BAYESIAN ESTIMATION OF MOTION FIELDS FROM IMAGE SEQUENCES

by

Janusz Konrad

M.Eng.

Department of Electrical Engineering

McGill University

Montreal, Canada

June 1989

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

This is a penultimate version, which differs from the final version in a minor way.

Janusz Konrad

Rodzicom – Wiktorii i Leopoldowi,

oraz żonie – Edycie.

ABSTRACT

This thesis addresses the ill-posed problem of estimating two-dimensional motion in time-varying images. The approach proposed here uses the theory of stochastic processes at the formulation and solution stages. Independent Gaussian random variables are used to model the relationship between motion fields and images, and vector and binary Markov random fields are used to model motion and motion discontinuity fields, respectively. These models, combined using Bayes rule, result in Gibbsian *a posteriori* probability distribution from which the Maximum *A Posteriori* Probability (MAP) and the Minimum Expected Cost (MEC) estimation criteria are derived. Optimization of these criteria is performed using stochastic relaxation. The MAP estimation is extended to handle large displacements via hierarchical approach. Deterministic approximations to some of the stochastic algorithms proposed are derived and compared with their stochastic counterparts. The colour information is incorporated into the estimation process. Numerous experimental results are included. Two of the methods proposed are applied to motion-compensated interpolation and shown to reduce certain type of errors.

SOMMAIRE

Le présent mémoire examine le problème partiellement défini de l'estimation du mouvement bidimensionnel dans les séquences d'images dynamiques. L'approche proposée ici utilise la théorie des processus stochastiques aux étapes de la formulation du problème et de sa solution. Des variables aléatoires gaussiennes indépendantes sont utilisées pour modéliser la relation entre les champs de déplacement et les images, tandis que des champs aléatoires markoviens, vectoriels et binaires, sont utilisés pour modéliser respectivement le champ de déplacement et le champ de discontinuité du déplacement. Ces modèles, combinés en utilisant la règle de Bayes, résultent en une fonction de répartition de probabilité *a posteriori* de type Gibbs à partir de laquelle les critères d'estimation de Probabilité *A posteriori* Maximale (PAM) et de Coût Espéré Minimal (CEM) sont obtenus. L'optimisation de ces critères est accomplie par relaxation stochastique. A l'aide d'une méthode hiérarchique, l'estimation PAM est généralisée aux situations où les déplacements sont grands. Des approximations déterministes à certains des algorithmes stochastiques proposés sont obtenues et comparées aux méthodes stochastiques correspondantes. L'information couleur est incorporée dans le processus d'estimation. Plusieurs résultats expérimentaux sont inclus. Deux des méthodes proposées sont appliquées au problème de l'interpolation d'images avec compensation pour le mouvement et l'on constate qu'elles réduisent certains types d'erreurs.

ACKNOWLEDGMENTS

I am greatly indebted to my supervisor Professor Eric Dubois for his patient guidance throughout the course of my research. His door stayed always opened to me as well as to other students, and even very busy he always found time to discuss important (or less so) issues.

My thanks also go to Professor Amar Mitiche for inspiring discussions on various topics from computer vision, motion in particular, to Professors Michael Shalmon and Ravi Mazumdar for shedding light on some more difficult aspects of stochastic processes, and to the members of my Ph.D. committee: Professor Peter Kabal and Professor Martin Levine.

An important role has also been played by fellow student Claude Bergeron. Our heated discussions on motion estimation have resulted in many new ideas, and have significantly contributed to the algorithms developed here. Also, some of his software was invaluable for subjective evaluation of motion field quality. The experimental results would not have been possible without excellent computer facilities at INRS-Télécommunications, which are gratefully acknowledged. In particular I would like to thank Professor Peter Kabal for his guidance in the world of software, and to Dr. Vishwa Gupta of Bell-Northern Research for introducing me to programming the NMX-432 array processor.

I am also grateful to my supervisor, to INRS-Télécommunications and to McGill University for financial assistance over the last five years.

Finally, a special thanks goes to my parents Wiktoria and Leopold, who supported me from far away, and to my wife Edyta, who stayed right by me when I needed it most. Their love, patience and support enabled me to continue through the darkest moments when the light at the end of the tunnel was fading.

TABLE OF CONTENTS

LIST OF FIGURES

NOTATION

a	constant in annealing schedule
α	constant in a potential for single-element line clique
$\mathbf{b}$	*base* displacement field
$\dot{\mathbf{b}}$	initial *base* displacement field
β, $\beta_{\mathbf{d}}$, β_l	constants in Gibbs probability distributions
c, $c_{\mathbf{d}}$, c_l	cliques
$\mathcal{C}$, $\mathcal{C}_{\mathbf{d}}$, $\mathcal{C}_l$, $\mathcal{C}_h$, $\mathcal{C}_v$, $\mathcal{C}_{hv}$, $\mathcal{C}_{45}$, $\mathcal{C}_{-45}$	sets of cliques
$\mathbf{d}$	displacement field (sample)
$\boldsymbol{d}$	the true underlying displacement field
$\overline{\mathbf{d}}$	expectation of displacement random field $\mathbf{D}$
$\widehat{\mathbf{d}}$	estimate of the true displacement field
$\widehat{\mathbf{d}}^*$	optimal estimate of the true displacement field
$\mathbf{D}$	displacement random field
Δt	normalized distance between temporal positions of the displacement field and the preceding image field
g	observed image (data)
$\tilde{g}$	interpolated observed image at positions not on sampling grid
G	image random field
Γ, Γ^*	random fields in Markov chain Υ
γ, ϖ	sample fields in Markov chain Υ
$\mathbf{h}$	*incremental* displacement field
$\dot{\mathbf{h}}$	initial *incremental* displacement field
$\widehat{\mathbf{h}}$	estimate of *incremental* displacement field
$\mathcal{I}_k$	interpolation operator (from grid κ to grid $\kappa - 1$)
l	displacement discontinuity field (sample) or line field
$\boldsymbol{l}$	the true underlying displacement discontinuity field
$\widehat{l}$	estimate of the true displacement discontinuity field
$\widehat{l}^*$	optimal estimate of the true displacement discontinuity field
$\Lambda_{\mathbf{d}}$, Λ_g	lattice (sampling structure) of displacement field $\mathbf{d}$, image g
$\lambda_{\mathbf{d}}$, λ_g, λ_l	weighting constants for image, displacement and line energies
$M_{\mathbf{d}}$	number of vectors in one displacement field
M_g	number of pels in one image field
n, n_d	noise samples

N	noise random field
$\mathcal{N}$, $\mathcal{N}_{\mathbf{d}}$, $\mathcal{N}_l$, $\mathcal{N}_{(\mathbf{d},l)}$	neigbourhood systems
$\mathcal{N}^i$	i-th order neigbourhood system
ν_j	weights in gradient computation of three-component image
$\eta(\mathbf{x})$	neighbourhood at spatial position $\mathbf{x}$
Ω	state-space of Markov chain Υ
P	probability measure
p_{n_d}	Gaussian distribution of noise sample
π	Gibbs probability distribution
$\tilde{\mathbf{r}}$, $\tilde{\mathbf{r}}$	displaced pel difference for one and three components
$\mathbf{s}$	spatio-temporal position
$\mathcal{S}'_{\mathbf{d}}$, $\mathcal{S}'_g$, $\mathcal{S}'_l$	state-spaces of single displacement vector, image pel and displacement discontinuity
$\mathcal{S}_{\mathbf{d}}$, $\mathcal{S}_g$, $\mathcal{S}_l$	state-spaces of displacement, image and displacement discontinuity fields
σ^2	variance of noise term (DPD model)
t	temporal position (time)
t_-, t_+	temporal positions of the preceding and following image fields
T, T_0, T_f	temperature (parameter in simulated annealing)
$T^h_{\mathbf{d}}, T^v_{\mathbf{d}}, T_{\mathbf{d}}$	horizontal, vertical and temporal sampling periods for lattice $\Lambda_{\mathbf{d}}$
T^h_g, T^v_g, T_g	horizontal, vertical and temporal sampling periods for lattice Λ_g
τ	discretized time in the evolution of Markov chain Υ
τ_{60}	inter-field distance of 1/60 sec
Θ, θ	positive-definite functions in MEC estimation
u	the true underlying image
U, U_g, $U_{\mathbf{d}}$, U_l	energy functions in Gibbs probability distributions
$U^i_{\mathbf{d}}$, U^i_l	local energies used in the Gibbs sampler
Υ	Markov chain formed by sample vectors and line elements
$\mathbf{v}$	instantaneous velocity
V, $V_{\mathbf{d}}$, V_l	potential functions
φ	temperature schedule
$\mathbf{x}$	spatial position for image pel, displacement vector or line element
Ξ	transition probability matrix of Markov chain Υ
Z, $Z_{\mathbf{d}}$, Z_l	normalizing constants in Gibbs probability distribution

ψ_h, ψ_v	cosets (shifted lattices) of horizontal and vertical line elements
Ψ	union of cosets over which the displacement discontinuity field l is defined

Chapter 1 INTRODUCTION

1.1 MOTIVATION

This thesis addresses the problem of estimating 2-D motion from spatio-temporally sampled image sequences. Such a problem is one of the two aspects of the widely used term "motion estimation" in the field of computer vision. The other aspect is the so called 3-D motion, which aims at recovery of three-dimensional motion parameters of objects or of an observer. There exist various approaches to estimating 2-D motion from dynamic images, but they can usually 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, colour, contrast etc. The high-level methods, 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 these features to solve the correspondence problem. Instead of intensity or colour matching, contours or even items from a list are matched according to some pre-specified syntax.

1.1.1 Importance of 2-D motion estimation

The goal of 2-D motion estimation is to obtain a sequence of dense motion fields, also called *optical flow*, faithfully representing movements in a time-varying image. This estimation problem has a wide range of applications such as:

1. time-varying image processing: motion-compensated interpolation for sampling structure conversion, motion-compensated filtering for noise reduction,
2. image sequence compression: motion-compensated coding for bit rate reduction,
2. computer vision (robotics): structure from motion for passive navigation, 3-D motion from 2-D motion for passive navigation and object tracking.

1.1.2 Drawbacks of existing methods

Not surprisingly, since the problem is important, it is also difficult. The difficulty of estimating dense 2-D motion fields is due to two factors: ill-posedness and complexity. The problem is ill-posed since many different vector fields can explain the same data. The complexity of the problem is dependent on its dimensionality, which is high since typically several thousands of unknowns have to be computed simultaneously.

Simple estimation algorithms, like block matching, frequently fail to produce good results. Such methods rely on the data (images) only and do not attempt to explicitly model motion fields. Consequently, every motion vector is computed from local intensity values without any regard to the motion of its neighbouring picture elements. To overcome this deficiency global formulations have been proposed. Instead of minimizing a local objective function, global functions over the complete motion field have been used. Horn and Schunck [41] have formulated such a global criterion as a compromise between an error derived from the motion constraint equation and a motion smoothness error. Hildreth [39] has 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 [69] has 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 resulting cost functional is quadratic with respect to the estimated quantity, and is usually minimized either by establishing necessary conditions for

optimality and solving a set of linear equations (Gauss-Seidel relaxation in the Horn-Schunck approach), or by directly applying a general optimization method (conjugate gradients in Hildreth's approach). The major drawback of the Horn-Schunck method 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 too. Also, the formulation using the motion constraint equation requires the evaluation of data derivatives, which is an ill-posed problem itself. Finally, the method fails to work properly for large displacements as the assumption of locally linear variation of image intensity is violated, especially since the purely temporal derivative is used in the algorithm. These deficiencies frequently result in erroneous results, and in particular cause overestimation of velocities of high-contrast intensity discontinuities. Due to the space-invariant smoothing operation, the estimates also tend to be erroneous at the motion boundaries. The major drawback of Hildreth's approach is the need to know intensity contours e.g., edges, before performing motion estimation. Also it is not clear how to propagate the boundary estimates, especially if the contours are not closed. Nagel's extension suffers, like the Horn-Schunck algorithm, from fixed spatio-temporal positions of motion estimates, however he used more elaborate Beaudet operators to compute image derivatives. He also suggested a modification to the Horn-Schunck algorithm to avoid computation of temporal intensity derivative, although failed to give any justification for this modification. Nagel reported an improvement over the Horn-Schunck method due to the "oriented smoothness" (space-variant) operator, however his motion model disregards motion discontinuities and still causes oversmoothing at the motion boundaries.

1.1.3 Multidisciplinary problem

It is clear from the applications listed that 2-D motion estimation is a multidisciplinary problem. It may be used in time-varying image compression for video, in 3-D motion recovery for passive navigation as well as in object tracking for traffic control. This first application may be classified as a part of multidimensional digital signal processing, while the other two belong to computer vision and robotics. The area of motion estimation can benefit from theory developed in all these fields, however it is also constrained by them

in the sense that the goals to be achieved are specified by applications. For example, in video the image quality improvement is the ultimate goal, and velocity oversmoothing at a low-gradient intensity edge may not degrade it, while in robotics precise motion boundaries should be known to reliably recover structure from motion. Otherwise, a robot arm may miss its target, or even destroy it.

Another area with which motion estimation intersects in this thesis is the theory of stochastic processes, and in particular Markov random fields. Unlike the few proposed probabilistic methods of estimating motion (to be reviewed in Section 2.3.5), this thesis offers a broader view on stochastic modeling of motion and suggests stochastic solution methods to solve such stochastic formulations.

1.2 STOCHASTIC APPROACH TO MOTION ESTIMATION

In the field of applied stochastic processes, the Markov random field models have been successfully used for image modeling [92], [38]. Based on such models Geman and Geman [26] have proposed a theoretical basis for image restoration using *stochastic relaxation* methods, and have shown impressive results. Following the idea of Kirkpatrick *et al.* [52] they have also formalized *simulated annealing*, proving three important theorems.

Taking into account the success of Markov random field modeling of images as well as the drawbacks of the Horn-Schunck-type algorithms, this thesis approaches the 2-D motion estimation problem from a point of view of stochastic process theory. A Bayesian formulation of motion estimation is proposed, where the motion fields are modeled by a vector Markov random field, and are related to the images through independent Gaussian random variables. From these stochastic models a criterion for Maximum *A Posteriori* Probability estimation is derived. Since this criterion is a non-quadratic function of the estimates, and in general is multimodal, there exist many local minima and such methods as conjugate gradients, steepest descent etc., will not find the global minimum unless started sufficiently close to this global optimum. To locate the global minimum it is proposed in this thesis to use *simulated annealing* [26]. This method, under certain conditions, is able to find the global minimum regardless of the initial state. Two versions of *simulated annealing*

are derived: discrete state-space and continuous state-space. The discrete state-space algorithm is a straightforward implementation of the *Gibbs sampler* as proposed by Geman and Geman, and does not require data derivatives. It can be classified as a pel matching algorithm with a smoothness constraint. For the continuous state-space MAP estimation, however, it is possible to locally approximate the non-quadratic objective function by a quadratic one. This results in a Gaussian transitional probability (instead of Gibbsian) for which random deviates can be easily generated. This method belongs to a class of spatio-temporal gradient techniques with a motion smoothness constraint. It is demonstrated that this method is a stochastic generalization of Horn-Schunck-type algorithms. Also another Bayesian formulation, based on the idea proposed by Marroquin [62], minimizing an expected error, is investigated. Simplified to the Minimum Mean Squared Error estimation it is solved by *stochastic relaxation* and averaging.

To handle large displacements efficiently the MAP estimation is extended by incorporating a hierarchy of resolution levels. In this way the computational effort is reduced significantly. It is formally demonstrated for the 1-D space-invariant matching problem, that low-pass filtering of the data results in low-pass filtering of the objective function. Hence it can be concluded that multimodal objective functions can be "smoothed-out" to become unimodal if sufficient low-pass filtering is provided. Consequently, minimization becomes a trivial task, as is demonstrated on a 1-D example. This important result does not easily extend to the 2-D space-variant motion estimation, but the general idea can probably be extrapolated from the 1-D case, as seems to be confirmed by the practical examples.

The globally smooth motion fields are not sufficiently precise to describe motion in typical TV images. Rigid body motion is characterized by a piecewise smooth motion field, which includes discontinuities along smooth curves. A globally smooth field cannot accurately represent such properties. The problem is even more pronounced for the hierarchical approach, which tends to introduce oversmoothing at the motion boundaries. To accommodate a piecewise smooth description of motion fields, a motion model which explicitly allows discontinuities in the motion field is adopted via the so called *line process* (a coupled binary Markov random field interleaved spatially with the vector Markov random field). This two-layer model is shown to improve the estimates significantly.

Throughout the course of research I found that quite frequently the motion cue from luminance only is insufficient for reliable motion estimation, and that the results can be significantly improved by incorporating other cues. Colour is investigated as such an additional cue in the MAP context, and is shown to improve the estimates in some situations.

Since the stochastic motion estimation techniques proposed in this thesis are characterized by increased computational effort, faster deterministic solution methods were also investigated. Instead of the discrete state-space MAP estimation, the *iterated conditional modes* proposed by Besag [11] can be used. However this approximation is clearly inferior. If instantaneous reduction of a temperature parameter to zero is applied (*quenching*) to the continuous state-space MAP estimation, this method results in an update resembling the Horn-Schunck approach with certain modifications. Those modifications are discussed in detail, and their impact on the estimation process is demonstrated. Simulations performed show superiority of *stochastic relaxation* compared to deterministic Horn-Schunck-type methods for critical sequences. However, for less demanding (or ambiguous) data the difference between the methods is significantly reduced, sometimes even unnoticeable. This observation suggests that for more "bumpy" objective functions, the deterministic algorithm is unable to escape local minima, while *simulated annealing* avoids them very skillfully. The difference between estimates produced by the stochastic and deterministic counterparts is reduced when they are implemented in a hierarchical manner. Clearly, a smoother objective function incorporates fewer local minima, hence even a deterministic algorithm has a better chance of finding the global minimum.

1.3 THESIS OVERVIEW

In Chapter 2 some definitions and assumptions are presented, followed by the discussion of the ill-posed nature of motion estimation and a survey of motion estimation methods. Then, the hierarchical and regularization methods used in image processing and computer vision are discussed, as well as the stochastic modeling and estimation. The chapter is concluded with an example of 1-D space-invariant signal matching to demonstrate behaviour of some optimization methods used in this thesis.

Chapter 3 introduces the terminology, briefly summarizes Markov random fields and *Gibbs* distributions, and proposes the Maximum *A Posteriori* Probability and the Minimum Expected Cost estimation criteria. Subsequently, the necessary ingredients of such formulation: the *structural model*, the *observation model* and the *displacement field model*, are discussed, and the final form of the *a posteriori* probability is derived. Appendix 3.A contains a derivation of the Minimum Expected Cost estimator.

In Chapter 4 the stochastic solutions to earlier formulations are presented. First, the *Monte Carlo* methods are briefly described, and then more detailed analysis of *stochastic relaxation* via the *Metropolis algorithm* and via the *Gibbs sampler* is included. This is followed by the discussion of *simulated annealing* to obtain the Maximum *A Posteriori* Probability estimator and of the *Law of Large Numbers for Markov chains* to obtain the Minimum Expected Cost estimator. Then, the continuous state-space *Gibbs sampler* is derived, and the issue of image intensity interpolation is discussed. The chapter is concluded with numerous experimental results obtained from various image sequences. In appendices at the end of the chapter proofs related to both *stochastic relaxation* algorithms as well as the discussion of intensity interpolation are presented.

Chapter 5 extends the Maximum *A Posteriori* Probability estimation to a hierarchy of resolutions. First, the relevance of filtering in a hierarchical approach is discussed, and then the algorithm is described. Subsequently the filters for pyramid generation are designed, and the adjustment of some parameters according to the resolution level is presented. Again the chapter ends with experimental results. Appendix 5.A includes the proof of a theorem relating filtering operations on the data and on the objective function.

Chapter 6 introduces the piecewise smooth model for motion. First, a new Maximum *A Posteriori* Probability criterion is formulated, and then, based on the two-layer model, the *a posteriori* probability is derived. Also the *Gibbs sampler* for the *line process* is discussed. As usual the chapter concludes with numerous examples.

In Chapter 7 colour is incorporated into the structural and observation models. The objective function is derived, and some experimental results shown.

Chapter 8 brings deterministic approximations to the Maximum *A Posteriori* Probability estimation for the discrete and continuous state-spaces. Theoretical and practical

comparisons with the stochastic counterparts are carried out.

In Chapter 9 two algorithms are applied to motion-compensated interpolation. Their performance is compared subjectively via visual inspection and objectively – through an error criterion.

Finally, Chapter 10 summarizes this thesis, discusses the contributions and indicates some open questions.

Chapter 2 BACKGROUND

As stated in the introduction, this thesis addresses the problem of estimating a 2-D motion of picture elements in a spatio-temporal (3-D) sequence of images. It is not the intention of this work to investigate the recovery of the motion in the original 3-D scene.

In this chapter some basic definitions and frequently used assumptions will be given, ill-posedness of 2-D motion estimation will be demonstrated, and the major motion estimation techniques will be reviewed. Then, techniques proposed to deal with ill-posedness will be described, followed by a brief discussion of hierarchical methods as well as of stochastic modeling and estimation. The chapter will be concluded with a simple example of 1-D signal matching to illustrate the difficulties associated with minimization of multimodal functions.

2.1 DEFINITIONS

In order to review the existing motion estimation techniques, some frequently used assumptions and basic definitions are given below.

Images, which are the input for motion estimation, are usually formed by the projection of a three-dimensional scene onto an image plane. It is clear that any perceived motion in the image can be caused by some true motion (in the scene or between the observer and the scene), or by a changing illumination of the scene. This perceived motion is called *optical flow*, and is defined by Horn [42] as *"the apparent motion of the image brightness pattern"*. This term is widely used in the computer vision literature. Since it is very difficult to distinguish between the intensity (brightness) changes due to the true motion and due to illumination effects, constant scene illumination is often assumed. With this assumption,

the perceived changes in the image are due only to some true motion, and consequently a fixed point on a moving object should ideally have the same intensity in subsequent images, unless it disappears (leaves the field of view or is occluded). Hence, it is frequently assumed that image intensity along the motion trajectories is constant.

The 2-D vector field $\mathbf{v}(\mathbf{x},t)$ of instantaneous velocities of points in the image plane as a function of spatial position $\mathbf{x}=[x,y]$ and time t, is called a *velocity field*. It is defined everywhere in the image plane except at the occluding motion boundaries (discontinuities). Given a spatio-temporal position $(\mathbf{x},t)$, $\mathbf{v}(\mathbf{x},t)$ is a vector consisting of two components: $\mathbf{v}(\mathbf{x},t)=[v_x(\mathbf{x},t),v_y(\mathbf{x},t)]$ (note that $\mathbf{v}$ is a row vector). For images sampled in the temporal direction (sampling interval T), the concept of the velocity field is replaced by that of the *displacement field* $\mathbf{d}(\mathbf{x},t)$, which is defined as a mapping of image points from the image at time t into the corresponding image points at time $t+T$. The displacement field is defined as well only for points also visible in the next frame. The term *motion field* will be used to refer to either a velocity field or a displacement field.

Let $f(\mathbf{x},t)$ denote the image intensity as a function of space and time. Then, the assumption of constant image intensity along the motion trajectories can be formally expressed as the zero directional derivative of $f(\mathbf{x},t)$ in the direction $\mathbf{z}$:

$$\frac{df(\mathbf{x},t)}{ds}=\mathbf{z}\cdot\nabla f=0 \tag{2.1}$$

where $\mathbf{z}=[\mathbf{v}(\mathbf{x},t),1]/\sqrt{\|\mathbf{v}(\mathbf{x},t)\|^2+1}$ is a unit vector in the direction of motion and $\nabla=[\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial t}]^T$ is a spatio-temporal gradient (note that ∇ is a column vector). Substituting for $\mathbf{z}$ in the above equation, the *motion constraint equation* [41], [73] can be obtained:

$$\mathbf{v}(\mathbf{x},t)\cdot\nabla_{\mathbf{x}}f+\frac{\partial f}{\partial t}=0 \tag{2.2}$$

where $\nabla_{\mathbf{x}}=[\frac{\partial}{\partial x},\frac{\partial}{\partial y}]^T$ is a spatial gradient. Equation (2.2) is widely used in derivation of various motion estimation algorithms.

For images sampled in time ($t=kT$ for some integer k) it is more natural to describe motion in terms of a displacement field $\mathbf{d}(\mathbf{x},t)$. Define the *displaced frame difference* as

$$\tilde{\mathbf{r}}_f(\mathbf{d},\mathbf{x},t)=f(\mathbf{x}+\mathbf{d}(\mathbf{x},t),t+T)-f(\mathbf{x},t)\qquad \text{all } (\mathbf{x},t). \tag{2.3}$$

Let $\boldsymbol{d}(\mathbf{x},t)$ be the true motion field for some given images. $\boldsymbol{d}$ is a continuous function of spatial position $\mathbf{x}$ and a discrete function of temporal position $t=kT$. Under the

assumption of constant intensity along the motion trajectory $\boldsymbol{d}$, the following relationship holds

$$\tilde{r}_f(\boldsymbol{d}, \mathbf{x}, t) = 0 \qquad \text{all } (\mathbf{x}, t). \tag{2.4}$$

This equation can also be used in deriving motion estimation algorithms.

A second order assumption with respect to the intensity has been used to estimate motion, as well [88], [9]. If the spatial gradient of intensity is constant along the motion trajectories, then the directional derivative of this gradient should be zero

$$\frac{d(\nabla_{\mathbf{x}}^T f)}{ds} = \mathbf{z} \cdot \nabla(\nabla_{\mathbf{x}}^T f) = 0. \tag{2.5}$$

The vector equation above can be rewritten as

$$\mathbf{v} \cdot \nabla_{\mathbf{x}}(\nabla_{\mathbf{x}}^T f) + \frac{\partial}{\partial t}(\nabla_{\mathbf{x}}^T f) = 0 \tag{2.6}$$

where $\nabla_{\mathbf{x}}(\nabla_{\mathbf{x}}^T)$ denotes a (spatial) Hessian matrix:

$$\nabla_{\mathbf{x}}(\nabla_{\mathbf{x}}^T) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial x \partial y} \\ \frac{\partial^2}{\partial x \partial y} & \frac{\partial^2}{\partial y^2} \end{bmatrix}. \tag{2.7}$$

Since (2.6) is a vector equation, it provides constraints for both vector components, hence should not suffer as severely from the aperture problem [39] as the scalar equation (2.2).

2.2 ILL-POSED NATURE OF MOTION ESTIMATION

The estimation of 2-D motion from a sequence of images is an inverse problem. A 2-D motion in an image, whether it is a consequence of a projected 3-D motion onto the image plane or it is just a 2-D rearrangement of positions of intensity patterns, modifies the image content over time. The goal is to "undo" that operation and to recover the action which had produced that effect. Consequently, this recovery operation is an inverse problem, and not surprisingly has been identified as *ill-posed*.

According to Hadamard's definition, a *well-posed* problem is characterized by the following properties:

1. *existence*: for each data there exists a solution,
2. *uniqueness*: the solution is unique,
3. *continuity*: the solution is related to the data in a continuous manner.

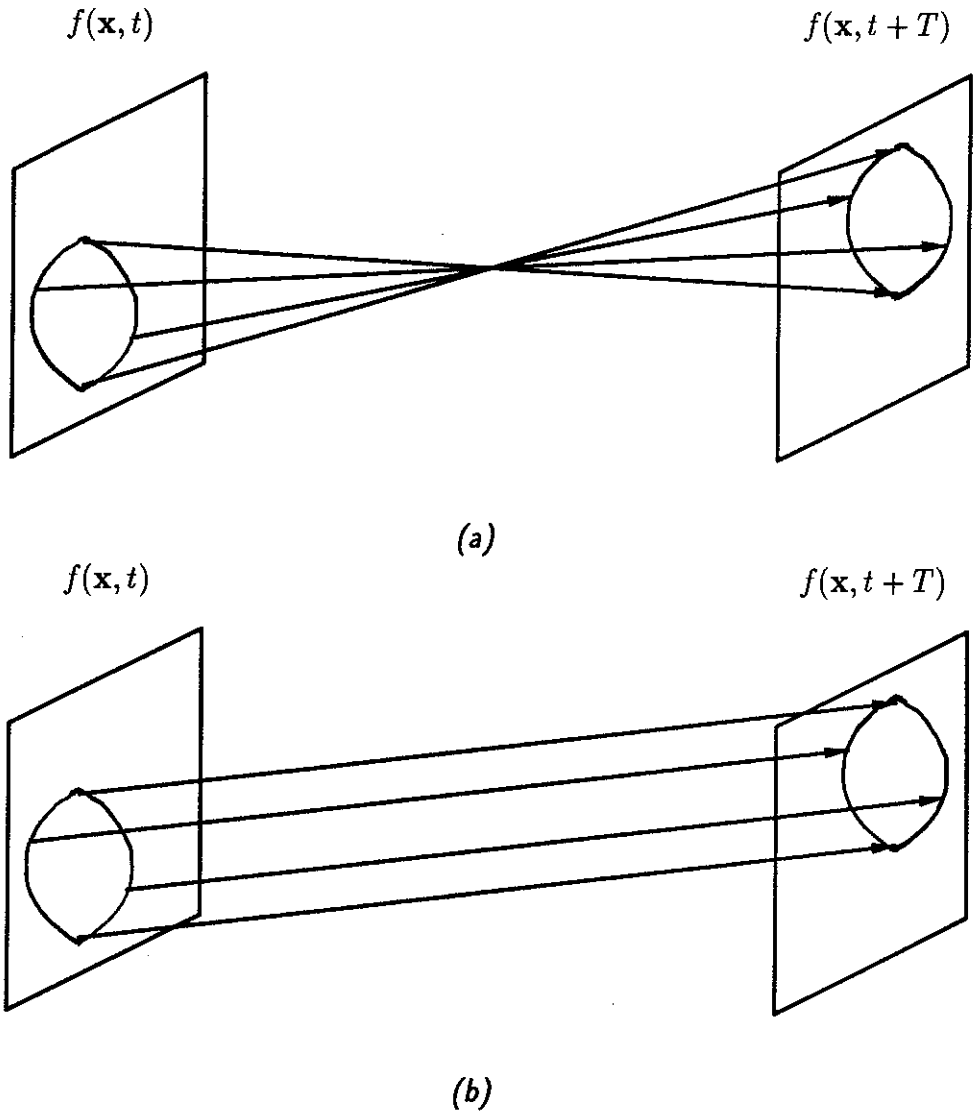


Fig. 2.1 Example of ill-posed nature of motion estimation (non-uniqueness): complex motion and possible deformation (a), and simple translational motion (b). Our experience tells us that example (b) agrees better with what we expect.

Violation of any of the above conditions makes a problem *ill-posed*. Clearly, motion estimation is ill-posed, because:

1. for the data containing occlusion areas there is no solution in such areas (violation of existence),
2. for given data there are many different motion fields satisfying the data (violation of uniqueness, Fig. 2.1),
3. for some intensity structures even slight local modification of this intensity may cause significant change in the computed vector length and/or orientation (violation of continuity).

The non-uniqueness of motion recovery is illustrated in Fig. 2.1. The image is simply an intensity surface with one iso-luminance contour shown. Assuming unchanged intensity between the images, Fig. 2.1.a shows one possible set of motion vectors explaining the data, while Fig. 2.1.b shows another one. Both types of motion explain accurately the data (intensities), however the field in Fig. 2.1.b seems to agree far better with our expectation of actual action in this image. Our experience tells us that things move coherently and most frequently those moving things are rigid bodies, hence we expect neighbouring points on the same object to move with similar velocities in similar directions. Concluding, we expected translational motion of the surface (Fig. 2.1.b) rather than complex motion with possible deformation as in Fig. 2.1.a.

2.3 SURVEY OF MOTION ESTIMATION METHODS

There exist two classes of motion estimation algorithms: those which extract 3-D motion parameters from a sequence of 2-D projections, and those which estimate dense velocity (displacement) fields disregarding any relationship between the objects and the camera. In the first case, a rigid body motion is usually assumed, which can be decomposed into translation, rotation and zooming components, and relatively few motion parameters are estimated from an overdetermined set of constraints. This class of methods will not be examined here, since the goal of this work is to obtain dense velocity fields, and secondly the typical TV imagery is too complex to be closely approximated by the simple motion models considered there.

The following methods, belonging to the second class, will be briefly discussed in the next few sections: histogram-based, transform-domain, matching and spatio-temporal gradient.

2.3.1 Histogram-based methods

One of the early motion estimation techniques was the gradient intensity transform method developed by Fennema and Thompson [23]. Based on the gradients estimated using the Sobel operator, they computed for each image point a constraint curve similar to

the motion constraint line (2.2), but positioned in the polar coordinates. The true velocity vector should be situated at the intersection of all constraint curves evaluated for image points belonging to a moving object. Since in practice there are numerous intersections, they partition the polar velocity space into velocity cells and accumulate a count number of motion constraint curve intersections with each velocity cell. Once all the image points have been analyzed, the peak value in the histogram identifies the velocity estimate.

This technique works quite well for a single object undergoing a simple translation, but it is not well suited for other motions (e.g., rotation), and may also present difficulties for images with multiple objects. The problem becomes especially difficult when several smaller objects are translating with the same velocity. From the histogram peak it will be impossible to distinguish such a case from a single large object, with its area equal to the total area of the small objects, moving with the same velocity. The technique of Fennema and Thompson uses local constraints in the global decision, which can be viewed as a bottom-up process. Once a decision is made there is no top-down feedback to use this global information in the local constraints reevaluation. The local structure of the constraints is lost, hence inability to tell the difference between several small and one large object undergoing the same translation.

Schunck [80] improved the above approach by clustering constraint lines for a given spatial area of the image. He performed 1-D cluster analysis to exclude contributions from across the motion boundaries. His technique can distinguish multiple objects and the background needs not be stationary. He presented some results but only for moving "random dots" pattern.

2.3.2 Transform-domain methods

Unlike other methods described in this chapter, which operate in the signal domain, the transform-domain techniques operate in a different domain. If a transformation is such that motion in the image produces characteristic modification of the image transform, then this motion can be detected and sometimes evaluated quantitatively in the transform domain.

The Fourier transform is used most frequently, and its shift property is the basis of the *Fourier-phase* method [37], [43]. Suppose $F(\omega_x, \omega_y)$ is the Fourier transform of the image

$f(x,y)$. Then, if this image undergoes a uniform translation $[d_x, d_y]$,

$$f(x-d_x, y-d_y) \Longleftrightarrow F(\omega_x, \omega_y) \cdot e^{-j2\pi(\omega_x d_x + \omega_y d_y)}.$$

The phase difference between the Fourier transforms of $f(x,y)$ and $f(x-d_x, y-d_y)$ evaluated over a number of frequencies results in an overconstrained system of linear equations, which can be solved for example by the least squares method. In practice this method will work only for single objects moving across a uniform background. Moreover, the positions of pixels with the obtained velocity vector are not known, hence the assignment of the velocity to an object must be performed in some other way. Also, care must be taken of the non-uniqueness of the Fourier phase function since the integral multiples of 2π are indistinguishable.

Another approach is the *spatio-temporal frequency* method originated by Gafni and Zeevi [24], [25], and generalized by Jacobson and Wechsler [46]. If a time-varying image $f(x,y,t)$ is uniformly translating with some constant velocity $[v_x, v_y]$, then

$$f(x,y,t) = f(x-v_x t, y-v_y t, 0) = f(x,y,0) * \delta(x-v_x t, y-v_y t, 0),$$

where δ is the Dirac delta function and "$*$" denotes convolution. It can be shown that the Fourier transform of $f(x,y,t)$ is zero everywhere in the $(\omega_x, \omega_y, \omega_t)$ space except the plane:

$$\omega_x v_x + \omega_y v_y + \omega_t = 0.$$

For each velocity (v_x, v_y) a *velocity polling function* is computed by integrating over different planes in the spatio-temporal frequency space. The velocity corresponding to the maximum of the polling function is chosen as an estimate. Obviously this approach is still limited to estimation of a single velocity, however Jacobson and Wechsler have proposed the use of the spatio-temporal/spatio-temporal frequency representation via the Wigner distribution instead of the spatio-temporal frequency representation via the Fourier transform. They claim that dense velocity fields can be obtained from this representation, but no examples of such fields are given in their work. The viability of the method is also questionable when discretized images have to be considered (3-D transform based on only two images ?).

The third type of transform-domain methods is the *phase correlation*, frequently used in image registration. The idea is to first perform a 2-D Fourier transform of each image, then

multiply together their corresponding frequency components, and take an inverse Fourier transform. The result, called a *correlation surface*, is nothing else but an outcome of full image matching performed in an efficient way †. The method works very well for translational motion, gives sub-pixel accuracy and is applicable to the estimation of large displacements (tens of pixels). Multiple moving objects can be also handled by this method, however the problem of assigning velocity estimates to spatial positions has to be circumvented in a different way e.g., as proposed by Thomas [86]. After identifying possible velocity estimates as n highest peaks on the correlation surface, he uses those candidates to perform local matching operations (e.g., block matching) to choose the locally best candidate. This method is a good example of a bottom-up data aggregation (Fourier transform) for some global decision process, combined with top-down feedback (block matching using globally computed candidates) for reexamination of the local constraints. Unfortunately, the method does not work well for motions departing from simple translation e.g., rotation, zoom or elastic body motion. This promising variation of the transform-domain approach suffers also from drawbacks characteristic for the block matching (Section 2.3.3.1).

2.3.3 Matching algorithms

Matching techniques have been designed to solve the short- and long-range *correspondence problem*, which associates certain structures in one image with corresponding structures in subsequent images. Various, image-dependent or image-independent, structures can be used. At first, these structures are identified in a reference image, and then an organized search for the corresponding structures is performed in the following images. Usually this search attempts to optimize some criterion in order to find the best match.

Since practically every motion estimation algorithm involves matching explicitly or implicitly, it is not clear how to classify some methods (e.g., Paquin and Dubois [73]). In this review it will be assumed that any method employing searching to derive the optimum correspondence between two (or more) images is a matching technique.

† Note that the sequence of operations given above is equivalent to a convolution of two images, which can be also viewed as a cross-correlation.

2.3.3.1 Region matching

A simple image-independent structure which can be used to solve the correspondence problem is a fixed size block of points. Basic assumption in such block matching algorithm is that motion is locally translational and slowly varying. The motion vectors are computed over a dense grid (e.g., identical with the image grid) or are assumed constant over a block (e.g., equal to the image block) when only a single vector is estimated for each block. Then, an optimization problem, based on some objective criterion, is formulated with respect to a displacement estimate. As such an objective criterion the following functions have been used:

1. correlation function (maximization) [8] †,
2. mean square error (minimization) [48],
3. mean of the absolute frame difference (minimization) [82],
4. thresholded absolute frame difference (minimization) [72].

The optimization problem can be solved by performing a very simple exhaustive search: computation of the objective function for every possible vector from its state-space, and choosing the one which offers the best value of the objective function. This inefficient approach finds the global optimum (within the given state-space) of this local optimization problem. To speed-up the search procedure other methods have been proposed:

1. 2D-logarithmic search [48],
2. three-step search [53],
3. modified conjugate direction search [82].

These techniques assume monotonicity of the objective function which in general is not true. For small blocks, however, this function may frequently be unimodal.

An interesting approach to correlation matching has been proposed by Anandan [3], [4]. He applied local analysis of the correlation surface by using principle axis decomposition, which resulted in high and low confidence direction estimates. He also proposed *heuristic* confidence measures, which together with the principle directions were later used in formulating the objective function to be minimized (velocity constraint error and velocity smoothness error).

† Comparative analysis of various correlation measures for motion analysis can be found in [14].

Block matching is a very simple technique to implement and relatively fast. It becomes much slower, however, when sub-pixel accuracy e.g., via bilinear or biquadratic interpolation, is required. In the case of a single vector estimate per block, it produces motion vector variations only at the block boundaries. To reduce this effect and to simultaneously more effectively deal with the motion boundaries, the displacement block size can be reduced. In the limit, when vectors are assigned on a pel by pel basis, this method becomes prohibitively expensive and rather than performing a search for a matching block, minimization techniques are used, and instead the method should be classified as a spatio-temporal gradient technique (e.g., [71] in Section 2.3.4).

Block matching is usually considered a suitable method for the short-range correspondence, while it is inappropriate for the long-range one. This can be explained by the fact that as the state-space for each block vector grows, the number of local minima grows as well, and the more efficient search methods frequently fail to attain the global minimum while the exhaustive search becomes too costly.

From the point of view of image coding this method may be sufficient as it attempts to minimize a prediction error in the direction of motion, but the resulting estimate may have little in common with the true motion in the image which is crucial for applications like motion-compensated interpolation.

2.3.3.2 Feature matching

The block matching approach does not take particular advantage of the image regions with high information content, but uses all the available data equally. Analysis of the human visual system suggests that the regions of high information content very significantly contribute to the motion perception. Thus, the use of more complex structures than blocks of points, seems promising. One of the first attempts in this direction was the work by Potter where he attempted to match simply intensity discontinuities [76] and templates [77], and also by Aggarwal and Duda [1] who used polygon vertices for matching. Also other high level features such as edges, termination points, corners [7], [58] or even complete image regions [1] have been used to solve the correspondence problem. Being based on the search for characteristic (and usually distinct) features of the image, feature matching is

suitable for the long-range correspondence problem.

Feature matching is a much more complex technique than block matching, since it heavily borrows from scene analysis (segmentation, feature extraction etc.). In fact, only schemes for analysis of relatively simple images have been proposed so far.

2.3.4 Spatio-temporal gradient methods

An important alternative to the techniques presented above are the *spatio-temporal gradient* methods. This class of techniques relies on the relationship between the spatial and temporal image intensity gradients.

The first attempts to use such gradients for motion computation were presented by Limb and Murphy [61], and Cafforio and Rocca [17]. Both have used a ratio of finite frame differences versus finite element differences to calculate the velocity estimate. In later work Netravali and Robbins [71] have minimized a displaced frame difference, which after linearization around some initial displacement estimate resulted in an iterative update algorithm. The direction of this update (for pel recursive algorithm) coincides with the negative direction of the gradient of their objective function (displaced frame difference), hence their approach can be classified as a steepest descent method. As the initial displacement estimate at each spatial location they used a horizontally preceding displacement, and this initial estimate was assumed to be fixed over a block for the displaced frame difference computation. An improvement to this method has been proposed by Paquin and Dubois [73]. They formulated a minimization problem, where the objective function was a sum (over a certain volume) of the square of linearized displaced pel differences. They obtained an iterative scheme which differs from the result of Netravali and Robbins in the choice of the initial estimate (temporal instead of horizontal predecessor) and its variation over the summation volume (variable instead of constant). Hence, the method provides displacement estimates at every spatial location and not a single one per block of pels.

The important work of Horn and Schunck [41], which will be examined more closely because of its relationship with some methods proposed in this thesis, has produced explicitly the motion constraint equation in its scalar form (2.2), which is a direct consequence of the constant image intensity assumption along the motion trajectories. In their paper they gave

the geometrical interpretation of the motion constraint equation, which is a constraint line in velocity coordinates. Since an overdetermined set of local constraints frequently fails to give a unique or any solution, they minimized a constraint error. The scalar equation (2.2), however, can produce only the velocity component along the local intensity gradient. To overcome this limitation they augmented their original objective function (motion constraint error) with another function reflecting the motion field smoothness. They formulated the following continuous minimization problem:

$$\min_{\mathbf{v}} \int\int (e_m^2(\mathbf{v}) + \lambda \cdot e_s^2(\mathbf{v}))dx\,dy, \tag{2.8}$$

where the motion constraint error e_m is the left-hand side of (2.2):

$$e_m(\mathbf{v}) = \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{\partial f}{\partial t} = \frac{\partial f}{\partial x} v_x + \frac{\partial f}{\partial y} v_y + \frac{\partial f}{\partial t}, \tag{2.9}$$

and the motion smoothness error is the sum of squared magnitudes of gradients of the velocity components:

$$e_s^2(\mathbf{v}) = \|\nabla_{\mathbf{x}} v_x\|^2 + \|\nabla_{\mathbf{x}} v_y\|^2 = (\frac{\partial v_x}{\partial x})^2 + (\frac{\partial v_x}{\partial y})^2 + (\frac{\partial v_y}{\partial x})^2 + (\frac{\partial v_y}{\partial y})^2. \tag{2.10}$$

The dependence of the intensity function on $(\mathbf{x}, t)$ and of the velocity function on $\mathbf{x}$ has been omitted for clarity of notation. The double integral in (2.8) extends over the entire image, and the parameter λ allows a balance between the impact of the data structure (through e_m) and the smoothness of motion field (through e_s) on the velocity estimates. Note that (2.8) is a global minimization problem, where the objective function is formulated over the entire image and entire motion field. At the time it was a completely new perspective on motion estimation.

From the necessary condition for optimality of the objective function in (2.8), Horn and Schunck have derived a set of two equations for each $\mathbf{x}$, and after discretization of f and $\mathbf{v}$ have used deterministic relaxation (Jacobi, Gauss-Seidel) to solve that linear system. The relaxation algorithm resulted in the following iterative update:

$$\mathbf{v}_{(i,j)}^{n+1} = \overline{\mathbf{v}}_{(i,j)}^{n} - \frac{\varepsilon_{(i,j)}(f, \overline{\mathbf{v}}^n)}{\mu_{(i,j)}(f)} \cdot \nabla_{(i,j)}^T f, \tag{2.11}$$

where the subscript (i,j) denotes a discretized spatial position $\mathbf{x}$, $\overline{\mathbf{v}}_{(i,j)}$ is a velocity vector averaged over some neighbourhood of (i,j) [41], superscript n denotes the iteration

number, and $\nabla_{(i,j)}$ is a discretized spatial gradient defined as

$$\nabla_{(i,j)} f = [f^x_{(i,j)}, f^y_{(i,j)}]^T,$$

where $f^x_{(i,j)}$ and $f^y_{(i,j)}$ are finite-difference approximations of the horizontal and vertical derivatives of f, respectively. The scalars $\varepsilon_{(i,j)}(f, \overline{\mathbf{v}}^n)$ and $\mu_{(i,j)}(f)$ are defined as follows:

$$\begin{aligned} \varepsilon_{(i,j)}(f, \overline{\mathbf{v}}^n) &= \overline{\mathbf{v}}^n_{(i,j)}) \cdot \nabla_{(i,j)} f + f^t_{(i,j)}, \\ \mu_{(,i,j)}(f) &= \lambda + \|\nabla_{(i,j)} f\|^2, \end{aligned} \tag{2.12}$$

where again $f^t_{(i,j)}$ is a finite-difference approximation to the temporal derivative of f.

Careful examination of the iterative update (2.11) and the scalars (2.12), will reveal that the solution can be rewritten as:

$$\mathbf{v}^{n+1}_{(i,j)} = \mathbf{v}^n_{(i,j)} - \frac{1}{2} \nabla^T_{\mathbf{v}^n_{(i,j)}} e_s^2(\mathbf{v}^n_{(i,j)}) - \frac{1}{2\mu_{(i,j)}(f)} \nabla^T_{\overline{\mathbf{v}}^n_{(i,j)}} e_m^2(\overline{\mathbf{v}}^n_{(i,j)}),$$

where $\nabla_{\mathbf{v}}$ is a gradient with respect to velocity $\mathbf{v}$. Note that with each iteration the solution advances in the direction of the gradient of the squared smoothness error and then in the direction of the scaled gradient of the squared constraint error (with respect to the average velocity). It is not exactly the steepest descent method, but it may be viewed as a two-step separable steepest descent.

A similar approach has been proposed by Hildreth [39]. As the motion constraint error e_m she used the difference between the measured velocity component orthogonal to the intensity contour and its estimate, while as the smoothness error e_s she used a velocity variation measure (e.g., gradient) along an intensity contour. Thus, the smoothness is enforced directionally along the intensity contour, and not homogeneously in spatial coordinates. Similarly formulated unconstrained minimization problem she solved by the method of conjugate gradients.

The relative success of Horn and Schunck's approach spawned a significant amount of interest in the global formulations. Nagel in his early work [68] has minimized a sum of squares of a quadratic approximation (involving second-order derivatives of f) of the displaced pel difference $\tilde{r}_f$. Later [69] he augmented this approach with a smoothness constraint based on the notion of the *gray value corner*. Unlike Horn and Schunck, he proposed to vary the smoothness constraint based on the characteristics of the data structure

using second-order derivatives of f. In his recent work [70] he argued about the redundancy of second-order derivatives in approximating $\tilde{r}_f$ (e_m term) and in weighting various components of the smoothing functional (e_s term). His oriented smoothness implemented through a weight matrix in the smoothing functional seems to contain relevant information about the data structure, and has a significant impact on the estimates around intensity discontinuities.

Cornelius and Kanade [18] have proposed to relax the intensity constancy assumption (along motion vectors) and added a space-varying offset to the constraint error. Gennert and Negahdaripour [28] have extended that by adding a constant intensity multiplier, and thus allowing the image pattern to vary linearly across the image. Both the offset and the multiplier were constrained by a smoothness operator, and were estimated simultaneously with the displacements. Also Krause [59] did not rely on the intensity constancy, but minimized a norm of a derivative of the intensity gradient (left-hand side of equation (2.6)). The second-order intensity derivatives resulting from differentiation of his objective function, have been estimated from a 2-D low-order polynomial fitting to the actual intensity.

A different approach to the local velocity estimation has been proposed by Tretiak and Pastor [88]. They have also relaxed the constant intensity assumption, but instead required constant derivative of the intensity gradient (2.5). Note that this is a vector equation, which results in a vector version of the motion constraint equation (2.6). Since the vector equation (2.6) provides two scalar equations, and since there are two unknowns v_x, v_y, this linear system is not underconstrained any more, and the aperture problem so characteristic of the scalar equation (2.2) is significantly reduced. This can be explained by the fact that the determinant of the Hessian matrix (2.7) is equal to gaussian curvature of the intensity surface at given t, which is large when local intensity contrast is large in all directions (a good candidate for velocity estimation).

2.3.5 Statistical approach to motion estimation

The motion estimation methods presented above were based on the deterministic relationships between the estimators (motion fields) and the observed data (images), and also

assumed no probabilistic models for the estimators themselves. Methods have also been developed approaching the problem statistically.

Schunck [80] proposed the maximum likelihood estimation of velocity vectors. As the likelihood function he used the probability of 2 line constraint measurements (orientation and distance from the origin) conditioned on the direction of motion. He derived an expression for this function assuming a uniform probability density of the constraint line orientation given the motion direction, but failed to precisely specify the velocity probability density. He proposed to use a set of measurements (assuming their independence), and to maximize the log-likelihood function using histogram techniques. The estimated direction permits computation of the velocity from the motion constrained equation. Schunck did not demonstrate any experimental results.

Also Martinez proposed to maximize the joint likelihood of motion and local image parameters [64]. From an additive white Gaussian noise model for observations he derived a Gaussian likelihood function, which is quadratic in image parameters but non-quadratic in motion vectors. He proposed to alternately estimate the image parameters (solution of a linear system) and the motion vectors (steepest descent method). He reported good results obtained with this method, however at high computational expense.

2.4 HIERARCHICAL METHODS IN IMAGE PROCESSING AND COMPUTER VISION

It has been an observation of many researchers that the performance of motion estimation algorithms can be significantly improved by pre-filtering of images. This observation can be explained by the following interactions:

1. block matching: the displaced frame difference (2.3) evaluated over a block is usually a multimodal function with respect to the displacement i.e., apart from the deepest valley in such a surface (global minimum for this local problem) there might be several less deep minima; smoothing of the image will result in smoothing of this surface too (proof for 1-D matching problem will be given in Appendix 5.A); with sufficient low-pass filtering this surface will become locally unimodal and any minimization method will attain the global minimum of this new optimization problem,
2. spatio-temporal gradient method: an important assumption of locally linear variation of the intensity function, which is used by gradient methods relying

on the motion constraint equation (2.2), is violated when the data structure is characterized by relatively high frequency content such as very sharp edges, periodical patterns with small repetition period, noise; smoothing reduces this high frequency content, and also suppresses noise, so that the data is closer to a locally linear behaviour; related is the problem of reliable estimation of the intensity derivatives – suppression of high frequencies and noise makes derivative estimation a better-conditioned problem [87].

The above observations have led to the interest in hierarchical algorithms which perform estimation over a pyramid of increasing image resolutions. The early examples of hierarchical processing are from other fields than optical flow estimation. The pyramidal representation was used for template matching [78] and image registration [91]. Terzopoulos [83] investigated the problem of surface reconstruction from sparse data using mechanical models (thin plate and membrane). He solved the minimization problem using the finite elements method applied over hierarchy of image resolutions. He also showed how the shape-from-shading and optical flow problems can be approached over a hierarchy of resolutions [84].

Burt proposed hierarchical operators for motion estimation, and suggested correlation and gradient-based methods [15]. He later developed a hierarchical correlation algorithm [16], where a pyramid of band-pass filtered images was used to obtain motion estimates at various levels of resolution (frequency bands). The estimates from different levels have not been combined into a full-resolution field, however. In other work, Glazer *et al.* [30] and Anandan [6] have performed hierarchical correlation matching over a pyramid of low-pass filtered images. They used lower resolution estimates for subsequent estimation at higher resolution levels. In his later work Glazer [31] also extended Horn and Schunck's approach by applying it in a hierarchical manner. Enkelmann [22] has similarly extended the *oriented smoothness* approach of Nagel [68], [69].

In his early work Burt [13] has developed fast algorithms for generating pyramids of low-pass or band-pass filtered images. His filtering relies on scale-invariant kernels which in the limit approximate a Gaussian for low-pass filtering and a Laplacian for band-pass filtering. Another fast algorithm for band-pass pyramid construction was proposed by Crowley and Stern [20]. It is not clear, however, how much filtering should be applied to the data in order to construct a pyramid suitable for motion estimation.

The filtering performed in a hierarchical algorithm allows it to disambiguate the matching problem to a certain extent. Reduced high-frequency content also allows it to perform matching at larger displacements. These clear advantages do not, however, speed-up the computation process. Since the high-detail data structure has been reduced, it should be expected that also such structures will be reduced in the estimator (motion field). Hence, it became apparent that the spatial grid of the motion field may be subsampled to such an extent as to accommodate the local variation of the data. In this way at low image resolution levels the displacement fields would be computed at sparse locations only. The density of the displacement sampling grid would increase with the resolution of the data. Hence two independent pyramids would be constructed: one for the images and one for the displacement field. The subsampling of the displacement grid improves the convergence of the hierarchical algorithm compared to the one with no estimator grid subsampling. This improvement is due to two factors:

1. number of vectors to be computed at lower resolution level is smaller (e.g., for interlevel subsampling by 2 there are about 4 times fewer vectors to be computed; the exact number depends on whether the pyramid is *even* or *odd* [13], [31]),
2. propagation of the displacement field structure (this applies to methods which involve certain relationships between neighbouring vectors) is faster because of the absolute distance between displacement vector positions.

There is a danger, however, that this improved expansion of motion characteristics may cause excessive smoothness across motion boundaries (invisible at such low-resolution scale) unrecoverable at subsequent higher resolution levels.

2.5 REGULARIZATION METHODS

Recall the definition of a well-posed problem from Section 2.2 and its consequences for motion estimation. It comes as no surprise that, like many early vision problems, motion estimation being an inverse problem is ill-posed. To deal with the ill-posed problems, quite frequently arising in applied science, two branches of mathematical analysis have been developed: the theory of *generalized inverses* and the *regularization* theory. A concise review of both approaches to solving the ill-posed problems as well as numerous references

are given by Bertero *et al.* [9]. Here, only the principle ideas underlying both approaches will be given in order to refer to them in some later sections of this thesis.

Assume that functional spaces X and Y, as well as a linear, continuous operator L from X to Y are given. The task of an inverse problem is to find, for some given $h \in Y$, a function $d \in X$ such that:

$$h = Ld.$$

The direct problem would be to compute h given d. The theory of generalized inverses attempts to solve the problem by minimizing a norm of a certain function. Inverses can be classified according to the choice of that function as follows:

1. *Least squares inverses*: the following variational problem is solved:

$$\min_d \|Ld - h\|_Y, \tag{2.13}$$

   where $\|\cdot\|_Y$ denotes the norm in Y. This problem results in the linear system $L^*Ld = L^*h$ (L^* is the adjoint of L) for which the existence and uniqueness of the solution depend on the properties of L.

2. *Generalized inverses*: the solution of (2.13) is sought such that it is of minimal norm:

$$\min_d \|d\|_X.$$

3. *C-Generalized inverses*: the solution of (2.13) is sought such that it is also minimal in a constraint space:

$$\min \|Cd\|_Z,$$

   where C is a linear operator from X into the constraint (functional) space Z.

An alternative to the generalized inverses are the regularization methods. The regularization method as proposed by Tikhonov will be briefly summarized next, but some other perspectives on regularization theory can be found in [9]. Probably the most investigated technique of regularization is the following optimization problem:

$$\min_d \|Ld - h\|_Y^2 + \lambda \cdot \|Cd\|_Z^2, \tag{2.14}$$

where the same notation is used as above. The parameter λ is called the regularization parameter, and $\|C \cdot\|$ is the stabilizing functional, which usually expresses some desired

property expected from the solution (e.g., smoothness, directionality). λ weights the compromise between data approximation and model fitting (expressed by the stabilizing functional). In spite of existing theory for the optimal choice of λ, such choice is probably the most difficult problem in regularization.

The theory of generalized inverses and regularization for non-linear problems †, has also been investigated but it is not well-developed yet. The conclusions from existing results are that the solutions may not exist, and even if they do, they may be not unique.

There are numerous examples of regularized approaches to the early vision problems [74]. In optical flow estimation, the methods proposed by Horn and Schunck [41], Hildreth [39], Anandan [5] are examples of regularization, where the ill-posedness, also understood as the aperture problem [39], has been treated by imposing the smoothness constraint on the solutions (the smoothest velocity field is sought from the set of possible velocity fields consistent with the measurements). Also numerical differentiation, used in edge detection and other early vision problems, has been formulated in the framework of regularization [75], [40], where the stabilizing functional uses the second derivative of the approximating function. Other early vision problems approached from the regularization point of view are: shape from shading [45], surface interpolation [32], [33], [34], [85], curve fitting in the presence of discontinuities [60].

2.6 STOCHASTIC MODELING AND ESTIMATION

Various physical phenomena, among them image acquisition, can be interpreted deterministically or statistically. The topics discussed above treated such phenomena in a deterministic manner i.e., it was assumed that the underlying phenomenon as well as its relationship with the observed data were known with probability 1. An alternative to this approach is to consider the underlying process and/or its relationship with the observations as samples of some random processes. For example, in image acquisition the observed image can be related to the underlying one through an additive random noise, and also the true underlying image can modeled for example by a Markov random field.

† The linear operator Ld is replaced by $A(d)$, where $A(\cdot)$ is a non-linear operator.

Three stages can be identified in such a stochastic approach to estimation: modeling, formulation and solution. The statistical modeling attempts to mathematically express the behaviour of physical processes based on the probability theory. The statistical formulation relates various phenomena (variables) in a probabilistic manner, and the statistical solution involves a probabilistic method which solves the result of the formulation stage. Note that the outcome of the statistical solution depends on random numbers used, hence in principle is not repeatable †.

The best known statistical models in image modeling are: autoregressive (AR), moving average (MA) and autoregressive moving average (ARMA) models [2]. The AR models employ a recursive filter driven by a random noise to approximate such quantities as image intensities [35], [47]. Similarly the MA models use a non-recursive filter also excited by random noise. The combination of the two is known as the ARMA model. A closely related concept of a Markov random field has also been used for modeling [92]. The AR equation can be viewed as a means of generating Markov random field samples. The Markov random fields have been used in the context of conditional probabilities rather than AR equations for texture modeling [38], [19] and image modeling [26]. Also other random fields, with neighbour probability dependence specified by a correlation function instead of the Markov property, have been used in image modeling [2].

Usually the formulation stage is closely related to the modeling. Examples of formulation (and modeling) will be given for the linear restoration problem:

$$g = h * f + n$$

where f is the original image, g is the observed image, n is noise, h is a filter impulse response and $*$ denotes linear convolution. Note that the additive noise and the deterministic filter h are elements of the model. The goal is to recover f from the knowledge of g and some additional information or assumptions. The following models and formulations can be used to solve the above problem:

1. **linear regression**: *model* - n is a random process with known mean and covariance; *formulation* - minimize the mean squared error between f and

† If pseudo-random numbers generated by some algorithm are used, then of course the results are repeatable.

its linear estimate; results in the estimator which is a conditional expectation of f given g,

2. **Wiener estimation**: *model* - f and n are random processes with known means and covariances; *formulation* - minimize the mean squared error between f and its linear estimate; results in the Wiener filtering,
3. **Bayesian estimation**: *model* - f and n are random processes with known joint probability distribution $P(f,g)$ (from which all marginal distributions can be computed); *formulation* - minimize the conditional expected cost $E_{f/g}[\theta(f-\widehat{f})]$, where θ is a positive definite function, $\widehat{f}$ is an estimate; minimization of this cost results in the optimal Bayesian estimator; if θ is the squared Euclidean norm, Bayesian estimation is simplified to the minimum mean squared error (MMSE) estimation; if θ is a Dirac impulse then Bayesian estimation results in the maximum *a posteriori* probability (MAP) estimation, and if additionally the *a priori* probability distribution is uniform then the maximum likelihood (ML) estimation is obtained.

In the case of linear regression and Wiener estimation, application of such tools as linear algebra and theory of generalized inverses (Section 2.5) results in a closed form solution, usually implemented in the Fourier transform domain. In general such closed form solution does not exist for Bayesian estimation, but even when it does, as in the case of MMSE estimation, its implementation is not straightforward (how do we compute a conditional mean of a 512 by 512 image ?). In such a situation stochastic methods are used, for example *Monte Carlo* methods [36], which are usually concerned with estimating numerical values of some unknown distribution parameters (e.g., expected value). The particularly well known example of *Monte Carlo* methods is the *Metropolis algorithm* [65], which was originally proposed to study the behaviour of a many-particle system in thermal equilibrium. A recent example in this class of techniques is the *Gibbs sampler* developed by Geman and Geman [26] specifically to generate samples of Markov random fields. Both methods can be used for estimation of certain distribution parameters, however they become especially useful for minimization of objective functions via *simulated annealing* [52], a stochastic minimization method which under certain conditions [89] is able to attain the global optimum. Some minimization problems resulting from the Bayesian estimation could also be solved via deterministic optimization techniques like the steepest descent, Newton or conjugate gradients methods. However, if their initial state is not "sufficiently close" †

† This notion depends on the modality of the objective function.

to the global minimum, they will be able to locate only a local minimum (Section 2.7).

More detailed and formal discussion of these three stages will be given in Chapters 3 and 4.

2.7 FINDING THE GLOBAL OPTIMUM: A SIMPLE EXAMPLE

In this section three approaches to an objective function minimization will be demonstrated on a simple 1-D example of segment matching. It will be shown how the difficulty with localization of the global optimum performed by a steepest-descent method (due to function multimodality) can be alleviated by using either the hierarchical or the stochastic approach.

Let $f(y)$ and $g(y)$ be real-valued functions of the real position y, and let d be a real number from $[-K, K]$. Consider the following objective function $\phi(d)$ as the matching error between f and g over the segment $[0, N-1]$ (d is assumed constant over this segment):

$$\phi(d) = \sum_{x=0}^{N-1} [f(x) - g(x+d)]^2.$$

To find the optimal displacement d^*, the solution to the following minimization problem must be found:

$$\phi(d^*) = \min_d \phi(d). \tag{2.15}$$

To solve (2.15), the following three methods will be used:

1. **Gauss-Newton optimization.**
   Assume that some initial displacement d_0 is known, and that higher than first-order terms in the Taylor expansion of $g(d)$ can be neglected. Then $\phi(d)$ can be approximated as follows:

   $$\phi(d) \approx \sum_{x=0}^{N-1} [f(x) - g(x+d_0) - (d-d_0)\frac{\partial g(x+d_0)}{\partial x}]^2,$$

   and from the necessary condition for optimality ($\partial\phi/\partial d = 0$) the following iterative update equation can be derived:

   $$d^n = d^{n-1} + \frac{\sum_{x=0}^{N-1}[f(x) - g(x+d^{n-1})]\frac{\partial g(x+d^{n-1})}{\partial x}}{\sum_{x=0}^{N-1}[\frac{\partial g(x+d^{n-1})}{\partial x}]^2},$$

   where n denotes iteration number and d^{n-1} is used instead of d_0. Note that the update term is proportional to the gradient of the objective function

$\phi(d)$, hence this method belongs to the steepest-descent algorithms. The process stops if total error over M consecutive iterations does not exceed some threshold θ. This convergence criterion can be expressed as follows:

$$\sum_{j=i}^{i+M} (d^j - d^{j-1})^2 < \theta.$$

2. **Hierarchical Gauss-Newton optimization.**
   First, a pyramid of low-pass filtered versions of signals f and g is produced. Then the standard Gauss-Newton minimization described above is applied to the most filtered f and g. The result of this estimation is supplied as the initial value d_0 to another Gauss-Newton minimization but applied to the less filtered signals f and g. This process is repeated until the full resolution estimate is obtained. The same convergence criterion as for the single-level method is used.

3. **Simulated annealing.**
   This method will be discussed in detail in Chapter 4, but it should be mentioned that simulated annealing is a stochastic method able under certain conditions to find the global minimum of a cost functional. It is basically a random search through the state-space of the estimator with gradual (very slow) lowering of a "temperature" parameter, so that after sufficiently long evolution the only possible mode is the one corresponding to the global minimum.

The signals f and g were generated by a moving-average process as follows:

1. a sequence r of independent uniformly distributed random numbers from the range [0.0,1.0) was generated,
2. a narrow-band low-pass linear-phase FIR filter was applied to the random sequence r to obtain signal f (Fig. 2.2),
3. to avoid perfect segment match, signal g was generated by superimposing white Gaussian noise (mean m=0.05, variance σ^2=0.04) on random sequence r, then applying to the result the same filtering as in 2., and finally shifting the outcome by the displacement d=100.0 (Fig. 2.2),
4. to obtain the "pyramid" of f and g signals for hierarchical Gauss-Newton approach (Fig. 2.3), inter-level low-pass filtering with a linear-phase FIR filter derived from Gaussian distribution with the standard deviation of 20.0 was used.

Fig. 2.4 shows the objective functions $\phi(d)$ for 4 resolution levels. Note that $\phi(d)$ after single filtering of f and g is still multimodal, and only after triple filtering it becomes unimodal.

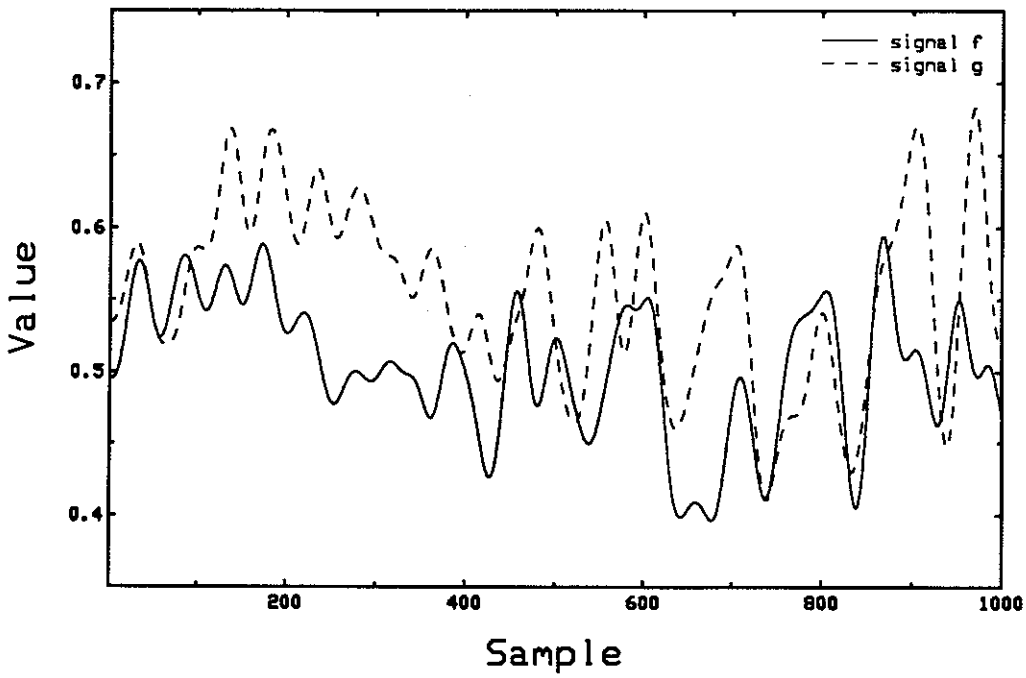


Fig. 2.2 Input signals f and g for segment matching (before MA filtering signal g was corrupted by white Gaussian noise with mean m=0.05 and variance σ^2=0.04, true displacement d=100.0).

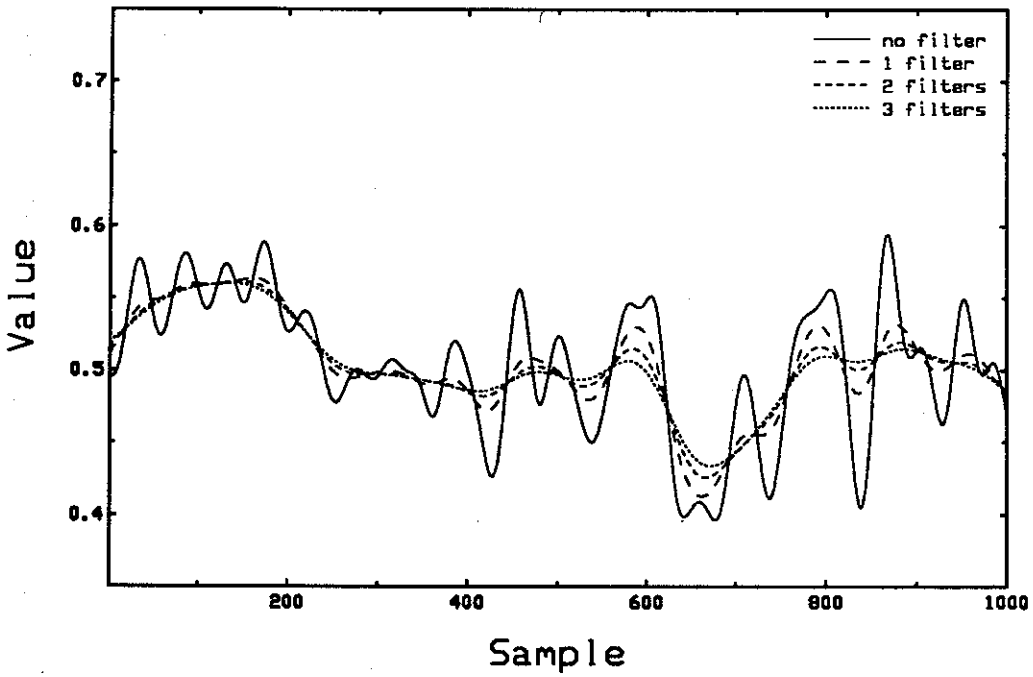


Fig. 2.3 Input signal f over 4 levels of resolution (filtering based on FIR approximation to the Gaussian with standard deviation of 20.0).

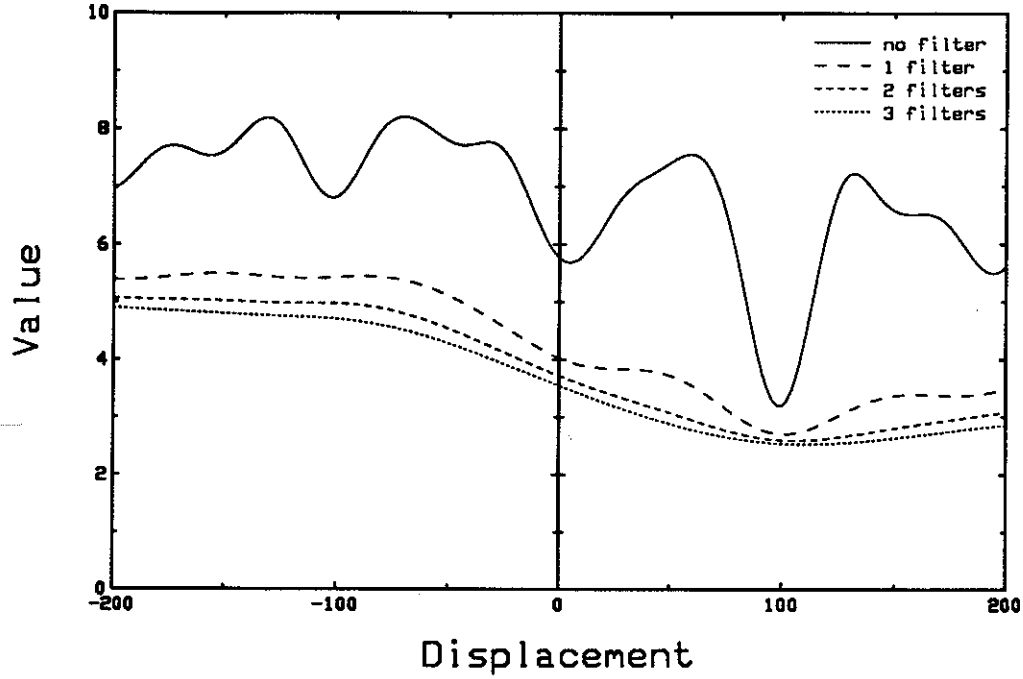


Fig. 2.4 Hierarchy of objective functions ϕ as a function of displacement (filtering based on FIR approximation to the Gaussian with standard deviation of 20.0).

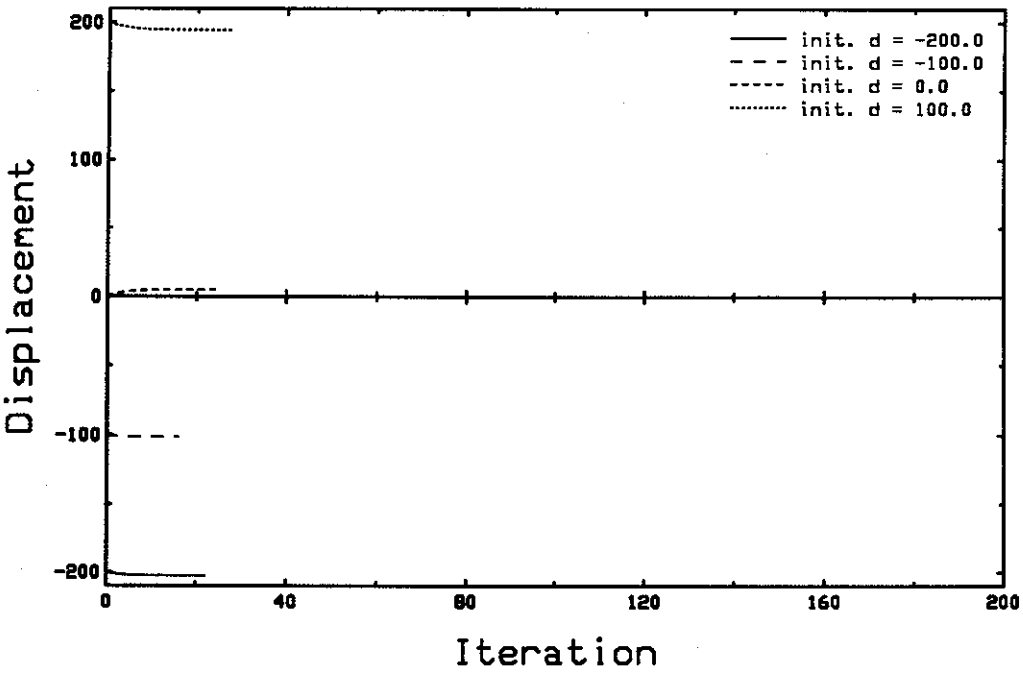


Fig. 2.5 Evolution of the estimation process for the Gauss-Newton method from 4 initial states (θ=0.0001).

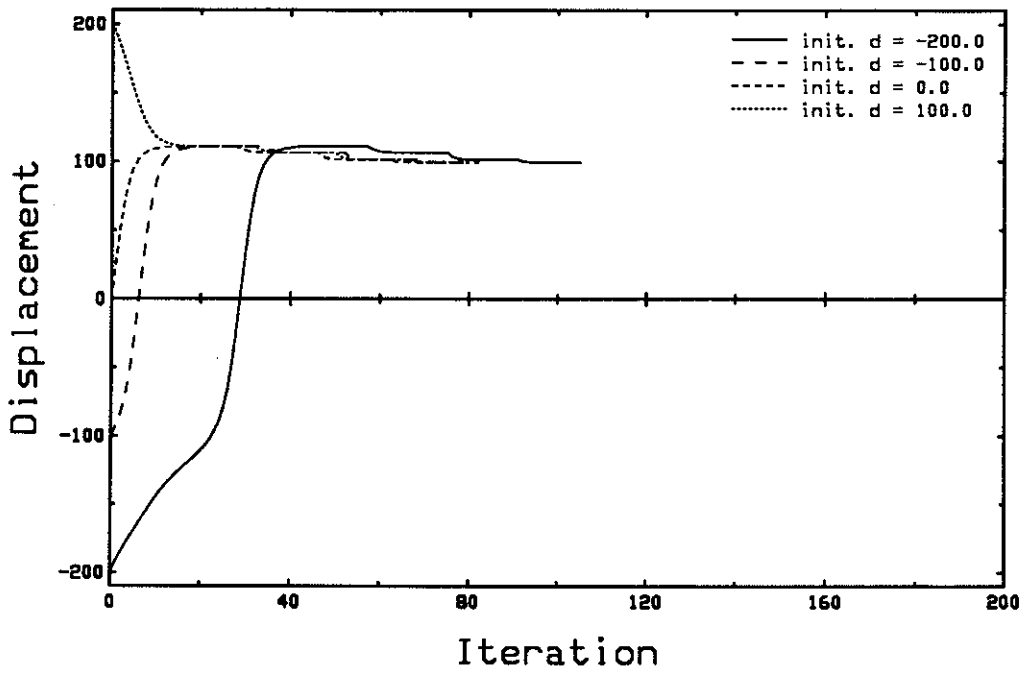


Fig. 2.6 Evolution of the estimation process for the hierarchical Gauss-Newton method from 4 initial states (θ=0.0001).

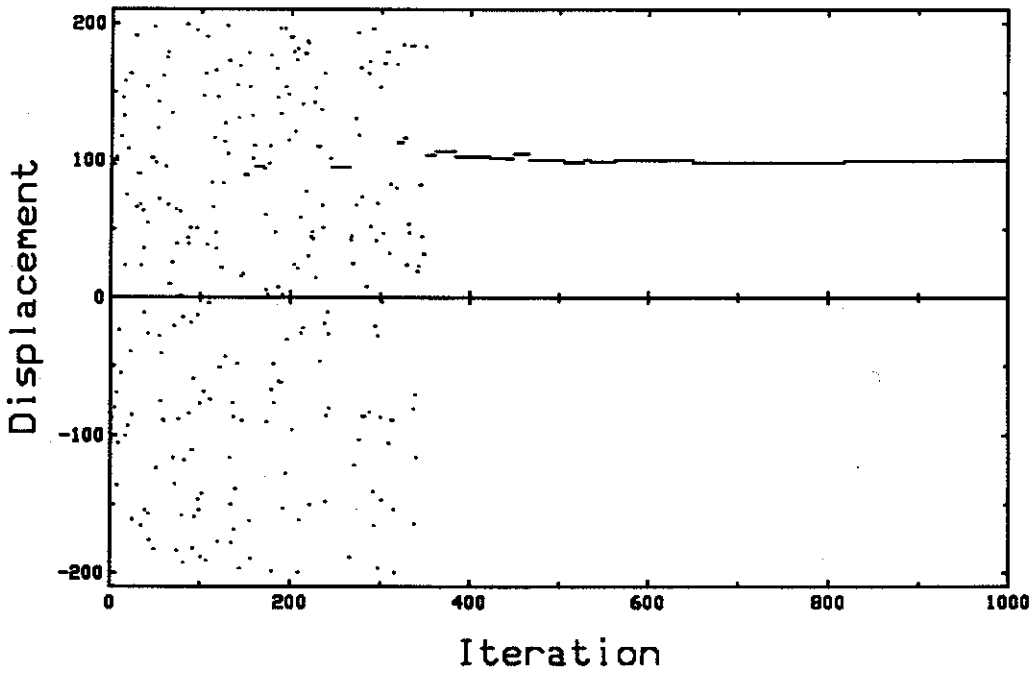


Fig. 2.7 Evolution of the estimation process for simulated annealing; initial state d_0=-200.0, initial temperature T_0=10.0.

For each method the estimation of d was performed starting from 4 different initial states (values of d_0). As the convergence criterion for the deterministic approach the stopping threshold θ=0.001 over M=10 iterations was used. Fig. 2.5 shows the iteration-wise evolution of the Gauss-Newton method. It is clear that the method is highly dependent on the initial state, and is not able to "climb" any of the encountered "hills". The 4-level hierarchical Gauss-Newton method, as it can be seen in Fig. 2.6, found the global optimum quite easily. At level 3 it found only an approximate estimate ($\approx$110.6, for initial displacement of -200.0), which was refined at subsequent levels ($\approx$105.9, 101.0, 99.1). In experiments with various amounts of hierarchical filtering it was observed that the success of the approach is very dependent on this filtering. In the last experiment the method of simulated annealing was used with the starting "temperature" parameter T_0=10.0 and 5% reduction every 10 iterations. The method performs a very random search initially (Fig. 2.7), but then it visits the correct state more and more often until it locks to the true displacement. Only one initial state d_0=-200.0 has been tested since the search is very random at the beginning (independence of simulated annealing from the initial state follows naturally). Note that no filtering was performed, hence the method is able to "climb" all the encountered "hills" of the full resolution ϕ. This method will be discussed in detail in Chapter 4, where its limitations in practical implementation will be also reported.

The exercise presented in this section aimed at showing two successful but completely different approaches to minimization of multimodal functions frequently encountered in computer vision. I will concentrate in this thesis mostly on the stochastic approach, however I will also present some deterministic approximations (to these stochastic algorithms) implemented hierarchically.

Chapter 3

BAYESIAN FORMULATION OF MOTION ESTIMATION

Relatively little research has been done to date in statistical approach to 2-D motion estimation. The only statistical estimation method which has been investigated is the maximum likelihood (ML) estimation.

In this chapter I propose to model the observed images (data) as random fields (RFs) via an *observation model*, and the motion (displacement) fields as Markov random fields (MRFs) – *motion (displacement) model*. These two models are combined into a single description by application of Bayes rule to the *a posteriori* probability of obtaining a certain motion field given the observed images. Two estimation criteria are proposed: maximum (total) *a posteriori* probability and minimum expected cost.

This chapter is organized as follows. First the terminology will be presented, followed by a brief overview of Gibbs distributions and Markov random fields. Then, the estimation criteria will be discussed, followed by the description of three models incorporated in these criteria. Finally, the *a posteriori* probability will be derived.

3.1 TERMINOLOGY

Let u denote the true underlying time-varying image to be defined in Section 3.4. The observed image g (also time-varying) is considered to be a sample of a random field (multidimensional stochastic process) G, and also to be related to the underlying image u via some random transformation. The image g is assumed to be sampled on a lattice Λ_g in R^3 (horizontal, vertical and temporal directions). Such a lattice is a collection of *sites* $\mathbf{s} \in R^3$ uniquely described by a sampling matrix [21].

Consider the true underlying image u defined over continuous spatio-temporal positions $(\mathbf{x}, t)$. Let u at time $t = t_1$ be called the preceding image and at $t = t_2$ – the following image. Excluding the occlusion and newly exposed areas, for every point in the preceding image there exists a corresponding point in the following image. Every such pair of points can be connected by a straight line. Since u is defined over continuous $(\mathbf{x}, t)$, these lines will intersect a plane located at t $(t_1 < t < t_2)$ over a dense set of locations. In other words, for each $(\mathbf{x}, t)$ there exists a line joining corresponding image points at times t_1 and t_2. Note that these lines coincide with the true motion trajectory at the end points (t_1 and t_2), and not necessarily between them. The true motion trajectory will intersect (in general) the plane at time t at location different from $(\mathbf{x}, t)$†.

