Bayesian Estimation of Motion Vector Fields

Janusz Konrad and Eric Dubois, *Senior Member, IEEE*

Abstract— **This paper presents a new approach to the estimation of 2-D motion vector fields from time-varying images. The approach is stochastic both in its formulation and in the solution method. The formulation involves the specification of a deterministic structural model along with stochastic observation and motion field models. Two motion models are proposed: a globally smooth model based on vector Markov random fields and a piecewise smooth model derived from coupled vector-binary Markov random fields. Two estimation criteria are studied. In the maximum *a posteriori* probability (MAP) estimation, the *a posteriori* probability of motion given data is maximized, whereas in the minimum expected cost (MEC) estimation, the expectation of a certain cost function is minimized. The MAP estimation is performed via *simulated annealing*, whereas the MEC algorithm performs iteration-wise averaging. Both algorithms generate sample fields by means of *stochastic relaxation* implemented via the *Gibbs sampler*. Two versions are developed: one for a discrete state space and the other for a continuous state space. The MAP estimation is incorporated into a hierarchical environment to deal efficiently with large displacements. Numerous experimental results of application of these algorithms to natural and computer-generated images with natural and synthetic motion are shown.**

Index Terms— **Bayesian estimation, Markov random fields, motion estimation, motion modeling, optical flow, simulated annealing, stochastic relaxation, 2-D motion.**

I. Introduction

MOTION computation has long been known as one of the fundamental and difficult problems in computer vision. It has been tackled from various angles with variable success. This paper introduces a stochastic approach to the computation of motion. The approach stems from well-known concepts used in stochastic modeling and reconstruction of images such as Bayesian estimation criteria [10], [27], Markov random field (MRF) models for images [35], [10], [7], *simulated annealing*, and iterated conditional modes (ICM) as solution methods [20], [10], [4]. In this paper, we extend these criteria, models, and solution methods to continuous and discrete motion vector fields, we discuss similarities and differences with existing algorithms, and we present numerous motion estimates for synthetic and natural motion.

There are two main aspects to motion estimation: computation of 3-D motion in space and computation of 2-D motion in the image plane, which is also called *optical flow*. In this paper, we are concerned with the latter problem. Knowledge of 2-D motion in a time-varying image can be used to infer 3-D motion, e.g., of a camera with respect to the environment, or to compute structure from motion; however, it can also be used directly in motion-compensated processing (e.g., interpolation, noise reduction) or compression of image sequences.

The existing approaches to estimating 2-D motion from dynamic images can be classified as either low-level or high-level computer vision algorithms. The class of algorithms presented here belongs to the former group, along with such methods as block matching, spatio-temporal gradient, or Fourier techniques, and is characterized by computation of motion based only on simple low-level image descriptors like intensity. The high-level methods, which are not considered here, rely on image analysis to extract high-level features of the data, such as edges, object boundaries, or complete objects, and use them to solve the correspondence problem.

The problem of motion computation has proved to be difficult due to two factors: ill posedness and complexity. The problem is ill posed since many different vector fields can explain the same data (images). The complexity of the problem is dependent on its dimensionality, which is high since typically, several thousand unknowns have to be computed simultaneously.

Simple estimation algorithms like block matching frequently fail to produce good results because they rely on the data only and do not attempt to explicitly model motion fields. Consequently, every motion vector is computed from local intensity values, disregarding the motion of neighboring picture elements. To overcome this deficiency, instead of minimizing a local objective function, global formulations over the complete motion field have been used. Horn and Schunck [17] proposed a global criterion as a compromise between an error derived from the motion constraint equation and a motion smoothness error. Hildreth [16] used the difference between the measured and the estimated velocity component orthogonal to an intensity contour and allowed smoothing only along such a contour. Nagel and Enkelmann [31] extended the Horn-Schunck method by using image structure in the smoothness term, thus allowing space-variant smoothing.

All three approaches can be classified as *regularization* (of the original ill-posed correspondence problem), where the smoothness term expresses *a priori* assumptions about the properties of motion. The major drawback of the Horn-Schunck and Nagel-Enkelmann methods is that the motion vectors are confined to fixed spatio-temporal positions; hence, to perform sampling structure conversion through motion-compensated interpolation, motion fields have to be interpolated as well. In addition, the formulation using the motion constraint equation requires evaluation of image derivatives,

Manuscript received March 13, 1990; revised February 21, 1992. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Strategic Grant STR0040524. Recommended for acceptance by Associate Editor N. Ahuja.

The authors are with the Institut National de la Recherche Scientifique INRS-Télécommunications, Verdun Quebec, Canada, H3E 1H6.

IEEE Log Number 9201771.

0162–8828/92$03.00 © 1992 IEEE

which is an ill-posed problem itself. Horn and Schunck have used a space-invariant smoothing operator, but it causes substantial oversmoothing at motion boundaries. This problem has been partially solved by the Nagel and Enkelmann "oriented smoothness" operator. Both methods have been extended to handle large displacements via a hierarchical approach [12], [9]. The major drawback of Hildreth's approach is the need to know intensity contours, e.g., edges, before performing motion estimation, which is especially cumbersome if the method is to be applied to a wide class of images (e.g., video). In addition, it is not clear how the boundary estimates are propagated, especially if the contours are not closed.

Most motion estimation techniques to date, including the three examples above, have been exclusively deterministic. There are a few notable exceptions, however. Schunck [33] and Martinez [28] proposed maximum likelihood estimation of motion based on a statistical model relating observations and unknowns. Murray and Buxton [30] used a Markov random field model for optical flow segmentation and solved it using *simulated annealing*. More recently, Bouthemy and Lalande [5] and Heitz and Bouthemy [14], [15] applied Markov random field models to motion detection and segmentation and solved the problem using Besag's ICM method [4].

In that which follows, a Bayesian approach to motion estimation derived from the Bayesian reconstruction of images as proposed by Geman and Geman [10] and Marroquin [27] is proposed. In Section II, estimation criteria are formulated, suitable models are proposed, and the resulting *a posteriori* probability is derived. In Section III, solutions to the stochastic formulations are proposed via stochastic relaxation. Section IV covers hierarchical extension of the maximum *a posteriori* (MAP) probability estimation, whereas Section IV presents numerous experimental results for natural images with synthetic and natural motion.

II. Formulation

A. Terminology

Let u denote the true underlying time-varying image understood as an illuminance pattern in some image plane obtained from the observed scene via an ideal optical system. The observed image g, which is related to the underlying image u via some random transformation, is considered to be a sample of a random field G. The image g is assumed to be quantized in amplitude and sampled on lattice Λ_g in R^3 (horizontal, vertical, and temporal directions). Such a lattice is a collection of *sites* $(\boldsymbol{x}, t) \in R^3$ [8], where $\boldsymbol{x}$ and t denote spatial and temporal positions, respectively.

Let u at times $t = t_-$ and $t = t_+$ be called the preceding and the following images, respectively. Disregarding the occlusion and newly exposed areas, for every point in the preceding image, there exists a corresponding point in the following image such that both arise from the projection of the same scene point. Every such pair of points can be connected by a straight line. Since u is defined over continuous $(\boldsymbol{x}, t)$, these lines will intersect a plane located at t $(t_- < t < t_+)$ over a dense set of locations. In other words, for each $(\boldsymbol{x}, t)$, there exists a line joining corresponding image points at times t_- and t_+. Note that these lines coincide with the true motion trajectory at the end points and not necessarily between them. The true motion trajectory will intersect (in general) the plane at time t at a location different from $(\boldsymbol{x}, t)$. It is also possible that more than one such linear trajectory will pass through $(\boldsymbol{x}, t)$.

Let the 2-D projections of line segments between t_- and t_+ on the plane at time t be referred to as the unknown (true) displacement field $\boldsymbol{d}$ associated with the underlying image u. It is not feasible to compute displacement vectors on a continuum of spatial positions; hence, $\boldsymbol{d}$ is assumed to be an array of vectors defined over the lattice $\Lambda_{\boldsymbol{d}}$ in R^3. In the literature, the cases where $\Lambda_{\boldsymbol{d}}$ is a sublattice of Λ_g : $\Lambda_{\boldsymbol{d}} \subset \Lambda_g$, or $\Lambda_{\boldsymbol{d}}$ is identical to Λ_g : $\Lambda_{\boldsymbol{d}} = \Lambda_g$, have been most frequently considered. In this paper, a more general situation, where $\Lambda_{\boldsymbol{d}}$ and Λ_g are arbitrary, will be investigated. Consequently, a displacement vector may be defined at a spatio-temporal position that does not belong to Λ_g. For $\Lambda_{\boldsymbol{d}} = \Lambda_g$ (e.g., for motion-compensated prediction), the linear trajectories introduce no errors because only the endpoints are of interest. In the case where $\Lambda_{\boldsymbol{d}} \neq \Lambda_g$ (e.g., for motion-compensated interpolation), however, there are three points of interest, and consequently, there will be an error introduced due to a departure of the motion trajectory from linearity in the interval (t_-, t_+). If this interval is small, such an error is expected to be minor.

The investigations that follow are valid for any lattices Λ_g and $\Lambda_{\boldsymbol{d}}$, but for simplicity, it is assumed that they are rectangular lattices with horizontal, vertical, and temporal sampling periods (T_g^h, T_g^v, T_g) and $(T_{\boldsymbol{d}}^h, T_{\boldsymbol{d}}^v, T_{\boldsymbol{d}})$, respectively. Consequently, consecutive image fields are spaced by T_g seconds, whereas motion fields are spaced by $T_{\boldsymbol{d}}$ seconds. Each field of the image sequence contains M_g picture elements (pixels), and each motion field consists of $M_{\boldsymbol{d}}$ vectors. Let the horizontal and vertical dimensions of motion fields be $M_{\boldsymbol{d}}^h$ and $M_{\boldsymbol{d}}^v$, respectively $(M_{\boldsymbol{d}} = M_{\boldsymbol{d}}^h \times M_{\boldsymbol{d}}^v)$. With the above assumptions, a site $\boldsymbol{s}_i = [kT_g^h, lT_g^v, mT_g] \in \Lambda_g$ (for some integer i, k, l, m) is an image pixel in the mth field with spatial coordinate (k, l).

The true displacement field $\tilde{\boldsymbol{d}}$ is assumed to be a sample (realization) from random field (RF) $\boldsymbol{D}$. Let $\hat{\boldsymbol{d}}$ be an estimate of $\tilde{\boldsymbol{d}}$, and let $\boldsymbol{d}$ denote any sample field from $\boldsymbol{D}$. Assuming a linear motion trajectory between two images, as discussed above, the definition of the displacement field is given below and is illustrated in Fig. 1.

Definition: The displacement field $\tilde{\boldsymbol{d}}$ defined over $\Lambda_{\boldsymbol{d}}$ is a set of 2-D vectors such that for all $(\boldsymbol{x}_i, t) \in \Lambda_{\boldsymbol{d}}$, the *preceding* image point $(\boldsymbol{x}_i - \Delta t \cdot \tilde{\boldsymbol{d}}(\boldsymbol{x}_i, t), t_-)$ has moved to the *following* point $(\boldsymbol{x}_i + (1.0 - \Delta t) \cdot \tilde{\boldsymbol{d}}(\boldsymbol{x}_i, t), t_+)$, where $t_- = t - \Delta t \cdot T_g$, $t_+ = t + (1.0 - \Delta t) \cdot T_g$, and $\Delta t = t/T_g - \lfloor t/T_g \rfloor$.

The displacement field $\tilde{\boldsymbol{d}}$ may not be defined at *all* points in $\Lambda_{\boldsymbol{d}}$, for example, due to occlusions and may not be unique, as in the case of a rotating uniform disk. Note that this definition places a displacement vector at any spatio-temporal position. The limiting cases when Δt=0.0 and Δt=1.0 correspond to forward and backward predictive-type estimation, respectively.

Since we impose no constraints on images, it can be

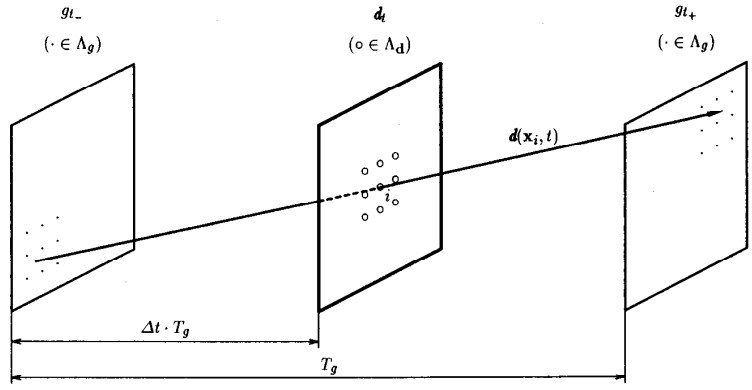


Fig. 1. Definition of the displacement field $\boldsymbol{d}_t$ for motion estimation from two image fields (only one vector is displayed). Sites of lattice $\Lambda_{\boldsymbol{d}}$ (o) are pivoting points for the displacement vectors, and thus, every vector $\boldsymbol{d}(\boldsymbol{x}_i, t)$ crosses site $(\boldsymbol{x}_i, t)$, whereas its ends are "free" to move and do not necessarily belong to lattice Λ_g.

expected that motion fields associated with such images will contain discontinuities at the boundaries of objects with different motion. To model sudden changes in motion vector length and/or orientation, we use the concept of motion discontinuity. The true motion discontinuities are curves in the image plane defined over continuous spatio-temporal coordinates $(\boldsymbol{x}, t)$ and are unobservable like the true motion fields. They are just a means of describing motion in an image and can be understood as indicator functions (e.g., binary) for each $(\boldsymbol{x}, t)$. Let the true (unknown) field of such discontinuities be denoted by $\tilde{\boldsymbol{l}}$. Since the computation of $\tilde{\boldsymbol{l}}$ over a continuum of spatial positions is not feasible, $\tilde{\boldsymbol{l}}$ will be represented by a discrete discontinuity field over a union of cosets (shifted lattices) [8] $\Psi_l = \psi_h \cup \psi_v$, where ψ_h and ψ_v are orthogonal cosets defined as follows (superscript T denotes a transposition):

$$\psi_h = \Lambda_{\boldsymbol{d}} + [0, T^v_{\boldsymbol{d}}/2, 0]^T, \quad \psi_v = \Lambda_{\boldsymbol{d}} + [T^h_{\boldsymbol{d}}/2, 0, 0]^T.$$

Cosets ψ_h and ψ_v identify positions at which $M_l = M^h_{\boldsymbol{d}} \times (M^v_{\boldsymbol{d}} - 1) + (M^h_{\boldsymbol{d}} - 1) \times M^v_{\boldsymbol{d}}$ motion discontinuities or *line elements* are estimated. The line elements are located midway between sites of $\Lambda_{\boldsymbol{d}}$ and take on the value 1 if a motion boundary intersects the line joining the neighboring displacement vector sites and 0 otherwise.

We assume that $\tilde{\boldsymbol{l}}$ is a sample field from RF L. Let $\hat{l}$ be an estimate of $\tilde{\boldsymbol{l}}$. In addition, let l be any sample field drawn from L. The RF L will be called a *line process*, and its sample l will be called a *line field*.

