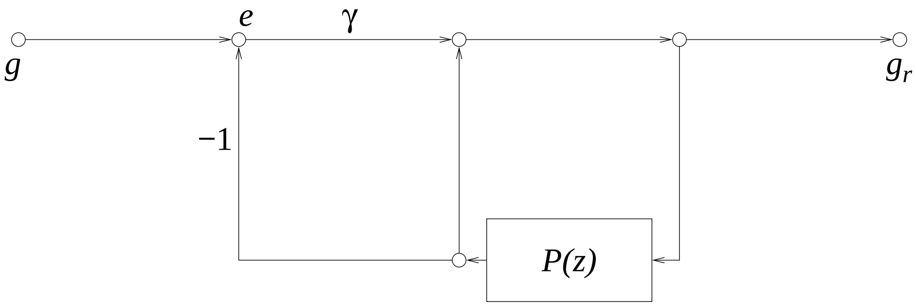

Figure 3.9: All-pole temporal recursive filter.

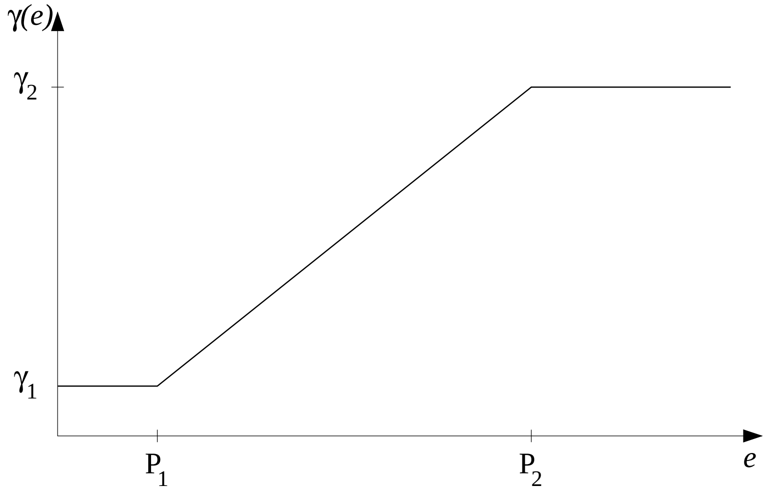

Figure 3.10: Nonlinear function relating multiplication coefficient γ to the prediction error.

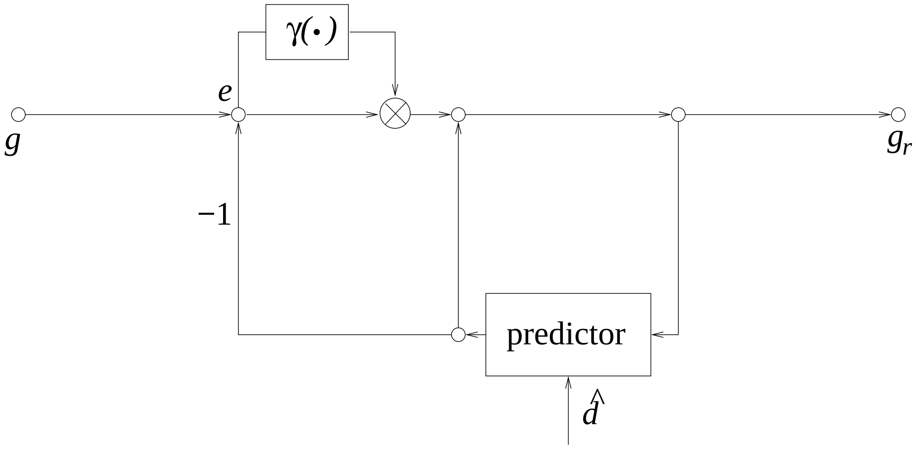

Figure 3.11: Block diagram of motion-compensated noise reducer.

from temporally subsampled signals, and field rate increase to reduce display artifacts or for slow-motion portrayal. In this case, purely temporal interpolation is involved. The second case is when input image samples exist at the given time (as in Figure 3.2(b)), but spatial aliasing may be present due to a non-orthogonal sampling structure, such as in interlaced sampling. In this case, temporal interpolation may give better results than spatial interpolation, and should be incorporated into the overall interpolation algorithm, perhaps jointly with spatial interpolation.