Let the 2-D projections of line segments between t_1 and t_2 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 unfeasible, however, to compute displacement vectors on a continuum of spatial positions, hence $\boldsymbol{d}$ is assumed to be sampled on a lattice $\Lambda_{\mathbf{d}}$ in R^3. In the literature the cases where

1. $\Lambda_{\mathbf{d}}$ is a sub-lattice of Λ_g : $\quad \Lambda_{\mathbf{d}} \subset \Lambda_g$, or
2. $\Lambda_{\mathbf{d}}$ is identical to Λ_g : $\quad \Lambda_{\mathbf{d}} = \Lambda_g$,

have been most frequently considered. In this thesis a more general situation, where $\Lambda_{\mathbf{d}}$ and Λ_g are arbitrary, will be investigated. Consequently a displacement vector may be defined at a spatio-temporal position which does not belong to Λ_g. Clearly, approximation of the true motion trajectory by a line will be precise when $\Lambda_{\mathbf{d}} = \Lambda_g$, for example in motion-compensated prediction. In the case, however, when $\Lambda_{\mathbf{d}} \neq \Lambda_g$, as in motion-compensated interpolation, there will be an error introduced due to a departure of motion trajectory from linearity in the interval (t_1, t_2). If this interval is small e.g., 1/60 sec., such an error should be minor.

The investigations which follow are valid for any lattices Λ_g, $\Lambda_{\mathbf{d}}$, however without loss of generality, it is assumed that they are rectangular lattices with horizontal, vertical

† To provide a more accurate approximation to the true motion trajectory, two vectors: between t_1 and t, and between t and t_2, would have to be defined. Then, however, the number of unknowns for each $(\mathbf{x}, t)$ would grow from 2 to 4, and the problem would be severely underconstrained.

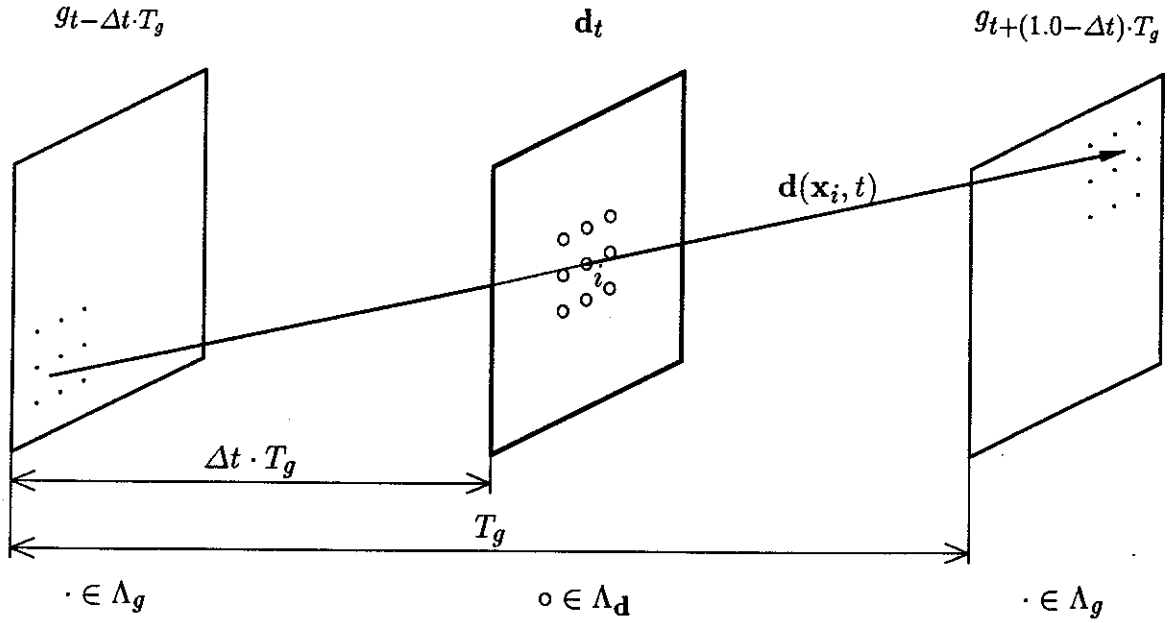


Fig. 3.1 Definition of the displacement field $\mathbf{d}_t$ for motion estimation from two image fields (only one vector is displayed). Sites of lattice $\Lambda_{\mathbf{d}}$ ($\circ$) are also pivoting points for the displacement vectors – every vector $\mathbf{d}(\mathbf{x}_i, t)$ crosses site $(\mathbf{x}_i, t)$ while its ends are "free" to move and do not necessarily belong to lattice Λ_g.

and temporal sampling periods (T_g^h, T_g^v, T_g) and $(T_{\mathbf{d}}^h, T_{\mathbf{d}}^v, T_{\mathbf{d}})$, respectively. Consequently consecutive image fields are spaced by T_g seconds, while motion fields are spaced by $T_{\mathbf{d}}$ seconds. Each field of the image sequence contains M_g picture elements (pels), and each motion field consists of $M_{\mathbf{d}}$ vectors. Let the horizontal and vertical dimensions of the motion fields be $M_{\mathbf{d}}^h$ and $M_{\mathbf{d}}^v$, respectively ($M_{\mathbf{d}} = M_{\mathbf{d}}^h \times M_{\mathbf{d}}^v$). The numbering order for the picture elements ($i = 1, ..., M_g$) and the motion vectors ($i = 1, ..., M_{\mathbf{d}}$) in a given field is arbitrary (e.g., horizontal scan). With the above assumptions a site $\mathbf{s}_i = [kT_g^h, lT_g^v, mT_g] \in \Lambda_g$ (for some integer i, k, l, m) is an image pel in the m-th field with spatial coordinate (k, l).

The true displacement field $\boldsymbol{d}(\mathbf{x}, t)$ is an array of vectors defined over the lattice $\Lambda_{\mathbf{d}}$, and is assumed to be a sample (realization) from random field $\mathbf{D}(\mathbf{x}, t)$. Let also $\mathbf{d}(\mathbf{x}, t)$ denote any sample field from $\mathbf{D}(\mathbf{x}, t)$. In general every motion field can be estimated from K_i image fields. Here, however, the case of K_i=2 will be investigated. Assuming a linear motion trajectory between two images, as discussed above, the definition of the displacement field is given below, and also illustrated in Fig. 3.1.

Definition: The displacement (motion) field $\mathbf{d}(\mathbf{x}, t)$ defined over $\Lambda_{\mathbf{d}}$ is a set of 2-D vectors such that for all $(\mathbf{x}_i, t) \in \Lambda_{\mathbf{d}}$ the (*preceding*) image point

$$(\mathbf{x}_i - \Delta t \cdot \mathbf{d}(\mathbf{x}_i, t), t - \Delta t \cdot T_g)$$

has moved to the (*following*) point:

$$(\mathbf{x}_i + (1.0 - \Delta t) \cdot \mathbf{d}(\mathbf{x}_i, t), t + (1.0 - \Delta t) \cdot T_g),$$

where

$$\Delta t = \frac{t}{T_g} - \left\lfloor \frac{t}{T_g} \right\rfloor$$

is the normalized (with respect to the inter-image distance T_g) temporal distance between the motion field position and the preceding image position.

Note that the above definition allows us to locate the displacement vector at any spatio-temporal position. In particular consider the following two limiting cases:

1. $t = m \cdot (1.0 + \varepsilon) \cdot T_g$, $\varepsilon \in R \quad \Rightarrow \quad \lim_{\varepsilon \to 0} \Delta t = 0.0$, and in the limit the preceding and the following image fields are defined as follows:
$$(\mathbf{x}_i, t) \longrightarrow (\mathbf{x}_i + \mathbf{d}(\mathbf{x}_i, t), t + T_g),$$
2. $t = m \cdot (1.0 - \varepsilon) \cdot T_g$, $\varepsilon \in R \quad \Rightarrow \quad \lim_{\varepsilon \to 0} \Delta t = 1.0$, and in the limit the preceding and the following image fields are defined as follows:
$$(\mathbf{x}_i - \mathbf{d}(\mathbf{x}_i, t), t - T_g) \longrightarrow (\mathbf{x}_i, t).$$

Essentially both cases are similar, since the temporal position of the motion field coincides with the temporal position of an image. The first case describes the forward estimation where the motion field is computed from the knowledge of the current and the following images. The second case is an example of backward estimation: motion field is obtained from the current and the preceding images. In order to avoid ambiguity, when the motion field and image temporal positions coincide, it is enough to restrict Δt to one of the following intervals: $[0.0, 1.0)$ or $(0.0, 1.0]$. The first of these two intervals will be used here.

Let an estimate of the true displacement field $\boldsymbol{d}$ for a given image sequence be denoted by $\widehat{\mathbf{d}}$. Let also the subscript t denote the restriction of a random field or of its realization to time t. Then, $\mathbf{d}_t$ is a realization of random field $\mathbf{D}_t$ ($\mathbf{D}$ at time t), while $\mathbf{d}(\mathbf{x}_i, t_j)$ is a

single displacement vector at spatial location $\mathbf{x}_i$ and time t_j. Hence, the following notation will be used:

$G, N, \mathbf{D}$ – random fields: observation (image) RF, noise RF and displacement RF; for instance $G(\mathbf{x}_i, t_j)$ is a random variable (RV) from the observation RF,

$g, n, \mathbf{d}$ – samples (realizations) of the random fields $G, N, \mathbf{D}$ respectively; for example $g(\mathbf{x}_i, t_j)$ is a sample of the observation RF at $(\mathbf{x}_i, t_j)$ or in other words the current value (intensity) of the random variable $G(\mathbf{x}_i, t_j)$,

$(G = g), (N = n), (\mathbf{D} = \mathbf{d})$ – understood as follows: the RF G has currently the realization g i.e., $G(\mathbf{x}_i, t_j) = g(\mathbf{x}_i, t_j)$ for all i, j,

$G_t, N_t, \mathbf{D}_t, g_t, n_t, \mathbf{d}_t$ – subscript t in any symbol of a RF or its realization means the restriction to the time instant t, hence $(G_{t_j} = g_{t_j}) \equiv (G = g, \text{at } t = t_j) \equiv (G(\mathbf{x}_i, t_j) = g(\mathbf{x}_i, t_j), \text{all } i)$.

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 pel state-space and $(\cdot)^M$ denotes an M-fold Cartesian product. Similarly, the random field $\mathbf{D}_t$ is defined over the state-space $\mathcal{S}_\mathbf{d} = (\mathcal{S}'_\mathbf{d})^{M_\mathbf{d}}$, where $\mathcal{S}'_\mathbf{d}$ is the single vector state-space. Two cases of $\mathcal{S}'_\mathbf{d}$ are considered:

1. $\mathcal{S}'_\mathbf{d}$ 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. $\mathcal{S}'_\mathbf{d} = R^2$ is a continuous state-space.

All state spaces correspond to a sufficiently fine quantization of the underlying continuous image intensities and displacements, so that their characteristic properties are preserved.

3.2 GIBBS DISTRIBUTION AND MARKOV RANDOM FIELDS

The following material is presented for completeness, however it can be found in various references, for example: Geman and Geman [26], Besag [11], [10], Kindermann and Snell [51].

3.2.1 Gibbs distribution

In order to define the *Gibbs* distribution the concepts of *neighbourhood system*, *clique* and *potential function* are needed. For the purpose of this chapter let Λ denote a lattice

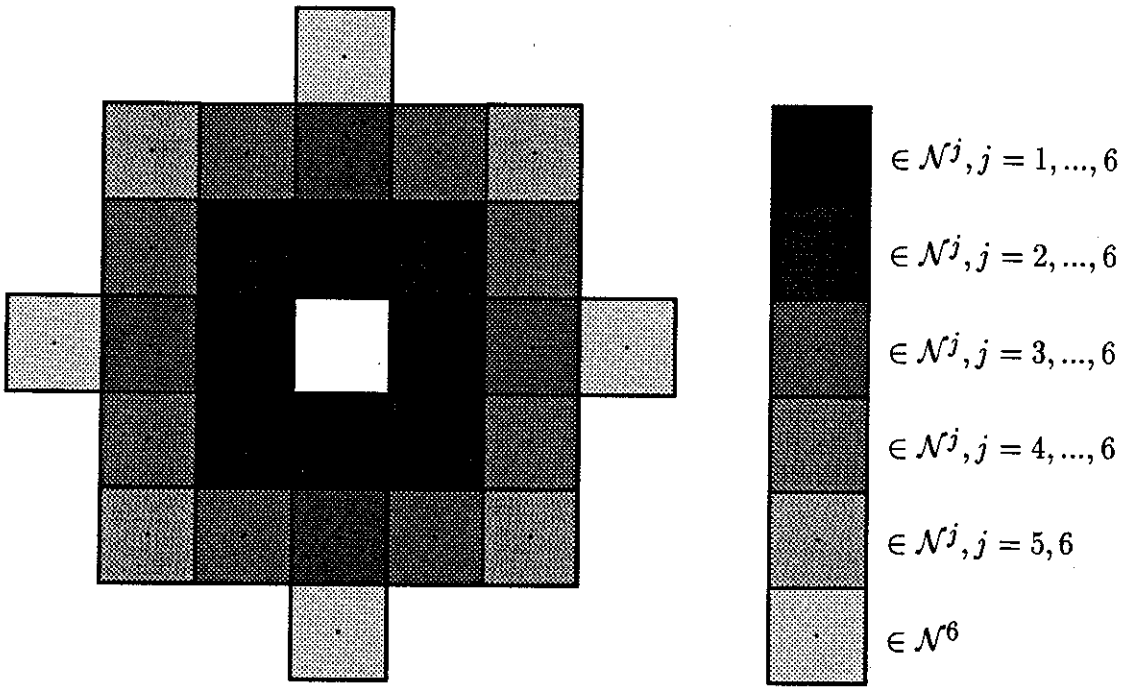


Fig. 3.2 Hierarchically organized neighbourhood systems of order 1 through 6: every j-th order system contains all sites of systems of order up to $(j-1)$.

with M sites, and let χ be a sample field from random field X defined over lattice Λ and state-space $\mathcal{S}$. With the above temporary definitions neighbourhood system is defined as follows.

Definition: A collection $\mathcal{N}$ of subsets of Λ

$$\mathcal{N} = \{\eta(\mathbf{s}_i) : \mathbf{s}_i \in \Lambda, \eta(\mathbf{s}_i) \subset \Lambda, 1 \leq i \leq M\}$$

is a *neighbourhood system* on Λ, if and only if, the *neighbourhood* $\eta(\mathbf{s}_i)$ of site $\mathbf{s}_i \in \Lambda$ satisfies the following conditions:

1. $\mathbf{s}_i \notin \eta(\mathbf{s}_i)$, **and**
2. if $\mathbf{s}_j \in \eta(\mathbf{s}_i)$, then $\mathbf{s}_i \in \eta(\mathbf{s}_j)$ for any $\mathbf{s}_i \in \Lambda$.

In this thesis random field models with only spatial (2-D) Markovian dependence are considered for motion. Hence, only 2-D neighbourhood systems will be investigated. As a consequence whenever a neighbourhood is discussed, the time coordinate t will be omitted and sometimes spatial coordinate $\mathbf{x}_i$ will be replaced by (k, l), since $\mathbf{x}_i = (kT_{\mathrm{d}}^h, lT_{\mathrm{d}}^v)$ for

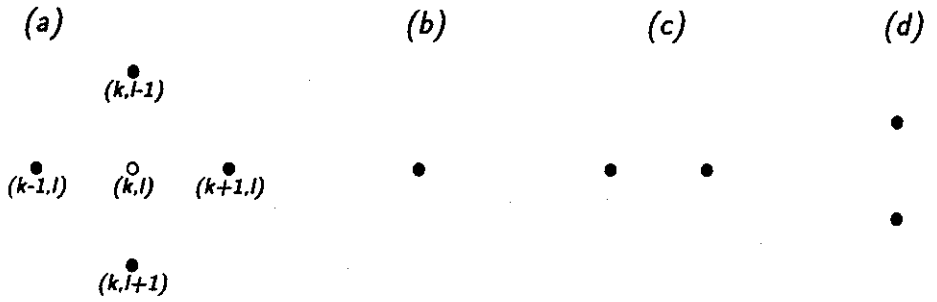


Fig. 3.3 First-order neighbourhood system $\mathcal{N}^1$ (a), and associated one-element (b), two-element horizontal (c) and vertical (d) cliques. Note that $\mathbf{x}_i = (k, l)$ for some integer k and l.

some integer k, l. Fig. 3.2 shows 2-D neighbourhood systems $\mathcal{N}^j, j=1,...,6$, in hierarchical order i.e., every higher-order structure includes all the lower-order ones. The first-order system $\mathcal{N}^1$, also known as the nearest-neighbour system, is commonly used in image modeling. The second-order neighbourhood system $\mathcal{N}^2$ comprises the sites of $\mathcal{N}^1$, and also additional diagonal sites. The higher-order systems are constructed similarly. Note that a neighbourhood system defined over Λ needs not be isotropic nor hierarchical. It should also be pointed out that the neighbourhood system is shift-invariant except at the image boundaries, where it must be redefined.

Definition: A *clique* c defined over a lattice Λ with respect to the neighbourhood system $\mathcal{N}$ is a subset of Λ such that:

1. two sites in a *clique* are neighbours i.e., $(i,j),(k,l) \in c$ and $(i,j) \neq (k,l) \Rightarrow (i,j) \in \eta(k,l)$, **or**
2. c consists of a single site.

The set of all cliques is denoted by $\mathcal{C}$. Cliques for the first- and second-order neighbourhood systems are shown in Fig. 3.3 and 3.4, respectively. Note that the cliques associated with $\mathcal{N}^2$ contain all the cliques associated with $\mathcal{N}^1$.

Definition: A *Gibbs distribution* with respect to the lattice Λ and the neighbourhood system $\mathcal{N}$ is a probability measure π on $\mathcal{S}$ such that

$$\pi(\chi) = \frac{1}{Z} e^{-U(\chi)/\beta}, \tag{3.1}$$

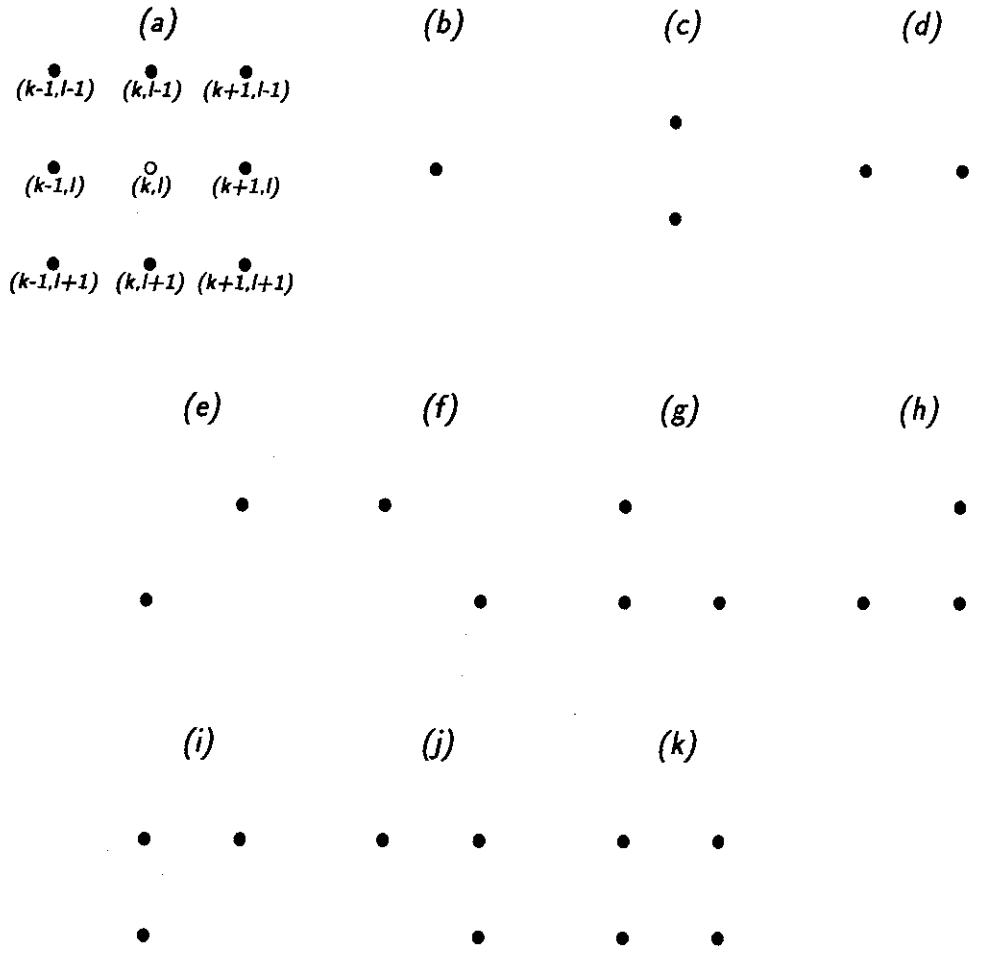


Fig. 3.4 Second-order neighbourhood system $\mathcal{N}^2$ (a), and associated cliques: one-element (b), two-element vertical (c), horizontal (d) and diagonal (e),(f), three-element (g),(h),(i),(j), and four-element (k). Note that $\mathbf{x}_i = (k, l)$ for some integer k and l.

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

$$U(\chi) = \sum_{c \in \mathcal{C}} V(\chi, c). \tag{3.2}$$

$V(\chi, c)$ is called a *potential function*, and depends only on those samples from χ which belong to the clique c. Z is called a *partition function*, and is a normalizing constant such that π is a probability measure.

3.2.2 Markov Random Fields

The discrete state-space MRF X is a multidimensional stochastic process with the

following properties:

1. $P(X = \chi) > 0, \quad \forall \chi \in \mathcal{S},$

2. $P(X_i = \chi_i | X_j = \chi_j, \forall j \neq i) = P(X_i = \chi_i | X_j = \chi_j, \forall j \in \eta(i)), \quad \forall i, \ \chi \in \mathcal{S},$

where P denotes a probability measure. Analogous property applies to the continuous state-space MRF when the probabilities P are replaced by cumulative distributions $F_X(\chi)$. Furthermore, assuming the existence of the probability density p (the differentiability of F_X), the above property applies directly with the densities p replacing the probabilities P. To keep the further derivations clear and simple, the discrete state-space notation (probability distribution P) will be used in this thesis. Exception to this rule will be where explicit continuous state-space expressions (probability density p) are needed.

In order to uniquely characterize a MRF the finite-dimensional joint probability distribution is necessary. To express this joint probability distribution of the complete random field X all initial and transitional (conditional) probability distributions are needed. This approach is cumbersome, because the conditional probability distributions must satisfy certain consistency conditions [10] (hence cannot be chosen arbitrarily), and because the computation of the joint distribution from these conditional distributions is usually difficult. Also the relationship between the form of a conditional probability distribution and the characteristic properties of a sample field (e.g., smoothness) is not obvious.

Such a clear and simple relationship can be provided through the Hammersley-Clifford theorem [10] which states that if X is a MRF on lattice Λ with respect to neighbourhood system $\mathcal{N}$, then the probability distribution of its sample realizations (configurations) is a Gibbs distribution (Eq. 3.1). This unique characterization of a MRF by a Gibbs distribution results in a straightforward relationship between qualitative properties of a MRF and its parameters via the potential functions V. Extension of the Hammersley-Clifford theorem to vector MRFs is straightforward (only a new definition of a state has to be provided).

3.3 ESTIMATION CRITERIA

The goal of this work is to estimate the true displacement field $\boldsymbol{d}(\mathbf{x}, t)$ corresponding to an underlying time-varying image $u(\mathbf{x}, t)$ on the basis of the observations $g(\mathbf{x}, t)$. As it was

assumed in Section 3.1, the estimation of $\boldsymbol{d}(\mathbf{x},t)$ will be based on 2 image fields (K_i=2). To simplify the notation, the temporal positions of the preceding and the following image fields will be denoted by $t_- = t - \Delta t \cdot T_g$ and $t_+ = t + (1.0 - \Delta t) \cdot T_g$, respectively (recall that the image sequence is temporally sampled). Let also $t_\pm$ denote either t_- or t_+.

3.3.1 Maximum a posteriori probability (MAP) estimation

The objective here is to determine the "best" or the "most likely" displacement field $\hat{\mathbf{d}}_t^* \in \mathcal{S}_\mathbf{d}$ given the observations g_{t_-}, g_{t_+}. Hence, $\hat{\mathbf{d}}_t^*$ must satisfy the relationship:

$$P(\mathbf{D}_t = \hat{\mathbf{d}}_t^* | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \geq P(\mathbf{D}_t = \hat{\mathbf{d}}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \quad \forall\, \hat{\mathbf{d}}_t \in \mathcal{S}_\mathbf{d}.$$

To obtain the posterior distribution for the discrete state-space case, Bayes rule for the discrete random variables can be applied as follows:

$$P(\mathbf{D}_t = \hat{\mathbf{d}}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \frac{P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \hat{\mathbf{d}}_t, G_{t_-} = g_{t_-}) \cdot P(\mathbf{D}_t = \hat{\mathbf{d}}_t | G_{t_-} = g_{t_-})}{P(G_{t_+} = g_{t_+} | G_{t_-} = g_{t_-})}, \quad (3.3.a)$$

while for the continuous $\mathcal{S}_\mathbf{d}$ the same rule for mixed random variables can be used:

$$p(\hat{\mathbf{d}}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \frac{P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \hat{\mathbf{d}}_t, G_{t_-} = g_{t_-}) \cdot p(\hat{\mathbf{d}}_t | G_{t_-} = g_{t_-})}{P(G_{t_+} = g_{t_+} | G_{t_-} = g_{t_-})}. \quad (3.3.b)$$

Again P is a probability distribution and p is a probability density. Note that since the probability in the denominator of (3.3) is not a function of the displacement process $\mathbf{D}_t$, it can be ignored, and the MAP estimate of $\boldsymbol{d}_t$ is the solution to the following optimization problem:

$$\max_{\hat{\mathbf{d}}_t} \Big[P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \hat{\mathbf{d}}_t, G_{t_-} = g_{t_-}) \cdot P(\mathbf{D}_t = \hat{\mathbf{d}}_t | G_{t_-} = g_{t_-}) \Big]. \quad (3.4)$$

To perform this optimization the explicit form of the conditional probability distributions involved in (3.4) must be known.

3.3.2 Minimum expected cost (MEC) estimation

Another approach to the estimation of $\boldsymbol{d}_t$ is to minimize the ensemble average over $\boldsymbol{d}_t, g_{t_-}, g_{t_+}$ of some positive definite cost functional measuring the error between the true

and estimated motion fields (minimum expected cost or MEC estimation) [62], [63]. Let this functional have the following form:

$$\Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t) = \sum_{i=1}^{M_{\mathbf{d}}} \theta(\boldsymbol{d}(\mathbf{x}_i, t), \hat{\mathbf{d}}(\mathbf{x}_i, t)),$$

where $\theta(\cdot,\cdot)$ is a positive definite function. The optimal Bayesian estimator $\hat{\mathbf{d}}_t^* \in \mathcal{S}_{\mathbf{d}}$ is defined as follows:

$$E_{\mathbf{d}_t, g_{t_-}, g_{t_+}}[\Theta(\mathbf{d}_t, \hat{\mathbf{d}}_t^*)] = \min_{\hat{\mathbf{d}}_t} E_{\mathbf{d}_t, g_{t_-}, g_{t_+}}[\Theta(\mathbf{d}_t, \hat{\mathbf{d}}_t)] \tag{3.5}$$

where $E_{\mathbf{d}_t, g_{t_-}, g_{t_+}}[\cdot]$ stands for the ensemble expectation over all configurations of $\mathbf{d}_t, g_{t_-}$ and g_{t_+}. Note that the estimates $\hat{\mathbf{d}}_t$, $\hat{\mathbf{d}}_t^*$ are functions of g_{t_-} and g_{t_+}, and can be thought of as mappings from R^{2M_g} to $R^{2M_{\mathbf{d}}}$. Expressing the expectation as a sum (discrete random variables), and using the positive-definiteness of $\theta(\cdot,\cdot)$, it can be shown (for the derivation consult Appendix 3.A) that (3.5) is equivalent to:

$$\min_{\hat{\mathbf{d}}(\mathbf{x}_i,t)} \sum_{\mathbf{r}\in\mathcal{S}'_{\mathbf{d}}} \theta(\mathbf{r}, \hat{\mathbf{d}}(\mathbf{x}_i,t)) \Big[\sum_{\mathbf{d}_t:\mathbf{d}(\mathbf{x}_i,t)=\mathbf{r}} P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \Big] \quad \forall i. \tag{3.6}$$

This means that the minimization with respect to the field $\hat{\mathbf{d}}_t$ in (3.5) can be achieved by individually minimizing marginal expected costs at each position $(\mathbf{x}_i, t)$.

Any further simplification of (3.6) requires explicit knowledge of the function θ. This function must reflect "goodness" of the estimate i.e., a worse displacement estimate should increase the value of θ, and a better one should reduce it. The following θ is used here:

$$\theta(\boldsymbol{d}(\mathbf{x}_i,t), \hat{\mathbf{d}}(\mathbf{x}_i,t)) = \|\boldsymbol{d}(\mathbf{x}_i,t) - \hat{\mathbf{d}}(\mathbf{x}_i,t)\|^2, \tag{3.7}$$

where $\|\cdot\|$ is the L^2 norm, for mathematical tractability. It can be shown (Appendix 3.A) that the solution to (3.6) with θ defined in (3.7) is

$$\hat{\mathbf{d}}^*(\mathbf{x}_i,t) = \sum_{\mathbf{r}\in\mathcal{S}'_{\mathbf{d}}} \mathbf{r} \Big[\sum_{\mathbf{d}_t:\mathbf{d}(\mathbf{x}_i,t)=\mathbf{r}} P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \Big] \quad \forall i, \tag{3.8.a}$$

or in other words it is the marginal conditional expectation $\overline{\mathbf{d}}(\mathbf{x}_i,t)$ of $\mathbf{D}(\mathbf{x}_i,t)$. This result could have been inferred directly from expressions (3.6) and (3.7), since for this choice of θ it is the minimum mean squared error (MMSE) estimation. For the continuous state-space case the optimal Bayesian estimator has the following form:

$$\hat{\mathbf{d}}^*(\mathbf{x}_i,t) = \int_{\mathbf{r}\in\mathcal{S}'_{\mathbf{d}}} \mathbf{r} \Big[\int_{\mathbf{d}_t:\mathbf{d}(\mathbf{x}_i,t)=\mathbf{r}} p(\mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) d\mathbf{d}_t \Big] d\mathbf{r} \quad \forall i. \tag{3.8.b}$$

To compute the conditional expectations from (3.8.a,.b) the *a posteriori* probability distribution $P(\mathbf{D}_t|G_{t_-}, G_{t_+})$ and density $p(\mathbf{d}_t|G_{t_-}, G_{t_+})$ are needed.

3.4 MODELS

To discuss the structural and the observation models, first the true underlying image u must be defined. In this thesis the true image u is understood as an illuminance pattern in some image plane obtained from the observed scene via an ideal optical system. Hence, u is defined over continuous spatio-temporal coordinates $(\mathbf{x}, t)$, and given $(\mathbf{x}_i, t_j)$, $u(\mathbf{x}_i, t_j)$ is a real number.

3.4.1 Structural model

In order to facilitate inference of motion from images it is necessary to assume a certain relationship or a "structural" model between motion vectors and image intensity values. Such a model is fundamental to any motion estimation algorithm. As reported in Section 2.1, it is usually assumed that no illumination effects are present in the scene viewed by the camera. Then, the constant image intensity or constant gradient (spatial or spatio-temporal) of that intensity along the motion trajectories is usually assumed as the "structural" model. Note that this assumption applies to the true underlying image u. The constant image intensity model will be used here as follows. Since the ultimate goal of this thesis is estimation of motion in temporally sampled sequences (TV), the notion of a displacement vector will be used instead of an instantaneous velocity. Assuming that over the time interval $[t_-, t_+]$ the intensity of the true underlying image u along the motion trajectory (true displacement $\boldsymbol{d}$) is constant, the following holds:

$$u(\mathbf{x} - \Delta t \cdot \boldsymbol{d}(\mathbf{x}, t), t_-) = u(\mathbf{x} + (1.0 - \Delta t) \cdot \boldsymbol{d}(\mathbf{x}, t), t_+). \tag{3.9}$$

A more complex model incorporating linear variation of intensity has been devised in [88], [28], [59], however it will not be considered here. Also as an open issue it still remains to account in this structural model for the occlusions and the newly exposed areas.

3.4.2 Observation model

As discussed above, the true underlying image u is an effect of an "ideal" projection of a scene onto an image plane. In reality, however, any image that will be used here has been acquired through a real video camera† and discretized using some electronic circuitry. This observed image g is a transformed copy of u after the following operations:

1. spatial optical filtering (non-ideal optical system),
2. spatio-temporal electronic filtering (sensor characteristics),
3. vertical and temporal sampling,
4. γ-correction,
5. horizontal electronic filtering (band-width limitation before horizontal sampling),
6. horizontal sampling,
7. quantization.

In comparison with the true image u the image g also incorporates certain noise and distortions. Their sources can be identified as the following:

1. image sensor noise,
2. quantization noise,
3. distortion due to aliasing.

To model the characteristic properties of the processes contributing to g is a very difficult task. First, consider a simplified case where g, u, n are continuous in value and defined over continuous $(\mathbf{x}, t)$. Let the observed image g be related to the true underlying image u as follows:

$$g(\mathbf{x}, t) = u(\mathbf{x}, t) + n(\mathbf{x}, t), \tag{3.10}$$

where $n(\mathbf{x}, t)$ are Gaussian random variables. Using the structural model (3.9) the following relationship can be easily verified:

$$\begin{aligned} &g(\mathbf{x} + (1.0 - \Delta t) \cdot \boldsymbol{d}(\mathbf{x}, t), t_+) - g(\mathbf{x} - \Delta t \cdot \boldsymbol{d}(\mathbf{x}, t), t_-) = \\ &n(\mathbf{x} + (1.0 - \Delta t) \cdot \boldsymbol{d}(\mathbf{x}, t), t_+) - n(\mathbf{x} - \Delta t \cdot \boldsymbol{d}(\mathbf{x}, t), t_-). \end{aligned} \tag{3.11}$$

The term on the left hand side of (3.11) is known as a displaced pel difference (DPD), and the one on the right is a displaced noise difference. Since the right hand side of relationship

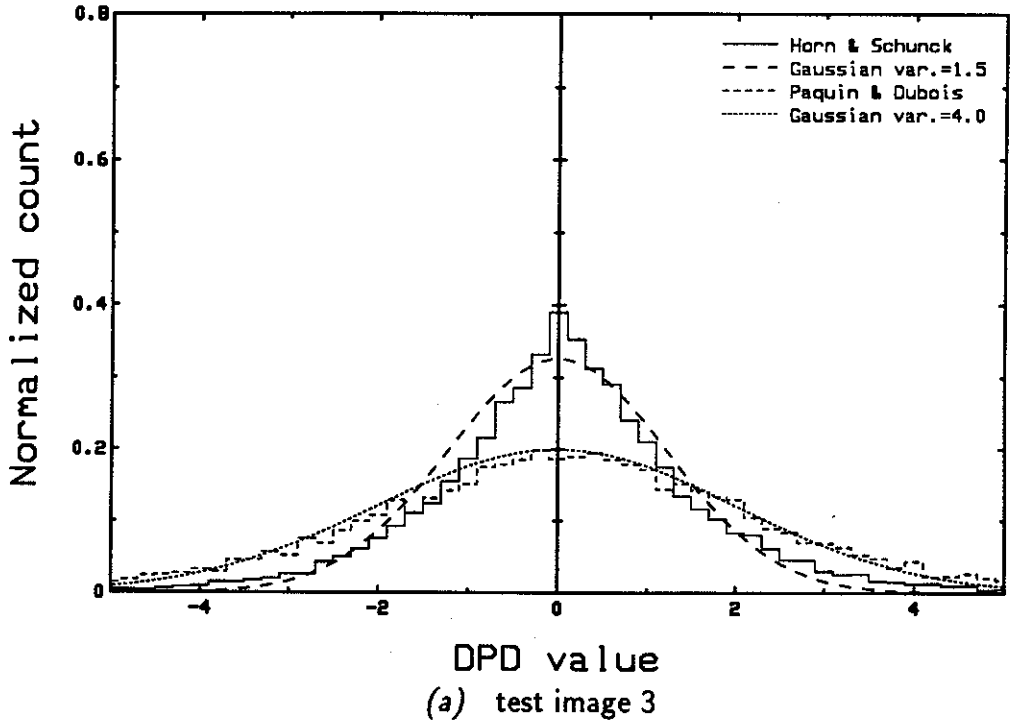
† Except for the test image 1 (Section 4.7) which has been generated synthetically.

(3.11) is a linear combination of Gaussian random variables, it is a Gaussian random variable itself with doubled variance.

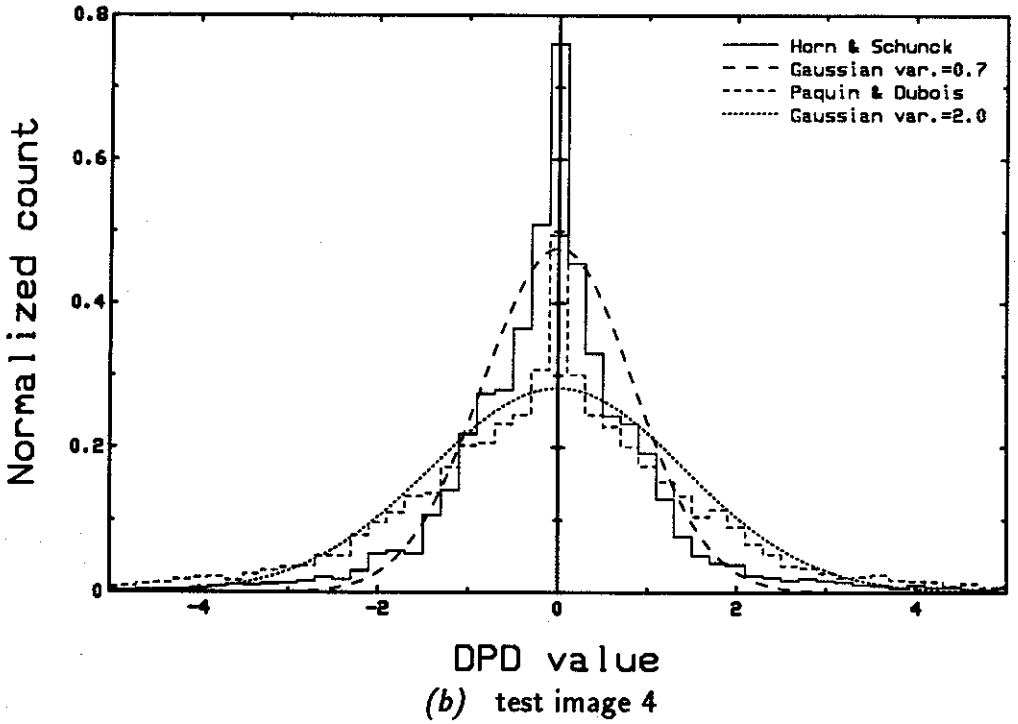
In reality other transformations and noise sources discussed at the beginning of this section should be taken into account too. It is very difficult, however, to theoretically derive a DPD model in such a complex case.

In order to gain more insight into the statistical properties of displaced pel difference (3.11), I have performed several simulations as follows. I have computed a number of histograms and autocorrelation functions of displaced pel differences obtained by some existing motion estimation techniques. I chose reliable methods which made no assumptions as far as the statistical properties of the DPDs are concerned. In particular, I chose the method proposed by Horn and Schunck [41] implemented exactly as described in the paper, and also the method of Paquin and Dubois [73]. Each method was applied to the test images presented in Figs. 4.9 and 4.10. The histograms and the autocorrelation functions of displaced pel difference, are presented in Fig. 3.5 and Fig. 3.6, respectively. Note that the histograms in Fig. 3.5.a very closely resemble Gaussian distributions and those in Fig. 3.5.b are not too far from Gaussians too. The narrow peaks for zero DPD in Fig. 3.5.b are due to large stationary area (background) in the test image 4. Apart from this departure from "gaussianity" the shape of the histograms is quite close to a Gaussian distribution. Obviously other distributions could have been fitted to those histograms e.g., Laplacian, however due to mathematical tractability of the Gaussian distribution I chose this approximation as the DPD model. Note that the variance of such approximation depends on the image material and motion estimation technique used. As far as the autocorrelation functions are concerned (Fig. 3.6), it can be clearly seen that the impulse in the center of the plot is quite similar to the Dirac impulse and indicates near independence between the displaced pel differences. Again, to a first approximation it may be assumed that the displaced pel differences are independent random variables.

Based on above derivation for the simplified case, and on empirical observations, the independent, identically distributed discrete random variables drawn from the Gaussian distribution with variance σ^2, are proposed to model the displaced pel differences (3.11).

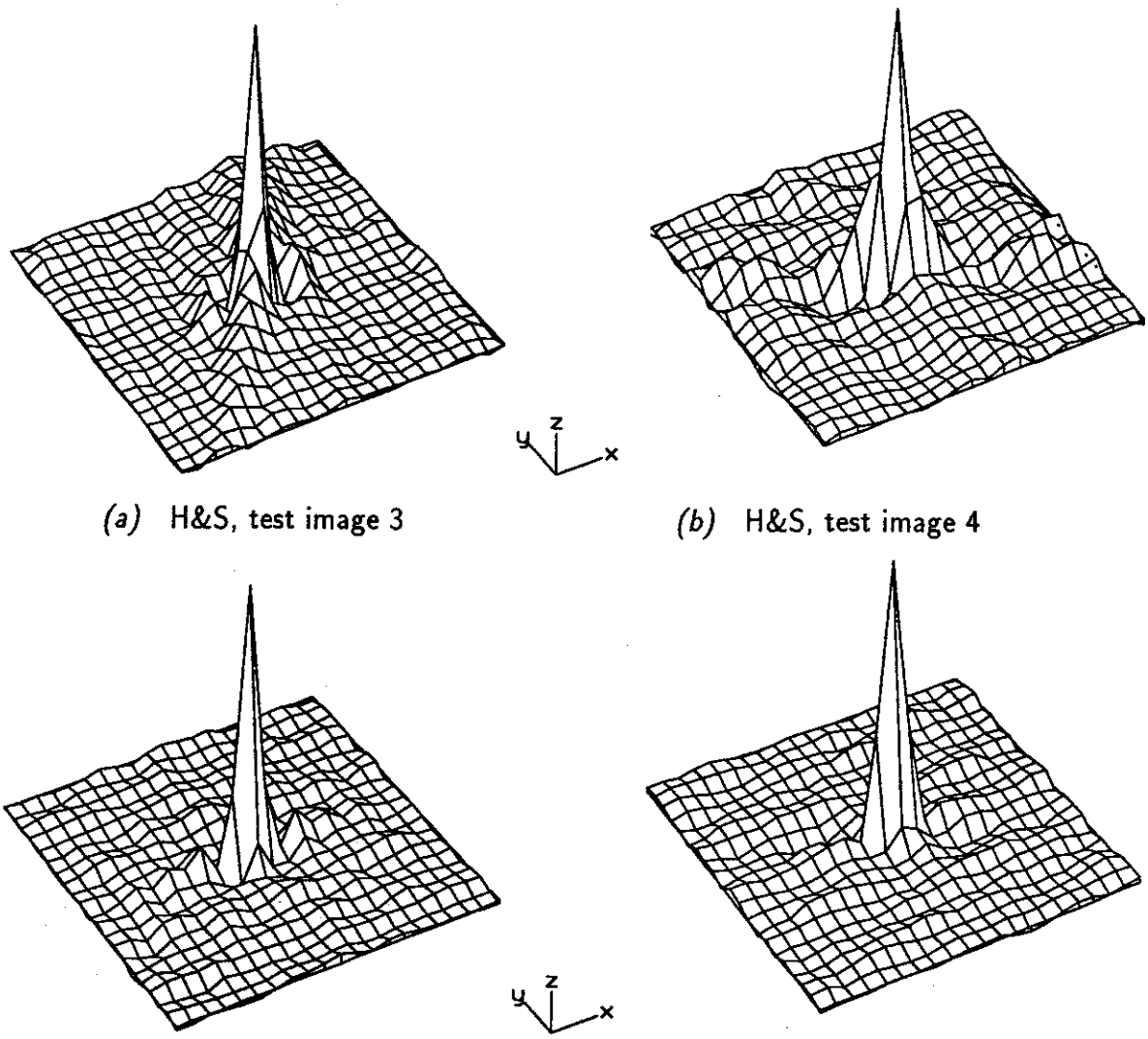


(a) test image 3



(b) test image 4

Fig. 3.5 Histogram of displaced pel difference obtained by algorithms proposed by Horn and Schunck [41] and by Paquin and Dubois [73] for the test images 3 and 4 (Section 4.7).



(a) H&S, test image 3

(b) H&S, test image 4

(c) P&D, test image 3

(d) P&D, test image 4

Fig. 3.6 Autocorrelation function of displaced pel difference obtained by algorithms proposed by Horn and Schunck [41] and by Paquin and Dubois [73] for the test images 3 and 4 (Section 4.7).

Consequently the equation (3.11) can be written in the following form:

$$g(\mathbf{x} + (1.0 - \Delta t) \cdot \boldsymbol{d}(\mathbf{x}, t), t_+) - g(\mathbf{x} - \Delta t \cdot \boldsymbol{d}(\mathbf{x}, t), t_-) = n_d(\mathbf{x}, t), \qquad (3.12)$$

where again g is discrete and $(\mathbf{x}, t)$ belongs to lattice $\Lambda_{\mathbf{d}}$, and $n_d(\mathbf{x}, t)$ is the discrete random variable described above. Since $n_d(\mathbf{x}, t)$ is an *iid* (independent identically distributed)

Gaussian random variable it follows immediately from (3.12) that:

$$P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \hat{\mathbf{d}}_t, G_{t_-} = g_{t_-}) = \prod_{i=1}^{M_d} p_{n_d}(\tilde{g}(\mathbf{x}_i + (1.0 - \Delta t) \cdot \hat{\mathbf{d}}(\mathbf{x}_i, t), t_+) - \tilde{g}(\mathbf{x}_i - \Delta t \cdot \hat{\mathbf{d}}(\mathbf{x}_i, t), t_-)) = (2\pi\sigma^2)^{-M_d/2} \cdot e^{-U_g(g_{t_+}|\hat{\mathbf{d}}_t, g_{t_-})/2\sigma^2}, \quad (3.13)$$

where p_{n_d} is a Gaussian distribution with variance σ^2, and $\tilde{g}$ denotes an intensity value at locations $(\mathbf{x}_i - \Delta t \cdot \hat{\mathbf{d}}(\mathbf{x}_i, t), t_-)$, $(\mathbf{x}_i - (1.0 - \Delta t) \cdot \hat{\mathbf{d}}(\mathbf{x}_i, t), t_+) \notin \Lambda_g$ obtained by some interpolation (e.g., bilinear, biquadratic). The resulting energy U_g is defined as follows:

$$U_g(g_{t_+}|\hat{\mathbf{d}}_t, g_{t_+}) = \sum_{i=1}^{M_d} [\tilde{g}(\mathbf{x}_i + (1.0 - \Delta t) \cdot \hat{\mathbf{d}}(\mathbf{x}_i, t), t_+) - \tilde{g}(\mathbf{x}_i - \Delta t \cdot \hat{\mathbf{d}}(\mathbf{x}_i, t), t_-)]^2. \quad (3.14)$$

To simplify the notation let $\tilde{r}$ denote the displaced pel difference:

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

3.4.3 Displacement field model

It can be observed that in most scenes motion is a result of position change of rigid or almost rigid bodies. After projection onto the image plane, that three-dimensional motion becomes a two-dimensional motion of some 2-D objects. The optical flow in such an image consists of patches of similar (orientation and length) vectors, with possible discontinuities at the motion boundaries. In this chapter I will discuss only a coarse approximation to such motion properties i.e., no motion discontinuities will be incorporated into the motion model. The discontinuous motion case will be described in Chapter 6. Therefore, it will be assumed here that motion fields are smooth functions of spatial position $\mathbf{x}$ (fixed t). In fact $\mathbf{d}(\mathbf{x}, t)$ is significantly smoother than the image itself. Based on this observation and on successful application of Markov random fields (MRFs) to image modeling [38], [26], I propose here to model the displacement field $\mathbf{d}(\mathbf{x}, t)$ by a 2-D vector MRF (VMRF) $\mathbf{D}_t$.

Since a VMRF differs from a scalar MRF only by the definition of a state (e.g., in R^2 instead of R), the properties described in (Section 3.2.2) hold for VMRFs too. Due to the Hammersley-Clifford theorem the Gibbs distribution (3.1) is a clear and effective way to

characterize a random field $\mathbf{D}_t$. A virtually unrestricted choice of potential functions $V_{\mathbf{d}}$[†] in this distribution provides a means of approximating local properties of $\mathbf{D}_t$. In this thesis the first- and second-order neighbourhood systems $\mathcal{N}_{\mathbf{d}}^1, \mathcal{N}_{\mathbf{d}}^2$ depicted in Figs. 3.3 and 3.4, will be investigated. The first-order system consists of 2-element vector cliques (Fig. 3.3.c,.d)

$$\begin{aligned} \mathcal{C}_h &= \left\{c_{\mathbf{d}} = \{\mathbf{x}_i, \mathbf{x}_j\} : \quad \mathbf{x}_i - \mathbf{x}_j = [T_{\mathbf{d}}^h, 0]\right\} \\ \mathcal{C}_v &= \left\{c_{\mathbf{d}} = \{\mathbf{x}_i, \mathbf{x}_j\} : \quad \mathbf{x}_i - \mathbf{x}_j = [0, T_{\mathbf{d}}^v]\right\} \\ \mathcal{C}_{hv} &= \{\mathcal{C}_h \cup \mathcal{C}_v\}, \qquad (\mathbf{x}_i, t), (\mathbf{x}_j, t) \in \Lambda_{\mathbf{d}} \end{aligned} \tag{3.15}$$

which represent only horizontal and vertical bindings ($\mathcal{C}_{\mathbf{d}}^1 = \mathcal{C}_{hv}$). The second-order system augments the above cliques with two-element diagonal, as well as three- and four-element cliques. In this thesis, however, neighbourhood $\mathcal{N}_{\mathbf{d}}^2$ with only horizontal and vertical (3.15), and diagonal cliques (Fig. 3.4 c,d,e,f)

$$\begin{aligned} \mathcal{C}_{45} &= \left\{c_{\mathbf{d}} = \{\mathbf{x}_i, \mathbf{x}_j\} : \quad \mathbf{x}_i - \mathbf{x}_j = [T_{\mathbf{d}}^h, T_{\mathbf{d}}^v]\right\} \\ \mathcal{C}_{-45} &= \left\{c_{\mathbf{d}} = \{\mathbf{x}_i, \mathbf{x}_j\} : \quad \mathbf{x}_i - \mathbf{x}_j = [-T_{\mathbf{d}}^h, T_{\mathbf{d}}^v]\right\}, \end{aligned} \tag{3.16}$$

is investigated ($\mathcal{C}_{\mathbf{d}}^2 = \{\mathcal{C}_{hv} \cup \mathcal{C}_{45} \cup \mathcal{C}_{-45}\}$).

The choice of the potential function $V_{\mathbf{d}}$ defined over a clique is crucial to characterization of the VMRF model. In general this potential could be a function of $\mathbf{d}$ and g, hence incorporating the image information into the motion model. In this chapter, however, I assume the statistical independence of the motion model from the images (or in other words, functional dependence of $V_{\mathbf{d}}$ only on $\mathbf{d}$). A more complex model, taking such a relationship into account, will be discussed in Chapter 6.

I choose the following potential function $V_{\mathbf{d}}$ over a two-element clique $c_{\mathbf{d}} \in \mathcal{C}_{\mathbf{d}}$:

$$V_{\mathbf{d}}(\mathbf{d}_t, c_{\mathbf{d}}) = V(\mathbf{d}(\mathbf{x}_i, t), \mathbf{d}(\mathbf{x}_j, t)) = \|\mathbf{d}(\mathbf{x}_i, t) - \mathbf{d}(\mathbf{x}_j, t)\|^2, \qquad c_{\mathbf{d}} = \{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{C}_{\mathbf{d}} \tag{3.17}$$

where $\|\cdot\|$ is a norm in R^2 e.g., L^2 (note that the summation in (3.2) is now over i and j). This particular potential captures the smoothness of the displacement field process $\mathbf{D}_t$; for $\mathbf{d}(\mathbf{x}_i, t) = \mathbf{d}(\mathbf{x}_j, t)$ the potential is zero, and the probability of such a configuration is

---

[†] The general concepts of neighbourhood, neighbourhood system, clique, set of cliques, potential etc., presented in Section 3.2.1 will be used now for random field $\mathbf{D}_t$ and will be identified by subscript $\mathbf{d}$, for example $\Lambda_{\mathbf{d}}$, $\mathcal{N}_{\mathbf{d}}$, $c_{\mathbf{d}}$, $\mathcal{C}_{\mathbf{d}}$, $V_{\mathbf{d}}$.

high, while any deviation from this equality causes a smooth reduction in the probability of such an arrangement.

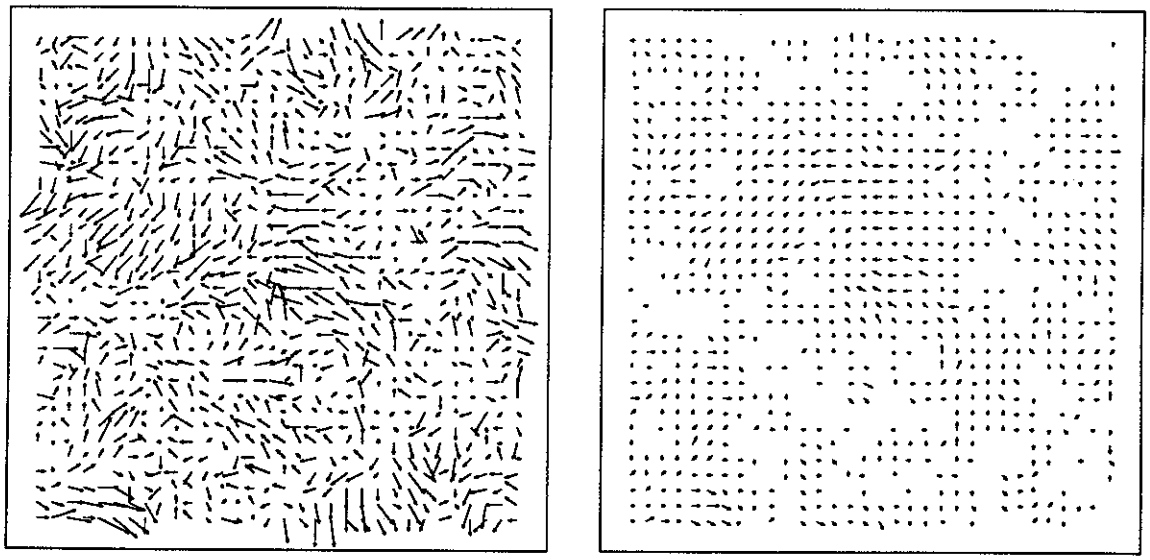
In this chapter it will be assumed that the knowledge of a single image (e.g., $G_{t\_}$) does not provide any information to computation of the motion field $\mathbf{d}(\mathbf{x},t)$. This is a coarse approximation, since sometimes knowledge of one image may help in estimating a motion field (for example in a uniform intensity area it is unlikely to have an abrupt change in length or orientation of motion vectors). This approximation will be improved upon in Chapter 6 by using a more complex motion model. With the above assumption $\mathbf{D}_t$ and $G_{t\_}$ are independent and the probability $P(\mathbf{D}_t{=}\mathbf{d}_t|G_{t\_}{=}g_{t\_})$ will have the following Gibbs form:

$$P(\mathbf{D}_t = \mathbf{d}_t|G_{t\_} = g_{t\_}) = \pi(\mathbf{d}_t) = \frac{1}{Z_d}e^{-U_\mathbf{d}} = \frac{1}{Z_d}e^{-\sum_{c_\mathbf{d}\in\mathcal{C}_\mathbf{d}} V_\mathbf{d}(\mathbf{d}_t,c_\mathbf{d})/\beta_\mathbf{d}}, \qquad (3.18)$$

where the set of cliques $\mathcal{C}_\mathbf{d}$ is equal to $\mathcal{C}_\mathbf{d}^1$ for $\mathcal{N}_\mathbf{d}^1$ or $\mathcal{C}_\mathbf{d}^2$ for $\mathcal{N}_\mathbf{d}^2$. For the continuous state-space case the probability density $p(\mathbf{d}_t|G_{t\_}{=}g_{t\_})$ has the same form as that in (3.18).

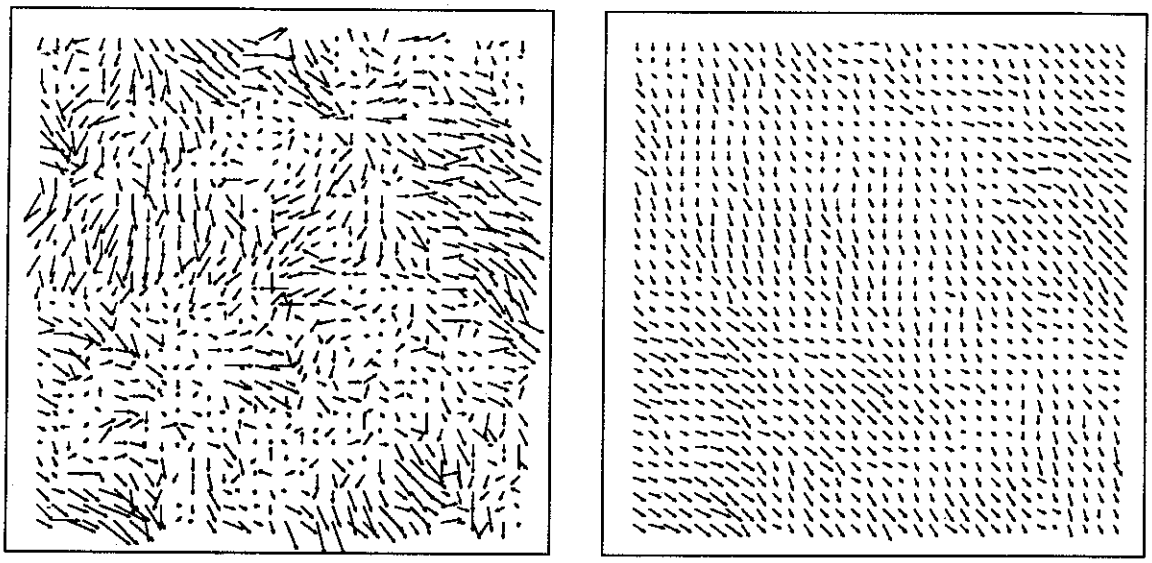
To demonstrate that VMRF is a valid model for displacement fields, I generated several unconstrained VMRF samples (unconstrained by image intensities). The samples were generated by the *Gibbs sampler* (to be described in Section 4.2.3) from the *a priori* Gibbs distribution (3.18) with potential (3.17). Both first- and second-order neighbourhood systems were used, as well as different values of the parameter $\beta_\mathbf{d}$, reflecting the sample field "activity" (the higher $\beta_\mathbf{d}$, the more active or chaotic the sample field). The *Gibbs sampler* was supplied with a random configuration as the initial state. After sufficient time it produced only the most likely samples, which for small value of $\beta_\mathbf{d}$ were homogeneous fields of identical vectors (due to the form of the potential function (3.17)). The average length of these vectors should be approximately equal to the expected value of the initial configuration. The state space $\mathcal{S}_\mathbf{d}^l$ for each vector was discrete with maximum displacement d_{max}=2.0 and N_d=17 possible levels in each direction. The generating distribution for the initial state was independent for both vector components and piecewise-uniform i.e., uniform over $[-d_{max}, 2m]$ and $(2m, d_{max}]$ with probability 0.5 of falling into any of the intervals. It can be easily verified that such distribution has mean equal to m, and for $m = 0$ it becomes a uniform distribution over the range $[-d_{max}, d_{max}]$.

Figs. 3.7, 3.8, 3.9 and 3.10 show VMRF samples for two different values of $\beta_\mathbf{d}$ after 50 iterations of the *Gibbs sampler*. The sample from Fig. 3.7 has been generated with the



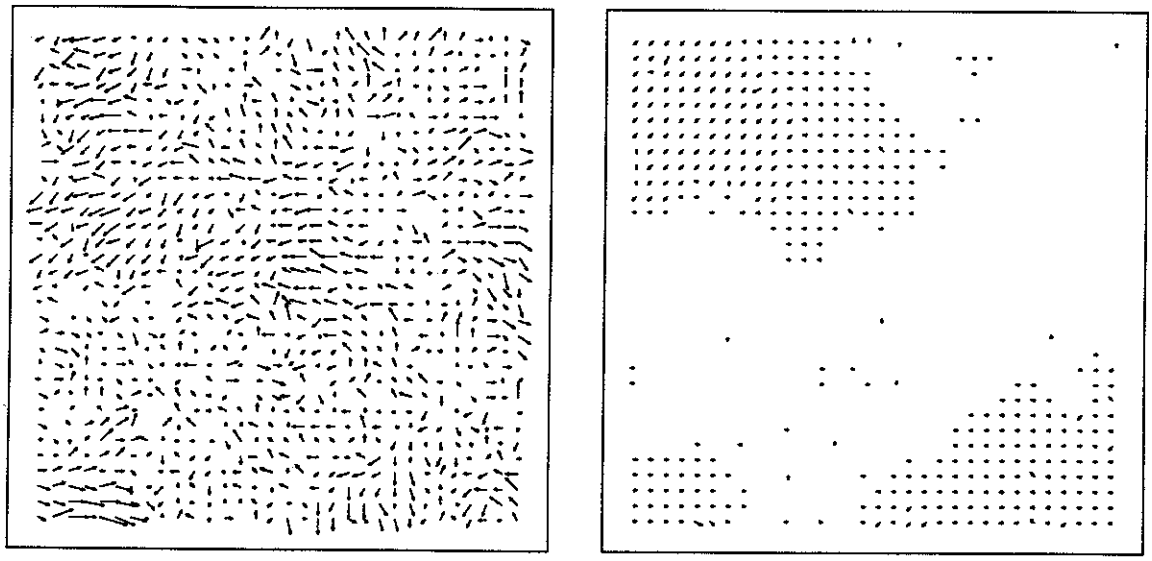
(a) $\beta_{\mathbf{d}}$=1.0 *(b)* $\beta_{\mathbf{d}}$=0.1

Fig. 3.7 VMRF samples for potential function (3.17), neighbourhood system $\mathcal{N}_{\mathbf{d}}^{1}$ and two values of parameter $\beta_{\mathbf{d}}$, initialized by a uniformly distributed field with mean m=(0.0,0.0).



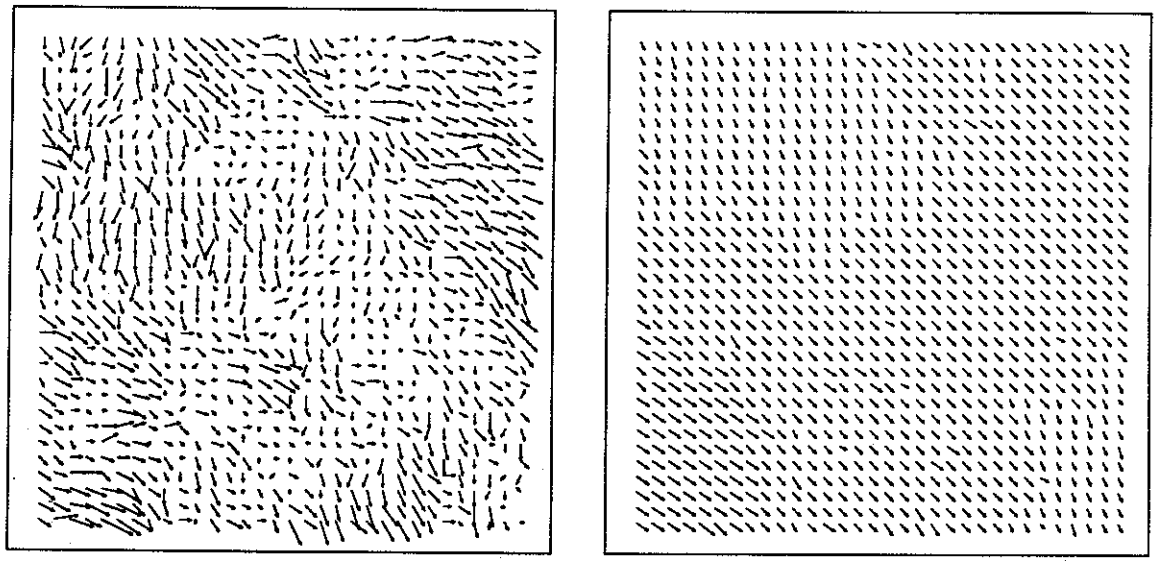
(a) $\beta_{\mathbf{d}}$=1.0 *(b)* $\beta_{\mathbf{d}}$=0.1

Fig. 3.8 VMRF samples for potential function (3.17), neighbourhood system $\mathcal{N}_{\mathbf{d}}^{1}$ and two values of parameter $\beta_{\mathbf{d}}$, initialized by a piecewise-uniformly distributed field with mean m=(0.5,0.5).



(a) $\beta_{\mathbf{d}}$=1.0 *(b)* $\beta_{\mathbf{d}}$=0.1

Fig. 3.9 VMRF samples for potential function (3.17), neighbourhood system $\mathcal{N}_{\mathbf{d}}^2$ and two values of parameter $\beta_{\mathbf{d}}$, initialized by a uniformly distributed field with mean m=(0.0,0.0).



(a) $\beta_{\mathbf{d}}$=1.0 *(b)* $\beta_{\mathbf{d}}$=0.1

Fig. 3.10 VMRF samples for potential function (3.17), neighbourhood system $\mathcal{N}_{\mathbf{d}}^2$ and two values of parameter $\beta_{\mathbf{d}}$, initialized by a piecewise-uniformly distributed field with mean m=(0.5,0.5).

first-order neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$ from a uniformly distributed initial configuration with mean m=(0.0,0.0). For $\beta_{\mathbf{d}}$=1.0 the sample field consists of quite random vectors, only locally "smooth", while for $\beta_{\mathbf{d}}$=0.1 the vectors are well "ordered" but much shorter. The sample field from Fig. 3.8 has been also produced with the first-order neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$ but from a piecewise-uniformly distributed initial configuration with mean m=(0.5,0.5). Note that the influence of $\beta_{\mathbf{d}}$ is similar to that in Fig. 3.7, but the vectors for $\beta_{\mathbf{d}} = 0.1$ oscillate around the mean value of the initial configuration. Figs. 3.9 and 3.10 present similar results except that the second-order system $\mathcal{N}_{\mathbf{d}}^2$ with cliques from $\mathcal{C}_{\mathbf{d}}^2$ was used. Again the influence of $\beta_{\mathbf{d}}$ is similar as before, but the vectors are more ordered now. This is due to the increased size of the neighbourhood system (added diagonal cliques) in spite of still only two-element cliques being used.

The VMRF samples shown in the figures suggest that the values of $\beta_{\mathbf{d}}$ smaller than 1.0 give smooth displacement fields, and the values above 1.0 produce more "chaotic" fields. Thus, to model a slowly varying motion $\beta_{\mathbf{d}}$'s of the order of 0.1 should be used, however an exact value cannot be established.

3.5 A POSTERIORI PROBABILITY

Using equations (3.13) and (3.18) in (3.3) the following Gibbs form can be obtained for the posterior probability distribution:

$$\begin{aligned}\pi(\hat{\mathbf{d}}_t|G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) &= P(\mathbf{D}_t = \hat{\mathbf{d}}_t|G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \\ &= \frac{1}{Z} e^{-U(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+})}\end{aligned} \tag{3.19}$$

where Z is a new normalizing constant (incorporating the probability $P(G_{t_-}, G_{t_+})$ from (3.3), $(2\pi\sigma^2)^{-M_{\mathbf{d}}/2}$ from (3.13) and Z_d from (3.18)). The new energy function $U(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+})$ is defined as follows

$$U(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+}) = \lambda_g \cdot \sum_{i=1}^{M_{\mathbf{d}}} [\tilde{r}(\hat{\mathbf{d}}(\mathbf{x}_i, t), \mathbf{x}_i, t, \Delta t)]^2 + \lambda_{\mathbf{d}} \cdot U_{\mathbf{d}}(\hat{\mathbf{d}}_t), \tag{3.20}$$

and $\lambda_g = 1/2\sigma^2$, $\lambda_{\mathbf{d}} = 1/\beta_{\mathbf{d}}$. Again for the continuous state-space case, the mixed *a posteriori* probability density function has the same form as that in (3.19). The neighbourhood system for this new Gibbs distribution is the same as that for the *a priori* distribution

$P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-})$ since the DPD model is based on the independently distributed noise random variables. Should this model be correlated (e.g., filtered noise), the neighbourhood system would need to be redefined appropriately [26].

The exponential *a posteriori* probability distribution (3.19) results in the following form for the MAP estimation (3.4):

$$\min_{\hat{\mathbf{d}}_t} U(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+}) = \min_{\hat{\mathbf{d}}_t} [\lambda_g \cdot U_g(g_{t_+} | \hat{\mathbf{d}}_t, g_{t_+}) + \lambda_{\mathbf{d}} \cdot U_{\mathbf{d}}(\hat{\mathbf{d}}_t)] \tag{3.21}$$

Note that the energy function U in (3.21) consists of two terms. The first one U_g is a sum of squared DPDs over the entire image, and attempts to describe the matching problem of the data g_{t_-}, g_{t_+} by the motion field $\hat{\mathbf{d}}$. As suggested in Section 2.2, this correspondence problem is ill-posed. The second term $U_{\mathbf{d}}$ in (3.21) is responsible for conforming to the properties of some *a priori* model (Section 3.4.3).

The two-term formulation of the energy function (3.20) can be viewed as regularization of the original correspondence problem (DPD only), as defined in Tikhonov's formulation (Section 2.5). Then, $U_{\mathbf{d}}$ plays the role of a stabilizing functional and $\lambda_{\mathbf{d}}/\lambda_g$ is a regularization parameter. Hence, the Bayesian formulation comprises, as a specific case, the regularization, which has been frequently used in computer vision (Section 2.5).

In order for the conditional *a posteriori* probability of an estimate (3.19) to be high, the energy (3.20) must be low, hence this estimate must well explain the data and conform to the motion model characteristic properties. For the model proposed by potential (3.17) such a property is smoothness of a displacement field. Due to this additional constraint, the estimation problem becomes well-posed. Recall the example of motion estimation ill-posedness from Fig. 2.1. Now, with the displacement field smoothness required simultaneously with the data matching, it is easy to see that the translational motion (Fig. 2.1.b) will provide lower total energy (3.20), and should be chosen as the better (in the view of assumed model) solution.

In the formulation (3.21) the ratio $\lambda_{\mathbf{d}}/\lambda_g$ plays an important role weighting the confidence in the data and in the *a priori* model. A modification of λs has an effect on the estimator, however the magnitude of this effect is highly dependent on the data itself. Recall that σ^2 is a variance of the displaced pel difference (Gaussian) model. Its value can be

estimated given a displacement field, however the other parameter β_{d}, which characterizes displacement field "activity", is far more difficult to compute. This parameter embodies our prior expectation as to the degree of randomness which the estimator should incorporate. When MRFs are used in estimation of such observables as images or textures, β_{d} 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 like motion, such computation of β_{d} is not possible (at least up to now). It must be chosen *ad hoc*. Consequently there is no point in estimating σ^2 (λs may be chosen instead).

Appendix 3.A. DERIVATION OF MEC ESTIMATOR

In this appendix the optimal (discrete state-space) Bayesian estimator with respect to a positive definite cost functional Θ will be derived. Let this functional be of the following separable form (as proposed in Section 3.3.2):

$$\Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t) = \sum_{i=1}^{M_{\mathbf{d}}} \theta(\boldsymbol{d}(\mathbf{x}_i, t), \hat{\mathbf{d}}(\mathbf{x}_i, t)), \tag{3.A.1}$$

where $\theta(\cdot,\cdot)$ is a positive definite function, and $\boldsymbol{d}_t$, $\hat{\mathbf{d}}_t$ are again the true displacement field and its estimate, respectively. The optimal Bayesian estimator $\hat{\mathbf{d}}_t^* \in \mathcal{S}_d$ is a displacement field defined as follows:

$$E_{\boldsymbol{d}_t, g_{t_-}, g_{t_+}}[\Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t^*)] = \min_{\hat{\mathbf{d}}_t} E_{\boldsymbol{d}_t, g_{t_-}, g_{t_+}}[\Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t)], \tag{3.A.2}$$

where $E_{\boldsymbol{d}_t, g_{t_-}, g_{t_+}}[\cdot]$ is the ensemble expectation over all configurations of $\boldsymbol{d}_t$, g_{t_-} and g_{t_+} (t_- and t_+ are temporal positions of the preceding and the following images, respectively). Note that an estimator with the smallest expected cost (in the sense of Θ) is being sought over all possible configurations of the data g and of the true displacements $\boldsymbol{d}$.

The following theorem, which is a 2-D version of that proposed by Marroquin [62], extends his results to the vector MRFs.

Theorem: Given a positive definite cost functional Θ, the optimal (minimum expected cost) estimate of a vector field $\boldsymbol{d}_t$ with respect to Θ can be obtained by minimizing independently the marginal expected cost for each vector, i.e.,

$$\hat{\mathbf{d}}^*(\mathbf{x}_i, t) = \mathbf{q} \in \mathcal{S}'_{\mathbf{d}} : \sum_{\mathbf{r} \in \mathcal{S}'_{\mathbf{d}}} \theta(\mathbf{r}, \mathbf{q}) \cdot P(\mathbf{D}(\mathbf{x}_i, t) = \mathbf{r} | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \leq \sum_{\mathbf{r} \in \mathcal{S}'_{\mathbf{d}}} \theta(\mathbf{r}, \mathbf{p})) \cdot P(\mathbf{D}(\mathbf{x}_i, t) = \mathbf{r} | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+})$$

for all $\mathbf{p} \neq \mathbf{q}$ ($\mathbf{p} \in \mathcal{S}'_{\mathbf{d}}$), and for $i = 1, ..., M_{\mathbf{d}}$.

Proof:

Since the fields $\boldsymbol{d}_t, g_{t_-}, g_{t_+}$ have discrete finite state-spaces the expectation will be expressed in terms of finite summations rather than integrals (the continuous state-space case). Then, using the Bayes rule the expectation (3.A.2) can be expressed as follows:

$$\begin{aligned} E_{\boldsymbol{d}_t, g_{t_-}, g_{t_+}}[\Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t)] = & \sum_{\substack{\boldsymbol{d}_t \in \mathcal{S}_{\mathbf{d}} \\ g_{t_-}, g_{t_+} \in \mathcal{S}_g}} \Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t) \cdot P(\mathbf{D}_t = \boldsymbol{d}_t, G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \\ & \sum_{g_{t_-}, g_{t_+} \in \mathcal{S}_g} \Big[\sum_{\boldsymbol{d}_t \in \mathcal{S}_{\mathbf{d}}} \Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t) \cdot P(\mathbf{D}_t = \boldsymbol{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \Big] \\ & \cdot P(G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \\ & \sum_{g_{t_-}, g_{t_+} \in \mathcal{S}_g} Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+}) \cdot P(G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \\ & E_{g_{t_-}, g_{t_+}}[Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+})], \end{aligned} \tag{3.A.3}$$

where the functional Q_1 is defined as:

$$Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_{t_+}) = \sum_{\boldsymbol{d}_t \in \mathcal{S}_{\mathbf{d}}} \Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t) \cdot P(\mathbf{D}_t = \boldsymbol{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}). \tag{3.A.4}$$

Since Θ is positive definite and P is a probability measure, Q_1 is positive definite too. The estimator $\hat{\mathbf{d}}_t$ is a mapping from space $\mathcal{S}_g \times \mathcal{S}_g$ into space $\mathcal{S}_{\mathbf{d}}$, and also is a function of the data g_{t_-}, g_{t_+}[†]. In other words, $\hat{\mathbf{d}}_t$ depends on the data g_{t_-}, g_{t_+} but the choice of $\hat{\mathbf{d}}_t$ for different data is independent. The same independence applies to Q_1. The expectation in (3.A.3) is a linear combination of positive definite functionals Q_1, hence the minimization can be performed with respect to the functionals Q_1 independently for different data:

$$\begin{aligned} \min_{\hat{\mathbf{d}}_t} E_{\boldsymbol{d}_t, g_{t_-}, g_{t_+}}[\Theta(\boldsymbol{d}_t, \hat{\mathbf{d}}_t)] &= \min_{\hat{\mathbf{d}}_t} E_{g_{t_-}, g_{t_+}}[Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_+)] \\ &= E_{g_{t_-}, g_{t_+}}[\min_{\hat{\mathbf{d}}_t} Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_+)]. \end{aligned} \tag{3.A.5}$$

Using in (3.A.5) the definition of Q_1 from (3.A.4) and the definition of separable Θ from (3.A.1), and then appropriately expanding and grouping the summations (the summation with respect to field $\boldsymbol{d}_t$ is expanded as a multiple summation with respect to its individual

[†] For simplicity of notation the dependence of estimate $\hat{\mathbf{d}}_t$ and of the optimal estimate $\hat{\mathbf{d}}_t^*$ on the data g_{t_-}, g_{t_+} has been omitted.

vectors) it follows that:

$$\begin{aligned}
&\min_{\hat{\mathbf{d}}_t} Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_+) = \\
&\min_{\hat{\mathbf{d}}_t} \sum_{\boldsymbol{d}_t \in \mathcal{S}_\mathbf{d}} \sum_{i=1}^{M_\mathbf{d}} \theta(\boldsymbol{d}(\mathbf{x}_i,t), \hat{\mathbf{d}}(\mathbf{x}_i,t)) \cdot P(\mathbf{D}_t = \boldsymbol{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \\
&\min_{\hat{\mathbf{d}}_t} \sum_{i=1}^{M_\mathbf{d}} \sum_{\mathbf{r} \in \mathcal{S}'_\mathbf{d}} \sum_{\boldsymbol{d}_t : \boldsymbol{d}(\mathbf{x}_i,t) = \mathbf{r}} \theta(\mathbf{r}, \hat{\mathbf{d}}(\mathbf{x}_i,t)) \cdot P(\mathbf{D}_t = \boldsymbol{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \\
&\min_{\hat{\mathbf{d}}_t} \sum_{i=1}^{M_\mathbf{d}} \sum_{\mathbf{r} \in \mathcal{S}'_\mathbf{d}} \theta(\mathbf{r}, \hat{\mathbf{d}}(\mathbf{x}_i,t)) \cdot P(\mathbf{D}(\mathbf{x}_i,t) = \mathbf{r} | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}),
\end{aligned} \tag{3.A.6}$$

where the marginal conditional probability of displacement vector positioned at $(\mathbf{x}_i, t)$ is defined as follows:

$$P(\mathbf{D}(\mathbf{x}_i,t) = \mathbf{r} | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \sum_{\boldsymbol{d}_t : \boldsymbol{d}(\mathbf{x}_i,t) = \mathbf{r}} P(\mathbf{D}_t = \boldsymbol{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}).$$

Defining another functional Q_2 as

$$Q_2(\hat{\mathbf{d}}(\mathbf{x}_i,t), g_{t_-}, g_{t_+}) = \sum_{\mathbf{r} \in \mathcal{S}'_\mathbf{d}} \theta(\mathbf{r}, \hat{\mathbf{d}}(\mathbf{x}_i,t)) \cdot P(\mathbf{D}(\mathbf{x}_i,t) = \mathbf{r} | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}),$$

permits the following modification of the minimization (3.A.6)

$$\begin{aligned}
\min_{\hat{\mathbf{d}}_t} Q_1(\hat{\mathbf{d}}_t, g_{t_-}, g_+) &= \min_{\hat{\mathbf{d}}_t} \sum_{i=1}^{M_\mathbf{d}} Q_2(\hat{\mathbf{d}}(\mathbf{x}_i,t), g_{t_-}, g_{t_+}) \\
&= \sum_{i=1}^{M_\mathbf{d}} \min_{\hat{\mathbf{d}}(\mathbf{x}_i,t)} Q_2(\hat{\mathbf{d}}(\mathbf{x}_i,t), g_{t_-}, g_{t_+}).
\end{aligned} \tag{3.A.7}$$

The last line follows from the fact that the functional Q_2 is positive definite (θ is a positive definite function and P is a probability measure), and also that for every i, Q_2 is a function of only one estimate vector $\hat{\mathbf{d}}(\mathbf{x}_i,t)$. Finally, the minimization in (3.A.7) is equivalent to the inequality condition given in the theorem evaluated at each i. □

Note that the continuous state-space case is very similar; the theorem and the proof will hold with the summations over $\mathbf{d}_t$ and $\mathbf{r}$ replaced by integrals, and with probabilities P replaced by density functions p.

In order to complete the derivation of the MEC estimator, the function θ must be known explicitly. In this work I assume the following form for θ:

$$\theta(\boldsymbol{d}(\mathbf{x}_i,t), \hat{\mathbf{d}}(\mathbf{x}_i,t)) = \|\boldsymbol{d}(\mathbf{x}_i,t) - \hat{\mathbf{d}}(\mathbf{x}_i,t)\|^2, \tag{3.A.8}$$

where $\|\cdot\|$ is the L^2 norm, since it reflects the "goodness" of the estimate and also since it is mathematically tractable. Using the definition of θ and the above theorem, the optimal Bayesian estimator $\widehat{\mathbf{d}}^*(\mathbf{x}_i,t)=\mathbf{q}\in\mathcal{S}'_{\mathbf{d}}$ $(i=1,...,M_{\mathbf{d}})$ is defined as follows

$$\sum_{\mathbf{r}\in\mathcal{S}'_{\mathbf{d}}}\|\mathbf{r}-\mathbf{q}\|^2\cdot P(\mathbf{D}(\mathbf{x}_i,t)=\mathbf{r}|G_{t_-}=g_{t_-},G_{t_+}=g_{t_+})\leq$$
$$\sum_{\mathbf{r}\in\mathcal{S}'_{\mathbf{d}}}\|\mathbf{r}-\mathbf{p}\|^2\cdot P(\mathbf{D}(\mathbf{x}_i,t)=\mathbf{r}|G_{t_-}=g_{t_-},G_{t_+}=g_{t_+}),\quad \text{all } \mathbf{p}\in\mathcal{S}'_{\mathbf{d}},\mathbf{p}\neq\mathbf{q},$$

or after expanding the L^2 norm and applying some arithmetic:

$$\|\overline{\mathbf{r}}-\mathbf{q}\|^2\leq\|\overline{\mathbf{r}}-\mathbf{p}\|^2\qquad\forall\mathbf{p}\in\mathcal{S}'_{\mathbf{d}},\mathbf{p}\neq\mathbf{q},$$

where $\overline{\mathbf{r}}$ denotes the conditional expectation of $\mathbf{r}$ given the data. Hence the MEC estimator $\widehat{\mathbf{d}}^*(\mathbf{x}_i,t)=\mathbf{q}$ $(i=1,...,M_{\mathbf{d}})$ is equal to the following marginal conditional expectation:

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

Again, for the continuous state-space case ($\mathcal{S}'_{\mathbf{d}}=R$) appropriate MEC estimator can be expressed as follows:

$$\widehat{\mathbf{d}}^*(\mathbf{x}_i,t)=\int_{\mathbf{r}\in\mathcal{S}'_d}\mathbf{r}\Big[\int_{\mathbf{d}_t:\mathbf{d}(\mathbf{x}_i,t)=\mathbf{r}}p(\mathbf{d}_t|G_{t_-}=g_{t_-},G_{t_+}=g_{t_+})d\mathbf{d}_t\Big]d\mathbf{r}\quad\forall i.$$

Chapter 4

STOCHASTIC SOLUTION TO MOTION ESTIMATION

In Chapter 3 the motion estimation problem was formulated on the basis of stochastic processes theory. This chapter will discuss some solution methods. In particular, stochastic methods will be proposed to solve the MAP and the MEC estimation problems.