Let the subscript t denote restriction of a random field or of its realization to time t. Thus, $\boldsymbol{d}_t$ will stand for a realization of random field $\boldsymbol{D}_t$ ($\boldsymbol{D}$ at time t).

It is assumed that the random field G_t is defined over the discrete state space $\mathcal{S}_g = (\mathcal{S}'_g)^{M_g}$, where $\mathcal{S}'_g$ is the single pixel state space corresponding to quantized image intensities, and $(\cdot)^M$ denotes an M-fold Cartesian product. Similarly, the random field $\boldsymbol{D}_t$ is defined over the state space $\mathcal{S}_{\boldsymbol{d}} = (\mathcal{S}'_{\boldsymbol{d}})^{M_{\boldsymbol{d}}}$, where $\mathcal{S}'_{\boldsymbol{d}}$ is the single vector state space. Two cases are considered:

1. $\mathcal{S}'_{\boldsymbol{d}}$ is a discrete state space, i.e., a square 2-D grid over the range $[-d_{max}, d_{max}]$ with N_d possible levels in each direction.
2. $\mathcal{S}'_{\boldsymbol{d}} = R^2$ is a continuous state space.

It is also assumed that the random field L is defined over the discrete state space $\mathcal{S}_l = (\mathcal{S}'_l)^{M_l}$, where $\mathcal{S}'_l$ is the single line element state space. For the purpose of this work, it is assumed that RF L is binary and that $\mathcal{S}'_l$ consists of two states: 0 (no motion discontinuity) and 1 (motion discontinuity "on"). Possible extensions of this state space to nonbinary spaces (e.g., incorporating the directionality of line elements) are not considered in this paper but can be found in [10] in the context of image modeling.

B. Estimation Criteria

The objective is to jointly estimate the pair $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{l}})$ of true displacement and line fields at time t corresponding to an underlying image u on the basis of observations g.

1) MAP Estimation: The "best" or "most likely" displacement field estimate $\widehat{\boldsymbol{d}}^*_t \in \mathcal{S}_{\boldsymbol{d}}$ and line field estimate $\widehat{l}^*_t \in \mathcal{S}_l$ given the observations g_{t_-}, g_{t_+} must satisfy the relationship:

$$P(\boldsymbol{D}_t = \widehat{\boldsymbol{d}}^*_t, L_t = \widehat{l}^*_t | g_{t_-}, g_{t_+}) \geq P(\boldsymbol{D}_t = \widehat{\boldsymbol{d}}_t, L_t = \widehat{l}_t | g_{t_-}, g_{t_+}) \quad \forall \widehat{\boldsymbol{d}}_t \in \mathcal{S}_{\boldsymbol{d}}, \widehat{l}_t \in \mathcal{S}_l$$

where P is the conditional discrete probability distribution of the motion and line fields given the observations. Applying the Bayes rule for discrete random variables, the above posterior distribution can be factored as follows:

$$P(\boldsymbol{D}_t = \boldsymbol{d}_t, L_t = l_t | g_{t_-}, g_{t_+}) = \frac{P(G_{t_+} = g_{t_+} | \boldsymbol{d}_t, l_t, g_{t_-}) P(\boldsymbol{D}_t = \boldsymbol{d}_t, L_t = l_t | g_{t_-})}{P(G_{t_+} = g_{t_+} | g_{t_-})}. \quad (1)$$

Note that since $P(G_{t_+} = g_{t_+} | g_{t_-})$ is not a function of $(\boldsymbol{D}_t, L_t)$, it can be ignored when maximizing $P(\boldsymbol{D}_t = \widehat{\boldsymbol{d}}_t, L_t = \widehat{l}_t | g_{t_-}, g_{t_+})$ with respect to $(\widehat{\boldsymbol{d}}_t, \widehat{l}_t)$. Thus, the MAP estimate of the pair $(\tilde{\boldsymbol{d}}_t, \tilde{\boldsymbol{l}}_t)$ is the solution to the following optimization

problem:

$$\max_{(\widehat{\boldsymbol{d}}_t,\widehat{l}_t)}\Big[P(G_{t_+}=g_{t_+}|\widehat{\boldsymbol{d}}_t,\widehat{l}_t,g_{t_-})P(\boldsymbol{D}_t=\widehat{\boldsymbol{d}}_t,L_t=\widehat{l}_t|g_{t_-})\Big].$$

If displacement vectors are defined over the continuous state space $\mathcal{S}_{\boldsymbol{d}}=R^2$, then by using the Bayes rule for mixed random variables, the discrete probability distribution $P(\boldsymbol{D}_t=\widehat{\boldsymbol{d}}_t,L_t=\widehat{l}_t\mid g_{t_-})$ is replaced by the mixed continuous/discrete probability distribution $p(\widehat{\boldsymbol{d}}_t,L_t=\widehat{l}_t|g_{t_-})$.

2) Minimum Expected Cost (MEC) Estimation: Another approach to the estimation of motion is based on Bayesian criteria developed by Marroquin for reconstruction of images [27]. The goal is to minimize the ensemble average with respect to $\boldsymbol{D}_t,G_{t_-},G_{t_+}$ of some positive definite cost functional measuring the error between the true and estimated motion fields (*minimum expected cost* (MEC) estimation). Here, we will only discuss the MEC estimation for the globally smooth motion model, i.e., without discontinuity process L_t. Let the cost functional have the following form:

$$\Theta(\boldsymbol{d}_t',\boldsymbol{d}_t'')=\sum_{i=1}^{M_{\boldsymbol{d}}}\theta(\boldsymbol{d}'(\boldsymbol{x}_i,t),\boldsymbol{d}''(\boldsymbol{x}_i,t))$$

where θ is a positive definite function, and $\boldsymbol{d}'$, $\boldsymbol{d}''$ are two motion vector fields. The cost associated with approximating the true motion field $\tilde{\boldsymbol{d}}_t$ by estimate $\widehat{\boldsymbol{d}}_t$ is $\Theta(\tilde{\boldsymbol{d}}_t,\widehat{\boldsymbol{d}}_t)$. The optimal Bayesian estimate $\widehat{\boldsymbol{d}}_t^*\in\mathcal{S}_{\boldsymbol{d}}$ is defined as follows:

$$E_{\boldsymbol{D}_t,G_{t_-},G_{t_+}}[\Theta(\boldsymbol{d}_t,\widehat{\boldsymbol{d}}_t^*)]=\min_{\widehat{\boldsymbol{d}}_t}E_{\boldsymbol{D}_t,G_{t_-},G_{t_+}}[\Theta(\boldsymbol{d}_t,\widehat{\boldsymbol{d}}_t)]\quad(2)$$

where $E_{\boldsymbol{D}_t,G_{t_-},G_{t_+}}[\cdot]$ stands for the ensemble expectation over all configurations of $\boldsymbol{D}_t,G_{t_-}$ and G_{t_+}. Note that the estimates $\widehat{\boldsymbol{d}}_t$ and $\widehat{\boldsymbol{d}}_t^*$ are functions of g_{t_-} and g_{t_+} and can be thought of as mappings from R^{2M_g} to $R^{2M_{\boldsymbol{d}}}$. Expressing the expectation as a sum and using the positive definiteness of θ, it can be shown [24] that (2) is equivalent to

$$\min_{\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t)}\sum_{\boldsymbol{r}\in\mathcal{S}_{\boldsymbol{d}}'}\theta(\boldsymbol{r},\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t))\cdot\Big[\sum_{\boldsymbol{d}_t:\boldsymbol{d}(\boldsymbol{x}_i,t)=\boldsymbol{r}}P(\boldsymbol{D}_t=\boldsymbol{d}_t|g_{t_-},g_{t_+})\Big]\quad\forall i.\quad(3)$$

This means that the minimization with respect to $\widehat{\boldsymbol{d}}_t$ can be achieved by minimizing the individual marginal expected costs at each position $(\boldsymbol{x}_i,t)$.

Further simplification of (3) requires explicit knowledge of the function θ. This function must reflect "goodness" of an estimate, i.e., a worse estimate should increase the value of θ, and a better one should reduce it. We use the following θ:

$$\theta(\tilde{\boldsymbol{d}}(\boldsymbol{x}_i,t),\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t))=\|\tilde{\boldsymbol{d}}(\boldsymbol{x}_i,t)-\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t)\|^2\quad(4)$$

where $\|\cdot\|$ is the L^2 norm for mathematical tractability. The solution to minimization (3) with θ defined in (4) is the minimum mean squared error (MMSE) estimate expressed through the marginal conditional expectation of $\boldsymbol{D}(\boldsymbol{x}_i,t)$:

$$\widehat{\boldsymbol{d}}^*(\boldsymbol{x}_i,t)=\sum_{\boldsymbol{r}\in\mathcal{S}_{\boldsymbol{d}}'}\boldsymbol{r}\Big[\sum_{\boldsymbol{d}_t:\boldsymbol{d}(\boldsymbol{x}_i,t)=\boldsymbol{r}}P(\boldsymbol{D}_t=\boldsymbol{d}_t|g_{t_-},g_{t_+})\Big]\quad\forall i.\quad(5)$$

The continuous state space case is identical except for integrals replacing the summations and probability density replacing the discrete probability distribution. To compute the conditional expectation from (5), the *a posteriori* discrete probability distribution $P(\boldsymbol{D}_t=\boldsymbol{d}_t\mid g_{t_-},g_{t_+})$ or density $p(\boldsymbol{d}_t\mid g_{t_-},g_{t_+})$ is needed.

C. Models

In order to solve the MAP and MEC estimation problems, the posterior distribution from (1) must be known explicitly. In this section, we determine the constituent probabilities from (1) by formulating appropriate models. First, from a structural model relating motion to the underlying image and from a model for the observation process, we propose the likelihood $P(G_{t_+}=g_{t_+}\mid\boldsymbol{d}_t,l_t,g_{t_-})$. Then, we directly postulate a displacement field model for $P(\boldsymbol{D}_t=\boldsymbol{d}_t,L_t=l_t|g_{t_-})$.

1) Structural Model: To facilitate inference of motion from images, it is fundamental to specify a *structural* model relating motion vectors and image intensity values. It is usually assumed that image intensity or its spatial gradient is constant along motion trajectories. Applying the intensity constancy assumption to the true underlying image u along the true motion trajectories $\tilde{\boldsymbol{d}}$ in time interval $[t_-,t_+]$ results in the following relationship:

$$u(\boldsymbol{x}-\Delta t\cdot\tilde{\boldsymbol{d}}(\boldsymbol{x},t),t_-)=u(\boldsymbol{x}+(1.0-\Delta t)\cdot\tilde{\boldsymbol{d}}(\boldsymbol{x},t),t_+).\quad(6)$$

A more complex model incorporating linear variation of intensity has been devised in [34] and [11]; however, it will not be considered here.

2) Observation Model: The true underlying image u results from the "ideal" projection of a scene onto an image plane. In reality, images are acquired with a video camera and discretized using electronic circuitry. Hence, the observed image g is a transformed copy of u after such operations as filtering, sampling, γ correction, and quantization, and it also incorporates image sensor noise, quantization noise, distortion due to aliasing, etc.

Extrapolating the relationship (6) to the observed image g, we model the *displaced pixel differences* (DPD's)

$$\tilde{r}(\tilde{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t)=\tilde{g}(\boldsymbol{x}_i+(1.0-\Delta t)\cdot\tilde{\boldsymbol{d}}(\boldsymbol{x}_i,t),t_+)-\tilde{g}(\boldsymbol{x}_i-\Delta t\cdot\tilde{\boldsymbol{d}}(\boldsymbol{x}_i,t),t_-)$$

for $(\boldsymbol{x}_i,t)\in\Lambda_{\boldsymbol{d}}$ by independent Gaussian random variables. $\tilde{g}(\boldsymbol{x},t)$ above denotes an intensity value at $(\boldsymbol{x},t)\notin\Lambda_g$ obtained by interpolation. We have investigated various interpolation schemes [23] and have concluded that although matching algorithms are relatively insensitive to the type of interpolator, spatio-temporal gradient methods require interpolators that provide continuity of both the intensity and its first derivative. We thus use a third-order (bicubic) separable interpolator, as proposed by Keys [19], which has this property.

Given these assumptions, we can write[1]

$$P(\tilde{G}_{t_+} = \tilde{g}_{t_+} \mid \boldsymbol{d}_t, l_t, g_{t_-}) = (2\pi\sigma^2)^{-M_{\boldsymbol{d}}/2} \cdot e^{-U_g(\tilde{g}_{t_+} \mid \boldsymbol{d}_t, g_{t_-})/2\sigma^2} \quad (7)$$

where energy U_g is defined as

$$U_g(\tilde{g}_{t_+} \mid \boldsymbol{d}_t, g_{t_-}) \approx \sum_{i=1}^{M_{\boldsymbol{d}}} [\tilde{r}(\boldsymbol{d}(\boldsymbol{x}_i, t), \boldsymbol{x}_i, t)]^2 \quad (8)$$

and where $\tilde{G}_{t_+}$ and $\tilde{g}_{t_+}$ denote the random and sample image fields at time t_+ spatially interpolated at positions pointed to by displacement vectors. We need, however, $U_g(g_{t_+} \mid \boldsymbol{d}_t, g_{t_-})$, i.e., energy at locations of the image sampling lattice Λ_g at time t_+. We could account for this discrepancy by interpolating the motion field $\boldsymbol{d}_t$ to get a field of motion vectors terminating on the lattice points at t_+; however, this will not significantly change the energy U_g, and thus, we simply assume that $U_g(g_{t_+} \mid \boldsymbol{d}_t, g_{t_-}) = U_g(\tilde{g}_{t_+} \mid \boldsymbol{d}_t, g_{t_-})$.

3) Displacement Field Model: It can be observed that in most scenes, motion is the result of a change of position of near rigid bodies. After projection onto the image plane, that 3-D motion becomes a 2-D motion of 2-D objects. The motion field in such an image consists of patches of similar (orientation and length) vectors with possible discontinuities at motion boundaries. Therefore, it will be assumed here that motion fields are smooth functions of spatial position $\boldsymbol{x}$ (fixed t) except for occasional abrupt changes in vector length and/or orientation. In fact, displacement fields are significantly smoother than the images themselves. Based on this observation and on successful application of MRF's to image modeling [13], [10], here, we propose to jointly model the motion and motion discontinuity fields by a pair $(\boldsymbol{D}_t, L_t)$ of coupled vector and binary MRF's.

The traditional approach to characterization of MRF's via the finite-dimensional joint distributions is cumbersome and does not provide a simple and clear relationship between the distribution parameters and the properties of a MRF. Due to the Hammersley-Clifford theorem [3], however, a MRF with respect to neighborhood system $\mathcal{N}$ is uniquely characterized by a *Gibbs* distribution with respect to $\mathcal{N}$. Since a vector MRF (VMRF) differs from a scalar MRF only by the definition of a state (in R^2 instead of R), the properties of scalar MRF's hold for VMRF's as well.