Motion-compensated interpolation takes the form

$$\widehat{g}(\mathbf{x},t) = \sum_{t_j \in \mathcal{I}_t^{\mathbf{x}}} b_j \widetilde{g}(\mathbf{c}(t_j;\mathbf{x},t),t_j), \tag{3.59}$$

where the trajectory covers only points visible at $(\mathbf{x},t)$. Since the intensity varies slowly along the motion trajectory, fairly simple 1-D filters can be used, such as equal coefficients $b_j = 1/|\mathcal{I}_t^{\mathbf{x}}|$, or coefficients that decrease with distance from t. As with prediction and noise reduction, the coefficients can be made dependent on the motion trajectory parameters, and thus on position with respect to the sampling lattice. It should be noted that errors in motion estimates can lead to erroneously interpolated values that can seriously impair the picture, thus making highly reliable motion estimates a necessity for motion-compensated interpolation. If the field to be interpolated is available at the motion estimation phase (as in some coding systems where the motion field is transmitted) much better interpolation results can often be obtained [13].

In the second case of spatiotemporal interpolation, the motion-compensated interpolation should be combined with a spatial interpolation. As an example, consider the conversion of an image sequence from interlaced to progressive format. Referring to Figure 3.8, we see that if the vertical component of the motion is near zero, a motion-compensated temporal interpolation will be effective (where the horizontal motion is compensated). However, as the vertical velocity increases, the temporal interpolation is less effective and spatial interpolation might as well be applied. Thus the overall interpolation scheme is given by

$$\widehat{g}(\mathbf{x},t) = \xi \widehat{g}_T(\mathbf{x},t) + (1-\xi)\widehat{g}_S(\mathbf{x},t) \tag{3.60}$$

where $\widehat{g}_T(\mathbf{x},t)$ is obtained by motion-compensated temporal interpolation, as in equation (3.59), and $\widehat{g}_S(\mathbf{x},t)$ is obtained by spatial interpolation. The coefficient ξ can be based simply on d_y as previously, or it can be based on a more complex local analysis comparing the relative efficacy of temporal versus spatial interpolation [49].

The spatial interpolation can be a simple fixed vertical interpolation, such as cubic interpolation. However, because of the significant spatial aliasing present in single fields of interlaced sequences, more sophisticated spatial interpolation schemes may be advantageous. For example, directional interpolation based on local orientation analysis has been shown to provide a useful improvement in interpolation of diagonal edges, reducing the serration effect produced by pure vertical interpolation [50].

3.4 Summary

Estimates of 2-D motion are required in order to perform different types of temporal processing of time-varying images along trajectories of motion. In this chapter, we have presented a general Bayesian formulation for the estimation of motion trajectories from observed image sequences. By an appropriate parametrization of the trajectories, the formulation can handle nonlinear trajectories that result from accelerating motion. Occlusion effects such as newly exposed or disappearing image features are also included. General multichannel image observations (e.g., three color components) are assumed and incorporated into a vector image formulation. A maximum a posteriori probability (MAP) estimation criterion has been developed on the basis of a number of models: a model for the observation process, a structural model relating the unknown motion to the underlying image, and an a priori model of the motion field, including occlusion and discontinuity tags. Most estimation criteria that have been proposed can be seen to be special cases of this general formulation. Some of these special cases have been identified.

We have pointed out methods appropriate for minimization of the resulting cost function, specifically stochastic and deterministic relaxation methods. The importance of multiresolution and multiscale methods in efficient localization of a near global optimum has also been addressed.

Several applications of motion-compensated processing to the problems of prediction, interpolation and smoothing have been discussed. General methods for motion-compensated predictor design have been proposed, including some methods that can deal with the problematic effects of interlace. These predictors can be used in both motion-compensated coding and motion-compensated noise reduction. Finally, motion-compensated spatiotemporal interpolation, required for effective sampling structure conversion, has been addressed.