The chapter starts with a brief overview of *Monte Carlo* procedures, followed by a detailed description of two examples of such procedures: the *Metropolis algorithm* and the *Gibbs sampler*. Then, a general optimization method called *simulated annealing* is discussed and shown to be applicable to the MAP estimation, followed by the solution of the MEC estimation using the *Law of Large Numbers for Markov chains*. The following sections discuss a continuous state-space MAP estimation and the spatial image interpolation for computation of motion. Finally, four test images and numerous estimation results are presented.

4.1 MONTE CARLO METHODS

Monte Carlo methods constitute a branch of experimental mathematics concerned with experiments on random numbers. Out of two types of problems handled by Monte Carlo methods – probabilistic and deterministic (according to whether or not they are directly related to stochastic processes) – the former type will be considered in this thesis. The simplest Monte Carlo approach to the probabilistic problem is to observe random numbers, chosen in such a way that they directly simulate the physical random process of the original problem, and to infer the desired solution from the behaviour of these random numbers.

Most Monte Carlo work is concerned with estimating the unknown numerical value of a parameter of some distribution. The ultimate goal is to find an unbiased, minimum-variance estimator (linear or non-linear). There exist various approaches to Monte Carlo estimation [36] e.g., *crude* Monte Carlo, *stratified sampling*, *importance sampling*, *control variates*, *antithetic variates* etc.

The fundamental differences between the above approaches can be explained on the example of expected value estimation for some given distribution. Assume that this distribution is sufficiently complex to prevent standard analytical treatment. Then, Monte Carlo methods can be applied to obtain the mean.

In the *crude* Monte Carlo method, the estimate is obtained by random sampling of the argument domain, computation of the distribution function values at these points and computation of the estimate. The argument domain samples are drawn from the uniform distribution.

In *stratified sampling* the argument domain is divided into subintervals (strata) and each of these intervals is assigned a number of sample points. Subsequently, the crude Monte Carlo method is applied to each of the strata. The specification of the strata may be as simple as intervals of equal length. A better way, however, is to choose the strata so that the variation of distribution function is the same in each subinterval. The number of sample points per stratum should also be proportional to the variation of the function over this stratum. The relative efficiency (product of labour ratio and variance ratio) of stratified sampling is about 10 with respect to crude Monte Carlo.

Importance sampling also attempts to modify the argument domain sampling in order to provide higher reliability of the estimate. The idea is to concentrate the distribution of the sample points in those parts of the domain that are of highest "importance" instead of spreading them out evenly. In order not to bias the estimate, the appropriately modified distribution function (non-uniform distribution of the sample points is taken into account) is used in the estimation process rather than the unmodified one. The relative efficiency of importance sampling is about 10, also with respect to the crude Monte Carlo method.

Control variates is another method attempting to reduce the variation of the distribution function. A simple function (*control variate*) such that:

1. it can be treated in the usual analytical way (e.g., integration),
2. it closely "mimics" the distribution function under investigation,

is chosen. Obviously these two requirements are conflicting since the distribution function cannot be treated analytically (assumption at the beginning) and hence the control variate cannot follow it exactly. Assuming that the estimation process is a linear operator (e.g., integration) it can now be performed independently for the control variate (analytically) and for the difference between the distribution function and the control variate (crude Monte Carlo). The relative efficiency of the control variates method is about 30 with respect to the crude Monte Carlo method.

The *antithetic variates* method is similar to the control variates: it applies a control function to "mimic" the behaviour of the distribution function, but it also permutes the order of subintervals to make the variation of the sum of the rearranged functions as constant as possible. Its relative efficiency is highly dependent on the permutation type, but it ranges from about 30 to several thousands.

Only a few simple examples of Monte Carlo estimation have been briefly discussed above, however it is clear that Monte Carlo estimation is more an approach than an algorithm. The choice of the method is highly dependent on the problem; while some methods may be of insufficient speed and/or precision for a given application, the other ones may be very suitable for this particular case. In the following section two examples of Monte Carlo methods will be discussed.

4.2 STOCHASTIC RELAXATION

Stochastic relaxation is an iterative, site-replacement procedure for generating sample configurations from a MRF described by a Gibbs distribution π. The characteristic property of every stochastic relaxation scheme is that configuration changes resulting in energy increase are permitted. By contrast, deterministic algorithms only allow modifications of states leading to reduction of energy, and hence are mostly trapped in local minima.

4.2.1 General form of Metropolis algorithm

The *Metropolis algorithm* was originally invented to study certain problems in statistical

mechanics [65]. In particular, the goal was to calculate the properties of any substance which may be considered as composed of numerous interacting individual elements (e.g., molecules in a gas or atoms in binary alloys). The systems studied consisted of hundreds of identical, interacting components, which precluded the usual analytical treatment. The Metropolis algorithm, being an example of Monte Carlo estimation, permitted the study of the equilibrium properties (e.g., ensemble averages), time-evolution and low-temperature behaviour of such large systems.

Define Γ to be a state random variable, γ to be a state sample and Ω to be a discrete phase-space or state-space for γ (these definitions will be generalized in the following section).

If a system is in *thermal equilibrium* and its state γ has energy $U(\gamma)$, then the probability distribution in phase-space of the point representing γ is proportional to

$$e^{-U(\gamma)/\beta}, \tag{4.1}$$

where $\beta = k\mathrm{T}$, T is absolute temperature of the surroundings, and k is the Boltzmann constant. According to ergodic theory, the proportion of time that the system spends in state γ is also proportional to (4.1). Usually one needs to compute the expectation $<f>$ of some state-function $f(\gamma)$

$$<f>=\frac{\sum_\gamma f(\gamma)e^{-U(\gamma)/\beta}}{\sum_\gamma e^{-U(\gamma)/\beta}}. \tag{4.2}$$

$<f>$ could be evaluated by the crude Monte Carlo estimation, but the exponential factor means that major part of the summation is concentrated in a very small region of the phase-space. Therefore importance sampling should be used instead, and samples from the probability distribution

$$\pi(\gamma) = \frac{e^{-U(\gamma)/\beta}}{\sum_\gamma e^{-U(\gamma)/\beta}} \tag{4.3}$$

should be generated. Direct use of (4.3) is impossible since the denominator is unknown, but the algorithm proposed by Metropolis *et al.* [65] circumvents this problem.

Let $\tau = 0, 1, 2, ...$ be the discretized time in the evolution of a discrete chain. Let the state random variable at time $\tau - 1$ be $\Gamma(\tau - 1)$, and the proposed (new) state random

variable also at time $\tau - 1$ be $\Gamma^*(\tau - 1)$. Let ϖ be a sample state from the phase-space, and

$$\pi(\varpi)/\pi(\gamma) = e^{-U(\varpi)/\beta + U(\gamma)/\beta} = e^{-\Delta U/\beta},$$

where $\Delta U = U(\varpi) - U(\gamma)$ is the energy difference between the two states. With the above notation the general form of the Metropolis algorithm is given below.

General Metropolis algorithm

1. if $\Gamma(\tau - 1) = \gamma$ select a new state $\varpi \in \Omega$ from a probability distribution P such that
$$Q_{\gamma\varpi} = P(\Gamma^*(\tau - 1) = \varpi | \Gamma(\tau - 1) = \gamma) = P(\Gamma^*(\tau - 1) = \gamma | \Gamma(\tau - 1) = \varpi) = Q_{\varpi\gamma},$$
2. if $\Delta U \leq 0$ take $\Gamma(\tau) = \varpi$,
3. if $\Delta U > 0$ take
$$\Gamma(\tau) = \begin{cases} \varpi & \text{with probability } e^{-\Delta U/\beta} \\ \gamma & \text{with probability } 1 - e^{-\Delta U/\beta}. \end{cases}$$

The condition in 1. simply requires from probability distribution P that the transitional probabilities be symmetric. It remains to demonstrate that the above algorithm generates a Markov chain with the steady-state (limiting, equilibrium) probability distribution π or in other words, that it generates samples from the distribution π. First, it is necessary to show that π is a unique invariant measure of Markov chain Υ generated by the above algorithm. The proof, based on the work by Hammersley and Handscomb [36], is presented in Appendix 4.A. By the standard result from the theory of Markov chains (Theorem 1.3, Chapter 3 in [49]) it follows that since the state-space is finite, and also since the chain is irreducible ($\pi(\gamma) > 0$, $\forall \gamma \in \Omega$) and aperiodic by construction, the invariant distribution π is also the steady-state probability distribution of the Markov chain Υ.

Finally, note that the original Metropolis algorithm [65] is a special case of the algorithm presented above. The transition probabilities $Q_{\gamma\varpi}$ in that algorithm are independent of the

previous state $\Gamma(\tau-1)$ and also are uniform:

$$Q_{\gamma\varpi} = P(\Gamma^*(\tau-1)=\varpi|\Gamma(\tau-1)=\gamma) = P(\Gamma^*(\tau-1)=\varpi) = Q_{\varpi}$$

$$Q_{\varpi\gamma} = P(\Gamma^*(\tau-1)=\gamma|\Gamma(\tau-1)=\varpi) = P(\Gamma^*(\tau-1)=\gamma) = Q_{\gamma}$$

$$Q_{\varpi} = Q_{\gamma},$$

hence satisfying the symmetry requirement.

4.2.2 Metropolis algorithm for motion estimation

The Metropolis algorithm described in the previous section can be adapted to generating random vectors (2-D random variables), and used later in motion estimation through the joint probability distribution (3.19) which is Gibbsian.

To accommodate the more complex case of motion estimation the symbols $\Gamma, \gamma, \varpi, \Omega$ introduced in the previous section will now be redefined. Let Υ be a chain (discrete-time process) with the state-space (set of all possible configurations) Ω defined as follows:

$$\Omega = \{\gamma = (\gamma_1, ..., \gamma_{M_{\mathbf{d}}}): \gamma_i = \mathbf{d}(\mathbf{x}_i, t) \in \mathcal{S}'_{\mathbf{d}}, \text{ all } i\} = (\mathcal{S}'_{\mathbf{d}})^{M_{\mathbf{d}}}.$$

Clearly, γ is a possible configuration of chain Υ, and is also a set of individual states (vectors) γ_i assigned to spatial positions $\mathbf{x}_i$. γ can also be viewed as a sample motion field from the state-space (phase-space) Ω. Let ϖ be a single sample state (vector) from $\mathcal{S}'_{\mathbf{d}}$. Define $\gamma^{(\varpi,j)}$ to be a configuration identical to configuration γ except at position j where its value is ϖ i.e.,

$$\gamma^{(\varpi,j)} = (\gamma_1, ..., \gamma_{j-1}, \varpi, \gamma_{j+1}, ..., \gamma_{M_{\mathbf{d}}}), \qquad \text{for some } \varpi \in \mathcal{S}'_{\mathbf{d}}.$$

Let also $\Gamma = \{\Gamma_1, \Gamma_2, ..., \Gamma_{M_{\mathbf{d}}}\}$ be a set of random vectors (bivariates) or a random field with possible states in Ω.

The Metropolis algorithm produces a Markov chain such that after long enough evolution its states are distributed according to the invariant measure π. The values of the time-indexed random field $\Gamma(\tau)$ of this chain are complete motion fields i.e., $\Gamma(\tau)=\gamma = \{\gamma_1, \gamma_2, ..., \gamma_{M_{\mathbf{d}}}\}$. After sufficiently long evolution of the chain Υ motion field samples of higher probability will occur more frequently than those of low probability. The generated states will contain valuable information as to the statistical properties of the motion

fields. The subsequent investigations of $\pi(\gamma)$ will also apply if the conditional distribution $\pi(\gamma|G_{t_-}=g_{t_-}, G_{t_+}=g_{t_+})$ is used instead. Since the single state γ corresponds to a field of vectors defined over a lattice, it is not clear how to select a new candidate state. If two or more vectors are modified at a time, their effects might counteract, however when one vector is modified at a time, only its impact on the energy value counts. Hence, in order to propose a candidate motion field, first, a random location (uniformly distributed) in this field is generated, and second, a random vector (uniformly distributed) is produced. Then, corresponding energies of the old and new states are calculated and compared. If the new state reduces the system energy then it is unconditionally accepted, otherwise it is accepted with probability $e^{-\Delta U/\beta}$. The Metropolis algorithm for motion estimation is given below.

Metropolis algorithm for motion estimation

1. generate a random uniformly distributed spatial position j $((\mathbf{x}_j, t) \in \Lambda_{\mathbf{d}})$ i.e., such that:

$$P(\mathbf{x}_1) = P(\mathbf{x}_2) = ... = P(\mathbf{x}_j) = ... = P(\mathbf{x}_{M_{\mathbf{d}}}),$$

   where P is a probability,
2. for location j generate a uniformly distributed vector ϖ so that the new state is $\gamma^{(\varpi,j)}$,
3. compute the energy increment $\Delta U = U(\gamma^{(\varpi,j)}) - U(\gamma)$ resulting from changing the state of j-th vector from γ_j to ϖ,
4. if $\Delta U \leq 0$ change the state i.e., set $\Gamma(\tau) = \gamma^{(\varpi,j)}$,
5. if $\Delta U > 0$ change the state with probability $e^{-\Delta U/\beta}$ i.e.,

$$\Gamma(\tau) = \begin{cases} \gamma^{(\varpi,j)} & \text{with probability } e^{-\Delta U/\beta} \\ \gamma & \text{with probability } 1 - e^{-\Delta U/\beta}. \end{cases}$$

The energy increment, fundamental to the Metropolis algorithm, can be also expressed in terms of displacement vectors $\mathbf{d}(\mathbf{x}_i, t)$. Let $z \in \mathcal{S}'_{\mathbf{d}}$ $(z = \varpi)$ denote some new proposed vector (state) at spatial position n_τ. Recall that only one vector is modified at a time, while the other ones remain unchanged. Then, using the energy function (3.20) computed

in Section 3.5, the energy increment at n_τ can be expressed as follows:

$$\Delta U^{n_\tau} = \lambda_g \cdot \left[(\tilde{r}(z, \mathbf{x}_{n_\tau}, t, \Delta t))^2 - (\tilde{r}(\mathbf{d}(\mathbf{x}_{n_\tau}, t), \mathbf{x}_{n_\tau}, t, \Delta t))^2\right] + \lambda_{\mathbf{d}} \cdot \Big[\sum_{i:\mathbf{x}_i \in \eta_{\mathbf{d}}(\mathbf{x}_{n_\tau})} V(z, \mathbf{d}(\mathbf{x}_i, t)) - \sum_{i:\mathbf{x}_i \in \eta_{\mathbf{d}}(\mathbf{x}_{n_\tau})} V(\mathbf{d}(\mathbf{x}_{n_\tau}, t), \mathbf{d}(\mathbf{x}_i, t))\Big]. \quad (4.4)$$

If the new vector z reduces the energy U ($\Delta U < 0$), then it is accepted unconditionally. If $\Delta U > 0$, the vector z is accepted with probability $e^{-\Delta U/\beta}$. Hence, the larger the ΔU, the less likely is the acceptance of such a state.

4.2.3 Gibbs sampler

The *Gibbs sampler* is another method of generating samples from the Gibbs distribution π (3.1). Unlike the Metropolis algorithm, which was proposed in 1953, the Gibbs sampler is a relatively new technique. It was developed in the early 1980's by Geman and Geman [26]. It can be classified, with respect to the Monte Carlo methods, as an example of importance sampling, since rather than producing uniformly distributed states of a Markov chain, it generates samples from the Gibbs distribution. After sufficiently long evolution of this chain, the Gibbs sampler, like the Metropolis algorithm, produces more frequently such states which have higher probability of occurrence and less frequently those of lower likelihood.

The following description of the Gibbs sampler will be presented directly in the context of VMRFs since originally [26] it was intended to generate MRF samples, which was not the case for the Metropolis algorithm. Let the symbols $\Upsilon, \Gamma, \gamma, \varpi, \Omega$ be defined in the same way as in Section 4.2.2. Let τ denote again the time index in the evolution of the Markov chain Υ. Then, the evolution $\Gamma(\tau - 1) \rightarrow \Gamma(\tau)$ of the Gibbs sampler is described by the following relationship:

$$P(\Gamma_i(\tau) = \gamma_i, \forall i) = \pi(\Gamma_{n_\tau} = \gamma_{n_\tau} | \Gamma_j = \gamma_j, j \neq n_\tau) \cdot P(\Gamma_j(\tau - 1) = \gamma_j, j \neq n_\tau), \qquad (i, j : (\mathbf{x}_i, t), (\mathbf{x}_j, t) \in \Lambda_{\mathbf{d}}), \quad (4.5)$$

where n_τ is the spatial position at which the replacement takes place at time τ. It is clear from (4.5) that at each epoch only one site undergoes a possible change, hence configurations $\Gamma(\tau - 1)$ and $\Gamma(\tau)$ are either identical (no change) or they differ at coordinate n_τ. A new state is chosen at coordinate n_τ by drawing a sample from the local conditional

characteristics of distribution π. More specifically, a state $\gamma_{n_\tau} \in \mathcal{S}_{\mathbf{d}}$ is chosen from the conditional distribution of Γ_{n_τ} given the observed states of the neighbouring sites $\Gamma_i(\tau - 1)$ $\forall i : \mathbf{x}_i \in \eta_{\mathbf{d}}(\mathbf{x}_{n_\tau})$.

Hence, the time evolution of the chain Υ can be defined by the transition probability matrix $\Xi(\tau)$ at time τ. This matrix is everywhere zero except for the entries (ζ, γ) (ζ - row, γ - column) such that $\zeta = \gamma^{(\varpi, n_\tau)}$ for some $\varpi \in \mathcal{S}'_{\mathbf{d}}$:

$$\Xi_{\zeta,\gamma}(\tau) = \begin{cases} \pi(\Gamma_{n_\tau} = \gamma_{n_\tau} | \Gamma_j = \gamma_j, j \neq n_\tau), & \text{if } \zeta = \gamma^{(\varpi, n_\tau)} \text{ for some } \varpi \in \mathcal{S}'_{\mathbf{d}}, n_\tau \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

From the above construction of the chain Υ it follows that it is a Markov chain. This chain has a finite state-space, and is aperiodic and irreducible ($\pi(\gamma) > 0$ for all $\gamma \in \Omega$). The transition probability matrix Ξ of the chain Υ is time-dependent (hence Υ is non-homogeneous) which can be demonstrated as follows. Consider the transition from state $\zeta \in \mathcal{S}_{\mathbf{d}}$ to another state $\gamma \in \mathcal{S}_{\mathbf{d}}$ at time τ. Assume, without loss of generality, that $\zeta = \gamma^{(\varpi, n_\tau)}$ for some $\varpi \in \mathcal{S}'_{\mathbf{d}}$ such that $\varpi \neq \zeta_{n_\tau}$, hence a change of state takes place at n_τ. Since every state from the state-space $\mathcal{S}'_{\mathbf{d}}$ has non-zero probability, this transition has non-zero probability: $\Xi_{\zeta,\gamma}(\tau) > 0$. Now, move to the time $\tau + 1$ and consider again the transition from ζ to γ. Since already $\zeta = \gamma^{(\varpi, n_\tau)}$ ($\varpi \neq \zeta_{n_\tau}$), in order to obtain the state γ the transition $\zeta_{n_\tau} \rightarrow \varpi$ at n_τ must take place. But at $\tau + 1$ only transition at coordinate $n_{\tau+1}$ is allowed, hence $\Xi_{\zeta,\gamma}(\tau+1) = 0$ while $\Xi_{\zeta,\gamma}(\tau) > 0$, and time-dependence of transition probability matrix Ξ follows.

It can be shown (Appendix 4.B) that the invariant distribution of the chain Υ is the Gibbs distribution π:

$$\pi(\gamma) = \big(\pi\Xi(\tau)\big)_\gamma = \sum_\zeta P(\Gamma(\tau) = \gamma, | \Gamma(0) = \zeta) \cdot \pi(\zeta). \tag{4.7}$$

The proof of convergence of the Gibbs sampler (convergence of the distribution of $\Gamma(\tau)$) is quite complex, and can be found in the paper by Geman and Geman [26] in the proof of Theorem A. It is also shown there that the equilibrium distribution of the Gibbs sampler is equal to its invariant distribution which by equation (4.7) is Gibbsian.

In order to completely specify the Gibbs sampler the transition probability matrix must be known. Since the equilibrium distribution is Gibbsian, it follows that (Bayes rule and the law of total probability):

$$\begin{aligned}\pi(\Gamma_{n_\tau} = \gamma_{n_\tau} | \Gamma_i = \gamma_i, i \neq n_\tau) &= \frac{\pi(\gamma)}{\pi(\Gamma_i = \gamma_i, i \neq n_\tau)} \qquad (i : (\mathbf{x}_i, t) \in \Lambda_{\mathbf{d}}) \\ &= \frac{\pi(\gamma)}{\sum_{\varpi \in \mathcal{S}'_{\mathbf{d}}} \pi(\gamma^{(\varpi, n_\tau)})}.\end{aligned} \tag{4.8}$$

The relationship (4.8) will also hold if the Gibbs distribution $\pi(\gamma)$ is replaced by a conditional Gibbs distribution. To simplify the notation γ's shall be used for motion vectors instead of $\mathbf{d}$'s. Then, the *a posteriori* probability distribution (3.19) can be expressed as follows:

$$\pi(\gamma | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \frac{1}{Z} \cdot e^{-U(\gamma, g_{t_-}, g_{t_+})}, \tag{4.9}$$

where the energy function U is defined as

$$U(\gamma, g_{t_-}, g_{t_+}) = \lambda_g \cdot \sum_{i=1}^{M_{\mathbf{d}}} [\tilde{\mathbf{r}}(\gamma_i, \mathbf{x}_i, t, \Delta t)]^2 + \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}} \in \mathcal{C}_{\mathbf{d}}} V_{\mathbf{d}}(\gamma, c_{\mathbf{d}}).$$

With the above notation the conditional probability (4.8) takes the following form:

$$\begin{aligned}&\pi(\Gamma_{n_\tau} = \gamma_{n_\tau} | \Gamma_i = \gamma_i, i \neq n_\tau, G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \\ &= \frac{\pi(\gamma | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+})}{\sum_{\varpi \in \mathcal{S}'_{\mathbf{d}}} \pi(\gamma^{(\varpi, n_\tau)} | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+})} \\ &= \frac{\frac{1}{Z} \cdot e^{-U(\gamma, g_{t_-}, g_{t_+})/\beta}}{\sum_{\varpi \in \mathcal{S}'_{\mathbf{d}}} \frac{1}{Z} \cdot e^{-U(\gamma^{(\varpi, n_\tau)}, g_{t_-}, g_{t_+})/\beta}}.\end{aligned} \tag{4.10}$$

The energy functions from the numerator and denominator can be decomposed as follows:

$$\begin{aligned}U(\gamma, g_{t_-}, g_{t_+}) =& \lambda_g \cdot \Big(\sum_{\substack{j=1 \\ j \neq n_\tau}}^{M_{\mathbf{d}}} [\tilde{\mathbf{r}}(\gamma_j, \mathbf{x}_j, t, \Delta t)]^2 + [\tilde{\mathbf{r}}(\gamma_{n_\tau}, \mathbf{x}_{n_\tau}, t, \Delta t)]^2 \Big) + \\ & \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}} : \mathbf{x}_{n_\tau} \notin c_{\mathbf{d}}} V_{\mathbf{d}}(\gamma, c_{\mathbf{d}}) + \\ & \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}} : \mathbf{x}_{n_\tau} \in c_{\mathbf{d}}} V_{\mathbf{d}}(\gamma, c_{\mathbf{d}})\end{aligned} \tag{4.11.a}$$

$$\begin{aligned}U(\gamma^{(\varpi, n_\tau)}, g_{t_-}, g_{t_+}) =& \lambda_g \cdot \Big(\sum_{\substack{j=1 \\ j \neq n_\tau}}^{M_{\mathbf{d}}} [\gamma_j, \tilde{\mathbf{r}}(\mathbf{x}_j, t, \Delta t)]^2 + [\tilde{\mathbf{r}}(\varpi, \mathbf{x}_{n_\tau}, t, \Delta t)]^2 \Big) + \\ & \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}} : \mathbf{x}_{n_\tau} \notin c_{\mathbf{d}}} V_{\mathbf{d}}(\gamma^{(\varpi, n_\tau)}, c_{\mathbf{d}}) + \\ & \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}} : \mathbf{x}_{n_\tau} \in c_{\mathbf{d}}} V_{\mathbf{d}}(\gamma^{(\varpi, n_\tau)}, c_{\mathbf{d}}).\end{aligned} \tag{4.11.b}$$

The first term in (4.11.b) is clearly independent of ϖ. Since the sum in the third term extends only over the cliques which do not contain $\mathbf{x}_{n_\tau}$, this term is also independent of ϖ. Hence, appropriate exponential factors can be extracted from under the summation in the denominator of (4.10) and cancelled with identical factors in the numerator. Then, the conditional probability (4.10) can be written as:

$$\pi(\Gamma_{n_\tau} = \gamma_{n_\tau} | \Gamma_i = \gamma_i, i \neq n_\tau, G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \frac{e^{-\lambda_g \cdot [\tilde{r}(\gamma_{n_\tau}, \mathbf{x}_{n_\tau}, t, \Delta t)]^2 - \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}}: \mathbf{x}_{n_\tau} \in c_{\mathbf{d}}} V_{\mathbf{d}}(\gamma, c_{\mathbf{d}})}}{\sum_{\varpi \in \mathcal{S}'_{\mathbf{d}}} e^{-\lambda_g \cdot [\tilde{r}(\varpi, \mathbf{x}_{n_\tau}, t, \Delta t)]^2 - \lambda_{\mathbf{d}} \cdot \sum_{c_{\mathbf{d}}: \mathbf{x}_{n_\tau} \in c_{\mathbf{d}}} V_{\mathbf{d}}(\gamma^{(\varpi, n_\tau)}, c_{\mathbf{d}})}}.$$

Since $\pi(\gamma | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+})$ the original notation using $\mathbf{d}$'s can be used now, and the final form of the conditional probability driving the Gibbs sampler is

$$P(\mathbf{D}(\mathbf{x}_{n_\tau}, t) = \hat{\mathbf{d}}(\mathbf{x}_{n_\tau}, t) | \mathbf{D}(\mathbf{x}_j, t) = \hat{\mathbf{d}}(\mathbf{x}_j, t), j \neq n_\tau, G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \frac{e^{-U_{\mathbf{d}}^{n_\tau}(\hat{\mathbf{d}}(\mathbf{x}_{n_\tau}, t) | \hat{\mathbf{d}}_t, g_{t_-}, g_{t_+})}}{\sum_{z \in \mathcal{S}'_{\mathbf{d}}} e^{-U_{\mathbf{d}}^{n_\tau}(z | \hat{\mathbf{d}}_t, g_{t_-}, g_{t_+})}}, \tag{4.12}$$

where the local (conditional) energy function $U_{\mathbf{d}}^i$ for displacement vector update is defined as

$$U_{\mathbf{d}}^i(z | \hat{\mathbf{d}}_t, g_{t_-}, g_{t_+}) = \lambda_g \cdot [\tilde{r}(z, \mathbf{x}_i, t, \Delta t)]^2 + \lambda_{\mathbf{d}} \cdot \sum_{j: \mathbf{x}_j \in \eta_{\mathbf{d}}(\mathbf{x}_i)} V(z, \hat{\mathbf{d}}(\mathbf{x}_j, t)). \tag{4.13}$$

Note that the partition function Z does not appear in the conditional probability expression (4.12), hence there is no need to evaluate it. In order to sample from this distribution the energy for each possible $z \in \mathcal{S}'_{\mathbf{d}}$ must be computed (Appendix 4.C). The more states there are in the state-space $\mathcal{S}'_{\mathbf{d}}$, the more computations are needed to evaluate (4.12). This approach will not obviously work for the continuous state-spaces $\mathcal{S}'_{\mathbf{d}}$. This case will be tackled via a different approach later.

Practical implementation of the Gibbs sampler in the case of the vector MRFs is more complex than that of scalar MRFs [26], [62], and can be found in Appendix 4.C.

4.3 SOLVING THE MAP ESTIMATION: SIMULATED ANNEALING

The stochastic relaxation schemes described in the previous sections can generate samples from VMRF $\mathbf{D}_t$ distributed according to the Gibbs measure $\pi(\mathbf{d}_t)$. Hence, the more likely samples will be frequently encountered and the less likely ones will show up rarely. This is not enough to find the state (or states) maximizing the *posterior* probability $P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \pi(\mathbf{d}_t)$. The ultimate goal would be to devise an algorithm generating samples from $\mathbf{D}_t$ according to a uniform distribution on the set of configurations attaining the global minimum of U. Such an algorithm, proposed for the first time independently by Kirkpatrick *et al.* [52] and by Černy, is generally known as *simulated annealing*, but other names like *Monte Carlo annealing*, *statistical cooling* or *probabilistic hill climbing* have also been used.

The simulated annealing algorithm is based on the analogy between the process of *annealing of solids* and the problem of solving large combinatorial optimization problems. In physics, annealing denotes a process in which the temperature of a solid in a *heat bath* is increased to a point at which all particles of the solid randomly arrange themselves in the liquid phase, followed by cooling through slowly lowering the temperature of the heat bath. If the initial (maximum) temperature is sufficiently high and the cooling is sufficiently slow, the particles attain the configuration of the minimum energy.

Since the thermal equilibrium of a system in a state γ with energy $U(\gamma)$ is described by the Boltzmann distribution (4.1), the annealing process can be described as follows. For every temperature value T the solid is allowed to reach the thermal equilibrium. While the temperature decreases, the Boltzmann distribution concentrates around the states of the lowest energy and eventually, once the temperature approaches zero, only the minimum energy states have non-zero probability of occurrence. If, however, the cooling is too rapid i.e., the solid is not allowed to reach the thermal equilibrium for each temperature value, defects can be "frozen" into the solid and metastable amorphous structures can be reached rather than the low energy crystalline lattice structure. In a process known as *quenching* the temperature of the heat bath is reduced instantaneously, resulting in particle freezing.

In simulated annealing the behaviour of the solid is simulated by generating sample

configurations from the Gibbs (Boltzmann) distribution with the energy function suitably crafted for given optimization problem, while the temperature T is replaced by the "temperature" parameter **T**, which is reduced according to some annealing schedule (e.g., logarithmic). In the reminder of this thesis the parameter **T** will be referred to as a temperature.

Since the transition probabilities depend on the temperature **T** two formulations of the algorithm can be distinguished:

1. a *homogeneous* algorithm described by a sequence of homogeneous Markov chains; each chain is generated at constant value of **T** which is changed only between subsequent chains,
2. an *inhomogeneous* algorithm described by a single inhomogeneous Markov chain; the value of **T** is changed between subsequent transitions.

In order to define the annealing schedule the following parameters have to be specified:

1. initial value of the temperature, $\mathbf{T}_0$,
2. final value of the temperature, $\mathbf{T}_f$,
3. length of Markov chains,
4. a rule φ for changing the temperature, $\mathbf{T}_n = \varphi(\mathbf{T}_0, n)$.

There is no clear advantage of one algorithm over the other. In the case of the homogeneous algorithm, the number of iterations at each value of **T** must be defined (this number will depend on the decrements of **T**). For the inhomogeneous simulated annealing the temperature decrement between subsequent transitions has to be specified. Fig. 4.1 shows block diagram of the inhomogeneous simulated annealing.

I will follow the approach of Geman and Geman [26]. They have used the inhomogeneous algorithm, and have proved (Theorem B), that if every replacement site is visited infinitely often (in practice it means that it is not omitted) and if

1. $\mathbf{T}(t) \to 0$ as $t \to \infty$, and
2. $\mathbf{T}(t) \geq M\Delta / \log t$ for all $t \geq t_0$ $(t_0 \geq 2)$,

then with time $t \to \infty$ the chain will converge to the global optimum for any starting configuration. In the above expressions M denotes the number of elements in one MRF sample (e.g., number of pixels in an image) and Δ is the largest absolute difference in energies associated with states differing at only one coordinate. Note that due to the logarithmic decrements of the temperature, in order to reduce the initial value $\mathbf{T}(t_0)$ by

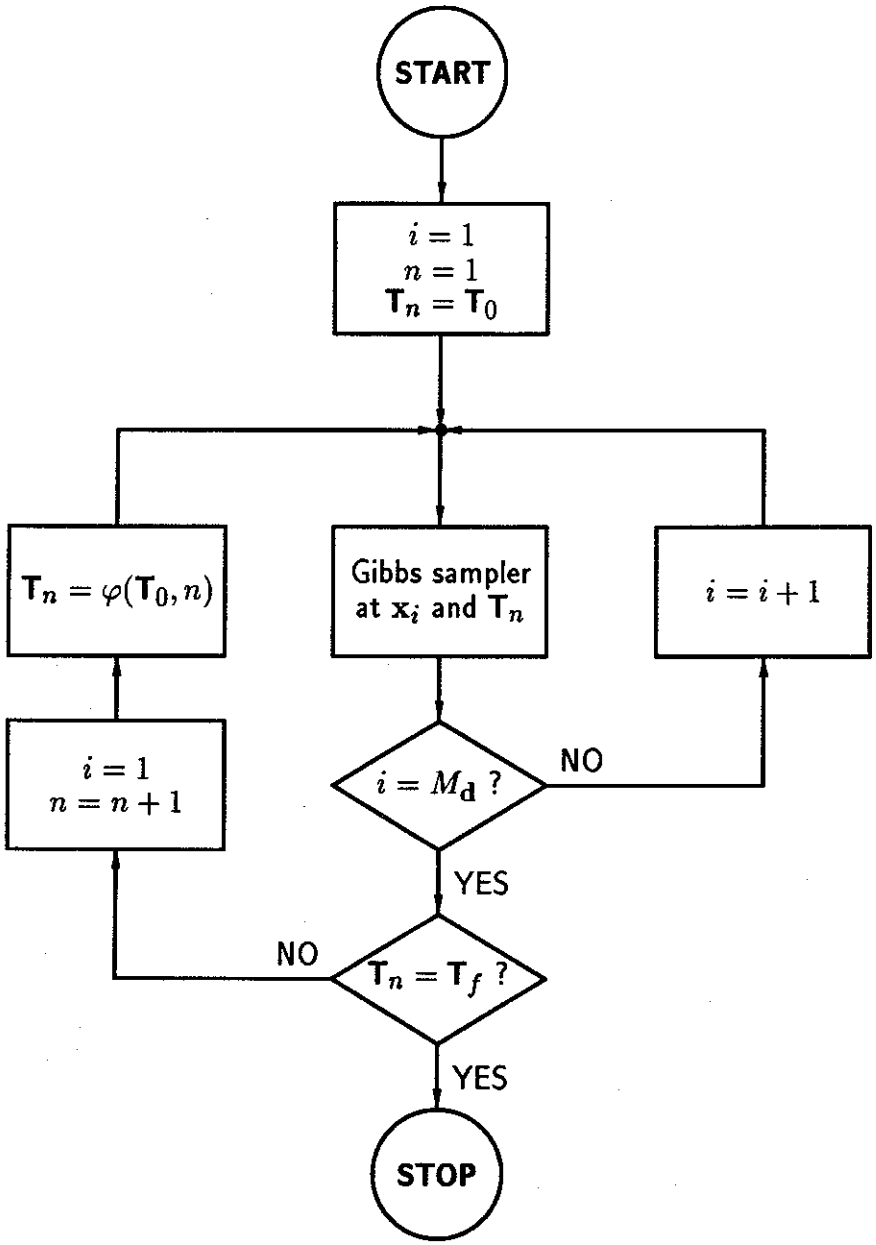


Fig. 4.1 Block diagram of the inhomogeneous simulated annealing algorithm based on the Gibbs sampler.

a factor of k, $(t_0)^k$ iterations (full scans of a displacement field) are required. Since the interesting range of temperatures (at which the structure is well-organized) falls below 1.0, the initial temperature value $M\Delta$ is quite impractical. More tight bounds on the initial temperature have been also obtained [89], but still are impractical. As a common practice the initial temperature is chosen *ad hoc*. Running several experiments with different values of that temperature, one can find the smallest one such that it does not degrade the solution

compared with the larger ones.

Another practical problem is posed by the logarithmic annealing schedule, as proposed by Geman and Geman. In order to obtain a "very organized" solution, the final temperature $\mathbf{T}_f$ must be quite small, independently of the choice of the initial temperature, for example of the order of 0.01. Unfortunately, the required number of iterations to attain the final temperature $\mathbf{T}_f$ from the initial $\mathbf{T}_0$ grows exponentially with $\mathbf{T}_0/\mathbf{T}_f$. If the initial temperature cannot be lowered any more without affecting the quality of the solution, then only the annealing schedule can be modified. In the experiments involving simulated annealing I will use either the optimal logarithmic schedule

$$\varphi(\mathbf{T}_0, n) = \mathbf{T}_0 \cdot \frac{\log 2}{\log (n+1)},$$

where n is the iteration number, or the exponential schedule

$$\varphi'(\mathbf{T}_0, n) = \mathbf{T}_0 \cdot a^{(n-1)},$$

where $0.0 < a < 1.0$. The exponential schedule allows to attain the final temperature in a reasonable number of iterations, but has to be used with caution since a large temperature decrement between iterations may trap the chain in a local minimum. Both schedules for initial temperature $\mathbf{T}_0$=1.0 are shown in Fig. 4.2 (a=0.980 for the exponential schedule). Note that after 200 iterations the logarithmic schedule attained the final temperature of 0.1307 while the exponential schedule gave 0.0179.

The optimality of the logarithmic schedule has been proved only for the Gibbs sampler, and hence it is not necessarily optimal for the Metropolis algorithm. As described in Sections 4.2.2 and 4.2.3 the Metropolis algorithm accepts or rejects states sampled from a uniform distribution, while the Gibbs sampler generates such states from a local conditional distribution. This suggests that the convergence rate of the Gibbs sampler towards the steady-state distribution is faster, however its complexity is higher. On the other hand due to its slower convergence rate, the Metropolis algorithm requires that the temperature modification be less frequent (between the temperature changes, a homogeneous Markov chain is generated), and hence more iterations be performed, but it is much simpler computationally. Comparison of the complexities per update for scalar MRFs can be found in [62],

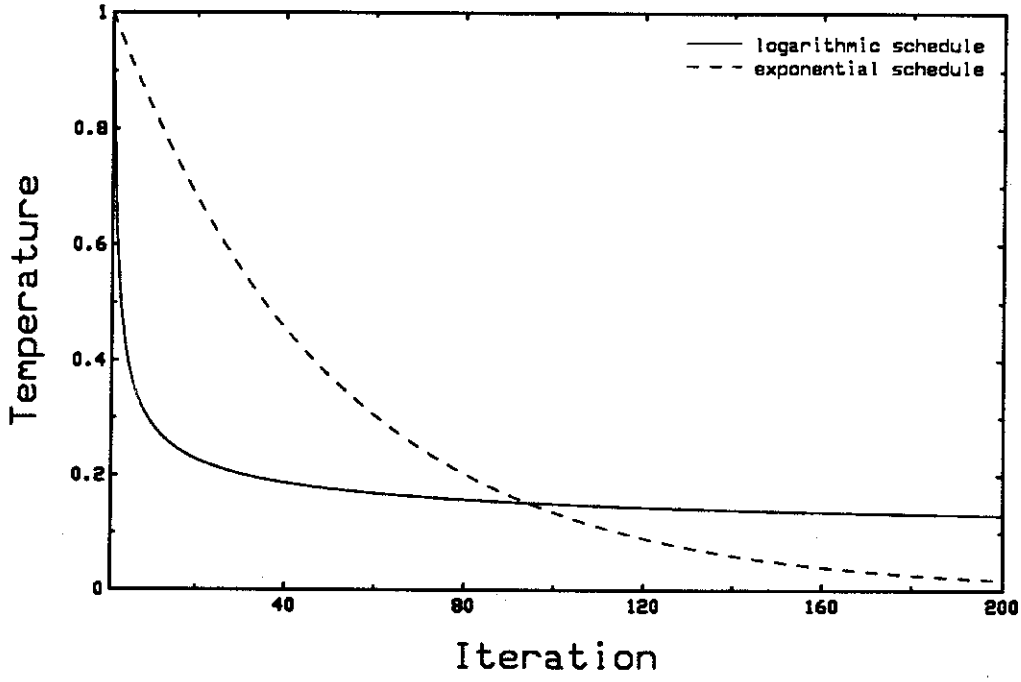


Fig. 4.2 The logarithmic and exponential (a=0.980) annealing schedules starting at T_0=1.0 over 200 iterations.

however an overall conclusion as to the total efficiency is difficult because the equivalent lengths of Markov chains cannot be easily established.

4.4 SOLVING THE MEC ESTIMATION: LLN FOR MARKOV CHAINS

It is clear from the construction of the Markov chain Υ that it is also a *regenerative process*, and that the time instants of the returns to a given state $\gamma \in \Omega$ constitute a *renewal process*. Now the *Law of Large Numbers (LLN) for Markov chains* (Proposition 5.9 and Theorem 5.10 in [79]) can be applied. Taking the function $f(X)$ in [79] as identity gives

$$\frac{1}{n}\sum_{\tau=0}^{n-1}\gamma_i(\tau) \quad \rightarrow \quad \overline{\mathbf{d}}(\mathbf{x}_i,t) \qquad (n\rightarrow\infty) \tag{4.14}$$

where $\gamma(\tau)$ is the configuration generated by the Gibbs sampler at time τ. Hence, running the Gibbs sampler or the Metropolis algorithm (or for that matter any other algorithm generating appropriate Markov chain) sufficiently long, and taking separately the time averages at spatial locations $\mathbf{x}_i$ will approximate the conditional mean of the vector $\mathbf{d}(\mathbf{x}_i,t)$ (remember that the generation of γ's is conditioned on the observations g_{t_-}, g_{t_+}).

The clear advantage of the above approach over the simulated annealing algorithm is that it requires no annealing schedule, since the generation of samples happens at some constant temperature **T**. Instead of the four parameters of the schedule (initial and end temperatures, length of Markov chain, and temperature change rule) now only one temperature, at which the process evolves, is necessary. This temperature, however, controls the state-rejection rate of the generation algorithm. The higher the parameter **T** the higher the rejection rate and the more chaotic the generated samples. The lower the **T**, the lower the rejection rate and the more orderly the structure of the generated realizations.

4.5 GIBBS SAMPLER FOR THE CONTINUOUS STATE-SPACE $\mathcal{S}_{\mathbf{d}}$

In Chapter 3 the *a posteriori* probability density $p(\hat{\mathbf{d}}_t|G_{t_-}=g_{t_-}, G_{t_+}=g_{t_+})$ was derived. It has the same form as the *a posteriori* distribution for discrete $\mathbf{d}_t$'s, but since $\mathcal{S}_{\mathbf{d}}=R^2$ is continuous the displacement Gibbs distribution becomes a density. Multiplied by the likelihood $P(G_{t_+}=g_{t_+}|\mathbf{D}_t=\hat{\mathbf{d}}_t, G_{t_-}=g_{t_-})$ it results in the *a posteriori* probability density $p(\hat{\mathbf{d}}_t|G_{t_-}=g_{t_-}, G_{t_+}=g_{t_+})$.

The conditional density driving the Gibbs sampler will have the same form as the conditional probability (4.12). The difference is that sample vectors from the continuous state-space $\mathcal{S}_{\mathbf{d}}=R^2$ have to be generated rather than from a discrete one. The discrete state-space Gibbs sampler must compute the complete conditional distribution (4.12) at each $(\mathbf{x}_i, t)$ (for explanation consult Appendix 4.C). This is a highly time-consuming task. To sample $\mathcal{S}_{\mathbf{d}}$ with infinitely small increments is even less feasible, hence a different approach must be used.

Recall the local energy function $U_{\mathbf{d}}^{\mathbf{i}}$ of the conditional probability driving the Gibbs sampler (4.13). Note that the first term is quadratic with respect to displaced pel difference $\tilde{r}$, while the second one is quadratic (given the potential (3.17)) in $\hat{\mathbf{d}}_t$. If the first term could be approximated by a quadratic form in $\hat{\mathbf{d}}_t$, then $U_{\mathbf{d}}^{\mathbf{i}}$ would be quadratic and the conditional density would be Gaussian. There exist efficient techniques for generating normal bivariates, hence such an approach would significantly speed up the estimation process.

Assume that an approximate estimate $\dot{\mathbf{d}}_t$ 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 pel difference $\tilde{r}$ can be expressed as follows:

$$\tilde{r}(\hat{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)=\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)+(\hat{\mathbf{d}}(\mathbf{x}_i,t)-\dot{\mathbf{d}}(\mathbf{x}_i,t))\cdot\nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t), \tag{4.15}$$

where the spatial gradient of $\tilde{r}$ is defined as

$$\nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)=\begin{bmatrix}\tilde{r}^x(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\\ \tilde{r}^y(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\end{bmatrix}= \tag{4.16}$$
$$\begin{bmatrix}\frac{\partial\tilde{g}(\mathbf{x}_i-\Delta t\cdot\dot{\mathbf{d}}(\mathbf{x}_i,t),t_-)}{\partial x}\cdot\Delta t+\frac{\partial\tilde{g}(\mathbf{x}_i+(1.0-\Delta t)\cdot\dot{\mathbf{d}}(\mathbf{x}_i,t),t_+)}{\partial x}\cdot(1.0-\Delta t)\\ \frac{\partial\tilde{g}(\mathbf{x}_i-\Delta t\cdot\dot{\mathbf{d}}(\mathbf{x}_i,t),t_-)}{\partial y}\cdot\Delta t+\frac{\partial\tilde{g}(\mathbf{x}_i+(1.0-\Delta t)\cdot\dot{\mathbf{d}}(\mathbf{x}_i,t),t_+)}{\partial y}\cdot(1.0-\Delta t)\end{bmatrix}.$$

To minimize the total energy U via simulated annealing the temperature $\mathbf{T}$ is slowly reduced to zero. Including the temperature $\mathbf{T}$ in the weights λ'_g and $\lambda'_{\mathbf{d}}$ defined as follows

$$\lambda'_g=\lambda_g/\mathbf{T},\qquad \lambda'_{\mathbf{d}}=\lambda_{\mathbf{d}}/\mathbf{T},$$

permits to write the local energy $U^i_{\mathbf{d}}$ driving the Gibbs sampler at location $(\mathbf{x}_i,t)$ as:

$$\begin{aligned}U^i_{\mathbf{d}}(\hat{\mathbf{d}}(\mathbf{x}_i,t)|\hat{\mathbf{d}},g_{t_-},g_{t_+})\approx&\lambda'_g\cdot[\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)+\\ &(\hat{\mathbf{d}}(\mathbf{x}_i,t)-\dot{\mathbf{d}}(\mathbf{x}_i,t))\cdot\nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2+\\ &\lambda'_{\mathbf{d}}\cdot\sum_{j:\,\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_i)}V(\hat{\mathbf{d}}(\mathbf{x}_j,t),\hat{\mathbf{d}}(\mathbf{x}_i,t)),\end{aligned} \tag{4.17}$$

where $\dot{\mathbf{d}}$ is fixed. The above local energy is quadratic with respect to $\hat{\mathbf{d}}$. It can be shown that the conditional probability density (4.12) with the above energy is a 2-D Gaussian with the following mean vector at location $(\mathbf{x}_i,t)$ (for the derivation of the mean vector and the covariance matrix see Appendix 4.D):

$$\mathbf{m}=\overline{\mathbf{d}}(\mathbf{x}_i,t)-\frac{\varepsilon_i}{\mu_i}\nabla^T_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t),$$

where the scalars ε_i and μ_i are defined as follows

$$\begin{aligned}\varepsilon_i&=\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)+(\overline{\mathbf{d}}(\mathbf{x}_i,t)-\dot{\mathbf{d}}(\mathbf{x}_i,t))\cdot\nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\\ \mu_i&=\xi_i\frac{\lambda'_{\mathbf{d}}}{\lambda'_g}+\|\nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\|^2,\end{aligned} \tag{4.18}$$

and $\overline{\mathbf{d}}(\mathbf{x}_i,t)$ is an average vector

$$\overline{\mathbf{d}}(\mathbf{x}_i,t)=\frac{1}{\xi_i}\sum_{j:\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_i)}\hat{\mathbf{d}}(\mathbf{x}_j,t), \tag{4.19}$$

where $\xi_i=|\eta_{\mathbf{d}}(\mathbf{x}_i)|$ denotes the size of neighbourhood $\eta_{\mathbf{d}}(\mathbf{x}_i)$ e.g., 4 for the neighbourhood system $\mathcal{N}^1_{\mathbf{d}}$. Note that a neighbourhood does not include the central vector. The horizontal

and vertical component variances σ_x^2, σ_y^2, as well as the correlation coefficient ρ, which comprise the covariance matrix M, have the following form

$$\begin{bmatrix}\sigma_x^2\\ \sigma_y^2\end{bmatrix} = \frac{1}{2\xi_i\lambda'_{\mathbf{d}}\mu_i}\begin{bmatrix}\xi_i\frac{\lambda'_{\mathbf{d}}}{\lambda'_g} + [\tilde{r}^y(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2\\ \xi_i\frac{\lambda'_{\mathbf{d}}}{\lambda'_g} + [\tilde{r}^x(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2\end{bmatrix}$$

$$\rho\sigma_x\sigma_y = \frac{-\tilde{r}^x(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\tilde{r}^y(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)}{2\xi_i\lambda'_{\mathbf{d}}\mu_i}.$$

The initial vector $\dot{\mathbf{d}}$ can be assumed zero throughout the estimation process, but then with increasing displacement vector estimates the error due to intensity non-linearity would significantly increase. Hence, it is better to "track" an intensity pattern by modifying $\dot{\mathbf{d}}$ accordingly. An interesting result can be obtained when it is assumed that at every iteration of the Gibbs sampler $\dot{\mathbf{d}} = \overline{\mathbf{d}}$ i.e., the initial (approximate) displacement field is equal to the average from the previous iteration, and also that the neighbourhood $\mathcal{N}_{\mathbf{d}}^1$, resulting in ξ_i=4, is used. Then, the estimation process can be described by the following iterative equation:

$$\hat{\mathbf{d}}^{n+1}(\mathbf{x}_i,t) = \overline{\mathbf{d}}^n(\mathbf{x}_i,t) - \frac{\varepsilon_i}{\mu_i}\nabla_{\mathbf{d}}^T\tilde{r}(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t) + \boldsymbol{n}_i, \tag{4.20}$$

where n denotes the iteration number, and

$$\begin{aligned}\varepsilon_i &= \tilde{r}(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\\ \mu_i &= 4\frac{\lambda_{\mathbf{d}}}{\lambda_g} + \|\nabla_{\mathbf{d}}\tilde{r}(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\|^2.\end{aligned} \tag{4.21}$$

$\boldsymbol{n}_i$ is a Gaussian bivariate with the following component variances and correlation coefficient:

$$\begin{bmatrix}\sigma_x^2\\ \sigma_y^2\end{bmatrix} = \frac{\mathsf{T}}{8\lambda_{\mathbf{d}}\mu_i}\begin{bmatrix}4\frac{\lambda_{\mathbf{d}}}{\lambda_g} + [\tilde{r}^y(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2\\ 4\frac{\lambda_{\mathbf{d}}}{\lambda_g} + [\tilde{r}^x(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2\end{bmatrix}$$

$$\rho\sigma_x\sigma_y = -\mathsf{T}\frac{\tilde{r}^x(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)\tilde{r}^y(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)}{8\lambda_{\mathbf{d}}\mu_i}.$$

The Gibbs sampler for the continuous state-space $\mathcal{S}_{\mathbf{d}}$ described above results in a spatio-temporal gradient estimation method, while the discrete state-space Gibbs sampler from Section 4.2.3 is an example of explicit (pel) matching algorithm. This important difference is due to the Taylor expansion used in approximating the displaced pel difference $\tilde{r}$ in (4.13) by the linear form (4.15).

Note the similarity of the iterative update equation (4.20) to the update equation of the Horn-Schunck algorithm (2.11). Except for the displaced pel difference replacing the motion

constraint equation and the inclusion of the random variable n_i they are identical. In fact if one applied the same Gibbs sampler procedure to a discretized version of objective function (2.8), the Horn-Schunck algorithm would be obtained with appropriate bivariate term n_i. It is interesting that similar update equations result from two different approaches: Horn and Schunck establish necessary conditions for optimality and solve them by deterministic relaxation, while here a 2-D Gaussian distribution is fitted into the conditional probability driving the Gibbs sampler. For T=0 and $\dot{\mathbf{d}} = \overline{\mathbf{d}}$ 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 instantaneous freezing instead of slow temperature decay. More details on this similarity will be given in Chapter 8.

At the beginning, when the temperature is high, the random term n_i has large variance and the estimates assume quite random values seeking the optimum. 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. Note that the variance σ_x^2 of the horizontal component for fixed λs and $\tilde{r}^y$ decreases with growing $\tilde{r}^x$. It means that when there is significant horizontal gradient (detail) in the image structure the uncertainty of the estimate in horizontal direction is small. The same applies to σ_y^2. Hence, the algorithm takes into account the image structure in determining the amount of randomness allowed at a given temperature.

4.6 SPATIAL IMAGE INTERPOLATION

In the practical implementation of any motion estimation algorithm operating with sub-pixel accuracy the image intensities at locations $(\mathbf{x}_i, t_\pm) \notin \Lambda_g$ (recall that Λ_g is the image lattice) must be known. Such intensities were denoted by $\tilde{g}$, for example in the energy increment (4.4) of the Metropolis algorithm or in the local energy (4.13) driving the Gibbs sampler. In this section spatial interpolation algorithms computing $\tilde{g}$ will be investigated.

The values of $\tilde{g}$ can be either interpolated or approximated from g. Let $(\mathbf{x}_i, t_\pm)$ be a spatio-temporal position at which the image intensity is required. The interpolation process performs local modeling of the intensity in such a way that

$$(\mathbf{x}_i, t_\pm) \in \Lambda_g \Rightarrow \tilde{g}(\mathbf{x}_i, t_\pm) = g(\mathbf{x}_i, t_\pm)$$

In the approximation process, however, a surface fitting is performed and this relationship is not satisfied (in general). Such a fitting can be considered a low-pass filtering and will not be pursued in this section since filtering of the intensities will be discussed in Chapter 5.

An interpolation scheme suitable for motion estimation should be characterized by the following properties::

1. efficiency, since it will be repeated hundreds of thousands of times for each motion field; as the consequence it should be simple and have only local support,
2. continuity with respect to the image intensities g, so that a small change of the known intensity (on Λ_g) will not result in a large change of the interpolated intensity (or its derivative),
3. continuity with respect to the position $\mathbf{x}$, so that a small change of the position e.g., $\mathbf{x} = \mathbf{x}_i - \Delta t \cdot \widehat{\mathbf{d}}(\mathbf{x}_i, t)$, will not result in a large change of the interpolated intensity (or its derivative).

The constraint on the image intensity derivatives is not relevant in the case of stochastic estimation via the discrete state-space Gibbs sampler, since the derivatives of g are not involved. However, for the continuous state-space Gibbs sampler discussed in the last section, the constraint on image intensity derivatives is vital.

Considering the above requirements, such complex schemes as the spline interpolation, which requires spline computation for each set of data, have to be excluded. In order to satisfy the efficiency requirement, the separable low-order polynomial interpolators will be used. A separable 2-D spatial interpolator can be constructed as a cascade of 1-D horizontal and vertical interpolators, hence is more efficient than a full 2-D interpolator. 1-D interpolators of 1-st, 2-nd and 3-rd order will be discussed here.

From now until the end of this section 1-D notation will be used. Let w be an input signal defined over a lattice Λ, and $\widetilde{w}$ be an interpolated value of w defined over R. Let x, $y \in R$ be arbitrary positions. Let Λ have a sampling period δ so that $\Lambda = \{x: x = j\delta, j \in I\}$, where I is a set of integers. If $\lfloor x \rfloor_\Lambda$ denotes the nearest lattice point from x with smaller or equal coordinate

$$\lfloor x \rfloor_\Lambda = \lfloor \frac{x}{\delta} \rfloor \cdot \delta,$$

then $\Delta x = x - \lfloor x \rfloor_\Lambda$ is the distance from the site x to the nearest preceding lattice point.

With the above notation the 2-, 3- and 4-point Lagrange interpolators can be described by the following equations:

1. **linear**: $0.0 \leq \Delta x < 1.0$

$$\tilde{w}(x) = (1 - \Delta x) \cdot w(\lfloor x \rfloor_\Lambda) + \Delta x \cdot w(\lfloor x \rfloor_\Lambda + \delta), \tag{4.22}$$

2. **quadratic**: $-0.5 \leq \Delta x < 0.5$

$$\begin{aligned} \tilde{w}(x) = & \frac{1}{2}\Delta x(\Delta x - 1) \cdot w(\lfloor x \rfloor_\Lambda - \delta) \\ & - (\Delta x - 1)(\Delta x + 1) \cdot w(\lfloor x \rfloor_\Lambda) \\ & + \frac{1}{2}\Delta x(\Delta x + 1) \cdot w(\lfloor x \rfloor_\Lambda + \delta), \end{aligned} \tag{4.23}$$

3. **cubic**: $0.0 \leq \Delta x < 1.0$

$$\begin{aligned} \tilde{w}(x) = & -\frac{1}{6}\Delta x(\Delta x - 1)(\Delta x - 2) \cdot w(\lfloor x \rfloor_\Lambda - \delta) \\ & + \frac{1}{2}(\Delta x + 1)(\Delta x - 1)(\Delta x - 2) \cdot w(\lfloor x \rfloor_\Lambda) \\ & - \frac{1}{2}(\Delta x + 1)\Delta x(\Delta x - 2) \cdot w(\lfloor x \rfloor_\Lambda + \delta) \\ & + \frac{1}{6}(\Delta x + 1)\Delta x(\Delta x - 1) \cdot w(\lfloor x \rfloor_\Lambda + 2\delta). \end{aligned} \tag{4.24}$$

The above equations, suitable for implementation, are not very useful in the analysis of continuity of the interpolators. Consider the interpolation process from the systems point of view. Assuming that only linear (in the systems sense) interpolators are considered, the following convolution can be used to describe the interpolator operation:

$$\tilde{w}(x) = \sum_{y \in \Lambda} w(y) \cdot a(x - y), \qquad x \in R \tag{4.25}$$

where $\tilde{w}$ is defined over R and a is the impulse response of an interpolator (linear filter) defined over R. Note that also a derivative of $\tilde{w}$ at arbitrary position x can be computed as follows (due to the linearity of convolution (4.25)):

$$\frac{\partial \tilde{w}}{\partial x} = \sum_{y \in \Lambda} w(y) \frac{\partial a(x - y)}{\partial x}, \qquad x \in R. \tag{4.26}$$

From the equations (4.22), (4.23) and (4.24), and the convolution (4.25), the impulse responses $a(x)$ of the interpolators can be derived (Appendix 4.E). They are piecewise polynomials of the first-, second- and third-order, respectively, and are plotted in Fig. 4.3. Note that while the linear and cubic interpolators have continuous impulse responses, the

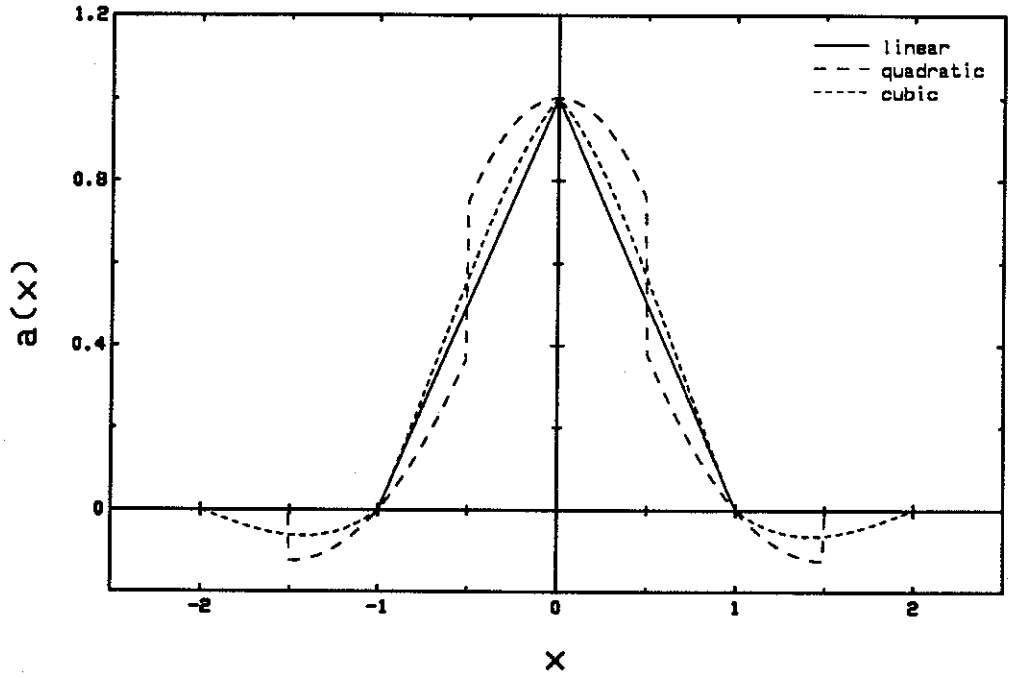


Fig. 4.3 Impulse responses of the linear, quadratic and cubic interpolators.

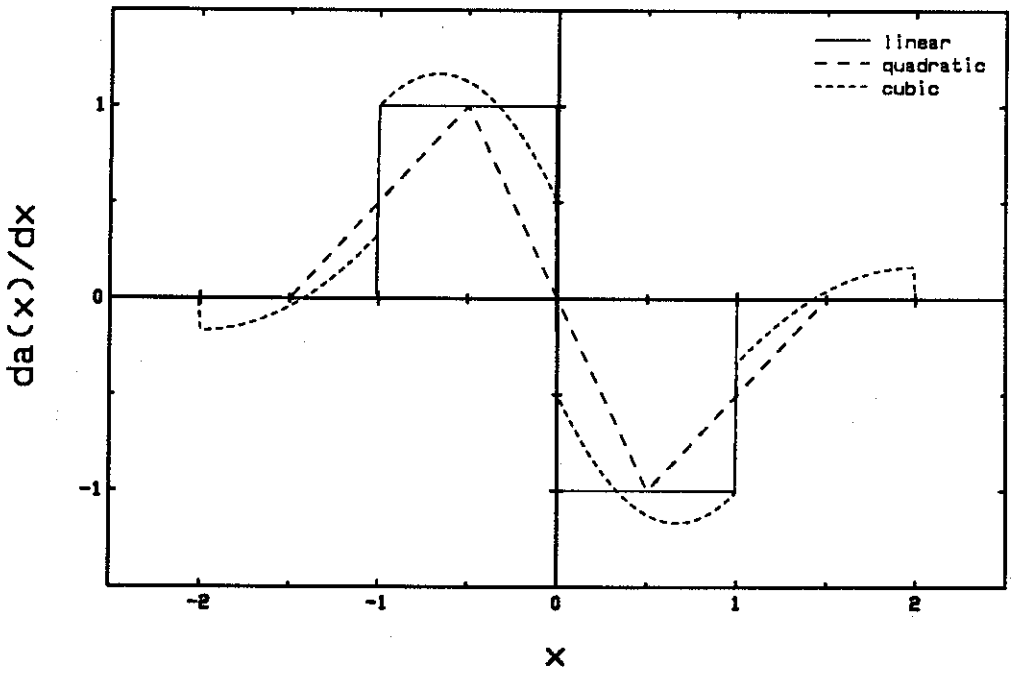


Fig. 4.4 Impulse response derivatives of the linear, quadratic and cubic interpolators.

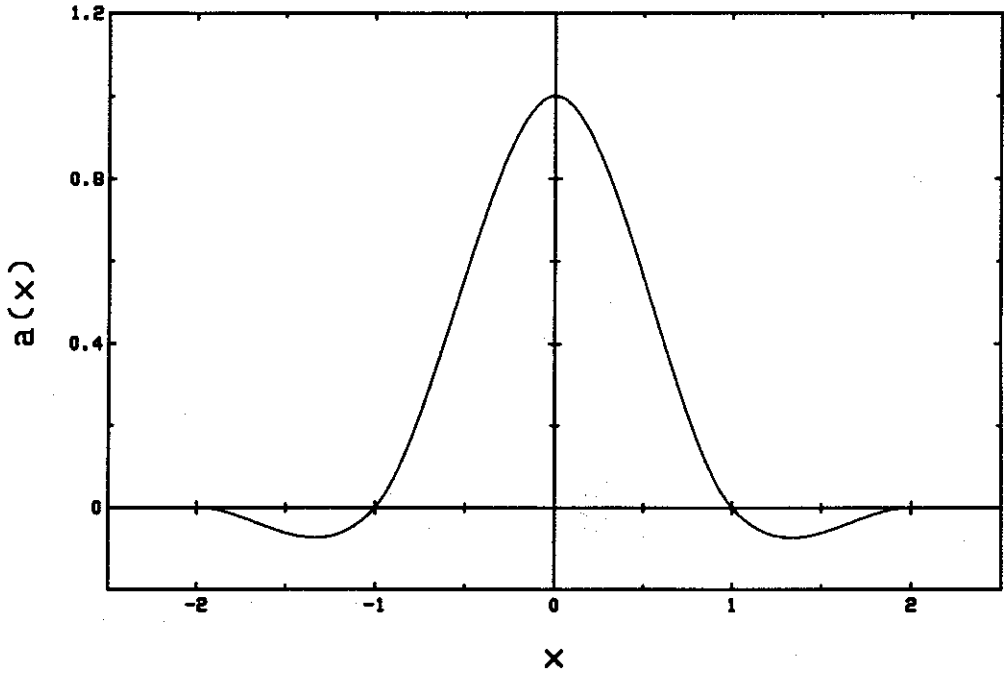


Fig. 4.5 Impulse response of cubic interpolator with C^1-continuous impulse response proposed by Keys [50].

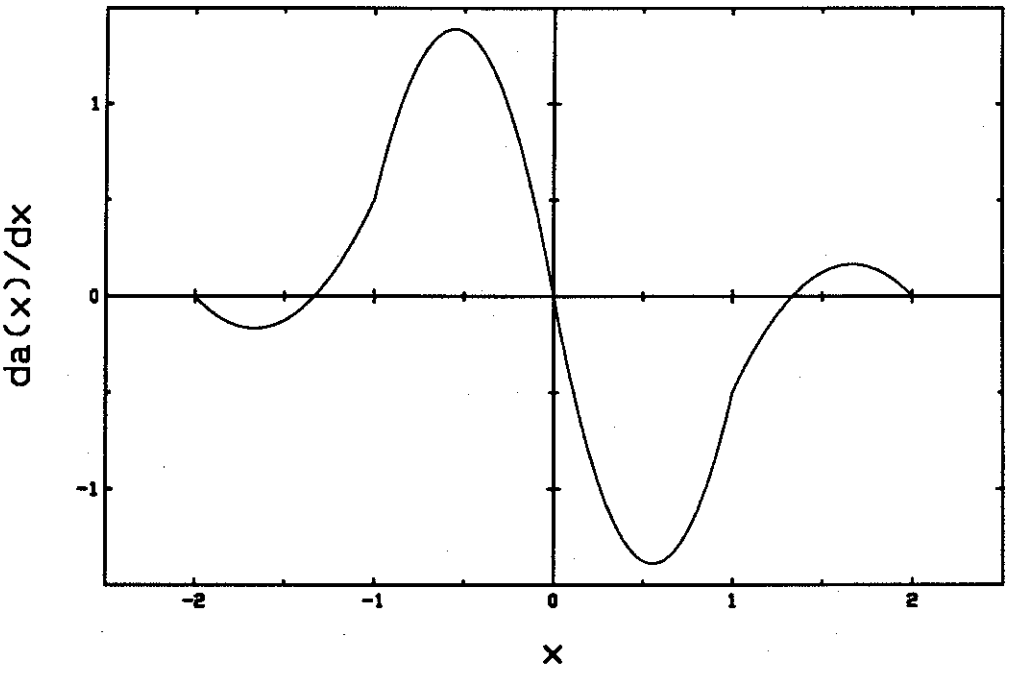


Fig. 4.6 Impulse response derivative of cubic interpolator with C^1-continuous impulse response proposed by Keys [50].

quadratic interpolator is discontinuous at x=-1.5,-0.5,0.5,1.5. The derivatives of $a(x)$ (Appendix 4.E) are plotted in Fig. 4.4. Note that the derivative of the impulse response is discontinuous for the linear (x=-1.0,0.0,1.0) and cubic (x=-1.0,1.0) interpolators.

By choosing the separable low-order polynomial form for the interpolator, the requirement 1, posed at the beginning of this section, has been satisfied. Also the requirement 2 is fulfilled, as the interpolator is a linear discrete filter with coefficients from the range $[0.0, 1.0]$ (equations (4.22), (4.23), (4.24)). In order to satisfy the third requirement, the interpolator impulse response must be continuous.

The problem of designing C^1-continuous (continuous up to the first derivative) impulse responses of linear and quadratic interpolators is overconstrained. It is, however, possible with the cubic interpolator, since there are 8 unknown coefficients in the piecewise polynomial description of its impulse response and only 7 constraints providing continuity of $a(x)$ and $\partial a(x)/\partial x$ (Appendix 4.E). This one degree of freedom has been used by Keys [50] to design a C^1-continuous cubic interpolator such that its output signal agrees with the first three terms of the Taylor series expansion of the input signal. The impulse response of this optimal cubic interpolator and its derivative are shown in Figs. 4.5 and 4.6.

4.7 TEST IMAGES

The algorithms described in this chapter have been tested on a number of synthetic and natural images with synthetic or natural motion. The next 3 sections describe the test images used throughout this thesis.

The ultimate goal of the motion estimation investigated here is its application to TV images, hence all the test images used subsequently had been stored in a displayable line-interlaced format. Such format consists of fields separated in time by τ_{60}=1/60 sec, and containing 106 lines with 256 pels per line. The odd fields are offset vertically by half of the inter-line distance. The estimation is performed on the inter-field basis, hence the terms field and image are used interchangeably. The theoretical dynamic range of the luminance component is [0,255], while in practice it is usually limited to [40,200]. Unless otherwise indicated only luminance fields are used. The test images shown subsequently in the figures,

and referred to as fields number 0, are in fact two interleaved fields number 0 and displayed as a frame. The white rectangular contours encircle the area actually used in the estimation process.

4.7.1 Test image 1: synthetic data, synthetic motion

To provide a quantitative test for the motion estimation algorithms proposed here, I used a pattern based on the concept of a random dot stereogram. Fig. 4.7 shows the 0-th field of a 24 field sequence which will be referred to as the test image 1. The 0-th field consists of random, uniformly distributed numbers from the range [40,200]. The subsequent even fields are exact copies of the 0-th field, except that a 50 by 20 pel rectangle (note that due to the interlace, the effective aspect ratio of the rectangle is around 5:4) in the center of the encircled area has been moved by $\mathbf{d}_s$=(2.0,1.0) with respect to the previous even field. The odd fields are exact replicas of the preceding even fields. The white frame encircles the area of 77 by 49 pels subsequently used for estimation.

4.7.2 Test image 2: natural data, synthetic motion

This test image provides a synthetic motion of natural data obtained from a video camera. Fig. 4.8 shows the 0-th field of a 24 field sequence which will be referred to as the test image 2. The background is provided by the test image 3, while the 45 by 20 pels rectangle in the center is obtained from another image as follows. That image had been first prefiltered by a 2-D low-pass separable linear-phase FIR filter to minimize aliasing after the subsampling. Then it was subsampled by 4 producing a contracted copy of the original image. In subsequent fields of the test image 2 the same (stationary) background image was used, while the moving pels in the rectangle were obtained from appropriately shifted pels in the prefiltered image. The subsampling factor of 4 provides the 1/4 pel precision of displacement vectors. Unlike in the test image 1, this test pattern permits non-integer displacements in which case there is no perfect data matching (more realistic situation). The white frame encircles the area of 77 by 49 pels.

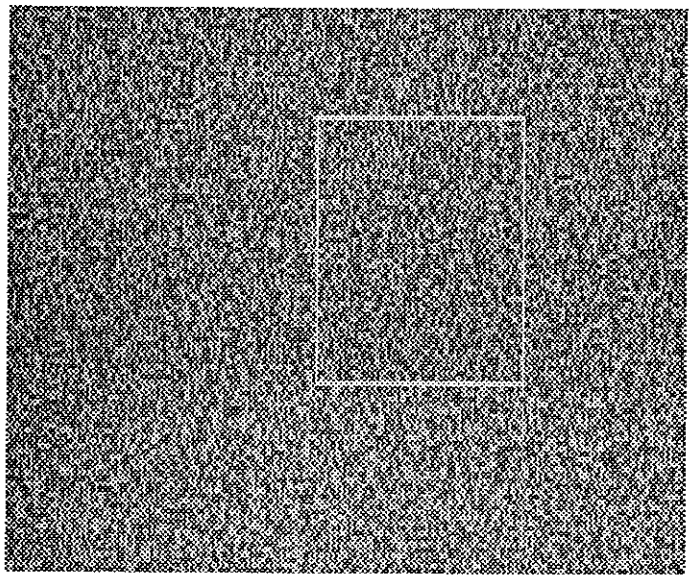


Fig. 4.7 Test image 1, field number 0; a 50 by 20 pels rectangle in the center moves by $\mathbf{d}_s$=(2.0,1.0) with every even field, the odd fields are identical to the preceding even fields, the white frame encircles the area used in estimation.

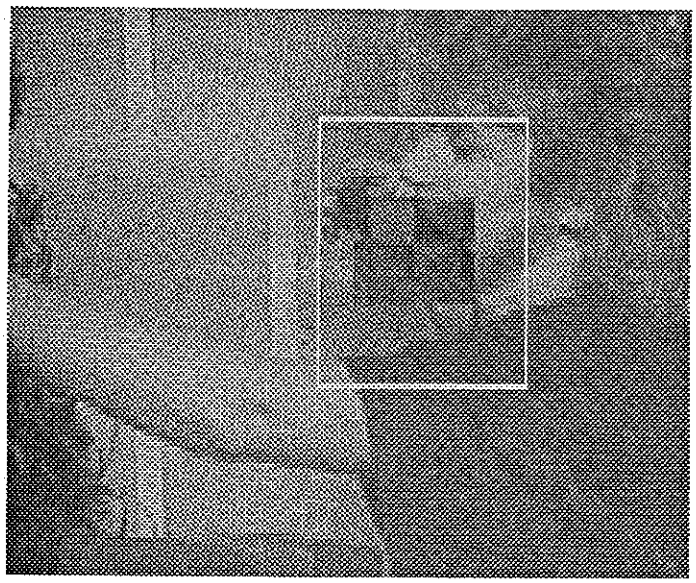


Fig. 4.8 Test image 2, field number 0; a 45 by 20 pels rectangle in the center moves by $\mathbf{d}_s$=(0.75,0.25) with every field, the white frame encircles the area used in estimation.

Fig. 4.9 Test image 3, field number 0; the white frame encircles the area used for estimation.

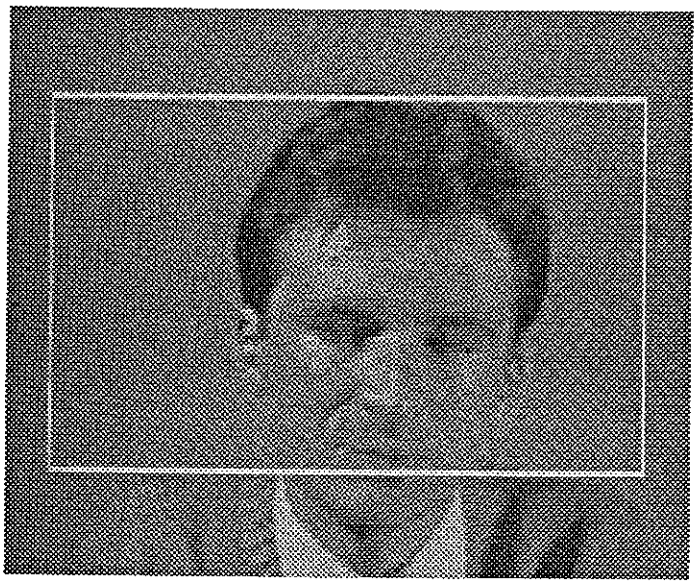


Fig. 4.10 Test image 4, field number 0; the white frame encircles the area used for estimation.

4.7.3 Test image 3 and 4: natural data, natural motion

Figs. 4.9 and 4.10 show the 0-th fields of two 24 field sequences obtained with a video camera. No filtering or any other processing has been applied to those sequences after their acquisition. The acquisition process included as usual camera filtering, sampling and quantization as described in Section 3.4.2. There is some aliasing present in the data due to insufficient filtering before sampling.

4.8 EXPERIMENTAL RESULTS

In this section some experimental results will be presented. First the results of application of the MAP and the MEC estimation to all four test images will be discussed. Then, some results for the test images 1 and 2 corrupted with noise will be also shown.

Note that, as defined in Section 4.3, an iteration means a complete scan of a displacement field i.e., $M_{\mathbf{d}} = M_{\mathbf{d}}^h \times M_{\mathbf{d}}^v$ attempted modifications of displacement vectors. The stochastic relaxation used to produce the results in this chapter has been based either on the discrete state-space $\mathcal{S}'_{\mathbf{d}}$ with d_{max}=2.0 and N_d=17 levels in each direction or on the continuous state-space. Unless otherwise indicated the parameter $\lambda_{\mathbf{d}}$ is set to 1.0 in all experiments. Of course only the ratio $\lambda_{\mathbf{d}}/\lambda_g$ provides a weight between matching and smoothing, however the absolute values of $\lambda_{\mathbf{d}}$ and λ_g have significant impact on the choice of the initial temperature $\mathbf{T}_0$.

The motion estimates presented in the following sections have been obtained from pairs of images (fields) separated by $T_g = 2\tau_{60}$, and with Δt=0.0 (i.e., forward estimation) and $\Lambda_{\mathbf{d}} = \Lambda_g$.

Since the true motion fields are known for the test images 1 and 2 (except for the occlusion and newly exposed areas), it is possible to assess the quality of motion field estimates. The Mean Squared Error and the bias measuring the departure of motion field estimate $\widehat{\mathbf{d}}$ from the known motion field $\mathbf{d}_s$, are defined as follows:

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

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

where $\mathcal{R}$ denotes the spatial rectangle to which synthetic motion had been applied. MSE and $bias$ are computed for test images 1 and 2 only, and are shown below appropriate motion field estimates.

Due to the interlace the vertical distance between neighbouring pixels in a field is about twice larger than the horizontal distance. This effect is taken into account in the potential function (3.17) by multiplying the vertical components of displacement vectors by 2.0. Consequently there is more weight given to the vertical components which is reflected in the mean squared error.

4.8.1 Results for test image 1

The two images used for estimation match exactly except for the newly exposed and occlusion areas. The match occurs at the displacement (0.0,0.0) in the background, and at (2.0,1.0) in the center rectangle. Due to this almost perfect matching, the estimation should strongly rely on the data while very little on the model. In fact it seems that except for those newly exposed and occlusion areas, just the data should be sufficient for motion recovery. Consequently in the experiments with test image 1 the parameter λ_g was set to 1.0, and λ_{d} was varied.

4.8.1.1 MAP estimation

The results of MAP estimation based on simulated annealing with the discrete state-space Gibbs sampler as described in Section 4.3, applied to the test image 1 are shown in Fig. 4.11. The synthetic displacement applied to the 50 by 20 pels rectangle in the center, which the estimation process attempts to recover, is shown in Fig. 4.11.a. The estimates from Figs. 4.11.b,.c,.d have been produced for $\lambda_{\mathrm{d}}/\lambda_g$ =0.0, 0.05, 1.0 respectively, with the first-order neighbourhood system $\mathcal{N}_{\mathrm{d}}^1$ and bilinear spatial interpolation of the image. The exponential annealing schedule with T_0=1.0 and a=0.980 over 200 iterations has been used in all three cases. Such schedule results in final temperature T_f=0.0179.

For the ratio $\lambda_{\mathrm{d}}/\lambda_g$=0.0 (Fig. 4.11.b), the estimation process disregarded the displacement model completely. This result indicates that trusting just the data may be misleading, which is not unexpected, since the algorithm uses only single pel matching and not blocks of

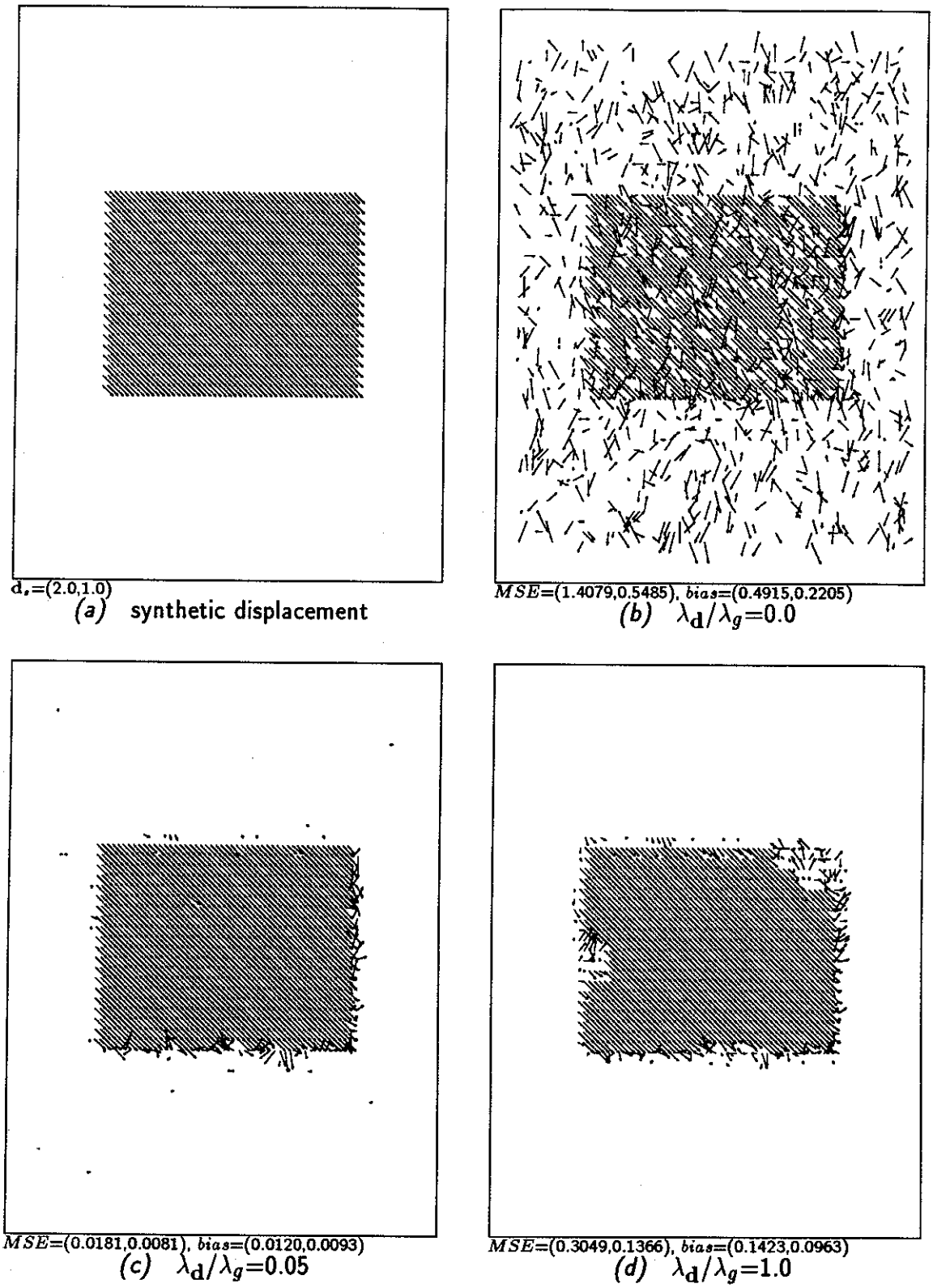


Fig. 4.11 Discrete state-space MAP estimates: test image 1, neighb. $\mathcal{N}_d^1$, bilinear interp., exponential schedule, T_0=1.0, a=0.980, 200 iter.

pels, in which case the estimate would be closer to the true motion. Since there is no model constraint the process seeks any vector minimizing a DPD, and hence it is ill-posed. To restrict the class of admissible solutions a (weak) adherence to a displacement model should be enforced. Fig. 4.11.c shows the estimate for $\lambda_{\mathbf{d}}/\lambda_g$=0.05 (weak model constraint), which is a dramatic improvement compared to the previous result, both subjectively and in terms of MSE. Relying to a higher degree on the model deteriorates the estimate, as can be observed in Fig. 4.11.d for $\lambda_{\mathbf{d}}/\lambda_g$=1.0. It turns out that this clearly suboptimal solution is due to the interaction between the weight ratio $\lambda_{\mathbf{d}}/\lambda_g$ and the temperature **T**. With $\lambda_{\mathbf{d}}/\lambda_g$=1.0 the local energy $U_{\mathbf{d}}^i$ is higher (complete displacement energy contributes to the local energy) than for small $\lambda_{\mathbf{d}}/\lambda_g$, and hence the relative temperature is smaller. That this departure of the estimate from the true motion is caused by a low relative temperature rather than directly by a high ratio $\lambda_{\mathbf{d}}/\lambda_g$ has been confirmed by computing a motion estimate for $\lambda_{\mathbf{d}}/\lambda_g$ =1.0 and $\mathbf{T}_0$=10.0. The result turned out to be almost identical to that from Fig. 4.11.c.