Recall that the properties of a motion model are described by the discrete probability distribution $P(\boldsymbol{D}_t = \boldsymbol{d}_t, L_t = l_t \mid g_{t_-})$ from (1), which can be factored using the Bayes rule:

$$P(\boldsymbol{D}_t = \boldsymbol{d}_t, L_t = l_t \mid g_{t_-}) = P(\boldsymbol{D}_t = \boldsymbol{d}_t \mid l_t, g_{t_-}) \cdot P(L_t = l_t \mid g_{t_-}).$$

If $P(\boldsymbol{D}_t = \boldsymbol{d}_t \mid l_t, g_{t_-})$ and $P(L_t = l_t | g_{t_-})$ are Gibbsian, then $P(\boldsymbol{D}_t = \boldsymbol{d}_t, L_t = l_t \mid g_{t_-})$ is also Gibbsian, and the pair $(\boldsymbol{d}_t, l_t)$ has Markovian properties. Since a single image field is expected to contribute little information to the motion vector model, the conditioning on g_{t_-} in $P(\boldsymbol{D}_t = \boldsymbol{d}_t \mid l_t, g_{t_-})$ is omitted as an approximation. However, motion discontinuities will most likely occur at the positions of intensity discontinuities at object boundaries, which is expressed through conditioning on g_{t_-} in $P(L_t = l_t \mid g_{t_-})$.

Under assumed Markov properties, the probability governing the displacement RF $\boldsymbol{D}_t$ can be expressed by the Gibbs distribution

$$P(\boldsymbol{D}_t = \boldsymbol{d}_t \mid l_t) = \frac{1}{Z_{\boldsymbol{d}}} e^{-U_{\boldsymbol{d}}(\boldsymbol{d}_t \mid l_t)/\beta_{\boldsymbol{d}}} \quad (9)$$

where $Z_{\boldsymbol{d}}$ is a normalizing constant (*partition* function), $\beta_{\boldsymbol{d}}$ is a constant controlling characteristic properties of $\boldsymbol{D}_t$, and the (conditional) *energy* function $U_{\boldsymbol{d}}(\boldsymbol{d}_t \mid l_t)$ is defined as

$$U_{\boldsymbol{d}}(\boldsymbol{d}_t \mid l_t) = \sum_{c_{\boldsymbol{d}} = \{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in C_{\boldsymbol{d}}} V_{\boldsymbol{d}}(\boldsymbol{d}_t, c_{\boldsymbol{d}})[1 - l(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, t)]. \quad (10)$$

$c_{\boldsymbol{d}}$ is a clique of vectors, whereas $C_{\boldsymbol{d}}$ is a set of all such cliques derived from a neighborhood system $\mathcal{N}_{\boldsymbol{d}}$ defined over lattice $\Lambda_{\boldsymbol{d}}$. $(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, t) \in \Psi_l$ denotes a site of line element located between vector sites $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $V_{\boldsymbol{d}}$ is a *potential* function crucial to characterization of the properties of displacement process $\boldsymbol{D}_t$. This is a straightforward extension of the scalar MRF model for images as proposed by Geman and Geman [10].

The energy function (10) can be understood as follows. There exists a cost $V_{\boldsymbol{d}}(\boldsymbol{d}_t, c_{\boldsymbol{d}})$ associated with each vector clique $c_{\boldsymbol{d}}$ that increases if a motion field sample locally departs from the assumed *a priori* model characterized by $\beta_{\boldsymbol{d}}$ and $V_{\boldsymbol{d}}$. If, however, the line element separating the displacement vectors from clique $c_{\boldsymbol{d}}$ is "turned on" $(l(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, t) = 1)$, there is no cost associated with the clique $c_{\boldsymbol{d}}$. In this way, there is no penalty for introducing an abrupt change in length or orientation of a displacement vector. The ability to zero the cost associated with vector cliques by inserting a line element must be penalized, however. Otherwise, a line field with all elements "on" would give the zero displacement energy (10). This penalty is provided by the line field model described in the next section.

To specify the *a priori* displacement model, the potential function $V_{\boldsymbol{d}}$, the neighborhood system $\mathcal{N}_{\boldsymbol{d}}$, and the cliques $c_{\boldsymbol{d}}$ have to be specified. Bearing in mind the assumed smoothness of $\boldsymbol{d}_t$, we define the potential function $V_{\boldsymbol{d}}$ as follows:

$$V_{\boldsymbol{d}}(\boldsymbol{d}_t, c_{\boldsymbol{d}}) = V(\boldsymbol{d}(\boldsymbol{x}_i, t), \boldsymbol{d}(\boldsymbol{x}_j, t)) = \|\boldsymbol{d}(\boldsymbol{x}_i, t) - \boldsymbol{d}(\boldsymbol{x}_j, t)\|^2, \quad c_{\boldsymbol{d}} = \{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in C_{\boldsymbol{d}} \quad (11)$$

where $\|\cdot\|$ is a norm in R^2, e.g., L^2. This particular potential well captures the smoothness of the displacement field process $\boldsymbol{D}_t$ [21]. We use the first-order neighborhood system $\mathcal{N}^1_{\boldsymbol{d}}$ depicted in Fig. 2(a), which consists of two-element horizontal and vertical vector cliques (Fig. 2(b) and (c)). Note that every displacement vector has four vector neighbors and four line neighbors.

To demonstrate that this VMRF is a valid model for displacement fields, we generated unconstrained (by image intensities) VMRF samples disregarding the line process. Fig. 3 shows such samples for the first-order neighborhood system

[1] Note that $\boldsymbol{d}_t$ constitutes a complete description of motion, and line field l_t is only an aid in estimation of $\boldsymbol{d}_t$. Hence, the conditioning on l_t can be dropped, thus giving $U(\tilde{g}_{t_+} | \boldsymbol{d}_t, g_{t_-})$.

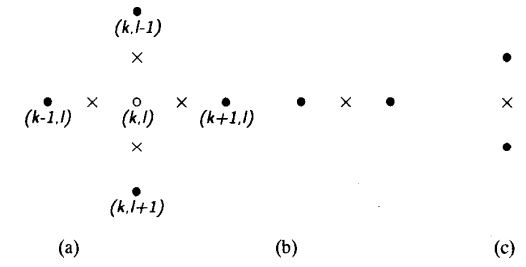


Fig. 2. (a) First-order neighborhood system $\mathcal{N}_{\boldsymbol{d}}^1$ for vector field $\boldsymbol{d}_t$ defined over $\Lambda_{\boldsymbol{d}}$ with discontinuities l_t defined over Ψ_l; (b) horizontal cliques; (c) vertical cliques (o–center vector site, •–vector site, ×–line site).

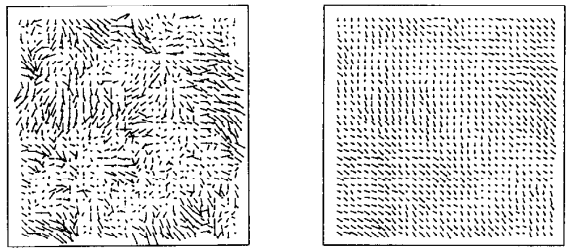


Fig. 3. VMRF samples for potential function (11) and neighborhood system $\mathcal{N}_{\boldsymbol{d}}^1$ initialized by a random vector field with mean (0.5, 0.5): (a) $\beta_{\boldsymbol{d}}$=1.0; (b) $\beta_{\boldsymbol{d}}$=0.1.

and two different values of $\beta_{\boldsymbol{d}}$ after 50 iterations of the *Gibbs sampler* (Section III) starting from random initial configuration with mean (0.5, 0.5). Note that for smaller value of $\beta_{\boldsymbol{d}}$, the sample field is less "chaotic." The value of $\beta_{\boldsymbol{d}}$ can be viewed as an "activity" or "ordering" measure. Thus, to model a slowly varying motion $\beta_{\boldsymbol{d}}$ of the order of 0.1 seems to be appropriate; however, an exact value cannot be established.

4) Line Field Model: The line field model is based on a binary MRF (BMRF) L_t and is described by the Gibbs probability distribution

$$P(L_t = l_t \mid g_{t-}) = \frac{1}{Z_l} e^{-U_l(l_t|g_{t-})/\beta_l}, \tag{12}$$

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

$$U_l(l_t \mid g_{t-}) = \sum_{c_l \in \mathcal{C}_l} V_l(l_t, g_{t-}, c_l) \tag{13}$$

where c_l is a line clique, and $\mathcal{C}_l$ is a set of all line cliques derived from a neighborhood system $\mathcal{N}_l$ defined over Ψ_l. The line potential function V_l provides a penalty associated with the introduction of a line element.

The second-order neighborhood system $\mathcal{N}_l^2$ for the "dual" sampling structure Ψ_l is shown in Fig. 4. Note that since the union of cosets $\Psi_l = \psi_h \cup \psi_v$ identifies positions of horizontal and vertical line elements, two neighborhood systems are defined (Figs. 4(a) and (b)). Every line element has eight line neighbors and two vector neighbors. There are two types of four-element line cliques. The cross-shaped cliques from Fig. 4(c) are the same as those used in [10] and aim at modeling the shape of motion boundaries (small penalty for straight lines and high penalty for intersections; see Fig. 5(a)). The

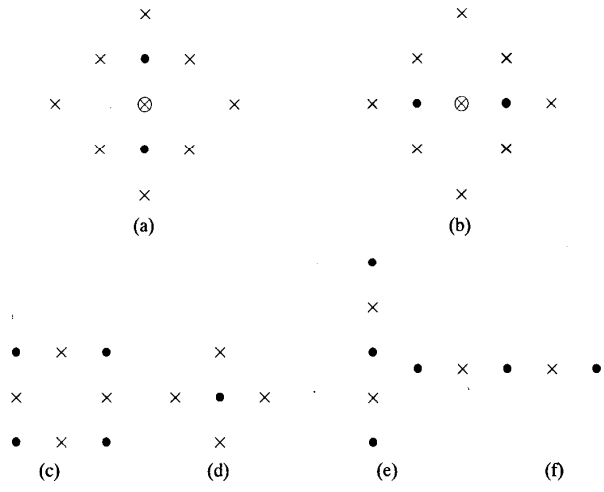


Fig. 4. Second-order neighborhood system $\mathcal{N}_l^2$ for line field l_t defined over Ψ_l: (a) Horizontal line element; (b) vertical line element; (c), (d) four-element cliques; (e), (f) two-element cliques (○–center line site, ×–line site, •–vector site).

square-shaped cliques from Fig. 4(d) are employed in order to exclude isolated vectors ($V_{l_4} = \infty$; see Fig. 5(b)) when all four line elements are "on." The two-element vertical cliques of horizontal line elements (Fig. 4(e)) and the horizontal cliques of vertical line elements (Fig. 4(f)) are used following Marroquin [27] to prevent formation of double edges (Fig. 5(c)). Fig. 5 shows possible configurations (up to a rotation) as well as related costs, which were chosen experimentally. Only one configuration of the square-shaped four-element clique (Fig. 5(b)) is shown since other configurations are accounted for in the cross-shaped and two-element cliques.

Note that the *a priori* probability of the line process (12) is conditioned on the observations. It means that image information g_{t-} should be considered when computing the line samples l_t. In general, a 3-D scene giving rise to a motion discontinuity will also contribute to an intensity edge. Only under specific circumstances will a motion discontinuity not correspond to an edge of intensity. Hence, similarly to Hutchinson *et al.* [18], we assume that an introduction of a line element should coincide with an intensity edge. We use the following potential function for one-element cliques:

$$V_{l_1}(l_t, g_{t-}, c_l) = \begin{cases} \frac{\alpha}{(\nabla_v g_{t-})^2} l_h(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, t) & \text{for horizontal } c_l = \{\boldsymbol{x}_i, \boldsymbol{x}_j\} \\ \frac{\alpha}{(\nabla_h g_{t-})^2} l_v(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, t) & \text{for vertical } c_l = \{\boldsymbol{x}_i, \boldsymbol{x}_j\} \end{cases}$$

where l_h, l_v are horizontal and vertical line elements, ∇_h, ∇_v are horizontal and vertical components of the spatial gradient at position $(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, t)$, and α is a constant. The above potential introduces a penalty only if a line element is "on," and the appropriate gradient is relatively small. The total potential function for the line field can be expressed as

$$V_l(l_t, g_{t-}, c_l) = V_{l_4}(l_t, c_l) + V_{l_2}(l_t, c_l) + V_{l_1}(l_t, g_{t-}, c_l) \tag{14}$$

where V_{l_4} and V_{l_2} are tabulated in Fig. 5, and V_{l_1} is given above.

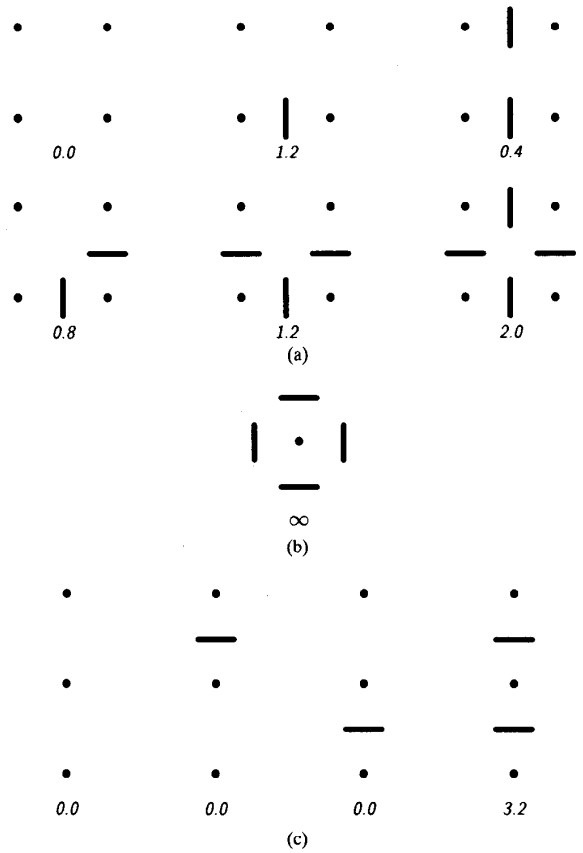


Fig. 5. Costs V_{l_4}, V_{l_2} associated with various configurations (up to a rotation): (a) Four-element cross-shaped cliques; (b) four-element square-shaped clique (other configurations are accounted for in (a) and (c)); (c) two-element cliques (•–vector site, — –line element "turned on").

To demonstrate that VMRF/BMRF is a valid model for discontinuous displacement fields, we generated unconstrained (by image intensities) samples from this model. Fig. 6 shows such samples for $\beta_{\boldsymbol{d}}$=0.1 and two different values of β_l after 50 iterations of the *Gibbs sampler* starting from a random line field with mean 0.0 (other parameters are the same as for Fig. 3). Note that, again, for smaller value of β_l, the sample field is more organized.

