

4

On Gibbs-Markov Models for Motion Computation

J. Konrad and C. Stiller¹

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

Abstract

In this chapter we present Gibbs-Markov models for 2-D motion in the context of their application to video coding and processing. We study nonlinear trajectory model that incorporates both velocity and acceleration. Although the maximum *a posteriori* probability criterion is the preferred choice for most motion estimation algorithms based on Gibbs-Markov models, we discuss the more general Bayesian criterion, including the merits of several loss functions. We describe various models for the likelihood and prior probability distributions, but we concentrate on pixel-, block- and region-based motion models. We propose a new motion model that incorporates acceleration into the affine model. This contribution is mainly theoretical, however we present some experimental results to underline essential differences between models discussed.

4.1 Introduction

In this chapter we are concerned with Gibbs-Markov models used in the computation of 2-D motion from time-varying images. Our goal is to propose a general formulation that incorporates nonlinear motion trajectories, multichannel (vector) observations and that is applicable at various levels of image detail, for example at pixel, block and (arbitrarily-shaped) region level. To achieve this objective we link individual Gibbs-Markov models by the *a posteriori* probability that we eventually exploit in various Bayesian estimation criteria. We discuss the most popular one, the *maximum a posteriori probability* criterion, along with other criteria based on different Bayes risk functions.

Estimation of 2-D motion from dynamic images is a typical inverse problem (the direct problem being the formation of time-varying intensities due to object/camera motion) that is *ill-posed* and as such is very difficult to solve [2]. Despite this difficulty many approaches to the problem have been proposed in the last two decades; methods presented in [34, 21, 33] are but 3 examples of dozens of methods developed. This activity can be partially attributed to

¹Present address: Robert Bosch GmbH, D-31132 Hildesheim, Germany

the rapid development of digital video compression techniques in which 2-D motion plays an essential role.

Efficient encoding of time-varying images is essential for economical use of network or storage resources 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 motion. Since the correlation of image intensity or color is very high along the direction of motion, the knowledge of motion helps in removing interimage redundancy, as is the case in predictive or hybrid (predictive/transform or predictive/subband) coding compensated for motion. In fact, algorithms currently used in videoconferencing, digital and high-definition TV are of the hybrid type [40]. Other applications that can greatly benefit from the knowledge of motion are sampling structure conversion and noise reduction. Again, due to the high correlation along motion trajectories, motion-compensated interpolation is the most effective tool in both cases. In the sampling structure conversion missing samples can be reliably recovered, whereas in noise reduction noise can be suppressed without altering image features. A very good discussion of those and related issues can be found in [40].

The remainder of this chapter is organized as follows. In the next two sections, a framework for the description of motion models is established. Then, various statistical estimation criteria are discussed. Finally, pixel-, block- and region-based Gibbs-Markov motion models are presented. The main focus of this contribution is the description of various motion models; some experimental results are included but more can be found in our previous publications.

4.2 Framework

As pointed out in the introduction, we are interested in the computation of motion in the context of video processing and compression. Therefore, we assume that time-varying images from which motion is computed are obtained by a camera that projects a 3-D scene onto a 2-D image plane. Furthermore, we assume that every point in the image corresponds to a single point in the 3-D scene; transparent or reflective surfaces are not accounted for. 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. Since the coordinates x and y of the projection of a point in the 3-D scene onto the 2-D image plane vary in time t , it is useful to consider the trajectory of an image point in a conceptual 3-D xyt space.

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} [14]. $\mathbf{c}(\tau; \mathbf{x}, t)$ describes a 2-D trajectory in the image plane, while $(\mathbf{c}(\tau; \mathbf{x}, t), \tau)$ describes a 3-D trajectory in the xyt space. Clearly, there is a unique mapping between the two trajectories.

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

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

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. An image point may also accelerate along a nonlinear 2-D trajectory, thus tracing a nonlinear 3-D trajectory in the xyt space.

Since it is difficult, if not impossible, to estimate a continuum of motion trajectories in the xyt space, we limit our task to the estimation of segments of trajectories $\mathbf{c}(\tau; \mathbf{x}, t)$ for τ in some time interval containing t , where (\mathbf{x}, t) is defined on a sampling lattice $\Lambda_c \subset R^3$. For simplicity we consider only orthorhombic lattices Λ_c and we assume that there are M locations (pixels) in Λ_c at time t . A generalization is possible if we apply trajectory \mathbf{c} to a region of support. Then, \mathbf{c} may describe (Fig. 4.1):

1. motion of a pixel at $(\mathbf{x}, t) \in \Lambda_c$,
2. motion of a rectangular block of pixels $\mathcal{B}(\mathbf{x}, t)$ with the center at \mathbf{x} ,
3. motion of an arbitrarily-shaped region \mathcal{R} .

The issue of model support is further discussed in the next section and also in sections describing the three motion models.

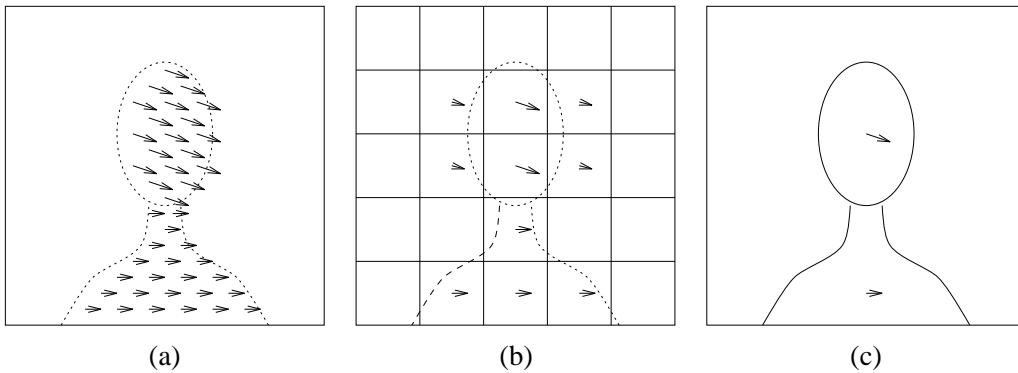


Figure 4.1: Schematic representation of motion for three regions of support: (a) pixel, (b) block and (c) arbitrarily-shaped region.

A trajectory \mathbf{c} is usually estimated from intensity or luminance images. Since there is no particular reason to use only luminance for motion estimation, we consider a more general case where motion is estimated from several cues simultaneously, e.g., components of a color image or combination of range and infrared data. Consequently, we develop models that are based on vector observations. Let \mathbf{u} be the *true* underlying K -component image that is continuous in amplitude and in coordinates, and let $\mathbf{g} = [g_1, g_2, \dots, g_K]^T$ be an observed discrete image.

We take into account occlusion effects present in dynamic images by defining an *occlusion field* $o(\mathbf{x}, t)$ with samples on 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 will model them by parametric functions \mathbf{c}^p over some time interval; details of such a parametrization will be provided in Section 4.3. Since parameters of these functions may change rapidly at object boundaries, we permit such a variation by defining a motion *discontinuity field*² $l(\mathbf{x}, t)$ over a union of two orthorhombic cosets specifying positions of horizontal and vertical discontinuities [26]. Another way to allow such a discontinuity is by the introduction of a generic

²Field l is often called a *line process* [16], while a single variable is called a *line element*.

segmentation field $s(\mathbf{x}, t)$ defined on Λ_c , identifying the object to which the site (\mathbf{x}, t) belongs [38].

We model the trajectories \mathbf{c} , discontinuities l , occlusions o and segmentations s as samples of *Markov random fields* (MRFs) \mathbf{C} , L , O , S , respectively. Since the characterization of a MRF through conditional probabilities is nearly impossible due to potential inconsistency problems, we use the *Hammersley-Clifford theorem* [3] and describe the MRF by a *Gibbs distribution*³.

Let the subscript t denote the restriction of an image \mathbf{g} to time instant t , i.e., \mathbf{g}_t . Since we consider t to be the reference point, i.e., time at which the unknown attributes are estimated, such a restriction is implicit for \mathbf{c} , l , o , s , and therefore omitted.

Since the models to be discussed may associate trajectories \mathbf{c} with various combinations of motion attributes, in order to simplify notation we will use γ (random field Γ) to denote such generic attributes, e.g., $\gamma = (o, l)$ ($\Gamma = (O, L)$). Thus, our goal is to find *optimal* estimates of (\mathbf{c}, γ) corresponding to the true underlying image \mathbf{u} based on observations $\mathcal{G} = \{\mathbf{g}_\tau : \tau \in \mathcal{I}_t\}$, where \mathcal{I}_t denotes the set of time instants of images \mathbf{g} used in the estimation, i.e., $\mathcal{I}_t = \{\tau : \mathbf{g}_\tau \text{ is used in the estimation of } (\mathbf{c}, \gamma)\}$.

4.3 Motion trajectory models

A trajectory $\mathbf{c}(\tau; \mathbf{x}, t)$ mathematically describes the 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 with constant velocity $\mathbf{v}(\mathbf{c}(\tau; \mathbf{x}, t), \tau) = \mathbf{v}(\mathbf{x}, t)$, we define a *displacement* \mathbf{d} as follows

$$\mathbf{d}(\tau; \mathbf{x}, t) = \mathbf{v}(\mathbf{x}, t) \cdot (\tau - t).$$

Then, the associated linear trajectory can be expressed by

$$\mathbf{c}(\tau; \mathbf{x}, t) = \mathbf{x} + \mathbf{v}(\mathbf{x}, t) \cdot (\tau - t). \quad (4.1)$$

Consequently, for linear motion the task is to find, for each pixel (\mathbf{x}, t) , the two components of the velocity $\mathbf{v}(\mathbf{x}, t)$ or displacement $\mathbf{d}(\tau; \mathbf{x}, t)$. This is the predominant model used to date.