Note that the estimate from Fig. 4.11.c is very uniform inside of the rectangle where it attains exactly the value of the synthetic displacement. The two leftmost columns and the top row of the rectangle are the newly exposed areas, while the two rightmost columns and the bottom row are the occluded areas. Since Δt=0.0 the intensities from $g(\mathbf{x}, t_-)$ are being matched with those at t_+, and unreliable estimates should be expected in the occlusion areas (the motion model does not take such an effect into account). This is exactly what happens in Fig. 4.11.c. Obviously for Δt=1.0 (backward estimation) the newly exposed areas would be troublesome, and both occlusion and newly exposed areas would play a role when $0.0 < \Delta t < 1.0$.

Also the inhomogeneous simulated annealing implemented via the Metropolis algorithm has been applied to the test image 1. As discussed in Section 4.3 the Metropolis algorithm is simpler per update than the Gibbs sampler, however it requires more iterations. I have applied the Metropolis algorithm over 15,000 iterations with the exponential temperature schedule modified every 75 iterations. Subjectively the result was almost identical to the estimate obtained with the Gibbs sampler (Fig. 4.11.c), however the attained energies were slightly higher. The comparison of the computational effort for both methods is not

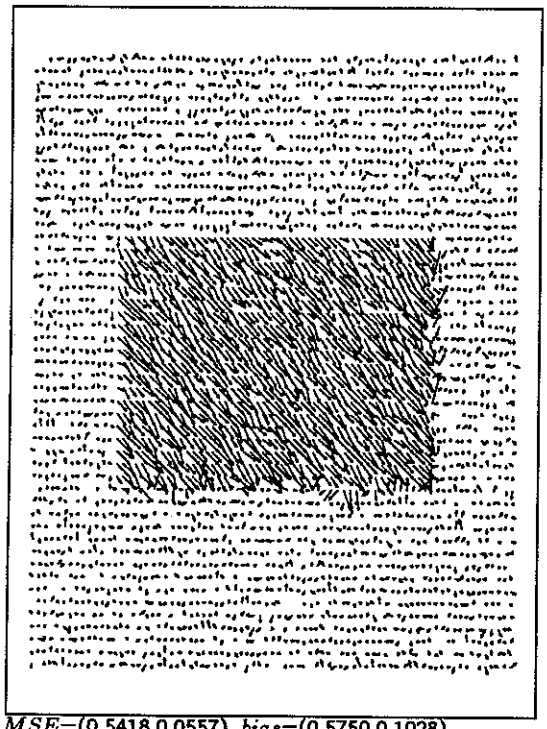
straightforward (what is the minimal effort to obtain given result ?), however the measured CPU time was very similar in both cases.

Using other exponential annealing schedules and also lowering the initial temperature $\mathbf{T}_0$ to as low value as 0.01 (with λ_d/λ_g=0.05) had negligible effect on the final result. This suggests that the objective (energy) function for this particular data has rather few and not very deep local minima, while the global minimum is probably very deep. This observation, later confirmed in Section 8.1, is not surprising because the data cue is very strong due to the "gray value corners" at almost every spatial location.

Application of the continuous state-space MAP estimation to the test image 1 resulted in an estimate which had little in common with the true motion. This is not surprising since as a spatio-temporal gradient technique it relies on the relationship between the spatial and temporal intensity gradients, and those are absolutely unrelated in the test image 1 (the random dots are statistically independent). This, however, is not the case for the test image 2 as it will be demonstrated in Section 4.8.2.1.

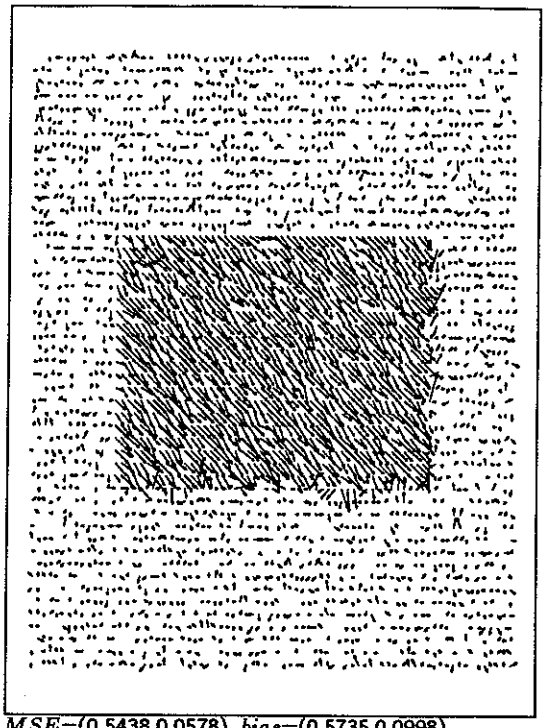
4.8.1.2 MEC estimation

Fig. 4.12 shows two MEC estimates obtained with the discrete state-space Gibbs sampler for constant temperatures **T**=1.0 (a) and **T**=0.1 (c). Again $\mathcal{N}_d^1$, bilinear interpolation and 200 iterations were used. The time average, which approximates the ensemble expectation (Section 4.4), has been computed over the last 150 iterations. Note that due to the averaging process the displayed vectors have continuous rather than discrete components. The corresponding displacement fields after quantization to the closest of N_d states from the $[-d_{max}, d_{max}]$ range are shown in Fig .4.12.b,.d. Note significantly fewer spurious vectors in the stationary background, especially in Fig. 4.12.d. The results demonstrate that MEC estimation can also provide reliable estimates without the need to use an annealing schedule. A constant temperature **T** at which the Markov chain evolves, however, must be specified. This temperature controls directly the randomness or "chaos" in the estimates: the lower the temperature the more organized (spatially) the estimate is. Too low a temperature, however, may prove suboptimal because only the most likely states will be produced rather than a whole range of states.



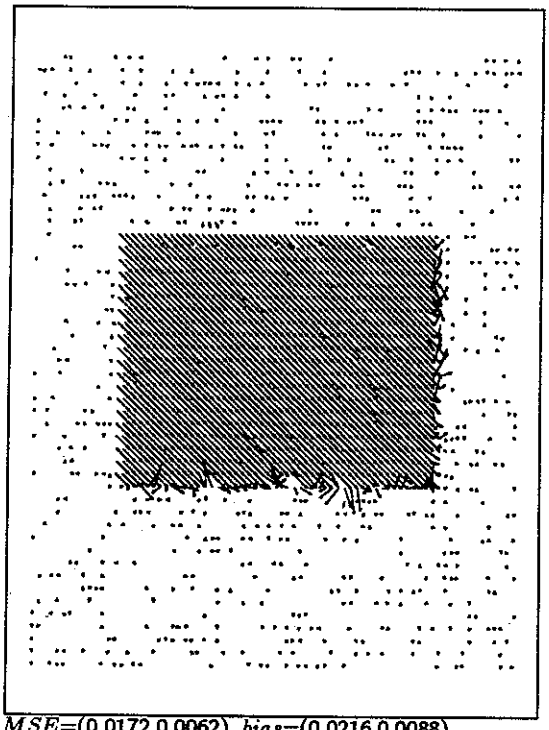
MSE=(0.5418,0.0557), $bias$=(0.5750,0.1028)

(a) T=1.0



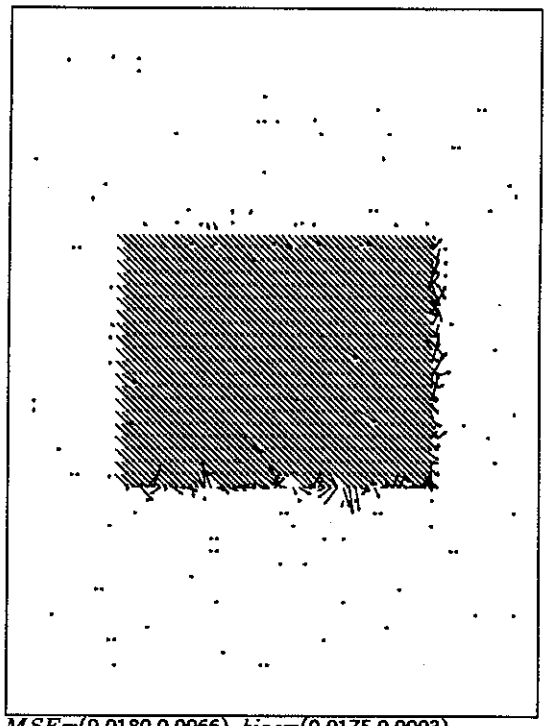
MSE=(0.5438,0.0578), $bias$=(0.5735,0.0998)

(b) T=1.0, quantized



MSE=(0.0172,0.0062), $bias$=(0.0216,0.0088)

(c) T=0.1



MSE=(0.0180,0.0066), $bias$=(0.0175,0.0093)

(d) T=0.1, quantized

Fig. 4.12 MEC estimates: test image 1, λ_d/λ_g=0.05, neighb. $\mathcal{N}_d^1$, bilinear interp., 200 iter.

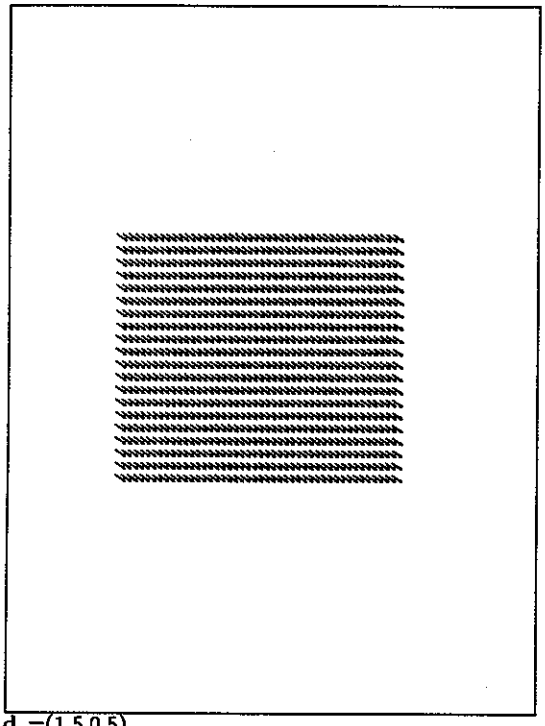
4.8.2 Results for test image 2

This test image contains the natural data in motion hence there is no perfect matching and consequently 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" $\beta_{\mathbf{d}}$ of the order of 0.1–1.0, the ratio $\lambda_g/\lambda_{\mathbf{d}}$ should be of the order of 20.0. Obviously the exact value cannot be established, however it will be demonstrated that even a 2 orders of magnitude change in the ratio $\lambda_g/\lambda_{\mathbf{d}}$ will not incur disastrous effects.

4.8.2.1 MAP estimation

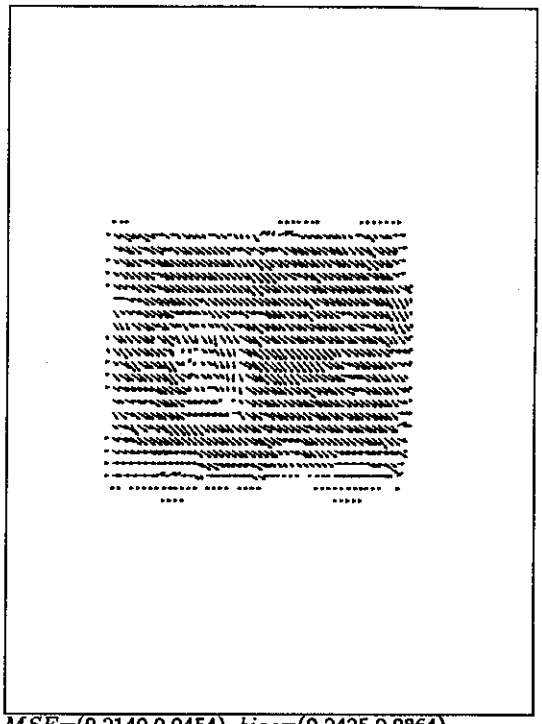
Fig. 4.13.a shows the displacement applied to the center rectangle which the algorithms try to recover. Figs. 4.13.b,.c,.d show the results of discrete state-space MAP estimation applied to test image 2 for three values of the ratio $\lambda_{\mathbf{d}}/\lambda_g$: 4.0 (b), 20.0 (c), 100.0 (d), neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interpolator and exponential annealing schedule starting at $\mathbf{T}_0$=1.0 with a=0.980 over 200 iterations. Clearly the result from Fig. 4.13.c is smoother than the one from Fig. 4.13.b, however no disastrous effects can be observed. Even more smooth, but very feasible, field can be seen in Fig. 4.13.d for $\lambda_{\mathbf{d}}/\lambda_g$ ratio of 100.0. It shows that the adjustment range for the ratio $\lambda_{\mathbf{d}}/\lambda_g$ is quite large.

Since there is no perfect matching between the images, the spatial interpolation type should have an impact on the results. Fig. 4.14.a shows the MAP estimate obtained for the same parameters as those used for the result from Fig. 4.13.c, except that bilinear spatial interpolation was used instead of the Keys bicubic interpolation. Observe that both subjectively (smoothness) and objectively (mean squared error in the center rectangle) this estimate is inferior to the one from Fig. 4.13.c. To demonstrate the role of the annealing schedule in simulated annealing, Figs. 4.14.b,.c show the estimates obtained for the standard set of parameters but different annealing schedules. Fig. 4.14.b shows the estimate produced for $\mathbf{T}_0$=0.1. Clearly the estimate is subjectively suboptimal, as is also confirmed by the higher total energy U. In another experiment, also a higher initial temperature $\mathbf{T}_0$=30.0 has been used, with no significant subjective or objective (energy) effect. In practical implementation an important parameter is the final temperature $\mathbf{T}_f$,



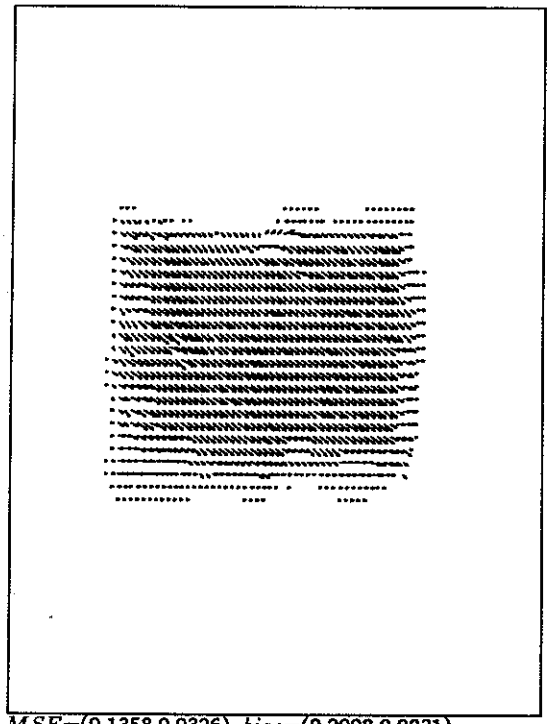
$\mathbf{d}_*$=(1.5,0.5)

(a) synthetic displacement



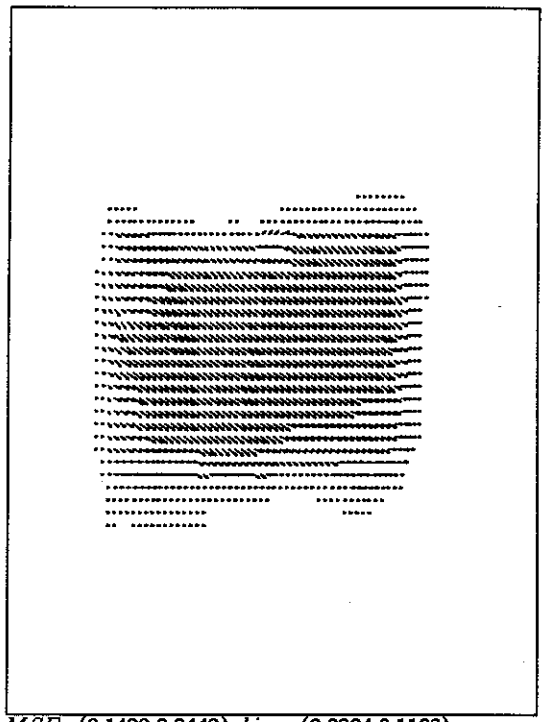
MSE=(0.2140,0.0454), $bias$=(0.2425,0.0864)

(b) $\lambda_{\mathbf{d}}/\lambda_g$=4.0



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

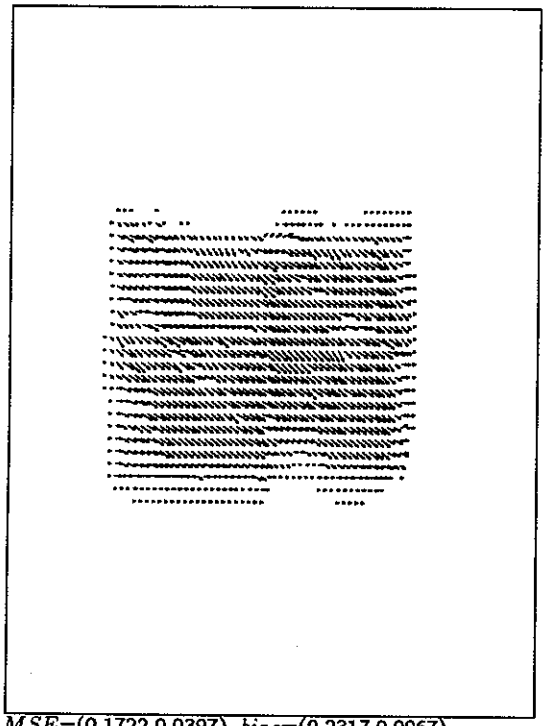
(c) $\lambda_{\mathbf{d}}/\lambda_g$=20.0



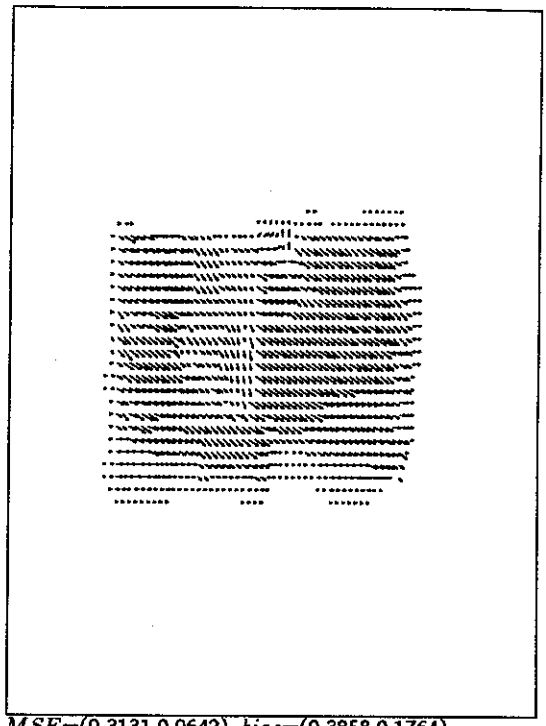
MSE=(0.1422,0.0442), $bias$=(0.2294,0.1189)

(d) $\lambda_{\mathbf{d}}/\lambda_g$=100.0

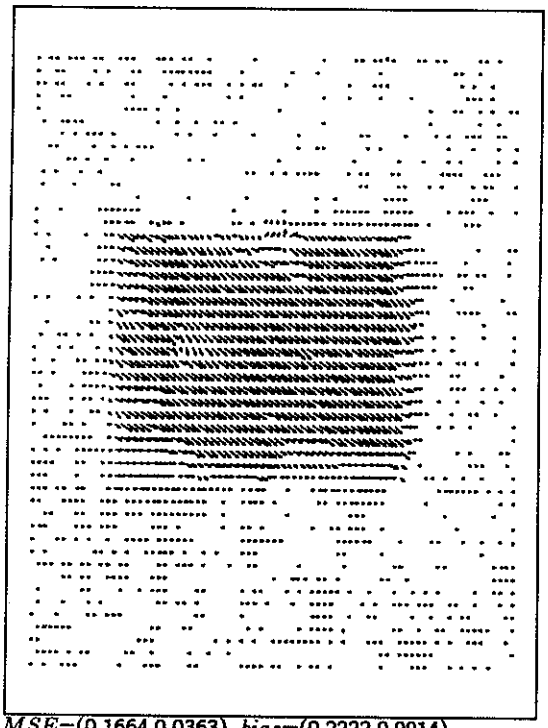
Fig. 4.13 Discrete state-space MAP estimates for various $\lambda_{\mathbf{d}}/\lambda_g$: test image 2, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}_0$=1.0, a=0.980, 200 iter.



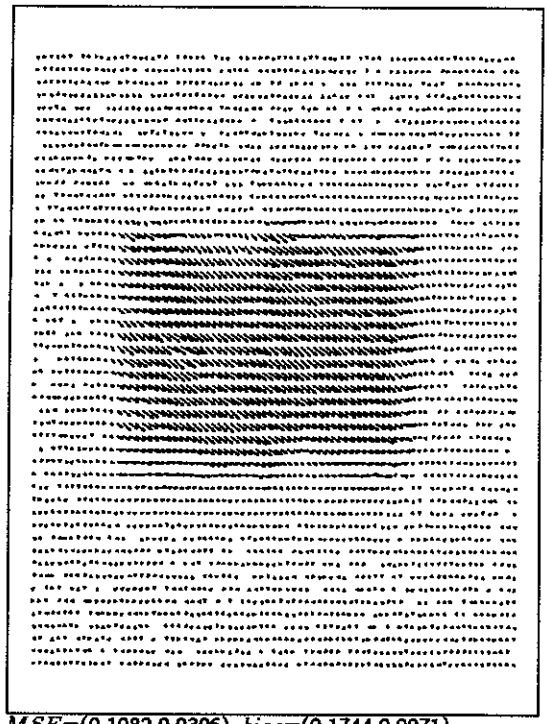
MSE=(0.1722,0.0397), $bias$=(0.2317,0.0967)
(a) bil. interp., T_0=1.0, a=0.980



MSE=(0.3131,0.0642), $bias$=(0.3858,0.1764)
(b) T_0=0.1, a=0.980



MSE=(0.1664,0.0363), $bias$=(0.2222,0.0914)
(c) logarithmic sched., T_0=1.0



MSE=(0.1082,0.0306), $bias$=(0.1744,0.0971)
(d) continuous, T_0=5.0, a=0.9944

Fig. 4.14 Discrete (a,b,c) and continuous (d) state-space MAP estimates: test image 2, λ_d/λ_g=20.0, neighb. $\mathcal{N}_d^1$, bilinear (a) or Keys bicubic (b,c,d) interp., exponential (a,b,d) or logarithmic (c) schedule, 200 (a,b,c) or 1000 (d) iter.

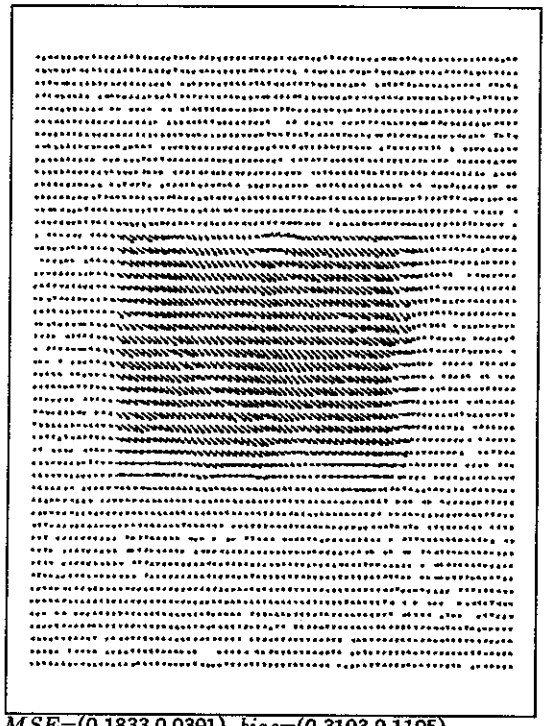
which theoretically should slowly attain 0.0. The value 0.0 cannot be implemented, but for a=0.980 the final temperature after 200 iterations is $\mathbf{T}_f$=0.0179. Further reduction of the temperature to 0.001 did not result in significant changes. Too high a final temperature may leave the process in a fairly chaotic state. Fig. 4.14.c shows an estimate obtained with the logarithmic temperature schedule (Section 4.3) from $\mathbf{T}_0$=1.0, which after 200 iterations attains $\mathbf{T}_f$=0.1307. Clearly the process is on its way to an estimate probably at least as good as the one from Fig. 4.13.b, however a too high final temperature leaves numerous vectors still trying to sample some sub-optimal states. The further linear reduction of the temperature to attain $\mathbf{T}_f$=0.01, produces an estimate very close to the one from Fig. 4.13.c.

Also the continuous state-space MAP estimation (Section 4.5) has been applied to the test image 2. The result is shown in Fig. 4.14.d for the same parameters as those applied to obtain the result from Fig. 4.13.c except for the exponential annealing schedule which started at $\mathbf{T}_0$=5.0 and used a=0.9944† over 1000 iterations. The need for higher initial temperature can be explained by the fact that in the continuous state-space case not only the intensity interpolation is involved but also the intensity derivative computation (which is an ill-posed problem itself [74]). The false vector estimates can be due to both the intensity value and its derivative, hence a higher initial temperature is needed to overcome all the local minima. That the continuous state-space case is more "delicate" (requires less abrupt state changes), especially at low temperatures, seems to be confirmed by the need to use long annealing. More abrupt temperature changes result in suboptimal estimates, especially in the triangle (center of the moving rectangle) containing vertical bars with horizontal repetition period close to the displacement per frame.

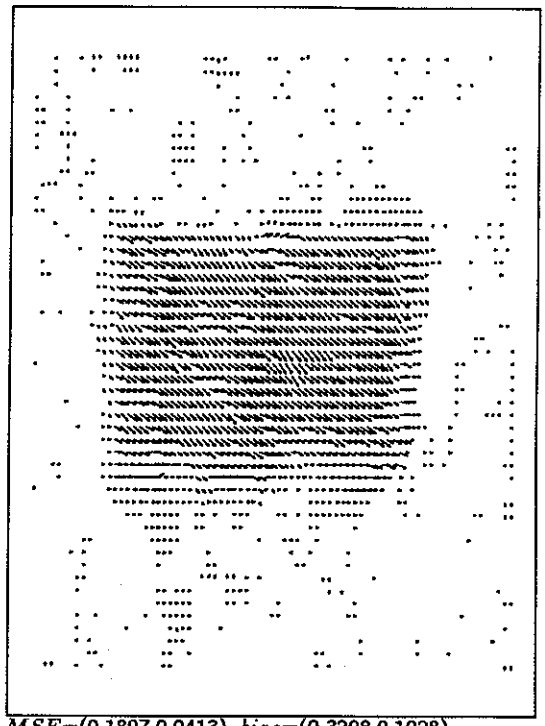
Also the Metropolis algorithm has been applied to the test image 2. Implemented over 15,000 iterations it produced a similar estimate to that from Fig. 4.13.c, however with somewhat higher energy. Since for similar computational effort it results in higher energy values, only the Gibbs sampler will be used in subsequent experiments.

4.8.2.2 MEC estimation

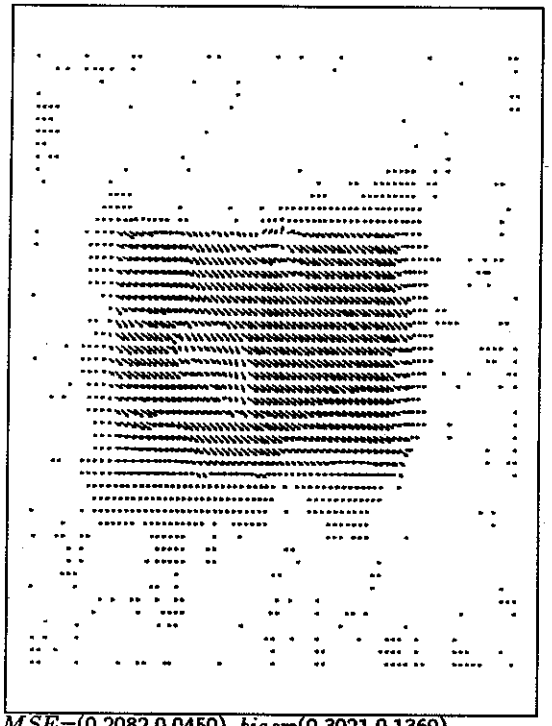
† This choice of a has been dictated by the final temperature $\mathbf{T}_f$=0.0179, used in the discrete-state space simulated annealing, in order to compare the results. It has not been optimized in any way.



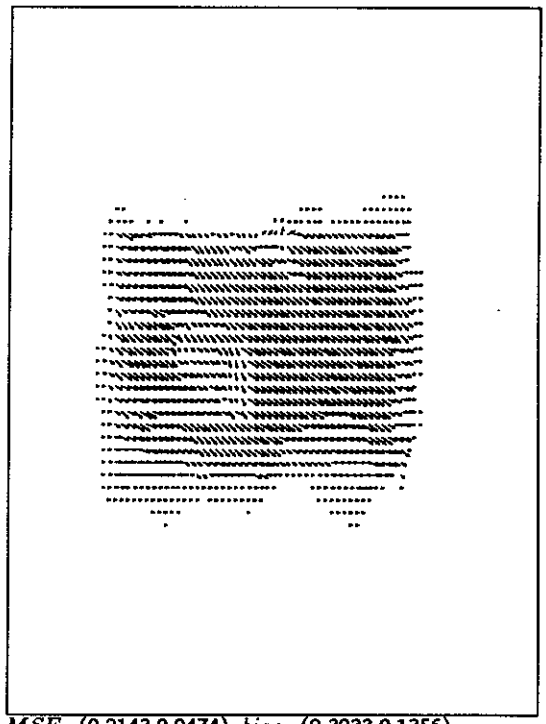
MSE=(0.1833,0.0391), $bias$=(0.3193,0.1105)
(a) T=1.0



MSE=(0.1897,0.0413), $bias$=(0.3208,0.1028)
(b) T=1.0, quantized



MSE=(0.2082,0.0450), $bias$=(0.3021,0.1369)
(c) T=0.1



MSE=(0.2143,0.0474), $bias$=(0.3033,0.1356)
(d) T=0.1, quantized

Fig. 4.15 MEC estimates: test image 2, $\lambda_{\mathbf{d}}/\lambda_g = 20.0$, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., 200 iter.

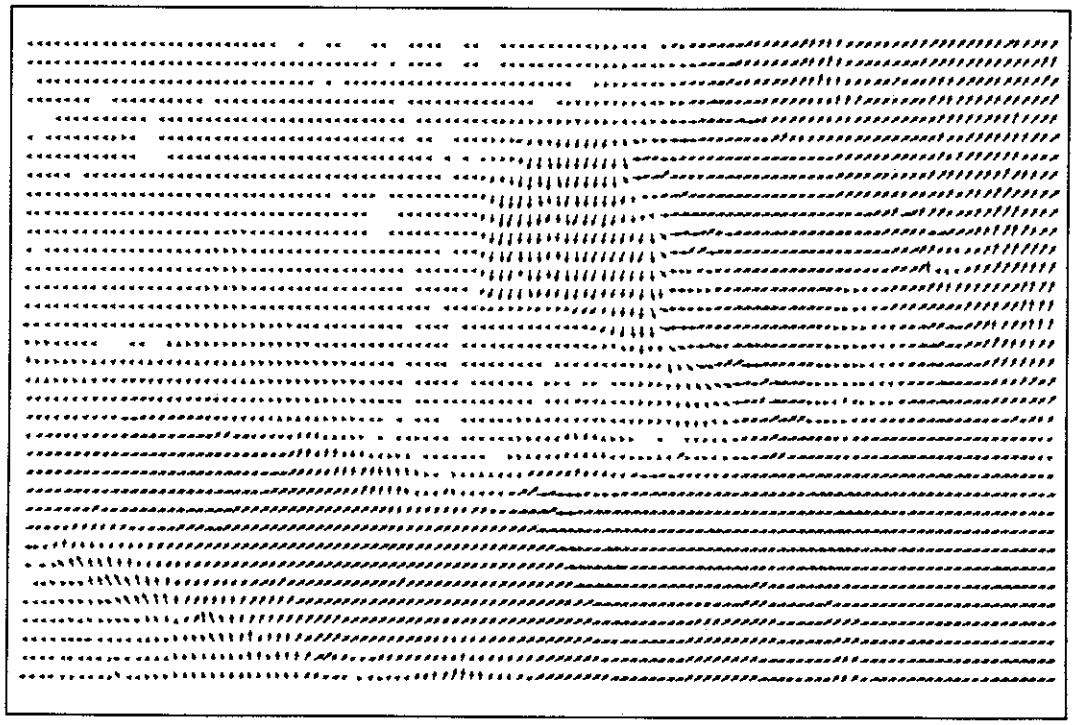
Figs. 4.15.a,.c show the MEC estimate obtained for temperatures **T**=1.0 and 0.1, respectively. Other relevant parameters are the same as for the result from Fig. 4.13.c. The time average has been computed over the last 150 iterations. The estimate for **T**=1.0 seems to be better subjectively, since for **T**=0.1 there is a pronounced triangle of smaller velocities in the center of the rectangle. Also the mean squared error for both components is significantly smaller for **T**=1.0. That the temperature **T**=0.1 is too small seems to be confirmed by a poor MAP estimate commencing at $\mathbf{T}_0$=0.1 (Fig. 4.14.b). For **T**=1.0 the estimate inside of the rectangle is close to that shown in Fig. 4.13.b, however it is not in the stationary background, where small chaotically oriented vectors are present. After quantization of the MEC estimate to the 0.25 pel precision (as used by the Gibbs sampler in this case), the result is dramatically improved in the background as shown in Figs. 4.15.b,.d.

4.8.3 Results for test images 3 and 4

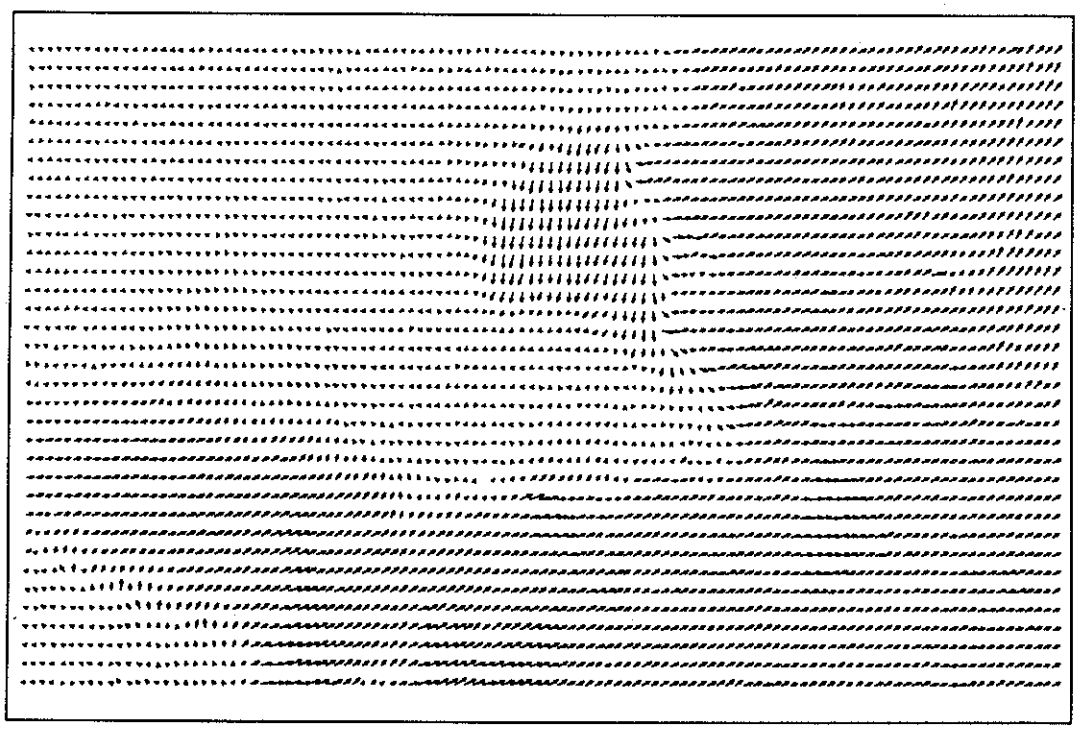
The test images 3 and 4 contain camera captured scenes with moving subjects. Since the data is natural, there is no perfect matching, and consequently the estimation process should rely to a higher degree on the *a priori* motion model. Like in the previous section the ratio $\lambda_{\mathbf{d}}/\lambda_g$ of the order of 20.0 seems to be a reasonable choice.

4.8.3.1 MAP estimation

Figs. 4.16.a,.b and 4.17.a,.b show discrete and continuous state-space MAP estimates obtained from the test images 3 and 4. In both cases $\lambda_{\mathbf{d}}/\lambda_g$=20.0, the first order neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$ and exponential temperature schedule were used. The discrete estimation used bilinear interpolation and initial temperature $\mathbf{T}_0$=1.0 with a=0.980 over 200 iterations, while the continuous estimation used Keys bicubic interpolation and schedule starting at $\mathbf{T}_0$=5.0 with a=0.9944 over 1000 iterations. Keys bicubic interpolation used in discrete state-space estimation resulted in almost identical estimates as the ones from Fig. 4.16.a and 4.17.a. This seems to confirm the earlier observation (Section 4.6) that since the discrete state-space Gibbs sampler uses no intensity derivatives it should be more robust to image intensity interpolation. Like in the case of test image 2, the continuous state-space estimation required higher $\mathbf{T}_0$ and longer annealing schedule to attain results

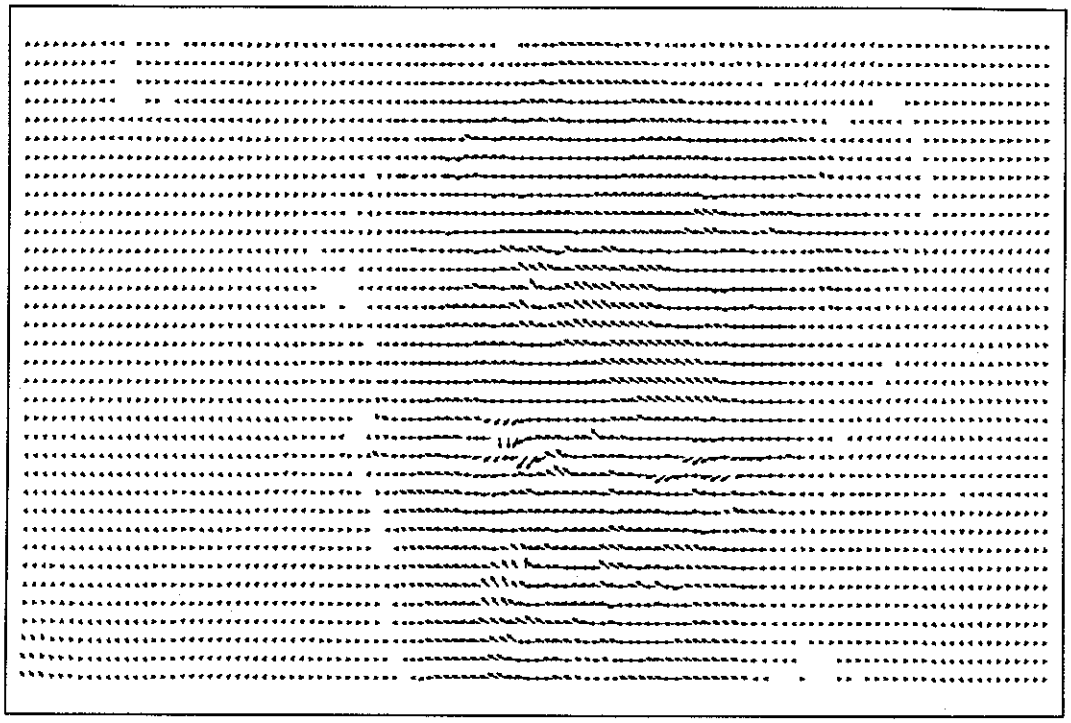


(a) discrete, bilinear interp., $\mathbf{T}_0$=1.0, a=0.980, 200 iter.

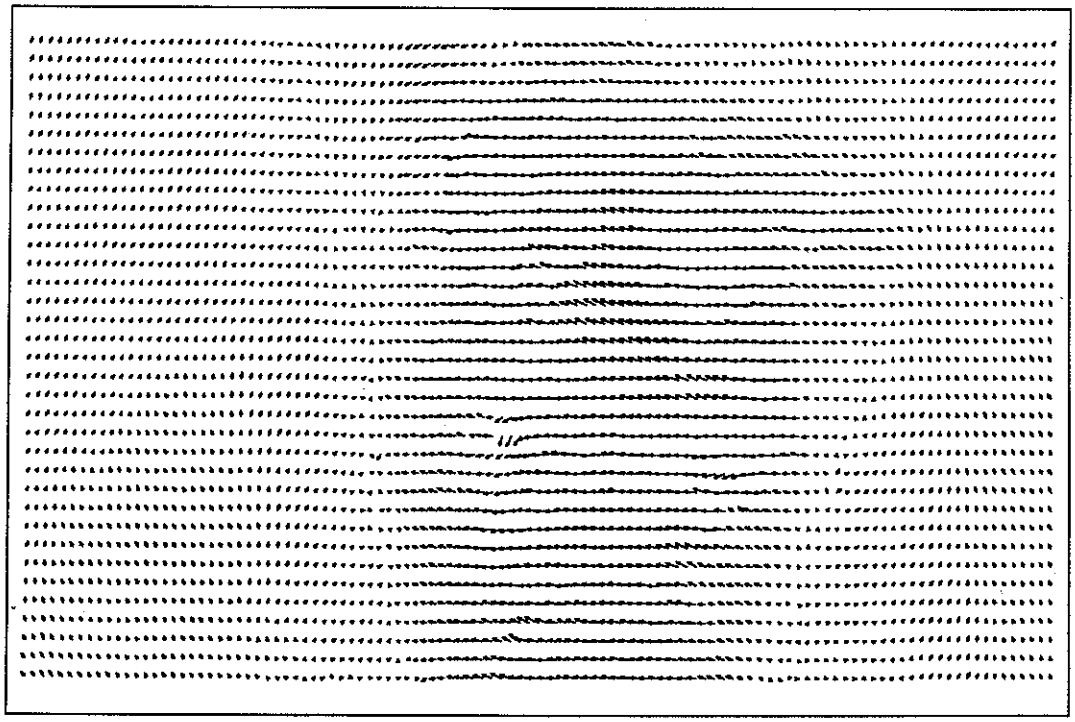


(b) continuous, Keys bicubic interp., $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter.

Fig. 4.16 Discrete and continuous state-space MAP estimates: test image 3, $\lambda_{\mathbf{d}}/\lambda_g$=20.0, neighb. $\mathcal{N}_{\mathbf{d}}^{1}$, exponential schedule.



(a) discrete, bilinear interp., $\mathbf{T}_0$=1.0, a=0.980, 200 iter.



(b) continuous, Keys bicubic interp., $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter.

Fig. 4.17 Discrete and continuous state-space MAP estimates: test image 4, $\lambda_{\mathbf{d}}/\lambda_g$=20.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, exponential schedule.

similar to the discrete case. Note, however, that due to smaller computational effort per iteration, the continuous state-space estimation with Keys bicubic interpolation, which has much higher complexity than bilinear interpolation, was still about an order of magnitude faster than the discrete state-space estimation.

All estimates are smooth within moving objects and this smoothing is also applied across motion boundaries. This can be explained by the fact that motion boundaries (inferred from the data) are not as strong as in the test image 1. Note that the continuous state-space estimates are smoother than the discrete state-space results, especially on the neck in the test image 3. The motion boundaries, however, are more oversmoothed too. The smoothness of the continuous state-space estimates is confirmed by significantly lower total energy than for the discrete state-space, which is due to longer annealing schedule.

It must be added that the test image 4 is rather difficult for estimation, because the illumination effects are not negligible there (the shadow of the hair on the forehead) and because of strong newly exposed areas (the hair suddenly exposed from behind the right ear), both not accounted for in the motion model.

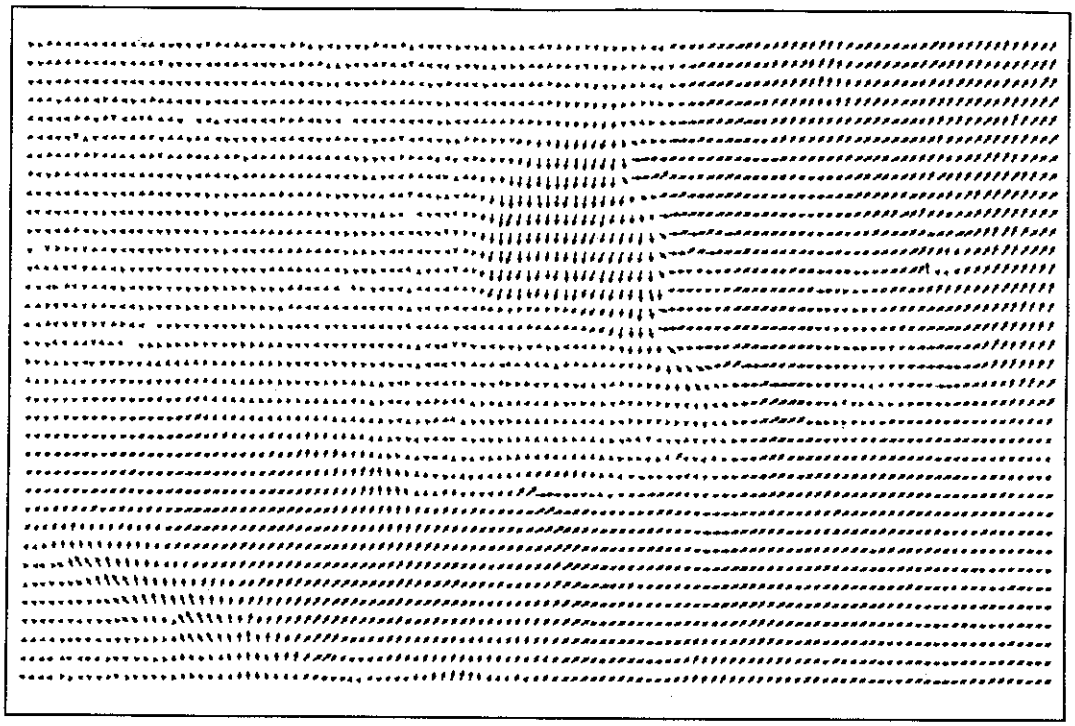
The influence of a too low initial temperature $\mathbf{T}_0$ on the motion estimates from the test image 3, as well as an example estimate early into annealing can be found in [54].

4.8.3.2 MEC estimation

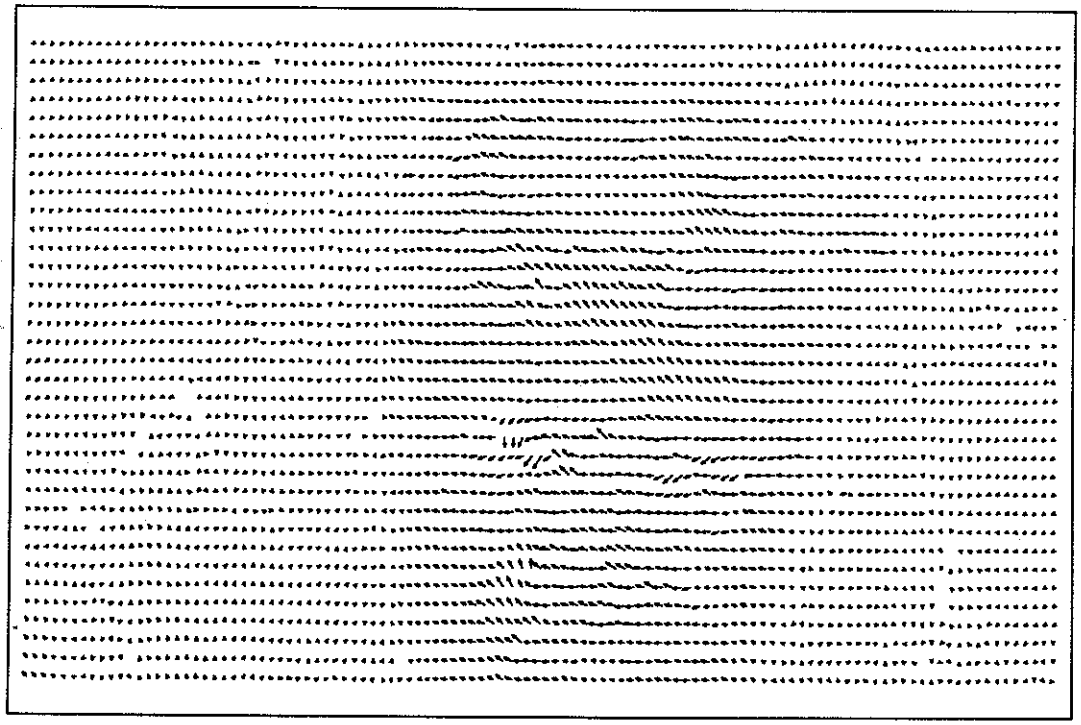
Figs. 4.18.a,.b show the MEC estimate obtained for temperature $\mathbf{T}$=1.0. Other relevant parameters are the same as for the results from previous section. The time average has been computed over the last 150 iterations, and displayed without quantization. Note that subjectively the MEC estimates are very similar to the MAP estimates from Fig. 4.16.a and 4.17.a. Their energies are higher, however, which is not surprising since the MEC estimation does not attempt to maximize the *a posteriori* probability (minimize the energy U), but rather to minimize the mean squared difference between the true motion and its estimate.

4.8.4 Results for test images 1 and 2 corrupted by noise

To test the robustness of both algorithms in the presence of noise, additive white Gaussian noise with the variance σ_a^2=20.0 has been superimposed on the test images 1

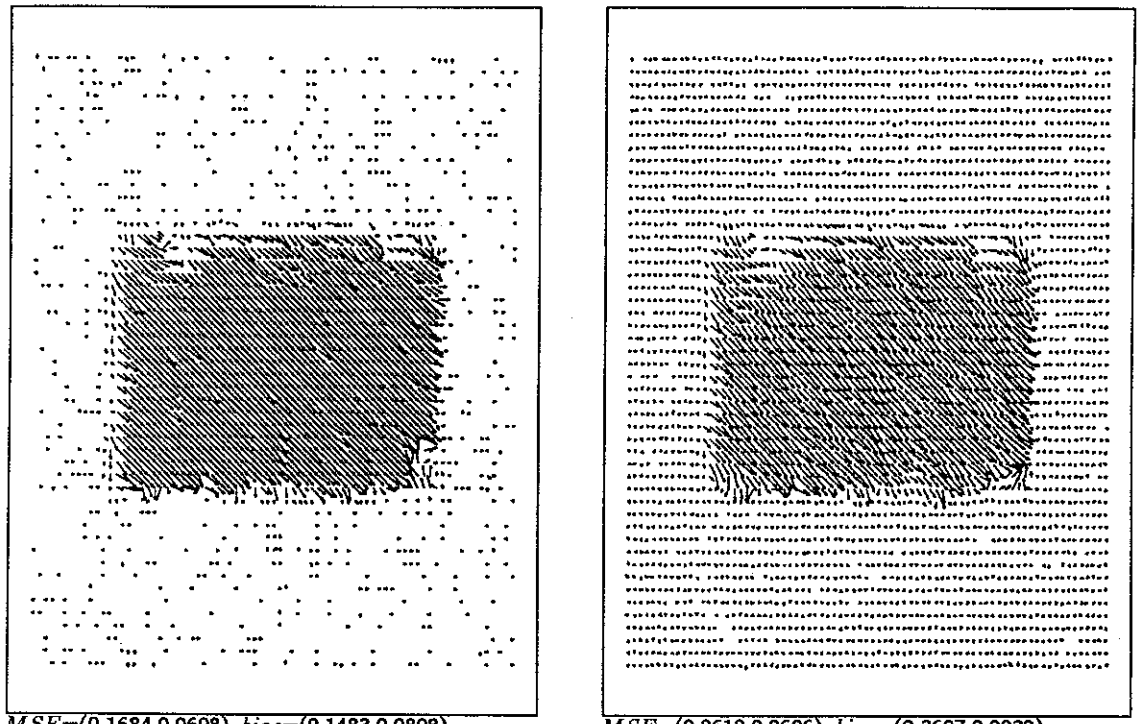


(a) test image 3



(b) test image 4

Fig. 4.18 MEC estimates: test images 3 and 4, $\lambda_{\mathbf{d}}/\lambda_g$=20.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bilinear interp., T=1.0, 200 iter.



MSE=(0.1684,0.0698), $bias$=(0.1483,0.0808)
(a) MAP, $\mathbf{T}_0$=20.0, a=0.965

MSE=(0.2612,0.0606), $bias$=(0.3607,0.0932)
(b) MEC, $\mathbf{T}$=2.0

Fig. 4.19 MAP and MEC estimates from data corrupted by white Gaussian noise (σ_a^2=20.0): test image 1, $\lambda_{\mathbf{d}}/\lambda_g$= 100.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bilinear interp., exponential schedule (a), 200 iter.

and 2. Since the noise is white with variance 20.0 and $\beta_{\mathbf{d}}$ has been chosen in the range 0.1–1.0, the original ratio $\lambda_{\mathbf{d}}/\lambda_g$ should be augmented by $2\sigma_a^2/\beta_{\mathbf{d}}$ valued at around 100.0 ($\beta_{\mathbf{d}}$=0.4).

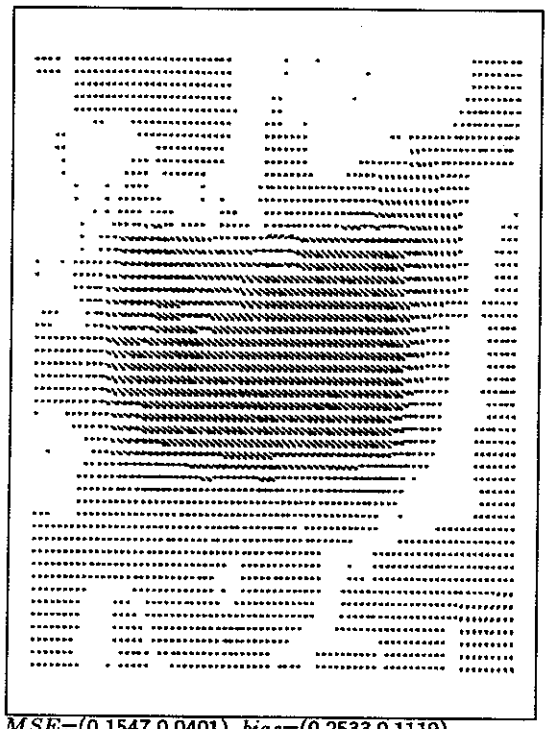
Figs. 4.19.a,.b show, respectively, the MAP and the MEC estimates of motion from the test image 1 corrupted by additive white Gaussian noise. In both cases the first order neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$, the bilinear interpolation and $\lambda_{\mathbf{d}}/\lambda_g$= 100.0 were used. The MAP estimation used the exponential annealing schedule with initial temperature $\mathbf{T}_0$=20.0 and a=0.965, while the MEC estimator was produced for $\mathbf{T}$=2.0, both over 200 iterations. Note that in spite of significant noise (σ_a^2=20.0) both estimates are quite close to the corresponding estimates obtained from the images without noise (Fig. 4.11.c and Fig. 4.12.c respectively). These results demonstrate that both the MAP and the MEC estimation are quite robust to noise. The ratio $\lambda_{\mathbf{d}}/\lambda_g$ could have been chosen differently since the value of $\beta_{\mathbf{d}}$ is not known, however even as low a value of $\beta_{\mathbf{d}}$ as 0.1 did not incur disastrous effects. With the value of 0.4 used here, the results for noisy and noiseless data are quite

comparable.

Fig. 4.20 shows the estimates of motion from the test image 2 also corrupted by additive white Gaussian noise. The first order neighbourhood system $\mathcal{N}_{\mathrm{d}}^{1}$, the Keys bicubic interpolation and the ratio $\lambda_{\mathrm{d}}/\lambda_{g}=$ 120.0 have been used in each case. The discrete state-space MAP estimation from Fig. 4.20.a used the exponential annealing schedule with T_0=20.0 and a=0.965 over 200 iterations, while the continuous state-space MAP from Fig. 4.20.b was produced for a=0.993 over 1000 iterations. The MEC estimation from Figs. 4.20.c,.d used the constant temperature T_0=1.0 over 200 iterations. The estimates again resemble quite well the corresponding true motion fields, however they are not as good as the estimates from noiseless data (mean squared error). Poorer performance for the noise corrupted test image 2 is not surprising, since the motion cue is much weaker here than in the test image 1. Subsequently imposition of significant noise (σ_a^2=20.0) "masks" the motion to certain extent. In the test image 1, however, due to the "gray value corners" almost everywhere, such masking effect is much less pronounced.

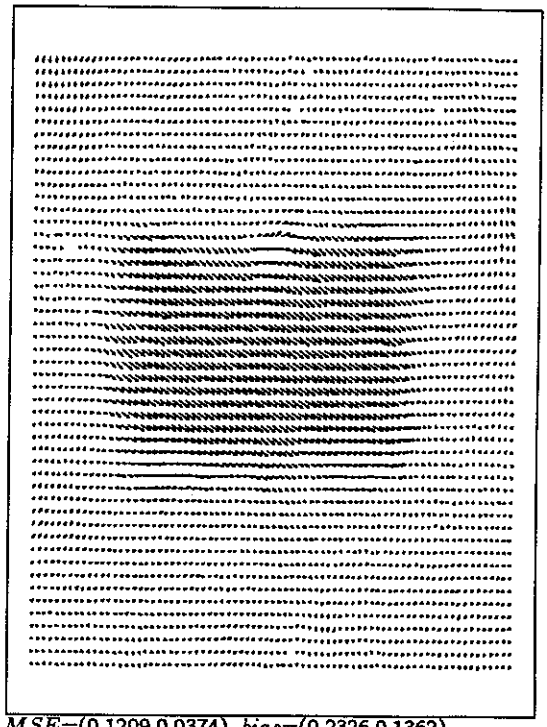
Note that both MAP estimates are quite similar except for the upper part of the moving rectangle where the continuous state-space MAP estimate outperforms the discrete state-space estimate. This is confirmed by the mean squared error, however the parametrizing energies are quite similar.

From the above experiments, as well as from the experiments with noisy test image 3 [55], it was concluded that the MAP and the MEC estimates are similar. No superiority of the MEC estimation in noisy environment has been noticed as observed by Marroquin [62] with respect to scalar MRFs applied to image reconstruction. This may be due to the fact that only noise with variance of 20.0 has been tested, and also that the state-spaces used were much larger here than in his case.



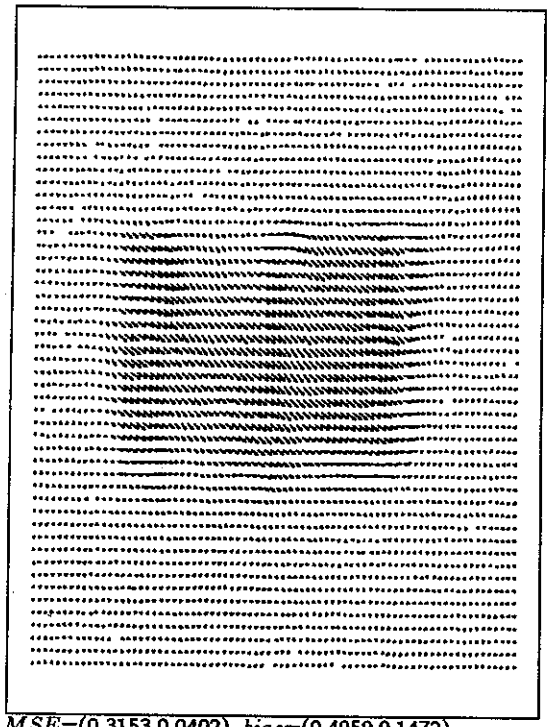
MSE=(0.1547,0.0401), $bias$=(0.2533,0.1119)

(a) MAP discrete



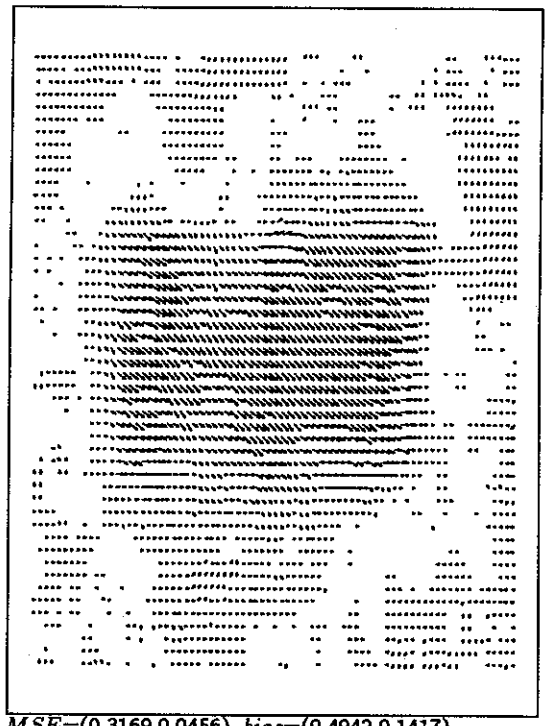
MSE=(0.1209,0.0374), $bias$=(0.2326,0.1362)

(b) MAP continuous



MSE=(0.3153,0.0402), $bias$=(0.4959,0.1472)

(c) MEC



MSE=(0.3169,0.0456), $bias$=(0.4942,0.1417)

(b) MEC, quantized

Fig. 4.20 MAP and MEC estimates from data corrupted by white Gaussian noise (σ_a^2=20.0): test image 2, $\lambda_{\mathbf{d}}/\lambda_g$= 120.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}_0$=20.0, a=0.965 over 200 iter. (a) or a=0.993 over 1000 iter. (b) or $\mathbf{T}$=1.0 over 200 iter. (c,d).

Appendix 4.A. INVARIANT DISTRIBUTION OF THE METROPOLIS ALGORITHM

In this appendix it is shown that the Gibbs distribution π is the invariant distribution of the Metropolis algorithm presented in Section 4.2.1. The reasoning below follows that presented in [36].

Let $\alpha_{\gamma\varpi}$ be defined as the following ratio:

$$\alpha_{\gamma\varpi} = \frac{\pi(\varpi)}{\pi(\gamma)} = e^{-U(\varpi)/\beta + U(\gamma)/\beta} = e^{-\Delta U/\beta}.$$

Following the notation from Section 4.2.1 note that $Q_{\gamma\varpi}$'s are transitional probabilities and are symmetric (by assumption). Hence they satisfy the following relationships

$$Q_{\gamma\varpi} \geq 0, \qquad \sum_{\varpi} Q_{\gamma\varpi} = 1, \qquad Q_{\gamma\varpi} = Q_{\varpi\gamma}. \tag{4.A.1}$$

A little reflection shows that the Metropolis algorithm described in Section 4.2.1 can be described by the following probability expressions:

$$\begin{aligned} \gamma \neq \varpi: \qquad & P_{\gamma\varpi} = \begin{cases} Q_{\gamma\varpi} \cdot \alpha_{\gamma\varpi} & \text{if} \quad \alpha_{\gamma\varpi} < 1 \\ Q_{\gamma\varpi} & \text{if} \quad \alpha_{\gamma\varpi} \geq 1 \end{cases} \\ \gamma = \varpi: \qquad & P_{\gamma\gamma} = Q_{\gamma\gamma} + \sum_{\varpi:\alpha_{\gamma\varpi}<1} Q_{\gamma\varpi} \cdot (1 - \alpha_{\gamma\varpi}) \end{aligned} \tag{4.A.2}$$

That $P_{\gamma\varpi}$'s really constitute a stochastic matrix can be concluded from the following:

1. $P_{\gamma\varpi} \geq 0$ since $Q_{\gamma\varpi} \geq 0$ (transitional probabilities), $\pi(\gamma), \pi(\varpi) \geq 0$ (π is a probability measure) and by (4.A.2),
2. $\sum_{\varpi} P_{\gamma\varpi} = 1$ since

$$\begin{aligned} \sum_{\varpi} P_{\gamma\varpi} &= P_{\gamma\gamma} + \sum_{\varpi \neq \gamma} P_{\gamma\varpi} \\ &= Q_{\gamma\gamma} + \sum_{\varpi:\,\alpha_{\gamma\varpi}<1} Q_{\gamma\varpi} \cdot (1 - \alpha_{\gamma\varpi}) + \sum_{\varpi:\,\alpha_{\gamma\varpi}<1} Q_{\gamma\varpi} \cdot \alpha_{\gamma\varpi} + \sum_{\varpi:\,\alpha_{\gamma\varpi}\geq 1} Q_{\gamma\varpi} \\ &= Q_{\gamma\gamma} + \sum_{\varpi:\,\alpha_{\gamma\varpi}<1} Q_{\gamma\varpi} + \sum_{\varpi:\,\alpha_{\gamma\varpi}\geq 1} Q_{\gamma\varpi} \\ &= Q_{\gamma\gamma} + \sum_{\varpi \neq \gamma} Q_{\gamma\varpi} \\ &= \sum_{\varpi} Q_{\gamma\varpi} = 1. \end{aligned}$$

In order to show that π is the unique invariant measure of the generated Markov chain it is enough to demonstrate that

$$\pi(\varpi) = \sum_{\gamma} \pi(\gamma) P_{\gamma\varpi}.$$

Consider the following three cases:

1. $\alpha_{\gamma\varpi} = 1$: then by (4.A.2) it follows that

$$P_{\gamma\varpi} = Q_{\gamma\varpi} = Q_{\varpi\gamma} = P_{\gamma\varpi},$$

and therefore $\pi(\gamma) \cdot P_{\gamma\varpi} = \pi(\varpi) \cdot P_{\varpi\gamma}$,

2. $\alpha_{\gamma\varpi} < 1$: then also by (4.A.2)

$$P_{\gamma\varpi} = Q_{\gamma\varpi} \cdot \alpha_{\gamma\varpi} = Q_{\varpi\gamma} \cdot \alpha_{\gamma\varpi} = P_{\varpi\gamma} \cdot \alpha_{\gamma\varpi},$$

and therefore $\pi(\gamma) \cdot P_{\gamma\varpi} = \pi(\varpi) \cdot P_{\varpi\gamma}$,

3. $\alpha_{\gamma\varpi} > 1$: again by (4.A.2)

$$P_{\gamma\varpi} = Q_{\gamma\varpi} = Q_{\varpi\gamma} = P_{\varpi\gamma} \cdot \alpha_{\gamma\varpi},$$

and also $\pi(\gamma) \cdot P_{\gamma\varpi} = \pi(\varpi) \cdot P_{\varpi\gamma}$.

Consequently the balance equation $\pi(\gamma)P_{\gamma\varpi} = \pi(\varpi)P_{\varpi\gamma}$ holds for all values of γ, ϖ, and since $P_{\gamma\varpi}$'s form a stochastic matrix, it follows that:

$$\sum_{\gamma} \pi(\gamma) \cdot P_{\gamma\varpi} = \sum_{\gamma} \pi(\varpi) \cdot P_{\varpi\gamma} = \pi(\varpi) \cdot \sum_{\gamma} P_{\varpi\gamma} = \pi(\varpi).$$

□

Appendix 4.B. INVARIANT DISTRIBUTION OF THE GIBBS SAMPLER

Using the same notation as in Section 4.2.3 it will be shown here that the Gibbs distribution π is the invariant distribution of the Gibbs sampler [26].

For a fixed time τ and some $\zeta \in \Omega$ it follows that:

$$\begin{aligned} \big(\pi\Xi(\tau)\big)_{\gamma} &= \sum_{\zeta} \pi(\zeta) \cdot \Xi_{\zeta,\gamma}(\tau) \\ &= \sum_{\varpi \in \mathcal{S}'_{\mathbf{d}}} \pi(\gamma^{(\varpi,n_\tau)}) \cdot \Xi_{\gamma^{(\varpi,n_\tau)},\gamma}(\tau). \end{aligned} \tag{4.B.3}$$

The last summation extends over all possible $\varpi \in \mathcal{S}'_{\mathbf{d}}$. Note, however, that the transition probability of going from state ϖ at site n_τ (time τ) to a new state γ_{n_τ} does not depend on the state of the site n_τ, but only on the states of its neighbours. Hence the following relationship holds:

$$\Xi_{\gamma^{(\vartheta,n_\tau)},\gamma}(\tau) = \Xi_{\gamma^{(\varpi,n_\tau)},\gamma}(\tau) \qquad (\text{ for any } \vartheta \in \mathcal{S}'_{\mathbf{d}}).$$

Since this summand is independent of the previous state ϖ it can be considered a constant in the summation, and it follows that

$$\begin{aligned}\left(\pi\Xi(\tau)\right)_\gamma =& \Xi_{\gamma^{(\vartheta,n_\tau)},\gamma}(\tau)\cdot \sum_{\varpi\in\mathcal{S}'_{\mathbf{d}}} \pi(\gamma^{(\varpi,n_\tau)}) \qquad (\text{ for any } \vartheta\in\mathcal{S}'_{\mathbf{d}})\\ =& \pi(\Gamma_{n_\tau}=\gamma_{n_\tau}|\Gamma_i=\gamma_i,\forall i:(\mathbf{x}_i,t)\in\Lambda_{\mathbf{d}},i\neq n_\tau)\cdot\\ & \pi(\Gamma_i=\gamma_i,\forall i:(\mathbf{x}_i,t)\in\Lambda_{\mathbf{d}},i\neq n_\tau)\\ =& \pi(\gamma).\end{aligned} \tag{4.B.4}$$

Appendix 4.C. IMPLEMENTATION OF THE GIBBS SAMPLER FOR VECTOR MRFS

In practice the Gibbs sampler for Vector MRFs differs from that for Scalar MRFs. The way it has been implemented for the purpose of this research is presented below.

The conditional probability $P(\mathbf{D}_t=\hat{\mathbf{d}}_t|G_{t_-}=g_{t_-},G_{t_+}=g_{t_+})$ driving the Gibbs sampler has a non-quadratic local energy $U^i_{\mathbf{d}}$ with respect to the displacement vectors $\hat{\mathbf{d}}_t$ (due to the displaced pel difference). This precludes the use of a simple transformation of uniform or normal random variables efficiently generated by standard routines. Hence the complete probability distribution (local energy) must by computed at each location $(\mathbf{x}_i,t)$.

Since the states to be generated are 2D vectors, let the state-space $\mathcal{S}'_{\mathbf{d}}$ of each vector be the Cartesian product of state-spaces in horizontal and vertical directions ($\mathcal{S}'_{\mathbf{d}_h}$ and $\mathcal{S}'_{\mathbf{d}_v}$ respectively):

$$\mathcal{S}'_{\mathbf{d}}=\mathcal{S}'_{\mathbf{d}_h}\times\mathcal{S}'_{\mathbf{d}_v},\qquad M_h=|\mathcal{S}'_{\mathbf{d}_h}|,\qquad M_v=|\mathcal{S}'_{\mathbf{d}_v}|,$$

where $|\mathcal{S}'_{\mathbf{d}}|$ stands for cardinality of set $\mathcal{S}'_{\mathbf{d}}$. Let z be a vector from $\mathcal{S}'_{\mathbf{d}}$. Let also z_i and z_j be any two members of $\mathcal{S}'_{\mathbf{d}_h}$ and $\mathcal{S}'_{\mathbf{d}_v}$, respectively. Then the cumulative distribution arrays A (2D) and B (1D) are computed as follows:

$B_0=0$
For $j=1,...,M_v$ **compute**
  $A_{(0,j)}=0$
  For $i=1,...,M_h$ **compute**
    $z=(z_i,z_j)$

$$A_{(i,j)} = A_{(i-1,j)} + e^{-U^l(z,g_{t_-},g_{t_+})}$$

Continue

$$B_j = B_{j-1} + A_{(M_h,j)}$$

Continue

The new vector $\boldsymbol{w} \in \mathcal{S}'_{\mathbf{d}}$ is obtained by generating two pseudo-random uniformly distributed numbers (r_h, r_v). First, r_v is generated in the range $(0, B_{M_v}]$, and $\boldsymbol{w}^v$ is assigned the following value:

$$\boldsymbol{w}^v = \boldsymbol{z}_j, \qquad r_v \in (B_{j-1}, B_j]. \tag{4.C.5}$$

Then r_h is generated in the range $(0, A_{(M_h,j)}]$ (for j from (4.C.5)), and $\boldsymbol{w}^h$ is assigned the value

$$\boldsymbol{w}^h = \boldsymbol{z}_i, \qquad r_h \in (A_{(i-1,j)}, A_{(i,j)}]. \tag{4.C.6}$$

Appendix 4.D. MEAN AND COVARIANCE FOR THE CONTINUOUS STATE-SPACE GIBBS SAMPLER

The notation used here is that of Section 4.5. Let the subscripts x and y denote horizontal and vertical components of appropriate vectors i.e., $\hat{\mathbf{d}} = [\hat{d}_x, \hat{d}_y]$ and $\dot{\mathbf{d}} = [\dot{d}_x, \dot{d}_y]$. Using the definition of potential (3.17) rewrite the local energy (4.17) as follows

$$\begin{aligned} U^i_{\mathbf{d}}(\hat{\mathbf{d}}(\mathbf{x}_i,t)|\hat{\mathbf{d}}, g_{t_-}, g_{t_+}) = \lambda'_g \cdot [&\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t) - \\ &\tilde{r}^x(\dot{\mathbf{d}}(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t) \cdot \dot{d}_x(\mathbf{x}_i,t) - \tilde{r}^y(\dot{\mathbf{d}}(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t) \cdot \dot{d}_y(\mathbf{x}_i,t) + \\ &\tilde{r}^x(\dot{\mathbf{d}}(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t) \cdot \hat{d}_x(\mathbf{x}_i,t) + \tilde{r}^y(\dot{\mathbf{d}}(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t) \cdot \hat{d}_y(\mathbf{x}_i,t)]^2 + \\ &\lambda'_{\mathbf{d}} \cdot \sum_{j:\, \mathbf{x}_j \in \eta_{\mathbf{d}}(\mathbf{x}_i)} [\hat{d}_x(\mathbf{x}_i,t) - \hat{d}_x(\mathbf{x}_j,t)]^2 + [\hat{d}_y(\mathbf{x}_i,t) - \hat{d}_y(\mathbf{x}_j,t)]^2. \end{aligned}$$

To simplify the notation the dependence of $\tilde{r}^x$ and $\tilde{r}^y$ on $(\dot{\mathbf{d}}(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t)$ as well as the dependence of $\hat{d}_x$ and $\hat{d}_y$ on $(\mathbf{x}_i, t)$ will be omitted for the rest of this appendix. After some arithmetical manipulations $U^i_{\mathbf{d}}$ can be rewritten as

$$U^i_{\mathbf{d}}(\mathbf{d}(\mathbf{x}_i,t)|\mathbf{d}, g_{t_-}, g_{t_+}) = A \cdot (\hat{d}_x)^2 + B \cdot (\hat{d}_y)^2 + C \cdot \hat{d}_x \cdot \hat{d}_y + D \cdot \hat{d}_x + E \cdot \hat{d}_y + F, \tag{4.D.7}$$

where the parameters A, B, C, D, E are defined as follows

$$A = \lambda'_g(\tilde{r}^x)^2 + \xi_i\lambda'_{\mathbf{d}},$$
$$B = \lambda'_g(\tilde{r}^y)^2 + \xi_i\lambda'_{\mathbf{d}},$$
$$C = 2\lambda'_g\tilde{r}^x\tilde{r}^y,$$
$$D = 2[\lambda'_g\tilde{r}^x(\tilde{r} - \tilde{r}^x\dot{d}_x - \tilde{r}^y\dot{d}_y) - \xi_i\lambda'_{\mathbf{d}}\overline{d}_x],$$
$$E = 2[\lambda'_g\tilde{r}^y(\tilde{r} - \tilde{r}^x\dot{d}_x - \tilde{r}^y\dot{d}_y) - \xi_i\lambda'_{\mathbf{d}}\overline{d}_y],$$

and F is a constant depending on the neighbouring estimates $(\hat{\mathbf{d}}(\mathbf{x}_j, t) : \mathbf{x}_j \neq \mathbf{x}_i)$, approximate estimate $\dot{\mathbf{d}}$ and displaced pel difference at $\dot{\mathbf{d}}$. The vector $\overline{\mathbf{d}} = [\overline{d}_x, \overline{d}_y]$ denotes the average over the neighbouring estimates, and has been defined in (4.19).

A bivariate Gaussian distribution of the vector $\hat{\mathbf{d}}$ can be described by the following expression:

$$p(\hat{\mathbf{d}}) = \frac{1}{2\pi(detM)^{\frac{1}{2}}} e^{-\frac{1}{2}(\hat{\mathbf{d}}-\mathbf{m})M^{-1}(\hat{\mathbf{d}}-\mathbf{m})^T},$$

where $\mathbf{m} = [m_x, m_y]$ is the mean and the inverse of the covariance matrix is defined as follows

$$M^{-1} = \frac{1}{\sigma_x^2\sigma_y^2(1-\rho^2)} \begin{bmatrix} \sigma_y^2 & -\rho\sigma_x\sigma_y \\ -\rho\sigma_x\sigma_y & \sigma_x^2 \end{bmatrix}.$$

σ_x^2 and σ_y^2 denote the variances of horizontal and vertical components respectively, and ρ is the correlation coefficient.

The exponent of this bivariate Gaussian distribution can be expressed in the same form as the local energy (4.D.7). After some manipulations and comparing appropriate terms it can be shown that

$$A = \frac{1}{2\sigma_x^2(1-\rho^2)},$$
$$B = \frac{1}{2\sigma_y^2(1-\rho^2)},$$
$$C = -\frac{\rho}{\sigma_x\sigma_y(1-\rho^2)},$$
$$D = \frac{1}{\sigma_x(1-\rho^2)}(\frac{\rho m_y}{\sigma_y} - \frac{m_x}{\sigma_x}),$$
$$E = \frac{1}{\sigma_y(1-\rho^2)}(\frac{\rho m_x}{\sigma_x} - \frac{m_y}{\sigma_y}).$$

Since F can be dropped (it is just a constant scaling the distribution so that it is a probability measure), there are 5 equations and 5 unknowns. Some laborious manipulations

will show that the means, the variances and the correlation factor are described by the following expressions:

$$m_x = \bar{d}_x - \frac{\tilde{r}(\mathbf{x}_i, \Delta t, \dot{\mathbf{d}}) + \tilde{r}^x(\bar{d}_x - \dot{d}_x) + \tilde{r}^y(\bar{d}_y - \dot{d}_y)}{\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^x)^2 + (\tilde{r}^y)^2} \cdot \tilde{r}^x$$

$$m_y = \bar{d}_y - \frac{\tilde{r}(\mathbf{x}_i, \Delta t, \dot{\mathbf{d}}) + \tilde{r}^x(\bar{d}_x - \dot{d}_x) + \tilde{r}^y(\bar{d}_y - \dot{d}_y)}{\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^x)^2 + (\tilde{r}^y)^2} \cdot \tilde{r}^y$$

$$\sigma_x^2 = \frac{\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^y)^2}{2\xi_i \lambda'_d (\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^x)^2 + (\tilde{r}^y)^2)}$$

$$\sigma_y^2 = \frac{\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^x)^2}{2\xi_i \lambda'_d (\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^x)^2 + (\tilde{r}^y)^2)}$$

$$\rho\sigma_x\sigma_y = \frac{-\tilde{r}^x\tilde{r}^y}{2\xi_i \lambda'_d (\xi_i \frac{\lambda'_d}{\lambda'_g} + (\tilde{r}^x)^2 + (\tilde{r}^y)^2)}.$$

Appendix 4.E. 1-D INTERPOLATION

In this appendix the impulse responses of linear, quadratic and cubic interpolators will be computed from the interpolating equations (Section 4.6), and a C^1 continuous cubic interpolator will be derived. The notation will be that used in Section 4.6.