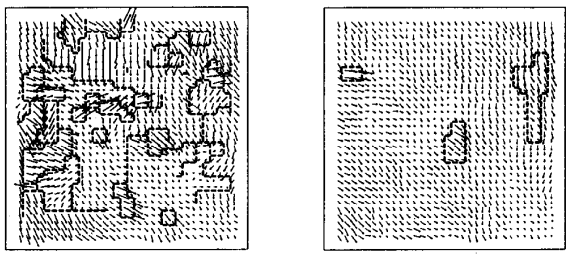
D. A Posteriori Probability

Combining the conditional likelihood (7), the displacement *a priori* probability (9), and the line *a priori* probability (12) via (1) and substituting $(\widehat{\boldsymbol{d}}_t, \widehat{l}_t)$ for $(\boldsymbol{d}_t, l_t)$, the following Gibbs form of the *a posteriori* probability can be obtained:

$$P(\boldsymbol{D}_t = \widehat{\boldsymbol{d}}_t, L_t = \widehat{l}_t \mid g_{t_-}, g_{t_+}) = \frac{1}{Z} e^{-U(\widehat{\boldsymbol{d}}_t, \widehat{l}_t, g_{t_-}, g_{t_+})} \quad (15)$$

where Z is a new normalizing constant (incorporating the probability $P(G_{t_+} = g_{t_+} | g_{t_-})$ from (1), $(2\pi\sigma^2)^{-M_{\boldsymbol{d}}/2}$ from (7), Z_d from (9), and Z_l from (12)), whereas the new energy function $U(\widehat{\boldsymbol{d}}_t, \widehat{l}_t, g_{t_-}, g_{t_+})$ is defined as follows:

$$U(\widehat{\boldsymbol{d}}_t, \widehat{l}_t, g_{t_-}, g_{t_+}) = \lambda_g U_g(\widetilde{g}_{t_+} \mid \widehat{\boldsymbol{d}}_t, g_{t_-}) + \lambda_{\boldsymbol{d}} U_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}_t \mid \widehat{l}_t) + \lambda_l U_l(\widehat{l}_t \mid g_{t_-}). \quad (16)$$



(a) (b)

Fig. 6. VMRF/BMRF samples for potential functions (11) and (14), neighborhood system $\mathcal{N}_{\boldsymbol{d}}^1 \cup \mathcal{N}_l^2$, and $\beta_{\boldsymbol{d}}$=0.1 initialized by a random vector field with mean (0.5, 0.5) and by a random line field with mean 0.0: (a) β_l=0.25; (b) β_l=0.05.

The conditional energies in the above relationship are defined in (8), (10), and (13), respectively, and $\lambda_g = 1/(2\sigma^2)$, $\lambda_{\boldsymbol{d}} = 1/\beta_{\boldsymbol{d}}$, $\lambda_l = 1/\beta_l$. The neighborhood system for this new Gibbs distribution is a combination of $\mathcal{N}_{\boldsymbol{d}}$ and $\mathcal{N}_l$ since the DPD model is based on the independently distributed noise random variables. Should this model be correlated (e.g., filtered noise), the neighborhood system for $\boldsymbol{D}$ would need to be redefined appropriately [10].

Having shown that the posterior distribution (1) is Gibbsian, it follows that the MAP estimation can be achieved by means of the following minimization:

$$\min_{\{\widehat{\boldsymbol{d}}_t, \widehat{l}_t\}} \lambda_g U_g(\widetilde{g}_{t_+} \mid \widehat{\boldsymbol{d}}_t, g_{t_-}) + \lambda_{\boldsymbol{d}} U_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}_t \mid \widehat{l}_t) + \lambda_l U_l(\widehat{l}_t \mid g_{t_-}). \quad (17)$$

Note that the functional to be minimized consists of three terms: U_g is a sum of squared DPD's over the entire image and describes the ill-posed matching problem of the data (g_{t_-}, g_{t_+}) by the motion field $\widehat{\boldsymbol{d}}_t$, $U_{\boldsymbol{d}}$ is responsible for conforming to the properties of the *a priori* displacement model, and U_l allows for occasional discontinuities according to the *a priori* line model. The three-term formulation of the energy function (16) can be viewed as regularization of the original correspondence problem (DPD only). Then, $\lambda_{\boldsymbol{d}} U_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}_t \mid \widehat{l}_t) + \lambda_l U_l(\widehat{l}_t \mid g_{t_-})$ is a stabilizing functional, and $1/\lambda_g$ is a regularization parameter. Hence, the Bayesian formulation comprises, as a specific case, the regularization method, which has been frequently used in computer vision [2].

The objective function in (17) is similar to that used in [18], which is derived from the original formulation of Horn and Schunck [17] with the additional nonstochastic motion discontinuity model. We pursue the stochastic approach by using two coupled MRF's and a random displaced pixel difference instead of the motion constraint equation. In addition, the line cliques are different, and the penalty for introducing a line element is a continuous function of data rather than a binary one [18]. Most importantly, however, we will use a stochastic relaxation algorithm for minimization of (17).

In the formulation (7), the ratios $\lambda_{\boldsymbol{d}}/\lambda_g$ and $\lambda_{\boldsymbol{d}}/\lambda_l$ play an important role weighting the confidence in the data and in the *a priori* model. A modification of any λ has an effect on the estimate; however, the magnitude of this effect is highly dependent on the data itself. Recall that σ^2 is a variance of the

displaced pixel difference (Gaussian) model; its value can be estimated given a displacement field. However, the parameters $\beta_{\boldsymbol{d}}$ and β_l, which characterize the *a priori* motion model, are far more difficult to compute. When MRF's are used in estimation of such observables as images or textures, β 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, and thus, it is usually chosen *ad hoc*. Consequently, there is no point in estimating σ^2; the ratios $\lambda_{\boldsymbol{d}}/\lambda_g$ and $\lambda_{\boldsymbol{d}}/\lambda_l$ may be chosen instead.

III. Stochastic Solution to MAP and MEC Estimation

The MAP estimation expressed through optimization (17) is a very complex problem because of the number of unknowns involved (typically tens of thousands) and because of multimodality of the objective function, which depends on $\widehat{\boldsymbol{d}}_t$ through the observations g. The MEC estimation problem is also characterized by a large number of unknowns and by the necessity to compute the marginal conditional distributions in (5).

A. Solving the MAP Estimation: Simulated Annealing

To solve the optimization problem (17), we use the method of *simulated annealing* [20], [10], which is based on the analogy with the process of *annealing* of solids. In simulated annealing, the behavior of a solid is simulated by generating sample configurations from a Gibbs distribution with the energy function suitably crafted for the given optimization problem. A "temperature" parameter $\mathbf{T}$, which replaces the true temperature in chemical annealing, is introduced into the *a posteriori* probability (15) as follows:

$$P(\boldsymbol{D}_t = \widehat{\boldsymbol{d}}_t, L_t = \widehat{l}_t \mid g_{t_-}, g_{t_+}) = \frac{1}{Z} e^{-U(\widehat{\boldsymbol{d}}_t, \widehat{l}_t, g_{t_-}, g_{t_+})/\mathbf{T}}. \tag{18}$$

The sample configurations are produced using *stochastic relaxation* (e.g., the *Metropolis* algorithm or the *Gibbs sampler*), which will be described in Section III-C. Similarly to Geman and Geman, we incorporate the *Gibbs sampler* into the *inhomogeneous* simulated annealing algorithm[2] with the annealing schedule specified by the initial and final temperatures $\mathbf{T}_0$ and $\mathbf{T}_f$ and the temperature change rule $\mathbf{T}_k = \varphi(\mathbf{T}_0, k)$ at iteration k.[3] In [10], it was proved (Theorem B) that if a sufficiently high initial temperature $\mathbf{T}_0$ is used, if the temperature $\mathbf{T}_k$ attains 0 with at most logarithmic rate, and if every site is visited infinitely often, then with time $t \to \infty$, the system will converge to the global optimum for any starting configuration.

The theoretical value of $\mathbf{T}_0$ is usually impractically high, and therefore, it is typically established through experimentation. Another practical problem is posed by the logarithmic annealing schedule. In order to obtain a "very organized" solution, the final temperature $\mathbf{T}_f$ after N iterations must be quite small. Unfortunately, the required number of iterations to attain the final temperature $\mathbf{T}_f$ from $\mathbf{T}_0$ grows exponentially with $\mathbf{T}_0/\mathbf{T}_f$. If the initial temperature cannot be lowered more without affecting the quality of the solution, then only the annealing schedule can be modified. In the experiments described in Section V, we have used the exponential schedule $\varphi(\mathbf{T}_0, k) = \mathbf{T}_0 \cdot a^{(k-1)}$, where a is slightly less than 1.0. Such a schedule reaches a low temperature in fewer iterations than the logarithmic one with the same number of iterations [10] but has to be used with caution since a large temperature decrement between iterations may trap the chain in a local minimum.

[2]Simulated annealing is called *homogeneous* if the Markov chain it generates can be decomposed into a sequence of stationary chains, i.e., with constant temperatures. It is called *inhomogeneous* if such decomposition does not exist, i.e., transition probabilities are time dependent through the temperature.

[3]An iteration is understood here as a complete scan of the unknown field, i.e., $M_{\boldsymbol{d}}$ or $M_{\boldsymbol{d}} + M_l$ attempted modifications of individual unknowns.

B. Solving the MEC Estimation: LLN for Markov Chains

The properties of a Markov chain constructed by the *Metropolis* algorithm or the *Gibbs sampler* are such that it is a *regenerative process*, and the time instants of the returns to a given state constitute a *renewal process*. Hence, the *law of large numbers (LLN) for Markov chains* [32] can be applied, and the marginal conditional expectation (5) can be approximated as follows:

$$\widehat{\boldsymbol{d}}^*(\boldsymbol{x}_i, t) \approx \frac{1}{N} \sum_{k=0}^{N-1} \boldsymbol{d}^k(\boldsymbol{x}_i, t) \tag{19}$$

where $\boldsymbol{d}^k(\boldsymbol{x}_i, t)$ is a displacement vector estimate at iteration number k, and N is the total number of iterations.

The advantage of the above approach over simulated annealing is that it requires no annealing schedule since the generation of samples occurs at a constant temperature $\mathbf{T}$. This temperature controls the state-rejection rate of the generation algorithm. If the parameter $\mathbf{T}$ is higher, the lower the rejection rate and the more chaotic the generated samples. If the $\mathbf{T}$ is lower, the higher the rejection rate and the more orderly the structure of generated realizations. To choose optimal $\mathbf{T}$, Marroquin proposed a method that maximizes the likelihood of a solution with respect to $\mathbf{T}$ [27].

C. Stochastic Relaxation

To implement the MAP and MEC estimation algorithms discussed above, samples from MRF's $\boldsymbol{D}_t$ and L_t are needed. Such samples can be provided by *stochastic relaxation* such as the *Metropolis* algorithm [29] or the *Gibbs sampler* [10]. A very important feature of stochastic relaxation methods is that they produce states according to probabilities of their occurrence, i.e., the unlikely states are also generated (less frequently, however). When incorporated into simulated annealing, this property allows local minima to be escaped, unlike the case of descent methods. We will use the Gibbs sampler since, despite its high complexity per iteration, in our tests, it provided a lower overall computational burden than the Metropolis algorithm [24].

The Gibbs sampler for discrete displacement process $\boldsymbol{D}_t$ is a site-replacement procedure that generates a new vector at every position $(\boldsymbol{x}_i, t) \in \Lambda_{\boldsymbol{d}}$ according to the following marginal

conditional discrete probability distribution [25] derived from probability (18):

$$P(\boldsymbol{D}(\boldsymbol{x}_i,t)=\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t) \mid \widehat{\boldsymbol{d}}(\boldsymbol{x}_j,t), j\neq i, \widehat{l}_t, g_{t_-}, g_{t_+}) = \frac{e^{-U^i_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}_t,\widehat{l}_t,g_{t_-},g_{t_+})/\mathbf{T}}}{\sum_{\boldsymbol{z}\in\mathcal{S}'_{\boldsymbol{d}}} e^{-U^i_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}^z_t,\widehat{l}_t,g_{t_-},g_{t_+})/\mathbf{T}}} \tag{20}$$

with the local displacement energy function $U^i_{\boldsymbol{d}}$ defined as

$$U^i_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}^z_t,\widehat{l}_t,g_{t_-},g_{t_+}) = \lambda_g[\widetilde{r}(\boldsymbol{z},\boldsymbol{x}_i,t)]^2 + \lambda_{\boldsymbol{d}} \sum_{j:\boldsymbol{x}_j\in\eta_{\boldsymbol{d}}(\boldsymbol{x}_i)} \|\boldsymbol{z}-\widehat{\boldsymbol{d}}(\boldsymbol{x}_j,t)\|^2[1-\widehat{l}(\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle,t)] \tag{21}$$

where $\eta_{\boldsymbol{d}}(\boldsymbol{x}_i)$ is a spatial neighborhood of displacement vector at $\boldsymbol{x}_i$, and $\widehat{\boldsymbol{d}}^z_t$ denotes a displacement field identical to field $\widehat{\boldsymbol{d}}_t$, except for spatial location $\boldsymbol{x}_i$, where the vector is z. Similarly, it can be demonstrated [24] that the conditional probability driving the Gibbs sampler for displacement discontinuities can be expressed at spatio-temporal location $(\boldsymbol{y}_i,t)\in\Psi_l$ as follows:

$$P(L(\boldsymbol{y}_i,t)=\widehat{l}(\boldsymbol{y}_i,t) \mid \widehat{l}(\boldsymbol{y}_j,t), j\neq i, \widehat{\boldsymbol{d}}_t, g_{t_-}) = \frac{e^{-U^i_l(\widehat{l}_t,\widehat{\boldsymbol{d}}_t,g_{t_-})/\mathbf{T}}}{\sum_{z\in\mathcal{S}'_l} e^{-U^i_l(\widehat{l}^z_t,\widehat{\boldsymbol{d}}_t,g_{t_-})/\mathbf{T}}} \tag{22}$$

where the local line energy function U^i_l is defined as

$$U^i_l(\widehat{l}^z_t,\widehat{\boldsymbol{d}}_t,g_{t_-}) = \lambda_{\boldsymbol{d}} \sum_{\substack{c_{\boldsymbol{d}}=\{\boldsymbol{x}_m,\boldsymbol{x}_n\}:\\ \langle\boldsymbol{x}_m,\boldsymbol{x}_n\rangle=\boldsymbol{y}_i}} \|\widehat{\boldsymbol{d}}(\boldsymbol{x}_m,t)-\widehat{\boldsymbol{d}}(\boldsymbol{x}_n,t)\|^2[1-z] + \lambda_l \sum_{c_l:\boldsymbol{y}_i\in c_l} V_l(\widehat{l}^z_t,g_{t_-},c_l) \tag{23}$$

and $\widehat{l}^z_t$ denotes a line field that is identical to field $\widehat{l}_t$, except for spatial location $\boldsymbol{y}_i=<\boldsymbol{x}_m,\boldsymbol{x}_n>$, where the line element is z.

1) Discrete State Space Gibbs Sampler: The discrete state space Gibbs sampler scans the displacement and line fields and generates new states according to the Gibbs marginal probability distributions (20) and (22), respectively.

At each spatial location of random field $\boldsymbol{D}_t$, a complete bivariate discrete probability distribution is computed (probability for all possible displacement vector states). Then, the 2-D distribution is accumulated along one direction (e.g., horizontal) to obtain a 1-D marginal cumulative distribution, and a random state is generated from this univariate distribution. The vertical state is generated by sampling the univariate cumulative distribution obtained from the 2-D discrete probability distribution for the known horizontal state. The calculation of the complete probability distribution at each location results in a high computational load per iteration of the discrete state space Gibbs sampler. However, this is offset by faster convergence than the Metropolis algorithm, which samples the states from a uniform discrete probability distribution and only then rejects the unlikely ones.