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

$$\mathbf{c}(\tau; \mathbf{x}, t) = \mathbf{x} + \mathbf{v}(\mathbf{x}, t) \cdot (\tau - t) + \frac{1}{2} \cdot \mathbf{a}(\mathbf{x}, t) \cdot (\tau - t)^2. \quad (4.2)$$

The 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 accounting for second-order effects. This model is relatively new and only recently has it been demonstrated to benefit motion computation both in the Fourier-transform domain [7] and in the original space-time domain [5]. Furthermore, the model could be extended to higher-order effects (i.e., derivative of acceleration) as suggested in [9].

Since trajectories \mathbf{c} have been written so far as general functions of \mathbf{x} and t , they 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

³In order to facilitate understanding of the models, a brief review of Markov random fields, Gibbs distributions and the relationship between them is given in Appendix A.

by a parametric function $\mathbf{c}^p(\tau; \mathbf{x}, t)$ uniquely identified by the parameter vector \mathbf{p} . With this assumption \mathbf{c}^p belongs to a finite-dimensional space.

In particular, for the linear trajectory model (4.1) the parameter vector \mathbf{p}_i takes the following form:

$$\mathbf{p}_i = [v_x(\mathbf{x}_i, t) \ v_y(\mathbf{x}_i, t)]^T,$$

and the search for an estimate is executed in R^2 . Similarly, each quadratic trajectory (4.2) is described by the following parameter vector in R^4 :

$$\mathbf{p}_i = [v_x(\mathbf{x}_i, t) \ v_y(\mathbf{x}_i, t) \ a_x(\mathbf{x}_i, t) \ a_y(\mathbf{x}_i, t)]^T$$

The above trajectory models have been developed for pixel-based algorithms, however one can easily imagine extending those models to blocks or arbitrarily-shaped regions. The linear trajectory model over square blocks has been used very successfully in the current video compression standards. Also, an acceleration has been allowed implicitly in the so-called “B”-frame mode of MPEG (independent backward and forward motion vectors). However, no quadratic trajectory models defined on blocks have been studied to date over more than three frames.

Similarly, only temporally-linear motion models have been studied to date in the case of arbitrarily-shaped regions, e.g., spatially-constant, spatially-quadratic or affine. The quadratic trajectory model (4.2) suggests an extension of region-based motion models to include acceleration. This is discussed in detail in Sections 4.7 and 4.8. In consequence, the temporally-quadratic model would involve estimation of motion parameters using multiple frames as opposed to two frames used today.

4.4 MAP and other Bayesian criteria

Since the goal of this chapter is to present statistical models for motion computation, we only consider estimation criteria derived from the statistical basis. Moreover, we constrain ourselves to a narrower class of Bayesian criteria due to their flexibility and practical value as demonstrated in various image processing [16] and computer vision [27] applications. In this class the *maximum a posteriori probability* (MAP) criterion has been explored first and consequently has been adopted by several researchers (image reconstruction [16], image segmentation [11, 29], motion segmentation [32, 6], motion estimation [26, 19]).

For the problem of joint estimation of motion trajectory field \mathbf{c} and its attribute γ , MAP estimation can be described as follows:

$$(\hat{\mathbf{c}}, \hat{\gamma}) = \arg \max_{(\mathbf{c}, \gamma)} P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G}) \quad (4.3)$$

where $(\hat{\mathbf{c}}, \hat{\gamma})$ is the MAP estimate of both fields. Above, the distribution $P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G})$ denotes a probability mass function where \mathbf{c} may be discrete- or continuous-valued and γ is discrete-valued.

Using the Bayes rule, the above maximization can be rewritten

$$(\hat{\mathbf{c}}, \hat{\gamma}) = \arg \max_{(\mathbf{c}, \gamma)} P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n}) \cdot P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathbf{g}_{t_n}). \quad (4.4)$$

Above, $t_n \in \mathcal{I}_t$ is an arbitrarily chosen time instant from \mathcal{I}_t and $\mathcal{G}^n = \{\mathbf{g}_\tau : \tau \in \mathcal{I}_t - \{t_n\}\}$. \mathbf{G}^n denotes a suitable random field for \mathcal{G}^n . To solve (4.4) for $(\hat{\mathbf{c}}, \hat{\gamma})$, the likelihood $P(\mathbf{G}^n =$

$\mathcal{G}^n|\mathbf{c}, \gamma, \mathbf{g}_{t_n}$) and the *a priori* probability distribution $P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma|\mathbf{g}_{t_n})$ must be known. These distributions are uniquely defined by the models to be described in the subsequent sections: the likelihood model is responsible for the relationship between the estimated motion (and/or its attributes) and the images, whereas the prior model takes care of the assumed properties of motion and attribute fields.

Another estimation criterion frequently used for the recovery of a continuously-changing characteristic, such as image intensity, has been the *mean-squared error* (MSE) [18]. It is not suitable, however, for problems where a generic variable is sought, e.g., occlusion or segmentation map of an image. To handle such cases, a more general class of criteria based on the Bayes risk, also referred to as *minimum expected cost* (MEC) criteria, can be used. In such a criterion, the Bayes *risk* R , defined as the expectation of a cost functional J (also called *loss*) measuring the discrepancy between the estimate and a random variable, is minimized:

$$R(\hat{\mathbf{c}}) = \int J(\mathbf{C}, \hat{\mathbf{c}})P(\mathbf{C}|\mathcal{G})d\mathbf{C}.$$

The special case of the MAP criterion (4.3) can be defined by a “hit or miss” loss function in the limit of $\Delta \rightarrow 0$:

$$J_1(\mathbf{C}, \hat{\mathbf{c}}) = -\frac{1}{\Delta^P} \text{rect}\left(\frac{\mathbf{C} - \hat{\mathbf{c}}}{\Delta}\right) \quad (4.5)$$

provided, the limit exists [35]. Above, $\text{rect}(\cdot)$ denotes the P -dimensional rectangular function:

$$\text{rect}(\mathbf{C}) = \begin{cases} 1 & \text{if } |C_i| < 0.5 \quad \forall i \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{C} = [C_1 C_2 \dots C_P]^T$.

Criteria other than MAP are well-known in the estimation theory, but have rarely been used for computer vision problems. Some examples in the context of image reconstruction and stereo matching are reported in [27].

In what follows, we concentrate on two Bayesian criteria for discrete random processes that seem particularly relevant to the problems studied here. For the estimation of motion trajectories \mathbf{c} , the minimum MSE (MMSE) estimation is defined based on the L^2 norm $\|\cdot\|$:

$$J_2(\mathbf{C}, \hat{\mathbf{c}}) = \sum_{i=1}^M \|\mathbf{C}(\mathbf{x}_i) - \hat{\mathbf{c}}(\mathbf{x}_i)\|^2. \quad (4.6)$$

The above criterion has been studied in the context of constant-velocity motion (displacement estimation) [26]. Although expected to outperform the MAP criterion for noisy observations, as reported for image segmentation [28], the MMSE criterion performed similarly. Possible reasons for this are discussed in [26], but further studies are needed.

Another Bayesian criterion given by the loss function

$$J_3(\Gamma, \hat{\gamma}) = \sum_{i=1}^M \{1 - \delta(\Gamma(\mathbf{x}_i) - \hat{\gamma}(\mathbf{x}_i))\}, \quad (4.7)$$

leads to the *maximum marginal a posteriori probability* (MMAP) estimate. Since $\delta(\cdot)$ is the discrete impulse function, the error at each site \mathbf{x}_i is binary and does not grow with the discrepancy

between Γ and $\hat{\gamma}$ as is the case in (4.6). Compared to the MAP estimation (loss function (4.5)) that maximizes the overall *a posteriori* distribution, MEC estimation based on the loss function J_3 maximizes the *a posteriori* marginal distribution at each site \mathbf{x}_i . Although seemingly less interesting, the MMAP estimation has been reported to perform well, and often better than the MAP estimation, in binary image segmentation under high noise levels [28]. Intuitively this may be explained as follows. For a high SNR the expected optimal segmentation error is close to zero, so that MAP and MMAP estimates coincide. If the SNR is low, however, the MAP estimator tends to be too conservative; one or dozens of mistakes are equally costly. On the other hand, in the case of the MMAP estimator few mistakes have only a marginal effect on the expected cost; the estimator *can* make a better, although a more risky, guess.

It can be shown, that for discrete-valued \mathbf{c} and for criterion (4.6) the optimal estimate $\hat{\mathbf{c}}(\mathbf{x}_i)$ is the value closest to the posterior mean

$$\bar{\mathbf{c}}(\mathbf{x}_i) = \sum_{\mathbf{c}} \sum_{\gamma} \mathbf{c}(\mathbf{x}_i) P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G}). \quad (4.8)$$

Thus, the optimal estimate $\hat{\mathbf{c}}$ is obtained by first computing the posterior mean $\bar{\mathbf{c}}$ (conditioned on \mathcal{G}) and then by selecting a vector \mathbf{c} closest to $\bar{\mathbf{c}}$. For a continuous-valued \mathbf{c} , the first summation in (4.8) would be replaced by an integral, and $\hat{\mathbf{c}}$ would be equal to $\bar{\mathbf{c}}$.

Similarly, it can be shown that for criterion (4.7), the optimal estimate $\hat{\gamma}(\mathbf{x}_i)$ is the value q that maximizes the posterior marginal:

$$P_i(q) = \sum_{\mathbf{c}} \sum_{\gamma: \gamma(\mathbf{x}_i)=q} P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G}). \quad (4.9)$$

Again, first the posterior marginal $P_i(q)$ must be computed for all q from the state space of $\gamma(\mathbf{x}_i)$, and then the optimal estimate $\hat{\gamma}$ is found by selecting q that gives the highest posterior marginal $P_i(q)$.