4.E.1 Impulse response of an interpolating filter

Note that the convolution (4.25) can be written as follows

$$\tilde{w}(x) = \ldots + a(x - \lfloor x \rfloor_\Lambda + \delta) \cdot w(\lfloor x \rfloor_\Lambda - \delta) + a(x - \lfloor x \rfloor_\Lambda) \cdot w(\lfloor x \rfloor_\Lambda) +$$
$$a(x - \lfloor x \rfloor_\Lambda - \delta) \cdot w(\lfloor x \rfloor_\Lambda + \delta) + \ldots, \quad x \in R,$$

where $\lfloor x \rfloor_\Lambda \in \Lambda$ as defined in Section 4.6. Since $x - \lfloor x \rfloor_\Lambda = \Delta x$, the above equation can be rewritten as

$$\tilde{w}(x) = \ldots + a(\Delta x + \delta) \cdot w(\lfloor x \rfloor_\Lambda - \delta) + a(\Delta x) \cdot w(\lfloor x \rfloor_\Lambda) +$$
$$a(\Delta x - \delta) \cdot w(\lfloor x \rfloor_\Lambda + \delta) + \ldots, \quad x \in R.$$

Comparison of the two terms of equation (4.22) for the linear interpolator with appropriate terms of the above equation will result in the following expression for the interpolator

impulse response $a(x)$:

$$a(x) = \begin{cases} 0 & \text{for } x < -1.0 \\ 1 + x & \text{for } -1.0 \leq x < 0.0 \\ 1 - x & \text{for } 0.0 \leq x < 1.0 \\ 1 + x & \text{for } 1.0 \leq x. \end{cases}$$

Similarly it can be easily verified that the impulse response for the quadratic interpolator is:

$$a(x) = \begin{cases} 0 & \text{for } x < -1.5 \\ \frac{1}{2}(x^2 + 3x + 2) & \text{for } -1.5 \leq x < -0.5 \\ -x^2 + 1 & \text{for } -0.5 \leq x < 0.5 \\ \frac{1}{2}(x^2 - 3x + 2) & \text{for } 0.5 \leq x < 1.5 \\ 0 & \text{for } 1.5 \leq x, \end{cases}$$

and for the cubic interpolator:

$$a(x) = \begin{cases} 0 & \text{for } x < -2.0 \\ \frac{1}{6}(x^3 + 6x^2 + 11x + 6) & \text{for } -2.0 \leq x < -1.0 \\ -\frac{1}{2}(x^3 + 2x^2 - x - 2) & \text{for } -1.0 \leq x < 0.0 \\ \frac{1}{2}(x^3 - 2x^2 - x + 2) & \text{for } 0.0 \leq x < 1.0 \\ -\frac{1}{6}(x^3 - 6x^2 + 11x - 6) & \text{for } 1.0 \leq x < 2.0 \\ 0 & \text{for } 2.0 \leq x. \end{cases}$$

The derivatives of the impulse responses follow immediately from the above expressions. The impulse responses and their derivatives for all three interpolators are shown in Figs. 4.3 and 4.4 of Section 4.6, respectively

4.E.2 C^1-continuous impulse response design

To design a C^1-continuous third-order piecewise-polynomial impulse response, the coefficient vector $[A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2]$ of the following function

$$a(x) = \begin{cases} 0 & \text{for } x < -2.0 \\ -A_2x^3 + B_2x^2 - C_2x + D_2 & \text{for } -2.0 \leq x < -1.0 \\ -A_1x^3 + B_1x^2 - C_1x + D_1 & \text{for } -1.0 \leq x < 0.0 \\ A_1x^3 + B_1x^2 + C_1x + D_1 & \text{for } 0.0 \leq x < 1.0 \\ A_2x^3 + B_2x^2 + C_2x + D_2 & \text{for } 1.0 \leq x < 2.0 \\ 0 & \text{for } 2.0 \leq x. \end{cases}$$

must be computed. Note that due to the required symmetry of the interpolator (linear-phase filter) only 2 polynomials are independent. To insure that the above impulse response is a valid interpolator and to provide the continuity of this response, the following constraints have to be satisfied:

1. $a(x = 0) = 1.0 \Rightarrow D_1 = 1.0$

2. $a(x = 1^-) = 0.0 \Rightarrow A_1 + B_1 + C_1 + D_1 = 0.0$
3. $a(x = 1^+) = 0.0 \Rightarrow A_2 + B_2 + C_2 + D_2 = 0.0$
4. $a(x = 2^-) = 0.0 \Rightarrow 8A_2 + 4B_2 + 2C_2 + D_2 = 0.0.$

Moreover, to provide continuity of the impulse response derivative also this set of constraints has to be satisfied:

1. $a'(x = 0^-) = a'(x = 0^+) = 1.0 \Rightarrow C_1 = 0.0$
2. $a'(x = 1^-) = a'(x = 1^+) = 0.0 \Rightarrow 3A_1 + 2B_1 + C_1 = 3A_2 + 2B_2 + C_2$
3. $a'(x = 2^-) = 0.0 \Rightarrow 12A_2 + 4B_2 + C_2 = 0.0.$

There are only 7 constraints while 8 unknown coefficients, hence 1 degree of freedom allows to specify one coefficient according to some extra criterion, for example as proposed by Keys [50]. He expressed all other coefficients in terms of A_2 and chose it so that the output (interpolated) signal agrees with the first three terms of the Taylor series expansion of the input signal. This criterion resulted in $A_2 = -0.5$, and provided the following impulse response:

$$a(x) = \begin{cases} 0 & \text{for } x < -2.0 \\ 0.5x^3 + 2.5x^2 + 4.0x + 2.0 & \text{for } -2.0 \leq x < -1.0 \\ -1.5x^3 - 2.5x^2 + 1.0 & \text{for } -1.0 \leq x < 0.0 \\ 1.5x^3 - 2.5x^2 + 1.0 & \text{for } 0.0 \leq x < 1.0 \\ -0.5x^3 + 2.5x^2 - 4.0x + 2.0 & \text{for } 1.0 \leq x < 2.0 \\ 0 & \text{for } 2.0 \leq x. \end{cases}$$

The impulse response and its derivative are plotted in Figs. 4.5 and 4.6 of Section 4.6, respectively

Chapter 5

HIERARCHICAL BAYESIAN ESTIMATION OF MOTION

In this chapter the Bayesian estimation based on the MAP criterion will be incorporated into the hierarchical framework. Many motion estimation methods, which cannot estimate large displacements due to violation of certain underlying assumptions (like linearity of image intensity), must use a hierarchical approach. In the case of the discrete state-space MAP estimation presented in Chapter 3, however, the only concern is computational efficiency. It will be demonstrated that a MAP estimator can attain the same optimum at a single scale, however at significantly increased computational cost.

In the next section I will explain why the low-pass filtering of images is so helpful in matching. Then, the hierarchical MAP estimation over discrete and continuous state-spaces will be presented, followed by the discussion of filters used for image pyramid generation and of adjustment of parameters λ. The chapter will be concluded with some experimental results.

5.1 WHY DOES IMAGE FILTERING HELP IN MOTION ESTIMATION ?

The benefit of image pre-filtering before performing motion estimation has been observed by numerous researchers (see Section 2.4), however usually it has been explained on the basis of informal reasoning. Here I will show why the filtering helps in a more rigorous way. I will consider the simple 1-D example of signal matching presented in Section 2.7.

Without loss of generality consider two infinite-length signals f and g. When matching of two signals is performed, they are usually assumed to be closely related rather than being

completely arbitrary. Hence, let g be a transformed and shifted copy of f:

$$g(x) = s(x) * f(x - d_0),$$

where $s(x)$ is an impulse response of a linear operator, $*$ is a linear convolution and d_0 is a known displacement. This displacement between the signals f and g can be estimated by minimizing the following objective function:

$$\phi(\widehat{d}) = \sum_{x=-\infty}^{\infty} [f(x) - g(x + \widehat{d})]^2,$$

which expresses the quality of matching i.e., the better the match, the smaller ϕ. To make the problem tractable mathematically assume that the estimator $\widehat{d}$ is constant over the whole domain (independent of x). Let f' and g' be the filtered versions of f and g:

$$f'(x) = h(x) * f(x)$$

$$g'(x) = h(x) * g(x),$$

where $h(x)$ is an impulse response of an LSI filter. Consider now the same objective function ϕ, but applied to the filtered data, and denote it by ϕ':

$$\phi'(\widehat{d}) = \sum_{x=-\infty}^{\infty} [f'(x) - g'(x + \widehat{d})]^2.$$

Using the above notation the relationship between $\phi(\widehat{d})$ and $\phi'(\widehat{d})$ is established by the following theorem:

Theorem: If $H(\nu)$ is the frequency response of the linear shift-invariant filter $h(x)$ with real-valued coefficients, then the Fourier transforms $\Phi(\nu)$ and $\Phi'(\nu)$ of $\phi(\widehat{d})$ and $\phi'(\widehat{d})$, respectively, are related through the following equation:

$$\Phi'(\nu) = \frac{1}{2\pi}\delta(\nu) \cdot (A' - A \cdot |H(0)|^2) + \Phi(\nu) \cdot |H(\nu)|^2,$$

where δ is a Dirac impulse and A, A' are constants dependent on signal f and operator s.

The proof of this theorem is given in Appendix 5.A.

The above equation shows that $\phi'(\widehat{d})$ is equal to a DC term plus $\phi(\widehat{d})$ convolved with a filter. If $h(x)$ is a zero-phase filter i.e., $h(x) = h(-x)$, then multiplication by $|H(\nu)|^2$ in the frequency domain corresponds to a cascade of two $h(x)$ filters in the time domain.

Hence, applying a low-pass filter to the data results in a double low-pass filtering of the objective function ϕ. If the data is such that ϕ is multimodal and has numerous local minima, the low-pass filtering operation will smooth-out some of those local minima. If the low-pass filtering is sufficiently severe, for example a cascade of filters is applied to construct a pyramid, then only one minimum may be left. Recall the 1-D example from Section 2.7. Fig. 2.4 shows the objective function ϕ over a hierarchy of resolutions obtained through low-pass filtering. Note that after single filtering ϕ is still multimodal. Even if it is difficult to see, this is the case after two filterings as well (a simple run of the Gauss-Newton algorithm from the initial displacement d_0=-200.0 confirms that). Only after the third filtering does ϕ become unimodal, allowing the optimization algorithm to quickly locate the optimum. There is, however, no certainty that this single minimum of the objective function for the filtered data will correspond exactly to the global minimum of the unfiltered data. Hence, the position of the optimum must be reevaluated at the higher resolutions of the data by running the same optimization algorithm starting from the lower-resolution estimate. The above procedure describes the very principle of hierachical approach to matching, motion estimation or any other suitable problem.

The filtering applied to the data makes the objective function (in the limit) unimodal. It can be also viewed as an attempt to make this function convex. This draws an immediate parallel with the *Graduated Non-Convexity* algorithm proposed by Blake and Zisserman [12]. They overcome the non-convexity of their objective function by constructing a sequence of approximations of the original objective function. At one end of the sequence is a convex approximation while at the other end is the original non-convex objective function, with some compromising functions between. The hierarchical approach can be viewed from this perspective as a sequence of objective functions for variable-resolution data, with the limiting case when the objective function is evaluated for averages (DC values).

5.2 HIERARCHICAL EXTENSION OF MAP ESTIMATION

5.2.1 Discrete state-space

The computational requirement of the discrete state-space Gibbs sampler employed

either in MAP or MEC estimation is linearly proportional to the size of the single displacement vector state-space $\mathcal{S}'_{\mathbf{d}}$. Recall that d_{max} denotes the maximum allowed horizontal and vertical displacements. Then, if δ denotes the step size in quantization of the state-space $\mathcal{S}'_{\mathbf{d}}$, its size can be expressed as follows:

$$|\mathcal{S}'_{\mathbf{d}}| = N_d^2, \qquad N_d = (1 + 2d_{max}/\delta).$$

Note that the spatial area covered by the state-space $\mathcal{S}'_{\mathbf{d}}$ is $(-d_{max} : d_{max}, -d_{max} : d_{max})$ with N_d^2 sampling points. Increasing either the maximum allowed displacement or reducing the step size by n, will increase the computational effort by approximately n^2 †. Hence, in order to handle the large displacements efficiently, while maintaining sufficient precision, a hierarchical technique must be devised for the Gibbs sampler. The approach I propose here is the non-recursive multigrid (coarse-to-fine resolution) algorithm. The general principle of this method can be explained as follows. An image pyramid of varying resolutions is constructed for example from the lowest resolution at the top to the full resolution at the bottom of the pyramid. The estimators (displacement fields) also form a pyramid of varying resolutions. The estimation starts at the top of the pyramid (coarse resolution), where the number of displacement lattice sites is small and the equilibrium state is located very quickly. Then, the estimate from this coarse level is interpolated to a finer resolution level, and is used as a coarse solution to be further refined. This hierarchical process is repeated until the full resolution estimate is obtained.

The necessary ingredient of the hierarchical approach is data filtering, however spatial subsampling is not. The data and/or estimator pyramids do not have to be spatially subsampled (from the bottom to the top). The data can be just filtered, without subsampling, and the estimators can be defined over the same full-resolution lattice $\Lambda_{\mathbf{d}}$ across the complete pyramid. The algorithm will work perfectly well, and will be still valuable for such estimation methods which cannot compute large displacements due to violation of some underlying assumptions (e.g., intensity linearity). In this form, however, it will not speed up the computations. It is the spatial subsampling of the displacement fields (moving up the

† For the maximum displacement d_{max}=2 and the step size δ=0.25 pel, the size of $\mathcal{S}'_{\mathbf{d}}$ is 17×17=289, while for d_{max}=4 the size will grow to 33×33=1089.

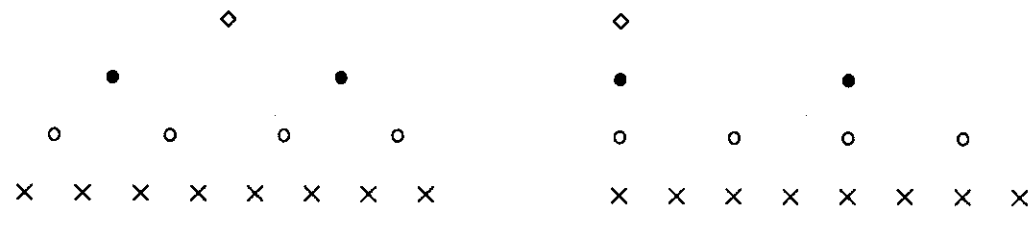
pyramid) which provides the computational gain. At the top of the pyramid the number of vectors to be estimated is relatively small, thus requiring only a fraction of the computational effort (compared to the bottom of the pyramid). Moreover, the increased absolute distance between the vectors enforces quick propagation of the smoothness constraints over large distances (to propagate the same constraint at full resolution over the same absolute distance would require many more iterations), and hence additionally improves the convergence rate of the algorithm. This solution is used at the next higher resolution level as, for example, the initial state. At this level more vectors are involved, but since the approximate solution is known the convergence is fast. The process is repeated until the bottom of the pyramid is reached. Note again, that in principle the images do not have to be subsampled when moving up the pyramid. Subsampling is just a way of reducing the storage requirements for the data pyramid, which is an important consideration for hardware implementations. It causes, however, data loss, since to obtain sub-pixel accuracy at a given level spatial image interpolation must be used from previously subsampled data. I will use here a "constant-width" pyramid for images and a regular pyramid for displacement fields.

There are two types of pyramids: even and odd. In both, the lower resolution level is constructed by spatial subsampling of appropriately filtered higher resolution level. The difference is that in the even pyramid the lower resolution samples are shifted by half of the sampling period of the higher resolution level, while in the odd pyramid there is no such shift (Fig. 5.1).

Let K_l be the number of resolution levels. The following sequence of sample fields

$$\{g_{t\pm}^{\kappa}, \kappa = 0, 1, ..., K_l - 1\}$$

will denote a "constant-width" pyramid of images. Naturally, $g_{t\pm}^{0}$ denotes the full resolution images, and the image $g_{t\pm}^{\kappa}$ is obtained by filtering the image $g_{t\pm}^{\kappa-1}$. The Gaussian (low-pass filtered) [22] and Laplacian (band-pass filtered) [6] pyramids of images have been used in hierarchical motion estimation. Also other filters such as Nyquist-like low-pass 2D separable FIR filters can be used [55]. In the next section I will discuss two types of low-pass filters which will be used later.



(a) even pyramid *(b)* odd pyramid

Fig. 5.1 Schematic (1-D) representation of even and odd pyramids for hierarchical data representation (the odd pyramid is used for displacement fields): κ=0,1,2,3 or ×,o,•,◇.

Let also the sequence of sample fields

$$\{\mathbf{d}_t^\kappa, \kappa = 0, 1, ..., K_l - 1\}$$

denote an odd pyramid of displacement field estimates at various resolution levels. Note that the filtering operation is not applicable in the case of estimator pyramid. Let s_κ denote the spatial subsampling factor of displacement field at level κ with respect to level 0. Then, the size of the field $\mathbf{d}_t^\kappa$ is $(\lfloor (M_{\mathbf{d}}^h - 1)/s_\kappa \rfloor + 1) \times (\lfloor (M_{\mathbf{d}}^v - 1)/s_\kappa \rfloor + 1)$.

Once an estimate is obtained at level κ, it must be transformed to the higher resolution level for subsequent improvement. This operation can be viewed either as a "parent-children" propagation or as an interpolation. I will consider the latter approach. Let the interpolation $\mathcal{I}_\kappa$ between levels κ and $\kappa - 1$ be expressed as follows:

$$\mathbf{b}_t^{\kappa-1} = \mathcal{I}_\kappa(\widehat{\mathbf{d}}_t^\kappa), \tag{5.1}$$

where $\widehat{\mathbf{d}}_t^\kappa$ denotes the final estimate at level κ, and $\mathbf{b}_t^{\kappa-1}$, called here the *base* displacement field at level $(\kappa - 1)$, is the result of interpolation $\mathcal{I}_\kappa$. I will investigate two types of interpolation for the odd pyramid:

1. simple repetition of κ-level estimates at the missing positions of level $(\kappa - 1)$ ("sample and hold"),
2. bilinear interpolation of κ-level estimates to obtain complete $(\kappa - 1)$-level field.

Usually hierarchical motion estimation algorithms use the previous-level estimate as an initial state at the subsequent level and allow arbitrary vectors (within a given state-space) after. Then, the spatial area covered by state-space $\mathcal{S}'_{\mathbf{d}}$ of each vector is identical at each

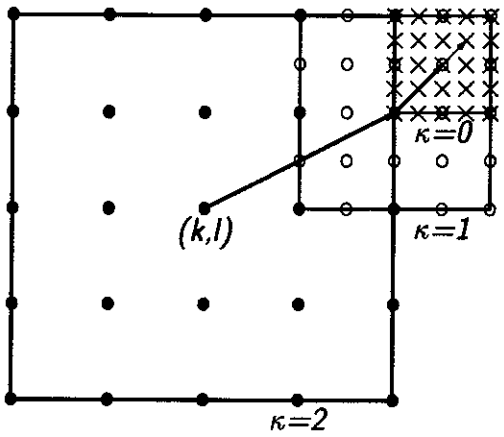


Fig. 5.2 Consecutive single-vector state-spaces for hierarchical estimation with K_l=3 and s_κ=2 (κ=0,1,2).

resolution. This strategy is not suitable for the discrete state-space Gibbs sampler, because the key problem is to speed up the computations or equivalently to restrict $|\mathcal{S}'_{\mathbf{d}}|$. However, if the solution from the previous level is close to the optimal one, only a limited spatial area around this initial solution can be searched for the new estimate. Let d^κ_{max} denote the maximum allowed displacement at level κ and let δ^κ specify the displacement step size at level κ. If the ratio $d^\kappa_{max}/\delta^\kappa$ is constant for all κ, then a sequence of state-spaces with constant $\mathcal{N}_{\mathbf{d}}$, but covering smaller and smaller areas with decreasing κ, results (Fig. 5.2). In other words at the top of the pyramid a search with large step size, thus covering large spatial area, is performed, while at the bottom only a small area is covered but with high precision.

The *base* displacement field $\mathbf{b}^\kappa_t$ will not be used at level κ merely as the initial state, but as a coarse solution which is fine-tuned at subsequent levels. The field $\mathbf{b}^\kappa_t$ is a fixed array of vectors, and is used to identify the centers of new single-vector state-spaces at level κ as depicted in Fig. 5.2. Application of the MAP estimation algorithm yields an *incremental* displacement field $\widehat{\mathbf{h}}^\kappa_t$, so that the final estimate at level κ is given by

$$\widehat{\mathbf{d}}^\kappa_t = \widehat{\mathbf{h}}^\kappa_t + \mathbf{b}^\kappa_t. \tag{5.2}$$

In this manner obviously a non-homogeneous total state-space $\mathcal{S}_{\mathbf{d}}$ has been generated. It has a finer precision around the expected final estimate $\widehat{\mathbf{d}}_t$.

The new energy function for the hierarchical MAP estimation can be expressed as follows:

$$U(\hat{\mathbf{h}}_t^\kappa, \mathbf{b}_t^\kappa, g_{t_-}, g_{t_+}) = \lambda_g^\kappa \cdot \sum_{i=1}^{M_\mathbf{d}} [\tilde{r}^\kappa(\hat{\mathbf{h}}^\kappa(\mathbf{x}_i,t) + \mathbf{b}^\kappa(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t)]^2 + \lambda_\mathbf{d}^\kappa \cdot U_\mathbf{d}(\hat{\mathbf{h}}_t^\kappa + \mathbf{b}_t^\kappa), \tag{5.3}$$

where the *incremental* displacement vectors $\hat{\mathbf{h}}^\kappa(\mathbf{x}_i,t)$ are variable, while the *base* displacement vectors $\mathbf{b}^\kappa(\mathbf{x}_i,t)$ are fixed for given κ. $\tilde{r}^\kappa$ is the displaced pel difference as before but evaluated for images $g_{t_\pm}^\kappa$.

Fig. 5.3 shows the flow graph of the algorithm for hierarchical MAP estimation of motion based on the above ideas. The initial *base* displacement field is denoted by $\dot{\mathbf{b}}_t$, the initial *incremental* displacement field at resolution level κ is $\dot{\mathbf{h}}_t^\kappa$, and the initial temperature at level κ is $\mathbf{T}^\kappa$.

The computational gain provided by the hierarchical approach compared to the single-level method can be evaluated as follows. Recall that the full-resolution displacement field size is $M_\mathbf{d}^h$ by $M_\mathbf{d}^v$. If K_l resolution levels are used then the total maximum displacement is $\sum_{\kappa=0}^{K_l-1} d_{max}^\kappa$. Consequently the computational effort involved in performing one iteration of the single-level method is

$$\zeta_1 = M_\mathbf{d}^h \times M_\mathbf{d}^v \times (1 + \frac{2}{\delta^0} \sum_{\kappa=0}^{K_l-1} d_{max}^\kappa)^2.$$

As far as the multi-level method is concerned recall that the inter-level subsampling at level κ is denoted by s_κ. Since the displacement field size at level κ is $\lfloor (M_\mathbf{d}^h - 1)/s_\kappa \rfloor + 1$ by $\lfloor (M_\mathbf{d}^v - 1)/s_\kappa \rfloor + 1$, the computational effort associated with K_l-level hierarchical algorithm is

$$\zeta_{K_l} = \sum_{\kappa=0}^{K_l-1} (\lfloor \frac{M_\mathbf{d}^h - 1}{\mathbf{s}_\kappa} \rfloor + 1) \times (\lfloor \frac{M_\mathbf{d}^v - 1}{\mathbf{s}_\kappa} \rfloor + 1) \times (1 + \frac{2}{\delta^\kappa} d_{max}^\kappa)^2.$$

The maximum allowed displacement at level κ is related to the maximum displacement at full-resolution level 0 as follows

$$d_{max}^\kappa = d_{max}^0 \cdot \prod_{i=0}^{\kappa} s_i,$$

where obviously s_κ^0=1. The displacement vector step sizes δ^κ can be arbitrary, however it is reasonable to increase this step size at low resolution levels to speed up the algorithm.

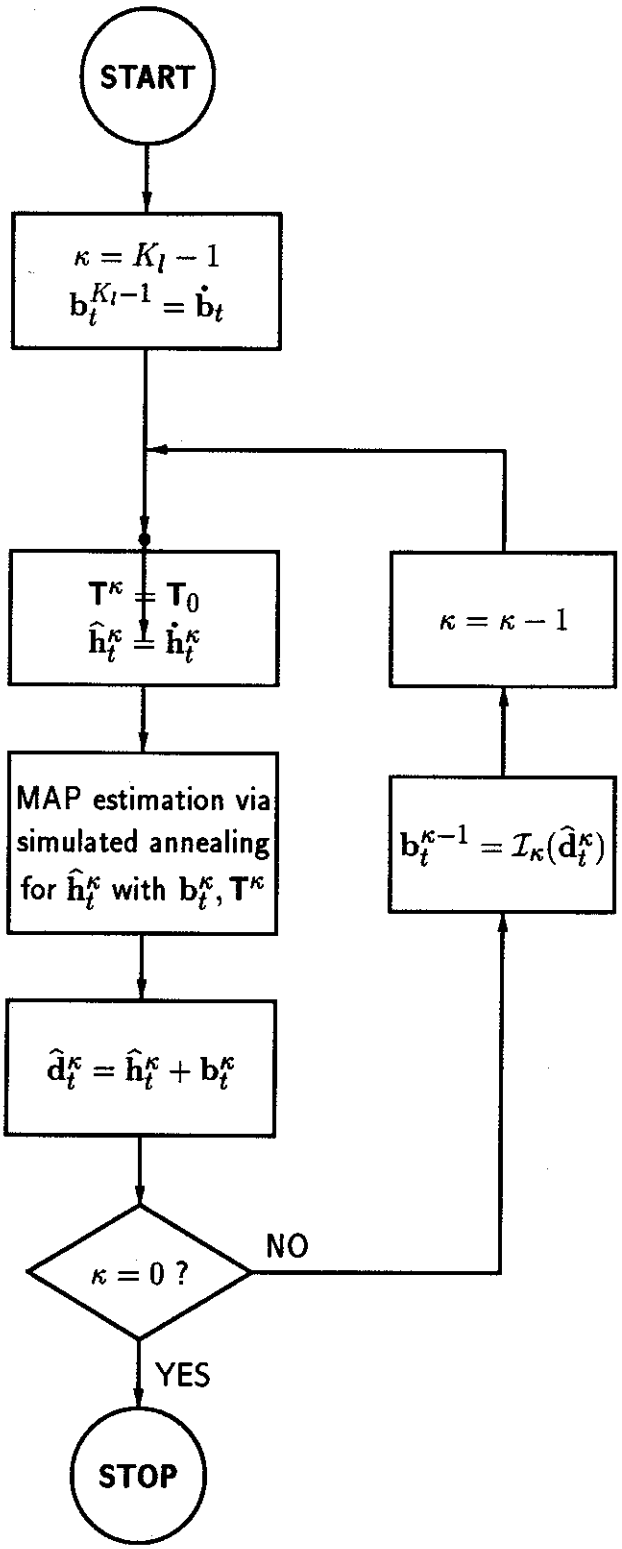


Fig. 5.3 Flow graph of the algorithm for non-recursive hierarchical MAP estimation of motion based on simulated annealing.

Here I will assume constant state-space size $|\mathcal{S}'_{\mathbf{d}}| = N_d^2$ over the hierarchy of scales. Hence, the step sizes will be defined as follows:

$$\delta^\kappa = \delta^0 \cdot \prod_{i=0}^{\kappa} s_i.$$

This results in $d^\kappa_{max}/\delta^\kappa = d^0_{max}/\delta^0$, and consequently in the state-space size $|\mathcal{S}'_{\mathbf{d}}| = N_d^2$ with $N_d = 1 + 2d^0_{max}/\delta^0$.

The computational gain provided by the hierarchical approach is clearly dependent on displacement field size, required maximum displacement, step size and inter-level subsampling. For the following set of parameters (used to produce some results of this chapter)

$$K_l = 3, \quad M^h_{\mathbf{d}} = 221, \quad M^v_{\mathbf{d}} = 69, \quad d^0_{max} = 1, \quad \delta^0 = 0.25, \quad s_1 = s_2 = 2,$$

the ratio ζ_1/ζ_{K_l} equals 30.4. By using more resolution levels or larger inter-level subsampling this ratio can be increased. Even with these modestly chosen parameters, the computational gain is quite impressive.

5.2.2 Continuous state-space

The ideas from previous section, including the block diagram from Fig. 5.3, apply also to MAP estimation over the continuous state-space of displacement vectors. The only difference is the actual implementation of the Gibbs sampler. Recall that at every resolution level only the *incremental* displacement $\widehat{\mathbf{h}}^\kappa_t$ is variable while the *base* displacement $\mathbf{b}^\kappa_t$ is constant. Using the relationship (5.2) in the iterative update (4.20) it follows that the update in the hierarchical algorithm can be described as

$$\begin{aligned}(\widehat{\mathbf{h}}^\kappa)^{n+1}(\mathbf{x}_i,t) =& (\overline{\mathbf{h}}^\kappa)^n(\mathbf{x}_i,t) + \overline{\mathbf{b}}^\kappa(\mathbf{x}_i,t) - \mathbf{b}^\kappa(\mathbf{x}_i,t) - \\ & \frac{\varepsilon_i}{\mu_i}\nabla^T_{\overline{\mathbf{h}}}\tilde{r}((\overline{\mathbf{h}}^\kappa)^n(\mathbf{x}_i,t) + \overline{\mathbf{b}}^\kappa(\mathbf{x}_i,t), \mathbf{x}_i, t, \Delta t) + n_i, \quad \forall i\end{aligned} \tag{5.4}$$

where $(\widehat{\mathbf{h}}^\kappa)^n$ denotes $\widehat{\mathbf{h}}$ at resolution level κ and iteration n. $\overline{\mathbf{b}}$ and $\overline{\mathbf{h}}$ denote averages of $\mathbf{b}$ and $\mathbf{h}$ respectively (Section 4.5). The constants ε_i and μ_i, and the covariance matrix are defined exactly as in Section 4.5 with $(\overline{\mathbf{h}}^\kappa)^n + \overline{\mathbf{b}}^\kappa$ replacing $\widehat{\mathbf{d}}^n$ and λ^κ_g, $\lambda^\kappa_{\mathbf{d}}$, replacing λ_g, $\lambda_{\mathbf{d}}$ respectively.

Except for the way the *incremental* displacement field is computed, other ingredients like filtering, inter-level interpolation etc., are identical to those used in the discrete state-space version. The computational savings due to the hierarchical structure of the algorithm

come from the reduced number of lattice points at lower resolution levels and subsequently from faster convergence at higher resolution levels because of the known coarse estimates. Unlike in the case of the discrete state-space algorithm, the state-space size is not a concern here, hence the algorithm is much less involved computationally.

5.3 FILTER CHOICE FOR HIERARCHICAL ESTIMATION

Most frequently the low-pass Gaussian and band-pass Laplacian filters have been used in motion estimation [31], [6], [22]. It is not quite clear, however, what frequency response (or other properties) such a filter should have. The estimator is subsampled spatially by factor s_κ between each two resolution levels or in other words its resolution is reduced by s_κ. Hence, it is reasonable to require that the data resolution also be reduced by s_κ, for example by filtering with $1/s_\kappa$-band filter [55]. Such a filter will assure no aliasing effects in the subsampled data, which can significantly deteriorate the quality of motion estimates. Here, despite the fact that the image pyramid is not subsampled spatially, I will still use this type of filter.

Two significantly different (frequency response) low-pass filters will be used in hierarchical MAP estimation:

1. separable Gaussian filter with 5 independent coefficients and variance 2.5, similar to that used by Enkelmann [22],
2. separable low-pass filter with 13 independent coefficients; each 1-D component filter is an approximation of the appropriate Nyquist filter, hence it will be called subsequently a Nyquist-like filter.

To generate the Gaussian pyramid the same filter is applied between every two consecutive resolution levels. Hence, a filtering operation from level κ to level $\kappa + n$ is equivalent to n-fold Gaussian filtering. The frequency response magnitudes for the single and double Gaussian filters are shown in Fig. 5.4. The Nyquist-like filtering has been implemented in a parallel manner i.e., the resolution level 1 is obtained from level 0 by approximation to half-band filtering (s_κ=2), level 2 from level 0 by approximation to quarter-band filtering etc. The frequency response magnitudes for both filters are also shown in Fig. 5.4. Note that the Gaussian filters have significantly lower cut-off frequency than the corresponding Nyquist-like filters. Since the Nyquist-like filters must compromise wide bandwidth

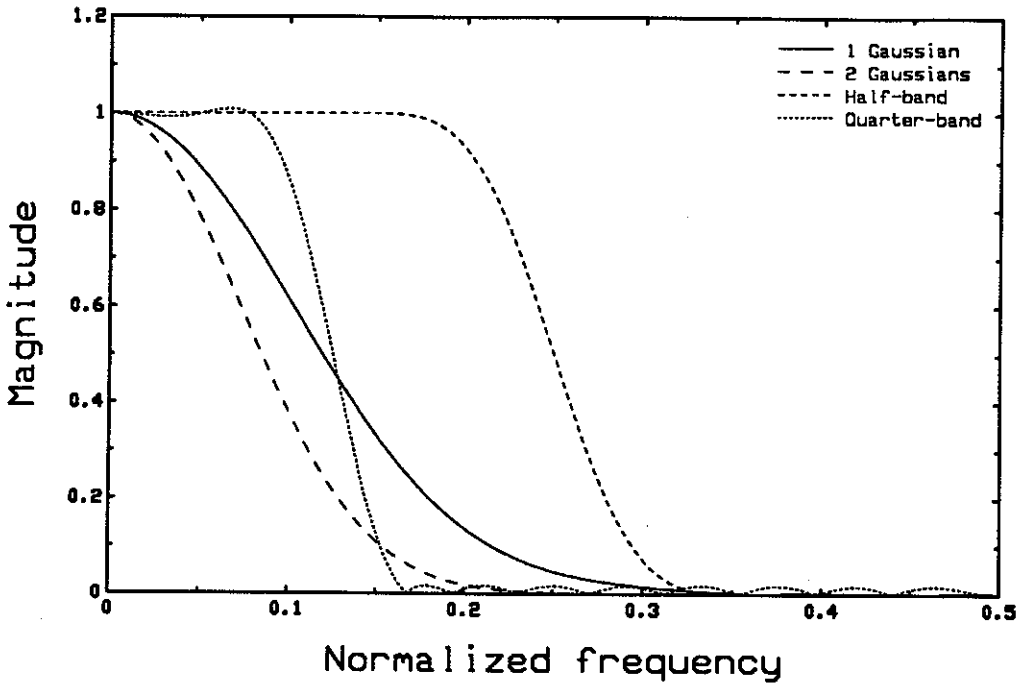


Fig. 5.4 Magnitude of the frequency response of Gaussian and Nyquist-like low-pass filters for 3-level (K_l=3) hierarchical motion estimation.

for maximum signal fidelity and immunity to aliasing, I chose to design the filters with at most 25% power aliased into the baseband (50% roll-off at half- and quarter-Nyquist rate respectively).

The benefits of data filtering for hierarchical estimation are counteracted by "distortions" introduced in the occlusion and newly exposed areas. Recall that the motion model does not distinguish such areas, and that it relies on intensity constancy along motion trajectories. Consider an image with an object moving across a uniform background and suddenly disclosing a structural discontinuity in this background. The filtering operation applied to the images is shift-invariant and will produce different intensity patterns in both images in the vicinity of the object border due to different background types. In spite of a possibly perfect intensity match at full resolution (except for the exposed and occluded areas), it will not be the case for higher levels in the pyramid, and the assumption of constant intensity along motion trajectories will not be satisfied. The extent of this violation will depend on the severity of structural discontinuity in the uncovered and occluded background, and on filter characteristics (coefficient values, mask size). Since the principle assumption

of algorithms presented here will not be satisfied in such case, serious difficulties at lower resolution levels can be expected. As will be demonstrated in Section 5.5, some images are more prone to such problems, like the moving random uncorrelated dots, while in typical TV imagery such effects are less pronounced.

5.4 DATA-MODEL COMPROMISE ACROSS THE RESOLUTIONS

In this section the problem of parameter ratio $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$ adjustment with changing data resolution will be discussed.

The need for such an adjustment can be explained informally as follows. The ratio $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$ reflects the amount of smoothness to be expected from the estimator. In the hierarchical approach, setting $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$ to a fixed value across the range of resolutions seems to be incorrect, since at lower resolution levels (top of the pyramid) the data is smoother and as such will produce smoother estimates †. Since some inherent vector smoothing is due to the smooth data, the ratio $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$ should be adjusted at lower resolution levels so that oversmoothed estimates would not be produced.

Recall that the displaced pel differences $\tilde{r}$ are modeled by *iid* Gaussian random variables. For such a strictly stationary (and consequently wide-sense stationary) discrete-space stochastic process, the power spectral density $\mathcal{S}(\omega_1,\omega_2)$ can be defined as follows

$$\mathcal{S}(\omega_1,\omega_2) = \sum_{m=-M}^{M}\sum_{n=-M}^{M} \mathcal{R}(m,n)\cdot e^{-j(\omega_1 m+\omega_2 n)}, \qquad -\pi \le \omega_1,\omega_2 \le \pi,$$

where $\{\mathcal{R}(m,n)\}$, $m,n = 0,1,...,M$ is an autocorrelation sequence, and (ω_1,ω_2) is the 2-D angular frequency. If the lower resolution data is obtained using a filter with frequency response $H(\omega_1,\omega_2)$, then the power spectral density $\mathcal{S}_f(\omega_1,\omega_2)$ of the filtered data can be computed from the following relationship

$$\mathcal{S}_f(\omega_1,\omega_2) = \mathcal{S}(\omega_1,\omega_2)\cdot|H(\omega_1,\omega_2)|^2.$$

Consequently the variance σ_f^2 of the filtered data is defined as:

$$\sigma_f^2 = \mathcal{R}(0,0) = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\mathcal{S}(\omega_1,\omega_2)\cdot|H(\omega_1,\omega_2)|^2 d\omega_1 d\omega_2,$$

† The fact that a smoother image will result in a smoother motion estimate can be explained by looking at high correlation of intensities in such an image, and hence high correlation of the gradients and displaced pel differences. In a spatio-temporal gradient estimation a highly correlated gradient will not produce uncorrelated motion vectors. The same applies to matching algorithms.

but since for *iid* random variables the power spectral density is flat ($\mathcal{S}(\omega_1,\omega_2)=\sigma^2$)

$$\sigma_f^2 = \frac{1}{4\pi^2}\cdot\sigma^2\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}|H(\omega_1,\omega_2)|^2 d\omega_1 d\omega_2. \qquad (5.5)$$

The above relationship establishes the dependence of displaced pel difference variance on filter characteristics. The narrower the filter bandwidth the smaller the variance σ_f^2, and consequently the smaller the ratio $\lambda_d^\kappa/\lambda_g^\kappa$.

The relationship (5.5) evaluated for the filters from Fig. 5.4 results in the following values of σ_f^2:

1. one Gaussian filter: σ_f^2=0.034·σ^2,
2. two Gaussian filters: σ_f^2=0.017·σ^2,
3. half-band Nyquist-like filter: σ_f^2=0.216·σ^2,
4. quarter-band Nyquist-like filter: σ_f^2=0.052·σ^2.

According to the above values theoretically the variance of DPDs for filtered images is approximately reduced by 30 and 60 for one and two Gaussian filters respectively, and by 5 and 20 for half- and quarter-band Nyquist-like filters. In reality the filtered DPD variance σ_f^2 may significantly depart from theoretical values because:

1. the DPDs are not exactly independent, hence their power spectral density $\mathcal{S}(\omega_1,\omega_2)$ is not constant in the $[-\pi,\pi]$ range,
2. since the filtering operation is shift invariant it may destroy the match around motion boundaries especially in presence of covered or occluded structural discontinuities,
3. severity of the distortion described above is increased at lower resolution levels where the discontinuities are closer to each other.

To estimate the severity of mismatch between theoretical and practical reduction of variance, I have applied estimates from Figs. 4.16.a, 4.17.a to the original and filtered images respectively. For single Gaussian filtering the variance went down by about 2.0, while for half-band Nyquist-like filtering it dropped by about 1.5. Interestingly, after another Gaussian filtering and after quarter-band Nyquist-like filtering the variance remained almost unchanged. This can probably be explained by the fact that no other effects than additive noise have been incorporated into the observation model and consequently into the DPD model. Those other effects reported in Section 3.4.2, however, are present in the test

images 2, 3, 4, and contribute to pel matching errors. In practice the variance reduction may be much smaller, and much smaller modifications of smoothness weights should be applied.

The ratio $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$ also involves the constant $\beta_{\mathbf{d}}$ from the Gibbs distribution $\pi(\mathbf{d}_t)$ which characterizes the properties of displacement field $\mathbf{d}_t$. It is not easy to establish the relationship between $\beta_{\mathbf{d}}$'s for different resolutions in a general case. Gidas [29] using the renormalization group approach showed under certain assumptions the relationship between Ising model parameters at different resolutions (scales). For isotropic 2-D Ising model, an equivalent of $\beta_{\mathbf{d}}$ increased by 1.21 between each two resolution levels (spatial subsampling by 2). That result cannot be directly extrapolated to compute $\beta_{\mathbf{d}}$ here, however it indicates the order of change to be expected from $\beta_{\mathbf{d}}$. Moreover, since the theoretical reduction of DPD variance due to filtering is exaggerated, it must be established *ad hoc* anyway. Hence even a precise modification of $\beta_{\mathbf{d}}$ would leave the overall ratio *ad hoc*.

Since at the top of the pyramid the algorithm starts with no information about motion, while moving downwards it uses the previous-level estimates, it is reasonable to require that the temperature **T** in simulated annealing be reduced too. In this way deeper local minima can be avoided when there is less confidence in the estimate. Such an approach has been used in image restoration over a hierarchy of scales via stochastic relaxation methods [62], [29].

5.5 EXPERIMENTAL RESULTS

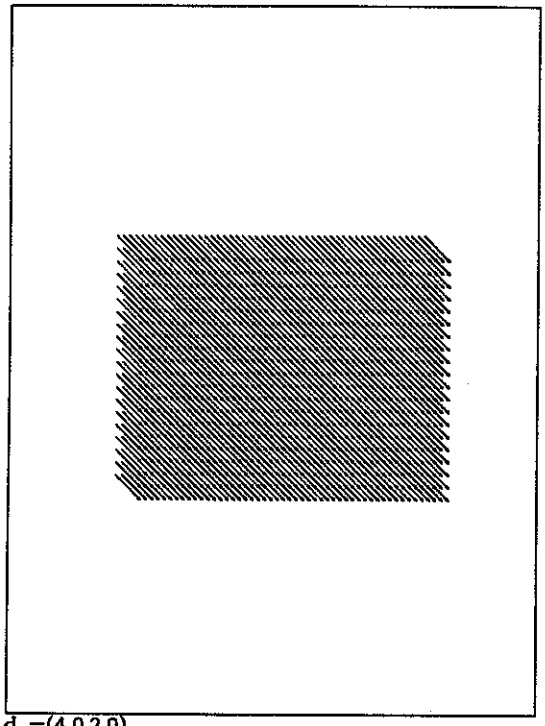
In this section some experimental results for hierarchical MAP estimation will be presented. The test images from Section 4.7 will be used with temporal spacing between images g_{t_-} and g_{t_+} of T_g=4τ_{60} for test images 1, 3, 4 or T_g=6τ_{60} for test image 2. In describing the results for hierarchical algorithms some parameters like ratios $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$, initial temperatures $\mathbf{T}_0^{\kappa}$ etc., varying with the resolution level, will be presented as vectors. For example $\mathbf{T}_0^{\kappa}$ = (1.0,2.0,4.0) will mean that the initial temperatures at levels 0, 1, 2 are 1.0, 2.0, 4.0 respectively. Taking into account the measured variances of filtered data and the 1.21 factor between $\beta_{\mathbf{d}}$'s at neighbouring resolution levels the $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$ ratios of

(20.0,9.0,7.0) will be used for Gaussian filters and (20.0,12.0,10.0) will be applied in the case of Nyquist-like filters.

5.5.1 Results for test image 1

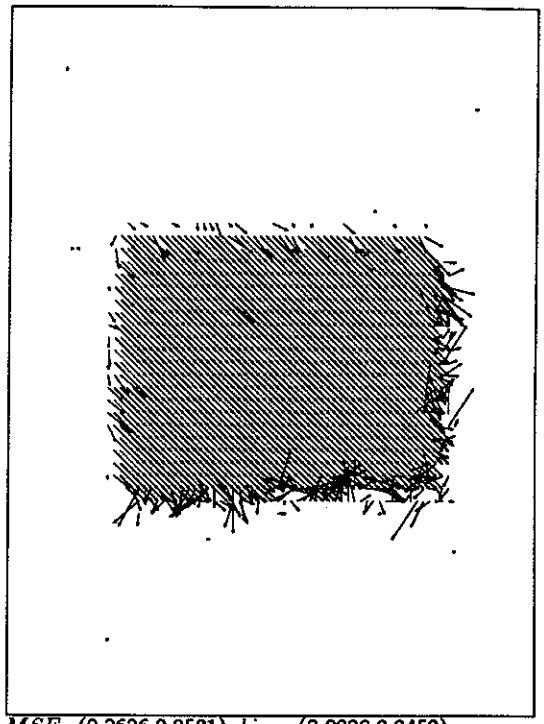
As was explained in Section 5.3, certain classes of images are not well suited for hierarchical motion estimation. The test image 1 turns out to belong to such a class. Since it consists of uncorrelated random dots, the newly exposed and occluded background areas are in no structural relationship with the rest of the background. Hence, the filtering operation will modify the intensity pattern around object borders differently for both images. Even if at full resolution there is a perfect matching of pels belonging to an object, after filtering this will not happen in the band around the border affected by the background pels. For the test image 1 the background consists of uncorrelated pels hence poor matching can be expected in some areas.

Fig. 5.5.a shows the synthetic displacement to be estimated. In Fig. 5.5.b the discrete state-space MAP estimate implemented at one resolution scale is presented, while Figs. 5.5.c,.d show the results of hierarchical MAP estimation over two resolution levels (K_l=2). In both cases the $\mathcal{N}_{\mathrm{d}}^1$ neighbourhood, bilinear interpolation, exponential annealing schedule and maximum displacement d_{max}^0=4.0 (N_d=33 for δ^0=0.25) were used. The single-level estimate was obtained for $\lambda_{\mathrm{d}}^0/\lambda_g^0$=0.05 with $\mathbf{T}_0$=1.0 and a=0.980 over 200 iterations. The hierarchical approach used the Gaussian filters and $\lambda_{\mathrm{d}}^\kappa/\lambda_g^\kappa$ = (0.05,10.0) to offset the introduction of mismatch due to filtering as discussed above. To avoid local minima due to the filtering, exponential annealing schedule with $\mathbf{T}_0^\kappa$=(1.0,10.0) and a=(0.9625,0.980) over (50,200) iterations was used. The single-level method attained very similar solution in 200 iterations with about triple computational effort of the 2-level method. This relatively small computational gain of the 2-level approach is due to the fact that significant estimation errors are expected at level κ=1, and in order to recover from those errors, a large state-space (N_d=33) must be used at full resolution. The gain is due to faster convergence at full resolution assured by knowledge of a coarse estimate from level 1. In more typical situations, where occlusion and newly exposed area effects are



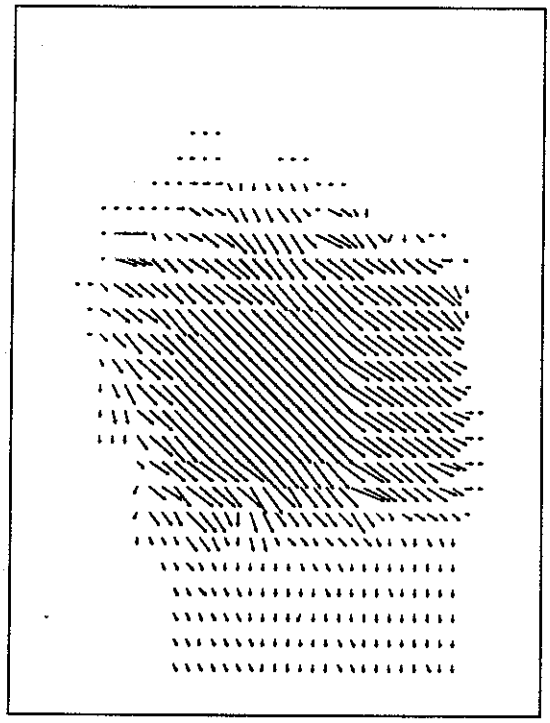
d_*=(4.0,2.0)

(a) synthetic displacement

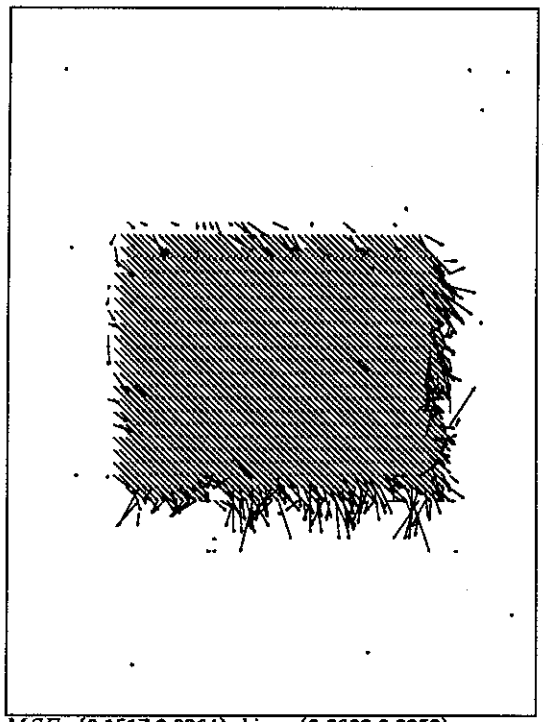


MSE=(0.2626,0.0581), $bias$=(0.0920,0.0450)

(b) single-level



(c) hierarchical, κ=1



MSE=(0.1517,0.0264), $bias$=(0.0693,0.0253)

(d) hierarchical, κ=0

Fig. 5.5 Hierarchical discrete state-space MAP estimates: test image 1, K_l=2, $\lambda_{\mathbf{d}}^{\kappa}/\lambda_g^{\kappa}$= (0.05,10.0), neighb. $\mathcal{N}_{\mathbf{d}}^1$, bilinear interp., exponential schedule, $\mathbf{T}_0^{\kappa}$= (1.0,10.0), 200 iter. for single-grid and (50,200) iter. for hierarchical approach.

less pronounced, smaller state-spaces can be used without degradation of estimate quality resulting in higher computational gain.

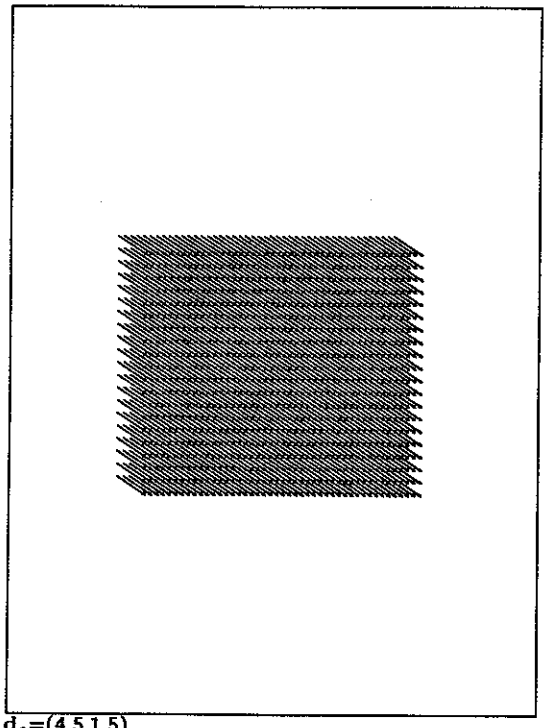
Application of the Nyquist-like filter caused even stronger effects. In fact the second-level estimate was far from the estimate from Fig. 5.5.c, which can be explained by the size of filter mask. Recall that such a filter extended over 25 pels by 25 pels area. Since the size of the moving rectangle is 50 by 20 pels, there is no single pel within the rectangle not affected by the background exposure or coverage. Consequently no perfect match exists in the filtered image. Note that the Gaussian filtered image included perfect match areas since the mask size was only 9 by 9.

5.5.2 Results for test image 2

Fig. 5.6.a shows the synthetic displacement field from test sequence 2 to be recovered, while Figs. 5.6.b,.c,.d show the discrete state-space MAP estimates at levels κ=2,1,0 respectively. Nyquist-like filters with $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$= (20.0,12.0,10.0), as well as neighbourhood $\mathcal{N}_{\mathbf{d}}^{1}$, Keys bicubic interpolation and exponential annealing schedule with $\mathbf{T}_0^{\kappa}$= (1.0,2.0,4.0) and a=0.980 over 200 iterations at each level were used. Since the mismatch due to filtering in the newly exposed and occlusion areas is much less severe than in the test image 1, smaller state-spaces with $N_d = 9$ and δ^0=0.25 were used at each level resulting in further computational gain. Fig.5.7.a shows the single-level estimate obtained for the same set of parameters except for the state-space which extended over d_{max}=5 with N_d=41 and the annealing schedule with $\mathbf{T}_0$=10.0 † and a=0.980 over 300 iterations. The hierarchical estimate is characterized by a higher energy, and is a little oversmoothed at the rectangle boundaries compared to the single-level estimate. Most importantly, however, it was incapable of correct estimation in the left-top corner of the rectangle, unlike the single-level method. In that area a bright background is uncovered in the second image and causes image breakup. The single-level method with sufficiently high $\mathbf{T}_0$ is able to estimate the motion correctly, however the hierarchical method cannot because of the filtering.

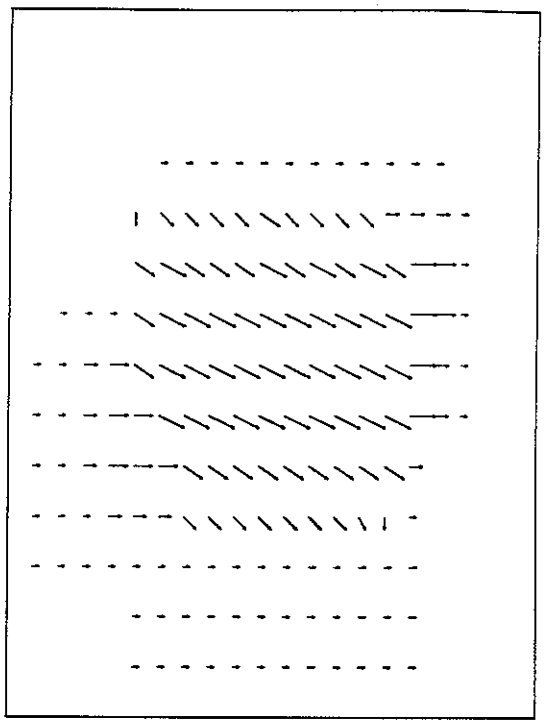
For the above set of parameters the computational gain of the hierarchical method with respect to the single-level approach was 15.6. Note, however, that the total maximum

† For $\mathbf{T}_0$=1.0 over 200 iterations the single-level method attained a sub-optimal solution.

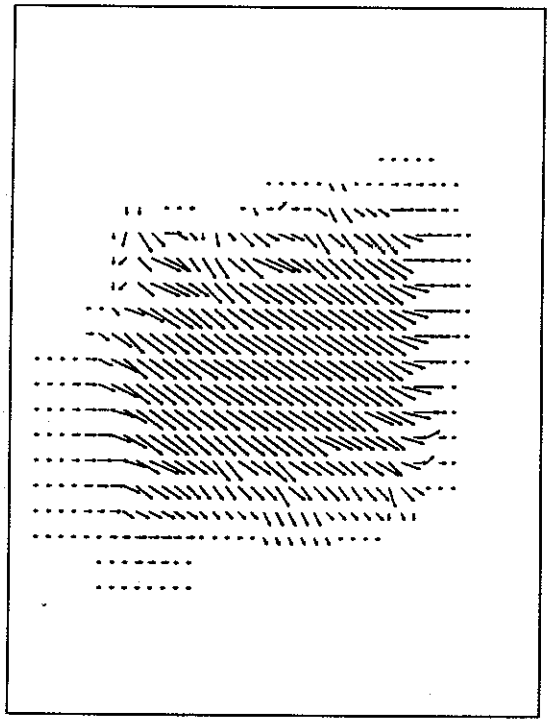


$\mathbf{d}_*$=(4.5,1.5)

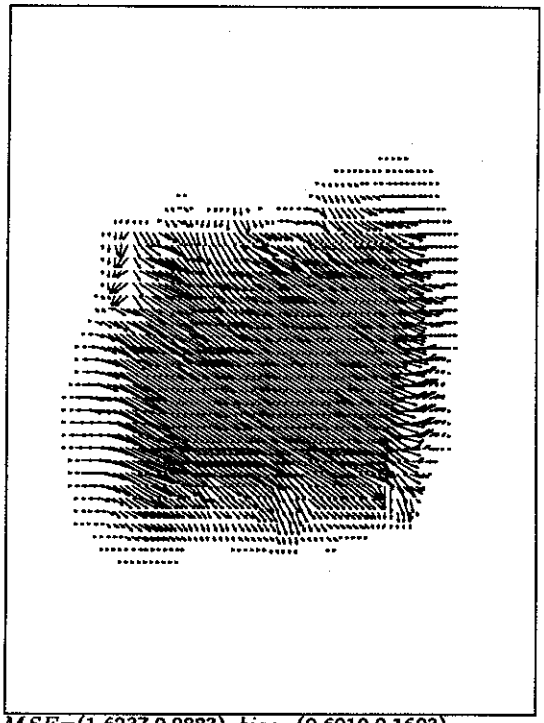
(a) synthetic displacement



(b) hierarchical, κ=2



(c) hierarchical, κ=1



MSE=(1.6237,0.0883), *bias*=(0.6919,0.1603)

(d) hierarchical, κ=0

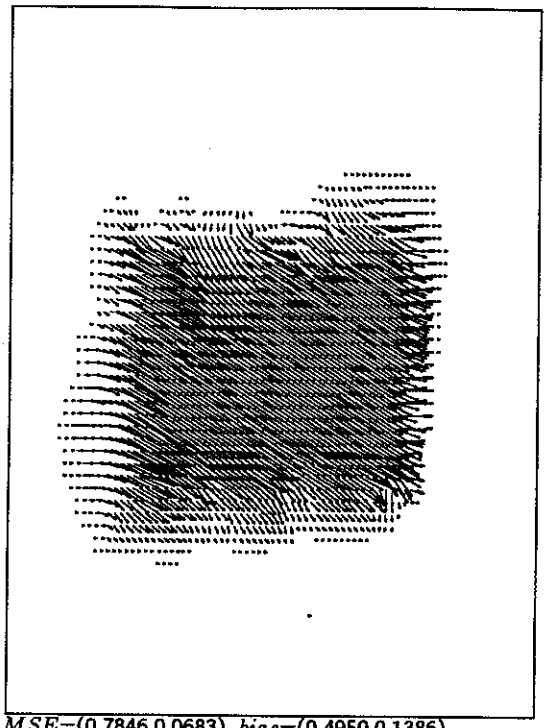
Fig. 5.6 Hierarchical discrete state-space MAP estimates: test image 2, K_l=3, $\lambda_{\mathbf{d}}^{\kappa}/\lambda_{g}^{\kappa}$= (20.0,12.0,10.0), neighb. $\mathcal{N}_{\mathbf{d}}^{1}$, Keys bicubic interp., exponential schedule, $\mathbf{T}^{\kappa}$=(1.0,2.0,4.0) and 200 iter. at each level.

displacement for the hierarchical method was 7 pels, while for the single-level method it was only 5 pels. If the maximum displacement d_{max} were set to 7 pels for the single-level approach, then the computational gain would be about 30.

Also the Gaussian filters from Fig. 5.4 have been used with $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$=(20.0,9.0,7.0) and other parameters unchanged. The estimates resembled subjectively the results from Fig. 5.6 very closely at each resolution level. The displacement energy U_{d} and the image energy U_g at full resolution level were higher, however, than for the estimates obtained with the Nyquist-like filters †. For both filter types a limited variation of ratios $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$ as well as different annealing schedules had negligible effect on the final result. Obviously the varying ratio $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$ affected the estimates at κ >0, but then the algorithm was able to recover from any sub-optimalities at subsequent higher resolution levels. This confirms that, unlike in the test image 1, the occlusion and exposure effects are not extremely pronounced in typical TV imagery. Also, the bilinear interpolation, instead of repetition, of displacement estimates between consecutive resolution levels has been used but had almost no impact at all on the final-level estimate.

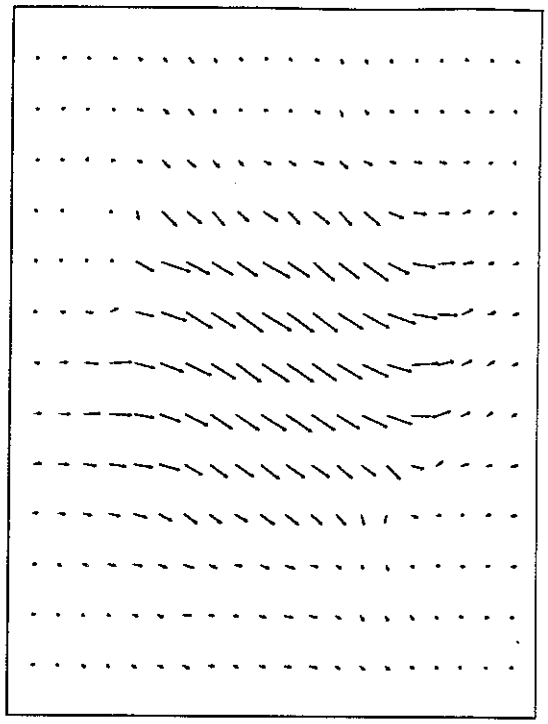
Figs. 5.7.b,.c,.d show the continuous state-space MAP estimates for levels κ=2,1,0 obtained with the same parameters as the results from Fig. 5.6 except for a=0.992 and 500 iterations. Also in this case varying weight ratios and annealing schedules had minimal effect on the final level estimates. First of all note that there are numerous small vectors in the stationary background (perfect match), while there were very few in the discrete state-space case. This effect can be explained by the inaccuracies in intensity derivatives computations in the continuous case versus a simple intensity match in the discrete case. Apart from the background effect note that for κ=1,2 the estimates are smoother in the rectangle-background transition region than in the discrete case, due to local averaging used to compute a new estimate. There is no explicit averaging in the case of the discrete state-space Gibbs sampler. The energies for the continuous case were significantly lower than for the discrete case at κ=2, somewhat lower at κ=1 and slightly higher at κ=0. The full resolution results slightly differ subjectively as well.

† The energies for $\kappa \neq 0$ cannot be compared since λ^{κ}'s are different.

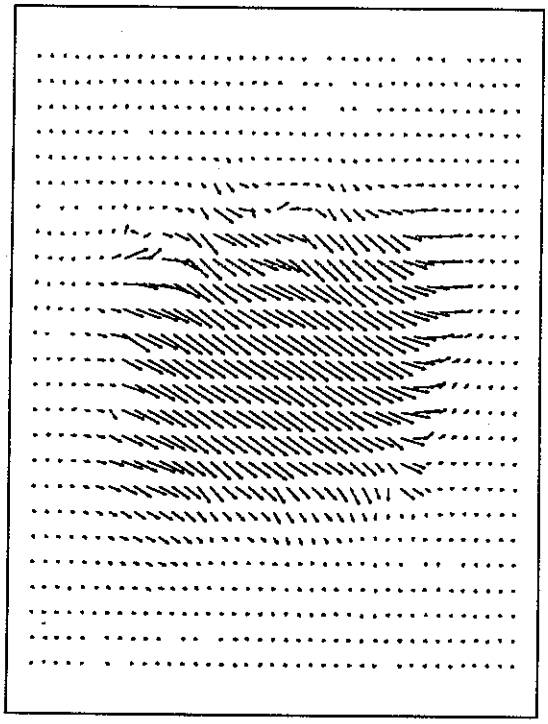


MSE=(0.7846,0.0683), $bias$=(0.4950,0.1386)

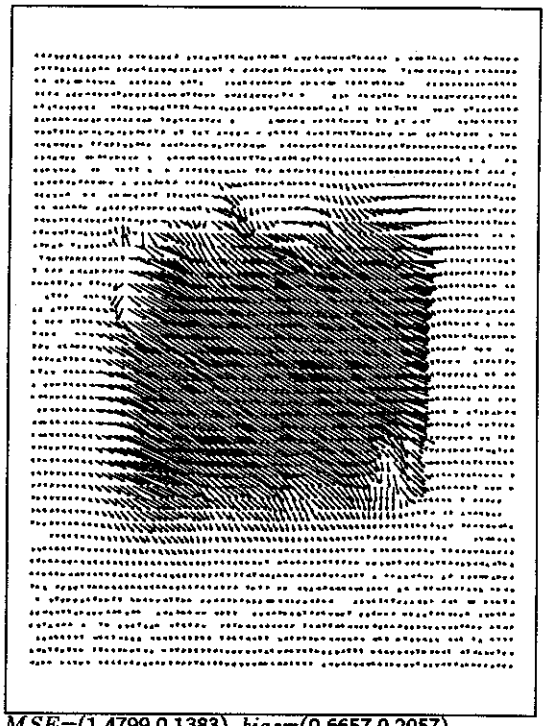
(a) single-level, discrete



(b) hierarchical, κ=2



(b) hierarchical, κ=1



MSE=(1.4799,0.1383), $bias$=(0.6657,0.2057)

(d) hierarchical, κ=0

Fig. 5.7 Single-level discrete and hierarchical continuous state-space MAP estimates: test image 2, K_l=3, $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$=(20.0,12.0,10.0), neighb. $\mathcal{N}_{\mathrm{d}}^{1}$, Keys bicubic interp., exponential schedule, $\mathbf{T}^{\kappa}$=(1.0,2.0,4.0), 500 iter. at each level.

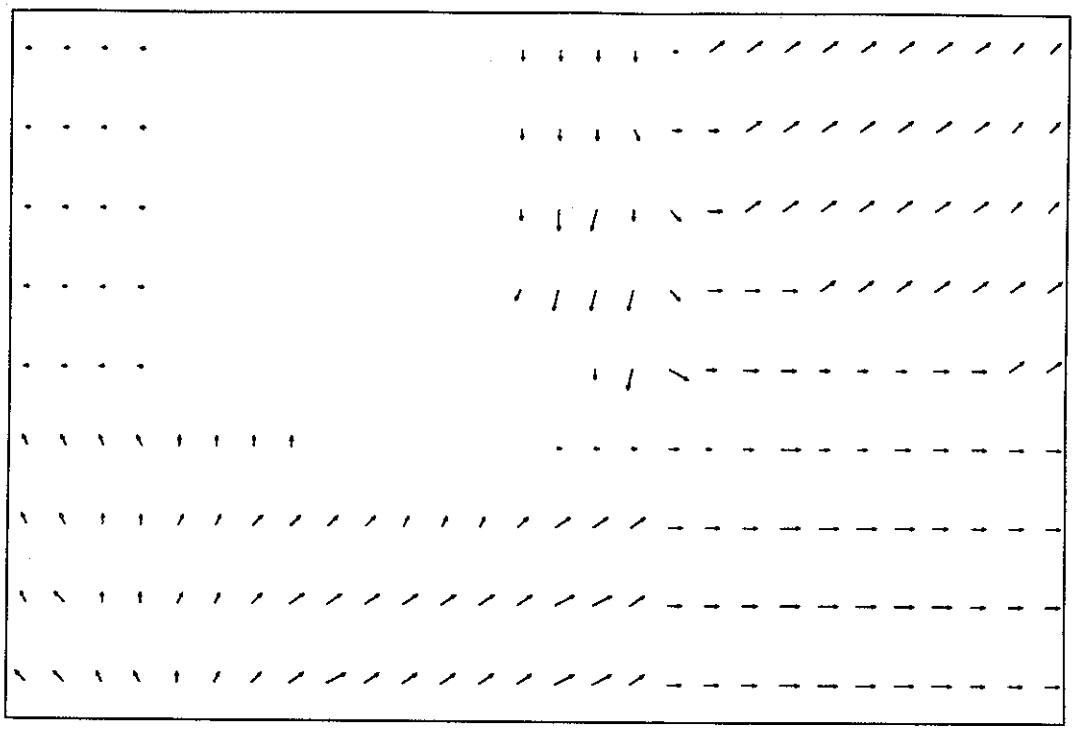
5.5.3 Results for test images 3 and 4

Figs. 5.8, 5.9, 5.10 show the level 2, 1, 0 motion estimates respectively, obtained from the test image 3 using the 3-level discrete and continuous state-space MAP estimation with Keys bicubic interpolator, Nyquist-like filters and $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$= (20.0,12.0,10.0) ratios. In both cases exponential annealing schedule with T^{κ}= (1.0,2.0,4.0) was used, however with a=0.980 over 200 iterations in the discrete estimation, and with a=0.992 over 500 iterations in the continuous estimation. Again limited variations of the ratios $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$ as well as of the annealing schedule and inter-level interpolation had small effect on the final results, however the constant ratio $\lambda_{\mathrm{d}}^{\kappa}/\lambda_{g}^{\kappa}$ =(20.0,20.0,20.0) resulted in significantly poorer motion estimate both subjectively and in terms of the parametrizing energies. As before, the estimate obtained with the Nyquist-like filter resulted in significantly lower energy values than with the Gaussian filter, however subjectively the estimates were quite similar.

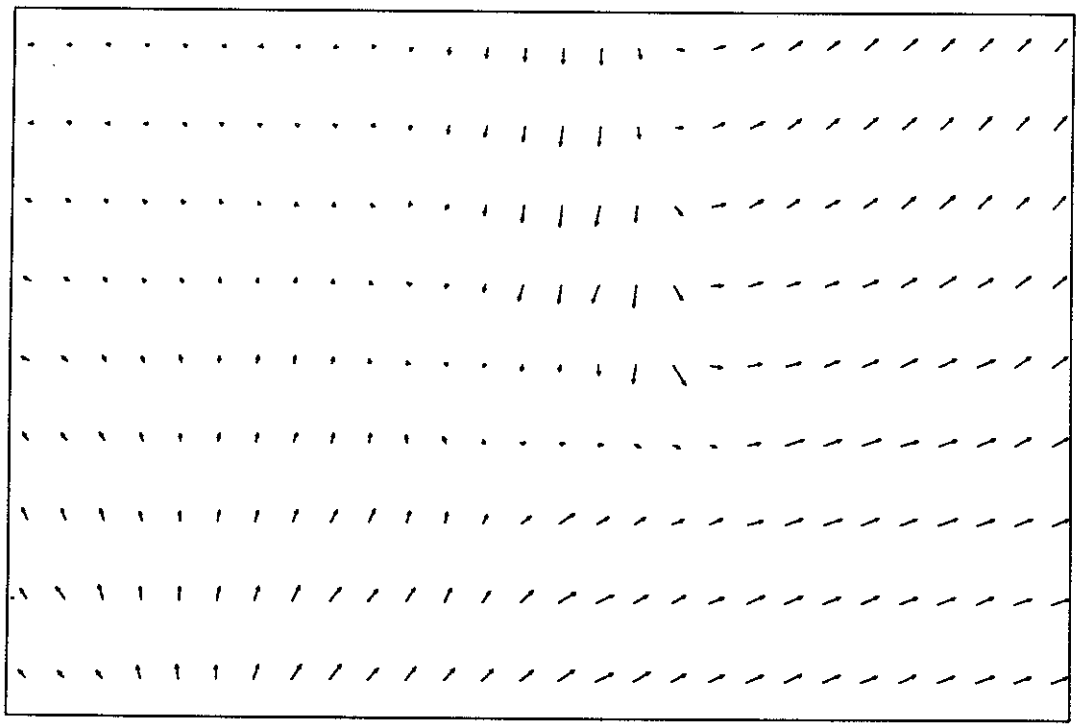
Note that both discrete and continuous state-space estimates are very similar at the full resolution level in spite of some differences at the lower resolution levels, especially for κ=2 when the continuous state-space estimate was significantly smoother. Observe also the evolution of motion field structure with increasing resolution. At level κ=2 only coarse motion is computed, while at level κ=1 the motion boundaries are already quite well established. At full resolution more details, like the boundaries of the hand and of the face show up, however there are still significant smoothing effects like above the hand or in the occlusion between the hand and the face. As far as the energies are concerned the continuous state-space estimate attained lower values due to a longer annealing schedule.

As shown at the end of Section 5.2.1 the computational gain of the hierarchical approach over the single-level method is around 30. Of course if the maximum displacement in the image is less than $\sum_{\kappa=0}^{K_l-1} d_{max}^{\kappa}$=7, then a smaller size state-space can be used, however even for the worst case 3-level estimation (displacement of 3.25 pels) the gain is still around 7. The motion estimate for test image 3 obtained with the single-level method has been almost identical to that from Fig. 5.10.a [55] confirming the ability of the single-level discrete state-space MAP estimation to compute also large displacements.

Figs. 5.11, 5.12, 5.13 show the level 2, 1, 0 motion estimates obtained from the test

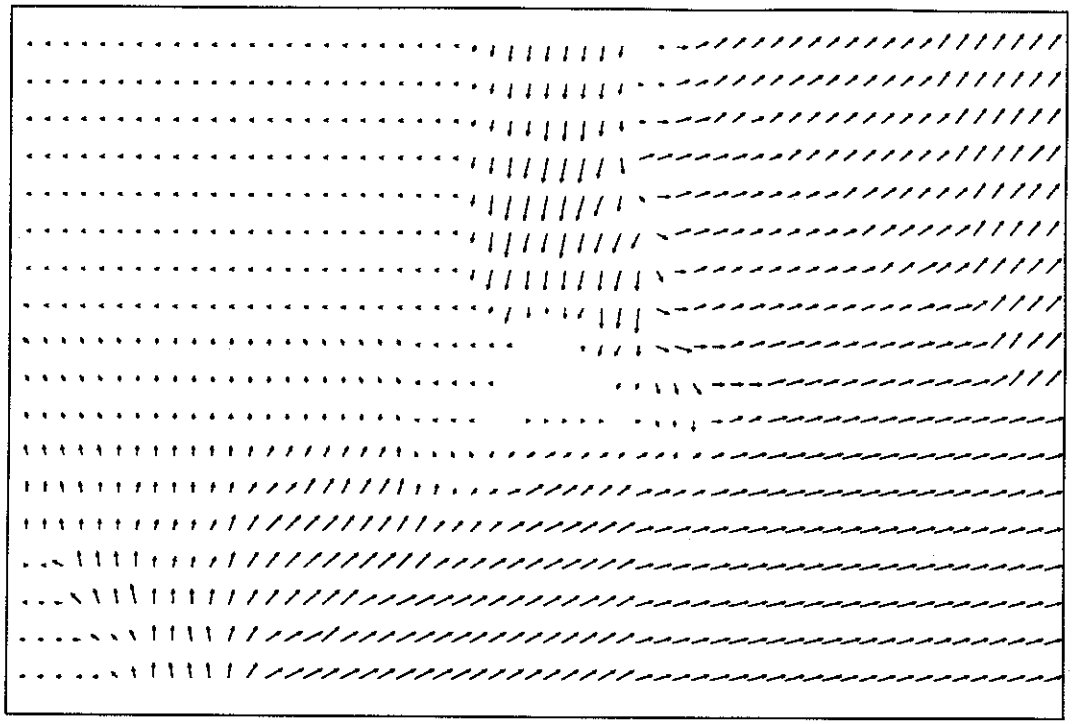


(a) discrete state-space, κ=2

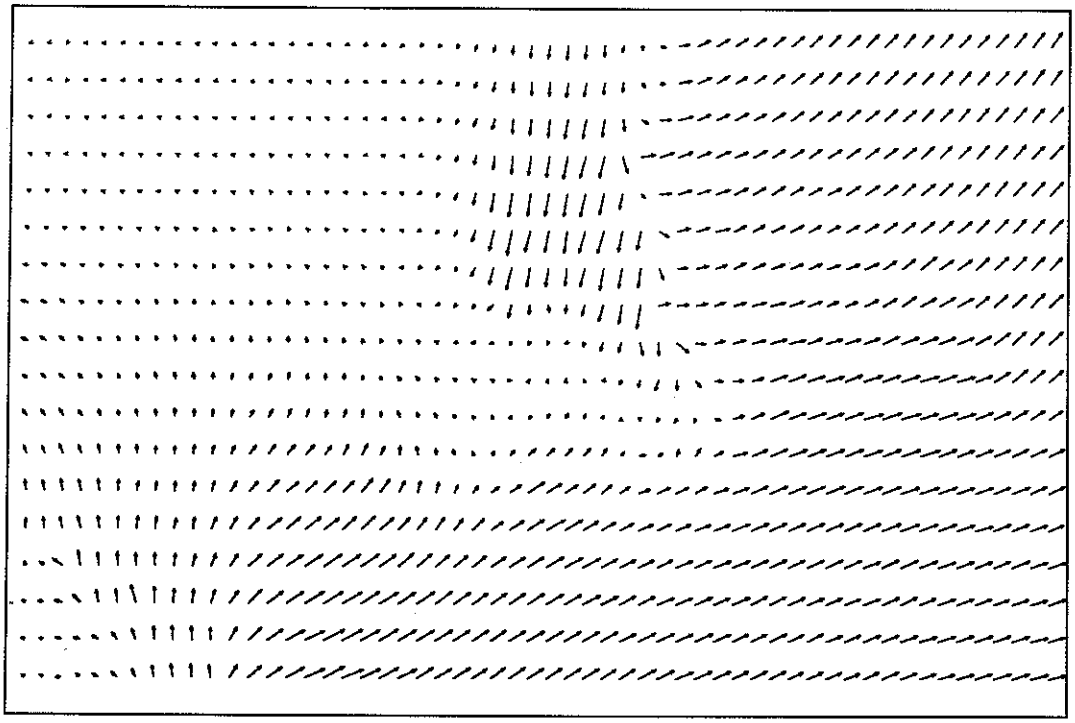


(b) continuous state-space, κ=2

Fig. 5.8 Hierarchical discrete and continuous state-space MAP estimates: test image 3, K_l=3, κ=2, $\lambda_{\mathbf{d}}/\lambda_g$= (20.0,12.0,10.0), neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}^{\kappa}$=(1.0,2.0,4.0), 200 (a) or 500 (b) iter.

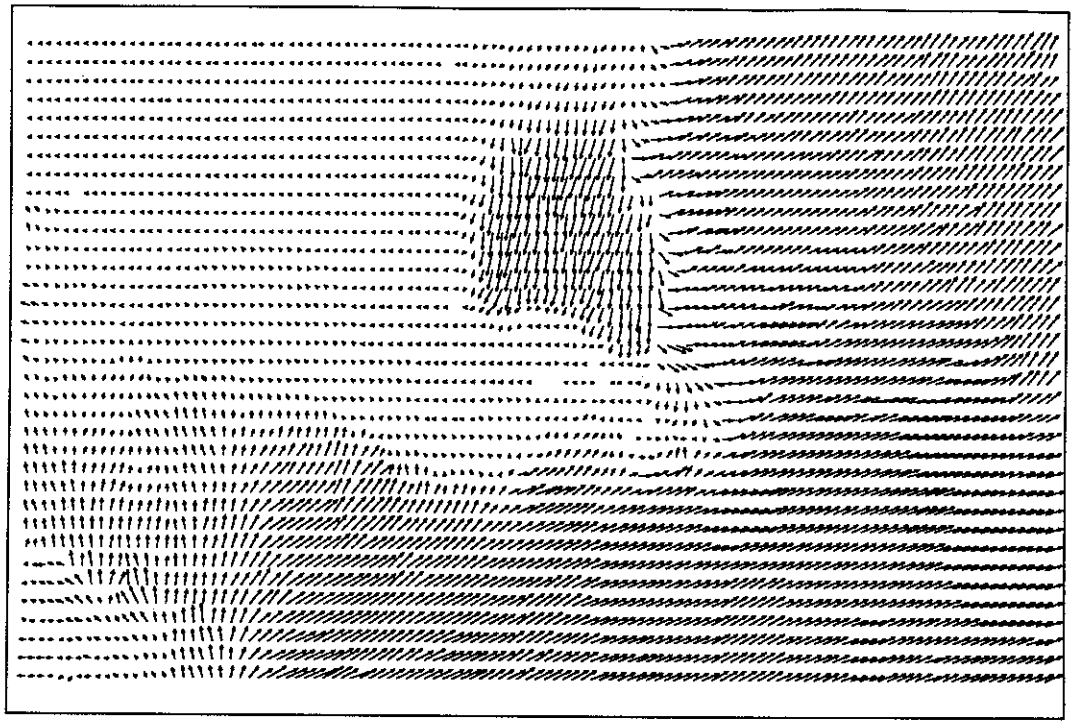


(a) discrete state-space, κ=1

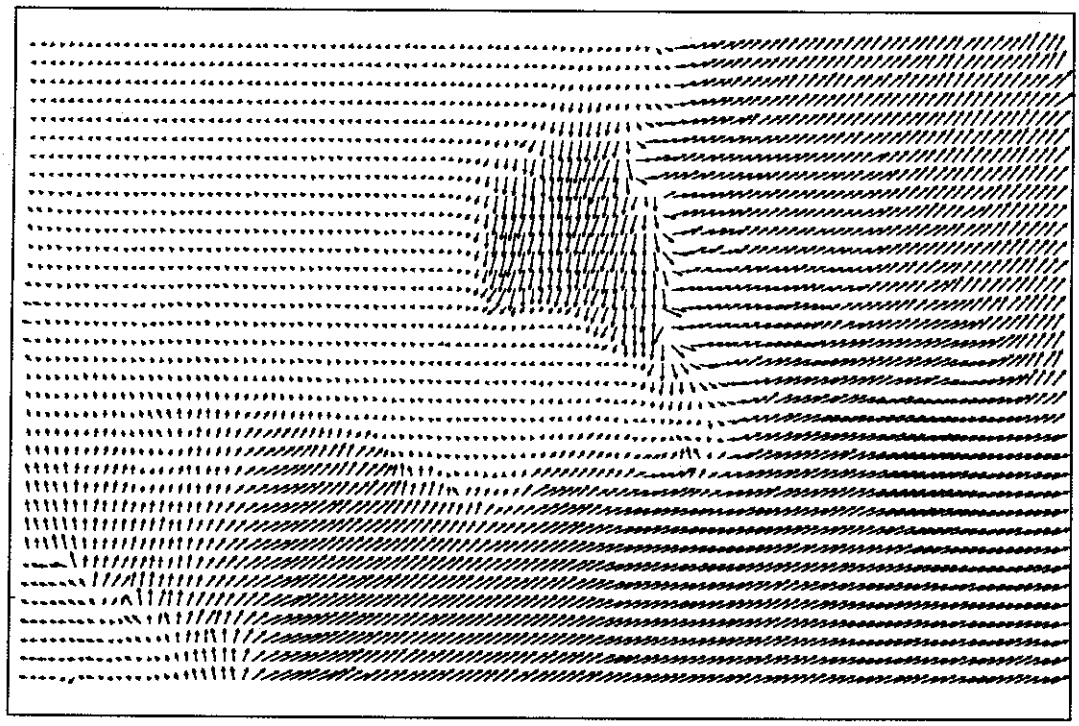


(b) continuous state-space, κ=1

Fig. 5.9 Hierarchical discrete and continuous state-space MAP estimates: test image 3, K_l=3, κ=1, $\lambda_{\mathrm{d}}/\lambda_g$= (20.0,12.0,10.0), neighb. $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}^{\kappa}$=(1.0,2.0,4.0), 200 (a) or 500 (b) iter.