Since the random field L_t is binary, only the probabilities for "on" and "off" states have to be computed. A new line state is generated based on those two probabilities.

2) Continuous State Space Gibbs Sampler for $\boldsymbol{D}_t$: Recall that in the case of the continuous state space ($\mathcal{S}_{\boldsymbol{d}}=R^2$), the marginal conditional discrete probability distribution defined in (20) becomes the probability density $p(\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t) \mid \widehat{\boldsymbol{d}}(\boldsymbol{x}_j,t), j\neq i, \widehat{l}_t, g_{t_-}, g_{t_+})$ with the same local energy $U^i_{\boldsymbol{d}}$ as defined in (21). Note that the first term in this energy is quadratic with respect to $\widetilde{r}$, whereas the second term is quadratic in $\widehat{\boldsymbol{d}}_t$. If the first term could be approximated by a quadratic form in $\widehat{\boldsymbol{d}}_t$, then $U^i_{\boldsymbol{d}}$ would be quadratic, and the conditional density would be Gaussian. There exist efficient techniques for generating normal bivariates, which would significantly speed up the estimation.

Assume that an approximate estimate $\dot{\boldsymbol{d}}_t$ of the displacement field is known and that the image intensity is locally approximately linear. Then, using the first-order terms of the Taylor expansion, the displaced pixel difference $\widetilde{r}$ can be expressed as follows:

$$\widetilde{r}(\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t) \approx \widetilde{r}(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t) + (\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t)-\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t))^T\nabla_{\boldsymbol{d}}\widetilde{r}(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t) \tag{24}$$

where the spatial gradient of $\widetilde{r}$ at $(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t)$ is defined as shown at the bottom of this page. Incorporating the linearization (24) into the local energy $U^i_{\boldsymbol{d}}$ (21), we obtain approximation (25) (which is shown at the bottom of the next page), where $\dot{\boldsymbol{d}}_t$ is fixed. To simplify the subsequent derivation, we temporarily use $\widetilde{r}_i$, $\widehat{\boldsymbol{d}}_i$, $\dot{\boldsymbol{d}}_i$, and $\widehat{l}_{ij}$ to denote $\widetilde{r}(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t)$, $\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t)$, $\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t)$, and $\widehat{l}(\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle,t)$, respectively. With this simplification, the energy (see (25)) can be written as follows:

$$U^i_{\boldsymbol{d}}(\widehat{\boldsymbol{d}}^z_t,\widehat{l}_t,g_{t_-},g_{t_+}) \approx \lambda_g\Big(\widetilde{r}_i+(\boldsymbol{z}-\dot{\boldsymbol{d}}_i)^T\nabla_{\boldsymbol{d}}\widetilde{r}_i\Big)^T\Big(\widetilde{r}_i+(\boldsymbol{z}-\dot{\boldsymbol{d}}_i)^T\nabla_{\boldsymbol{d}}\widetilde{r}_i\Big) + \lambda_{\boldsymbol{d}}\sum_{j:j\in\eta_i}(\boldsymbol{z}-\widehat{\boldsymbol{d}}_j)^T(\boldsymbol{z}-\widehat{\boldsymbol{d}}_j)(1-\widehat{l}_{ij}).$$

Let us fit the conditional probability density $p(\widehat{\boldsymbol{d}}(\boldsymbol{x}_i,t) \mid \widehat{\boldsymbol{d}}(\boldsymbol{x}_j,t), j\neq i, \widehat{l}_t, g_{t_-}, g_{t_+})$ with energy approximated as above

$$\nabla_{\boldsymbol{d}}\widetilde{r}(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t) = \begin{bmatrix}\widetilde{r}^x(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t)\\ \widetilde{r}^y(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),\boldsymbol{x}_i,t)\end{bmatrix} = \begin{bmatrix}\frac{\partial\widetilde{g}(\boldsymbol{x}_i-\Delta t\cdot\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),t_-)}{\partial x}\Delta t + \frac{\partial\widetilde{g}(\boldsymbol{x}_i+(1.0-\Delta t)\cdot\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),t_+)}{\partial x}(1.0-\Delta t)\\ \frac{\partial\widetilde{g}(\boldsymbol{x}_i-\Delta t\cdot\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),t_-)}{\partial y}\Delta t + \frac{\partial\widetilde{g}(\boldsymbol{x}_i+(1.0-\Delta t)\cdot\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t),t_+)}{\partial y}(1.0-\Delta t)\end{bmatrix}.$$

into the following 2-D Gaussian distribution:

$$p(\boldsymbol{z}) = \frac{1}{2\pi(\det \boldsymbol{M})^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m})^T \boldsymbol{M}^{-1}(\boldsymbol{z}-\boldsymbol{m})}$$

where $\boldsymbol{m}$ is the mean vector, and the covariance matrix $\boldsymbol{M}$ is defined as follows:

$$\boldsymbol{M} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}.$$

By completing the squares, we obtain the following equations (note that the temperature $\mathbf{T}$ from (20) must be taken into account):

$$\frac{1}{2}\boldsymbol{z}^T\boldsymbol{M}^{-1}\boldsymbol{z} = \frac{1}{\mathbf{T}}\Big(\lambda_g \boldsymbol{z}^T \nabla_{\boldsymbol{d}}\tilde{r}_i \nabla_{\boldsymbol{d}}^T \tilde{r}_i \boldsymbol{z} + \lambda_{\boldsymbol{d}} \boldsymbol{z}^T \boldsymbol{z} \sum_{j:j\in\eta_i}(1-l_{ij})\Big),$$

$$-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{M}^{-1}\boldsymbol{m} = \frac{1}{\mathbf{T}}\Big(\lambda_g \boldsymbol{z}^T \nabla_{\boldsymbol{d}}\tilde{r}_i (\tilde{r}_i - \dot{\boldsymbol{d}}_i^T \nabla_{\boldsymbol{d}}\tilde{r}_i) - \lambda_{\boldsymbol{d}} \boldsymbol{z}^T \sum_{j:j\in\eta_i} \hat{\boldsymbol{d}}_j(1-l_{ij})\Big)$$

which can be simplified to

$$\boldsymbol{M}^{-1} = \frac{2}{\mathbf{T}}(\lambda_g \nabla_{\boldsymbol{d}}\tilde{r}_i \nabla_{\boldsymbol{d}}^T \tilde{r}_i + \lambda_{\boldsymbol{d}}\xi_i I),$$

$$\boldsymbol{m} = -\frac{2}{\mathbf{T}}\boldsymbol{M}\Big(\lambda_g \nabla_{\boldsymbol{d}}\tilde{r}_i(\tilde{r}_i - \dot{\boldsymbol{d}}_i^T \nabla_{\boldsymbol{d}}\tilde{r}_i) - \lambda_{\boldsymbol{d}}\xi_i \bar{\boldsymbol{d}}_i\Big)$$

where I is the identity matrix, and ξ_i, $\bar{\boldsymbol{d}}_i$ are defined below. Computing appropriate inverses, the following parameters of the Gaussian distribution can be obtained:

$$\boldsymbol{M} = \frac{\mathbf{T}}{2\lambda_{\boldsymbol{d}}\xi_i\mu_i}(\mu_i I - \nabla_{\boldsymbol{d}}\tilde{r}_i \nabla_{\boldsymbol{d}}^T \tilde{r}_i),$$

$$\boldsymbol{m} = \bar{\boldsymbol{d}}_i - \frac{1}{\mu_i}\nabla_{\boldsymbol{d}}\tilde{r}_i\Big(\tilde{r}_i + (\bar{\boldsymbol{d}}_i - \dot{\boldsymbol{d}}_i)^T \nabla_{\boldsymbol{d}}\tilde{r}_i\Big)$$

where μ_i, the average vector $\bar{\boldsymbol{d}}_i$, and the scaling factor ξ_i are computed as follows:

$$\mu_i = \xi_i \frac{\lambda_{\boldsymbol{d}}}{\lambda_g} + \nabla_{\boldsymbol{d}}^T \tilde{r}_i \nabla_{\boldsymbol{d}}\tilde{r}_i,$$

$$\bar{\boldsymbol{d}}_i = \frac{1}{\xi_i} \sum_{j:\boldsymbol{x}_j\in\eta_{\boldsymbol{d}}(\boldsymbol{x}_i)} \hat{\boldsymbol{d}}_j(1-\hat{l}_{ij}),$$

$$\xi_i = \sum_{j:\boldsymbol{x}_j\in\eta_{\boldsymbol{d}}(\boldsymbol{x}_i)} (1-\hat{l}_{ij}).$$

Note that this definition of averaging takes into account the line elements and simply does not allow performance of such operation across a motion boundary. The horizontal and vertical component variances σ_x^2, σ_y^2, as well as the correlation coefficient ρ, which comprise the covariance matrix $\boldsymbol{M}$, have the following form:

$$\begin{bmatrix} \sigma_x^2 \\ \sigma_y^2 \end{bmatrix} = \frac{\mathbf{T}}{2\lambda_{\boldsymbol{d}}\xi_i\mu_i} \begin{bmatrix} \xi_i \frac{\lambda_{\boldsymbol{d}}}{\lambda_g} + (\tilde{r}_i^y)^2 \\ \xi_i \frac{\lambda_{\boldsymbol{d}}}{\lambda_g} + (\tilde{r}_i^x)^2 \end{bmatrix}$$

$$\rho\sigma_x\sigma_y = -\mathbf{T}\frac{\tilde{r}_i^x \tilde{r}_i^y}{2\lambda_{\boldsymbol{d}}\xi_i\mu_i}. \tag{26}$$

The initial vector $\dot{\boldsymbol{d}}_t$ can be assumed zero throughout the estimation process, but with increasing $\hat{\boldsymbol{d}}_t$, the error due to intensity nonlinearity would significantly increase. Hence, it is better to "track" an intensity pattern by modifying $\dot{\boldsymbol{d}}_t$ accordingly. An interesting result can be obtained when it is assumed that at every iteration of the Gibbs sampler, $\dot{\boldsymbol{d}}_t = \bar{\boldsymbol{d}}_t$, i.e., the approximate displacement field is equal to the average from the previous iteration. Then, returning to the original notation, the estimation process can be described by the following iterative equation:

$$\hat{\boldsymbol{d}}^{k+1}(\boldsymbol{x}_i, t) = \bar{\boldsymbol{d}}^k(\boldsymbol{x}_i, t) - \frac{\tilde{r}(\bar{\boldsymbol{d}}^k(\boldsymbol{x}_i,t), \boldsymbol{x}_i, t)}{\mu_i}\nabla_{\boldsymbol{d}}\tilde{r}(\bar{\boldsymbol{d}}^k(\boldsymbol{x}_i,t), \boldsymbol{x}_i, t) + \boldsymbol{n}_i \tag{27}$$

where k is the iteration number. $\boldsymbol{n}_i$ is a Gaussian bivariate with the covariance matrix M, which along with μ_i is defined as before except for $\bar{\boldsymbol{d}}_t^k$ replacing $\dot{\boldsymbol{d}}_t$.

The continuous state space Gibbs sampler described above results in a spatio-temporal gradient estimation method, whereas the discrete state space Gibbs sampler from the previous section is an example of an explicit (pixel) matching algorithm. This important difference is due to the Taylor expansion used in approximating the displaced pixel difference $\tilde{r}$ by the linear form (24).

Note the similarity of the iterative update equation (27) to the update equation of the Horn-Schunck algorithm [17]. Except for the displaced pixel difference replacing the motion constraint equation and the inclusion of the random vector $\boldsymbol{n}_i$, they are identical. It is interesting that similar update equations result from two different approaches: Horn and Schunck establish a necessary condition for optimality and solve a linear system by deterministic relaxation, whereas here, a 2-D Gaussian distribution is fitted into the conditional marginal probability density to obtain the Gibbs sampler update formula. For $\mathbf{T}$=0 and $\dot{\boldsymbol{d}}_t = \bar{\boldsymbol{d}}_t$, the continuous state space Gibbs sampler is equivalent to a variation of the Horn-Schunck algorithm, or in other words, their algorithm can be viewed as an instantly "frozen" simulated annealing.

Note that at the beginning, when the temperature is high, the random term $\boldsymbol{n}_i$ has large variance, and the estimates assume

$$U_{\boldsymbol{d}}^i(\hat{\boldsymbol{d}}_t^z, \hat{l}_t, g_{t_-}, g_{t_+}) \approx \lambda_g\Big[\tilde{r}(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t), \boldsymbol{x}_i, t) + (\boldsymbol{z} - \dot{\boldsymbol{d}}(\boldsymbol{x}_i,t))^T \nabla_{\boldsymbol{d}}\tilde{r}(\dot{\boldsymbol{d}}(\boldsymbol{x}_i,t), \boldsymbol{x}_i, t)\Big]^2 + \lambda_{\boldsymbol{d}} \sum_{j:\boldsymbol{x}_j\in\eta_{\boldsymbol{d}}(\boldsymbol{x}_i)} \|\boldsymbol{z} - \hat{\boldsymbol{d}}(\boldsymbol{x}_j,t)\|^2[1-\hat{l}(\boldsymbol{x}_i,\boldsymbol{x}_j),t)], \tag{25}$$

quite random values. As the temperature T is reduced to zero, the variances and the correlation coefficient get smaller, thus reducing the random term of the estimate. In the limit, the algorithm performs a deterministic update. In addition, observe that the variance σ_x^2 in (26) for fixed $\lambda_{\boldsymbol{d}}/\lambda_g$ and $\widehat{r}^y$ decreases with growing $\widehat{r}^x$. It means that if there is a significant horizontal gradient (detail) in the image structure, the uncertainty of the estimate in the horizontal direction is small. The same relationship applies to σ_y^2. Hence, the algorithm takes into account image structure in order to determine the amount of randomness allowed at a given temperature.

As stated at the beginning of this section, the continuous state space approach should be more efficient computationally due to generation of Gaussian bivariates, as opposed to generation of bivariates with arbitrary discrete probability distribution (Section III-C-1). Due to the complexity of both approaches, it is very difficult to calculate the number of multiplications and additions needed. We have found, however, that for a maximum displacement of two pixels and a quarter-pel precision (typical values for slow motion in broadcast applications), the computational cost for the continuous state space Gibbs sampler should be at least two orders of magnitude smaller than that for the discrete state space case. This finding has been confirmed experimentally and is discussed in Section V-B-2.

IV. Bayesian Estimation Over a Hierarchy of Resolutions

There are two major reasons for carrying out motion computation over a hierarchy of resolutions: to speed up the estimation process and to reduce violation of underlying assumptions. The discrete state space Gibbs sampler, because it is a stochastic exhaustive search method, requires a computational effort proportional to the size of the state space searched. When incorporated into simulated annealing at one resolution level, it is capable of locating the global optimum but at a substantial computational cost. In the case of the continuous state space Gibbs sampler, however, the major reason for the hierarchical approach is to minimize the violation of the assumed locally linear intensity variation.