Note that, similarly to the MAP estimation, the posterior distribution $P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G})$ is needed in both cases of the MEC estimation. Using the Bayes rule, this distribution can be expressed as follows:

$$P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G}) = \frac{P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n}) P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathbf{g}_{t_n})}{P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{g}_{t_n})}. \quad (4.10)$$

The fact that the denominator is independent of (\mathbf{c}, γ) permits an alternative formulation of the MAP estimation (4.4). Since \mathbf{C} and Γ are MRFs we know, by the Hammersley-Clifford theorem [3], that $P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G})$ is a Gibbs distribution (Appendix A)

$$P(\mathbf{C} = \mathbf{c}, \Gamma = \gamma | \mathcal{G}) = \frac{1}{Z} e^{-U(\mathbf{c}, \gamma, \mathcal{G})/\beta}.$$

Due to this particular functional form of the posterior distribution, the MAP estimation can be reduced to the minimization

$$(\hat{\mathbf{c}}, \hat{\gamma}) = \arg \min_{(\mathbf{c}, \gamma)} U(\mathbf{c}, \gamma, \mathcal{G})$$

that can be performed in various ways. The global optimum of U can be achieved by simulated annealing based on the Metropolis algorithm [30] or Gibbs sampler [16]. In each case a Markov

chain is generated for \mathbf{c} and γ that under certain conditions [16] converges to the global minimum of U . In the Metropolis algorithm, states are generated from the uniform distribution followed by acceptance/rejection based on decrement/increment of energy U . To allow occasional energy increase in search for the global minimum, states with energy increment may be accepted as well, however with lower probability. In the Gibbs sampler, under the assumption that $\hat{\gamma}$ is known (e.g., from previous iteration), random states are generated for $\hat{\mathbf{c}}(\mathbf{x}_i)$ based on the marginal probability $P(\mathbf{c}(\mathbf{x}_i)|\mathbf{c}(\mathbf{x}_j), i \neq j, \hat{\gamma}, \mathcal{G})$. Similarly, the Gibbs sampler is applied to $\gamma(\mathbf{x}_i)$ according to $P(\gamma(\mathbf{x}_i)|\gamma(\mathbf{x}_j), i \neq j, \hat{\mathbf{c}}, \mathcal{G})$. Simulated annealing based on the Gibbs sampler has been proven to converge, under certain conditions, to the global optimum [16]. Unfortunately the method is very intensive computationally. It has been argued, however, that almost optimal results can be achieved for some problems by less intensive deterministic minimization algorithms such as the “highest confidence first” method [8], the multiscale implementation [20] of the “iterated conditional modes” [4] or the multiresolution implementation of the Gauss-Newton minimization [25].

In order to find MEC estimates directly from equations (4.8) and (4.9), summations over all possible configurations of \mathbf{c} and γ must be carried out. For large M and finely-quantized \mathbf{c} or γ this may be prohibitively expensive. To alleviate the problem, Marroquin [28] has proposed to exploit statistical properties of Markov chains generated by the Metropolis algorithm or the Gibbs sampler. In particular, he proposed to exploit the regularity of the generated Markov chains and to approximate the posterior mean (4.8) by a sample mean and the posterior marginal distribution (4.9) by a frequency of occurrence. For sufficiently long chains both provide a good approximation and have been used in practice (image reconstruction [27], motion estimation [26]).

A few words about the relative merits of both criteria types are in order. While the MEC criteria based on the loss functions J_2 and J_3 draw expectation over large domains and therefore are more resilient to noise in the observations, various fast statistical and deterministic optimization methods lend themselves to an approximation of the MAP estimate. This is important from the computational point of view since most statistical estimation methods are highly intensive computationally. If we are bound, however, to use a statistical solution method (e.g., the Gibbs sampler), the computation of the posterior marginal or mean does not require establishing the delicate annealing schedule unlike in the case of simulated annealing (MAP). Moreover, coarse estimates can be established very rapidly, and subsequently refined to a higher precision.

4.5 Models for the likelihood distribution

The likelihood $P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n})$, responsible for the relationship between motion and image sequence, is an essential element of the measurement of (unobservable) motion. This relationship, called the *structural model*, addresses the direct problem of projecting moving objects onto the image plane. This has been traditionally studied at the level of pixels or blocks of pixels. Recently, also arbitrarily-shaped regions of an image have been used since they are expected to provide more stable properties.

Since we are interested in resolving the correspondence problem between few neighboring images only, a natural (and frequent) hypothesis made is that image brightness along motion trajectories be constant [34, 21]. This can be expressed as follows:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{c}(\tau; \mathbf{x}, t), \tau), \quad \forall \mathbf{x} \in \mathcal{R}, \quad (4.11)$$

where \mathbf{u} is the K -component underlying image and \mathcal{R} denotes an image area where this relationship applies (pixel, block of pixels or arbitrarily-shaped region).

Obviously, the relationship (4.11) does not hold when a change of scene illumination plays a role. To handle such a departure from brightness or color constancy, another approach must be used. One possibility is to model the difference between both sides of equation (4.11) by a smooth function representing a slowly varying illumination error [17, 31]. This allows handling of arbitrary illumination effects, but requires estimation of an additional field. Another possibility is to use an image property that is more stable under illumination change, e.g., a spatial variation of intensity. This reasoning has led to the following structural model:

$$\nabla \mathbf{u}(\mathbf{x}, t) = \nabla \mathbf{u}(\mathbf{c}(\tau; \mathbf{x}, t), \tau), \quad \forall \mathbf{x} \in \mathcal{R}, \quad (4.12)$$

where $\nabla = [\frac{\partial}{\partial x} \ \frac{\partial}{\partial y}]^T$ denotes the spatial gradient (we consider $\nabla \mathbf{u}$ to be a $2K$ -dimensional vector). The model (4.12) has been originally applied in a non-Bayesian context [41, 2], but later was also incorporated into MAP estimation [42].

Equations (4.11) and (4.12) 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 $\tilde{\mathbf{g}}$ of \mathbf{u} by a suitable operation (e.g., spatial interpolation) on the observed data \mathbf{g} . Then,

$$\tilde{\mathbf{g}}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, t) + \mathbf{e}(\mathbf{x}, t),$$

where \mathbf{e} is a K -component estimation error. Exploiting equation (4.11) we can write:

$$\begin{aligned} \tilde{\mathbf{g}}(\mathbf{x}, t) - \tilde{\mathbf{g}}(\mathbf{c}(\tau; \mathbf{x}, t), \tau) &= \\ \mathbf{e}(\mathbf{x}, t) - \mathbf{e}(\mathbf{c}(\tau; \mathbf{x}, t), \tau) &\triangleq \boldsymbol{\chi}(\mathbf{x}, t, \tau), \quad \forall \mathbf{x} \in \mathcal{R}, \end{aligned} \quad (4.13)$$

and for equation (4.12) we have

$$\nabla \tilde{\mathbf{g}}(\mathbf{x}, t) - \nabla \tilde{\mathbf{g}}(\mathbf{c}(\tau; \mathbf{x}, t), \tau) \triangleq \boldsymbol{\varsigma}(\mathbf{x}, t, \tau), \quad \forall \mathbf{x} \in \mathcal{R}. \quad (4.14)$$

$\boldsymbol{\chi}(\mathbf{x}, t, \tau)$ and $\boldsymbol{\varsigma}(\mathbf{x}, t, \tau)$ are noise-like K - and $2K$ -component terms, respectively, governed by a probability distribution depending on the statistics of the estimation error \mathbf{e} or its gradient. We assume that $\boldsymbol{\chi}(\mathbf{x}, t, \tau)$ and $\boldsymbol{\varsigma}(\mathbf{x}, t, \tau)$ are independent of $\tilde{\mathbf{g}}$.

Having proposed the structural model, we need to establish the functional form of the likelihood $P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n})$. Note that the pairwise differences of the interpolated image along motion trajectories have the properties of a random noise. Therefore, we assume that noise terms $\boldsymbol{\chi}(\mathbf{x}, t, \tau)$ and $\boldsymbol{\varsigma}(\mathbf{x}, t, \tau)$ depend only on the variability of $\tilde{\mathbf{g}}$ along motion trajectories, and that this variability for each trajectory is independent of variabilities along other trajectories. Thus, we assume that the likelihood for each trajectory through (\mathbf{x}, t) is Gibbsian:

$$P_{\mathbf{x}}(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n}) = \frac{1}{Z_{g_{\mathbf{x}}}} e^{-U_{g_{\mathbf{x}}}(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma) / \beta_g}, \quad (4.15)$$

where $\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}} = \{\tilde{\mathbf{g}}(\mathbf{c}(\tau; \mathbf{x}, t), \tau) : \tau \in \mathcal{I}_t\}$ is the set of interpolated observations along the trajectory through (\mathbf{x}, t) and $U_{g_{\mathbf{x}}}$ is a Gibbs energy function. Hence, the total likelihood is a product of distributions (4.15)

$$P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n}) = \prod_{\mathbf{x}} P_{\mathbf{x}}(\mathbf{G}^n = \mathcal{G}^n | \mathbf{c}, \gamma, \mathbf{g}_{t_n}) = \frac{1}{Z_g} e^{-U_g(\mathcal{G}, \mathbf{c}, \gamma) / \beta_g}, \quad (4.16)$$

where

$$U_g(\mathcal{G}, \mathbf{c}, \gamma) = \sum_{\mathbf{x}} U_{g_{\mathbf{x}}}(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma). \quad (4.17)$$

In the above formulation, we assume that for physical reasons trajectories do not intersect (assumed indirectly by excluding transparent and reflecting surfaces). We also assume that the trajectory $\mathbf{c}(\tau; \mathbf{x}, t)$ extends throughout the whole range of \mathcal{I}_t . However, if occlusions are taken into account, only pixels that are visible should contribute to the energy $U_{g_{\mathbf{x}}}(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma)$. The occlusion field o 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) a set $\mathcal{I}_t^{\mathbf{x}}$ can be defined. This set, called a *visibility set*, contains time instants from \mathcal{I}_t at which pixel (feature) from position (\mathbf{x}, t) is still visible. For examples and a discussion of the visibility sets see [5].