(a) discrete state-space, κ=0

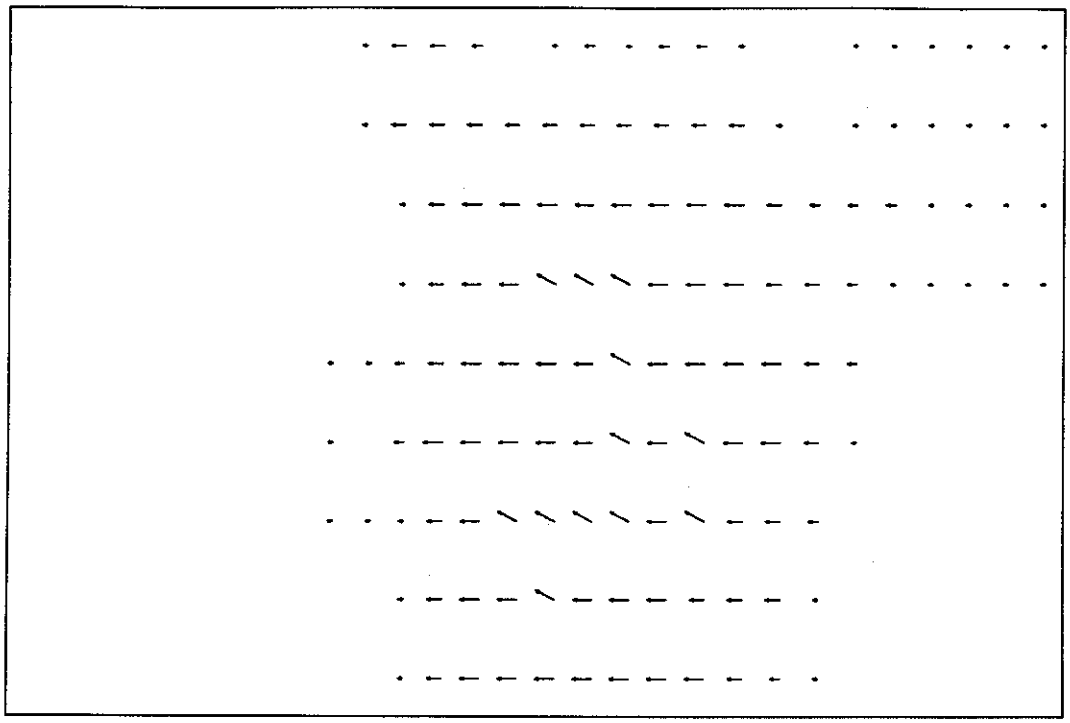


(b) continuous state-space, κ=0

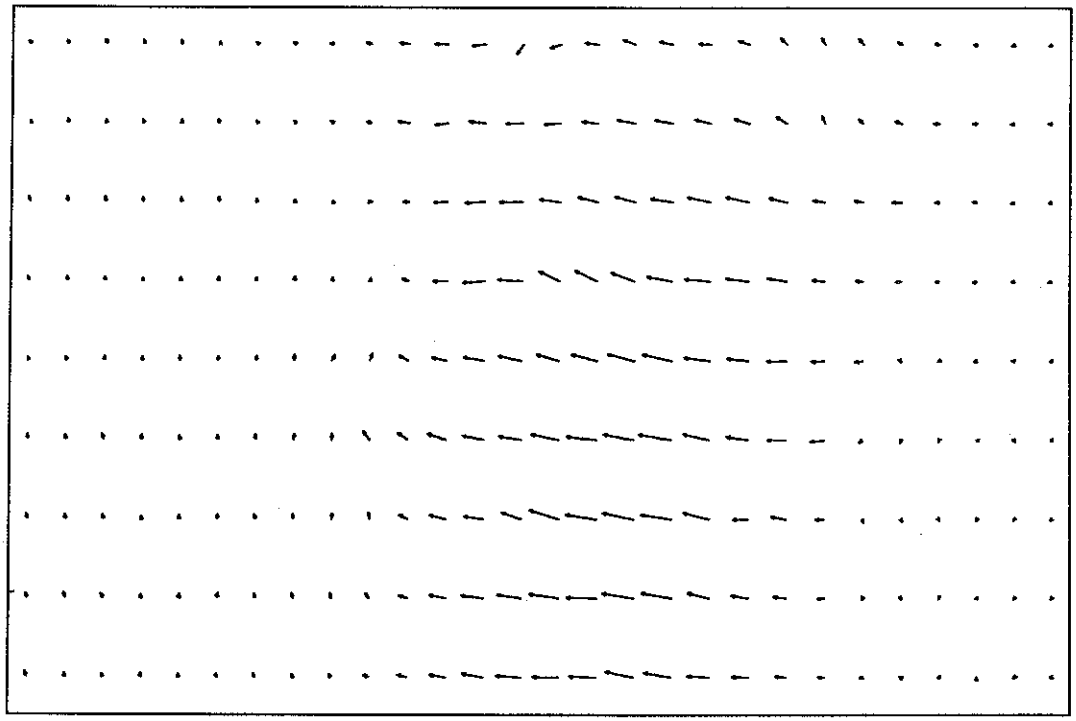
Fig. 5.10 Hierarchical discrete and continuous state-space MAP estimates: test image 3, K_l=3, κ=0, λ_d/λ_g= (20.0,12.0,10.0), neighb. $\mathcal{N}_d^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}^\kappa$=(1.0,2.0,4.0), 200 (a) or 500 (b) iter.

image 4 using the same parameters as for the test image 3. The same relative insensitivity to the weight and annealing schedule adjustment has been observed as for the test image 3.

Again both discrete and continuous state-space estimates are quite similar at the full resolution level in spite of some differences at the lower resolution levels. Also here the evolution of motion field structure with increasing resolution can be observed, however the motion boundaries are not as well defined as for the test image 3. This is due to mostly rotational motion of the head unlike the translational motion of the arm, hand and face in the test image 3. The continuous state-space estimate attained lower energy than the discrete state-space estimate, but as before this is due to a larger number of iterations.

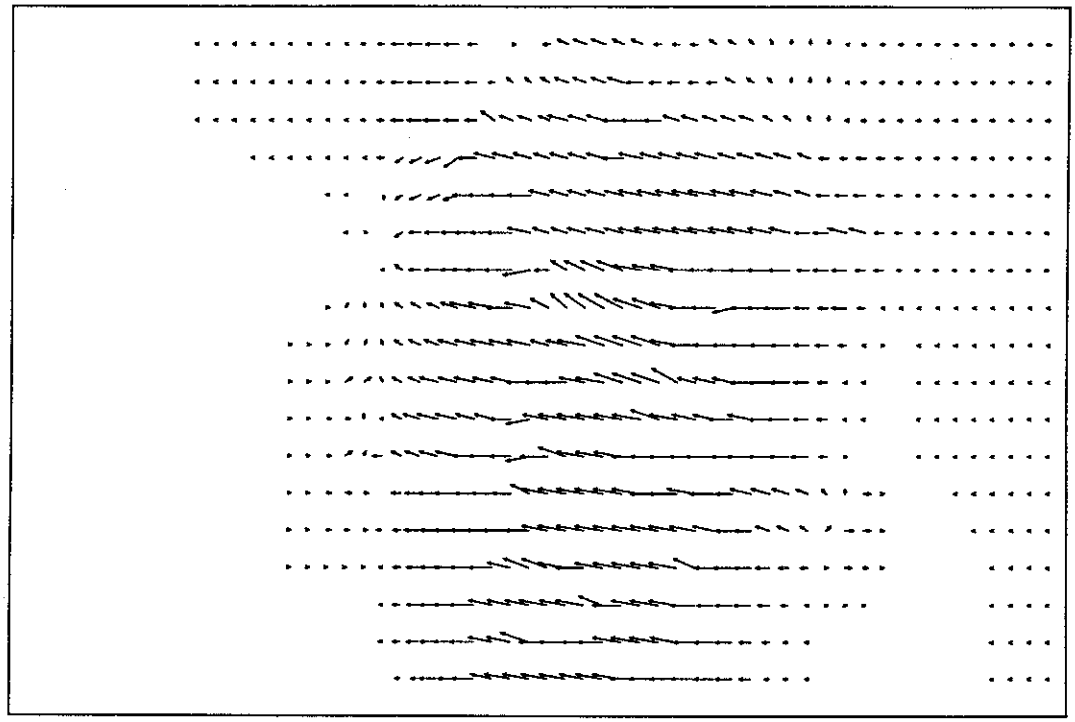


(a) discrete state-space, κ=2

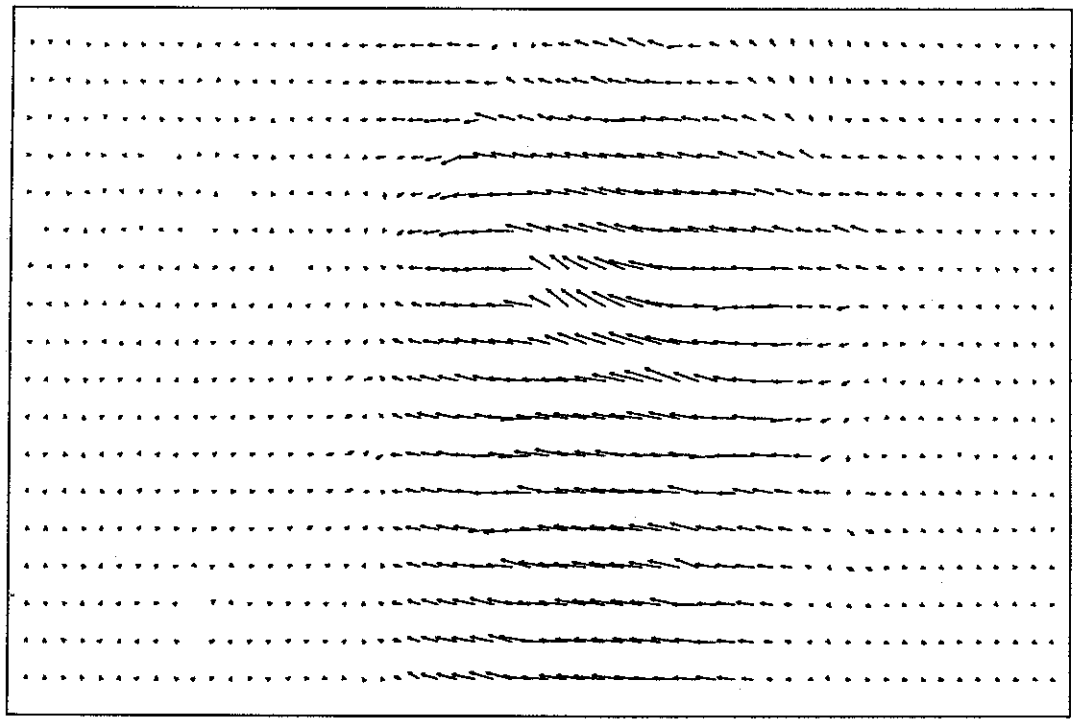


(b) continuous state-space, κ=2

Fig. 5.11 Hierarchical discrete and continuous state-space MAP estimates: test image 4, K_l=3, κ=2, $\lambda_{\mathrm{d}}/\lambda_g$= (20.0,12.0,10.0), neighb. $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}^{\kappa}$=(1.0,2.0,4.0), 200 (a) or 500 (b) iter.

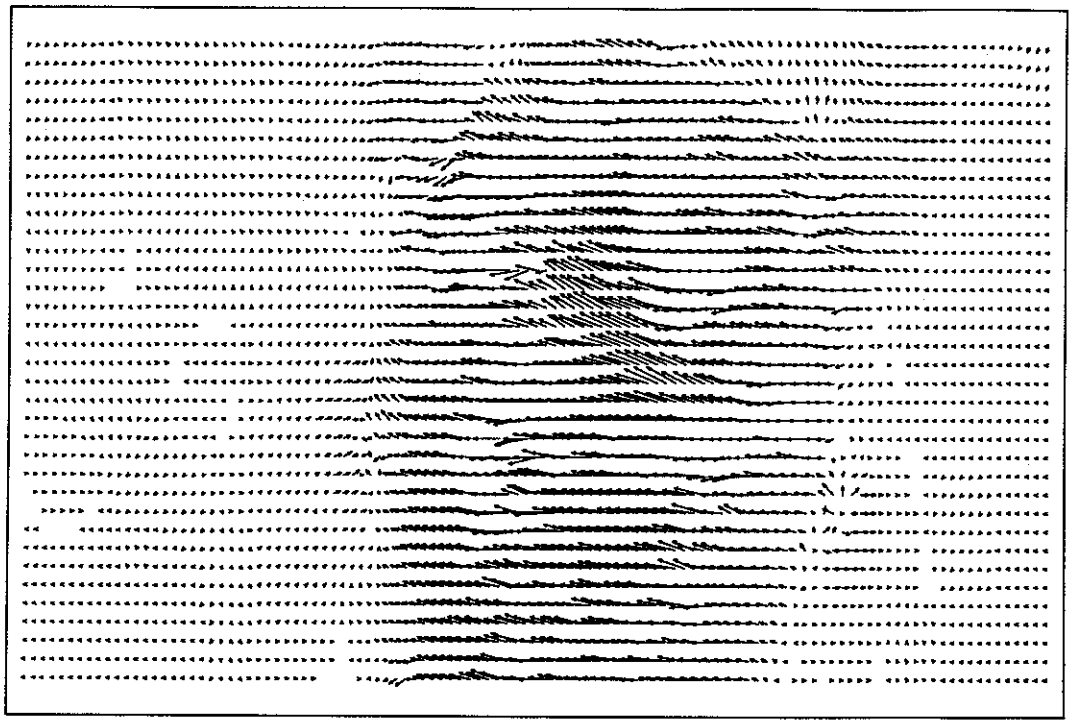


(a) discrete state-space, κ=1

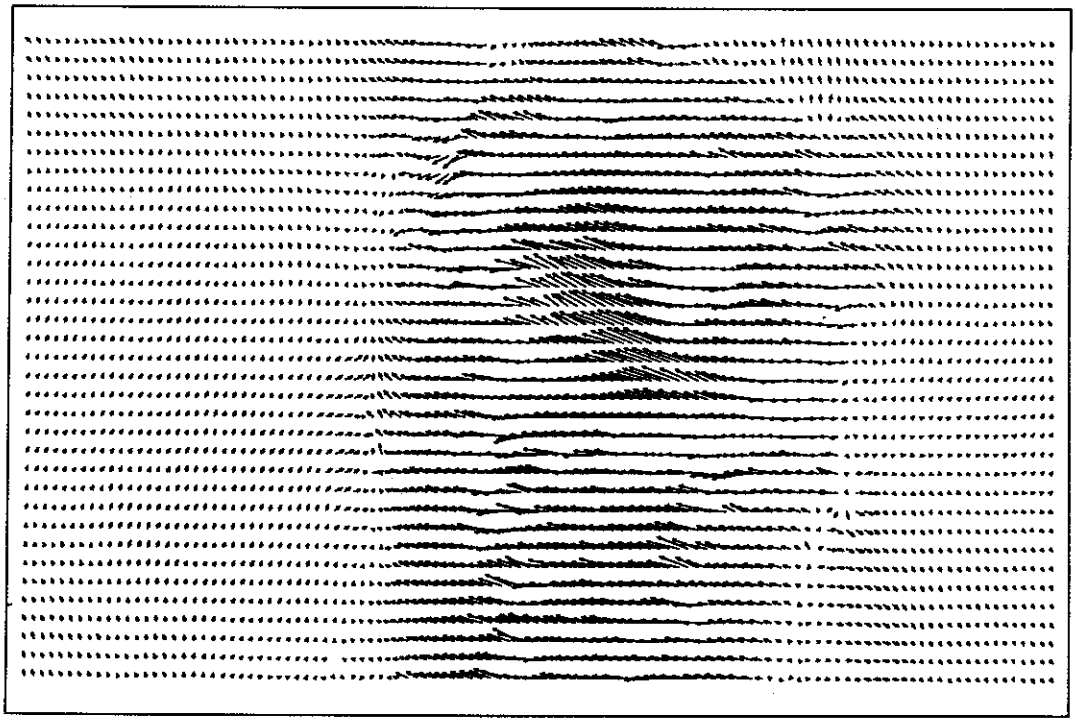


(b) continuous state-space, κ=1

Fig. 5.12 Hierarchical discrete and continuous state-space MAP estimates: test image 4, K_l=3, κ=1, λ_d/λ_g= (20.0,12.0,10.0), neighb. $\mathcal{N}_d^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}^\kappa$=(1.0,2.0,4.0), 200 (a) or 500 (b) iter.



(a) discrete state-space, κ=0



(b) continuous state-space, κ=0

Fig. 5.13 Hierarchical discrete and continuous state-space MAP estimates: test image 4, K_l=3, κ=0, $\lambda_{\mathbf{d}}/\lambda_g$= (20.0,12.0,10.0), neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}^{\kappa}$=(1.0,2.0,4.0), 200 (a) or 500 (b) iter.

Appendix 5.A. PROOF OF THE THEOREM FROM SECTION 5.1

The notation used here follows that used in Section 5.1. Let the waveform g be a transformed and shifted copy of the waveform f:

$$g(x) = s(x) * f(x - d_0),$$

where $s(x)$ is an impulse response of a linear operator, $*$ is a linear convolution and d_0 is a known displacement. d_0 can be estimated by minimizing the objective function:

$$\phi(\widehat{d}) = \sum_{x=-\infty}^{\infty} [f(x) - g(x + \widehat{d})]^2.$$

Assume that the estimator $\widehat{d}$ is constant over the whole domain (independent of x). Let f' and g' be the filtered versions of f and g:

$$f'(x) = h(x) * f(x)$$

$$g'(x) = h(x) * g(x),$$

where $h(x)$ is an impulse response of an LSI filter. Define a new objective function applied to the filtered data:

$$\phi'(\widehat{d}) = \sum_{x=-\infty}^{\infty} [f'(x) - g'(x + \widehat{d})]^2.$$

The relationship between $\phi(\widehat{d})$ and $\phi'(\widehat{d})$ is established by the following theorem:

Theorem: If $H(\nu)$ is the frequency response of the linear shift-invariant filter $h(x)$ with real-valued coefficients, then the Fourier transforms $\Phi(\nu)$ and $\Phi'(\nu)$ of $\phi(\widehat{d})$ and $\phi'(\widehat{d})$, respectively, are related through the following equation:

$$\Phi'(\nu) = \frac{1}{2\pi}\delta(\nu) \cdot (A' - A \cdot |H(0)|^2) + \Phi(\nu) \cdot |H(\nu)|^2,$$

where δ is a Dirac impulse and A, A' are constants dependent on signal f and operator s.

Proof:

Let $r(x, \widehat{d})$ be defined as follows:

$$r(x, \widehat{d}) = f(x) - g(x + \widehat{d}),$$

and let $R(\omega, \widehat{d})$ be the Fourier transform of $r(x, \widehat{d})$ with respect to x. Then, since g is a shifted and filtered copy of f (Section 5.1) it follows immediately that

$$R(\omega, \widehat{d}) = F(\omega) - F(\omega) \cdot S(\omega) \cdot e^{j\omega(\widehat{d} - d_0)}, \qquad (5.A.1)$$

where $F(\omega)$ and $S(\omega)$ are the Fourier transforms of $f(x)$ and $s(x)$, respectively.

Using the LSI property of the filter $h(x)$ and Parseval's theorem, the objective functions ϕ and ϕ' can be expressed as follows (note that x is discrete):

$$\phi(\hat{d}) = \sum_{x=-\infty}^{\infty} [r(x,\hat{d})]^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |R(\omega,\hat{d})|^2 d\omega$$
$$\phi'(\hat{d}) = \sum_{x=-\infty}^{\infty} [h(x) * r(x,\hat{d})]^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 \cdot |R(\omega,\hat{d})|^2 d\omega,$$

and consequently

$$\begin{aligned}\Phi(\nu) &= \int_{-\infty}^{\infty} \Big[\frac{1}{2\pi}\int_{-\pi}^{\pi} |R(\omega,\hat{d})|^2 d\omega\Big] \cdot e^{-j\nu\hat{d}} d\hat{d} \\ \Phi'(\nu) &= \int_{-\infty}^{\infty} \Big[\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 \cdot |R(\omega,\hat{d})|^2 d\omega\Big] \cdot e^{-j\nu\hat{d}} d\hat{d},\end{aligned} \tag{5.A.2}$$

where Φ and Φ' denote the Fourier transforms of ϕ and ϕ' respectively, and ν is a "displacement frequency". From (5.A.1) it can be shown that the squared magnitude of $R(\omega,\hat{d})$ is

$$\begin{aligned}|R(\omega,\hat{d})|^2 = |F(\omega)|^2 \cdot \{1 + S_R^2(\omega) + S_I^2(\omega) - 2S_R(\omega)cos[\omega(\hat{d}-d_0)] \\ + 2S_I(\omega)sin[\omega(\hat{d}-d_0)]\},\end{aligned} \tag{5.A.3}$$

where $S_R(\omega)$ and $S_I(\omega)$ are the real and imaginary parts of $S(\omega)$. Rearranging the order of integration, the expressions (5.A.2) can be rewritten as follows:

$$\begin{aligned}\Phi(\nu) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Q(\omega,\nu) d\omega \\ \Phi'(\nu) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Q(\omega,\nu) \cdot |H(\omega)|^2 d\omega,\end{aligned} \tag{5.A.4}$$

where $Q(\omega,\nu)$ is the Fourier transform of $|R(\omega,\hat{d})|^2$ with respect to $\hat{d}$. Based on (5.A.3) it can be shown that $Q(\omega,\nu)$ takes the following form:

$$\begin{aligned}Q(\omega,\nu) &= \int_{-\infty}^{\infty} |R(\omega,\hat{d})|^2 \cdot e^{-j\nu\hat{d}} d\hat{d} \\ &= |F(\omega)|^2 \cdot \Big\{[1 + S_R^2(\omega) + S_I^2(\omega)] \cdot \delta(\nu) - \\ &\quad [S_R(\omega)\cdot(cos\omega d_0 + sin\omega d_0) + S_I(\omega)\cdot(-cos\omega d_0 + sin\omega d_0)]\cdot\delta(\nu-\omega) - \\ &\quad [S_R(\omega)\cdot(cos\omega d_0 - sin\omega d_0) + S_I(\omega)\cdot(cos\omega d_0 + sin\omega d_0)]\cdot\delta(\nu+\omega)\Big\},\end{aligned}$$

where δ denotes a Dirac impulse. It is clear that the spectrum of $Q(\omega,\nu)$ is concentrated in the narrow pencils along the ω axis ($\delta(\omega)$), and along $\pm 45°$ axes ($\delta(\nu-\omega)$ and $\delta(\nu+\omega)$), as can be seen in Fig.5.A.1.

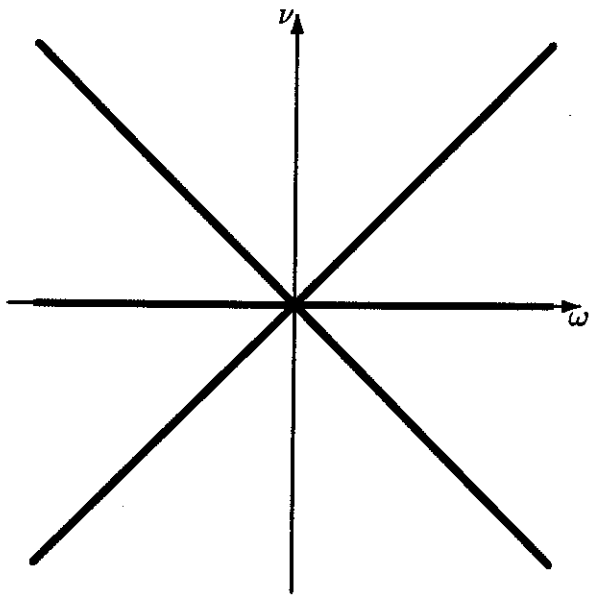


Fig. 5.A.1 2-D frequency occupancy by the function $Q(\omega, \nu)$.

Finally, using $Q(\omega, \nu)$ in (5.A.4) and performing the integration with respect to ω it can be shown that:

$$\begin{aligned}\Phi(\nu) =& \frac{1}{2\pi}\Big\{A \cdot \delta(\nu) - \\ & |F(\nu)|^2 \cdot [S_R(\nu) \cdot (cos\nu d_0 + sin\nu d_0) + S_I(\nu) \cdot (-cos\nu d_0 + sin\nu d_0)] - \\ & |F(-\nu)|^2 \cdot [S_R(\nu) \cdot (cos\nu d_0 + sin\nu d_0) + S_I(\nu) \cdot (cos\nu d_0 - sin\nu d_0)]\Big\},\end{aligned} \tag{5.A.5}$$

and

$$\begin{aligned}\Phi'(\nu) = & \frac{1}{2\pi}\Big\{A' \cdot \delta(\nu) - \\ & |F(\nu)|^2 \cdot [S_R(\nu) \cdot (cos\nu d_0 + sin\nu d_0) + S_I(\nu) \cdot (-cos\nu d_0 + sin\nu d_0)] \cdot |H(\nu)|^2 - \\ & |F(-\nu)|^2 \cdot [S_R(\nu) \cdot (cos\nu d_0 + sin\nu d_0) + S_I(\nu) \cdot (cos\nu d_0 - sin\nu d_0)] \cdot |H(-\nu)|^2\Big\},\end{aligned} \tag{5.A.6}$$

where

$$A = \int_{-\pi}^{\pi} |F(\omega)|^2 \cdot [1 + S_R^2(\omega) + S_I^2(\omega)] d\omega$$

$$A' = \int_{-\pi}^{\pi} |F(\omega)|^2 \cdot [1 + S_R^2(\omega) + S_I^2(\omega)] \cdot |H(\omega)|^2 d\omega.$$

The first terms in (5.A.5) and (5.A.6) are just DC offsets, while the next two terms are some spectra over frequency ν. Note that the last two terms in (5.A.6) are equal to those in (5.A.5) multiplied by the squared magnitude of the filter frequency response. Under the

reasonable assumption that filter coefficients are real it follows that:

$$\Phi'(\nu) = \frac{1}{2\pi}\delta(\nu) \cdot (A' - A \cdot |H(0)|^2) + \Phi(\nu) \cdot |H(\nu)|^2,$$

which tells that $\phi'(\widehat{d})$ is a filtered version of $\phi(\widehat{d})$ with certain DC offset. □

Chapter 6

PIECEWISE SMOOTH MODEL FOR MOTION

As was demonstrated in Chapter 3 the homogeneously smooth model for motion over the complete image is insufficient for accurate representation of motion properties in many TV image sequences. If such a sequence is characterized by a rigid body motion, its motion field should consist of patches of similarly oriented and long vectors. Discontinuities between those patches are known as motion boundaries, and usually correspond to occlusion borders. In this chapter a two-layer model comprising a VMRF model for motion vectors (similar to that presented in Chapter 3) and a binary MRF (BMRF) model for motion discontinuities will be presented.

In the next section a stochastic process modeling motion discontinuities will be defined, as well as its sampling structure and its state-space. Then, the MAP estimation criterion will be derived, followed by the description of the displacement field model with discontinuities and of the line field model itself. In the subsequent sections the *a posteriori* probability and the Gibbs sampler for coupled VMRF-BMRF model will be described. The chapter will be concluded with experimental results.

6.1 TERMINOLOGY

To describe sudden changes in motion vector length and/or orientation I will use the concept of motion discontinuity. The true motion discontinuities are defined over continuous spatio-temporal coordinates $(\mathbf{x}, t)$, and are unobservable like the true motion fields. They are just a means of describing the motion in an image and can be understood as indicator

functions (e.g., binary) for each $(\mathbf{x}, t)$. Let the unknown (true) field of such discontinuities be denoted by $\boldsymbol{l}$. Let $\boldsymbol{l}$ be a sample from a RF L and let l be any sample of motion discontinuities drawn from L. Let an estimate of $\boldsymbol{l}$ be denoted by $\widehat{l}$. The RF L will be called a *line process*, its sample l will be called a *line field* while individual discontinuities from l will be named *line elements*. I assume that the line elements are defined over a union of cosets (shifted lattices) [21] $\Psi_l = \psi_h \cup \psi_v$, where ψ_h and ψ_v are orthogonal cosets of horizontal and vertical line elements, respectively. Those cosets are defined as follows:

$$\psi_h = \Lambda_{\mathbf{d}} + [0, T_v^{\mathbf{d}}/2, 0]^T$$
$$\psi_v = \Lambda_{\mathbf{d}} + [T_h^{\mathbf{d}}/2, 0, 0]^T,$$

where $T_h^{\mathbf{d}}$ and $T_v^{\mathbf{d}}$ have been defined in Section 3.1 and superscript T denotes a transposition. Consequently there are $M_l = M_{\mathbf{d}}^h \times (M_{\mathbf{d}}^v - 1) + (M_{\mathbf{d}}^h - 1) \times M_{\mathbf{d}}^v$ horizontal and vertical line elements.

It is assumed that the random field L is defined over the discrete state-space $\mathcal{S}_l = (\mathcal{S}_l')^{M_l}$, where again $\mathcal{S}_l'$ is the single line element state-space. For the purpose of this work it is assumed that RF L is binary and $\mathcal{S}_l'$ consists of two states: 0 – absence of a line element (no motion discontinuity) and 1 – a line element "turned on" (motion discontinuity). Possible extensions of this state-space to non-binary spaces (e.g., incorporating the directionality of line elements) are not considered in this thesis, but can be found in [26].

6.2 ESTIMATION CRITERION

The objective here is again to find the true displacement field $\boldsymbol{d}(\mathbf{x}, t)$ corresponding to an underlying time-varying image $u(\mathbf{x}, t)$ on the basis of the observations $g(\mathbf{x}, t)$. Since the line field $\boldsymbol{l}(\mathbf{x}, t)$ is assumed to aid in describing the true motion field $\boldsymbol{d}(\mathbf{x}, t)$, it must be estimated, too.

In order to determine the most likely displacement field $\widehat{\mathbf{d}}_t^* \in \mathcal{S}_{\mathbf{d}}$ and line field $\widehat{l}_t^* \in \mathcal{S}_l$ given the observations g_{t_-}, g_{t_+}, a pair $(\widehat{\mathbf{d}}_t^*, \widehat{l}_t^*)$ which satisfies the relationship:

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

must be found. Applying Bayes rule for discrete random variables the following posterior distribution can be obtained (again sample fields $\mathbf{d}_t$, l_t, g_{t_-}, g_{t_+} are omitted):

$$P(\mathbf{D}_t, L_t|G_{t_-}, G_{t_+}) = \frac{P(G_{t_+}|\mathbf{D}_t, L_t, G_{t_-}) \cdot P(\mathbf{D}_t, L_t|G_{t_-})}{P(G_{t_+}|G_{t_-})} \tag{6.1}$$

In Chapter 3 it was assumed that the random field $\mathbf{D}$ at time t is statistically independent of the observation G at time t_- i.e., that the knowledge of a single image field provides no information about the motion. Consequently, since no line process L_t was used, the probability $P(\mathbf{D}_t|G_{t_-})$ could be simplified to $P(\mathbf{D}_t)$. That was a coarse approximation in the model, because in some circumstances certain single-image information can be used in motion computation. For example, it is unlikely to have a motion discontinuity in a constant-intensity image region, and such an occurrence should be penalized. However, it is possible to have a motion discontinuity at a position with significant appropriate (horizontal or vertical) gradient, and such occurrence should not be penalized. In this chapter I still assume no direct dependence between $\mathbf{D}_t$ and G_{t_-}, however I assume an indirect dependence through the line process L_t.

Note that since the probability in the denominator of (6.1) is not a function of the displacement process $\mathbf{D}_t$, it can be ignored when maximizing $P(\mathbf{D}_t, L_t|G_{t_-}, G_{t_+})$ with respect to $(\widehat{\mathbf{d}}_t, \widehat{l}_t)$, and the MAP estimate of the pair (d_t, l_t) is the solution to the following optimization problem:

$$\max_{(\widehat{\mathbf{d}}_t, \widehat{l}_t)} \Big[P(G_{t_+} = g_{t_+}|\mathbf{D}_t = \widehat{\mathbf{d}}_t, L_t = \widehat{l}_t, G_{t_-} = g_{t_-}) \cdot P(\mathbf{D}_t = \widehat{\mathbf{d}}_t, L_t = \widehat{l}_t|G_{t_-} = g_{t_-})\Big]. \tag{6.2}$$

6.3 MODELS

Since the new characterization of the displacement field affects only the *a priori* knowledge of motion, the same structural (Section 3.4.1) and observation (Section 3.4.2) models as before apply. Consequently only the models for motion and its discontinuities will be discussed.

6.3.1 Displacement field model with discontinuities

In Section 3.4.3 a globally (over the complete image) smooth displacement field model has been proposed, and a 2D VMRF $\mathbf{D}_t$ has been used to model the motion fields $\mathbf{d}_t$. Such a homogeneous smoothness assumption, however, is violated at the boundaries of moving objects. For example two vectors positioned across a motion boundary are subjected to the same smoothness constraint as other vectors not positioned across such a boundary. With this simple model the motion vectors in the vicinity of a motion boundary become oversmoothed. In order to reduce this effect a binary MRF model for discontinuities is introduced into the motion model. The idea of modeling discontinuities by a coupled MRF has been introduced by Geman and Geman [26] for image restoration, and subsequently used for boundary detection [27] and segmentation of moving planar surfaces [67]. A deterministic version of line field has been used in optical flow computation [44] via non-stochastic methods.

I will model the vector field – line field pair $(\mathbf{d}_t, l_t)$ by the vector-binary MRF field pair $(\mathbf{D}_t, L_t)$. This pair being jointly Markovian is characterized by the Gibbs distribution as follows:

$$P(\mathbf{D}_t, L_t) = \pi(\mathbf{d}_t, l_t) = \frac{1}{Z} e^{-U(\mathbf{d}_t, l_t)/\beta},$$

where U is an energy function, and Z, β are constants as usual. Using the Bayes rule this probability can be factored as follows:

$$P(\mathbf{D}_t, L_t) = P(\mathbf{D}_t | L_t) \cdot P(L_t),$$

which corresponds to decomposition of the energy $U(\mathbf{d}_t, l_t)$ into two terms: $U(\mathbf{d}_t | l_t)$ and $U(l_t)$ [26]. If $U(\mathbf{d}_t, l_t)$ and $U(l_t)$ are both non-negative energy functions, then $P(\mathbf{D}_t | L_t)$ and $P(L_t)$ are Gibbsian, and $\mathbf{D}_t$ given L_t as well as L_t by itself are Markovian.

As far as the probability $P(\mathbf{D}_t, L_t | G_{t\_})$ from (6.2) is concerned, it can be factored as follows:

$$P(\mathbf{D}_t, L_t | G_{t\_}) = P(\mathbf{D}_t | L_t, G_{t\_}) \cdot P(L_t | G_{t\_}) \tag{6.3}$$

Due to the assumed direct independence between D_t and $G_{t\_}$, $P(\mathbf{D}_t | L_t, G_{t\_})$ is equal to $P(\mathbf{D}_t | L_t)$ and consequently is Gibbsian. Also if $P(L_t | G_{t\_})$ is defined in such a way that

it equals $P(L_t) \cdot e^{-\sum V_{l_1}}$, where V_{l_1} is a non-negative potential, it will be Gibbsian (recall that $P(L_t)$ is Gibbsian). Consequently $P(\mathbf{D}_t, L_t|G_{t\_})$ is also Gibbsian.

Let $P(\mathbf{D}_t|L_t)$ be defined as follows:

$$\pi(\mathbf{d}_t|l_t) = \frac{1}{Z_\mathbf{d}} e^{-U_\mathbf{d}(\mathbf{d}_t|l_t)/\beta_\mathbf{d}}, \tag{6.4}$$

where $Z_\mathbf{d}$,$\beta_\mathbf{d}$ have the same meaning as before, and the conditional energy $U_\mathbf{d}(\mathbf{d}_t|l_t)$ is defined as:

$$U_\mathbf{d}(\mathbf{d}_t|l_t) = \sum_{c_\mathbf{d}=\{\mathbf{x}_i,\mathbf{x}_j\}\in\mathcal{C}_\mathbf{d}} V_\mathbf{d}(\mathbf{d}_t, c_\mathbf{d}) \cdot [1 - l(<\mathbf{x}_i, \mathbf{x}_j>, t)]. \tag{6.5}$$

Again $c_\mathbf{d}$ is a vector clique, while $\mathcal{C}_\mathbf{d}$ is a set of all vector cliques defined over $\Lambda_\mathbf{d}$. Only two-element cliques will be considered here. $V_\mathbf{d}$ is the same potential function as before, and $(<\mathbf{x}_i, \mathbf{x}_j>, t) \in \Psi_l$ denotes a site of line element located between vector sites $\mathbf{x}_i$ and $\mathbf{x}_j$ which belong to $\Lambda_\mathbf{d}$.

The energy function (6.5) can be understood as follows. There exists a cost $V_\mathbf{d}(\mathbf{d}_t, c_\mathbf{d})$ associated with each vector clique $c_\mathbf{d}$ which increases if a motion field sample locally departs from the assumed *a priori* model. This model is characterized by $\beta_\mathbf{d}$ and $V_\mathbf{d}$. If, however, the line element separating the displacement vectors from clique $c_\mathbf{d}$ is "turned on" ($l(<\mathbf{x}_i, \mathbf{x}_j>, t)=1$), there is no cost associated with the clique $c_\mathbf{d}$. 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 (6.5). This penalty is provided by the line field model described in the next section.

To specify the *a priori* displacement model (and the hence distribution) the neighbourhood system $\mathcal{N}_{(\mathbf{d},l)}$, cliques $c_\mathbf{d}$ and the potential function $V_\mathbf{d}$ have to be specified. In this chapter the first-order neighbourhood system $\mathcal{N}^1_{(\mathbf{d},l)}$ depicted in Fig. 6.1.a, which consists of 2-element vector cliques (Fig. 6.1.b, 6.1.c) defined in Section 3.4.3 is used. Note that every displacement vector has 4 vector neighbours and 4 line neighbours. The potential function over $c_\mathbf{d}$ is defined as before by (3.17).

Fig. 6.1 First-order neighbourhood system $\mathcal{N}^1_{(\mathrm{d},l)}$ for vector field $\mathbf{d}_t$ defined over $\Lambda_\mathbf{d}$ with discontinuities (line elements) l_t defined over Ψ_l(a), and associated horizontal (b) and vertical (c) cliques (o – center vector site, • – vector site, × – line site).

6.3.2 Line field model

The line field model is based on a binary MRF L_t, and is described by the Gibbs probability distribution

$$P(L_t|G_{t\_}) = \pi(l_t|g_{t\_}) = \frac{1}{Z_l} e^{-U_l(l_t|g_{t\_})/\beta_l}, \tag{6.6}$$

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

$$U_l(l_t|g_{t\_}) = \sum_{c_l \in \mathcal{C}_l} V_l(l_t, g_{t\_}, c_l), \tag{6.7}$$

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

The neighbourhood system for the "dual" sampling structure Ψ_l is shown in Fig. 6.2. Note that since the union of cosets $\Psi_l = \psi_h \cup \psi_v$ identifies positions of horizontal and vertical line elements, two neighbourhood systems are defined (Figs. 6.2.a, 6.2.b). Every line element has 8 line neighbours and 2 vector neighbours. There are two types of four-element line cliques. The cliques from Fig. 6.2.c are the same as used in [26] and aim at modeling the shape of motion boundaries, while the cliques from Fig. 6.2.d exclude isolated vectors. The two-element vertical cliques of horizontal line elements (Fig. 6.2.e) and the horizontal cliques of vertical line elements (Fig. 6.2.f) are used after Marroquin [62]. Possible configurations (up to a rotation) and related costs are shown in Fig. 6.3.a for the four-element clique from Fig. 6.2.c and in Fig. 6.3.b for the two-element clique. Note

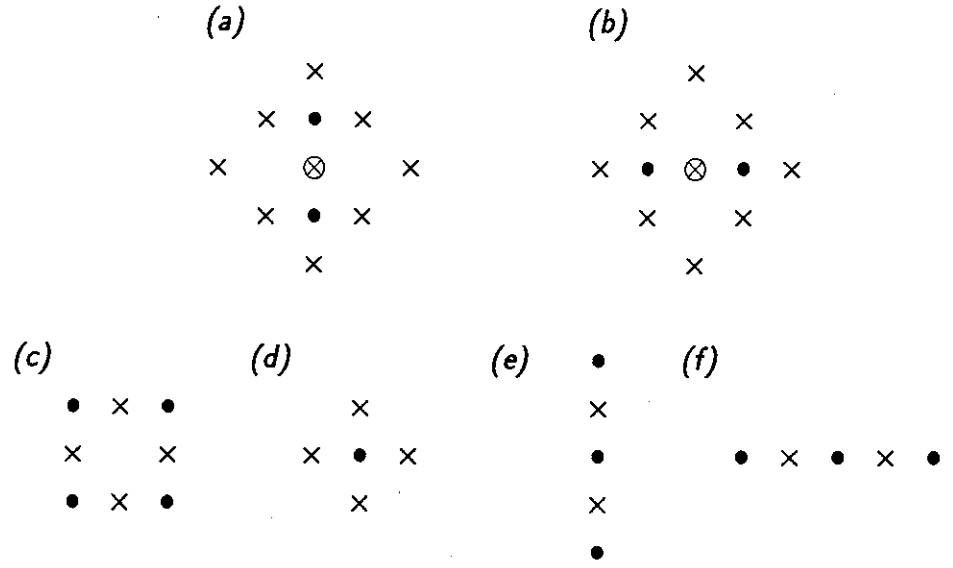


Fig. 6.2 Neighbourhood system $\mathcal{N}_l$ for line field l_t defined over Ψ_l: neighbourhood of horizontal (a) and vertical (b) line element, and associated four-element (c),(d) and two-element (e),(f) cliques ($\otimes$ – center line site, $\times$ – line site, $\bullet$ – vector site).

that such potentials encourage absence of line elements (V_{l_4}=0.0), slightly penalize straight lines (V_{l_4}=0.4) and corners (V_{l_4}=0.8), and more heavily penalize ends of lines (V_{l_4}=1.2) as well as the three-element (V_{l_4}=1.2) and the four-element (V_{l_4}=2.0) intersections. The "double edges" and sharp turns are discouraged by the high penalty (V_{l_2}=3.2) associated with the two-element cliques. The four-element clique from Fig. 6.2.d is used only to exclude isolated vectors by assigning $V_{l_4} = \infty$ when all four line elements are "on", since other configurations are incorporated in the clique from Fig. 6.2.c. The boundary conditions for the line cliques are handled in such a way as if there were a frame of line elements around the motion field. In this way motion boundaries extending beyond the field are discouraged unless there is a strong cue from the data to do so.

So far the line model has been defined based only on the relationship between the line elements and the displacement vectors. Note, however, that the *a priori* probability of the line process (6.6) is conditioned on the observations. It means that one should take into account the image information $g_{t\_}$ when computing the line samples l_t. If independence between L_t and $G_{t\_}$ were claimed, then only $P(l_t)=\pi(l_t)$ could have been used, and the motion discontinuities would have been inferred only from the displacement field $\mathbf{d}_t$. It is

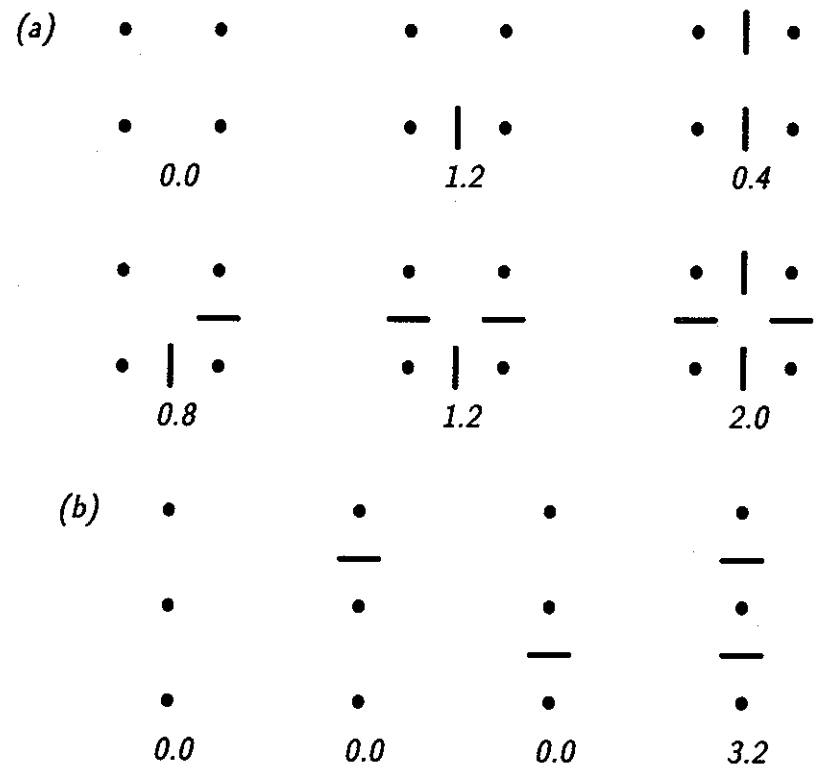


Fig. 6.3 Costs V_{l_4}, V_{l_2} associated with various configurations (up to a rotation) of the four-element (a) and two-element (b) cliques. (• – vector site, ▬ – line element "turned on").

clear, however, that there exists a relationship between image intensity and motion discontinuities. In general a 3D 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, I assume that an introduction of a line element should coincide with an intensity edge, and I will use the following potential function for one-element clique:

$$V_{l_1}(l_t, g_{t\_}, c_l) = \begin{cases} \frac{\alpha}{(\nabla_v g_{t\_})^2} \cdot l_h(<\mathbf{x}_i, \mathbf{x}_j>, t) & \text{for horizontal } c_l = \{\mathbf{x}_i, \mathbf{x}_j\} \\ \frac{\alpha}{(\nabla_h g_{t\_})^2} \cdot l_v(<\mathbf{x}_i, \mathbf{x}_j>, t) & \text{for vertical } c_l = \{\mathbf{x}_i, \mathbf{x}_j\}, \end{cases} \tag{6.8}$$

where l_h, l_v are horizontal and vertical line elements, ∇_h, ∇_v are horizontal and vertical components of the spatial gradient at position $(<\mathbf{x}_i, \mathbf{x}_j>, t)$, and α is a constant. Note that the potential V_{l_1} is non-negative, hence $P(L_t|G_{t\_})$ from (6.3) is Gibbsian like $P(L_t)$. The above potential introduces a penalty only if a line element is "on" and the appropriate

gradient is relatively small. For example with α=10.0 a vertical element at a position with horizontal gradient ∇_h=5.0 will cause a penalty of 0.4 i.e., equivalent to the smallest penalty of a non-zero line element (two in-line elements), while if ∇_h=10.0 the penalty will drop to 0.1.

The total potential function for the line field can be expressed as:

$$V_l(l_t, g_{t_-}, c_l) = V_{l_4}(l_t, c_l) + V_{l_2}(l_t, c_l) + V_{l_1}(l_t, g_{t_-}, c_l),$$

where V_{l_4} and V_{l_2} are tabulated in Fig. 6.3, and V_{l_1} is given above.

6.4 A POSTERIORI PROBABILITY

Combining the conditional likelihood $P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \hat{\mathbf{d}}_t, L_t = \hat{l}_t, G_{t_-} = g_{t_-})$ from (3.13), the displacement *a priori* probability $P(\mathbf{D}_t = \hat{\mathbf{d}}_t | L_t = \hat{l}_t)$ from (6.4) and the line *a priori* probability $P(L_t = \hat{l}_t | G_{t_-} = g_{t_-})$ from (6.6) via (6.1) the following Gibbs form of the *a posteriori* probability can be obtained

$$P(\mathbf{D}_t = \hat{\mathbf{d}}_t, L_t = \hat{l}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) = \frac{1}{P(G_{t_+} | G_{t_-})} \cdot \frac{1}{Z} e^{-U(\hat{\mathbf{d}}_t, \hat{l}_t, g_{t_-}, g_{t_+})} \quad (6.9)$$

where Z is a new normalizing constant, and the new energy function $U(\hat{\mathbf{d}}_t, \hat{l}_t, g_{t_-}, g_{t_+})$ is

$$U(\hat{\mathbf{d}}_t, \hat{l}_t, g_{t_-}, g_{t_+}) = \lambda_g \cdot U_g(g_{t_+} | \hat{\mathbf{d}}_t, g_{t_-}) + \lambda_{\mathbf{d}} \cdot U_{\mathbf{d}}(\hat{\mathbf{d}}_t | \hat{l}_t) + \lambda_l \cdot U_l(\hat{l}_t | g_{t_-}). \quad (6.10)$$

The conditional energies in the above relationship are defined in (3.14), (6.5) and (6.7) respectively, and λ_g=$1/(2\sigma^2)$, $\lambda_{\mathbf{d}}$=$1/\beta_{\mathbf{d}}$, λ_l=$1/\beta_l$.

Having shown that the posterior distribution (6.1) is Gibbsian, it follows that MAP estimation can be achieved by means of the following minimization

$$\min_{\{\hat{\mathbf{d}}_t, \hat{l}_t\}} \lambda_g \cdot U_g(g_{t_+} | \hat{\mathbf{d}}_t, g_{t_-}) + \lambda_{\mathbf{d}} \cdot U_{\mathbf{d}}(\hat{\mathbf{d}}_t | \hat{l}_t) + \lambda_l \cdot U_l(\hat{l}_t | g_{t_-}) \quad (6.11)$$

Note that the functional under minimization is again in a regularized form, where $\lambda_{\mathbf{d}} \cdot U_{\mathbf{d}}(\hat{\mathbf{d}}_t | \hat{l}_t) + \lambda_l \cdot U_l(\hat{l}_t | g_{t_-})$ is a stabilizing functional and $1/\lambda_g$ is a regularization parameter [74].

The objective function in (6.11) is similar to that used in [44], which is derived from the original formulation of Horn and Schunck [41] with additional deterministic motion

discontinuity model. I pursue the stochastic approach by using two coupled MRFs and the displaced pel difference to solve the 2D correspondence problem instead of the motion constraint equation. Also the line cliques are slightly different and the penalty for introducing a line element is continuous (inversely proportional to the squared magnitude of the spatial intensity gradient) rather than binary [44]. Most importantly, however, I will use a stochastic relaxation algorithm for minimization (6.11).

6.5 GIBBS SAMPLER FOR MOTION MODEL WITH DISCONTINUITIES

Since by Theorem A from [26] the scanning order is irrelevant for the convergence of $\pi(\cdot)$ to some steady-state distribution, the unknowns can be organized so that first the $\mathbf{d}$'s are updated and then the l's. In order to implement the Gibbs sampler the marginal conditional *a posteriori* probabilities for displacement vectors and for line elements must be known. Note that due to the Markovian property the condition that $j \neq i$ is equivalent to $j \in \eta_{\mathbf{d}}(\mathbf{x}_i)$, where $\eta_{\mathbf{d}}(\mathbf{x}_i) \in \mathcal{N}_{\mathbf{d}}$. Similarly as in Section 4.2.3 it can be shown that the conditional probability of a displacement vector at location $\mathbf{x}_{n_\tau}$ can be expressed as follows:

$$P(\mathbf{D}(\mathbf{x}_{n_\tau},t)=\widehat{\mathbf{d}}(\mathbf{x}_{n_\tau},t)|\mathbf{D}(\mathbf{x}_j,t)=\widehat{\mathbf{d}}(\mathbf{x}_j,t), j\neq n_\tau, \widehat{l}_t, g_{t_-}, g_{t_+}) = \frac{e^{-U_{\mathbf{d}}^{n_\tau}(\widehat{\mathbf{d}}(\mathbf{x}_{n_\tau},t)|\widehat{\mathbf{d}}_t,\widehat{l}_t,g_{t_-},g_{t_+})}}{\sum\limits_{z\in\mathcal{S}'_{\mathbf{d}}} e^{-U_{\mathbf{d}}^{n_\tau}(z|\widehat{\mathbf{d}}_t,\widehat{l}_t,g_{t_-},g_{t_+})}}, \qquad (\mathbf{x}_{n_\tau},t),(\mathbf{x}_j,t)\in\Lambda_{\mathbf{d}}. \tag{6.12}$$

For the potential function (3.17) the local displacement energy function $U_{\mathbf{d}}^{n_\tau}$ driving the Gibbs sampler is defined as

$$U_{\mathbf{d}}^{n_\tau}(z|\widehat{\mathbf{d}}_t,\widehat{l}_t,g_{t_-},g_{t_+}) = \lambda_g\cdot[\widetilde{r}(z,\mathbf{x}_{n_\tau},t,\Delta t)]^2 + \lambda_{\mathbf{d}}\cdot\sum_{j:\,\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_{n_\tau})}\|z-\widehat{\mathbf{d}}(\mathbf{x}_j,t)\|^2\cdot[1-\widehat{l}(<\mathbf{x}_{n_\tau},\mathbf{x}_j>,t)].$$

It can be also demonstrated that the conditional probability of a line element at location $(\mathbf{y}_{n_\tau},t)\in\Psi_l$ is

$$P(L(\mathbf{y}_{n_\tau},t)=\widehat{l}(\mathbf{y}_{n_\tau},t)|L(\mathbf{y}_j,t)=\widehat{l}(\mathbf{y}_j,t), j\neq n_\tau, \widehat{\mathbf{d}}_t, g_{t_-}) = \frac{e^{-U_l^{n_\tau}(\widehat{l}(\mathbf{y}_{n_\tau},t)|\widehat{l}_t,\widehat{\mathbf{d}}_t,g_{t_-})}}{\sum\limits_{z\in\mathcal{S}'_l} e^{-U_l^{n_\tau}(z|\widehat{l}_t,\widehat{\mathbf{d}}_t,g_{t_-})}}, \qquad (\mathbf{y}_{n_\tau},t),(\mathbf{y}_j,t)\in\Psi_l, \tag{6.13}$$

where for the line potential function V_l the local line energy function $U_l^{n_\tau}$ is defined as

$$U_l^{n_\tau}(z|\widehat{l}_t, \widehat{\mathbf{d}}_t, g_{t\_}) = \lambda_{\mathbf{d}} \cdot \sum_{\substack{\{\mathbf{x}_j, \mathbf{x}_k\}: \\ <\mathbf{x}_j, \mathbf{x}_k> = \mathbf{y}_{n_\tau}}} \|\widehat{\mathbf{d}}(\mathbf{x}_j, t) - \widehat{\mathbf{d}}(\mathbf{x}_k, t)\|^2 \cdot [1 - z] + \lambda_l \cdot \sum_{c_l:\, \mathbf{y}_{n_\tau} \in c_l} V_l(z, g_{t\_}, c_l).$$

Note that the local line energy is conditional on the data and the displacement field. Thus, in order to expect reasonable discontinuity estimates, the displacement field must be known to some extent. In practice it means that the Gibbs sampler for the displacement field should run for a number of iterations with no discontinuity field until a coarse estimate is obtained. Only then should the Gibbs sampler for the line elements be turned on, and subsequently interleaved with the displacement Gibbs sampler.

6.5.1 Gibbs sampler for the discrete state-space $\mathcal{S}_{\mathbf{d}}$

In order to generate displacement field samples from the probability distribution (6.12) the same procedure as that described in Appendix 4.C, but with the local energy $U_{\mathbf{d}}^{n_\tau}$ defined in the previous section, can be used. Line elements from the probability distribution (6.13) are obtained by computing the probabilities associated with possible states of a line element (0 or 1), and generating a random deviate according to these probabilities.

6.5.2 Gibbs sampler for the continuous state-space $\mathcal{S}_{\mathbf{d}} = R^2$

As in the case of the continuous state-space Gibbs sampler for motion model without discontinuities (Section 4.5), a first-order Taylor expansion will be applied to the displaced pel difference $\widetilde{r}$. It will be demonstrated that given the line process estimate $\widehat{l}_t$ the conditional probability (6.12) can be approximated by a Gaussian distribution, thus allowing simple generation of a 2-D vector deviate. Generation of a line sample according to the conditional probability (6.13) is obviously the same as in the previous section.

Recall the approximation (4.17) of the local energy for the motion model without discontinuities. Taking into account the *a posteriori* energy (6.10) of the model with such

discontinuities this approximation can be expressed as follows:

$$U^i_{\mathbf{d}}(\hat{\mathbf{d}}(\mathbf{x}_i,t)|\hat{\mathbf{d}},\hat{l}_t,g_{t_-},g_{t_+}) \approx$$
$$\lambda'_g \cdot [\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)+(\hat{\mathbf{d}}(\mathbf{x}_i,t)-\dot{\mathbf{d}}(\mathbf{x}_i,t))\cdot\nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2+ \quad (6.14)$$
$$\lambda'_{\mathbf{d}}\cdot\sum_{j:\,\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_i)} V(\hat{\mathbf{d}}(\mathbf{x}_j,t),\hat{\mathbf{d}}(\mathbf{x}_i,t))\cdot[1-\hat{l}(<\mathbf{x}_i,\mathbf{x}_j>,t)].$$

Given the line process $\hat{l}_t$, this energy is again quadratic in $\hat{\mathbf{d}}$, and again it can be shown that the conditional probability density (6.12) is a bivariate Gaussian. Note that the local energy (6.14) differs from the energy (4.17) by the line elements. The same update equation, ε_i and μ_i, as well as the means and the covariance matrix as those derived in Section 4.5 and Appendix 4.D apply, except for the definitions of ξ_i and $\overline{\mathbf{d}}$. It can be shown that taking into account the line process, ξ_i is defined as the following sum of line elements over $\eta_{\mathbf{d}}(\mathbf{x}_i)$:

$$\xi_i = \sum_{j:\,\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_i)} [1-\hat{l}(<\mathbf{x}_i,\mathbf{x}_j>,t)] \quad (6.15)$$

instead of being the cardinality of $\eta_{\mathbf{d}}(\mathbf{x}_i)$. It can be also easily demonstrated that the vector $\overline{\mathbf{d}}$ denotes the following weighted average over $\eta_{\mathbf{d}}(\mathbf{x}_i)$:

$$\overline{\mathbf{d}}(\mathbf{x}_i,t)=\frac{1}{\xi_i}\sum_{j:\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_i)}\mathbf{d}(\mathbf{x}_j,t)[1-\hat{l}(<\mathbf{x}_i,\mathbf{x}_j>,t)], \quad (6.16)$$

Note that this definition is different from (4.19). This modified definition of averaging takes into account the line elements and simply does not allow to perform such operation across a line element. If such an element is associated with a motion boundary (it should be), the averaging will not take effect across a motion boundary, which is a desirable property.

6.6 PIECEWISE SMOOTH MOTION MODEL OVER HIERARCHY OF RESOLUTIONS

The concept of motion discontinuity can be extended to a hierarchy of resolutions. Recall the image and displacement pyramids defined in Chapter 5, and define a discontinuity pyramid with line elements spatially interleaved with appropriate displacement vectors. Such motion discontinuities may not be as well defined as at the full resolution due to the filtered data and consequently smoother displacement field estimate. One can hope, however, that introduction of such discontinuities at low resolution levels will limit excessive smoothing

across moving edges. Even if not precisely positioned, these discontinuities can be corrected at subsequent higher resolution levels.

If $\widehat{l}_t^{\kappa}$ denotes the displacement discontinuity estimate at level κ, then the following minimization problem with respect to the pair $\{\widehat{\mathbf{h}}_t^{\kappa}, \widehat{l}_t^{\kappa}\}$ can be formulated at resolution level κ:

$$\min_{\{\widehat{\mathbf{h}}_t^{\kappa}, \widehat{l}_t^{\kappa}\}} \lambda_g \cdot U_g(g_{t+}^{\kappa} | \widehat{\mathbf{d}}_t^{\kappa} + \mathbf{b}_t^{\kappa}, g_{t-}^{\kappa}) + \lambda_{\mathbf{d}} \cdot U_{\mathbf{d}}(\widehat{\mathbf{d}}_t^{\kappa} + \mathbf{b}_t^{\kappa} | \widehat{l}_t^{\kappa}) + \lambda_l \cdot U_l(\widehat{l}_t^{\kappa} | g_{t-}^{\kappa}). \qquad (6.17)$$

Recall that the *base* displacement is fixed during the minimization and is only updated during inter-level interpolation. The estimate $\widehat{l}_t^{\kappa}$ has to be also transferred to subsequent resolution level κ-1. One possibility would be to simply pass the line elements from level κ to level κ-1, especially if the *even* pyramid is used (Fig. 5.1). The missing line elements would have to be somehow interpolated from the existing ones with eventual aid of the data (e.g., the intensity gradient). For the odd pyramid situation is a little more complex since the position of a line element at level κ coincides with a vector position at level κ-1. A "parent-children" propagation could be performed, thus generating two "children" from each "parent" line element. If there is a high penalty associated with such neighbouring parallel line elements, they will quickly disappear forming continuous single-element contours.

There is another alternative, however. 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. 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 iterations and is turned on once a coarse displacement estimate is known. This is consonant with the earlier remark that the discontinuity estimation should be started only after certain knowledge of the motion field had been acquired. This approach will be used for line process interpolation in the hierarchical estimation.

6.7 EXPERIMENTAL RESULTS

In this section some experimental results of motion estimation based on the piecewise smooth model will be presented. Introduction of another layer in the motion model requires

specification of the weighting coefficient λ_l. The selection of λ_l is not trivial, especially that in this formulation it must globally describe the trade-off between the displacement and line models. λ_l defines a threshold between variations of vectors belonging to the same patch of vectors and variations of vectors taken from different patches (across a motion boundary). It means that λ_{d} and λ_l should satisfy the following inequality

$$\lambda_{\mathrm{d}} \cdot V_{\mathrm{d}}^{+} < \lambda_l \cdot V_l^{\bullet} < \lambda_{\mathrm{d}} \cdot V_{\mathrm{d}}^{-},$$

where V_{d}^{+} denotes the maximum value of the displacement potential for two vectors considered to belong to the same patch and V_{d}^{-} is the minimum value of this potential for vectors from two different patches (across a motion boundary). $V_l^{\bullet}$ denotes the minimal value of line potential associated with the introduction of one line element, and for penalties proposed in Fig. 6.3 it is equal to 0.4 for an in-line element. Setting the threshold according to this rule will give the required balance between the line field "activity" and the displacement field smoothness for a deterministic algorithm. Since at this point I have no better way of estimating the value of λ_l I will use this rule for stochastic algorithms as well.

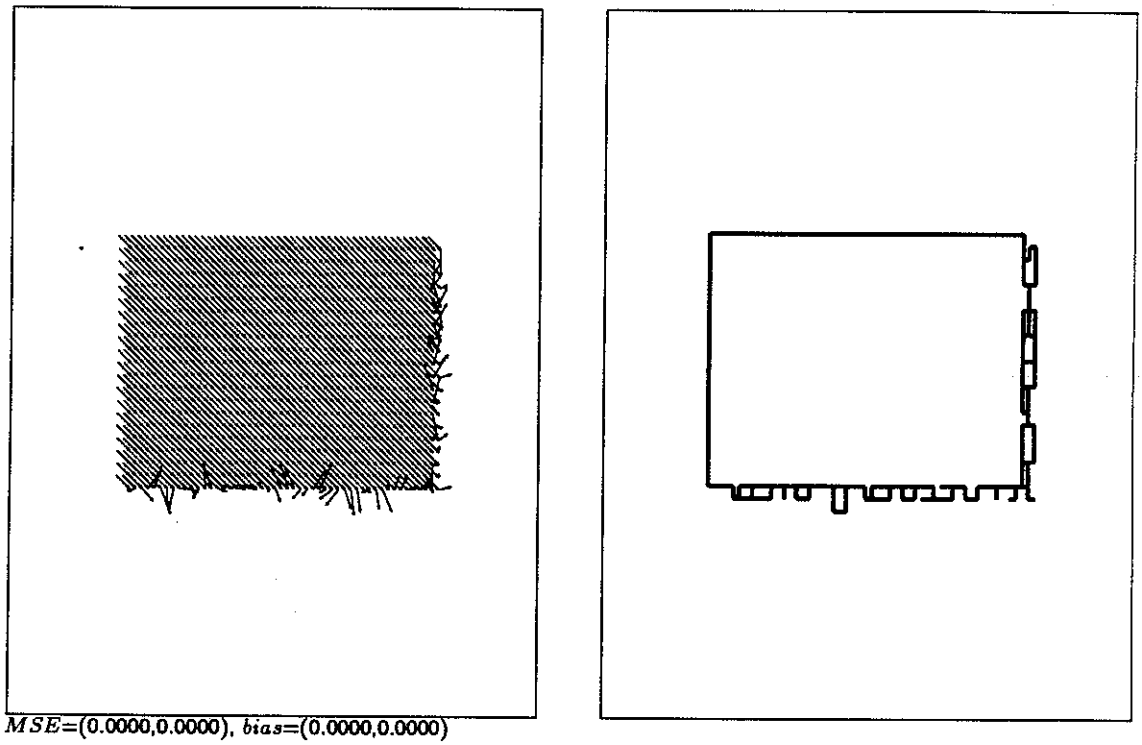
In order to perform estimation of motion discontinuities at least an approximate estimate of the motion field must be known. Hence, the line process is turned on only after a few dozen of iterations. The number of iterations has been chosen in all the experiments in such a way that the initial temperature for the line process is around 0.5.

In order to save space, whenever a motion field is sparse enough† the line elements will be displayed on the same graph.

6.7.1 Results for test image 1

As already demonstrated in Section 4.8.1 the test image 1 contains a very strong motion cue. This is further confirmed here by relative insensitivity of the algorithm to the choice of λ_l. Since the displacement vectors for the test image 1 are (2.0,1.0), it is reasonable to assume that $V_{\mathrm{d}}^{+}= 0.25^2+0.25^2$ (two vectors with horizontal and vertical components differing by at most 0.25 pel) and $V_{\mathrm{d}}^{-}= 1.0^2+1.0^2$. Consequently λ_l should be chosen

† Recall that for test images 3 and 4 the displayed motion fields are subsampled by 2 in each direction.



(a) displ. field *(b)* line field

Fig. 6.4 Discrete state-space MAP estimates with piecewise smooth motion model: test image 1, λ_d/λ_g= 0.05, λ_l/λ_d= 1.2, neighb. $\mathcal{N}_d^1$, bilinear interp., exponential schedule, T_0=1.0, a=0.9866, 400 iter.

from the interval $(0.3125{\cdot}\lambda_d, 5.0{\cdot}\lambda_d)$. In Fig. 6.4 the displacement and line fields are shown for the same set of parameters as used in estimation without the line model (Fig. 4.11.c), except for a longer annealing schedule of 400 iterations with a=0.9866. The line process was introduced after 60 iterations. Also $\lambda_l = 1.2{\cdot}\lambda_d = 0.06$, corresponding to a threshold equivalent to about 0.5 pel difference in vector components, was used. The line model was specified by the potentials discussed in Section 6.3.2, except for the potential V_{l_1} which is omitted (α=0.0) since intensity values lack spatial correlation in this image.

After 300 iterations, for which a=0.9866 was intended, the result was very good, but the total energy was still declining. This was confirmed by a few isolated line elements. Another 100 iterations brought the temperature down to 0.0046, and to the estimate demonstrated in Fig. 6.4. Due to the very strong motion cue in the data, the improvement, compared to the motion field estimate from Fig. 4.11.c, is rather minor: a few vectors close to the top and left motion boundaries have disappeared. The mean squared error and the

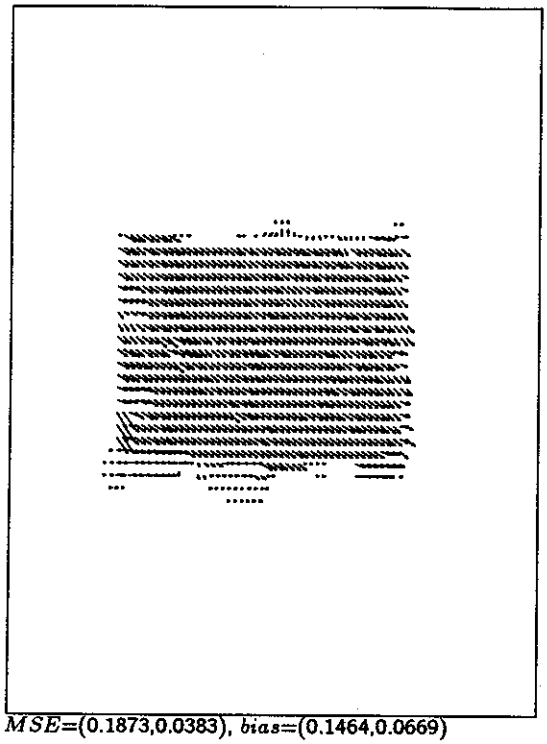
bias, however, have been reduced to zero ! Within the rectangle the estimate is exactly equal to the true motion ! Note that the well defined motion boundary along the top and left edges of the rectangle is correctly estimated by the algorithm. The other two edges corresponding to the occlusion border, where the motion boundary is not well defined†, 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.

The results for λ_l=0.3125·$\lambda_{\mathbf{d}}$ and λ_l=5.0·$\lambda_{\mathbf{d}}$ were very similar to the estimate from Fig. 6.4. They differed in the occlusion areas only. For λ_l=0.3125·$\lambda_{\mathbf{d}}$, which also required slightly longer annealing schedule, the partitioning of those areas was finer, while for λ_l=5.0·$\lambda_{\mathbf{d}}$ it was coarser.

6.7.2 Results for test image 2

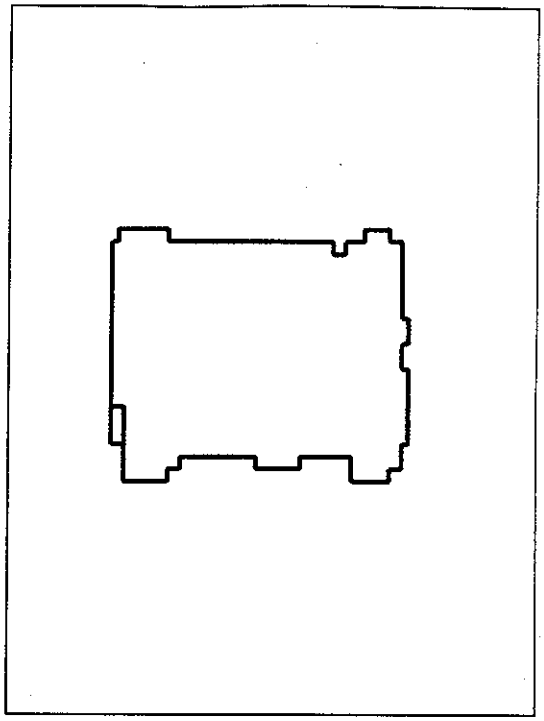
Figs. 6.5.a,.b show the displacement and line fields obtained by the discrete state-space MAP estimation with the same parameters as those used in estimation without the line model (Fig. 4.13.c), except for the length of the annealing schedule. An extended schedule of 300 iterations, a=0.9866 as well as $\lambda_l/\lambda_{\mathbf{d}}$= 1.2 were used, and the line process was introduced after 60 iterations. Note that line elements surround the moving rectangle and isolate the motion vectors inside from the stationary background. This results in a motion field that is slightly smoother inside the rectangle, especially close to its boundaries, and consequently the image energy is driven down by about 10%. Positioning of numerous line elements is incorrect, however, and the mean squared error is increased compared to the result with no line elements. Figs. 6.5.c,.d show an estimate after adding the potential V_{l_1} defined over single-element cliques. The constant α was chosen to be 10.0 to discourage formation of line elements in the areas of uniform intensity. Since higher α indirectly increases the overall line energy, the ratio $\lambda_l/\lambda_{\mathbf{d}}$ was reduced to 0.8. By enforcing the gradient penalty, the mean squared error was driven below that from Fig. 4.13.c, especially

† 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.

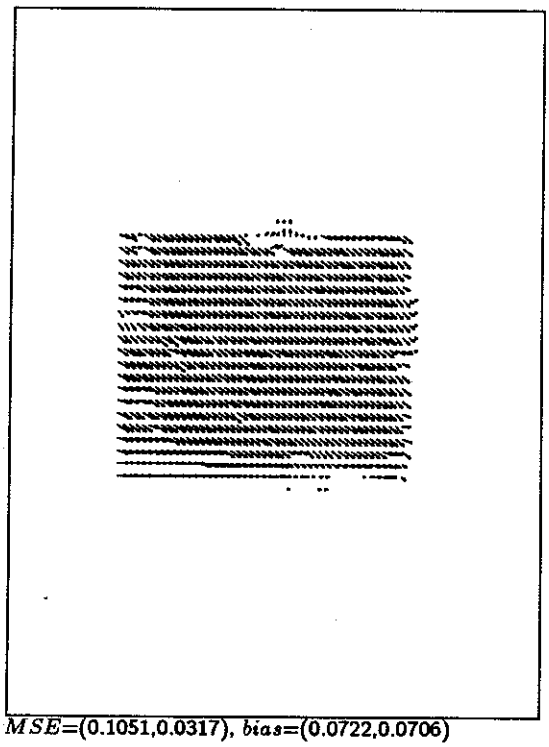


MSE=(0.1873,0.0383), $bias$=(0.1464,0.0669)

(a) $\lambda_l/\lambda_{\mathbf{d}}$=1.2, α=0.0

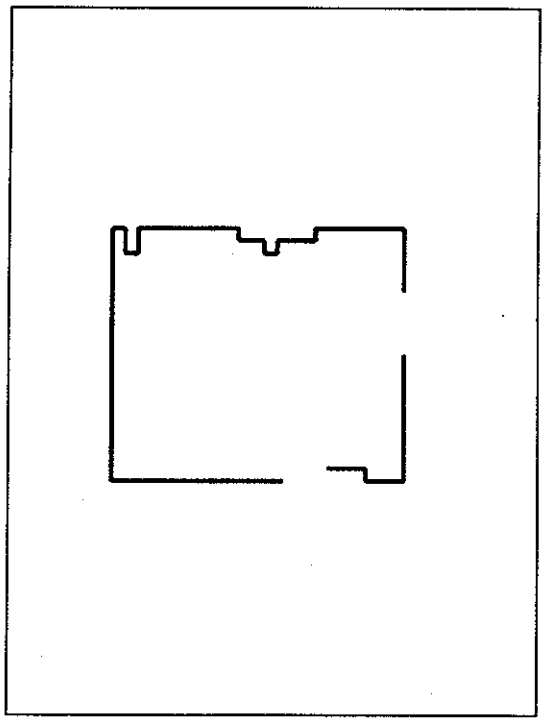


(b) $\lambda_l/\lambda_{\mathbf{d}}$=1.2, α=0.0



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

(c) $\lambda_l/\lambda_{\mathbf{d}}$=0.8, α=10.0



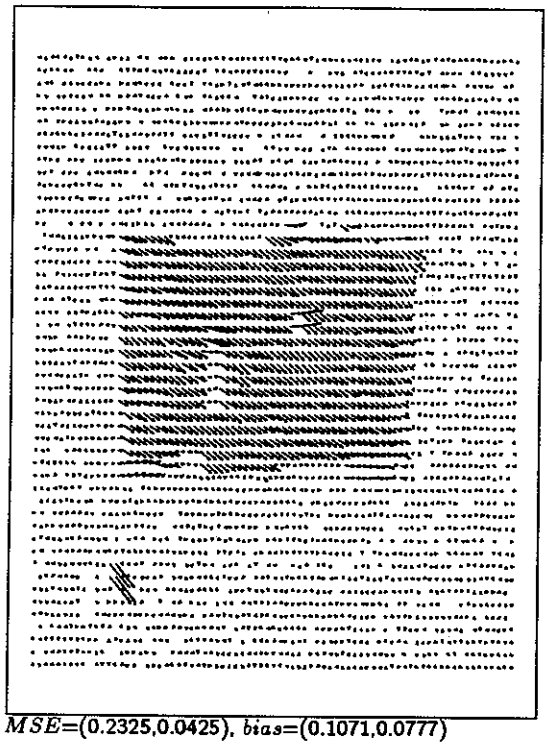
(d) $\lambda_l/\lambda_{\mathbf{d}}$=0.8, α=10.0

Fig. 6.5 Discrete state-space MAP estimates with piecewise smooth motion model: test image 2, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}_0$=1.0, a=0.9866, 300 iter.

for the horizontal component, and also the image energy was reduced by about 15%. This result is exactly what was expected from the piecewise smooth displacement model: to reduce the image energy without sacrificing motion smoothness.

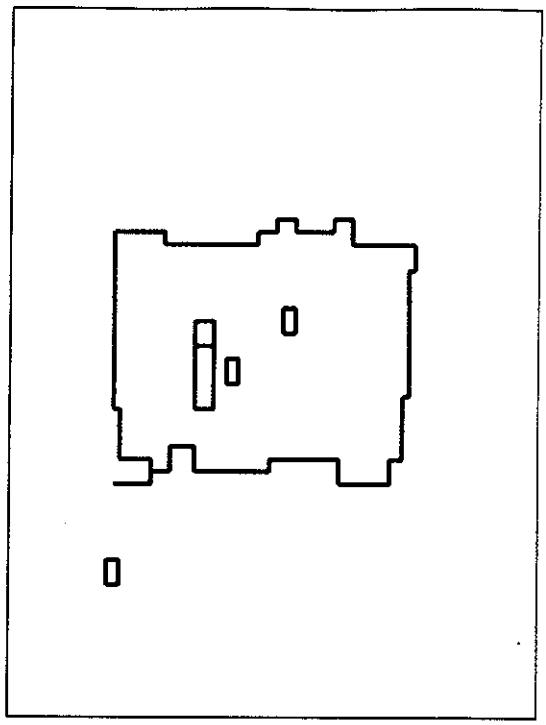
Similar results for the continuous state-space MAP estimation without and with the line model are shown in Fig. 6.6. The same parameters as above are used, except for the annealing schedule. As in the estimation from Fig. 4.14.d, $\mathbf{T}_0$=5.0, a=0.9944 and 1000 iterations are applied, and the line process is introduced after 400 iterations. Again, the mean squared error is increased (compared to the result from Fig. 4.14.d) if no gradient penalty is introduced, but it is reduced slightly after introduction of such a penalty. Note that some line elements are missing from the rectangle boundary. This effect is caused by a small intensity gradient in some parts of the rectangle contour. In both cases the image energy is reduced by over 20%, however the misplaced and ill-shaped discontinuity boundaries in Fig. 6.6.b cause substantial increase in the line energy.

Fig. 6.7 shows the continuous state-space MAP estimate with the piecewise smooth motion model over a hierarchy of resolutions. The same parameters have been used as for the estimate from Fig. 5.7. Since the displacement is larger (T_g=6τ_{60}), the ratio $\lambda_l/\lambda_{\mathrm{d}}$ has been increased to 1.8. The line process is turned on after (100,150,200) iterations, at respective resolution levels, to maintain similar initial temperature at each level. Since to generate the image pyramid low-pass filtering has been applied, the images for κ=2,1 contain much less spatial information. Taking this into account the constant α in computation of the single-element clique penalty has been reduced at subsequent resolution levels as follows: (10.0,3.0,1.0). Limited adjustment of these values did not incurr disastrous effects. Note that the discontinuity estimates at levels κ=2 and 1, despite lack of precision, still limit the spread of smoothness across the rectangle edge. The precision in positioning of line elements is improved at the full resolution level where, except for one area, it is exact. This imprecision is due to the image breakup in the left top corner of the rectangle, already discussed in Section 5.5.2. Observe a dramatic drop in the mean squared error compared to the corresponding hierarchical estimate not using the line process from Fig. 5.7. The estimate from Fig. 6.7.c is very close to the true motion, also visually. This

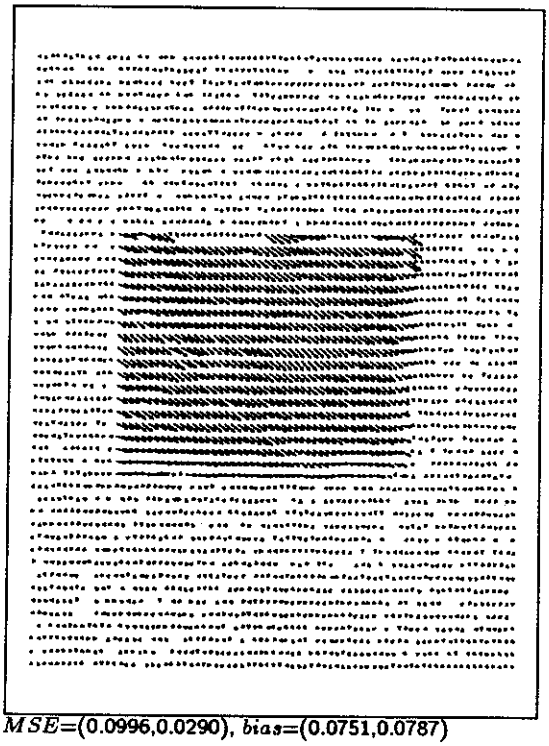


MSE=(0.2325,0.0425), $bias$=(0.1071,0.0777)

(a) $\lambda_l/\lambda_{\mathbf{d}}$=1.2, α=0.0

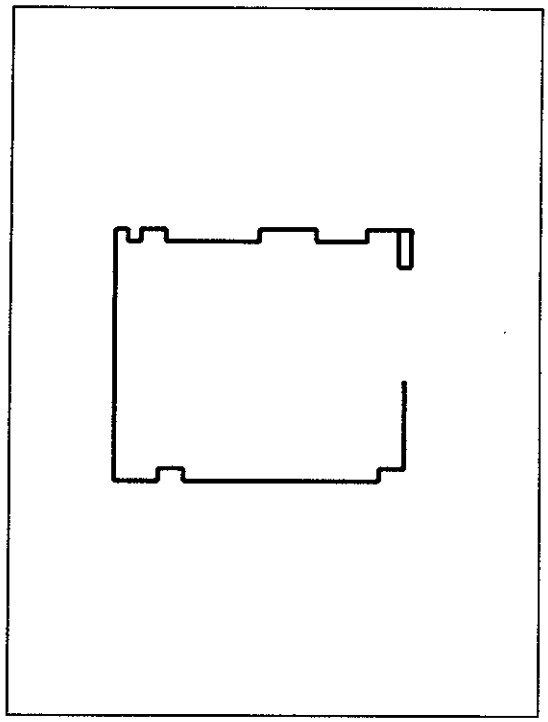


(b) $\lambda_l/\lambda_{\mathbf{d}}$=1.2, α=0.0



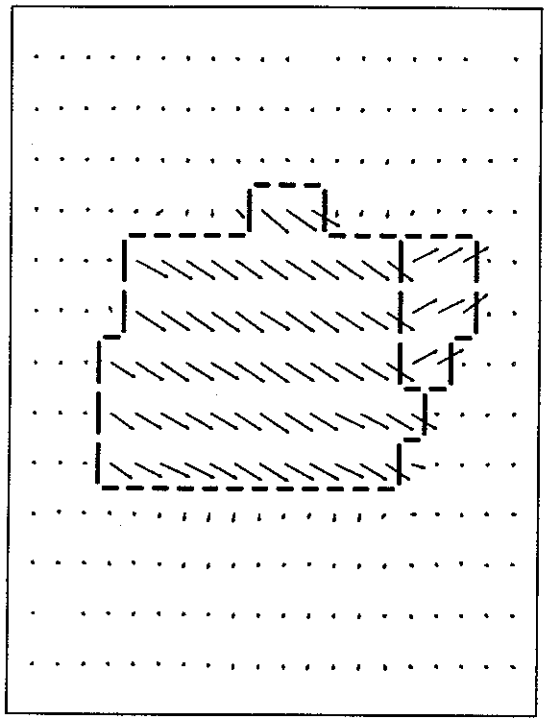
MSE=(0.0996,0.0290), $bias$=(0.0751,0.0787)

(c) $\lambda_l/\lambda_{\mathbf{d}}$=0.8, α=10.0

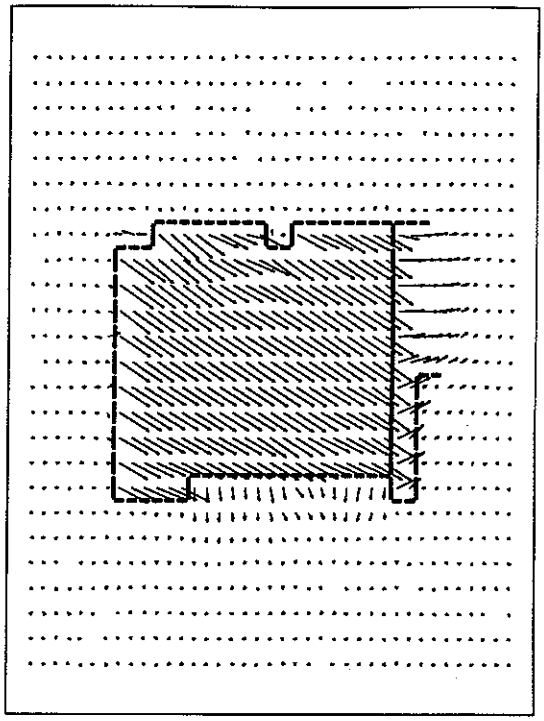


(d) $\lambda_l/\lambda_{\mathbf{d}}$=0.8, α=10.0

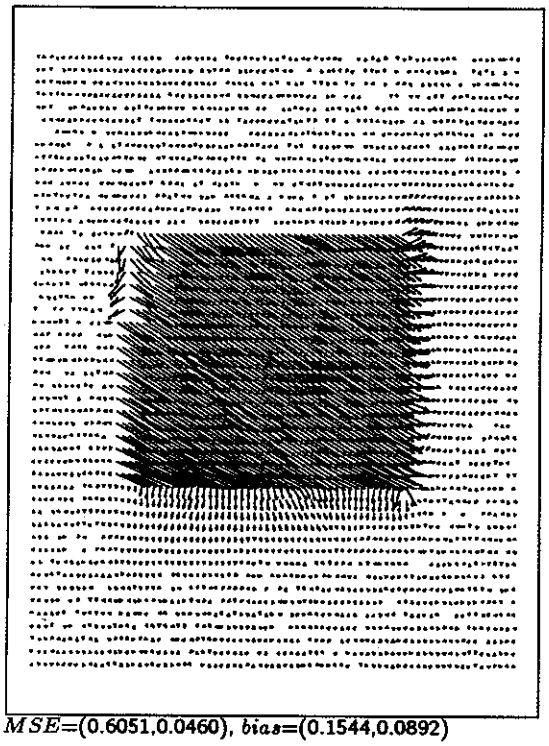
Fig. 6.6 Continuous state-space MAP estimates with piecewise smooth motion model: test image 2, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter.