The basic operation in a hierarchical algorithm is data prefiltering, which, when applied in a cascaded fashion, results in an even or odd[4] data pyramid [12]. Usually, after each stage of filtering, the data are subsampled [6] to provide a more compact representation at a given resolution level. In motion estimation with subpixel accuracy, however, such subsampling leads to unnecessary interpolation errors. Hence, we will use a constant-width odd pyramid for images. A similar pyramid is constructed for the displacement fields. Subsampling in such a pyramid is not obligatory, but since the data lack high-frequency content at lower resolution levels, it is reasonable to specify displacement vectors on a subsampled grid. Such subsampling will cause faster convergence of the estimation algorithm because of fewer vectors at the top of the pyramid and because of faster propagation of smoothness constraints over increased absolute distances.

Let the sequence of sample fields $\{g_{t_\pm}^\kappa, \kappa = 0, 1, \ldots, K_l - 1\}$ denote a constant-width odd pyramid of images with K_l being the number of resolution levels and subscript $t_\pm$ denoting either t_- or t_+. $g_{t_\pm}^0$ denotes the full resolution image, whereas the image $g_{t_\pm}^\kappa$ is obtained by filtering the image $g_{t_\pm}^{\kappa-1}$. The Gaussian (low-pass) [9] and Laplacian (bandpass) [1] pyramids of images have previously been used in hierarchical motion estimation. We used Nyquist-like low-pass 2-D separable FIR filters [24] to allow 2:1 subsampling with minimal aliasing.

Let the sequences $\{\widehat{\boldsymbol{d}}_t^\kappa, \kappa = 0, 1, \ldots, K_l - 1\}$ and $\{\widehat{l}_t^\kappa, \kappa = 0, 1, \ldots, K_l - 1\}$ denote an odd pyramid of displacement field estimates and an even pyramid of discontinuity field estimates, respectively.

Once an estimate is obtained at level κ, it must be transformed to the next higher resolution level for subsequent improvement. The interpolation $\mathcal{I}_\kappa$ between levels κ and $\kappa - 1$ can be expressed as follows:

$$(\boldsymbol{b}_t^{\kappa-1}, m_t^{\kappa-1}) = \mathcal{I}_\kappa(\widehat{\boldsymbol{d}}_t^\kappa, \widehat{l}_t^\kappa)$$

where $\boldsymbol{b}_t^{\kappa-1}$ and $m_t^{\kappa-1}$, here known as the *base* displacement and line fields at level $(\kappa - 1)$, are the results of interpolation $\mathcal{I}_\kappa$. For displacement interpolation, we use the neighbor repetition of κ-level estimates at the missing positions of level $(\kappa - 1)$. The interpolation of discontinuities defined over an even pyramid is less straightforward. Since, at lower resolution levels, the motion discontinuity estimates may be somewhat unreliable, it may not be useful to use them explicitly at the subsequent resolution level (m_t^κ=0). Instead, they can be used implicitly through the motion field, i.e., the discontinuities "encoded" into the motion field are passed through the displacement interpolation stage. The line process is absent for a number of iterations and is turned on once a coarse displacement estimate is known (other possibilities are discussed in [24]).

Usually, hierarchical motion-estimation algorithms use the previous-level estimate as the initial state at a subsequent level and allow arbitrary estimates (within a given state space) thereafter. This strategy is not suitable for the discrete state space Gibbs sampler since the key problem is to speed up the computations or, equivalently, to restrict the size of $\mathcal{S}'_{\boldsymbol{d}}$ for $\kappa > 0$. If the solution from the previous level is close to the optimal one, a limited spatial area around this initial solution can be searched for the new estimate.

Hence, the *base* displacement field $\boldsymbol{b}_t^\kappa$ will not be used at level κ merely as the initial state but as a coarse solution that is fine tuned at subsequent levels. The field $\boldsymbol{b}_t^\kappa$ is a fixed array of vectors and is used to identify the centers of new single-vector state spaces at level κ. Application of the MAP estimation algorithm yields an *incremental* displacement field $\widehat{\boldsymbol{h}}_t^\kappa$ so that the final estimate at level κ is given by $\widehat{\boldsymbol{d}}_t^\kappa = \widehat{\boldsymbol{h}}_t^\kappa + \boldsymbol{b}_t^\kappa$. Similarly, the final line field estimate at level κ is $\widehat{l}_t^\kappa = \widehat{k}_t^\kappa \oplus m_t^\kappa$ where $\widehat{k}_t^\kappa$ is an incremental line field, and $\oplus$ is a nonlinear (e.g., *modulo 2*) operator combining two line elements.

[4] In even pyramids, lower resolution pixels are half-pixel shifted with respect to the preceding higher resolution pixels, whereas in odd pyramids, they are aligned.

With the above notation, the new energy function for the hierarchical MAP estimation can be expressed as follows:

$$U(\hat{\boldsymbol{h}}_t^\kappa, \hat{k}_t^\kappa, \boldsymbol{b}_t^\kappa, m_t^\kappa, g_{t-}^\kappa, g_{t+}^\kappa) = \lambda_g^\kappa \sum_{i=1}^{M_{\boldsymbol{d}}} [\tilde{r}^\kappa(\hat{\boldsymbol{h}}^\kappa(\boldsymbol{x}_i, t) + \boldsymbol{b}^\kappa(\boldsymbol{x}_i, t), \boldsymbol{x}_i, t)]^2 + \lambda_{\boldsymbol{d}}^\kappa U_{\boldsymbol{d}}(\hat{\boldsymbol{h}}_t^\kappa + \boldsymbol{b}_t^\kappa | \hat{k}_t^\kappa \oplus m_t^\kappa) + \lambda_l^\kappa U_l(\hat{k}_t^\kappa \oplus m_t^\kappa | g_{t-}^\kappa)$$

where the incremental estimates $\hat{\boldsymbol{h}}^\kappa$ and $\hat{k}^\kappa$ are variable, whereas the base estimates $\boldsymbol{b}_t^\kappa$ and m_t^κ are fixed for given κ. $\tilde{r}^\kappa$ is the displaced pixel difference as before but evaluated for images $g_{t\pm}^\kappa$.

V. Experimental Results

A. Test Images

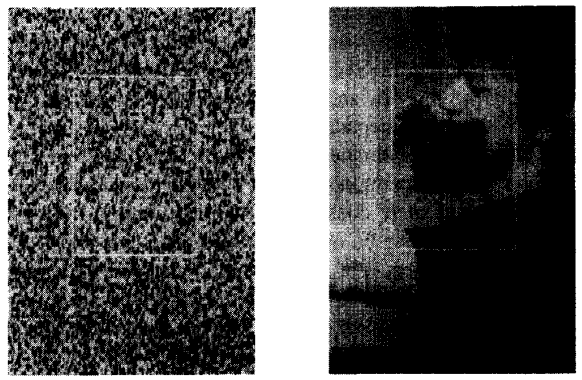
The algorithms described above have been tested on a number of images with both synthetic and natural motion. The images, which are either captured by a video camera or computer generated, have been stored in a displayable line-interlaced format with interfield distance $\tau_{60} = 1/60$ s.

To provide a quantitative test, we used two image sequences with synthetic motion. Fig. 7(a) shows a pattern based on the concept of a random dot stereogram. The 0th field of the sequence consists of random, uniformly distributed numbers from the range [40 200]. The subsequent even fields are exact copies of the 0th field, except for a 50 by 20 pixel rectangle $\mathcal{R} \in \Lambda_g$ in the center of the outlined area, which has been moved by $\boldsymbol{d}_s$=(2.0, 1.0) with respect to the previous even field. The odd fields are exact replicas of the preceding even fields. Fig. 7(b) shows the test image 2, which provides a synthetic motion of natural data obtained from a video camera. The background is provided by the test image 3, whereas the 45 by 20 pixel rectangle $\mathcal{R} \in \Lambda_g$ in the center is obtained from another image through low-pass filtering and subsampling. In subsequent fields, the same background image is used, whereas the moving pixels in the rectangle ($\boldsymbol{d}_s$ =(1.5, 0.5)) are obtained from appropriately shifted pixels in the prefiltered image. Unlike in the case of test image 1, this test pattern permits noninteger displacements, which is a more realistic situation since there is no perfect data matching. In both images, the white frame outlines the area of 77 by 49 pixels used in the estimation.

Fig. 8 shows the 0th field of a natural sequence obtained with a video camera. The acquisition process included, as usual, camera filtering, sampling, and quantization, as described in Section II-C-2. There is some aliasing present in the data due to insufficient vertical filtering before sampling (as is typical in most camera-captured imagery with interlace). No filtering or any other processing has been applied to the sequence after acquisition. The white frame outlines the area of 221 by 69 pixels used in estimation.

B. Estimation Results

The motion estimates presented below have been obtained from pairs of fields separated by $T_g = 2\tau_{60}$, with Δt=0.0 (i.e., forward estimation), and $\Lambda_{\boldsymbol{d}} = \Lambda_g$. Estimates for other Δt and $\Lambda_{\boldsymbol{d}}$ can be found in [24].



(a) (b)

Fig. 7. Test images with synthetic motion and a 77 by 49 pel window used in estimation: (a) Test image 1, $|\mathcal{R}| = 50 \times 20$ pixels, $\boldsymbol{d}_s$=(2.0, 1.0); (b) test image 2, $|\mathcal{R}| = 45 \times 20$ pixels, $\boldsymbol{d}_s$=(1.5, 0.5).

Fig. 8. Test image 3 with natural motion and a 221 by 69 pel window used in estimation.

The discrete state space stochastic relaxation used in simulations was based on $\mathcal{S}'_{\boldsymbol{d}}$ with d_{max}=2.0 and N_d=17 levels in each direction. This choice of parameters gives the upper bound of a quarter pixel on the predicted accuracy of motion vectors in the absence of such unaccounted effects as occlusions or illumination change. The continuous state space estimation, however, can, in principle, give finer accuracy if all underlying assumptions are satisfied. In practice, the observed accuracy in both cases depends on the extent to which assumptions are violated, which is difficult to assess, and on such parameters as annealing schedule, motion, and image models, etc. The annealing schedule is particularly difficult to establish since its parameters (initial temperature, number of iterations) are highly dependent on the data. At the moment, we do not know how to choose parameters of the algorithms in order to obtain given accuracy in practice, although we have been able to establish, through experimentation on synthetic-motion sequences, those models and parameter ranges that give lower mean square errors than others [24].

To model the motion, neighborhood systems from Figs. 2 and 4 with potentials (11) and (14) were used. Since the

logarithmic annealing schedule requires too many iterations to achieve very low temperatures, we used a suboptimal exponential schedule (Section III-A) with a close to 1.0. The initial temperature $\mathbf{T}_0$ and the number of iterations N have been chosen experimentally: $\mathbf{T}_0$ sufficiently high in order not to deteriorate the estimates and N sufficiently large in order to attain final temperatures around 0.1. We have found that a further increase of N does not substantially improve the final estimates, which can be explained by the temperature having reached the "plateau" of the annealing schedule. Certainly, instead of setting N in advance, one could monitor individual and/or total energies from (16) and stop the estimation process once energy fluctuations fall under a given threshold.

Since the true motion fields are known for the test images 1 and 2 (except for the occlusion and newly exposed areas), the quality of motion field estimates can be assessed through the mean squared error and the bias defined as follows:

$$MSE = E[(\boldsymbol{d}_s - \widehat{\boldsymbol{d}})^2] \approx \frac{1}{|\mathcal{R}|} \sum_{\boldsymbol{x}_i \in \mathcal{R}} [\boldsymbol{d}_s(\boldsymbol{x}_i, t) - \widehat{\boldsymbol{d}}(\boldsymbol{x}_i, t)]^2$$

$$bias = E[\boldsymbol{d}_s - \widehat{\boldsymbol{d}}] \approx \frac{1}{|\mathcal{R}|} \sum_{\boldsymbol{x}_i \in \mathcal{R}} [\boldsymbol{d}_s(\boldsymbol{x}_i, t) - \widehat{\boldsymbol{d}}(\boldsymbol{x}_i, t)].$$

1) Results for Test Image 1: Since the images used for estimation match exactly except for the newly exposed and occlusion areas, the estimation should strongly rely on the data and not so much on the displacement model. Consequently, λ_g is set to 1.0, and $\lambda_{\boldsymbol{d}}$ equals 0.05, resulting in a weak *a priori* constraint.

Fig. 9(a) shows a MAP estimate for the exponential annealing schedule with $\mathbf{T}_0$=1.0, a=0.98, and N=200. Note that except for few spurious small vectors in the background and the occluded bottom and right edges of the rectangle, the estimate is very close to the true motion. This is confirmed by small mean-squared error and bias. Application of the continuous state space MAP estimation resulted in an estimate that had little in common with the true motion. This is not surprising since the method relies on the relationship between the spatial and temporal intensity gradients, and those are absolutely unrelated in the test image 1.

Fig. 9(b) shows a MEC estimate obtained with the discrete state space Gibbs sampler for $\mathbf{T}$=0.1. The average (19), which approximates the ensemble expectation, has been computed over the last 150 iterations (N=200). This result demonstrates that MEC estimation can also provide reliable estimates without the need to use an annealing schedule.

Fig. 9(c) and (d) show the displacement and line estimates for the model incorporating motion discontinuities with λ_l/λ_d=1.2. Due to the very strong motion cue in the data, the subjective improvement, compared with the motion field estimate from Fig. 9(a), is rather minor. The mean squared error and the bias, however, have been reduced to zero! Within the rectangle, the estimate is exactly equal to the true motion. Note that the well-defined motion boundary along the top and left edges of the rectangle is correctly estimated by the algorithm. The other two edges corresponding to the occlusion border, where the motion boundary is not well defined,[5] are characterized by numerous line elements splitting the occlusion regions into small patches and thus disallowing significant contribution from vectors located there to the displacement energy. Note that since the motion model does not take the occlusion and exposure effects into account, poorer performance of the algorithm in such areas is not unexpected.