Many aspects of the general formulation presented remain to be investigated, and the relative importance of many of its features must be established. We believe that this formulation can be a useful framework for further research in advanced motion estimation and compensation techniques.

Appendix

In this appendix the definitions of a *Markov random field* and of the *Gibbs distribution* are given, and the relationship between them, the *Hammersley-Clifford theorem*, is described. For more detailed account, please consult [19],[5],[55].

Let sampling structure Ψ be a collection of sites in R^N [14]. A collection $\mathcal{N}$ of subsets of Ψ

$$\mathcal{N} = \{\eta(\mathbf{x}_i) :\ \mathbf{x}_i \in \Psi, \eta(\mathbf{x}_i) \subset \Psi, \forall i\} \tag{3.61}$$

is a *neighborhood system* on Ψ, if and only if, the *neighborhood* $\eta(\mathbf{x}_i)$ of site $\mathbf{x}_i \in \Psi$ satisfies both of the following conditions:

1. $\mathbf{x}_i \notin \eta(\mathbf{x}_i)$,
2. if $\mathbf{x}_j \in \eta(\mathbf{x}_i)$, then $\mathbf{x}_i \in \eta(\mathbf{x}_j)$ for any $\mathbf{x}_i \in \Psi$.

Examples of low-order neighborhood systems for orthogonal and non-orthogonal sampling structures Ψ over R^2 are given in Figures 3.3 and 3.5, respectively. For higher-order neighborhood systems see [19].

Random field over Ψ is a multidimensional stochastic process where each site in Ψ is assigned a random variable. A vector random field has a random vector (ensemble of random variables) assigned at each site in Ψ.

A *Markov random field* Υ is a random field with the following properties:

1. $P(\Upsilon = \upsilon) > 0, \quad \forall\, \upsilon \in \mathcal{S}$,
2. $P(\Upsilon_i = \upsilon_i | \Upsilon_j = \upsilon_j, \forall j \neq i) = P(\Upsilon_i = \upsilon_i | \Upsilon_j = \upsilon_j, \forall j \in \eta(i)), \quad \forall\, i, \forall \upsilon \in \mathcal{S}$,

where P denotes a probability measure and $\mathcal{S}$ is a state space. For a discrete $\mathcal{S}$, P is a probability for a given state, while for a continuous $\mathcal{S}$, P is replaced by the cumulative distribution F_Υ. If F_Υ is differentiable, the above property applies directly with the densities p replacing the probabilities P.

In order to define the *Gibbs distribution* the concepts of *clique* and *potential function* are needed. A *clique* θ defined over Ψ with respect to $\mathcal{N}$ is a subset of Ψ such that either θ consists of a single site or every pair of sites in θ are neighbors, i.e., $\mathbf{x}_i \in \eta(\mathbf{x}_j)\ \forall\{\mathbf{x}_i, \mathbf{x}_j\} \in \theta$. The set of all cliques is denoted by Θ. Examples of low-order cliques are shown in Figures 3.3 and 3.5.

Let υ be a sample field from random field Υ defined over Ψ and over state space $\mathcal{S}$. A *Gibbs distribution* with respect to Ψ and $\mathcal{N}$ is a probability measure π on $\mathcal{S}$ such that

$$\pi(\upsilon) = \frac{1}{Z} e^{-U(\upsilon)/\beta}, \tag{3.62}$$

where β, Z are constants, and the *energy function* U is of the form

$$U(\upsilon) = \sum_{\theta \in \Theta} V(\upsilon, \theta). \tag{3.63}$$

$V(\upsilon, \theta)$ is called a *potential function*, and depends only on those samples from υ which belong to the clique θ. Z is called a *partition function* and is a normalizing constant such that π is a probability measure. β is a parameter called *natural temperature.*

A MRF can be uniquely characterized by a finite-dimensional joint probability distribution, and thus all initial and transitional (conditional) probability distributions are needed. This approach is cumbersome, because the conditional probability distributions must satisfy certain consistency conditions [5], and because the computation of the joint distribution is usually difficult. Also, the relationship between the form of a conditional probability distribution and the characteristic properties of a sample field is not obvious. A clear and simple relationship can be provided, however, through the *Hammersley-Clifford theorem* [55],[5] which states that if Υ is a MRF on Ψ with respect to $\mathcal{N}$, then the probability distribution of its sample realizations is a Gibbs distribution with respect to Ψ and $\mathcal{N}$. This unique characterization of a MRF by a Gibbs distribution results in a straightforward relationship between qualitative properties of a MRF and its parameters via the potential functions V. Extension of the Hammersley-Clifford theorem to vector MRFs is straightforward (only a new definition of a state has to be provided).