There exists a considerable flexibility in the choice of the form of $U_{g_{\mathbf{x}}}(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma)$, 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

$$U_{g_{\mathbf{x}}}(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma) = \sum_{\theta_g} V_g(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma, \theta_g), \quad (4.18)$$

where θ_g is a clique defined on a suitable neighborhood system. Since the energy function $U_{g_{\mathbf{x}}}$ expresses the variability of \mathbf{g} along a trajectory, we must use at least two-element cliques in any of our models. Note, that up to this point our discussion of the likelihood was independent of the structural models. Thus, for the case of the structural model described by (4.11) and for two-element cliques of the form $\theta_g = \{\tau_1, \tau_2\}$, a possible potential function is

$$V'_g(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma, \theta_g) = \|\tilde{\mathbf{g}}(\mathbf{c}(\tau_1; \mathbf{x}, t), \tau_1) - \tilde{\mathbf{g}}(\mathbf{c}(\tau_2; \mathbf{x}, t), \tau_2)\|^2 \cdot \vartheta(\gamma, \tau_1, \tau_2) \quad (4.19)$$

where $\|\cdot\|$ is a suitable norm on the K -dimensional observation space and $\vartheta(\cdot)$ is a consistency function measuring whether matching is applicable. For example, for $\gamma = o$, ϑ should return 1 if a pixel is neither occluded nor exposed between τ_1 and τ_2 (normal matching), and 0 otherwise (matching makes no sense); sufficient penalty for the introduction of an occlusion label must discourage labeling all pixels as occluded or exposed. Similarly for $\gamma = s$, ϑ should return 1 if both pixels belong to the same object and 0 otherwise. The above potential penalizes deviation from intensity constancy along a trajectory and is often referred to as the *displaced pixel difference* (DPD). Similarly, for the other structural model (equation (4.12)), we have

$$V''_g(\tilde{\mathbf{g}}_{\mathbf{x}}^{\mathbf{c}}, \gamma, \theta_g) = \|\nabla \tilde{\mathbf{g}}(\mathbf{c}(\tau_1; \mathbf{x}, t), \tau_1) - \nabla \tilde{\mathbf{g}}(\mathbf{c}(\tau_2; \mathbf{x}, t), \tau_2)\|^2 \cdot \vartheta(\gamma, \tau_1, \tau_2). \quad (4.20)$$

This potential, on the other hand, penalizes the departure from the constancy of intensity (spatial) gradient. It may be referred to as the *displaced gradient difference*.

Note that for the norm above we can use any quadratic form $\boldsymbol{\chi}^T \mathbf{M} \boldsymbol{\chi}$ where \mathbf{M} is a positive-definite matrix. For most applications a diagonal matrix allowing a different relative weighting for each component or even an identity matrix (L_2 norm) is probably sufficient; both lead to Gauss-Markov distribution. It is also possible to replace the quadratic norm by any, not necessarily quadratic, distance measure and to establish non-stationary structural models [38]. There are many possibilities for potentials on cliques of three or more elements; for examples see [15].

Potentials given above apply directly to the estimation of motion from two frames. One can imagine that a suitable combination of such potentials could lead to a formulation based on

multiple frames. For the L_2 norm this has been done by using the sample intensity variance along a motion trajectory instead [5]. For such a formulation to fall into the Gibbs-Markov category, the equivalence between the sample variance and a Gibbs energy must be shown; a proof is included in Appendix B. Then, for $\vartheta(\cdot) = 1$ and $\theta_g = \{\tau_1, \tau_2\}$, each energy

$$U_{g_{\mathbf{x}}}(\tilde{\mathbf{g}}_{\mathbf{x}}^c, \gamma) = \sum_{\theta_g} V'_g(\tilde{\mathbf{g}}_{\mathbf{x}}^c, \gamma, \theta_g) = \rho \sum_{\tau \in \mathcal{I}_t} \|\tilde{\mathbf{g}}(\mathbf{c}(\tau; \mathbf{x}, t), \tau) - \zeta(\mathbf{x}, t)\|_{L_2}^2 \quad (4.21)$$

expresses the variability of intensity along motion trajectory through (\mathbf{x}, t) with respect to the sample mean $\zeta(\mathbf{x}, t)$

$$\zeta(\mathbf{x}, t) = \frac{1}{\text{Card}(\mathcal{I}_t)} \sum_{\tau \in \mathcal{I}_t} \tilde{\mathbf{g}}(\mathbf{c}(\tau; \mathbf{x}, t), \tau).$$

ρ is a constant discussed in Appendix B and has no particular importance since a compromise between various energy terms in $U(\mathbf{c}, \gamma, \mathcal{G})$ is usually achieved by weighting. The dependence of the sample variance and sample mean expressions on the attribute field γ can be made through the set \mathcal{I}_t . In (4.21) the summation range (\mathcal{I}_t) is the same for each trajectory. Later, in the section discussing occlusion models, we will replace \mathcal{I}_t with visibility sets $\mathcal{I}_t^{\mathbf{x}}$ dependent on $\gamma = o$ that will permit summations over $\tau \in \mathcal{I}_t^{\mathbf{x}}$ adapted to occlusion labels. Clearly, formulation using V'_g is explicitly dependent on γ .

4.6 Pixel-based motion models

In natural images motion fields are usually smooth functions of spatial position \mathbf{x} , except at motion boundaries. This observation has led to dense motion representations assigning a set of motion parameters to each pixel [34, 21]. As an alternative to these early deterministic models a Gibbs-Markov displacement model has been proposed [23]. In this contribution we use a more general model for motion; we model trajectories \mathbf{c}^p by continuous-valued vector MRFs \mathbf{C}^p . Since the parameters \mathbf{p} , rather than trajectories \mathbf{c}^p , obey the Markov property, we replace \mathbf{c}^p by \mathbf{p} in our derivations.

Having assumed that \mathbf{p} and γ are MRFs, we define the *a priori* distribution to be Gibbsian:

$$P(\mathbf{C}^p = \mathbf{c}^p, \Gamma = \gamma | \mathbf{g}_{t_n}) \triangleq P(\mathbf{P} = \mathbf{p}, \Gamma = \gamma | \mathbf{g}_{t_n}) = \frac{1}{Z_c} e^{-U_p(\mathbf{p}, \gamma, \mathbf{g}_{t_n}) / \beta_c}, \quad (4.22)$$

with Z_c, β_c being the usual constants (Appendix A), and with

$$U_p(\mathbf{p}, \gamma, \mathbf{g}_{t_n}) = \sum_{\theta_p} V_p(\mathbf{p}, \gamma, \mathbf{g}_{t_n}, \theta_p) + U_\gamma(\gamma, \mathbf{g}_{t_n}). \quad (4.23)$$

θ_p is a clique for parameter vectors \mathbf{p} derived from neighborhood \mathcal{N}_p defined over Λ_c . V_p is a potential function essential to the characterization of underlying properties of \mathbf{c}^p . Since γ describes generic motion attributes, an energy U_γ is used for now; details will be provided for each specific case.

The size of the neighborhood defines the order of the Markov field, i.e., the distance from the farthest samples affecting the current sample. The larger the order, the more contextual the bindings of the property being modeled, but at the same time the larger the computational

complexity of the model. Cliques and potential functions are closely related; a potential function must be defined for each type of clique (i.e., one-element, two-element). Cliques are responsible for the geometrical relationship between sample locations, whereas potential functions define the functional relationship between sample values.

4.6.1 Globally-smooth motion

To assure a globally-smooth trajectory field, we omit the attribute fields γ in the Gibbs distribution (4.22), i.e., we set $U_\gamma = 0$. To model the spatial smoothness of trajectories \mathbf{c}^p , V_p must be such that adjacent similar vectors \mathbf{p} give a small value of V_p (high probability), while dissimilar ones give a large value. For two-element cliques $\theta_p = \{\mathbf{x}_i, \mathbf{x}_j\}$, an often used potential is

$$V_p(\mathbf{p}, \gamma, \mathbf{g}_{t_n}, \theta_p) = (\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{Z}(\mathbf{g}_{t_n}) (\mathbf{p}_i - \mathbf{p}_j) \cdot \kappa(\gamma, \theta_p), \quad (4.24)$$

where $\mathbf{Z}(\mathbf{g}_{t_n})$ is a positive-definite weight matrix depending on the observations and $\kappa(\gamma, \theta_p)$ is a function that expresses dependence on the motion attributes γ . Clearly, for globally-smooth motion $\kappa(\gamma, \theta_p) = 1$. This potential captures smoothness of the random field \mathbf{C}^p since for $\mathbf{p}_i = \mathbf{p}_j$, $V_p = 0$. Experience has shown that first- and second-order neighborhoods \mathcal{N}_p are sufficient for quite accurate modeling of trajectories \mathbf{c}^p .

The dependence of probability (4.22) on the observations is expressed in (4.24) through the weight matrix $\mathbf{Z}(\mathbf{g}_{t_n})$. This matrix permits different weighting of horizontal and vertical as well as of lower- and higher-order components of \mathbf{p} . If $\mathbf{Z}(\mathbf{g}_{t_n})$ is the identity matrix, the Euclidean norm results. In general, $\mathbf{Z}(\mathbf{g}_{t_n})$ 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 a local image structure. This kind of adaptive smoothness constraint for the case of linear trajectories (4.1) has been proposed in [33].

The above Gibbs-Markov model has been extensively studied for linear trajectories: with the likelihood $P(\mathbf{G}^n = \mathcal{G}^n | \mathbf{p}, \mathbf{g}_{t_n})$ based on potential (4.19) [26] as well as based on potential (4.20) [42]. A quite different likelihood and a similar prior term have been proposed in [19]. The globally-smooth model has been also studied in the case of quadratic trajectories [5], but likelihood based on the multiple-frame energy (4.21) was used.



Figure 4.2: Original frames (a) #168 and (b) #171 from QCIF sequence “Carphone”.

Although this contribution is mainly theoretical, we are presenting some experimental results in order to illustrate the possible impact of presented models on the final estimate. Fig. 4.2 shows two original frames from a QCIF sequence often used in testing compression algorithms for very low bit rates. Fig. 4.3(a) shows a globally-smooth deterministic MAP estimate obtained for the DPD model (4.19) with $\vartheta(\cdot) = 1$ and motion model (4.24) with $\kappa(\cdot) = 1$ [25] (identity matrix used for $\mathbf{Z}(\mathbf{g}_{t_n})$). Note that the motion field is very smooth and many motion vectors are underestimated, e.g., in the car window.