(a) displ. + line, κ=2

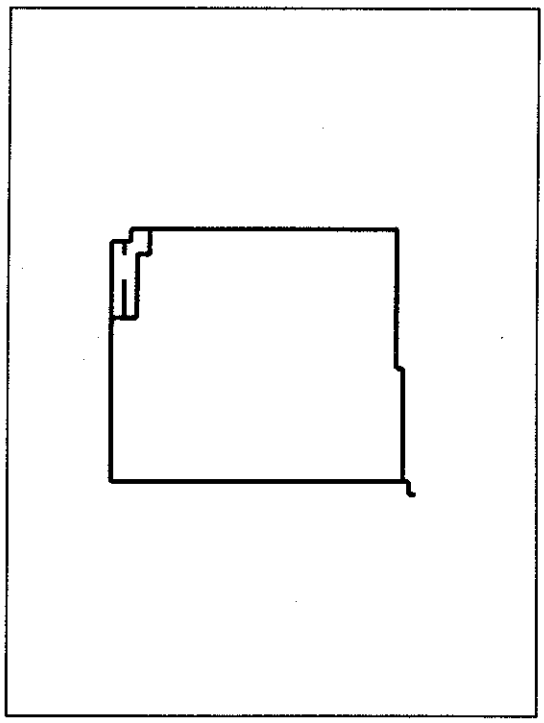


(b) displ. + line, κ=1



MSE=(0.6051,0.0460), $bias$=(0.1544,0.0892)

(c) displ., κ=0



(d) line, κ=0

Fig. 6.7 Hierarchical continuous state-space MAP estimates with piecewise smooth motion model: test image 2, K_l=3, $\lambda_{\mathrm{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathrm{d}}$= 1.8, α=10.0, neighb. $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interp., exponential schedule, $\mathbf{T}_0$=(1.0,2.0,4.0), ω=0.992, 500 iter. at each level.

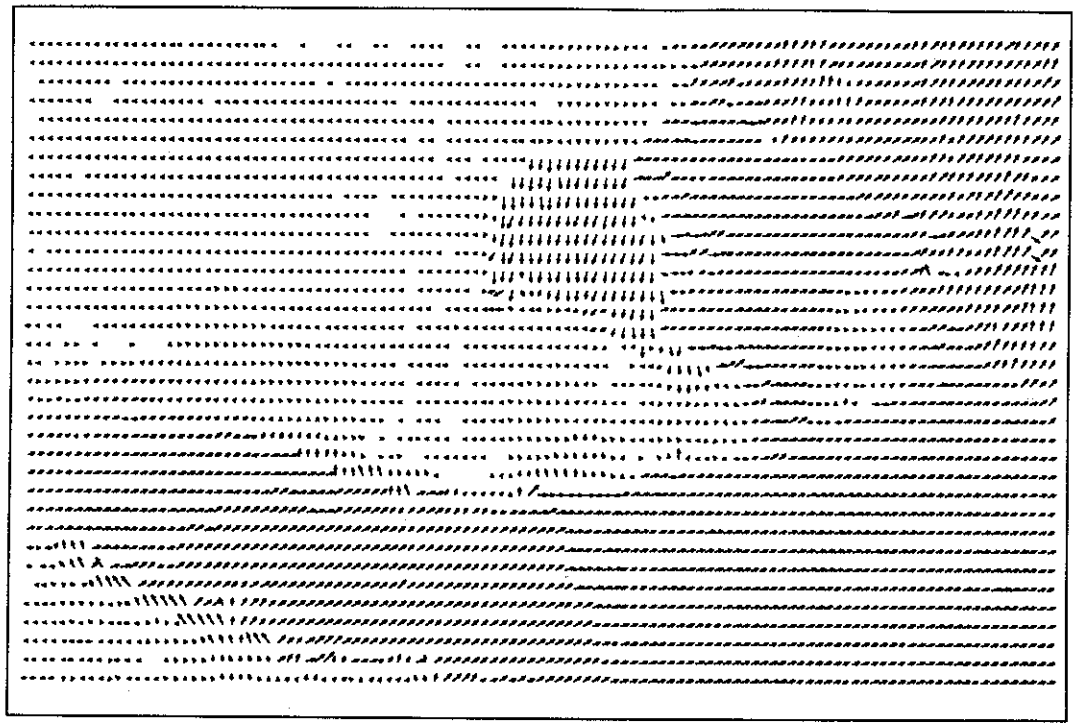
is furthermore confirmed by a 50% reduction of the image energy, again compared to the result from Fig. 5.7.

6.7.3 Results for test images 3 and 4

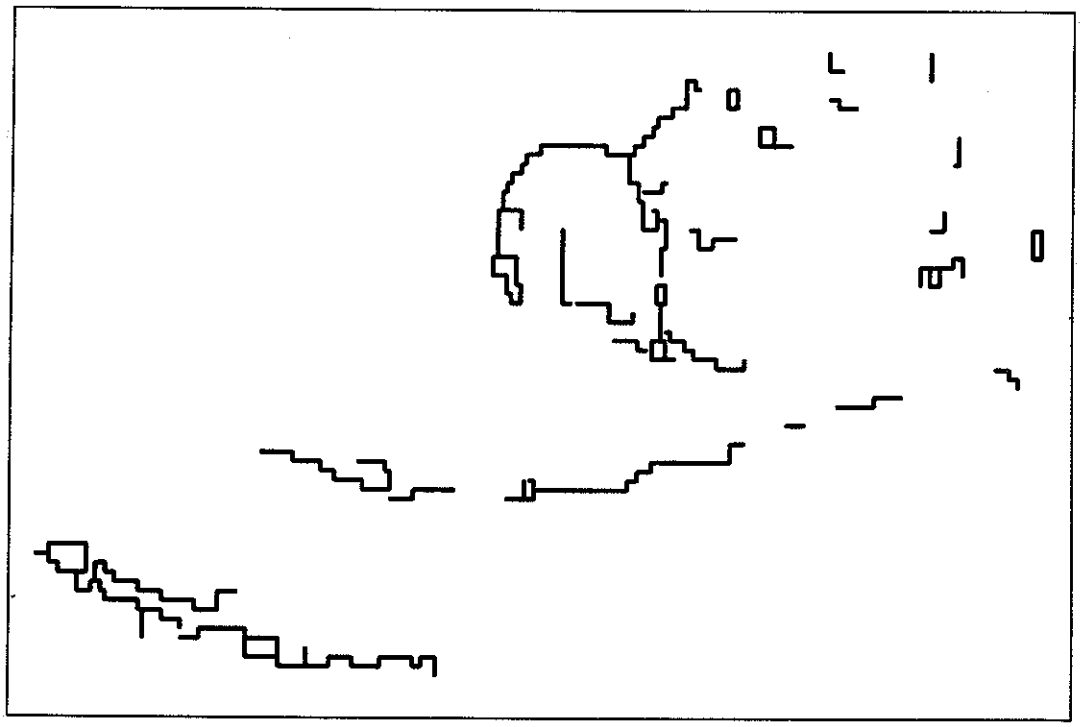
Except for the bilinear interpolation, the same parameters as used in the previous section have been applied to the test images 3 and 4. Fig. 6.8 shows the displacement and line fields obtained through the discrete state-space MAP estimation with the gradient penalty. Observe the well identified motion boundaries of the moving palm of the hand, of the face and of parts of the arm. Large fragments of motion boundaries are missing, probably due to insufficient intensity gradient there, however motion boundaries coinciding with substantial intensity gradients are easily detected. As in the case of test image 2, the image energy is reduced by about 5% compared to the result without the line model (Fig. 4.16.a) but the displacement field is at least as smooth as in Fig. 4.16.a.

Fig. 6.9 shows the continuous state-space MAP estimate with the same parameters. The contours formed by line elements are even more smooth in this case. Note the well identified borders of the moving hand, face and arm. The image energy is also substantially reduced while the motion field remains smooth. In some other experiments I have also applied the MAP estimation without the gradient penalty. In both cases (discrete and continuous) the motion boundaries were misplaced and ill-positioned. The discrete state-space estimate performed only slightly poorer in terms of the energies (compared to the estimate with the gradient penalty), but the continuous state-space result produced much lower image energy at a cost of boosted line energy. Overall, it performed significantly poorer than the estimate with the gradient penalty. Consequently, only the model with potential V_{l_1} will be used.

To demonstrate the impact of line process on the smoothness of motion field in the test image 3, Fig. 6.10 shows windows of 60 by 30 vectors taken from the central parts of motion fields presented in Fig. 4.16.b and Fig. 6.9.a. The two estimates differ only by the motion model, while other parameters are identical. Note the sharp transitions between the palm of the hand and the background or the face in Fig. 6.10.b. This transition is responsible for

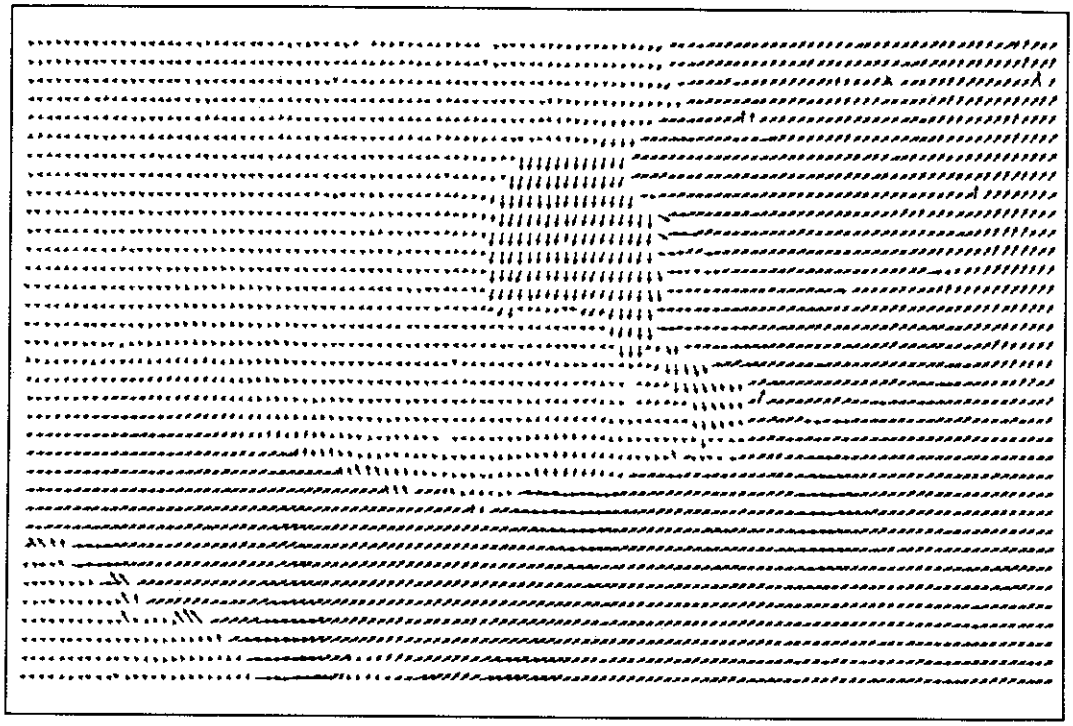


(a) displacement field

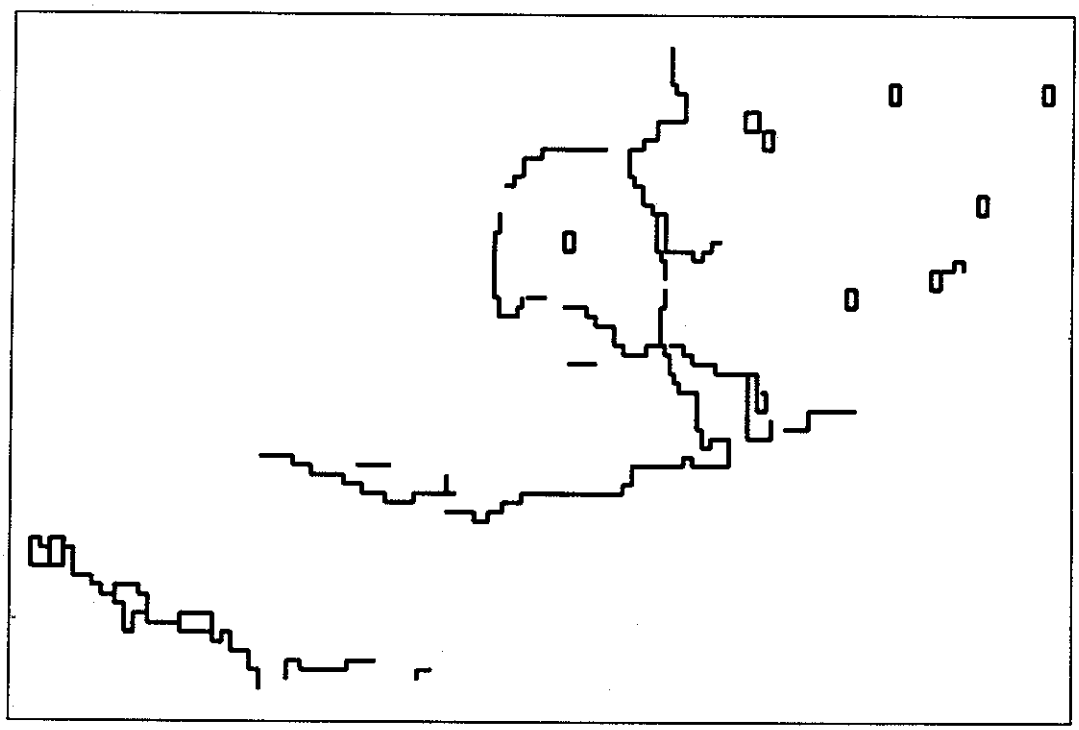


(b) line field

Fig. 6.8 Discrete state-space MAP estimates with piecewise smooth motion model: test image 3, $\lambda_{\mathbf{d}}/\lambda_g = 20.0$, $\lambda_l/\lambda_{\mathbf{d}} = 0.8$, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bilinear interp., exponential schedule, $\mathbf{T}_0$=1.0, a=0.9866, 300 iter.

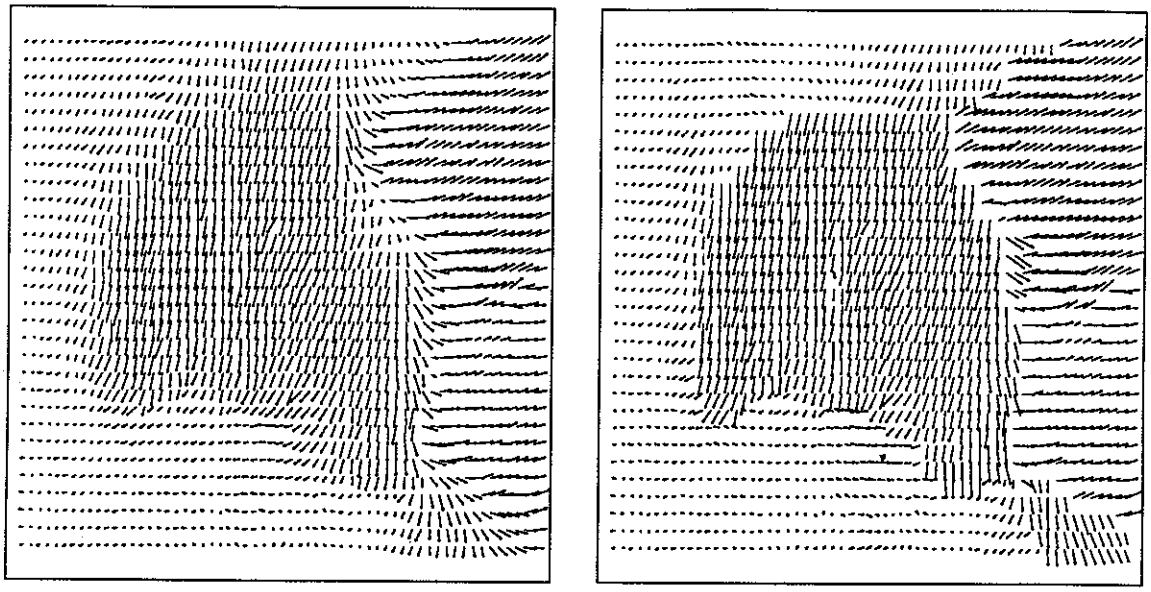


(a) displacement field



(b) line field

Fig. 6.9 Continuous state-space MAP estimates with piecewise smooth motion model: test image 3, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathbf{d}}$= 0.8, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bicubic interp., exponential schedule, T_0=5.0, a=0.9944, 1000 iter.



(a) globally smooth model *(b)* piecewise smooth model

Fig. 6.10 Central parts of continuous state-space MAP estimates without and with piecewise smooth motion model from Figs. 4.16.b and 6.9.a, respectively (no spatial subsampling of vectors is applied).

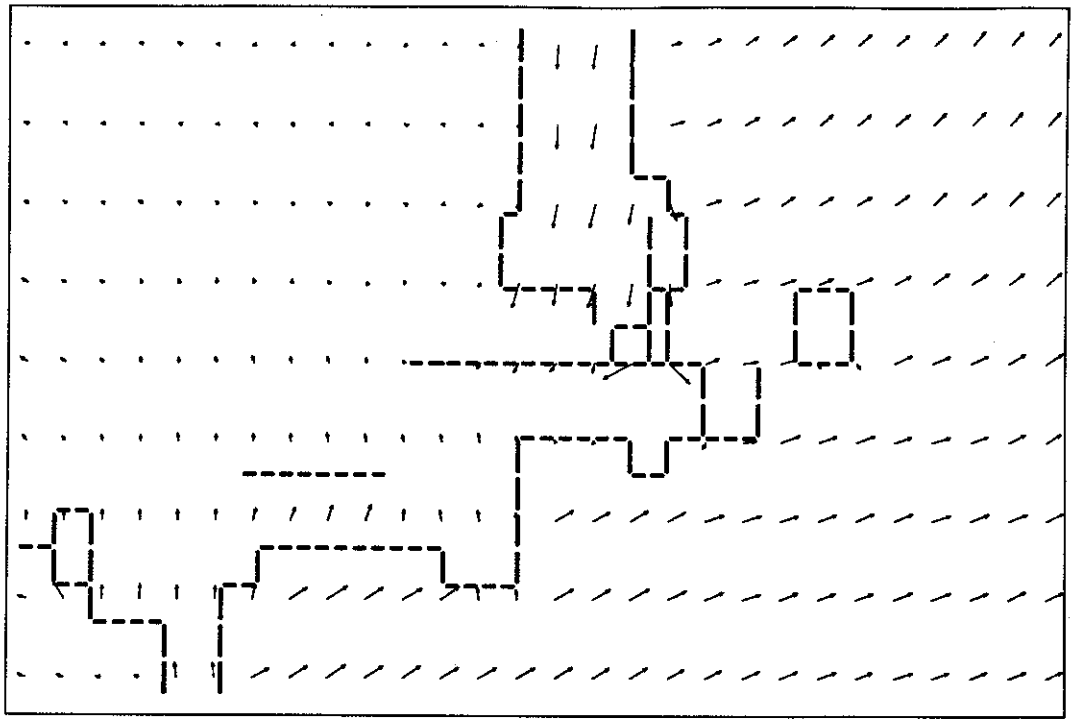
more precise motion portrayal and significant reduction of image energy without affecting smoothness of the motion field within moving objects.

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 $\lambda_{\mathbf{d}}/\lambda_g = 100.0$ for test image 3, confirming this observation, can be found in [57].

Figs. 6.11, 6.12 show the continuous state-space MAP estimates with the piecewise smooth motion model over a hierarchy of resolutions. The same parameters have been used as for the estimates from Figs. 5.8.b, 5.9.b, 5.10.b. Since the displacement is somewhat larger due to the temporal distance between images $T_g = 4\tau_{60}$, the ratio $\lambda_l/\lambda_{\mathbf{d}}$ has been increased from 0.8 to 1.0. The line process is turned on after (100,150,200) iterations to maintain similar initial temperature at each level. Note that the discontinuity estimates at levels $\kappa = 2$ 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 fact

that the motion field is as smooth as the one without the discontinuities model, but the image energy is reduced by over 25%.

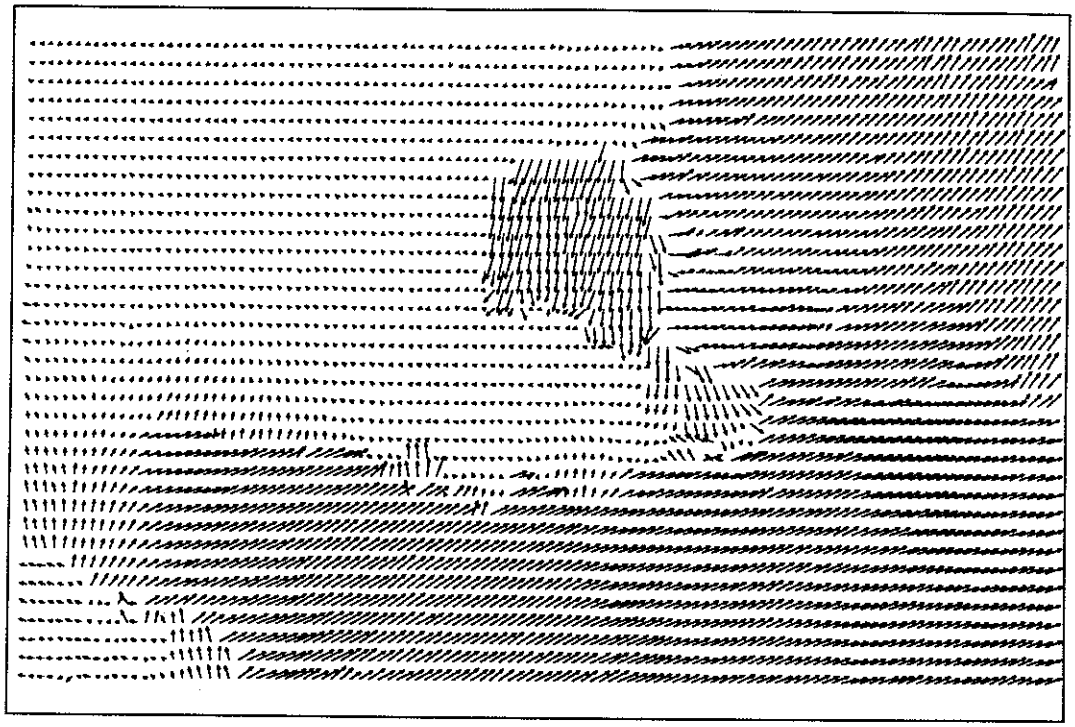
The same model and estimation parameters have been applied to the test image 4 and shown in Figs. 6.13, 6.14, however as expected it failed to produce as good results as in the case of other test images. The reasons are twofold. Firstly, the piecewise smooth motion model will work well only with images where moving objects generate well-defined motion boundaries. In the case of test image 4, however, the motion is primarily rotational, thus it does not contain well-defined motion boundaries. Secondly, as already reported, the illumination effects are not negligible because of the hair shadows on the forehead. The image energy was reduced significantly, but numerous line elements are not positioned on real motion boundaries.



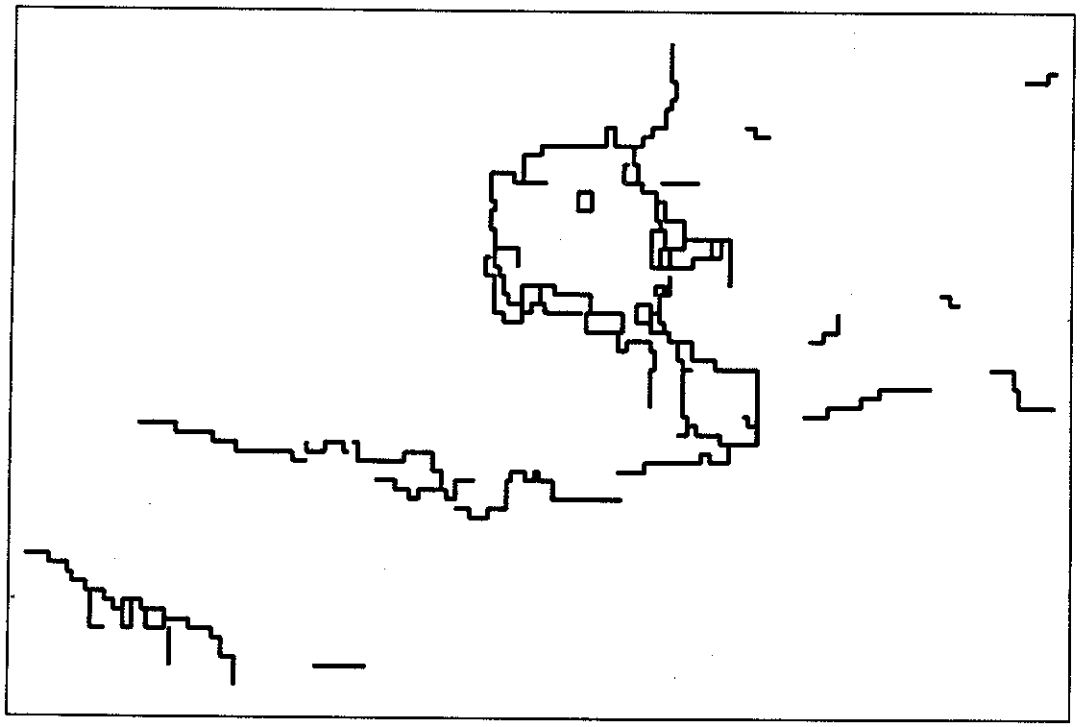
(a) displ. + line, κ=2

(b) displ. + line, κ=1

Fig. 6.11 Hierarchical continuous state-space MAP estimates with piecewise smooth motion model: test image 3, K_l=3, κ=2,1, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathbf{d}}$= 1.0, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bicubic interp., exponential schedule, T_0=(1.0,2.0,4.0), a=0.9944, 500 iter. at each level.

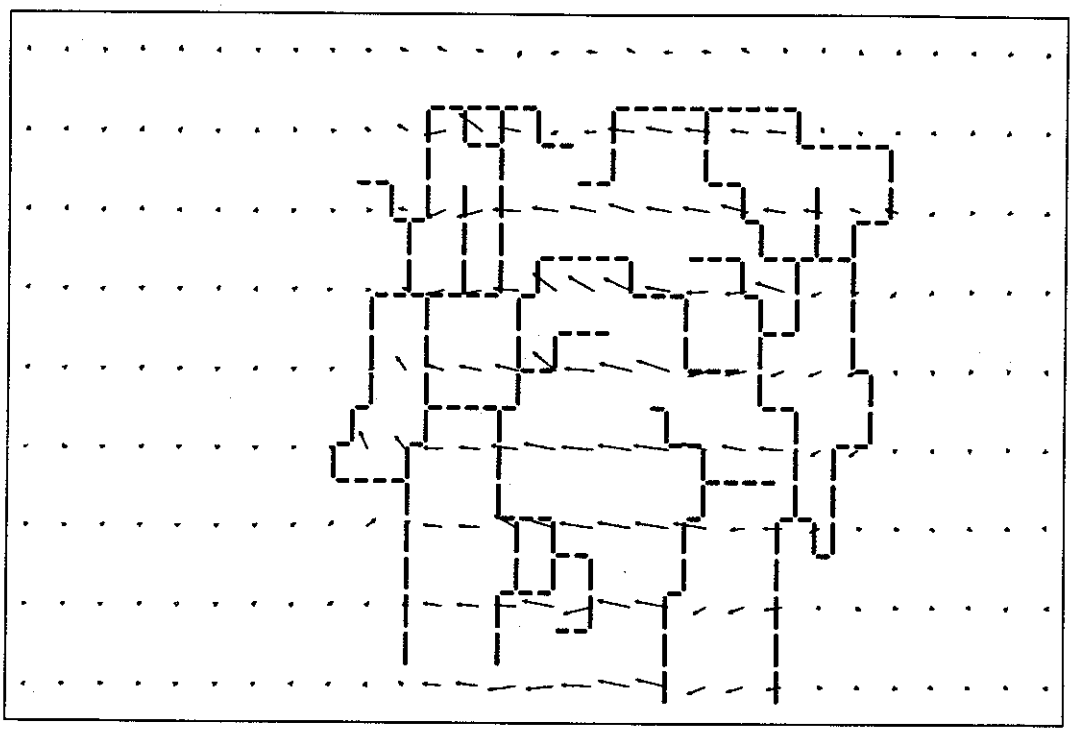


(a) displacement field, κ=0

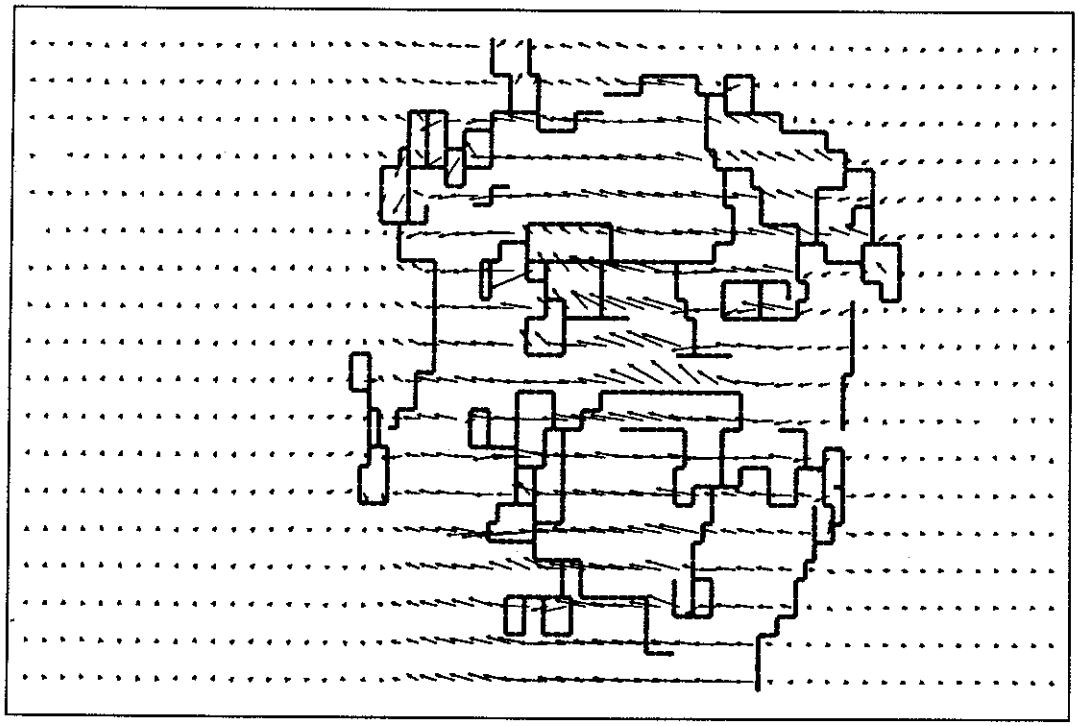


(b) line field, κ=0

Fig. 6.12 Hierarchical continuous state-space MAP estimates with piecewise smooth motion model: test image 3, K_l=3, κ=0, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathbf{d}}$= 1.0, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bicubic interp., exponential schedule, $\mathbf{T}_0$=(1.0,2.0,4.0), a=0.9944, 500 iter. at each level.

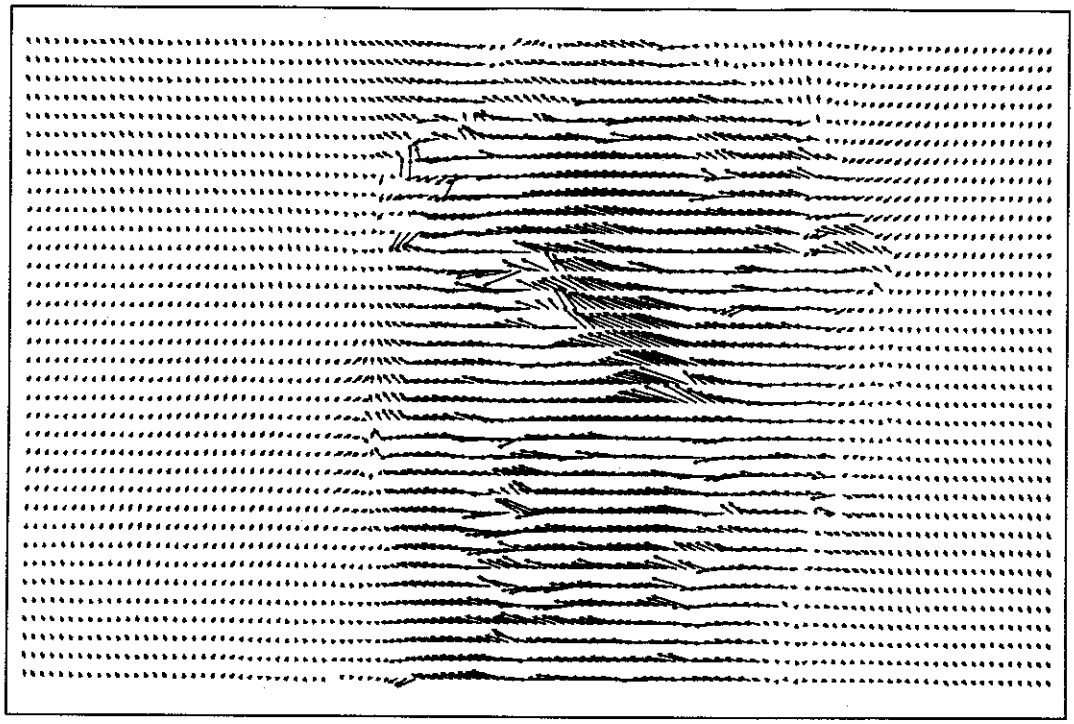


(a) displ. + line, κ=2

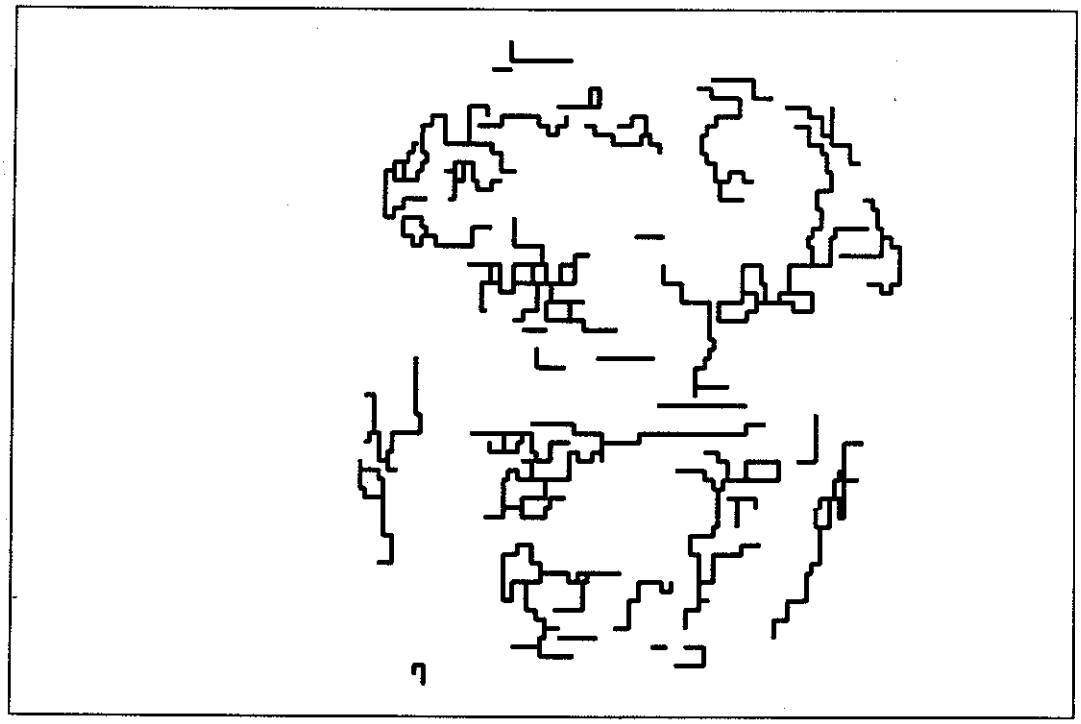


(b) displ. + line, κ=1

Fig. 6.13 Hierarchical continuous state-space MAP estimates with piecewise smooth motion model: test image 4, K_l=3, κ=2,1, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathbf{d}}$= 1.0, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bicubic interp., exponential schedule, $\mathbf{T}_0$=(1.0,2.0,4.0), a=0.9944, 500 iter. at each level.



(a) displacement field, κ=0



(b) line field, κ=0

Fig. 6.14 Hierarchical continuous state-space MAP estimates with piecewise smooth motion model: test image 4, K_l=3, κ=0, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathbf{d}}$= 1.0, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bicubic interp., exponential schedule, $\mathbf{T}_0$=(1.0,2.0,4.0), a=0.9944, 500 iter. at each level.

Chapter 7

COLOUR CUE IN MOTION ESTIMATION

The algorithms presented so far have used only the image luminance to compute motion fields. There exist, however, other cues which may be helpful in recovering 2-D motion. For example, Wohn *et al.* [90] proposed a multiconstraint method incorporating luminance as well as directional derivatives of that luminance to compute motion. They have also suggested using colour information (R-G-B signal) for motion recovery, however failed to give any details or show experimental results. Mitiche *et al.* [66] proposed a multiconstraint method which combines the luminance with multispectral functions (e.g., R-G-B signal) or with the output of spatial operators applied to the luminance (e.g., contrast, entropy) to compute optical flow, but also did not provide examples for colour images.

In this chapter I will discuss the extension of MAP estimation to include the colour cue. Some results of application of this method to images in Y-C1-C2 format will be presented.

7.1 INCORPORATING COLOUR INTO THE A POSTERIORI PROBABILITY

Up to this point the images which I have used for computation of motion contained luminance only. Scenes captured by a camera, however, are colorful and with a suitable imaging system these colours can be registered too. Depending on the type of system, R-G-B (red, green, blue components) or Y-C1-C2 (luminance and two chrominance components) format can be used to represent colour images. The R-G-B format contains three basic colours which are sampled with the same rate, while Y-C1-C2 contains chrominance components C1, C2 subsampled horizontally by 2 compared to the luminance Y. Conse-

quently the Y-C1-C2 format requires 2/3 of the storage necessary for the R-G-B format but at a cost of reduced chrominance resolution.

Does colour, apart from the pleasing effect, provide any useful perceptual information ? This question has been tackled in many ways, here however I am interested in this question in the context of motion perception. A simple experiment with switching off the chrominance on a colour monitor will show how much information is lost from an image. For example, certain contours and edges may disappear since the same intensity can have different hues. Also texture may look more pronounced in colour than in black-and-white. With this extra information from the colour cue one may hope for possible improvements in motion computation by applying an algorithm to all three components.

To generalize the subsequent derivation denote the three components of an image by g_1, g_2 and g_3. Recall the *a posteriori* probability (3.3) from Chapter 3. The likelihood $P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \mathbf{d}_t, G_{t_-} = g_{t_-})$ expresses the luminance matching between two images captured at instants t_- and t_+. Assuming that colours move coherently with luminance, the same matching should apply to both chrominances. Let $\mathbf{G}$ denote a 3-D vector random field, and let $\mathbf{g}$ be a sample field from $\mathbf{G}$ defined over the lattice Λ_g. Then, $\mathbf{g}$ can be understood as a sample image with three components (a 3-D vector) at each spatio-temporal position i.e., $\mathbf{g}(\mathbf{x}_i, t) = [g_1(\mathbf{x}_i, t), g_2(\mathbf{x}_i, t), g(\mathbf{x}_i, t)]$. Rewrite the *a posteriori* probability (3.3) as follows:

$$P(\mathbf{D}_t = \hat{\mathbf{d}}_t | \mathbf{G}_{t_-} = \mathbf{g}_{t_-}, \mathbf{G}_{t_+} = \mathbf{g}_{t_+}) = \frac{P(\mathbf{G}_{t_+} = \mathbf{g}_{t_+} | \mathbf{D}_t = \hat{\mathbf{d}}_t, \mathbf{G}_{t_-} = \mathbf{g}_{t_-}) \cdot P(\mathbf{D}_t = \hat{\mathbf{d}}_t | \mathbf{G}_{t_-} = \mathbf{g}_{t_-})}{P(\mathbf{G}_{t_+} = \mathbf{g}_{t_+} | \mathbf{G}_{t_-} = \mathbf{g}_{t_-})}. \quad (7.1)$$

Note that now the likelihood $P(\mathbf{G}_{t_+} = \mathbf{g}_{t_+} | \mathbf{D}_t = \mathbf{d}_t, \mathbf{G}_{t_-} = \mathbf{g}_{t_-})$ expresses the probability of obtaining the three-component signal $[g_1, g_2, g_3]_{t_+}$ from displacement $\hat{\mathbf{d}}(\mathbf{x}_i, t)$ and $[g_1, g_2, g_3]_{t_-}$. Define the 3-D displaced pel difference vector $\tilde{\mathbf{r}}$ as follows

$$\tilde{\mathbf{r}}(\mathbf{d}(\mathbf{x}_i, t), \mathbf{x}_i, t, \Delta t) = [\tilde{r}_1(\mathbf{d}(\mathbf{x}_i, t), \mathbf{x}_i, t, \Delta t), \tilde{r}_2(\mathbf{d}(\mathbf{x}_i, t), \mathbf{x}_i, t, \Delta t), \tilde{r}_3(\mathbf{d}(\mathbf{x}_i, t), \mathbf{x}_i, t, \Delta t)].$$

If the same line of reasoning as for the luminance-only DPD model (3.4.2) is applied, then

for the true motion field $\boldsymbol{d}$ the components of $\tilde{\mathbf{r}}$ are Gaussian RVs defined as follows

$$\begin{aligned}\tilde{\boldsymbol{r}}_j(\boldsymbol{d}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t) =& g_j(\mathbf{x}_i+(1.0-\Delta t)\cdot\boldsymbol{d}(\mathbf{x}_i,t),t_+)- \\ & g_j(\mathbf{x}_i-\Delta t\cdot\boldsymbol{d}(\mathbf{x}_i,t),t_-) \\ =& n_j(\mathbf{x}_i,t), \quad j=1,2,3, \text{ all } i.\end{aligned}$$

The $n_j(\mathbf{x}_i,t)$'s have been assumed independent for each i given j. It also seems reasonable to assume that they are independent for different j's, for example $n_1(\mathbf{x}_i,t)$ independent of $n_3(\mathbf{x}_i,t)$ for any i. Let σ_j^2 be the variance of Gaussian noise n_j for component g_j. Consequently the likelihood $P(\mathbf{G}_{t_+}=\mathbf{g}_{t_+}|\mathbf{D}_t=\mathbf{d}_t,\mathbf{G}_{t_-}=\mathbf{g}_{t_-})$ can be expressed as the following product of Gaussian distributions:

$$\begin{aligned}P(\mathbf{G}_{t_+}=\mathbf{g}_{t_+}|\mathbf{D}_t=\hat{\mathbf{d}}_t,\mathbf{G}_{t_-}=\mathbf{g}_{t_-}) = \prod_{j=1}^{3}\prod_{i=1}^{M_\mathbf{d}} p_{n_j}(\tilde{\mathbf{r}}(\hat{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)) = \\ (2\pi\sigma_1^2\sigma_2^2\sigma_3^2)^{-M_\mathbf{d}/2}\cdot e^{-U_\mathbf{g}(\mathbf{g}_{t_+}|\hat{\mathbf{d}}_t,\mathbf{g}_{t_-})},\end{aligned} \tag{7.2}$$

where the energy $U_\mathbf{g}$ is defined as follows:

$$U_\mathbf{g}(\mathbf{g}_{t_+}|\hat{\mathbf{d}}_t,\mathbf{g}_{t_-}) = \sum_{j=1}^{3}\frac{1}{2\sigma_j^2}\sum_{i=1}^{M_\mathbf{d}}[\tilde{\boldsymbol{r}}_j(\hat{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2. \tag{7.3}$$

Having defined the likelihood (7.2), the joint *a posteriori* probability can now be written in the Gibbs form:

$$P(\mathbf{D}_t=\hat{\mathbf{d}}_t|\mathbf{G}_{t_-}=\mathbf{g}_{t_-},\mathbf{G}_{t_+}=\mathbf{g}_{t_+}) = \frac{1}{Z}e^{-U(\hat{\mathbf{d}}_t,\mathbf{g}_{t_-},\mathbf{g}_{t_+})}$$

where Z is again a constant and the energy U is defined as follows:

$$\begin{aligned}U(\hat{\mathbf{d}}_t,\mathbf{g}_{t_-},\mathbf{g}_{t_+}) &= \sum_{j=1}^{3}\lambda_{g_j}\cdot\sum_{i=1}^{M_\mathbf{d}}[\tilde{\boldsymbol{r}}_j(\hat{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2+\lambda_\mathbf{d}\cdot U_\mathbf{d}(\hat{\mathbf{d}}_t) \\ &= \sum_{i=1}^{M_\mathbf{d}}\sum_{j=1}^{3}\lambda_{g_j}\cdot[\tilde{\boldsymbol{r}}_j(\hat{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2+\lambda_\mathbf{d}\cdot U_\mathbf{d}(\hat{\mathbf{d}}_t).\end{aligned} \tag{7.4}$$

The displacement energy $U_\mathbf{d}$ was defined in (3.18), while the constants λ_{g_j} are equal to $1/2\sigma_j^2$. The difference between the above energy function and the one derived in Section 3.5 is that here three components of the image are matched simultaneously instead of one. The weights between those three components are again inversely proportional to the variances of DPD noise models for individual components.

The conditional (marginal) probability driving the Gibbs sampler can be now computed from the total energy (7.4). It has the same form as defined by expression (4.12), except that the sample field g is replaced by the vector sample field $\mathbf{g}$:

$$P(\mathbf{D}(\mathbf{x}_{n_\tau},t)=\widehat{\mathbf{d}}(\mathbf{x}_{n_\tau},t)|\mathbf{D}(\mathbf{x}_j,t)=\widehat{\mathbf{d}}(\mathbf{x}_j,t), j\neq n_\tau, \mathbf{G}_{t_-}=\mathbf{g}_{t_-}, \mathbf{G}_{t_+}=\mathbf{g}_{t_+}) = \frac{e^{-U_{\mathbf{d}}^{n_\tau}(\widehat{\mathbf{d}}(\mathbf{x}_{n_\tau},t)|\widehat{\mathbf{d}},\mathbf{g}_{t_-},\mathbf{g}_{t_+})}}{\sum\limits_{z\in\mathcal{S}'_{\mathbf{d}}} e^{-U_{\mathbf{d}}^{n_\tau}(z|\widehat{\mathbf{d}},\mathbf{g}_{t_-},\mathbf{g}_{t_+})}}. \tag{7.5}$$

The local energy function $U_{\mathbf{d}}^{n_\tau}$ is different, however, since it incorporates matching with respect to three-components:

$$U_{\mathbf{d}}^{n_\tau}(z|\widehat{\mathbf{d}},\mathbf{g}_{t_-},\mathbf{g}_{t_+}) = \sum_{j=1}^{3}\lambda_{g_j}\cdot[\tilde{r}_j(z,\mathbf{x}_{n_\tau},t,\Delta t)]^2+\lambda_{\mathbf{d}}\cdot\sum_{j:\,\mathbf{x}_j\in\eta_{\mathbf{d}}(\mathbf{x}_{n_\tau})} V(z,\widehat{\mathbf{d}}(\mathbf{x}_j,t)). \tag{7.6}$$

Consequently the difference between the Gibbs sampler implemented for the above local energy function and the one described in Section 4.2.3 is in the DPD computation. Instead of one DPD applied only to the luminance component, there are DPDs computed for each component of the image e.g., Y, C1 and C2, and appropriately weighted.

Note that the use of three image components instead of one will not affect the two-layer motion model incorporating discontinuities, except for the potential V_{l_1} (6.8) defined over one-element line cliques. This potential relies on the squared magnitude of appropriate intensity gradient to penalize occurrence of line elements. With a three-component image it seems natural to weight the squared magnitudes of gradients obtained from individual components as follows:

$$(\nabla_\bullet \mathbf{g}_{t_-})^2 = \sum_{j=1}^{3}\nu_j\cdot(\nabla_\bullet g_{j_{t_-}})^2, \tag{7.7}$$

where $\nabla_\bullet$ denotes appropriate component of the gradient from (6.8), and ν_j is a weighting coefficient. The squared gradient component computed above can be used in the one-element clique potential to perform estimation from data in the Y-C1-C2 format.

7.2 EXPERIMENTAL RESULTS

In this section some experimental results of MAP motion estimation from colour sequences in Y-C1-C2 format will be presented. The algorithm will not be applied to the test

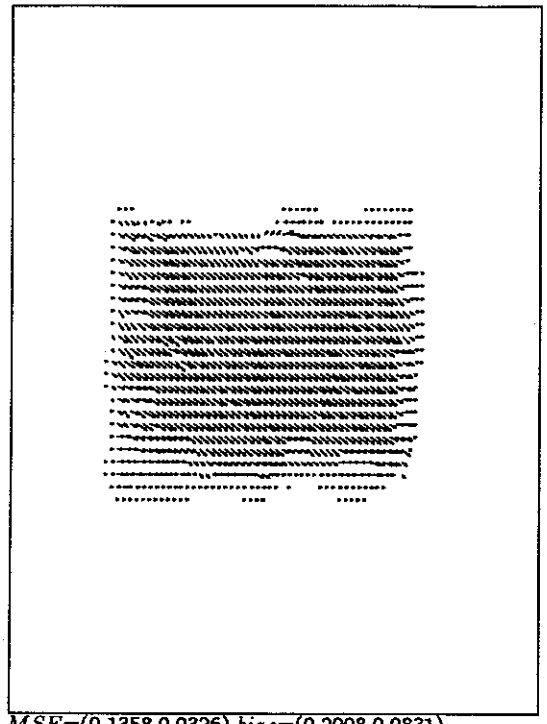
image 1, since as was demonstrated in Chapter 4 this image contains a very strong motion cue, and the resulting estimates had been already close to the true motion. Also the test image 4 will be omitted.

7.2.1 Results for test image 2

Fig. 7.1 compares motion estimates obtained from the test image 2 via the MAP criterion (discrete state-space Gibbs sampler) based on luminance matching only (a,b)†, and based on Y-C1-C2 matching (c,d). Both estimates have been obtained for the neighbourhood $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interpolation and exponential annealing schedule starting at $\mathbf{T}_0$=1.0 with a=0.980 over 200 iterations. The estimate from Fig. 7.1.a has been obtained for $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, while the estimate from Fig. 7.1.c was computed for $\lambda_{\mathbf{d}}/\lambda_{g_j}$= 6.667, j=1,2,3. In other words, in both cases the smoothing weight was 20.0, and it was uniformly distributed among the three components in estimation with colour. Figs. 7.1.b,.d show the vector fields of difference between the synthetic displacement field $\mathbf{d}_s$ from Fig. 4.13.a, and the motion fields from Figs. 7.1.a,.c, respectively. Note that the estimate obtained with the colour cue is smoother than the one without the colour, which is confirmed by the lower MSE displayed below the field, and also by fewer erroneous vectors in the central rectangle of the difference field.

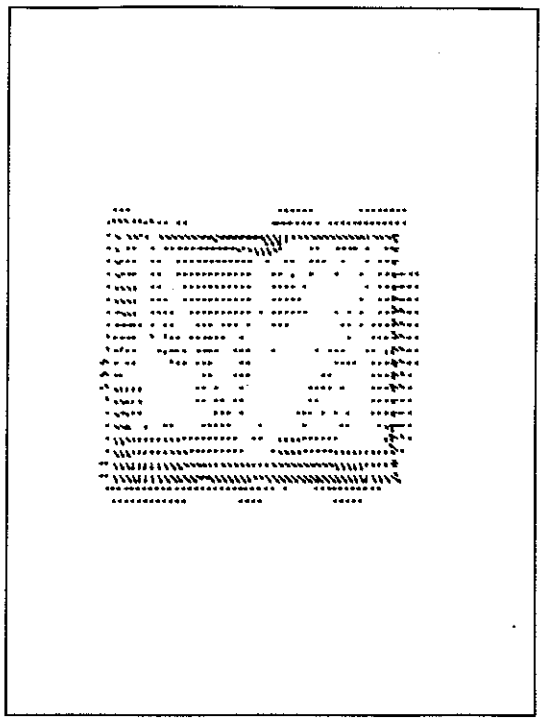
Also the motion model incorporating discontinuities has been tested on image 2. Fig. 7.2 shows the Y-C1-C2 estimate of motion field and of motion discontinuities field obtained with the same parameters as for the results from Fig. 7.1, plus equally weighted (ν_j=0.333, $\forall j$) squared magnitudes of gradients of the component signals used in the potential (6.8). Note a significant improvement over the no-colour estimate from Fig. 7.1.a in terms of motion field smoothness in the central rectangle. This estimate is also significantly improved over the one from Fig. 7.1.c in the vicinity of rectangle borders where the vectors are very uniform on the inside and disappear on the outside (this effect is somewhat less pronounced around the right-edge occlusion). Finally, compared to the already very good result from Fig. 6.5 (no colour, with line process), the estimate from Fig. 7.2 is superior in terms of vector field

† The result in Fig. 7.1.a is repeated from Fig. 4.13.c

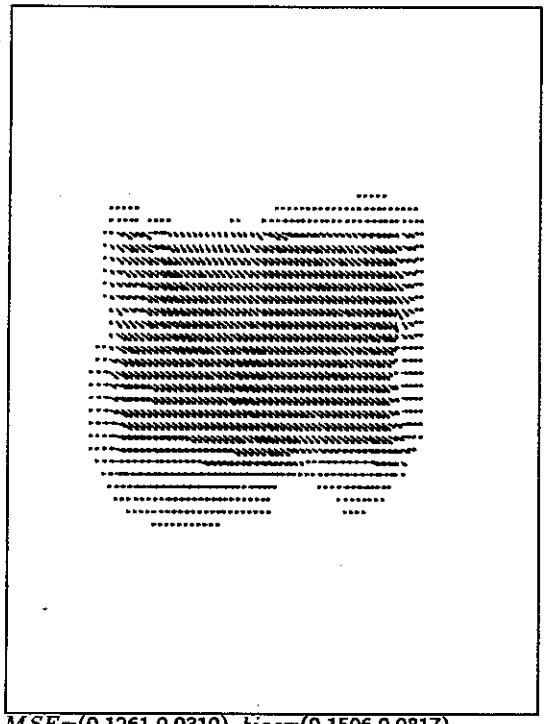


MSE=(0.1358,0.0326),$bias$=(0.2008,0.0831)

(a) luminance

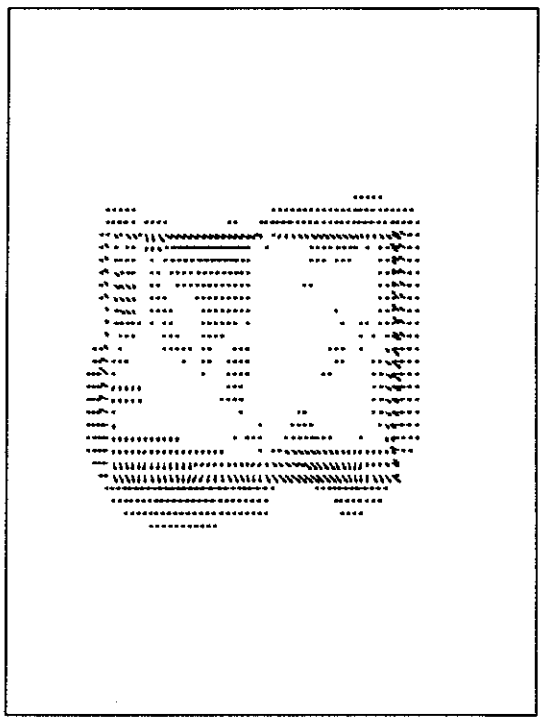


(b) diff. from true $\mathbf{d}_s$



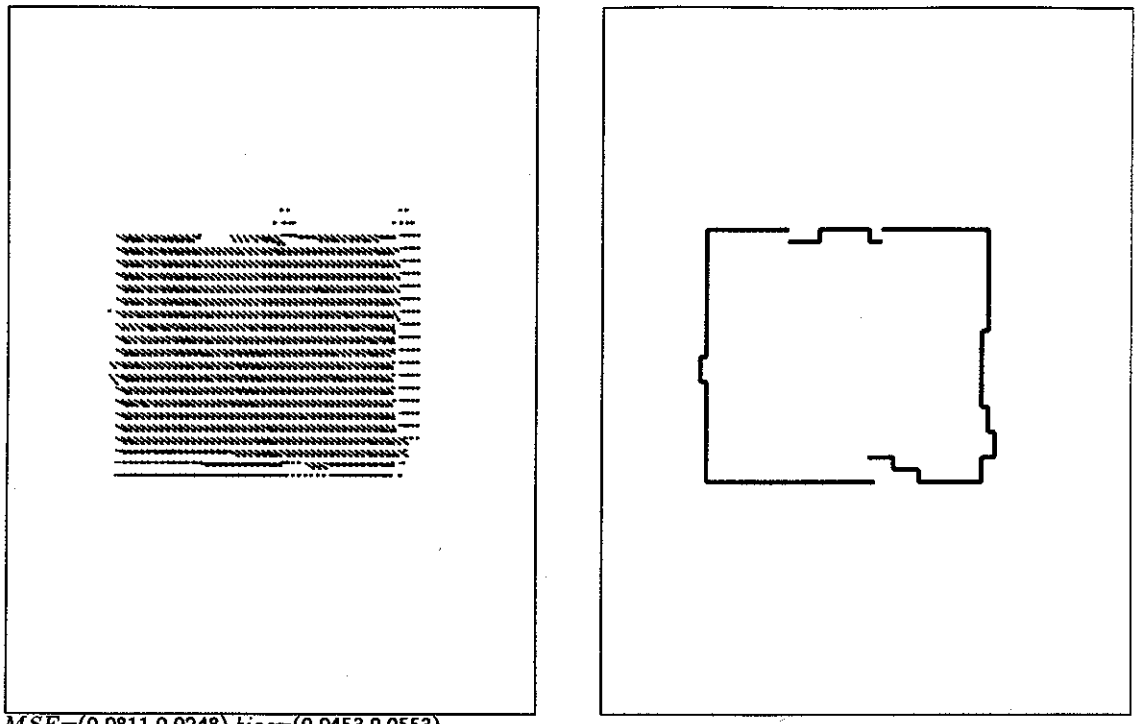
MSE=(0.1261,0.0310), $bias$=(0.1506,0.0817)

(c) Y-C1-C2



(d) diff. from $\mathbf{d}_s$

Fig. 7.1 Luminance and Y-C1-C2 discrete state-space MAP estimates: test image 2, $\lambda_{\mathbf{d}}/\lambda_g$=20.0 (a,b) and $\lambda_{\mathbf{d}}/\lambda_{g_j}$=6.667 (c,d) for j=1,2,3, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential sched., $\mathbf{T}_0$=1.0, a=0.980, 200 iter.



MSE=(0.0811,0.0248),$bias$=(0.0453,0.0553)

(a) displ. field *(b)* line field

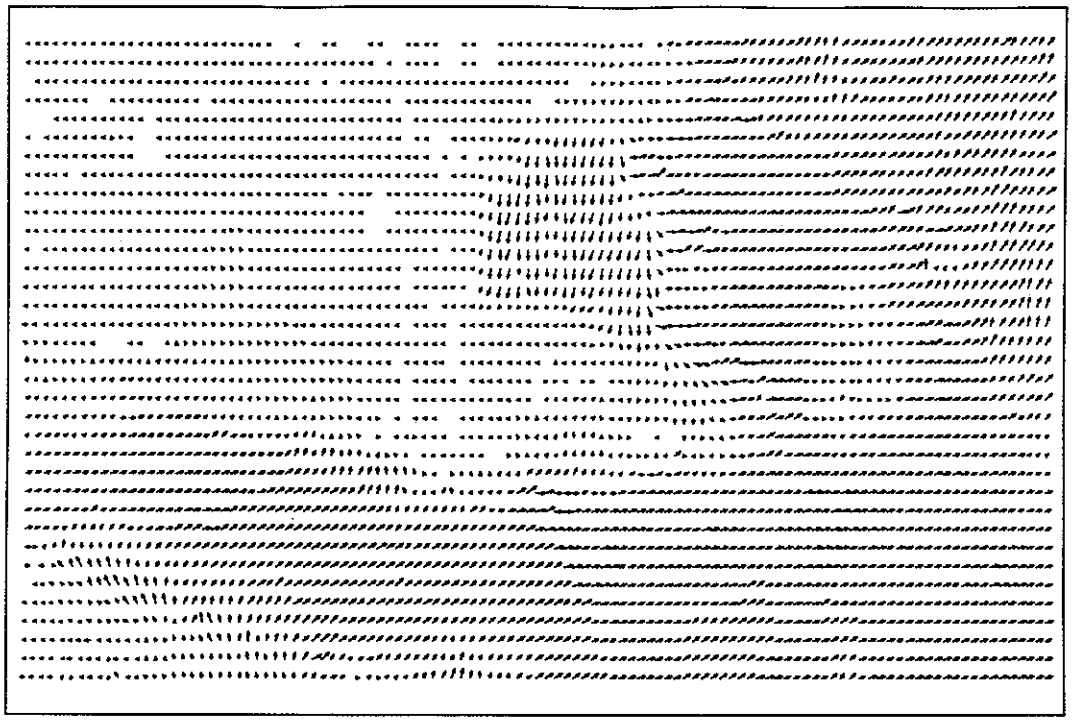
Fig. 7.2 Y-C1-C2 discrete state-space MAP estimate with piecewise smooth motion model: test image 2, $\lambda_{\mathbf{d}}/\lambda_{g_j}$=6.667 and ν_j=0.333 for j=1,2,3, $\lambda_l/\lambda_{\mathbf{d}}$=0.8, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., exponential sched., $\mathbf{T}_0$=1.0, a=0.9866, 300 iter.

smoothness within the rectangle, and also in terms of the mean squared error. This estimate is very close to the true motion field presented in Fig. 4.13.a. The artifacts are mostly due the occlusion areas where motion is undefined. The fidelity of the estimate is confirmed by significantly reduced mean squared error with respect to the estimates from Figs. 7.1.a,.c. The line field from Fig. 7.2.b, which ideally should be a rectangle, is still imprecise, however it clearly contributed to the reduction of MSE. Note that the parameters used in estimation were not optimized i.e., the same set of values as for the result with no colour (Fig. 7.1.a) has been used, and only the uniform contribution of individual components has been added.

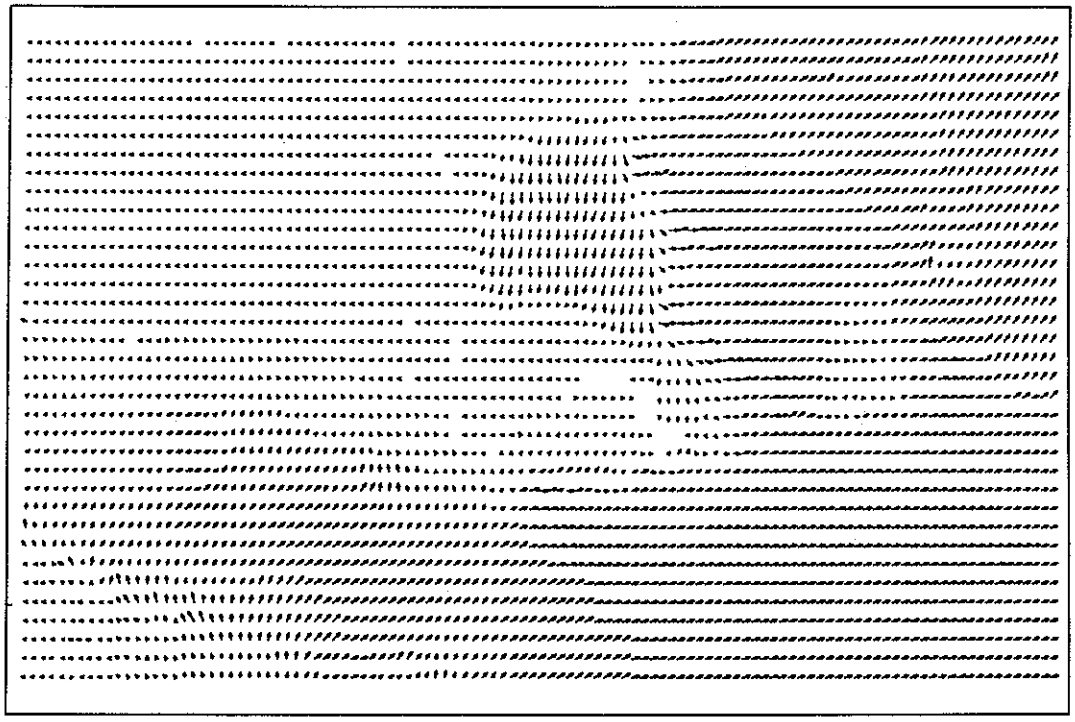
7.2.2 Results for test image 3

Fig. 7.3 shows motion estimates obtained from test image 3 using only luminance (a)†, and three components (b). The same parameters as in the previous section have

† Again the result in Fig. 7.3.a is repeated from Fig. 4.16.a

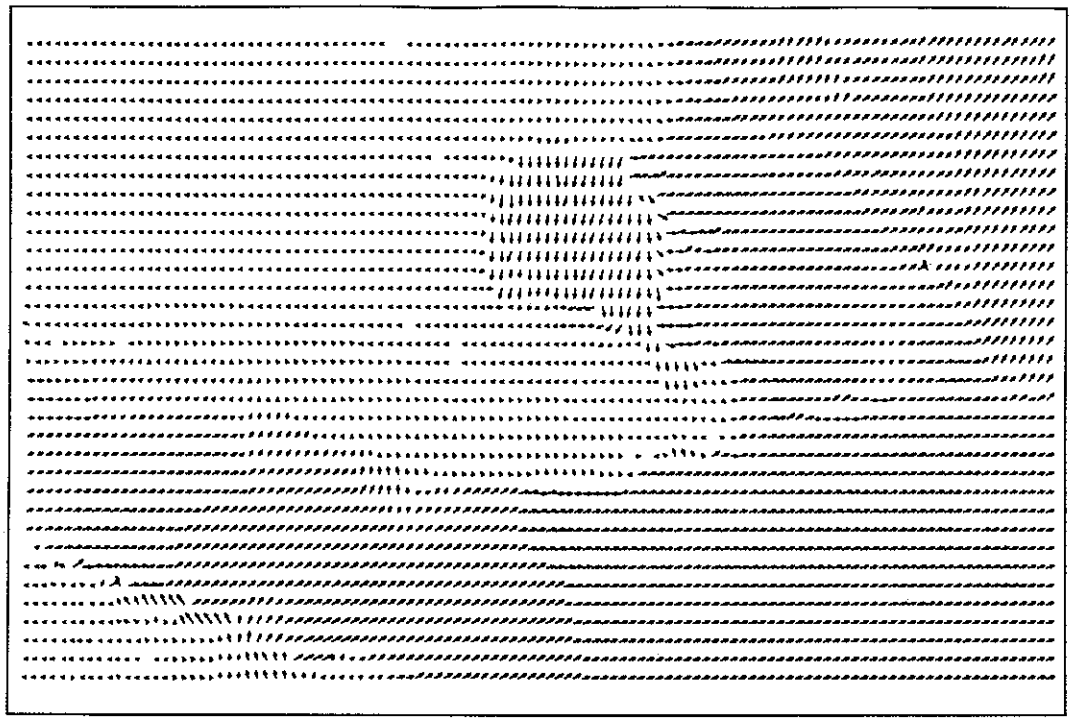


(a) luminance

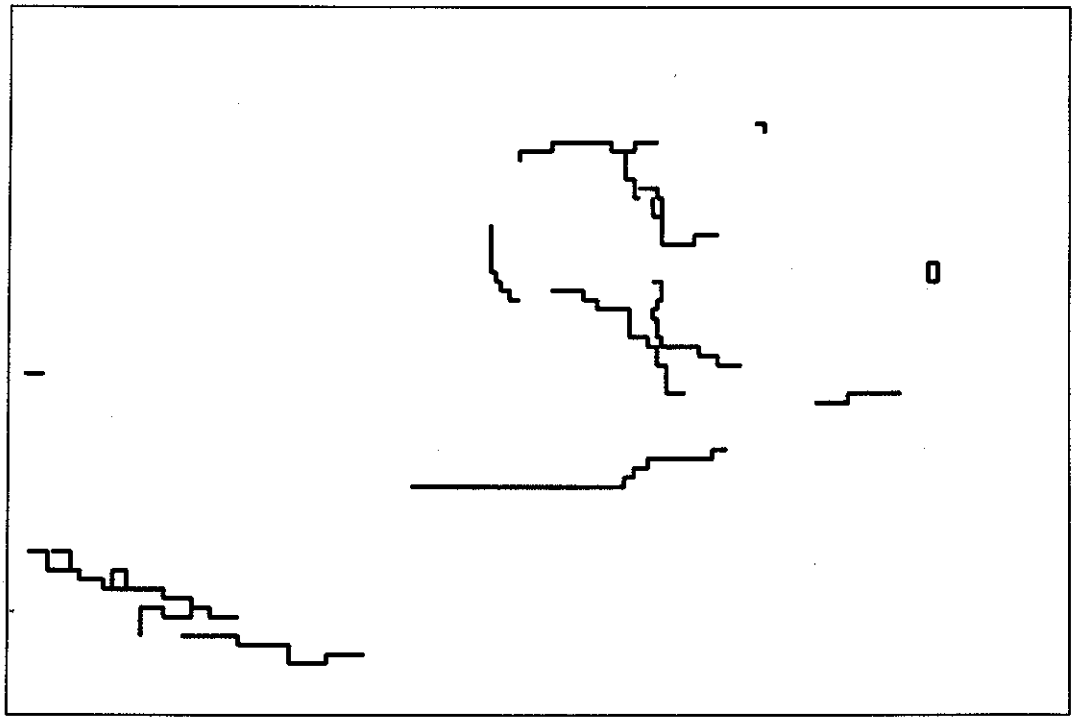


(b) Y-C1-C2

Fig. 7.3 Luminance and Y-C1-C2 discrete state-space MAP estimation: test image 3, $\lambda_{\mathrm{d}}/\lambda_g$=20.0 (a) and $\lambda_{\mathrm{d}}/\lambda_{g_j}$=6.667 (b) for j=1,2,3, neighb. $\mathcal{N}_{\mathrm{d}}^1$, bilinear interp., exponential sched., $\mathbf{T}_0$=1.0, a=0.980, 200 iter.



(a) displ. field



(b) line field

Fig. 7.4 Y-C1-C2 discrete state-space MAP estimate with piecewise smooth motion model: test image 3, $\lambda_{\mathbf{d}}/\lambda_{g_j}$=6.667 and ν_j=0.333 for j=1,2,3, $\lambda_l/\lambda_{\mathbf{d}}$=0.8, α=10.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bilinear interp., exponential sched., $\mathbf{T}_0$=1.0, a=0.9866, 300 iter.

been used, except for the bilinear interpolation. Note that although both estimates are similar at the first glance, a closer inspection will reveal that the estimate from Fig. 7.3.b is slightly smoother. Numerous isolated vectors from the luminance-only estimate (Fig. 7.3.a), differing significantly from their neighbours, have been aligned with those neighbours in the three-component estimate in spite of the same smoothing weight used. This smoothness is confirmed by the displacement energy for the three-component estimation which is about a half of the energy for luminance-only estimation.

To make the tests more complete, the two-layer model with estimation from Y-C1-C2 data has been also applied to the test image 3. The result is presented in Fig. 7.4 with the same set of parameters as those used for Fig. 7.3 plus uniform contribution of the three components to the image energy ($\lambda_{\mathrm{d}}/\lambda_{g_j}$=6.667, $\forall j$) and to the squared magnitude of the gradient (ν_j=0.333, all j). The improvement is not as significant as for the test image 2, however a closer inspection will demonstrate that the estimate is smoother than the one from Fig. 7.3.a and similar to Fig. 7.3.b. Also compared to the result from Fig. 6.8 the motion boundaries are better defined and less chaotic, but there are fewer of them.

Chapter 8 DETERMINISTIC APPROXIMATIONS TO STOCHASTIC MAP ESTIMATION

Stochastic methods based on the MAP and MEC estimation criteria (Chapters 3, 4) find the global optima of the appropriate cost functions provided that certain conditions are satisfied. Due to these conditions the methods are characterized by significant computational effort, especially the discrete state-space Gibbs sampler which requires computation of the complete local conditional probability distribution. In this chapter I will investigate deterministic optimization methods to approximate the discrete and continuous state-space MAP estimation.

In the next section the maximum marginal conditional *a posteriori* probability estimation will be investigated, and the results of its application to the test images will be shown. In the following section the Gauss-Newton optimization method will be used to minimize the MAP estimation criterion. It will be shown that this approximation is a modified version of the algorithm proposed by Horn and Schunck [41]. Also a deterministic approximation to the stochastic method incorporating motion discontinuities will be discussed. The results produced by these methods will be compared with the ones obtained via stochastic optimization.

8.1 MAXIMUM MARGINAL CONDITIONAL A POSTERIORI PROBABILITY (MMCAP) ESTIMATION

8.1.1 Algorithm description

Recall that the Gibbs sampler described in Section 4.2.3 generates discrete deviates from the marginal conditional *a posteriori* probability distribution defined in (4.12). Those

deviates are used to guide the chain towards the state of the global optimum. The local probability distribution describes the likelihood of a state at current position, given the states at neighbouring positions and the data. Due to randomness (actually pseudo-randomness) involved in this algorithm, the states obtained do not have to correspond to the maximum probability. Moreover, sometimes they can even correspond to a very low likelihood. This is the very principle of stochastic relaxation. But what would happen if the randomness were eliminated, and for example only the states corresponding to the maximum of the marginal conditional probability (4.12) were retained ? In this way the process would converge much faster than in simulated annealing, however it would not provide (in general) the global optimum in the sense of the MAP criterion.

The approach described above has been originally suggested by Besag [11] and termed by him *iterated conditional modes* (ICM). He argued that since it is difficult to maximize the joint *a posteriori* probability over the complete field, and since sometimes it is not profitable to do so (?), the random field (e.g., displacement) should be divided into a minimal number of disjoint sets such that any two random variables from a given set are conditionally independent given the states of the other sets. If the field is defined over a lattice (e.g., $\Lambda_{\mathbf{d}}$) those sets become cosets [21]. Since they are frequently distinguished by assigning a different colour to each of them, sometimes the name *coding colours* is used. Fig. 8.1 shows two cosets (colours) necessary to provide the independence for the $\mathcal{N}_{\mathbf{d}}^1$ neighbourhood system.

Since the random variables within one colour are conditionally independent given other colours, they can be optimized individually i.e., the following minimization problem can be solved

$$\max_{\widehat{\mathbf{d}}(\mathbf{x}_i,t)} P(\mathbf{D}(\mathbf{x}_i,t)=\widehat{\mathbf{d}}(\mathbf{x}_i,t)|\mathbf{D}(\mathbf{x}_j,t)=\widehat{\mathbf{d}}(\mathbf{x}_j,t), j\neq i, G_{t_-}=g_{t_-}, G_{t_+}=g_{t_+}) \qquad \forall i \in \text{one colour.} \tag{8.1}$$

Minimizing one colour after another will result in a complete iteration. If $\mathbf{D}_t$ is modeled by a MRF with neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$, the conditional probability in (8.1) can be replaced by the marginal conditional *a posteriori* probability (4.12)†. Equivalently, mini-

† This results in Maximum Marginal Conditional *A Posteriori* Probability (MMCAP) estimation.

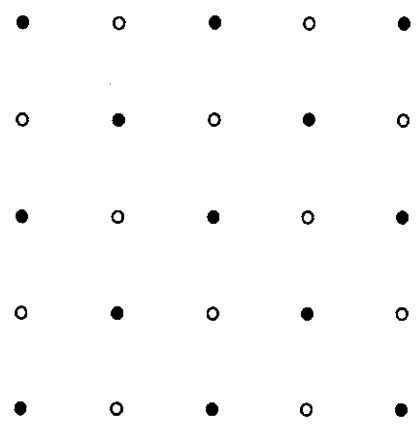


Fig. 8.1 Independent cosets (colours) • and ○ for rectangular lattice $\Lambda_{\mathbf{d}}$ and neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$: random variables associated with any two sites from the same coset are conditionally independent given the state of the elements of the other coset.

mization of the local energy $U_{\mathbf{d}}^i$ (4.13), which is a non-quadratic function of displacement vectors (via DPDs), can be performed. Since the state-space for the displacement vectors is discrete, either an exhaustive search or one of the efficient search techniques described in Section 2.3.3.1 must be applied. Standard optimization methods like Newton or conjugate gradients are not suitable, since they provide continuous instead of discrete solutions. If the displaced pel difference as a function of displacement vector $\widehat{\mathbf{d}}(\mathbf{x}_i, t)$ possesses more than one minimum, the fast search procedures may not be able to locate the smallest one, and consequently will not provide a MMCAP estimate. Hence, I will use the exhaustive search procedure i.e., the local energy $U_{\mathbf{d}}^i$ will be computed for each possible vector from $\mathcal{S}_{\mathbf{d}}'$, and the vector which minimizes this energy will be considered the estimate.

In the language of statistical mechanics the above process is equivalent to *quenching* or instantaneous freezing, in which the temperature is reduced to a minimum extremely rapidly. This process solidifies a material very quickly, however there remain various artifacts "frozen" into the solid and its state is far from the energy global minimum.

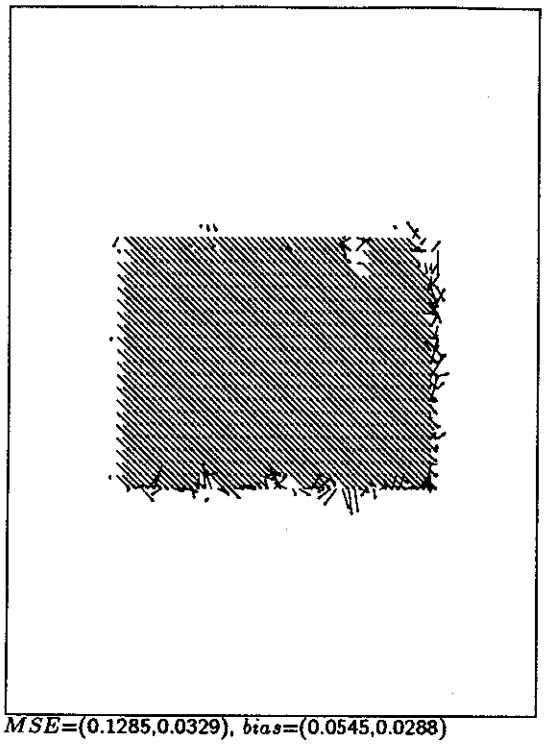
One should expect that the algorithm will converge in few iterations, and that the result will stay unchanged from then on, rather than oscillate like for the MAP or MEC estimation.

8.1.2 Experimental results

Fig. 8.2 shows the MMCAP estimates of motion in the test images 1 and 2. The estimate from Fig. 8.2.a was obtained for $\lambda_{\mathbf{d}}/\lambda_g$=0.05, first-order neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$ and bilinear interpolation. The algorithm, being an exhaustive search, needed only 7 iterations to converge. Note that the estimate is very close to the corresponding discrete state-space MAP estimate. This can be explained by the fact that there is very little interaction between neighbouring vectors, hence the estimation is based primarily on pel matching. In the limiting case, when $\lambda_{\mathbf{d}}/\lambda_g$=0.0, the joint probability $P(\mathbf{D}_t|g_{t_-}, g_{t_+})$ in MAP estimation can be expressed as a product of marginal likelihoods as in (3.13), which can be maximized individually. Consequently both the MAP and the MMCAP estimation minimize the same objective function, and the only difference is that MAP estimation does it randomly. If the motion cue is strong, similar results may be expected. For $\lambda_{\mathbf{d}}/\lambda_g \neq 0$ this is not the case, but since for the value 0.05 the interaction between displacement estimates is small, the MMCAP estimate still resembles the MAP estimate quite well. The mean squared error for the MMCAP estimate, however, is significantly higher than the error for the corresponding MAP estimate (Fig. 4.11.c).

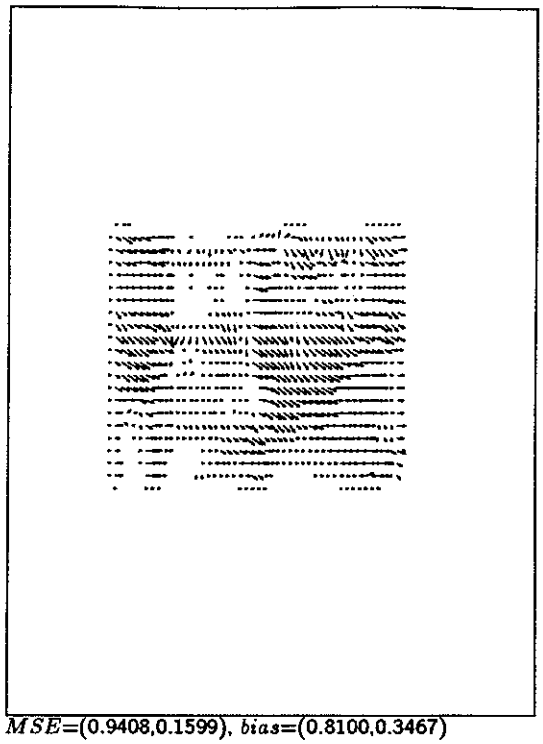
The use of higher-order spatial data interpolation (e.g., bicubic) had marginal effect on the MMCAP estimation, as expected, since the true motion points to the spatial locations at t_+ which belong to Λ_g. Also the second-order neighbourhood system $\mathcal{N}_{\mathbf{d}}^2$ did not improve the results significantly, since due to the strong data cue the motion model plays a minor role. The similarity of the MMCAP and MAP estimates suggests that the objective function for the test image 1 is close to a unimodal one. Another way of looking at it is to note that the data consist of "gray value corners" everywhere, hence are easy to match.