Acknowledgement

Our work on motion estimation and motion compensation has been funded by the Natural Sciences and Engineering Research Council of Canada, by the Communications Research Centre and by the Canadian Institute for Telecommunications Research.

References

[1] I. Abdelqader, S. Rajala, W. Snyder, and G. Bilbro, “Energy minimization approach to motion estimation,” *Signal Processing*, vol. 28, pp. 291–309, Sept. 1992.

[2] P. Anandan, “A unified perspective on computational techniques for the measurement of visual motion,” in *Proc. IEEE Int. Conf. Computer Vision*, pp. 219–230, June 1987.

[3] 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.

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

[5] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *J. Roy. Statist. Soc.*, vol. B 36, pp. 192–236, 1974.

[6] J. Besag, “On the statistical analysis of dirty pictures,” *J. Roy. Statist. Soc.*, vol. B 48, pp. 259–279, 1986.

[7] H. Boutrouille, “Réduction du niveau de bruit dans les séquences d’images video par filtrage compensé par le mouvement,” Tech. Rep. 91–29, INRS-Télécommunications, Sept. 1991.

[8] P. Burt, “Fast filter transforms for image processing,” *Comput. Vision, Graphics Image Process.*, vol. 16, pp. 20–51, 1981.

[9] R. Buschmann, “Improvement of optical flow estimation by HCF control and hierarchical block matching,” in *Proc. IEEE Workshop Vis. Sig. Process. Comm.*, pp. 270–273, Sept. 1992.

[10] C. Cafforio and F. Rocca, “Methods for measuring small displacements of television images,” *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 573–579, Sept. 1976.

[11] P. Chou and C. Brown, “Multimodal reconstruction and segmentation with Markov random fields and HCF optimization,” in *Proc. Image Understanding Workshop*, pp. 214–221, Apr. 1988.

[12] R. Depommier and E. Dubois, “Motion-compensated temporal prediction for interlaced image sequences,” in *Proc. IEEE Workshop Vis. Sig. Process. Comm.*, pp. 264–269, Sept. 1992.

[13] R. Depommier and E. Dubois, “Motion estimation with detection of occlusion areas,” in *Proc. IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. III.269–III.272, Mar. 1992.

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

[15] E. Dubois, “Motion-compensated filtering of time-varying images,” *Multidim. Syst. Sig. Process.*, vol. 3, pp. 211–239, 1992.

[16] E. Dubois and S. Sabri, “Noise reduction in image sequences using motion-compensated temporal filtering,” *IEEE Trans. Commun.*, vol. COM-32, pp. 826–831, July 1984.

[17] W. Enkelmann, “Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences,” *Comput. Vision, Graphics Image Process.*, vol. 43, pp. 150–177, 1988.

[18] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 891–905, Sept. 1991.

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

[20] S. Geman and C.-R. Hwang, “Diffusions for global optimization,” *SIAM J. Control and Optimization*, vol. 24, pp. 1031–1043, Sept. 1986.

[21] General Instruments Corporation, *DigiCipher HDTV system description*, Aug. 1991.

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

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

[24] D. Heeger, “Optical flow using spatiotemporal filters,” *Intern. J. Comput. Vision*, vol. 1, pp. 279–302, 1987.

[25] F. Heitz, P. Perez, and P. Bouthemy, “Parallel visual motion analysis using multiscale Markov Random Fields,” in *Proc. IEEE Workshop on Visual Motion*, pp. 30–35, Oct. 1991.

[26] E. Hildreth, “Computations underlying the measurement of visual motion,” *Artificial Intell.*, vol. 23, pp. 309–354, 1984.

[27] B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intell.*, vol. 17, pp. 185–203, 1981.