4.6.2 Piecewise-smooth motion

Although quite successful for some types of images, globally-smooth motion models are inappropriate in general; smoothness is enforced uniformly across the whole field. By introducing additional motion attributes γ , some of the problems associated with the global smoothness can be corrected.

Motion discontinuity models

A logical solution to oversmoothing is to prevent the smoothing at boundaries of moving objects. This can be done by selecting motion discontinuity field as the motion attribute; $\gamma = l$. Consequently, we modify the potential V_p (4.24) by redefining the function $\kappa(\cdot)$ as follows

$$\kappa(l, \theta_p) = 1 - l(\mathbf{x}_i, \mathbf{x}_j, t), \quad \theta_p = \{\mathbf{x}_i, \mathbf{x}_j\}, \quad (4.25)$$

where $\{\mathbf{x}_i, \mathbf{x}_j\}$ denotes a site of line element; \mathbf{x}_i and \mathbf{x}_j are first-order neighbors. The above choice for κ prevents smoothing when a motion discontinuity exists between sites \mathbf{x}_i and \mathbf{x}_j (i.e., $l(\mathbf{x}_i, \mathbf{x}_j, t) = 1$).

Since all motion discontinuities set to 1 would have brought the first term in (4.23) to zero, a penalty must be associated with the introduction of a motion discontinuity. This is achieved by associating the energy U_γ with a penalty:

$$U_\gamma(\gamma, \mathbf{g}_{t_n}) = \sum_{\theta_l} V_l(l, \mathbf{g}_{t_n}, \theta_l), \quad (4.26)$$

where θ_l is a clique of line elements defined over a union of two orthorhombic cosets [26]. The potential function V_l is set to a modest value for single-element cliques when $l(\mathbf{x}_i, \mathbf{x}_j, t) = 1$; introduction of a discontinuity is penalized. At the same time, it is important to note that a 3-D scene giving rise to a motion discontinuity also contributes to an intensity edge; cases when the two do not coincide are quite rare. To enforce such a coincidence, we set the same potential for single-element cliques to 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 the potential to a large value whenever motion discontinuity is attempted at locations where no intensity edge is present [22, 19]. The other approach is to make the potential inversely proportional to a norm of the local image gradient [24].

In order to model continuity of motion boundaries, a sufficiently large neighborhood system \mathcal{N}_l should be used. Example of such a neighborhood system can be found in [26]; cross-shaped cliques discourage creation of segments that intersect or do not have a continuation, while square-shaped cliques discourage formation of double lines and also inhibit generation of isolated trajectories.

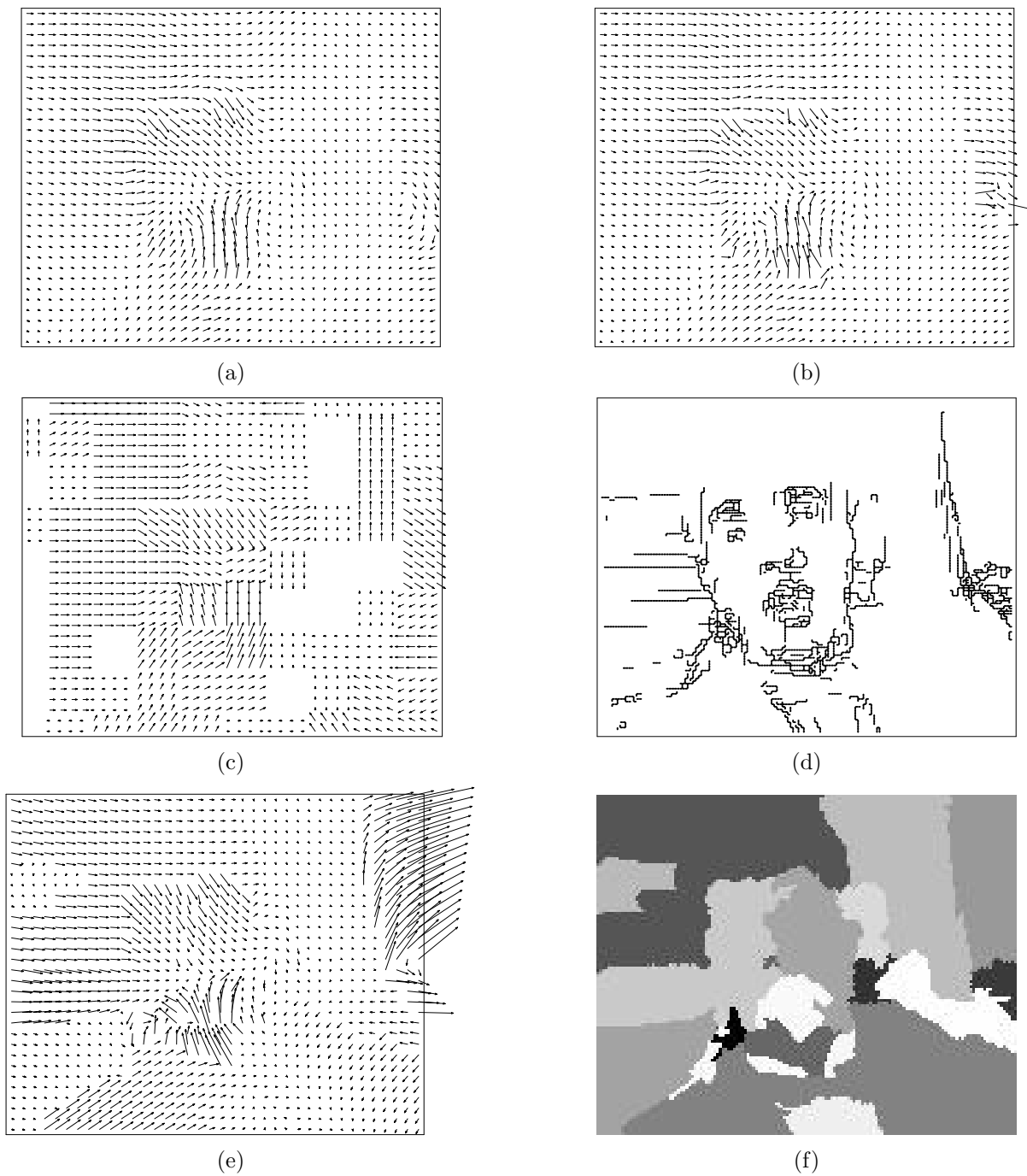


Figure 4.3: Motion estimates for the sequence from Fig. 4.2 shown as vector fields (magnified by 2 and subsampled by 4) for the following models: (a) globally-smooth pixel-based; (b,d) piecewise-smooth pixel-based; (c) block-based and (e,f) region-based. Details for each algorithm are given in the respective sections.

Fig. 4.3(b) shows a piecewise-smooth deterministic MAP estimate computed from the DPD model (4.19) with $\vartheta(\cdot) = 1$, motion model (4.24) with $\kappa(\cdot)$ from (4.25) and line potential functions V_l similar to those in [26]. Fig. 4.3(d) shows the associated line field. Note the motion field discontinuities at the face and car window boundaries⁴. However, since the line process does not model complete object boundaries, the improvement is local only; a better approach is to use a more stable boundary description such as a segmentation into regions.

Occlusion models

Occlusions play a very important role in the formation of images, and consequently in the process of estimating motion. We model occlusions by the MRF field O defined on Λ_c . The size of the state space for each occlusion label depends on $Card(\mathcal{I}_t)$; typically 3 or 5 states are used. Since the piecewise-smooth motion model has proven advantageous over the globally-smooth model, we also use the discontinuity field l . Consequently $\gamma = (o, l)$ and the energy U_γ can be defined as follows

$$U_\gamma(\gamma, \mathbf{g}_{t_n}) = \sum_{\theta_o} V_o(o, l, \mathbf{g}_{t_n}, \theta_o) + \sum_{\theta_l} V_l(l, \mathbf{g}_{t_n}, \theta_l).$$

θ_o is an occlusion clique derived from neighborhood \mathcal{N}_o defined over Λ_c , whereas the second term is the same as in (4.26).

Usually, low-order neighborhood systems with one - and two-element cliques are selected. The potential function V_o provides a penalty associated with an occlusion; otherwise energy (4.21) could be reduced freely by a suitable choice of an occlusion state. 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 two-element horizontal and vertical cliques, single-element cliques are used as well. 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 are required. The boundaries between different patches are expected to be occlusion boundaries that should coincide with motion discontinuities. Thus, the dependence of V_o on discontinuity field l should be exploited. Then, V_o should be set to 0 whenever two different occlusion states are separated by a motion discontinuity. Finally, discontinuities in areas of identical occlusion labels can be discouraged by assigning a high value to V_o whenever a discontinuity separates two identical labels.

Due to occlusions individual trajectories may start and end at different time instants. In consequence, we need to modify the likelihood distribution; only intensities of pixels that are visible should be matched. Therefore, we use the energy (4.21) with the sets \mathcal{I}_t replaced by the visibility sets \mathcal{I}_t^x . Since the visibility set \mathcal{I}_t^x is implicitly dependent on the occlusion state $o(\mathbf{x}, t)$, in the general case U_{g_x} is a function of $\gamma = o$.

In the simple case of estimation from 3 frames (at t_- , t and t_+), the direct Gibbs formulation (see the discussion of energy (4.21)) is straightforward; a good choice for U_{g_x} is

$$U_{g_x}(\tilde{\mathbf{g}}_x^c, \gamma) = V'_g(\tilde{\mathbf{g}}_x^c, \gamma, t_-, t) + V'_g(\tilde{\mathbf{g}}_x^c, \gamma, t, t_+).$$

If \mathcal{E} means that a pixel has been exposed between t_- and t , and \mathcal{C} means that a pixel has been covered between t and t_+ , then the consistency function $\vartheta(\cdot)$ for V'_g (4.19) should be selected in such a way that $\vartheta(\mathcal{E}, t_-, t) = 0$, $\vartheta(\mathcal{C}, t, t_+) = 0$, and otherwise equal to 1.