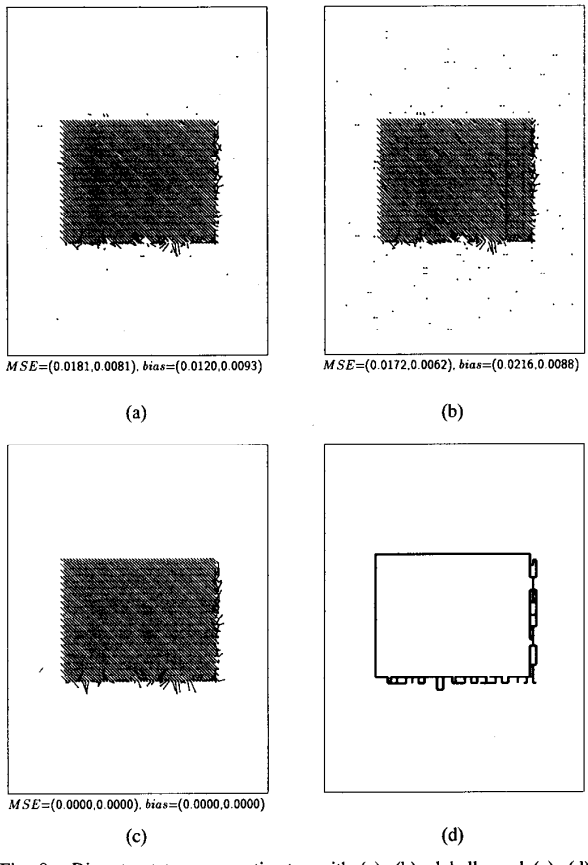
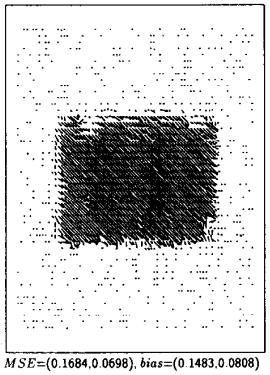


Fig. 9. Discrete state space estimates with (a), (b) globally and (c), (d) piecewise smooth motion model for test image 1, $\lambda_{\boldsymbol{d}}/\lambda_g$=0.05, bilinear interpolation: (a) MAP, $\mathbf{T}_0$=1.0, a=0.98, 200 iter.; (b) MEC, $\mathbf{T}$=0.1, 200 iter.; (c), (d) MAP $(\widehat{\boldsymbol{d}}_t, \widehat{l}_t)$, $\lambda_l/\lambda_{\boldsymbol{d}}$=1.2, $\mathbf{T}_0$=1.0, a=0.9866, 400 iter.

To test the robustness of the algorithms in the presence of noise, additive white Gaussian noise with the variance σ_a^2=20.0 has been superimposed on the test image 1. Choosing $\beta_{\boldsymbol{d}}$ in the range 0.1–1.0, the original ratio $\lambda_{\boldsymbol{d}}/\lambda_g$ should be augmented by $2\sigma_a^2/\beta_{\boldsymbol{d}}$ valued at around 100.0 ($\beta_{\boldsymbol{d}}$=0.4).

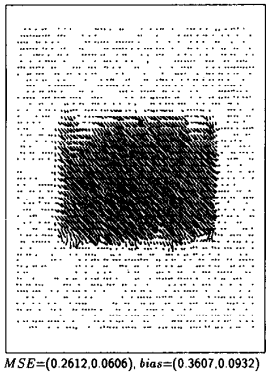
Fig. 10(a) and (b) show, respectively, the MAP and MEC estimates of motion from the noise-corrupted test image 1. Note that higher temperatures have been used in both cases. In spite of significant noise, both estimates resemble well the corresponding estimates obtained from the images without noise (Fig. 9(a) and (b)). These results demonstrate that both the MAP and the MEC estimation are quite robust to noise. The ratio $\lambda_{\boldsymbol{d}}/\lambda_g$ could have been chosen differently since the value of $\beta_{\boldsymbol{d}}$ is not known; however, values of $\beta_{\boldsymbol{d}}$ even as low

[5]This motion boundary is not well defined since in the occlusion areas (to the right and to the bottom of the rectangle), the motion itself is not well defined.



MSE=(0.1684,0.0698), *bias*=(0.1483,0.0808)

(a)



MSE=(0.2612,0.0606), *bias*=(0.3607,0.0932)

(b)

Fig. 10 Discrete state space estimates with globally smooth motion model for test image 1 corrupted by white Gaussian noise (σ_a^2=20.0), $\lambda_{\boldsymbol{d}}/\lambda_g$=100.0, bilinear interpolation, 200 iter: (a) MAP, $\mathbf{T}_0$=20.0, a=0.965; (b) MEC, **T**=2.0.

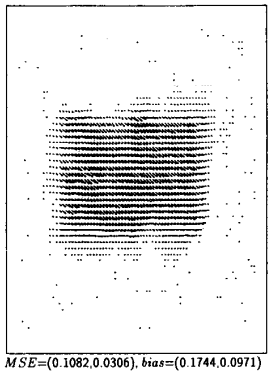
as 0.1 did not change the estimate dramatically. With the value of 0.4 used here, the results for noisy and noiseless data are quite comparable.

2) Results for Test Image 2: Since the test sequence 2 contains no perfect matching, the estimation process should rely to a higher degree on the *a priori* motion model. Assuming that the data contains noise with variance σ^2 of the order of 1.0–10.0 and that the motion field samples should be characterized by the "activity" $\beta_{\boldsymbol{d}}$ around 0.1–1.0, the ratio $\lambda_{\boldsymbol{d}}/\lambda_g$ should be on the order of 20.0. The exact value cannot be easily established, however, as it has been demonstrated in [24] that even a two orders of magnitude change in this ratio results in similar motion fields.

Fig. 11 shows the continuous and discrete state space MAP estimates for a globally smooth motion model. We have found that the continuous state space process requires smaller temperature changes, and thus, a longer annealing schedule starting at a higher initial temperature is used for the continuous case. The estimates are similar subjectively, but the continuous state space estimate performs slightly better in terms of MSE. Fig. 11(c) shows the MEC estimate for **T**=1.0. Note that it is quite similar to the MAP estimates; however, its MSE is higher.

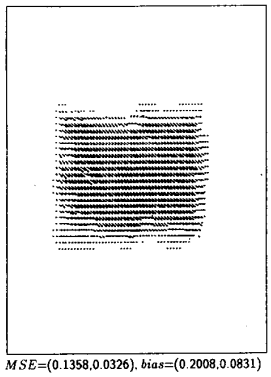
To evaluate the complexity of continuous and discrete approaches, we have measured the CPU time per iteration in both cases. The ratio of these times agrees well with the two orders of magnitude improvement (in favor of continuous state space approach) approximately evaluated in Section III-C-2. Taking into account, however, the need for a longer annealing schedule in the continuous case, the overall speed up due to the Gaussian approximation is closer to one order of magnitude.

Fig. 12 shows the displacement and line estimates for the discrete state space MAP algorithm with a piecewise smooth motion model. Note the substantially improved subjective quality of motion estimates at the rectangle boundaries, which is confirmed by the reduced MSE. The piecewise smooth motion model permits more abrupt transitions in the displacement estimates, which are thus closer to the true motion. The image energy U_g, which reflects the precision of matching, is



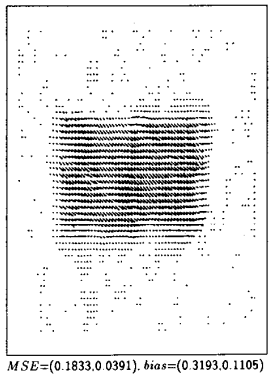
MSE=(0.1082,0.0306), *bias*=(0.1744,0.0971)

(a)



MSE=(0.1358,0.0326), *bias*=(0.2008,0.0831)

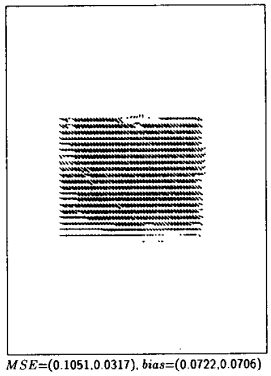
(b)



MSE=(0.1833,0.0391), *bias*=(0.3193,0.1105)

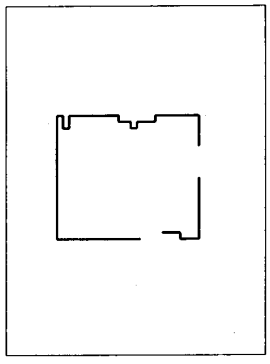
(c)

Fig. 11 Estimates for test image 2 with globally smooth motion model, $\lambda_{\boldsymbol{d}}/\lambda_g$=20.0, Keys bicubic interpolation: (a) Continuous state space MAP, $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter; (b) discrete state space MAP, $\mathbf{T}_0$=1.0, a=0.98, 200 iter; (c) discrete state space MEC, **T**=1.0, 200 iter.



MSE=(0.1051,0.0317), *bias*=(0.0722,0.0706)

(a)



(b)

Fig. 12. Discrete state-space MAP estimate with piecewise smooth motion model for test image 2, $\lambda_{\boldsymbol{d}}/\lambda_g$=20.0, $\lambda_l/\lambda_{\boldsymbol{d}}$=0.8, α=10.0, Keys bicubic interpolation, $\mathbf{T}_0$=1.0, a=0.9866, 300 iter: (a) $\hat{\boldsymbol{d}}_t$; (b) $\hat{l}_t$.

substantially reduced here compared with the estimate from Fig. 11(b) based on the globally smooth motion model.

In Fig. 13, the result of hierarchical three-level continuous state-space MAP estimation of motion from test image 2 with $T_g = 4\tau_{60}$ is presented. The piecewise smooth motion model

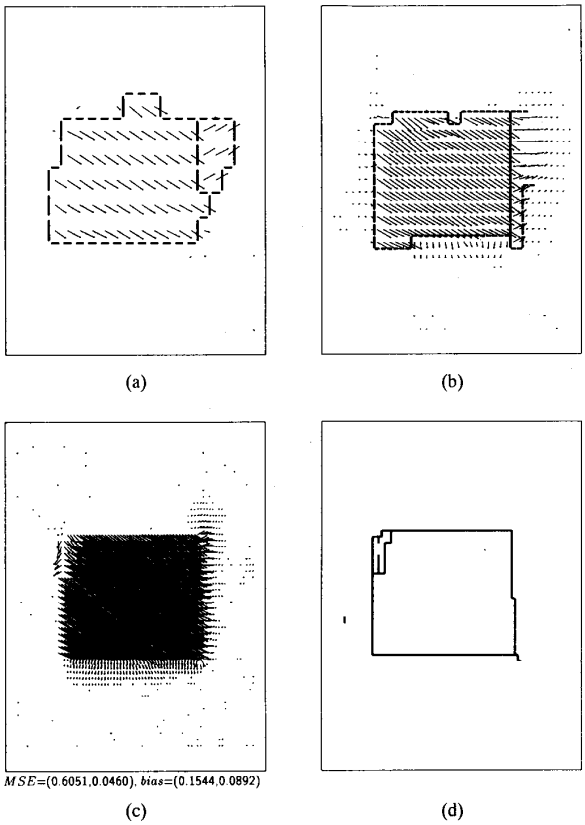


Fig. 13. Hierarchical continuous state-space MAP estimate with piecewise smooth motion model for test image 2, K_l=3, λ_d/λ_g=(20.0, 12.0, 10.0), λ_l/λ_d=1.8, α=10.0, Keys bicubic interpolation, $\mathbf{T}_0$=(1.0, 2.0, 4.0), a=0.992, 500 iter at each level: (a) κ=2; (b) κ=1; (c), (d) κ=0.

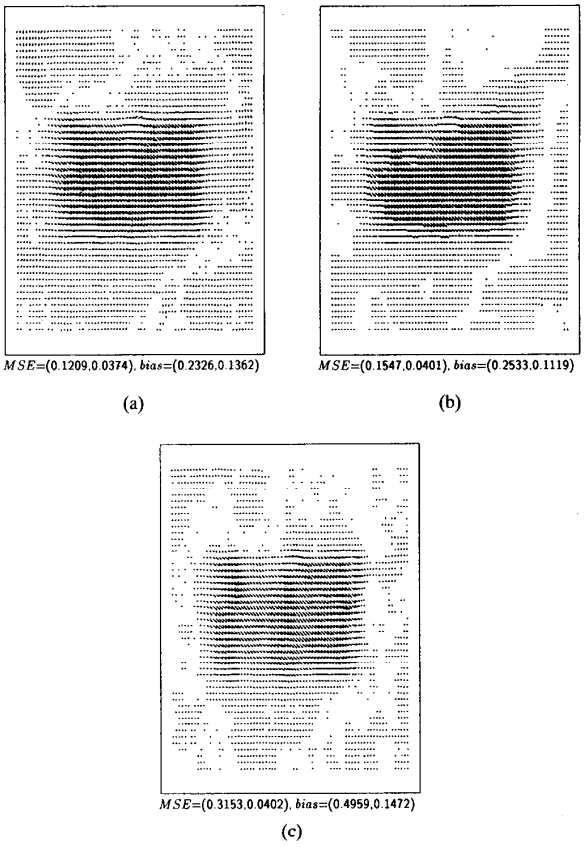


Fig. 14. Estimates with globally smooth motion model for test image 2 corrupted by white Gaussian noise (σ_a^2=20.0), λ_d/λ_g=120.0, Keys bicubic interpolation: (a) Continuous state space MAP, $\mathbf{T}_0$=20.0, a=0.993, 1000 iter; (b) discrete state space MAP, $\mathbf{T}_0$=20.0, a=0.965, 200 iter; (c) discrete state space MEC, $\mathbf{T}$=1.0, 200 iter.

has been applied in this case, and the Nyquist-like low-pass filtering has been used in data pyramid generation. Due to the filtering, the variance σ^2 of the DPD model is reduced at lower resolution levels, and the weights λ are adjusted as follows: λ_d/λ_g= 20.0, 12.0, and 10.0 at levels κ=0, 1, 2, respectively (a more detailed discussion of this adjustment can be found in [24]). Note that despite the smoothness constraint, the estimates at all levels clearly identify the moving rectangle. The transition is less abrupt in the newly exposed areas where motion is ill defined. The estimate is incorrect in the top-left corner of the rectangle, which is due to image breakup caused by a newly exposed area. These effects have not been accounted for in the motion models.

Fig. 14 shows estimates of motion from the test image 2 corrupted by additive white Gaussian noise. The ratio λ_d/λ_g= 120.0 has been used in each case. The initial temperature has been set to 20.0 for both MAP estimations, whereas the constant temperature of 1.0 has been used in the MEC algorithm. The estimates resemble quite well the corresponding true motion fields; however, they are not as good as the estimates from the noiseless data. Poorer performance for the noise-corrupted test image 2 (compared with the test image 1) is not surprising since its motion cue is much weaker, and consequently, imposition of noise masks the motion to a larger extent. In the test image 1, however, due to the "gray value corners" that are almost everywhere, such a masking effect is much less pronounced.

From the above experiments, as well as from the experiments with noisy test image 3 [22], we conclude that the MAP and MEC estimates are similar. No superiority of the MEC estimation in a noisy environment can be noticed, unlike the results observed by Marroquin [27] with respect to scalar MRF's applied to image reconstruction. This may be due to the fact that only noise with a variance of 20.0 has been tested and that the state spaces used were much larger here than in Marroquin's case.

3) Results for Test Image 3: Since test image 3 contains natural data as well as natural motion, the importance of the *a priori* motion model is reflected in increased ratio λ_d/λ_g=20.0. For clarity, the displayed vector fields have been subsampled by 2 in each direction.