[28] 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.

[29] A. Jain, *Fundamentals of digital image processing.* Information and System Sciences Series, Prentice Hall, 1989.

[30] J. Jain and A. Jain, “Displacement measurement and its application in interframe image coding,” *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1808, Dec. 1981.

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

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

[33] J. Konrad, *Bayesian estimation of motion fields from image sequences.* PhD thesis, McGill University, Dept. Electr. Eng., June 1989.

[34] J. Konrad, "Use of colour in gradient-based estimation of dense two-dimensional motion," in *Proc. Conf. Vision Interface VI'92*, pp. 103–109, May 1992.

[35] J. Konrad and E. Dubois, "Estimation of image motion fields: Bayesian formulation and stochastic solution," in *Proc. IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. 1072–1075, Apr. 1988.

[36] J. Konrad and E. Dubois, "Multigrid Bayesian estimation of image motion fields using stochastic relaxation," in *Proc. IEEE Int. Conf. Computer Vision*, pp. 354–362, Dec. 1988.

[37] 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.

[38] J. Konrad and E. Dubois, "Use of colour information in Bayesian estimation of 2-D motion," in *Proc. IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. 2205–2208, Apr. 1990.

[39] J. Konrad and E. Dubois, "Comparison of stochastic and deterministic solution methods in Bayesian estimation of 2D motion," *Image & Vision Computing*, vol. 9, pp. 215–228, Aug. 1991.

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

[41] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Communications ACM*, vol. 34, pp. 46–58, Apr. 1991.

[42] M. Liou, "Overview of the p x 64 kbits/s video coding standard," *Communications ACM*, vol. 34, pp. 59–63, Apr. 1991.

[43] 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.

[44] A. Mitiche, Y. Wang, and J. Aggarwal, "Experiments in computing optical flow with the gradient-based, multiconstraint method," *Pattern Recognition*, vol. 20, no. 2, pp. 173–179, 1987.

[45] C. Moloney and E. Dubois, "Estimation of motion fields from image sequences with illumination variation," in *Proc. IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. 2425–2428, May 1991.

[46] D. Murray and B. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 220–228, Mar. 1987.

[47] 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. PAMI-8, pp. 565–593, Sept. 1986.

[48] A. Netravali and J. Robbins, "Motion-compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.

[49] A. Nguyen and E. Dubois, "Adaptive interlaced to progressive image conversion," in *Proc. Int. Workshop on HDTV*, Nov. 1992.

[50] A. Nguyen and E. Dubois, "Spatial directional interpolation using steerable filters," in *Proc. Canadian Conf. Electr. Comp. Eng.*, pp. MA4.8.1–MA4.8.4, Sept. 1992.

[51] H. Nguyen and E. Dubois, "Representation of motion fields for image coding," in *Proc. Picture Coding Symposium*, pp. 8.4.1–8.4.5, 1990.

[52] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 74–90, Jan. 1987.

[53] T. Reuter, "Standards conversion using motion compensation," *Signal Processing*, vol. 16, pp. 73–82, 1989.

[54] M. Sezan, M. Ozkan, and S. Fogel, "Temporally adaptive filtering of noisy image sequences using a robust motion estimation algorithm," in *Proc. IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. 2429–2432, May 1991.

[55] F. Spitzer, "Markov random fields and Gibbs ensembles," *Amer. Math. Mon.*, vol. 78, pp. 142–154, Feb. 1971.

[56] C. Stiller and D. Lappe, "Gain/cost controled displacement-estimation for image sequence coding," in *Proc. IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. 2729–2732, May 1991.

[57] G. Thomas, "HDTV bandwidth reduction by adaptive subsampling and motion-compensated DATV techniques," *SMPTE J.*, vol. 96, pp. 460–465, May 1987.

[58] 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.

[59] K. Wohn, L. Davis, and P. Thrift, "Motion estimation based on multiple local constraints and nonlinear smoothing," *Pattern Recognition*, vol. 16, no. 6, pp. 563–570, 1983.

[60] Zenith and AT&T, *Digital Spectrum Compatible: Technical details*, Sept. 1991.