⁴Due to spatial subsampling by 4 to avoid vector overlapping, the motion discontinuities in Fig. 4.3(b) are partially masked; without subsampling they would have been very clear.

Segmentation models

Although the model based on motion discontinuities improves the quality of estimated motion fields when compared with the globally-smooth model, the improvement is rather local and often inconsistent. This is due to the fact that the modeled discontinuities are not closed and, in fact, very partitioned. A better solution to this problem is to model complete boundaries of moving objects; the contours are closed and usually simple, as are boundaries of real objects. Clearly, segmenting a motion field into disjoint objects explicitly accounts for motion discontinuities. Moreover, segmentation provides richer information and may lead to higher level tasks, such as recognition of objects. It also facilitates modeling object-specific properties such as average motion, flexibility of objects, or statistical characteristics of the prediction error [38].

If the segmentation is known, smoothness constraint can be applied to motion parameters within each segment and suspended across segment boundaries. Ideally, the segmentation should group in one segment all pixels arising from objects undergoing *one* motion. The segmentation can be represented by a generic label field $s \in \mathbb{N}$ on Λ_c . The segmentation partitions the lattice Λ_c into N disjoint subsets $\Lambda_1, \Lambda_2, \dots, \Lambda_N$, with $(\mathbf{x}, t) \in \Lambda_{s(\mathbf{x}, t)}$; for example $\Lambda_{s(\mathbf{x}, t)} = \Lambda_1$ for $s(\mathbf{x}, t) = 1$. These subsets form arbitrarily-shaped regions whose union is Λ_c . Since the smoothness constraint is to be suspended across segment boundaries, the potential V_p from (4.24) needs to be redefined by selecting a new function $\kappa(\cdot)$

$$\kappa(s, \theta_p) = \delta(s(\mathbf{x}_i, t) - s(\mathbf{x}_j, t)), \quad \theta_p = \{\mathbf{x}_i, \mathbf{x}_j\}. \quad (4.27)$$

Considering that motion trajectory \mathbf{c} describes the dynamics of one physical point on the image plane, its label s is expected to be constant along \mathbf{c} . At the same time, it is natural to expect spatially compact boundaries in the segmentation field. This can be formulated through a potential function by assigning a cost to any two-element clique that includes sites with different labels. Such two-element cliques may extend in the spatial direction or in the temporal direction along motion trajectories [37, 6]. By incorporating occlusions as well, we have $\gamma = (o, s)$ and we can define the energy U_γ as follows:

$$U_\gamma(\gamma, \mathbf{g}_{t_n}) = \sum_{\theta_o} V_o(o, s, \mathbf{g}_{t_n}, \theta_o) + \sum_{\theta_s} V_s(s, \mathbf{g}_{t_n}, \theta_s) + \sum_{\mathbf{x}} \sum_{\theta_t} V_t(s, \mathbf{g}_{t_n}, \theta_t, \mathbf{x}),$$

with potential functions

$$\begin{aligned} V_s(s, \mathbf{g}_{t_n}, \theta_s) &= \lambda_s(\mathbf{x}_i, \mathbf{x}_j)[1 - \delta(s(\mathbf{x}_i, t) - s(\mathbf{x}_j, t))], \quad \theta_s = \{\mathbf{x}_i, \mathbf{x}_j\} \\ V_t(s, \mathbf{g}_{t_n}, \theta_t, \mathbf{x}) &= \lambda_t(\mathbf{x}, t, \tau)[1 - \delta(s(\mathbf{x}, t) - \tilde{s}(\mathbf{c}(\tau; \mathbf{x}, t), \tau))], \quad \theta_t = \{t, \tau\}. \end{aligned}$$

\tilde{s} denotes a spatially-interpolated version of the segmentation s . The potential $V_o(o, s, \mathbf{g}_{t_n}, \theta_o)$ is the same as defined in the previous section, except for the line process which is replaced by the segmentation. The weight $\lambda_s(\cdot)$ determines directional preferences of typical realizations of the segmentation. For orthorhombic lattices directional preferences are reduced to a minimum for $\lambda_s(\mathbf{x}_i, \mathbf{x}_j) = \alpha/\|\mathbf{x}_i - \mathbf{x}_j\|$, where α is a constant and $\|\cdot\|$ denotes Euclidean norm. λ_t may be assigned a large value thus inhibiting the change of label along a motion trajectory. The above energy function assures good results already for low-order neighborhood systems in spatial direction (first- or second-order).

The weight λ_s may be made dependent on the observations \mathbf{g}_{t_n} . This would allow to assign higher cost to locations where intensity gradient is low, and therefore make a moving object boundary unlikely if not accompanied by a substantial gradient in the observation.

4.7 Block-based motion models

As pointed out in the introduction, various regions of support for the motion model may be used. In the case of pixel-based models, the amount of motion data is very large (at least 2 parameters per pixel). This is a stumbling block in video compression when motion needs to be transmitted in order to eliminate temporal redundancy. A compromise accepted in this case is to assign the same displacement vector (linear trajectory model) to a group of adjacent pixels, in particular to a rectangular block. Although not as precise in representing motion as the pixel-based trajectory model, block-based linear model proved very successful due to a remarkable reduction of motion information to transmit. The estimation method for such a model, called block matching, has become ubiquitous in all standard video encoders today.

The basic difference between pixel- and block-based models lies in the region of support. If we partition image \mathbf{g}_t into rectangular disjoint blocks $\mathcal{B}(\mathbf{x}, t)$ where \mathbf{x} is the block's center, then these centers form a new lattice Λ_c^* , which can be treated in the same way as Λ_c in the pixel-based case; each site has a vector \mathbf{p} of motion parameters assigned.

One possible extension of the basic block-based model would be to allow non-constant velocity in time. For example, the quadratic trajectory model (4.2) could be used. Then, $\mathbf{c}(\tau; \mathbf{x}_0, t)$ would describe motion of the block $\mathcal{B}(\mathbf{x}_0, t)$. As mentioned before, an implicit acceleration is allowed in the “B”-frame mode of MPEG.

A popular model to describe 2-D motion of a group of pixels is the affine model derived from 3-D planar rigid surface under parallel projection:

$$\mathbf{d}^p(\mathbf{x}) = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} p_3 & p_5 \\ p_4 & p_6 \end{bmatrix} (\mathbf{x} - \mathbf{x}_0), \quad \mathbf{x} \in \mathcal{B}(\mathbf{x}_0, t) \quad (4.28)$$

where \mathbf{x}_0 is a reference point, often the center of gravity of a region, and $\mathbf{p} = [p_1, \dots, p_6]^T$. Models with a higher number of parameters have also been proposed [43, 13, 39].

Another interesting extension would be to combine the quadratic trajectory model (4.2) with the affine model (4.28) and thus allow a spatial variation of velocity and/or acceleration within a block. A possible trajectory model is

$$\begin{aligned} \mathbf{c}(\tau; \mathbf{x}, t) = \mathbf{x} &+ \left(\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} p_3 & p_5 \\ p_4 & p_6 \end{bmatrix} (\mathbf{x} - \mathbf{x}_0) \right) (\tau - t) \\ &+ \left(\begin{bmatrix} p_7 \\ p_8 \end{bmatrix} + \begin{bmatrix} p_9 & p_{11} \\ p_{10} & p_{12} \end{bmatrix} (\mathbf{x} - \mathbf{x}_0) \right) (\tau - t)^2, \quad \mathbf{x} \in \mathcal{B}(\mathbf{x}_0, t) \end{aligned} \quad (4.29)$$

where $\mathbf{p} = [p_1, \dots, p_{12}]^T$. Although seemingly complicated, such a model could prove interesting for coding/processing over multiple frames but care would have to be taken of occlusions that start to play critical role in multiple-frame processing [5].

If blocks \mathcal{B} are sufficiently small compared to the size of moving objects, then adjacent blocks should have similar motion parameters \mathbf{p} unless separated by an object boundary. This suggests using a similar globally-smooth motion model as exploited for the pixel case. Note that expressions for the posterior distribution and for MAP estimation similar to those presented in Section 4.6 can be now written for the new lattice Λ_c^* .

A MAP estimation for affine block-based motion (no acceleration) has been proposed in [1] *via* the following maximization

$$\max_{\mathbf{p}} P(\mathbf{G}_{t_{n+1}} = \mathbf{g}_{t_{n+1}} | \mathbf{p}, \mathbf{g}_{t_n}) P(\mathbf{P} = \mathbf{p} | \mathbf{g}_{t_n}).$$

Modeling the likelihood by a Gaussian and the prior by a Gibbs distribution, the equivalent minimization can be written as

$$\min_{\mathbf{p}} \sum_i \sum_{\mathbf{x} \in \mathcal{B}(\mathbf{y}_i, t)} V'_g(\tilde{\mathbf{g}}_{\mathbf{x}}^c, t_n, t_{n+1}) + \lambda \sum_{\theta_B = \{k, l\}} \|\mathbf{p}_k - \mathbf{p}_l^k\|^2 + \|\mathbf{p}_l - \mathbf{p}_k^l\|^2, \quad (4.30)$$

where θ_B is a two-element block clique defined over Λ_c^* and \mathbf{p}_l^k denotes parameter vector for block l expressed with respect to the reference point of block k , e.g., the center of gravity of block k . The first term above assures matching of a block via affine parameters, whereas the second provides the smoothness between adjacent blocks. The smoothness term is slightly different from (4.24) because we seek smoothness of the motion field \mathbf{d}^p but our formulation is with respect to the parameter field \mathbf{p} . In order to compare motion parameters of neighboring blocks in a meaningful way, they need to be expressed with respect to a common point, e.g., center of gravity of one of the blocks. The two smoothness terms above assure symmetrical comparison, although for floating-point computations one term should suffice. Clearly, in the case of translational motion (only $p_1, p_2 \neq 0$) the two smoothness terms in (4.30) simplify to $\|\mathbf{p}_k - \mathbf{p}_l\|^2$.

With the above formulation the annoying motion discontinuities at block boundaries can be largely suppressed, and consequently the resulting prediction error can be made more uniform (less block structure). This appealing property comes at a substantial increase of computational complexity, and therefore did not find way into today's compression standards.