Fig. 15 shows a continuous state space MAP estimate, whereas the discrete state space MAP and MEC estimates are not presented here (they can be found in [22]) because of their similarity to the continuous state-space result. The discrete state-space MAP estimate has been outperformed, however, by the continuous one in terms of the parametrizing energies

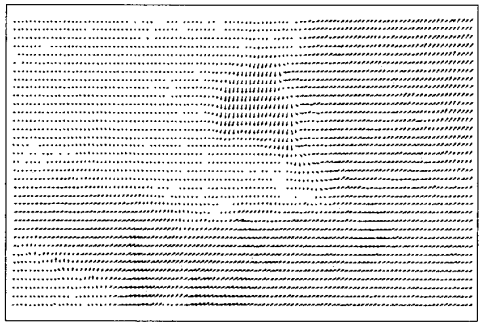
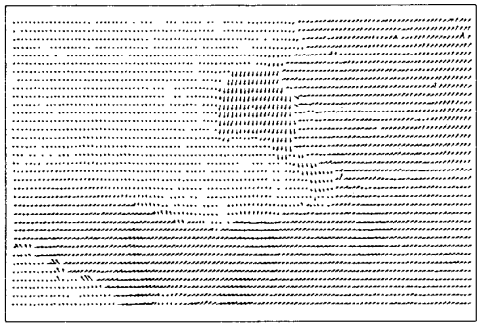
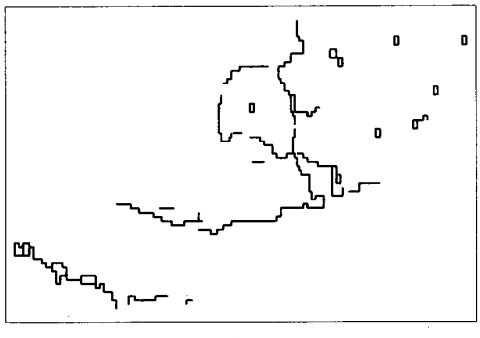


Fig. 15. Continuous state-space MAP estimate with globally smooth motion model for test image 3, $\lambda_{\boldsymbol{d}}/\lambda_g$=20.0, Keys bicubic interpolation, $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter.



(a)

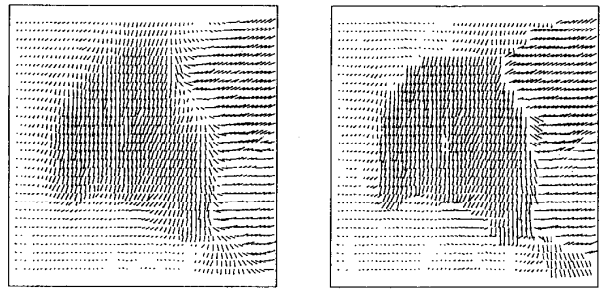


(b)

Fig. 16. Continuous state-space MAP estimate with piecewise smooth motion model for test image 3, $\lambda_{\boldsymbol{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\boldsymbol{d}}$=0.8, α=10.0, Keys bicubic interpolation, $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter: (a) $\widehat{\boldsymbol{d}}_t$; (b) $\widehat{l}_t$.

(the MEC estimate cannot be compared). The estimate from Fig. 15 is smooth within moving objects, and this smoothing is also applied across motion boundaries. This can be explained by the fact that motion boundaries are not as strong as in the test image 1.

Fig. 16 shows the displacement and line estimates obtained through the continuous state space MAP estimation with the piecewise smooth motion model. Observe the well-identified motion boundaries of the moving palm of the hand, of the face, and of parts of the arm. Large fragments of motion boundaries are missing due to insufficient intensity gradient there, but motion boundaries coinciding with substantial intensity gradients are easily detected. Again, the image energy



(a) (b)

Fig. 17. Central parts of continuous state-space MAP estimates (no spatial subsampling of vectors applied): (a) Globally smooth motion model from Fig. 15; (b) piecewise smooth motion model from Fig. 16(a).

U_g is significantly reduced compared with the result without the line model (Fig. 15).

To demonstrate the impact of line process on the smoothness of the motion field in the test image 3, Fig. 17 shows windows of 60 by 30 vectors taken from the central parts of motion fields presented in Figs. 15 and 16(a). Note the sharp transitions between the palm of the hand and the background or the face in Fig. 17(b). These transitions are responsible for more precise motion portrayal and significant reduction of image energy without affecting smoothness of the motion field within moving objects. One may expect that due to the introduction of a two-layer motion model, the smoothing can be increased substantially without effects across motion boundaries. An example of discrete state space MAP estimate with the ratio $\lambda_{\boldsymbol{d}}/\lambda_g$= 100.0 for test image 3 (confirming this observation) can be found in [25].

Finally, Fig. 18 shows the continuous state space MAP estimates with the piecewise smooth motion model over a hierarchy of resolutions. The same ratios $\lambda_{\boldsymbol{d}}/\lambda_g$ as the ones for test image 2 were used. The line model parameters $\lambda_l/\lambda_{\boldsymbol{d}}$=1.0 and α=10.0 as well as the annealing schedule $\mathbf{T}_0$=(1.0, 2.0, 4.0), a=0.9944, and N=500 have been chosen experimentally. The line process was turned on after 100 150 200 iterations to maintain similar initial temperature at each level. Note that the discontinuity estimates at levels κ=2 and 1 are not very precise; however, they limit the spread of smoothness across the hand and face contours. The precision in positioning of line elements is improved at the full resolution level when the contours of the hand, face, and parts of the arm are quite well established. The impact of the two-layer motion model is further confirmed by the 25% reduction in image energy U_g.

VI. Summary and Conclusions

This paper has presented a new stochastic approach to the estimation of 2-D motion from time-varying images. The approach is stochastic in both the formulation, where probabilistic models are used in the MAP or MEC estimation criteria, and in the solution, where stochastic relaxation is incorporated into simulated annealing or into averaging. Two motion models were proposed: a globally smooth model and a piecewise smooth model with motion discontinuities.

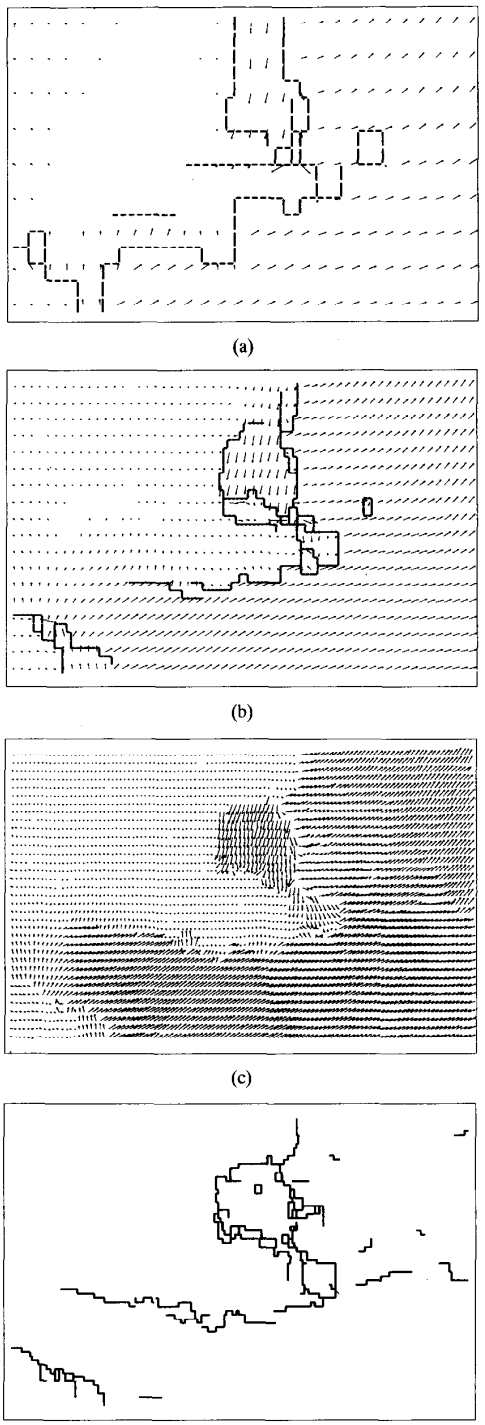


Fig. 18. Hierarchical continuous state-space MAP estimate with piecewise smooth motion model for test image 3, K_I=3, λ_d/λ_g= 20.0, λ_l/λ_d= 1.0, α=10.0, Keys bicubic interpolation, $\mathbf{T}_0$=(1.0, 2.0, 4.0), a=0.9944, 500 iter at each level: (a) κ=2; (b) κ=1; (c), (d) κ=0.

The stochastic relaxation for motion fields was implemented through the Gibbs sampler in two versions: discrete and continuous state space. The discrete version incorporated into simulated annealing resulted in a stochastic exhaustive pixel-matching algorithm with a smoothness constraint, whereas the continuous one was shown to be a stochastic generalization of the Horn-Schunck algorithm. Both methods were incorporated into a hierarchical framework to handle large displacements efficiently.

To verify the validity of proposed models and algorithms, numerous natural and computer-generated image sequences with natural and synthetic motion have been used. The MEC algorithm performed similarly to the MAP estimation both for standard and noise-corrupted images. The continuous state-space MAP estimation outperformed the discrete version both in terms of MSE (slightly) and in terms of computational efficiency (an order of magnitude). Compared with the globally smooth model, the piecewise smooth model provided more accurate motion portrayal both objectively and subjectively and simultaneously significantly reduced matching errors. The hierarchical version of MAP estimation was demonstrated to reliably estimate large displacements without a substantial increase in computational effort.

In conclusion, the stochastic approach presented here provides reliable and close to optimal motion estimates (given the assumptions). We have recently addressed the issue of the high complexity of the method, and we are currently investigating deterministic approximations to the above stochastic algorithms [26]. Further work is also required to extend the models for such conditions as occlusions and varying intensity along motion trajectories.

REFERENCES

[1] P. Anandan, "Measuring visual motion from image sequences," Ph.D. thesis, Univ. of Mass., May 1987.

[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] ——, "On the statistical analysis of dirty pictures," *J. Roy. Stat. Soc.*, vol. B 48, pp. 259–279, 1986.

[5] P. Bouthemy and P. Lalande, "Motion detection in an image sequence using Gibbs distributions," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, May 1989, pp. 1651–1654.

[6] P. Burt, "Fast filter transforms for image processing," *Comput. Vision Graphics Image Processing*, vol. 16, pp. 20–51, 1981.

[7] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, Jan. 1987.

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

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

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

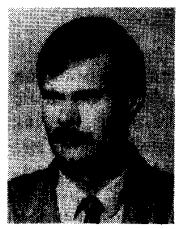
[11] M. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," Tech. Rep. 975, Mass. Inst. Technol. Artificial Intell. Lab., Cambridge, June 1987.

[12] F. Glazer, "Hierarchical motion detection," Ph.D. thesis, Univ. of Mass., Feb. 1987.

[13] M. Hassner and J. Sklansky, "The use of Markov random fields as models for texture," in *Image Modeling* (E. A. Rosenfeld, Ed.). New York: Academic, 1981, pp. 185–198.

[14] F. Heitz and P. Bouthemy, "Motion estimation and segmentation using a global Bayesian approach," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, 1990, pp. 2305–2308.

[15] ——, "Multimodal motion estimation and segmentation using Markov random fields," in *Proc. IEEE Int. Conf. Patt. Recogn.*, June 1990, pp. 378–383.
[16] E. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intell.*, vol. 23, pp. 309–354, 1984.
[17] B. P. Horn and B. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185–203, 1981.
[18] J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing motion using analog and binary resistive networks," *Comput.*, vol. 21, pp. 52–63, Mar. 1988.
[19] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-29, pp. 1153–1160, Dec. 1981.
[20] S. Kirkpatrick, C. Gelatt, Jr., and M. Vecchi, "Optimization by simulated annealing," *Sci.*, vol. 220, pp. 671–680, May 1983.
[21] J. Konrad and E. Dubois, "Estimation of image motion fields: Bayesian formulation and stochastic solution," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, Apr. 1988, pp. 1072–1075.
[22] ——, "Multigrid Bayesian estimation of image motion fields using stochastic relaxation," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 1988, pp. 354–362.
[23] J. Konrad, "Motion-compensated interpolation for TV frame-rate conversion," Tech. Rep. 26, INRS-Télécommunications, Oct. 1988.
[24] ——, "Bayesian estimation of motion fields from image sequences," Ph.D. thesis, Dept. of Elect. Eng., McGill Univ., Montreal, June 1989.
[25] J. Konrad and E. Dubois, "Bayesian estimation of discontinuous motion in images using simulated annealing," in *Proc. Conf. Vision Interface VI'89* June 1989, pp. 51–60.
[26] ——, "Comparison of stochastic and deterministic solution methods in Bayesian estimation of 2-D motion," *Image Vision Comput.*, vol. 9, pp. 215–228, Aug. 1991.
[27] J. Marroquin, "Probablistic solution of inverse problems," Ph.D. thesis, Mass. Inst. Technol., Cambridge, Sept. 1985.
[28] D. Martinez, "Model-based motion estimation and its application to restoration and interpolation of motion pictures," Ph.D. thesis, Mass. Inst. Technol., Cambridge, Aug. 1986.
[29] 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.
[30] D. Murray and B. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 220–228, Mar. 1987.
[31] H. -H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 565–593, Sept. 1986.
[32] S. Ross, *Applied Probability Models with Optimization Applications*. San Francisco: Holden-Day, 1970.
[33] B. Schunck, "Motion segmentation and estimation," Ph.D. thesis, Mass. Inst. Technol., Cambridge, May 1983.
[34] Ø. Tretiak and L. Pastor, "Velocity estimation from image sequences with second order differential operators," in *Proc. IEEE Int. Conf. Patt. Recogn.*, July 1984, pp. 16–19.
[35] J. Woods, "Two-dimensional discrete Markovian fields," *IEEE Trans. Inform Theory*, vol. IT-18, pp. 232–240, Mar. 1972.



Janusz Konrad was born in Szczecin, Poland, on Nov. 15, 1956. He received the M.Eng. degree with distinction from the Technical University of Szczecin, Poland, in 1980 and the Ph.D. degree, also with distinction, from McGill University, Montreal, Canada, in 1989, both in electrical engineering.

In July 1989, he joined the Institut National de la Recherche Scientifique, University of Québec as a research associate at the INRS-Télécommunications Centre in Montreal, Canada, where he worked on transmission of HDTV images. In May 1991, he became an Assistant Professor at INRS-Télécommunications. He has collaborated with the Visual Communications Group of Bell-Northern Research. His interests are in the areas of multidimensional signal processing, computer vision, and image sequence transmission.

Eric Dubois (S'72–M'77–SM'85) received the B.Eng. (honors) degree with great distinction and the M.Eng. degree from McGill University, Montreal, Canada in 1972 and 1974, respectively, and the Ph.D. degree from the University of Toronto, Toronto, Canada, in 1978, all in electrical engineering.

He joined the Institut National de la Recherche Scientifique (University of Quebec) in 1977, where he currently holds the position of Professor in the INRS-Télécommunications Centre, Montreal, Canada. He has also been an Auxiliary Professor in the Department of Electrical Engineering, McGill University, since 1981. In 1986, he was on sabbatical leave with the Advanced Television Research Program at the Media Laboratory, Massachusetts Institute of Technology, Cambridge. His research has centered on the source coding and processing of still and moving images and in multidimensional digital signal processing. He has collaborated extensively with the Image Services Department of Bell-Northern Research, the Engineering Department of the Canadian Broadcasting Corporation, and the Communications Research Center, Ottawa, Canada. His current work is in enhanced and high-definition television systems and in motion estimation with application to television signal processing.

Dr. Dubois was the co-recipient of the 1988 Journal Award from the Society of Motion Picture and Television Engineeers. He is a member of the Order of Engineers of Quebec.