Fig. 4.3(c) shows a block-based estimate obtained from minimization (4.30) for the case of linear trajectories (only $p_1, p_2 \neq 0$ in (4.29)). Note that despite the applied smoothing, blocks are still quite visible. This visibility could have been reduced by increasing λ , however at the cost of increased prediction error (important for coding).

4.8 Region-based motion models

Although the block support for motion model dramatically reduces the amount of motion information to be transmitted (as compared to the pixel support), it suffers from a major shortcoming. In the case of video coding, when the available bit rate is relatively small, blocks become clearly visible in the compressed images; the lower the rate, the more visible the blocks. The inter-block smoothness constraint discussed in the previous section rectifies the problem slightly, however for very low rates the problem persists. A natural extension, although at the cost of increased computational complexity, is motion representation based on arbitrarily-shaped regions. In such a representation a vector of motion parameters \mathbf{p} is assigned to a group of pixels (region) undergoing similar motion (usually a projection of a single 3-D object). The benefit, somewhat offsetting the high computational complexity, is the potential for the so-called "coding for content" where objects could be separately encoded, extracted, manipulated; in the receiver to reconstruct one object, only a sub-stream would need to be extracted and decoded.

Recall from Section 4.6 that the partition $\{\Lambda_1, \Lambda_2, \dots, \Lambda_N\}$ and segmentation $s(\mathbf{x}, t)$ are two complementary ways of describing a division of image sequence into disjoint sets. Clearly, they can be uniquely computed from each other: $s(\mathbf{x}, t) = n$ if and only if $\mathbf{x} \in \Lambda_n$.

From this point on we are going to work at the level of regions, i.e., given a partition $\{\Lambda_1, \Lambda_2, \dots, \Lambda_N\}$ we define an irregular grid using the centers of gravity of each Λ_n ; notions of adjacency and neighborhood are clear.

The pivoting point of our approach is the observation that motion boundaries most often coincide with intensity boundaries. A subset of the boundaries from a good intensity-driven segmentation should closely approximate most of the motion boundaries and therefore could be used as an initial segmentation. In order not to miss motion boundaries during subsequent stages, the initial intensity-based partition should be oversegmented. By subsequent merging, redundant boundaries may be eliminated. Therefore, the main idea of this approach is to first compute an oversegmented approximation to motion boundaries and then iteratively estimate \mathbf{p} and s .

In order to estimate \mathbf{p} a compromise between the *displaced region difference* (DRD) and the similarity of motion parameters among adjacent regions must be struck. Assuming the spatially-affine model⁵ (4.28) and the knowledge of images \mathbf{g}_{t-} and \mathbf{g}_t , this may be done *via* the following minimization:

$$\min_{\mathbf{p}} \sum_{n=1}^N \left[\sum_{\mathbf{x} \in \Lambda_n} V'_g(\tilde{\mathbf{g}}_{\mathbf{x}}^d, s, t_-, t) + \sum_{\theta_p = \{n,m\}} V_{sim}(\mathbf{p}_n, \mathbf{p}_m, \theta_p, N) \right]$$

where V'_g is defined in (4.19) and V_{sim} is a potential penalizing dissimilarity of motion parameters \mathbf{p} between neighboring regions. This is similar to smoothing in the block-based approach and makes sense only when far too many regions are present. In such a case the goal of smoothing is to facilitate subsequent merging of regions with similar motion. Clearly, the smoothing should be gradually disabled as the number of regions gets smaller. This can be achieved by making V_{sim} dependent on N to encourage similar motion of neighboring regions at the beginning of estimation (N large), and to discourage such similarity when N is close to optimal.

On the other hand, to estimate s a balance must be achieved between the DRD, the complexity of region boundaries and the number of regions N , for example *via*

$$\min_{\{s, N\}} \sum_{n=1}^N \left[\sum_{\mathbf{x} \in \Lambda_n} V'_g(\tilde{\mathbf{g}}_{\mathbf{x}}^d, s, t_-, t) + \sum_{\theta_p = \{n,m\}} V_s(\Lambda_n, \Lambda_m, \theta_p) \right] + \lambda N,$$

where $V_s(\Lambda_n, \Lambda_m, \theta_p)$ is a potential function penalizing complex boundary between regions n and m . The above minimization is not easy since pixel-by-pixel update of region boundaries is usually ineffective. A different strategy, for example moving the whole boundary between two regions by 1 pixel, may prove more effective. Note that we put N explicitly as the argument of the minimization to stress that both the boundary complexity and the number of regions have to be minimized. The reduction of N may be done through the pairwise merging of regions, i.e., comparison of the total cost before and after the merge.

We have carried out some preliminary investigations using the above models [10]. Figs. 4.3(e) and 4.3(f) show the motion and segmentation estimates for affine model (4.28) using both minimizations above except that inter-region smoothing was disabled ($V_{sim}=0$). Note the improved motion estimate in the car window compared to pixel- and block-based estimates. Also, the prediction error energy is significantly lower for the region-based motion estimate [10]. From the coding point of view, however, the gain is not obvious since the rate needed for motion attributes (\mathbf{p} and s) in the case of region-based approach is higher than that needed to transmit block displacements. Clearly, to minimize the overall rate, a judicious allocation of rate between

⁵Similar formulation could have been proposed for the temporally-quadratic spatially-affine motion model (4.29), but for simplicity of notation we discuss only the temporally-linear spatially-affine model (4.28).

different components (texture, motion, shape) must be carried out. This is a very active area of research today.

4.9 Summary

In this chapter we have presented Gibbs-Markov models for 2-D motion in the context of their application to video coding and processing. We have studied a trajectory model incorporating acceleration at pixel, block and region level. We have proposed a new motion model by permitting spatial variation of velocity and acceleration parameters *via* the affine model.

Although the MAP criterion seems to be the preferred choice for most of the statistically-based motion estimation algorithms, we have discussed at length the more general Bayesian criterion, including the merits of various loss functions. Consequently, we have developed the two components of the *a posteriori* distribution: the likelihood and the prior. First, we discussed various structural models leading to different likelihood functions, and then we described different prior models for motion. We have discussed in detail two types of pixel-based models: the globally smooth model and the piecewise-smooth model. In the latter model, motion discontinuities, occlusion tags and segmentation labels were exploited to prevent smoothing at moving boundaries. We have reviewed block-based models and we have suggested an extension to the affine motion model by incorporating the acceleration component. Finally, we have sketched a possible Gibbs-Markov model for region-based motion representation.

Many aspects of the models presented remain to be investigated, and the relative importance of many of its features must be established. We believe that the presented models are a useful framework for further research in advanced motion estimation algorithms.

Appendix A

Multidimensional Markov random fields have been introduced in the late 60's independently of Gibbs distributions that had been applied to the modeling of large systems of interacting particles before. The best known Gibbs distribution is probably the Ising model describing ferromagnetism. In the 70's, the important Hammersley-Clifford theorem was established stating equivalence between Gibbs distributions and Markov random fields. The founders provided only a preprint of the theorem; an early account of the theorem can be found in [3]. More details on Markov random fields and Gibbs distributions can be found in [36, 16, 12].

Let the sampling structure $\Omega = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ be a collection of sites in R^N . A collection \mathcal{N} of *neighborhoods* $\eta(\cdot)$ of all sites in Ω

$$\mathcal{N} = \{\eta(\mathbf{x}_i) : \mathbf{x}_i \in \Omega, \eta(\mathbf{x}_i) \subset \Omega, \forall \mathbf{x}_i\}$$

is a *neighborhood system* on Ω , if and only if both of the following conditions are satisfied:

1. all neighborhoods are self-excluding: $\mathbf{x}_i \notin \eta(\mathbf{x}_i)$,
2. the system is symmetric: if $\mathbf{x}_j \in \eta(\mathbf{x}_i)$, then $\mathbf{x}_i \in \eta(\mathbf{x}_j)$ for any $\mathbf{x}_i, \mathbf{x}_j \in \Omega$.

Neighborhood systems on lattices with shift-invariant neighborhoods (up to some suitable boundary conditions) are called *homogeneous*. Examples of low-order neighborhood systems for orthogonal and non-orthogonal sampling structures Ω over R^2 can be found in [16].

A random field over Ω is a multidimensional stochastic process where each site in $\mathbf{x}_i \in \Omega$ is assigned a random variable Υ_i . A vector random field has a random vector (ensemble of random variables) assigned at each site in Ω .

A *Markov random field* Υ with state space \mathcal{S} is a random field with the following properties:

1. $P(\Upsilon = v) > 0, \quad \forall v \in \mathcal{S},$
2. $P(\Upsilon_i = v_i | \Upsilon_j = v_j, \forall j \neq i) = P(\Upsilon_i = v_i | \Upsilon_j = v_j, \forall \mathbf{x}_j \in \eta(\mathbf{x}_i)), \quad \forall i, \forall v \in \mathcal{S},$

where P denotes a probability measure. 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 .

The univariate conditional distributions in the definition above are often referred to as *local characteristics* of the Markov random field. They completely specify the joint distribution (*global characteristic*) of the random field [3]

$$\frac{P(\Upsilon = v)}{P(\Upsilon = v^*)} = \prod_{i=1}^M \frac{P(\Upsilon_i = v_i | \Upsilon_j = v_j, \forall j < i, \Upsilon_k = v_k^*, \forall k > i)}{P(\Upsilon_i = v_i^* | \Upsilon_j = v_j, \forall j < i, \Upsilon_k = v_k^*, \forall k > i)}, \quad v, v^* \in \mathcal{S}.$$

Clearly, the global characteristic resulting from the local characteristics must be the same for any ordering of Ω . This imposes a consistency condition on the local characteristics which is difficult to validate in practice. For this reason, the modeling of a Markov random field by means of the local characteristics is not practical.

The latter difficulty does not occur with Gibbs distributions. In order to define the Gibbs distribution the concepts of clique and potential function are needed. A *clique* θ defined over Ω with respect to \mathcal{N} is a subset of Ω such that either θ consists of a single site or every pair of sites in θ are neighbors, i.e., $\mathbf{x}_i \in \eta(\mathbf{x}_j) \forall \mathbf{x}_i, \mathbf{x}_j \in \theta, i \neq j$. The set of all cliques is denoted by Θ .

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

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

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

$$U(v) = \sum_{\theta \in \Theta} V(v, \theta).$$

$V(v, \theta)$ is called a *potential function*, and depends only on those samples from v which belong to the clique θ . The real constant Z is called a *partition function* and normalizes the surface under π to 1. The only condition for π to be a valid probability measure is that the partition function be finite. Thus, the consistency problems that are nearly impossible to overcome in Markov random fields do not occur when Gibbs random fields are used. β is a parameter called *natural temperature*.

The equivalence, between Markov random fields and Gibbs distributions is provided through the important *Hammersley-Clifford theorem* [36, 3] which states that Υ is a MRF on Ω with respect to \mathcal{N} if and only if its probability distribution is a Gibbs distribution with respect to Ω and \mathcal{N} . This bijective 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).

Appendix B

In this appendix we demonstrate that for vector observations and for the L_2 norm, the sample variance formulation is equivalent to a Gibbs energy; this equivalence is used in equation (4.21).

To simplify the notation let $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L$ be the vector observations and let $L = \text{Card}(\mathcal{I}_t)$. We consider the sample variance (4.21) for the case of the L_2 norm. By multiplying individual terms and renaming indices we can write:

$$\begin{aligned} \sum_{i=1}^L \left\| \mathbf{y}_i - \frac{1}{L} \sum_{j=1}^L \mathbf{y}_j \right\|^2 &= \sum_{i=1}^L \left(\mathbf{y}_i - \frac{1}{L} \sum_{j=1}^L \mathbf{y}_j \right)^T \left(\mathbf{y}_i - \frac{1}{L} \sum_{j=1}^L \mathbf{y}_j \right) \\ &= \sum_{i=1}^L \mathbf{y}_i^T \mathbf{y}_i + \frac{1}{L} \left(\sum_{j=1}^L \mathbf{y}_j \right)^T \left(\sum_{j=1}^L \mathbf{y}_j \right) - \frac{2}{L} \left(\sum_{i=1}^L \mathbf{y}_i \right)^T \left(\sum_{j=1}^L \mathbf{y}_j \right) \\ &= \sum_{i=1}^L \mathbf{y}_i^T \mathbf{y}_i - \frac{1}{L} \left(\sum_{i=1}^L \mathbf{y}_i \right)^T \left(\sum_{j=1}^L \mathbf{y}_j \right). \end{aligned}$$

Rewriting the first term on the right-hand side we arrive at:

$$\begin{aligned} \sum_{i=1}^L \left\| \mathbf{y}_i - \frac{1}{L} \sum_{j=1}^L \mathbf{y}_j \right\|^2 &= \frac{1}{2L} \sum_{j=1}^L \sum_{i=1}^L \mathbf{y}_i^T \mathbf{y}_i + \frac{1}{2L} \sum_{j=1}^L \sum_{i=1}^L \mathbf{y}_j^T \mathbf{y}_j - \frac{1}{L} \left(\sum_{i=1}^L \mathbf{y}_i \right)^T \left(\sum_{j=1}^L \mathbf{y}_j \right) \\ &= \frac{1}{2L} \sum_{i=1}^L \sum_{j=1}^L (\mathbf{y}_i - \mathbf{y}_j)^T (\mathbf{y}_i - \mathbf{y}_j). \end{aligned}$$

Since the terms with $i = j$ do not contribute to the summation and since the double summation involves every clique $\{i, j\}$ twice, the factor $\frac{1}{2}$ vanishes and we have:

$$\sum_{i=1}^L \left\| \mathbf{y}_i - \frac{1}{L} \sum_{j=1}^L \mathbf{y}_j \right\|^2 = \frac{1}{L} \sum_{\{i,j\}} \|\mathbf{y}_i - \mathbf{y}_j\|^2.$$

The immediate consequence of this proof is that in a Gibbsian formulation, such as in (4.21), Gibbs energy may be replaced by the sample variance ($\rho = L$) which for multiple frames (high $\text{Card}(\mathcal{I}_t)$) is more intuitive.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Research Grant OGP0121619 (first author) and under International Canada Fellowship (second author). The assistance of V.-N. Dang in the preparation of experimental results is gratefully acknowledged.

References

- [1] 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.

- [2] M. Bertero, T. Poggio, and V. Torre, “Ill-posed problems in early vision,” *Proc. IEEE*, vol. 76, pp. 869–889, Aug. 1988.
- [3] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *J. Roy. Stat. Soc.*, vol. B 36, pp. 192–236, 1974.
- [4] J. Besag, “On the statistical analysis of dirty pictures,” *J. Roy. Stat. Soc.*, vol. B 48, pp. 259–279, 1986.
- [5] M. Chahine and J. Konrad, “Estimation and compensation of accelerated motion for temporal sequence interpolation,” *Signal Process., Image Commun.*, vol. 7, pp. 503–527, Nov. 1995.
- [6] M. Chang, M. Sezan, and A. Tekalp, “An algorithm for simultaneous motion estimation and scene segmentation,” in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. V.221–V.224, Apr. 1994.
- [7] W.-G. Chen, G. B. Giannakis, and N. Nandhakumar, “Spatio-temporal approach for time-varying image motion estimation,” in *Proc. IEEE Int. Conf. Image Processing*, pp. III.232–III.236, Nov. 1994.
- [8] 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.
- [9] G. Cortelazzo and M. Balanza, “Frequency domain analysis of translations with piecewise cubic trajectories,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-15, pp. 411–416, Apr. 1993.
- [10] V.-N. Dang, A.-R. Mansouri, and J. Konrad, “Motion estimation for region-based video coding,” in *Proc. IEEE Int. Conf. Image Processing*, pp. II.189–II.192, Oct. 1995.
- [11] H. Derin and H. Elliott, “Modeling and segmentation of noisy and textured images using Gibbs random fields,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, Jan. 1987.
- [12] H. Derin and P. Kelly, “Discrete-index Markov-type random processes,” *Proc. IEEE*, vol. 77, pp. 1485–1510, Oct. 1989.
- [13] N. Diehl, “Object-oriented motion estimation and segmentation in image sequences,” *Signal Process., Image Commun.*, vol. 3, pp. 23–56, Feb. 1991.
- [14] E. Dubois, “Motion-compensated filtering of time-varying images,” *Multidimens. Syst. Signal Process.*, vol. 3, pp. 211–239, 1992.
- [15] E. Dubois and J. Konrad, “Estimation of 2-D motion fields from image sequences with application to motion-compensated processing,” in *Motion Analysis and Image Sequence Processing* (M. Sezan and R. Lagendijk, eds.), ch. 3, pp. 53–87, Kluwer Academic Publishers, 1993.

- [16] 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.
- [17] M. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," Tech. Rep. 975, MIT Artificial Intelligence Laboratory, June 1987.
- [18] A. Habibi, "Two-dimensional Bayesian estimate of images," *Proc. IEEE*, vol. 60, pp. 878–883, July 1972.
- [19] F. Heitz and P. Bouthemy, "Multimodal estimation of discontinuous optical flow using Markov random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1217–1232, Dec. 1993.
- [20] F. Heitz, P. Perez, and P. Bouthemy, "Multiscale minimization of global energy functions in some visual recovery problems," *CVGIP: Image Underst.*, vol. 59, pp. 125–134, Jan. 1994.
- [21] B. Horn and B. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.
- [22] 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.
- [23] J. Konrad and E. Dubois, "Estimation of image motion fields: Bayesian formulation and stochastic solution," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. 1072–1075, Apr. 1988.
- [24] 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.
- [25] J. Konrad and E. Dubois, "Comparison of stochastic and deterministic solution methods in Bayesian estimation of 2D motion," *Image Vis. Comput.*, vol. 9, pp. 215–228, Aug. 1991.
- [26] 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.
- [27] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Am. Stat. Soc.*, vol. 82, pp. 76–89, Mar. 1987.
- [28] J. Marroquin, *Probabilistic solution of inverse problems*. PhD thesis, Massachusetts Institute of Technology, Dept. Electr. Eng. Comp. Sci., Sept. 1985.
- [29] R. Mester and U. Franke, "Statistical model based image segmentation using region growing, contour relaxation and classification," in *Proc. SPIE Visual Communications and Image Process.*, (Cambridge, MA), pp. 616–624, Nov. 1988.
- [30] 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.

- [31] C. Moloney and E. Dubois, "Estimation of motion fields from image sequences with illumination variation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. 2425–2428, May 1991.
- [32] 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.
- [33] 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.
- [34] A. Netravali and J. Robbins, "Motion-compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.
- [35] L. Scharf, *Statistical signal processing: detection, estimation and time series analysis*. Addison-Wesley Pub., 1990.
- [36] F. Spitzer, "Markov random fields and Gibbs ensembles," *Amer. Math. Mon.*, vol. 78, pp. 142–154, Feb. 1971.
- [37] C. Stiller, "Object oriented video coding employing dense motion fields," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. V.273–V.276, Apr. 1994.
- [38] C. Stiller and B. Hürtgen, "Combined displacement estimation and segmentation in image sequences," in *Int. Symp. on Fibre Optic Networks and Video Comm. EUROPTO*, vol. 1977, pp. 276–287, Apr. 1993.
- [39] C. Stiller and R. Suntrup, "Parametric object-motion estimation," in *Proc. Int. Symp. on Information Theory and its Applications*, pp. 633–637, Nov. 1992.
- [40] A. Tekalp, *Digital Video Processing*. Prentice Hall PTR, 1995.
- [41] 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.
- [42] P. Treves and J. Konrad, "Motion estimation and compensation under varying illumination," in *Proc. IEEE Int. Conf. Image Processing*, pp. I.373–I.377, Nov. 1994.
- [43] R. Tsai and T. Huang, "Estimating three-dimensional motion parameters of a rigid planar path," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 1147–1152, Dec. 1981.