Fig. 8.2.b shows the MMCAP estimate of motion from the test image 2 for the same set of parameters as used for the result from Fig. 4.13.c. Clearly the result has little in common with the true motion field, and also is very far from the good estimates provided by the MAP and MEC estimation, what is confirmed by the high mean squared error. This shows that unless the data provides a very strong motion cue the MMCAP estimation may provide very false solutions. As explained before, MMCAP estimation is just a simplification



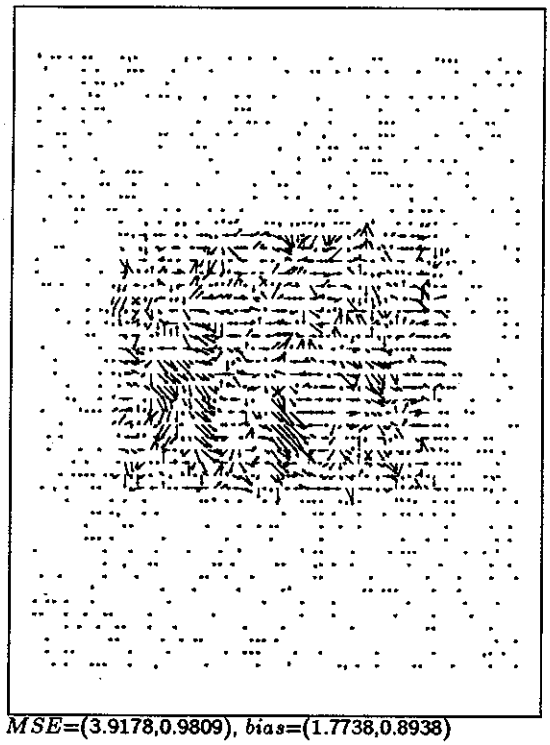
MSE=(0.1285,0.0329), $bias$=(0.0545,0.0288)

(a) test image 1, $\lambda_{\mathbf{d}}/\lambda_g$=0.05



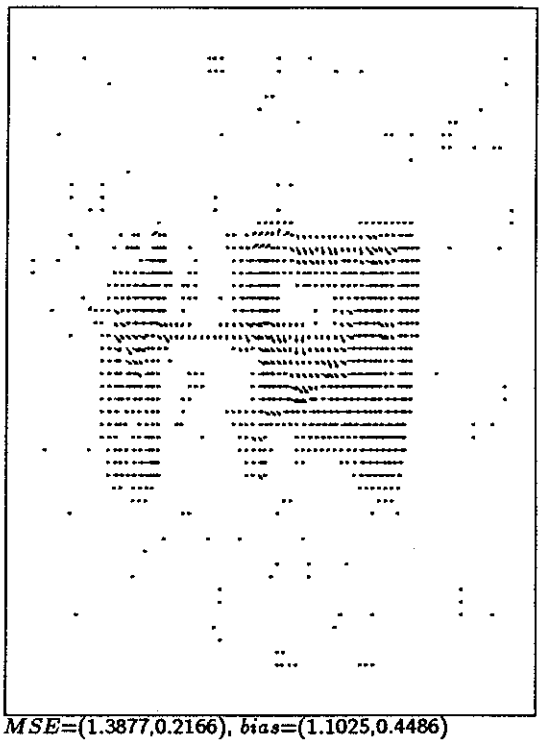
MSE=(0.9408,0.1599), $bias$=(0.8100,0.3467)

(b) test image 2, $\lambda_{\mathbf{d}}/\lambda_g$=20.0



MSE=(3.9178,0.9809), $bias$=(1.7738,0.8938)

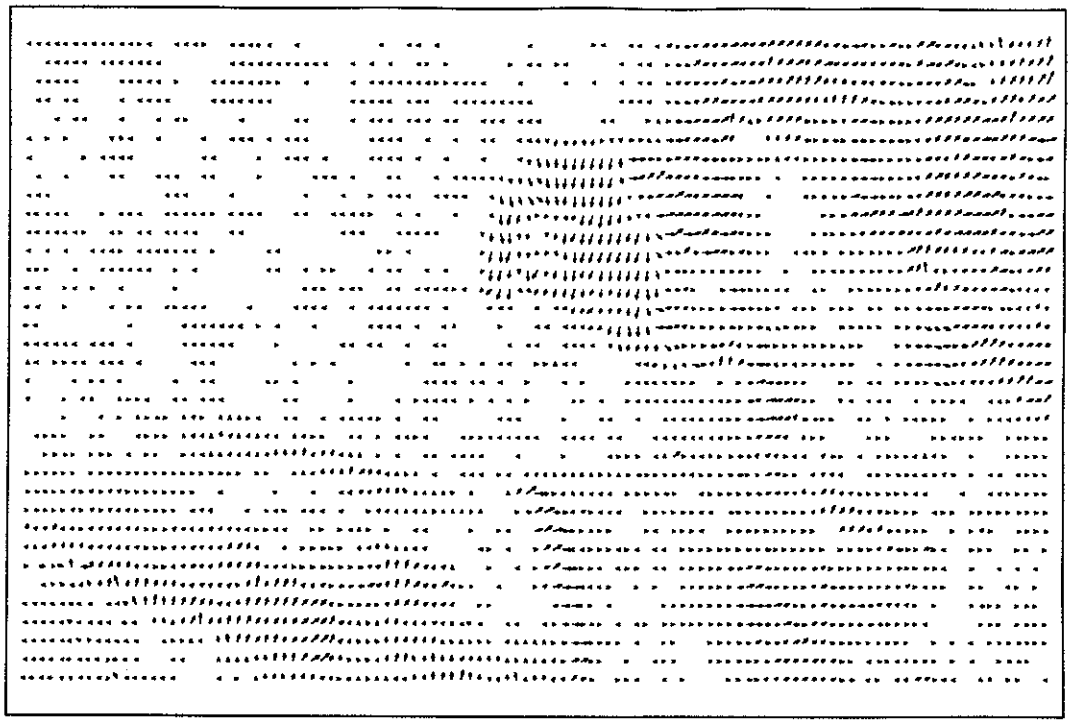
(c) test image 1 + noise, $\lambda_{\mathbf{d}}/\lambda_g$= 100.0



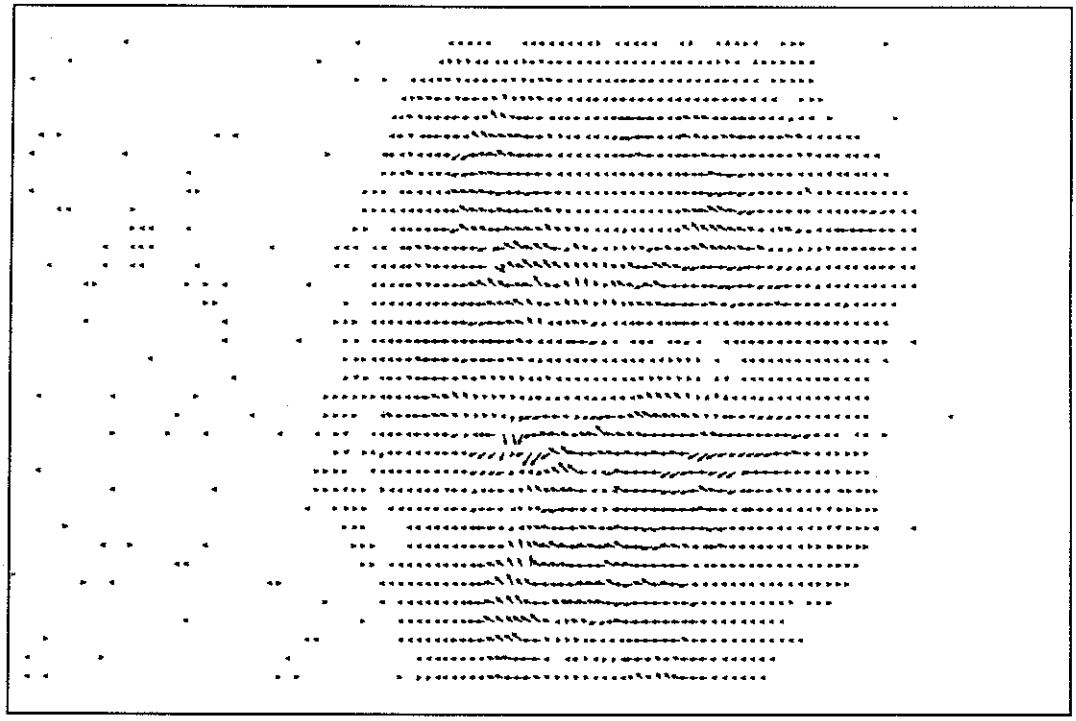
MSE=(1.3877,0.2166), $bias$=(1.1025,0.4486)

(d) test image 2 + noise, $\lambda_{\mathbf{d}}/\lambda_g$= 120.0

Fig. 8.2 MMCAP estimates: test images 1 and 2, neighbourhood $\mathcal{N}_{\mathbf{d}}^1$, bilinear (a,c) and Keys bicubic (b,d) interp.



(a) test image 3



(b) test image 4

Fig. 8.3 MMCAP estimates: test images 3 and 4, $\lambda_{\mathbf{d}}/\lambda_g = 20.0$, neighb. $\mathcal{N}_{\mathbf{d}}^1$, bilinear interp.

of MAP estimation, and does not attempt to maximize the joint *a posteriori* probability but only local marginal conditional probabilities. Concluding, a global rather than a local estimation criterion should be used for motion computation.

Even poorer performance of the MMCAP estimation can be observed in the presence of noise, as shown in Figs. 8.2.c,.d. The estimates for both test images do not resemble the true motion at all, and also are very far from the MAP and MEC estimates for the same data and parameters. The mean squared error is also much higher than for the corresponding estimates from the noiseless data.

Figs. 8.3.a,.b show the MMCAP estimation results for the test images 3 and 4. Since the image size is larger, the algorithm converged in 18 and 30 iterations, respectively. The results are poorer than for the stochastic relaxation, especially for the test image 3. Note that the algorithm failed to compute correctly the motion of the forearm and of the arm, except for the displacement vectors along the edge of the shirt sleeve. Also the vectors on the neck and parts of the face suggest that there is no motion, which is incorrect. The result for the test image 4 is somewhat better, however also there patches of very short or absent vectors do not reflect the true motion in the image.

Concluding, the quality of motion estimates obtained with the MMCAP estimation depends on the data, however even for moderately difficult images it is clearly inferior to the MAP estimation implemented via stochastic relaxation.

8.2 GAUSS-NEWTON MINIMIZATION OF THE MAP CRITERION

8.2.1 Algorithm description

Description of the continuous state-space approximation will be divided into three subsections. The first one will describe the basic algorithm, the second will briefly discuss the hierarchical extension, while the third one will augment the previous two by including discontinuities in the motion model.

8.2.1.1 Basic algorithm

Recall the minimization problem (3.21) resulting from the MAP estimation, and rewrite

it explicitly in terms of the displaced pel differences and potential functions:

$$\min_{\hat{\mathbf{d}}_t} \sum_{i=1}^{M_{\mathbf{d}}} \Big[\lambda_g \cdot [\tilde{r}(\hat{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2 + \lambda_{\mathbf{d}} \cdot \sum_{j:\mathbf{x}_j \in \eta_{\mathbf{d}}(\mathbf{x}_i)} V(\hat{\mathbf{d}}(\mathbf{x}_i,t),\hat{\mathbf{d}}(\mathbf{x}_j,t))\Big].$$

Note that due to symmetry of cliques $V(\hat{\mathbf{d}}(\mathbf{x}_i,t),\hat{\mathbf{d}}(\mathbf{x}_j,t))= V(\hat{\mathbf{d}}(\mathbf{x}_j,t),\hat{\mathbf{d}}(\mathbf{x}_i,t))$. Consequently, $\lambda_{\mathbf{d}}$ should be reduced by the size of clique $|c_{\mathbf{d}}|$, which for neighbourhood system $\mathcal{N}_{\mathbf{d}}^1$ is equal to 2. The function under minimization is not quadratic in terms of the displacement field estimate $\hat{\mathbf{d}}_t$, hence a general optimization procedure must be used. I will approach the problem similarly as in the case of the continuous state-space Gibbs sampler (Section 4.5) which is equivalent to applying Gauss-Newton minimization.

Assume that an approximate estimate $\dot{\mathbf{d}}_t$ of the displacement field is known, and that the image intensity is locally slowly varying. Using the first-order terms of the Taylor expansion linearize the displaced pel difference as in (4.15) with the spatial gradient of $\tilde{r}$ defined in (4.16). Using this linearization of $\tilde{r}$ and also the definition of the potential function (3.17), the minimization problem above can be approximated by the following form:

$$\min_{\hat{\mathbf{d}}_t} \sum_{i=1}^{M_{\mathbf{d}}} \Big[\lambda_g \cdot [\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t) + (\hat{\mathbf{d}}(\mathbf{x}_i,t) - \dot{\mathbf{d}}(\mathbf{x}_i,t)) \cdot \nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)]^2 + \lambda_{\mathbf{d}} \cdot \sum_{j:\mathbf{x}_j \in \eta_{\mathbf{d}}(\mathbf{x}_i)} \|\hat{\mathbf{d}}(\mathbf{x}_i,t) - \hat{\mathbf{d}}(\mathbf{x}_j,t)\|^2\Big].$$

Since the function under minimization is now quadratic with respect to $\hat{\mathbf{d}}_t$, necessary conditions for optimality can be established by differentiating this function with respect to $\hat{\mathbf{d}}(\mathbf{x}_i,t)$ for each $i= 1,...,M_{\mathbf{d}}$. The resulting equations have the following form:

$$2 \cdot \lambda_g \cdot \nabla_{\mathbf{d}}^T\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t) \cdot [\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t) + (\hat{\mathbf{d}}(\mathbf{x}_i,t) - \dot{\mathbf{d}}(\mathbf{x}_i,t)) \cdot \nabla_{\mathbf{d}}\tilde{r}(\dot{\mathbf{d}}(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t)] + 2 \cdot \lambda_{\mathbf{d}} \cdot \xi_i \cdot [\hat{\mathbf{d}}(\mathbf{x}_i,t) - \overline{\mathbf{d}}(\mathbf{x}_i,t)] = 0, \qquad \forall i,$$

where again $\overline{\mathbf{d}}$, defined in (4.19), denotes the average vector computed over neighbourhood $\eta_{\mathbf{d}}(\mathbf{x}_i)$, and $\xi_i = |\eta_{\mathbf{d}}(\mathbf{x}_i)|$ is the size of this neighbourhood. Solving the above vector equation, and assuming as before (Section 4.5) that $\dot{\mathbf{d}} = \overline{\mathbf{d}}$ at every iteration, the estimation process can be described by the (Gauss-Seidel) iterative update:

$$\hat{\mathbf{d}}^{n+1}(\mathbf{x}_i,t) = \overline{\mathbf{d}}^n(\mathbf{x}_i,t) - \frac{\varepsilon_i}{\mu_i}\nabla_{\mathbf{d}}^T\tilde{r}(\overline{\mathbf{d}}^n(\mathbf{x}_i,t),\mathbf{x}_i,t,\Delta t), \tag{8.2}$$

with ε_i and μ_i defined in (4.18), and superscript n denoting the iteration number.

Note that this iterative equation is exactly the same as equation (4.20) for the first-order neighbourhood system $\mathcal{N}_{\mathbf{d}}^{1}$† and for n_i=0. In such a case there is no uncertainty involved in the estimation process, and similarly as in the case of the MMCAP estimation (discrete state-space) the above algorithm is an example of quenching i.e., instantaneous reduction of temperature **T** to zero. Consequently, this rapid temperature reduction may not allow the system to attain the global minimum of the energy function.

Clearly, the deterministic approximation to the continuous state-space MAP estimation is a spatio-temporal gradient technique which can be viewed as a modified version of the Horn and Schunck algorithm described in Section 2.3.4. There are differences, however:

1. the modified algorithm (8.2) allows computation of displacement vectors for arbitrary $\Lambda_{\mathbf{d}}$ (arbitrary spatio-temporal positions $(\mathbf{x}, t)$) unlike the original Horn and Schunck algorithm in which $\Lambda_{\mathbf{d}} = \Lambda_g$+[0.5,0.5,0.5]^T,
2. the scalar ε_i is equal to a displaced pel difference in the modified version rather than to a motion constraint equation, both evaluated for $\overline{\mathbf{d}}(\mathbf{x}_i, t)$; consequently, no temporal derivative is needed,
3. the spatial intensity derivatives are computed from a separable polynomial intensity model in both images and appropriately weighted (4.16), instead of the finite difference approximation over a cube as proposed in [41].

The ability to estimate motion for arbitrary $\Lambda_{\mathbf{d}}$ is crucial for motion-compensated interpolation of sequences. The original Horn-Schunck algorithm would require 3-D interpolation of motion fields to obtain estimates at arbitrary position $(\mathbf{x}, t)$.

The use of a displaced pel difference instead of the motion constraint equation in ε_i is important because it allows intensity pattern tracking thus permitting more accurate intensity derivative computation, and also excludes necessity to compute the temporal derivative (actually, $\tilde{r}$ is an approximation to the directional derivative). The purely temporal derivative used in the Horn-Schunck algorithm is a reliable measure of temporal intensity change due to motion only if small displacements are applied to linearly varying intensity pattern. If the displacements are not small or if the intensity pattern is far from linearity, significant errors result, for example an overestimation at moving edges of high contrast. A

† Recall that for the first order neighbourhood system ξ_i equals 4, and ε_i and μ_i are defined in (4.21).

replacement of the motion constraint equation in ε_i by a displaced pel difference has been proposed by Nagel [69], however he did not provide any justification for such modification. He suggested computing $\tilde{r}$ as an average from $\tilde{r}$'s of the neighbouring vectors, while here it is evaluated at $\overline{\mathbf{d}}(\mathbf{x}_i, t)$.

The spatial derivative computation from an intensity model applied at the ends of vector $\overline{\mathbf{d}}(\mathbf{x}_i, t)$ is more general than that proposed by Horn and Schunck. Note, however, that it is important to have a C^1-continuous model as stressed in Section 4.6. In particular, for bilinear interpolation, $\Lambda_{\mathbf{d}} = \Lambda_g + [0.5, 0.5, 0.5]^T$ and $\overline{\mathbf{d}}(\mathbf{x}_i, t) = 0$ the finite difference approximation over a cube as proposed in [41] results.

8.2.1.2 Hierarchical algorithm

Recall the hierarchical continuous state-space MAP estimation from Section 5.2.2. Using the same notion of the *base* displacement field $\mathbf{b}_t$ and eliminating the random variable n_i, the following deterministic relaxation can be performed

$$(\hat{\mathbf{h}}^{\kappa})^{n+1}(\mathbf{x}_i, t) = (\overline{\mathbf{h}}^{\kappa})^{n}(\mathbf{x}_i, t) + \overline{\mathbf{b}}^{\kappa}(\mathbf{x}_i, t) - \mathbf{b}^{\kappa}(\mathbf{x}_i, t) - \frac{\varepsilon_i}{\mu_i} \nabla_{\mathbf{h}}^{T} \tilde{r}\Big((\overline{\mathbf{h}}^{\kappa})^{n}(\mathbf{x}_i, t) + \overline{\mathbf{b}}^{\kappa}(\mathbf{x}_i, t), \mathbf{x}_i, t, \Delta t\Big), \quad \forall i. \tag{8.3}$$

As before, the *base* displacement field $\mathbf{b}_t$ is constant at given resolution level, and only the *incremental* field $\hat{\mathbf{h}}_t$ is modified. The *base* displacement is updated only through the interpolation stage between two neighbouring resolution levels. Other components of the hierarchical approach, like filtering or inter-level interpolation, are the same as those used in the stochastic approach (Chapter 5).

This algorithm is related to the hierarchical methods proposed by Glazer [31] and Enkelmann [22]. Both of them use a more sophisticated multilevel control algorithm, while here simply a non-recursive, top-bottom procedure is used. Also the pyramid structure for the data is different as well as the interpolation scheme used for computation of intensities at non-integer positions. Also Enkelmann, following Nagel's oriented smoothness constraint, uses variable-space smoothing. Both formulations, however, are based on the Horn-Schunck approach and are similar to the one proposed here. Also the iterative updates are quite alike.

8.2.1.3 Algorithm incorporating motion discontinuities

Recall the iterative equation (4.20) for the continuous state-space MAP estimation. With the definitions of ξ_i and $\overline{\mathbf{d}}$ given in (6.15) and (6.16), respectively, it is also the update equation for the model with motion discontinuities. Again, after disregarding the random term n_i that update equation becomes the equation (8.2). The only difference is that ξ_i and $\overline{\mathbf{d}}$ are computed through (6.15) and (6.16), instead of (4.18) and (4.19).

This deterministic approximation is related to the algorithm proposed in [44] and based on the Horn-Schunck approach with addition of deterministic motion discontinuities. The major differences are in the use of the displaced pel difference here instead of the motion constraint equation as in [44], and in the choice of line potentials. In particular, the potential V_{l_1} for single-element cliques is binary (0 or ∞) in [44], while it varies continuously according to the local intensity gradient here. Otherwise the methods are identical.

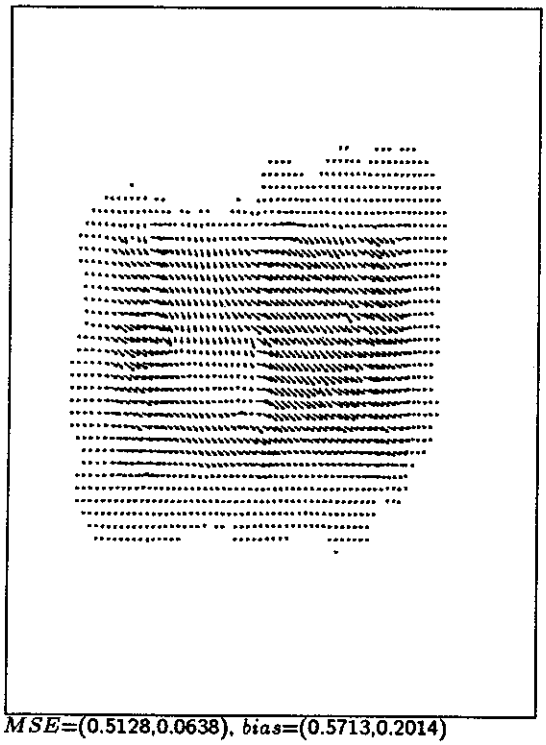
The extension of this deterministic approximation to a hierarchy of resolutions is straightforward after defining appropriate pyramids as discussed in Section 6.6.

8.2.2 Experimental results

In this section some experimental results of application of the deterministic approximation to the continuous state-space MAP estimation will be presented. The test image 1 will not be used since it consists of uncorrelated pels, and as such results in uncorrelated intensity gradients. Since every spatio-temporal gradient method relies on a close relationship between the spatial and temporal gradients, such a method is not expected to work well in the case of test image 1.

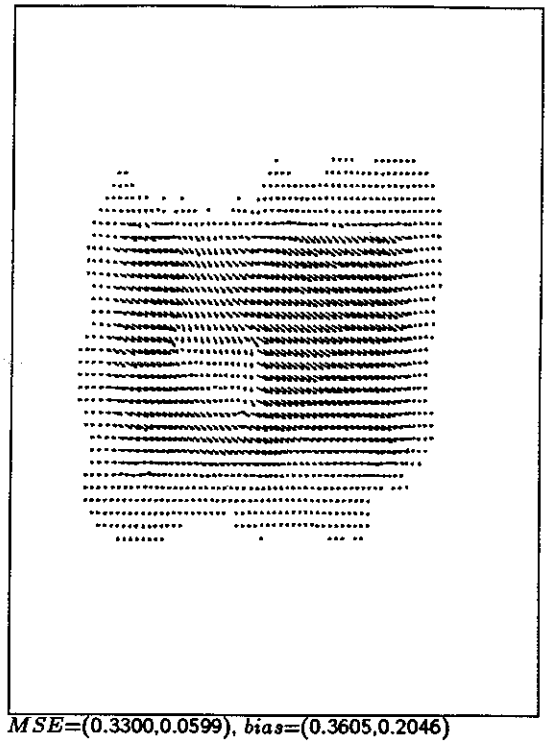
8.2.2.1 Basic algorithm

In Fig. 8.4.a the result of the original Horn-Schunck method, exactly implemented as suggested in the paper, applied to the test image 2 is shown for the ratio $\lambda_{\mathbf{d}}/\lambda_g = 20.0$ after 50 iterations. Recall that for this method $\Lambda_{\mathbf{d}} = \Lambda_g + [0.5, 0.5, 0.5]^T$. Consequently, to compare this result with other methods, Fig. 8.4.b shows the deterministic approximation to the continuous state-space MAP estimation as described above, while Fig. 8.4.c presents the continuous state-space MAP estimate itself. In both cases $\lambda_{\mathbf{d}}/\lambda_g$=20.0, Keys bicubic



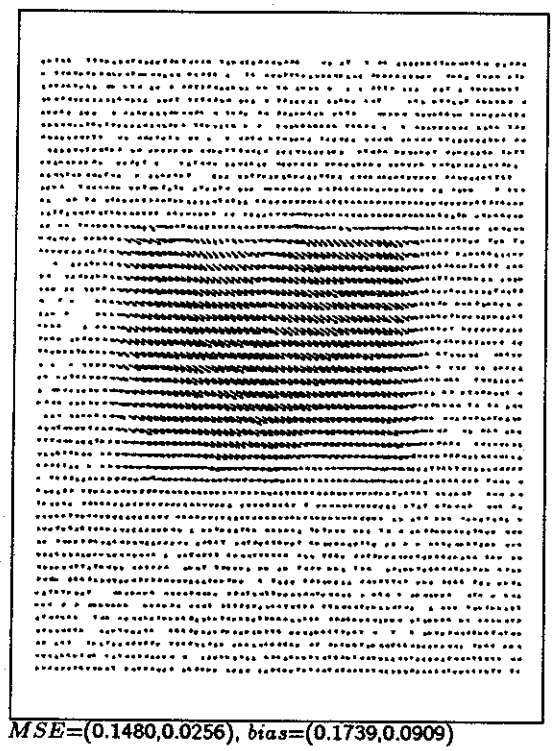
MSE=(0.5128,0.0638), $bias$=(0.5713,0.2014)

(a) Horn-Schunck, 50 iter.



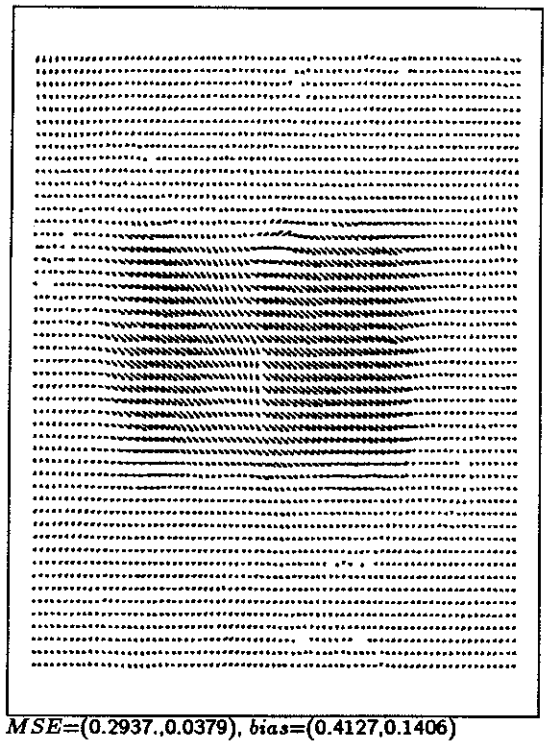
MSE=(0.3300,0.0599), $bias$=(0.3605,0.2046)

(b) determ., 50 iter.



MSE=(0.1480,0.0256), $bias$=(0.1739,0.0909)

(c) cont. MAP, 1000 iter.



MSE=(0.2937.,0.0379), $bias$=(0.4127,0.1406)

(d) determ., + noise, $\Lambda_{\mathbf{d}} = \Lambda_g$

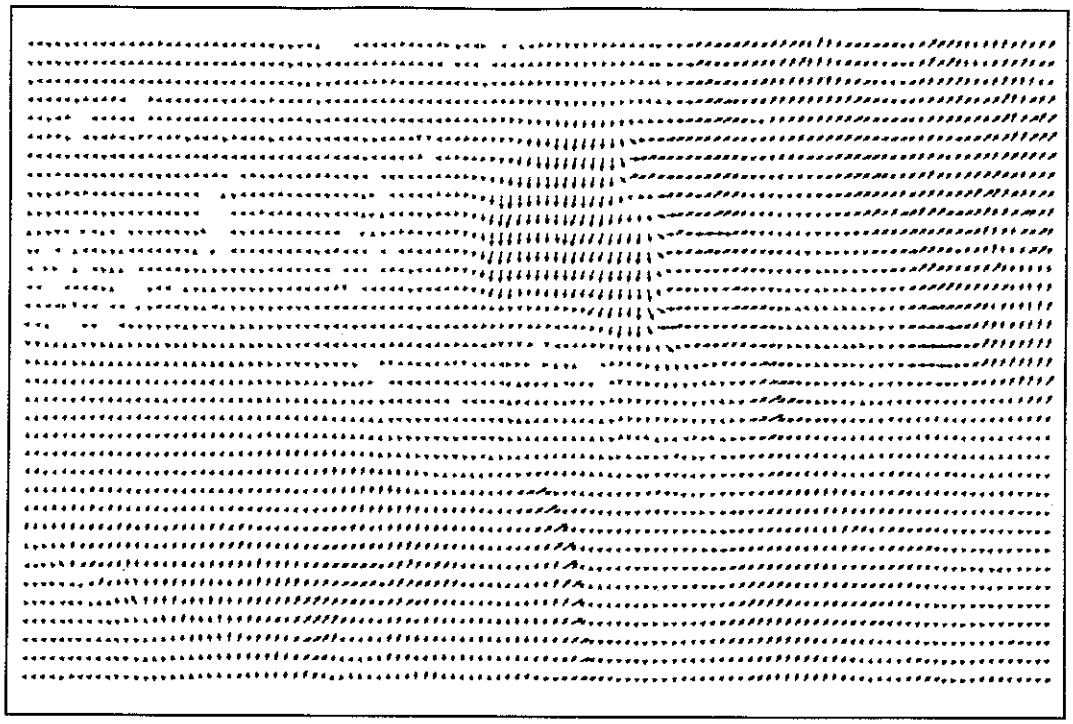
Fig. 8.4 Comparison of various algorithms for $\Lambda_{\mathbf{d}} = \Lambda_g + [0.5,0.5,0.5]^T$: test image 2, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0 (a,b,c) and 120.0 (d), neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., $\mathbf{T}_0$=5.0, a=0.9944 (c).

interpolation and $\Lambda_{\mathbf{d}} = \Lambda_g + [0.5, 0.5, 0.5]^T$† have been used. The deterministic method has used 50 iterations, while in the stochastic method T_0=5.0, a=0.9944 over 1000 iterations have been applied. Note that the Horn-Schunck algorithm produces the worst result, both subjectively and in terms of the MSE. The motion tends to be overestimated at strong edges (due to the purely temporal gradient), while it is underestimated in uniform areas. The deterministic approximation has produced a significantly lower MSE, and also subjectively the estimate is more uniform. Except for the visible triangle of underestimated displacements, the motion has been quite well computed. Superiority of the stochastic approach is clear from Fig. 8.4.c. The mean squared error is the lowest of the three estimates, and also subjectively this estimate is closest to the true motion. The image energy cannot be used as a parameter in the case of the Horn-Schunck approach, because this algorithm does not attempt to modify it. Such energy is, however, lower by about 5% in the case of the stochastic method compared to the deterministic one. This result is very significant, since not only theoretically, but also in practice, it may be profitable to slowly attain low temperatures rather than to perform instantaneous freezing. In other words, using a stochastic relaxation rather than deterministic does provide some gain, especially that in case of a continuous state-space the computational overhead is not very large (Section 4.5).

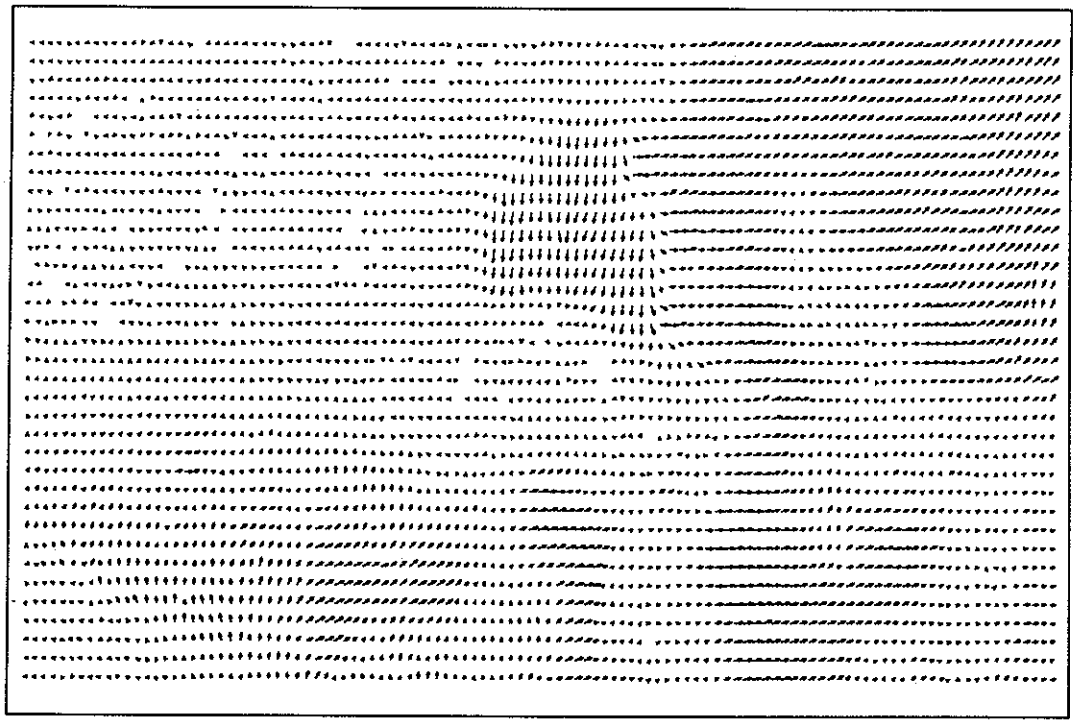
Observe that the deterministic estimate as well as the result provided by the Horn-Schunck algorithm have no vectors in the background, while the stochastic estimate has numerous small and differently oriented vectors. These vectors may be much smaller than the actual arrow size since only the zero length vectors are omitted while all other ones have identical size arrows. The background vectors in the stochastic estimate will eventually disappear with further reduction of the temperature to zero.

Fig. 8.4.d shows the deterministic estimate obtained from the noisy test image 2 for $\Lambda_{\mathbf{d}} = \Lambda_g$ and $\lambda_{\mathbf{d}}/\lambda_g$= 120.0 after 200 iterations. Compared to the corresponding stochastic estimate (Fig. 4.20.b), the deterministic result is quite good subjectively. Its mean squared error is much higher for the horizontal component, which is visible in the central part of the rectangle. In some other experiments it has been confirmed that the deterministic

† This results in Δt=0.5·T_g.



(a) Horn-Schunck algorithm



(b) deterministic approximation

Fig. 8.5 Estimates for the Horn-Schunck algorithm and the deterministic approximation to the continuous state-space MAP estimation for $\Lambda_{\mathbf{d}} = \Lambda_g + [0.5, 0.5, 0.5]^T$: test image 3, $\lambda_{\mathbf{d}}/\lambda_g$=20.0, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., 100 iter.

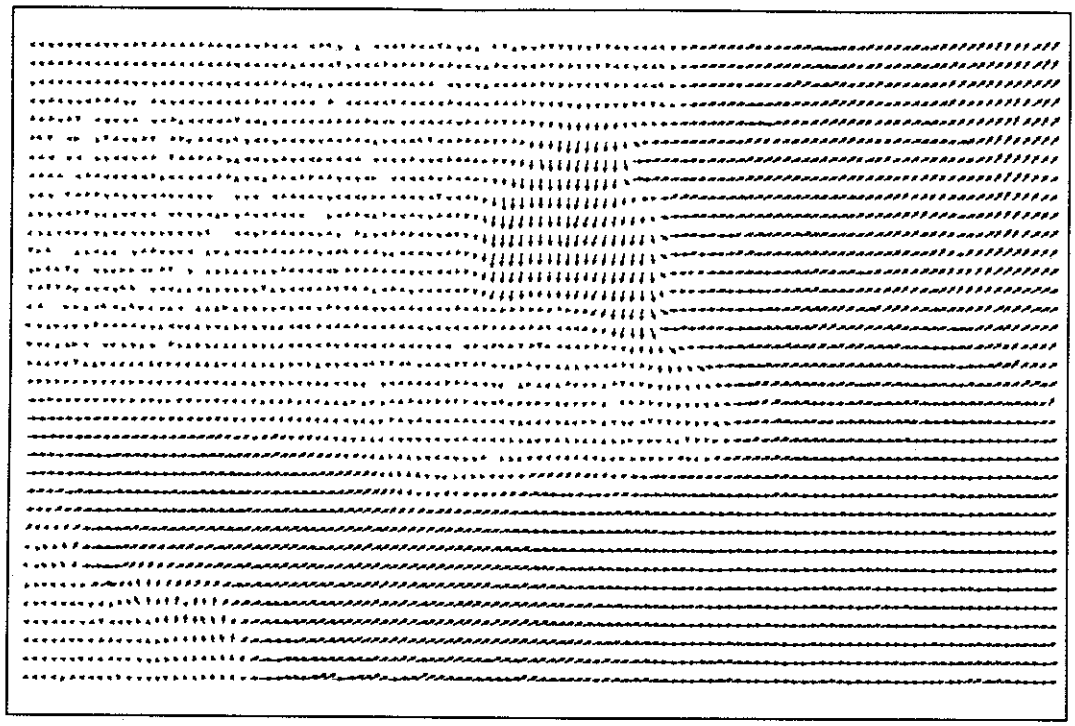


Fig. 8.6 Continuous state-space MAP estimate for $\Lambda_{\mathbf{d}} = \Lambda_g + [0.5, 0.5, 0.5]^T$: test image 3, $\lambda_{\mathbf{d}}/\lambda_g = 20.0$, neighb $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., $\mathbf{T}_0$=5.0, a=0.9944, 1000 iter.

algorithm performs quite well for high ratios $\lambda_{\mathbf{d}}/\lambda_g$ i.e., for significant smoothing. As far as the energies are concerned, the continuous state-space MAP estimate from Fig. 4.20.b has a higher energy than its deterministic approximation. However, after slowly reducing the temperature to 0.0001 over another 150 iterations, the stochastic algorithm attained an energy slightly lower than that of its deterministic counterpart†. This temperature reduction did not noticeably change the subjective quality of the vector field. Consequently, another very important observation can be made. After this extra temperature reduction, both energies (stochastic and deterministic algorithms) are similar, however subjectively the vector fields are quite different. This suggests that the objective function (3.20) is multimodal, and for the same value of this function quite different solutions can be obtained. Again, the stochastic relaxation algorithm has found a solution closer to the true motion field than the deterministic algorithm.

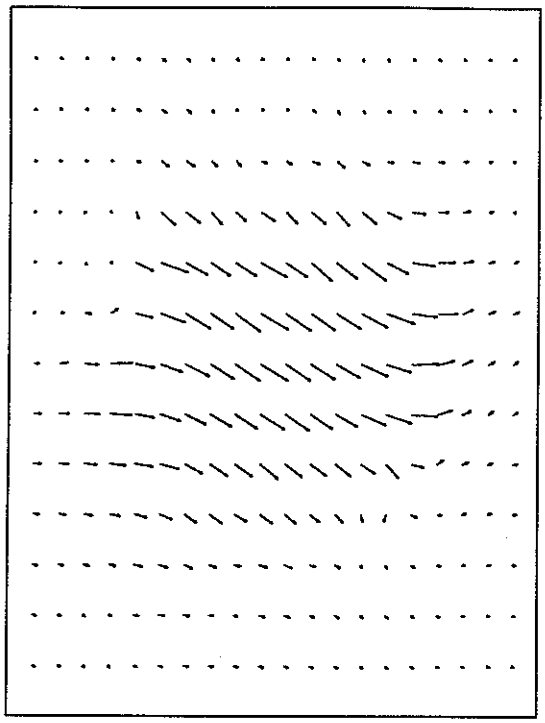
† Note that additional iterations in the deterministic algorithm will not drive the image energy down after a stationary point had been attained, while the stochastic method has another degree of freedom via the temperature, and unless it is already zero, further energy reduction is possible.

Similarly the three algorithms have been applied to the test image 3. Fig. 8.5.a shows the estimate resulting from the Horn-Schunck algorithm, and Fig. 8.5.b presents the result of deterministic approximation to the continuous state-space MAP estimation, both with the same parameters as for the test image 2. Note again that in the case of the Horn-Schunck method the motion vectors are overestimated at the edge of shirt sleeve and are underestimated in the uniform area to the right. The deterministic approximation performs slightly better: it is more uniform and has smaller edge effects. Both, however, are absolutely inferior to the continuous state-space MAP estimate via simulated annealing shown in Fig. 8.6. The estimate is smooth, and no edge effects are present. Again the deterministic estimate is characterized by a higher image energy (about 25%) than the stochastic MAP estimate.

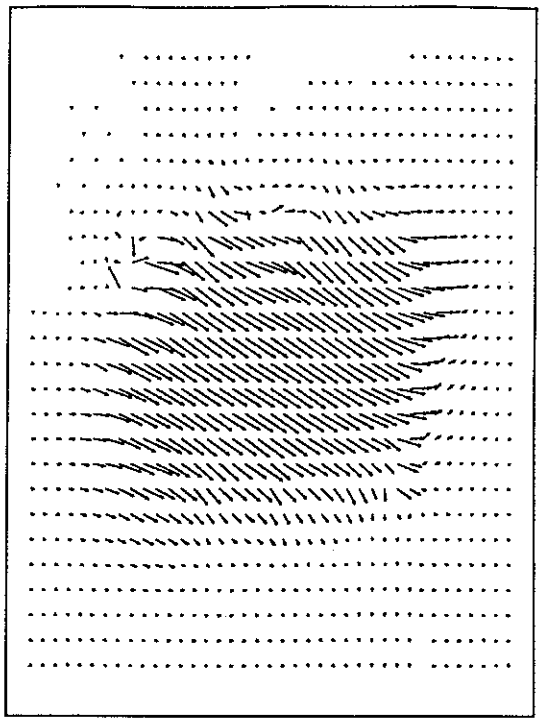
8.2.2.2 Hierarchical algorithm

Fig. 8.7 shows the deterministic approximations to the continuous state-space MAP estimation at three resolution levels for test image 2. First-order neighbourhood system, Keys bicubic interpolation and 200 iterations at each resolution level have been used. Observe that, compared to the stochastic hierarchical estimate from Fig. 5.7, it has a higher MSE for the horizontal component but lower error for the vertical one. In terms of the image energy it performs significantly poorer (about 50% higher energy). Subjectively it is very similar to the stochastic estimate except for the left top rectangle corner where the image breakup occurred.

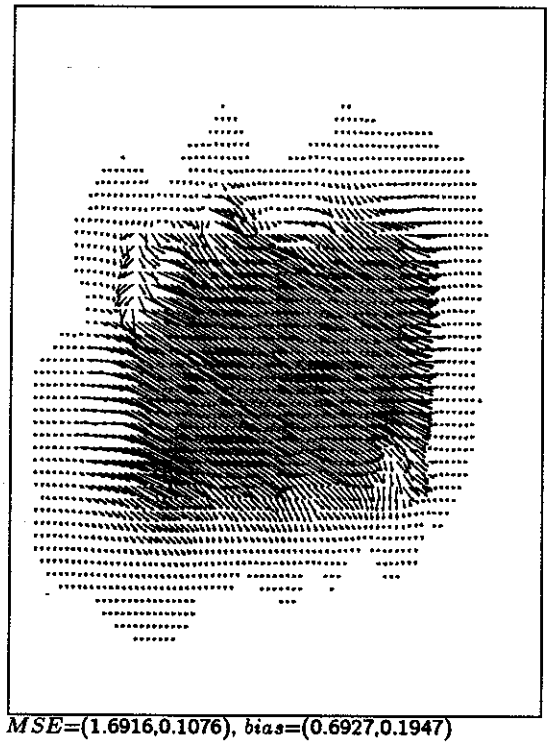
Figs. 8.8, 8.9 show estimates at three resolution levels obtained by the deterministic approximation implemented over a hierarchy of grids for test image 3. The estimates at levels κ=2,1 are very similar to the corresponding stochastic estimates from Figs. 5.8.b, 5.9.b. The full resolution estimate is also similar to the continuous state-space MAP estimate from Fig. 5.10, except for increased oversmoothing above the hand. The image energies are also comparable, however after another few dozens of iterations the energy of the stochastic estimate can be reduced below. Again, a clear superiority of the stochastic approach at single resolution level, is less pronounced when a hierarchy of resolutions is used.



(a) κ=2



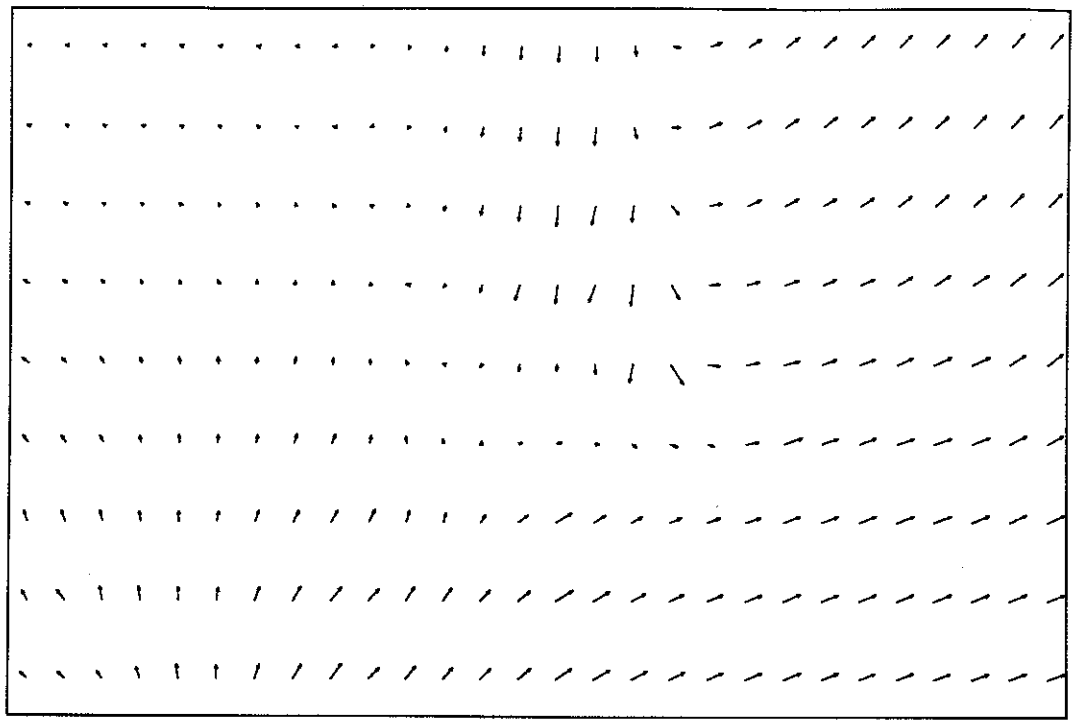
(b) κ=1



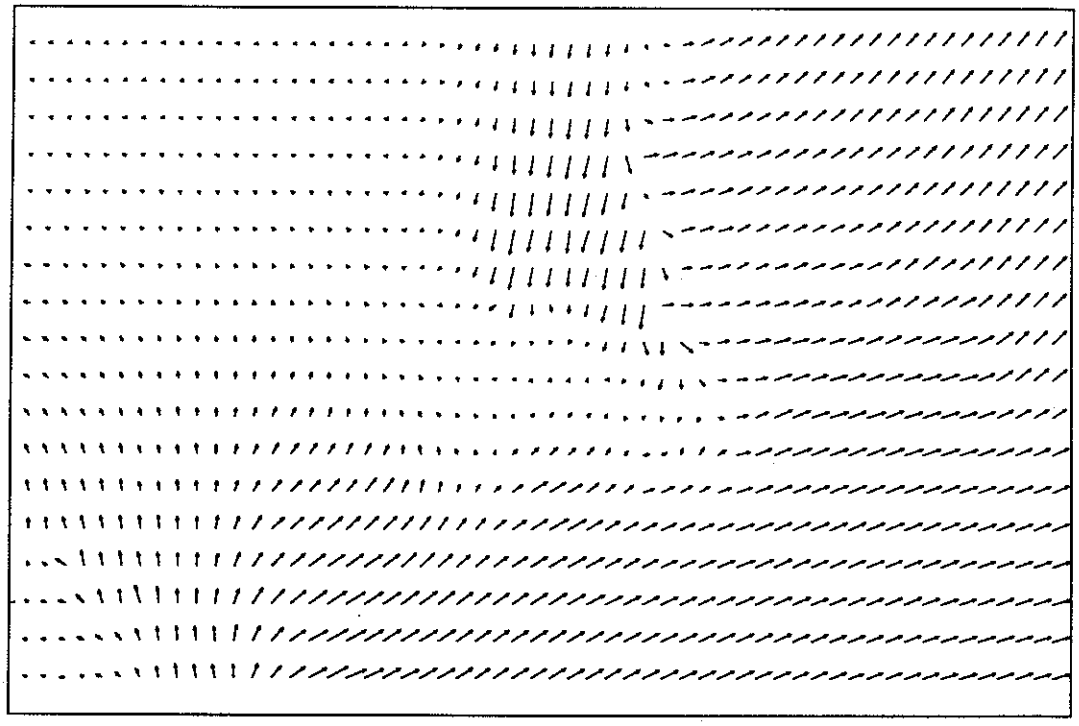
MSE=(1.6916,0.1076), $bias$=(0.6927,0.1947)

(c) κ=0

Fig. 8.7 Deterministic approximation to the hierarchical continuous state-space MAP estimation: test image 2, K_l=3, $\lambda_{\mathrm{d}}/\lambda_g$= 20.0, neighb. $\mathcal{N}_{\mathrm{d}}^{1}$, Keys bicubic interp., 200 iter. at each level.

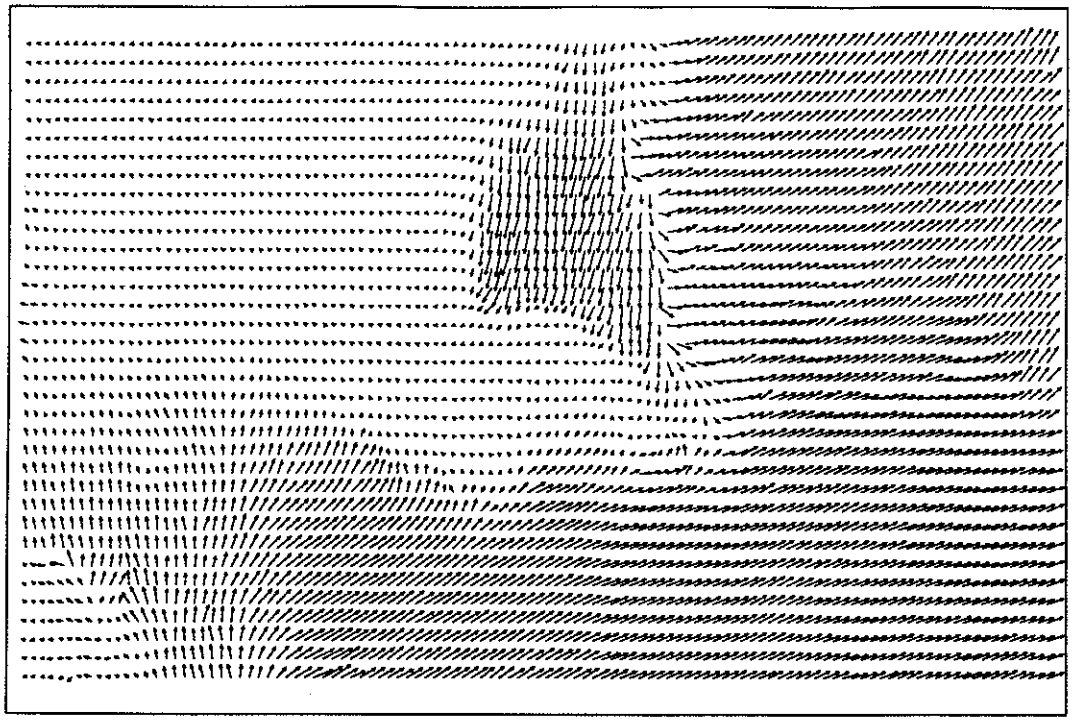


(a) κ=2



(b) κ=1

Fig. 8.8 Deterministic approximation to the hierarchical continuous state-space MAP estimation: test image 3, K_l=3, κ=2,1, $\lambda_{\mathrm{d}}/\lambda_g$=20.0, neighb. $\mathcal{N}_{\mathrm{d}}^{\mathrm{I}}$, Keys bicubic interp., 200 iter. at each level. Nyquist-like filtering.



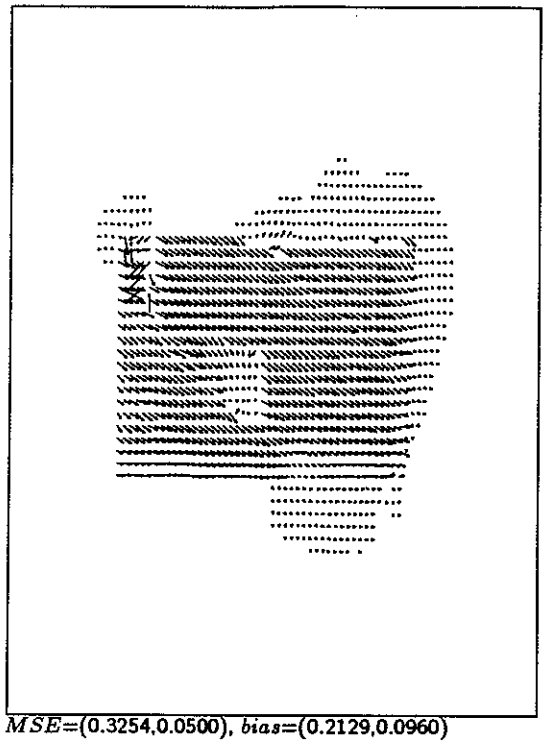
κ=0

Fig. 8.9 Deterministic approximation to the hierarchical continuous state-space MAP estimation: test image 3, K_l=3, κ=0, $\lambda_{\mathrm{d}}/\lambda_g$=20.0, neighb. $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interp., 200 iter. at each level. Nyquist-like filtering.

8.2.2.3 Algorithm incorporating motion discontinuities

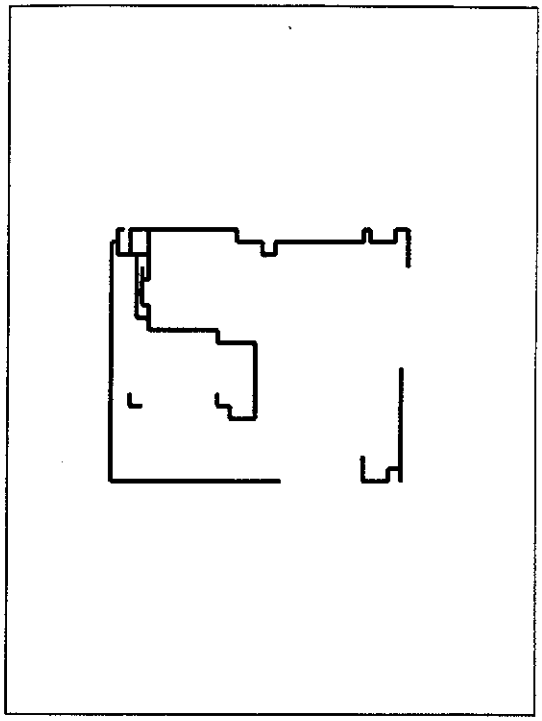
In this section some experimental results for deterministic algorithm with the two-layer motion model will be presented. In numerous experiments with this algorithm I have observed that to obtain results comparable with the stochastic estimates, the ratio $\lambda_l/\lambda_{\mathrm{d}}$ has to be significantly reduced. The values around 0.8-1.0 produced very few line elements. This may be explained by explicit averaging used in the deterministic algorithm. The continuous state-space MAP estimation uses similar averaging, but it also involves a randomness factor thus allowing switching line elements off and on, even if motion discontinuity does not quite allow it.

Figs. 8.10.a,.b show the deterministic estimate for test image 2. The parameters used are the same as for the stochastic estimate from Figs. 6.6.c,.d except for the ratio $\lambda_l/\lambda_{\mathrm{d}}$ which was 0.15. Note that both subjectively and in terms of the MSE the deterministic estimate is clearly inferior. The motion field in the center of the rectangle is not estimated

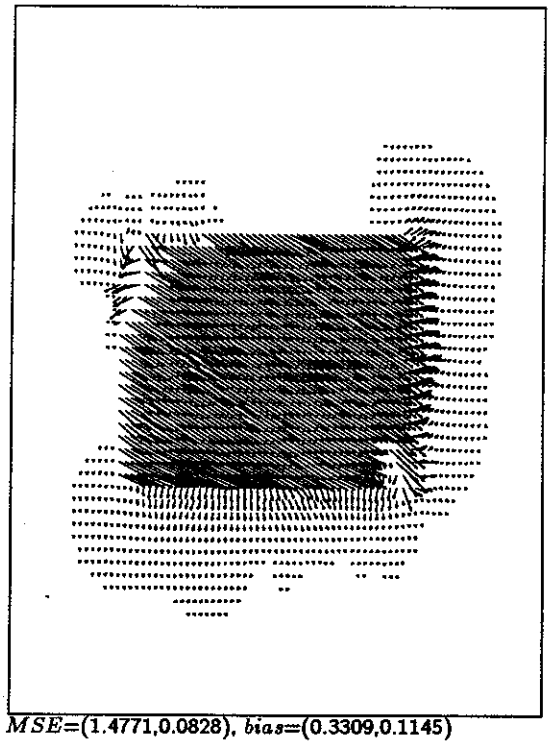


MSE=(0.3254,0.0500), $bias$=(0.2129,0.0960)

(a) single-level

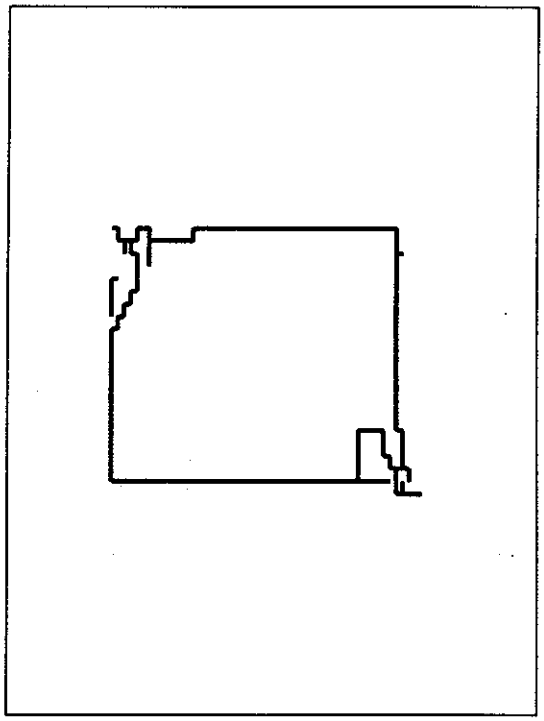


(b) single-level



MSE=(1.4771,0.0828), $bias$=(0.3309,0.1145)

(a) hierarchical, κ=0



(b) hierarchical, κ=0

Fig. 8.10 Deterministic approximation to the continuous state-space MAP estimation with piecewise smooth motion model: test image 2, $\lambda_{\mathbf{d}}/\lambda_g$= 20.0, $\lambda_l/\lambda_{\mathbf{d}}$= 0.15 (a,b) and 0.50 (c,d), neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., 1000 iter. (a,b) and 500 iter. at each level (c,d).

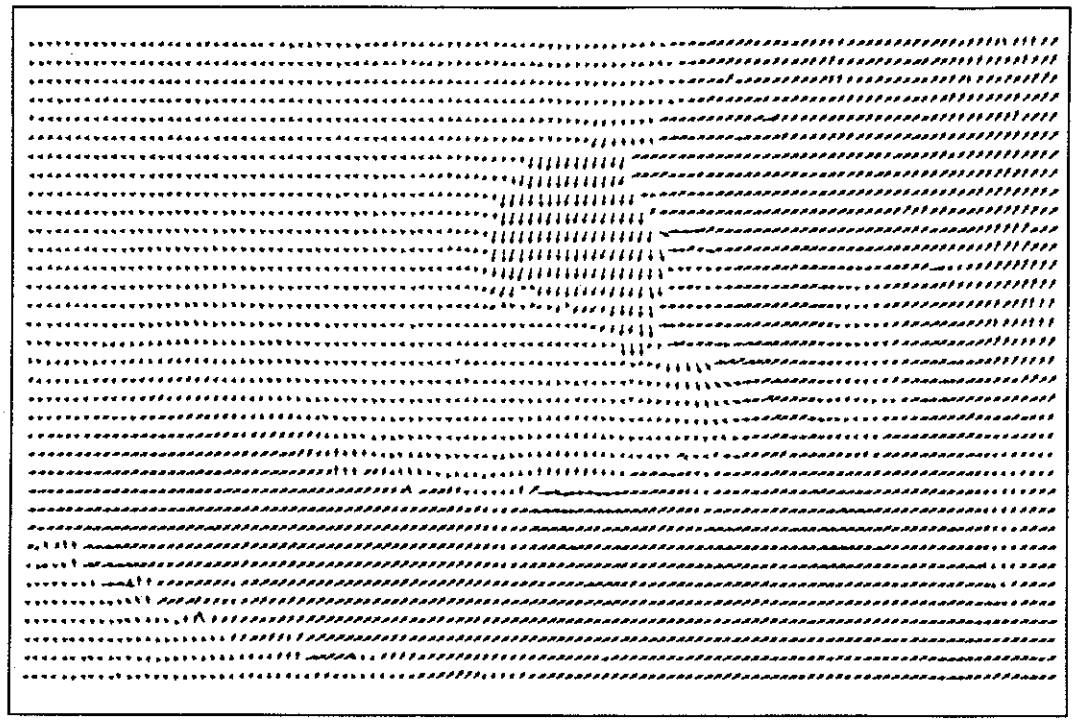
correctly, and in the left top corner the estimates are very inconsistent (image breakup). There are also numerous unended line segments.

Figs. 8.10.c,.d show the deterministic approximation to the continuous state-space MAP estimate with the piecewise smooth motion model over a hierarchy of resolutions (only the full resolution motion field is shown). Again identical parameters have been used as those for the stochastic estimate from Fig. 6.7 except for $\lambda_l/\lambda_{\mathbf{d}}$ which was 0.5. The estimate is characterized by a much higher mean squared error than the stochastic result. Subjectively, however, except for a more pronounced distortion in the troublesome left top corner of the rectangle, and some inconsistency in the bottom right corner, it is the same as the stochastic estimate, and very close to the true motion.

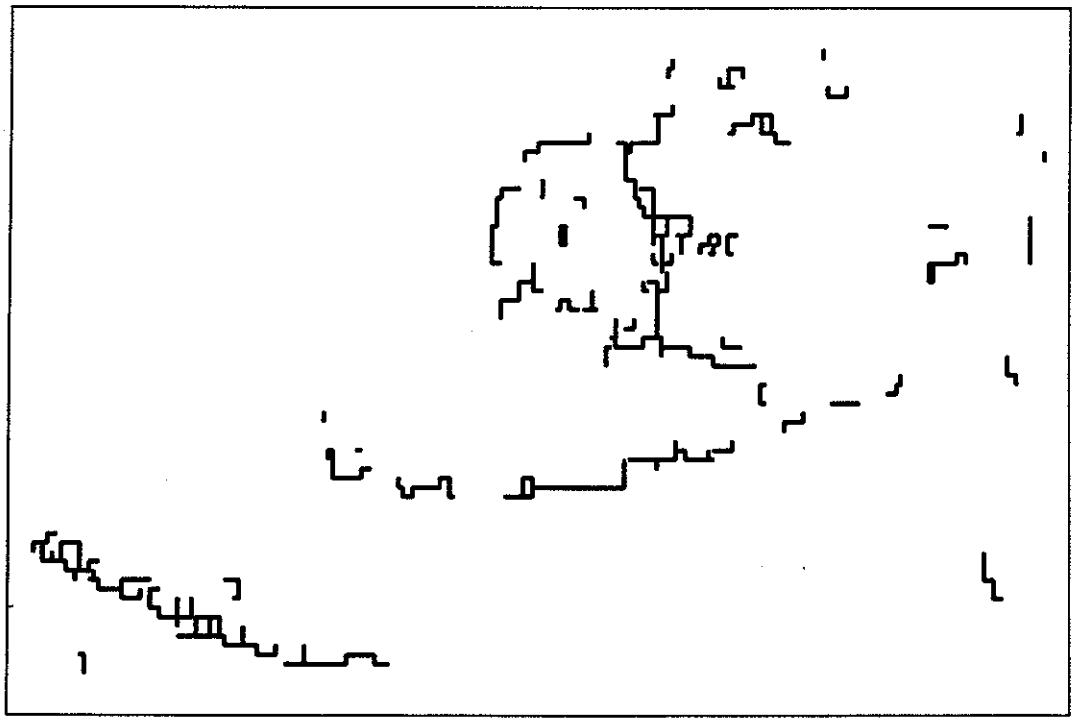
Fig. 8.11 presents the deterministic estimate of motion from test image 3 obtained with the same parameters as for the stochastic result from Fig. 6.9 except for the ratio $\lambda_l/\lambda_{\mathbf{d}}$ which was set to 0.15. Smaller values of this ratio resulted in many spurious line elements not reflecting motion boundaries, while larger values prevented formation of such boundaries where they should actually be. Compared with the stochastic result from Fig. 6.9 it is characterized by similar image energy, however the estimated motion boundaries are very fragmented and frequently unrelated to the true motion boundaries.

To complete the tests Fig. 8.12 shows the full resolution deterministic estimate with piecewise smooth motion model over a hierarchy of resolutions. Needless to say that the same parameters as those used for the result from Fig. 6.12 have been applied. This estimate is characterized by about 10% higher image energy than the stochastic solution from Fig. 6.12. The estimated motion boundaries are still fragmented, but subjectively this estimate is not very different from the stochastic result. The adjustment of ratio $\lambda_{\mathbf{d}}/\lambda_g$ towards larger values resulted in more continuous boundaries, but simultaneously many of them were missing.

Throughout the experiments I have observed that the deterministic algorithm is more sensitive to modifications of ratio $\lambda_{\mathbf{d}}/\lambda_g$ than the stochastic method, and also that obvious superiority of the stochastic algorithm at one resolution level is diminished once hierarchical estimation is used.

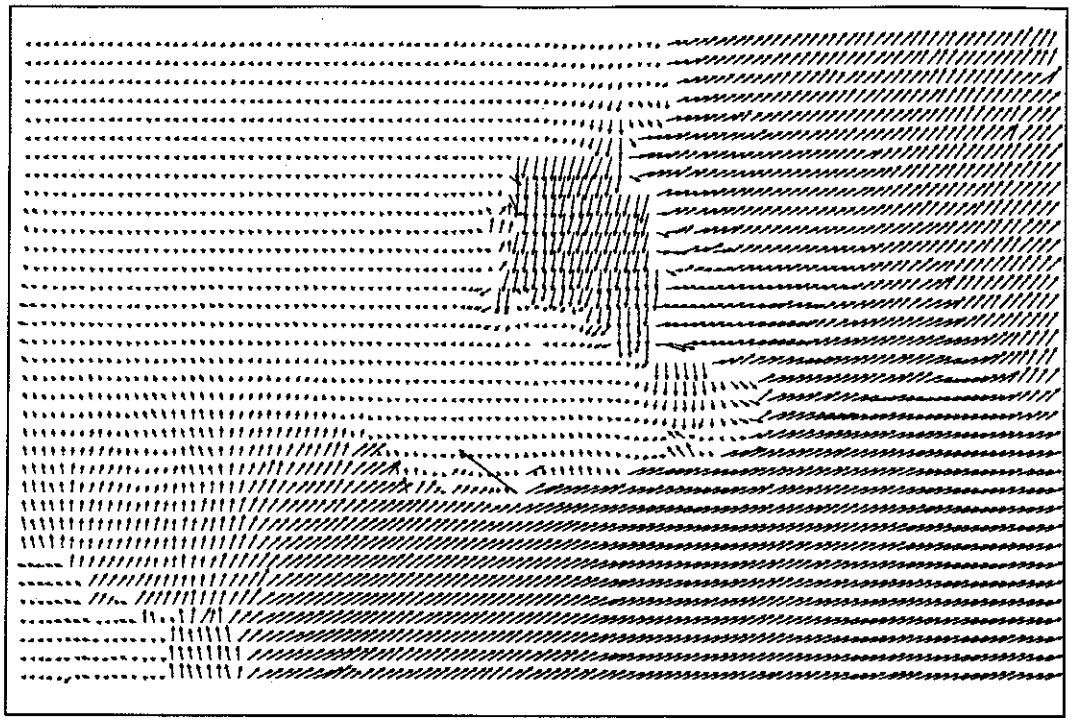


(a) displ. field

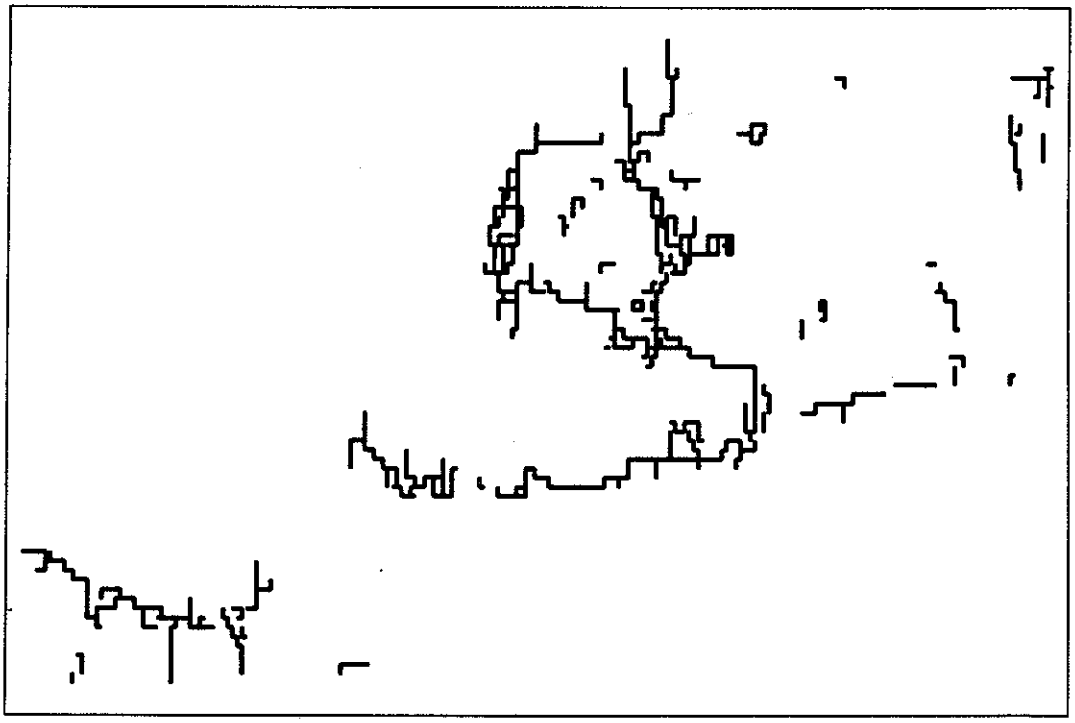


(b) line field

Fig. 8.11 Deterministic approximation to the continuous state-space MAP estimation with piecewise smooth motion model: test image 3, $\lambda_{\mathrm{d}}/\lambda_g$=20.0, $\lambda_l/\lambda_{\mathrm{d}}$=0.15, neighb. $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interp., 1000 iter.



(a) displ. field



(b) line field

Fig. 8.12 Deterministic approximation to the continuous state-space MAP estimation with piecewise smooth motion model over hierarchy of resolutions: test image 3, K_l=3, κ=0, $\lambda_{\mathbf{d}}/\lambda_g$=20.0, $\lambda_l/\lambda_{\mathbf{d}}$=0.20, neighb. $\mathcal{N}_{\mathbf{d}}^1$, Keys bicubic interp., 500 iter. at each level.

Chapter 9

MOTION-COMPENSATED INTERPOLATION

In previous chapters motion estimates resulting from different algorithms have been compared subjectively by evaluating the vector fields, and objectively, if the true motion had been known, by computing the mean squared error. The ultimate tests for the algorithms, however, are application specific. For example, in robotics it will be important how frequently a robot arm misses a target due to imprecise velocity estimates or incorrect localization of motion boundaries. In time-varying image processing and in image compression, the quality of the final displayed image is relevant. Thus, the quality of motion estimates will be assessed in this chapter by applying them to motion-compensated interpolation.

In the next section motion-compensated interpolation will be described. Then, two of the methods reported in earlier chapters will be applied to the test images 2 and 3, and the results will be compared.

9.1 MOTION-COMPENSATED INTERPOLATION

Motion-compensated interpolation is a technique which computes image intensity values specified on a sampling grid e.g., $\Lambda_{\mathbf{d}}$, from known images (image fields) specified on another lattice Λ_g, such that $\Lambda_{\mathbf{d}} \neq \Lambda_g$. This technique can be used in time-varying image compression via temporal subsampling, or in sampling structure conversion e.g., conversion between North American and European TV scanning standards.

If there is no motion in an image sequence, then a linear interpolation along the purely temporal direction will provide missing fields for $\Delta t \neq 0.0$. For missing lines or pels at

Δt=0.0, either spatial interpolation or purely temporal interpolation from the preceding and following images, can be used.

If there is motion in an image sequence, however, purely temporal interpolation fails to provide good results. For Δt=0.0 the missing lines and pels can be computed, at reduced resolution, by spatial interpolation, but for $\Delta t \neq 0.0$ motion must be taken into account.

Recall Fig. 3.1 where the definition of displacement field $\mathbf{d}_t$ is portrayed. If constant image intensity along motion trajectories is satisfied, then the intensity value at $(\mathbf{x}_i, t)$ should be identical to the intensity values at $(\mathbf{x}_i - \Delta t \cdot \mathbf{d}(\mathbf{x}_i, t), t_-)$ and $(\mathbf{x}_i + (1.0 - \Delta t) \cdot \mathbf{d}(\mathbf{x}_i, t), t_+)$. Obviously in practice such equality holds only approximately, and if illumination as well as occlusion effects are not negligible the error may be quite large. Assuming, however, that the intensity values at both ends of the displacement vector $\mathbf{d}(\mathbf{x}_i, t)$ are similar, the intensity value at $(\mathbf{x}_i, t)$ should be closely related to such intensities. The following linear interpolation along displacement vectors $\mathbf{d}(\mathbf{x}_i, t)$ will be used in subsequent experiments:

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

where $\hat{g}$ is the new interpolated intensity field defined over $\Lambda_{\mathbf{d}}$.

9.2 EXPERIMENTAL RESULTS

To test motion-compensated interpolation the algorithm has been applied to two 24-field image sequences which had been previously temporally subsampled. These sequences will be called the input sequences. Thus, motion fields are recovered from temporally sparse data, and used in the interpolation algorithm. Three sequences are generated for subjective evaluation:

1. interpolated (output) image sequence: for t such that $\Delta t \neq 0.0$ motion compensated interpolation is used, otherwise input fields are directly copied to the output sequence,
2. displaced field difference sequence: for t such that $\Delta t \neq 0.0$ displaced pel differences $\tilde{r}$, offset by 128 to accommodate positive and negative errors, are written at appropriate location $\mathbf{x}_i$, otherwise a field with fixed luminance of 128 is substituted,
3. original/interpolated difference sequence: difference between the input and output sequences is created.

The sequence of interpolated fields is used for direct comparison with the original (input) sequence. Sometimes it is difficult, however, to precisely locate the errors, hence also the difference sequence is created. This sequence identifies areas where an algorithm fails to faithfully reproduce the original image sequence. The DPD sequence shows the spatial distribution of image energy U_g which is an indicator of a mismatch between the images from which motion had been estimated. In an ideal case, when full correspondence between the images exists, it should be zero. Of course in reality, due to illumination effects, occlusion problems or violated assumptions of linear motion trajectory, local linearity of intensity function etc., such zero-valued field does not occur. To enhance the visibility of the errors, the DPD and (original/interpolated) difference fields presented in the following sections have been multiplied by 3.0 before adding 128.

The following algorithms have been used to generate motion field sequences, and then applied to motion-compensated interpolation:

1. deterministic approximation to the continuous state-space MAP estimation (or the modified Horn-Schunck algorithm) over a hierarchy of resolutions: K_l=3, $\lambda_{\mathrm{d}}/\lambda_g$=(20.0,12.0,10.0), first-order neighbourhood system $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interpolation, Nyquist-like filtering, 200 iterations at each level,
2. continuous state-space MAP estimation via stochastic relaxation with piecewise smooth motion model over a hierarchy of resolutions: K_l=3, $\lambda_{\mathrm{d}}/\lambda_g$=(20.0,12.0,10.0), $\lambda_l/\lambda_{\mathrm{d}}$=1.8 (test sequence 2) or 1.0 (test sequence 3), α=(10.0,3.0,1.0), first-order neighbourhood system $\mathcal{N}_{\mathrm{d}}^1$, Keys bicubic interpolation, Nyquist-like filtering, $\mathbf{T}_0$=(1.0,2.0,4.0), a=0.980, 200 iterations at each level.

In the following sections the above algorithms will be called 1 and 2, respectively. The vector fields obtained by both algorithms are very similar to those reported in Sections 6.7.2, 6.7.3 and 8.2.2.3, and will not be presented here.

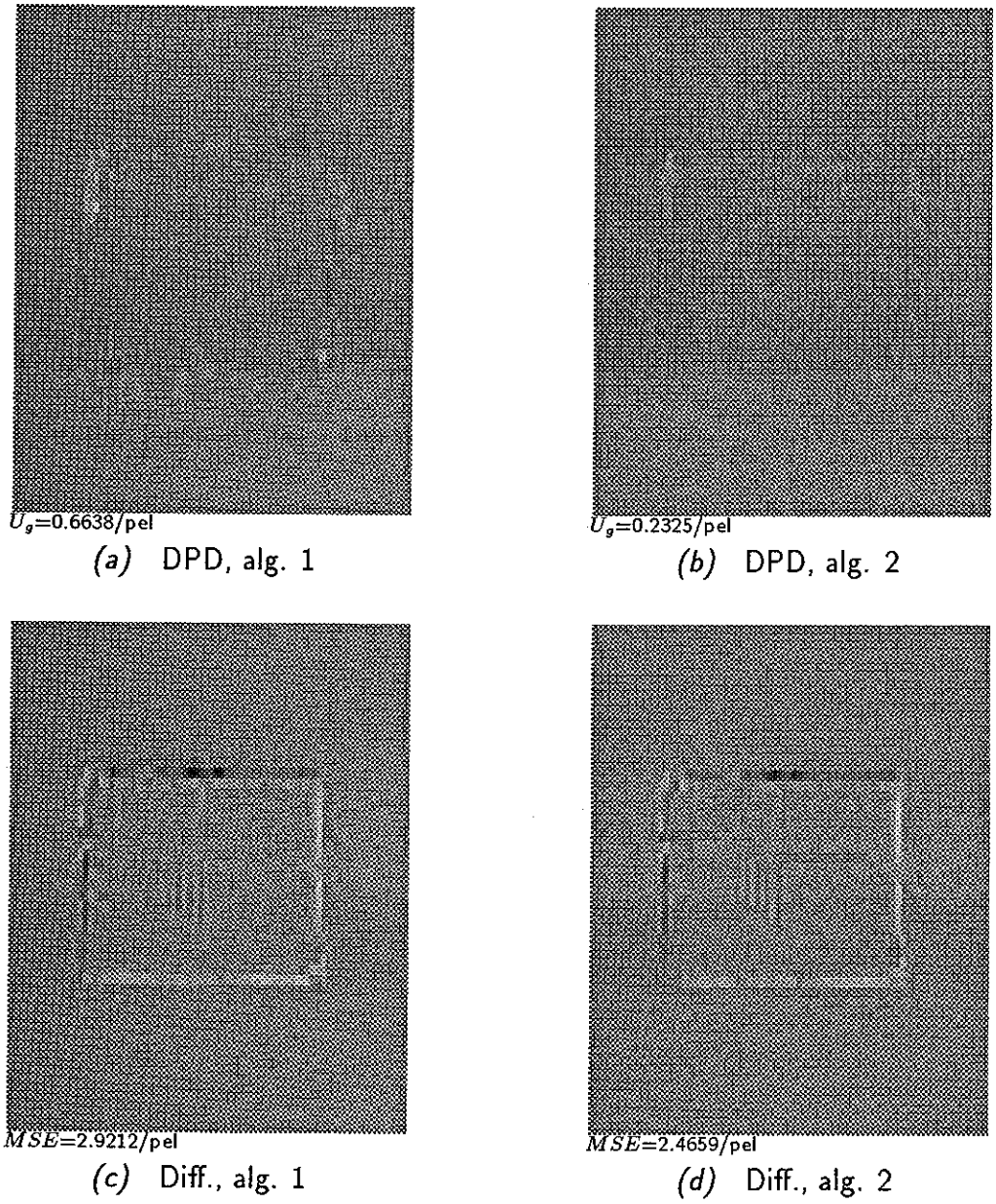
9.2.1 Test image 2

In order to simulate realistic data (no perfect matching) the temporal subsampling factor has been chosen to be 6 i.e., T_g=6τ_{60}. Thus from every 2 image fields 5 motion fields are computed.

Figs. 9.1.a,.b show the displaced field differences for both algorithms for motion fields number 2. The image energies U_g per pel are given below the images. Note the significant reduction of the DPD error for algorithm 2 compared to algorithm 1, both subjectively and in terms of U_g. Algorithm 1 has produced significant DPD errors which are distributed uniformly across the rectangle area, and visible in particular at the rectangle boundaries. There are rather few significant DPD errors produced by algorithm 2, and they appear in the image breakup area (top-left corner of the rectangle) and in the occlusion area (along the right edge). Unlike in the case of algorithm 1, the rectangle edges are hardly visible. This is exactly what the line process was supposed to provide: reduction of mismatch enforced by oversmoothing of vectors at motion boundaries. Matching constrained by spatial vector smoothness no longer has to be performed. With sufficient information in the data, this smoothness is broken and unconstrained matching can be carried out.

Figs. 9.1.c,.d show the original/interpolated difference fields number 2 for both algorithms. The mean squared error is displayed below both images. The MSE for the algorithm 2 is lower than that for the algorithm 1. This is also confirmed in the images. Note a spurious pattern extending from the rectangle boundary into the stationary background in Fig. 9.1.c. This pattern is due to oversmoothed vectors extending into the background where motion is absent. Observe that this pattern disappears completely in Fig. 9.1.d along the top and bottom edges, and is significantly reduced along the left and right edges. Due to the line process, a sudden transition from vectors inside of the rectangle to no vectors outside is permitted, reducing errors enforced by a smoothness-constrained matching. More pronounced effect in the vertical direction can be explained by a higher vertical smoothing resulting from twice larger inter-line than inter-pel distance (already discussed in Section 4.8).

When evaluating the moving pictures, with no error boost, the effects described above have been less pronounced. The reduction in the DPD error was still quite dramatic, however the original/interpolated difference was almost the same for both algorithms. In fact, when comparing both interpolated sequences with the original, certain artifacts remained in both, and there was no visible difference between the two algorithms except for the occlusion areas. These areas contained numerous annoying errors, but neither of the inter-



(a) DPD, alg. 1 *(b)* DPD, alg. 2

(c) Diff., alg. 1 *(d)* Diff., alg. 2

Fig. 9.1 Displaced field difference (a,b) and interpolated/original difference (c,d) for test image 2 (field number 2).

polated sequences could be considered better. These observations suggest that in spite of a significant improvements in the DPD and difference (boosted) errors, the problem of vector oversmoothing is not critical. Since oversmoothing occurs at the motion boundaries, and these usually coincide with occlusion borders, rather gentle oversmoothing errors tend to be masked by more pronounced occlusion errors. Errors due to oversmoothing are gentle, because if sufficient data structure (e.g., strong texture) is present in the background, the

vector coupling is less strong (either reduced displaced pel difference $\tilde{r}$ in the local energy (4.13) or increased update term in iterative equation (4.20)).

9.2.2 Test image 3

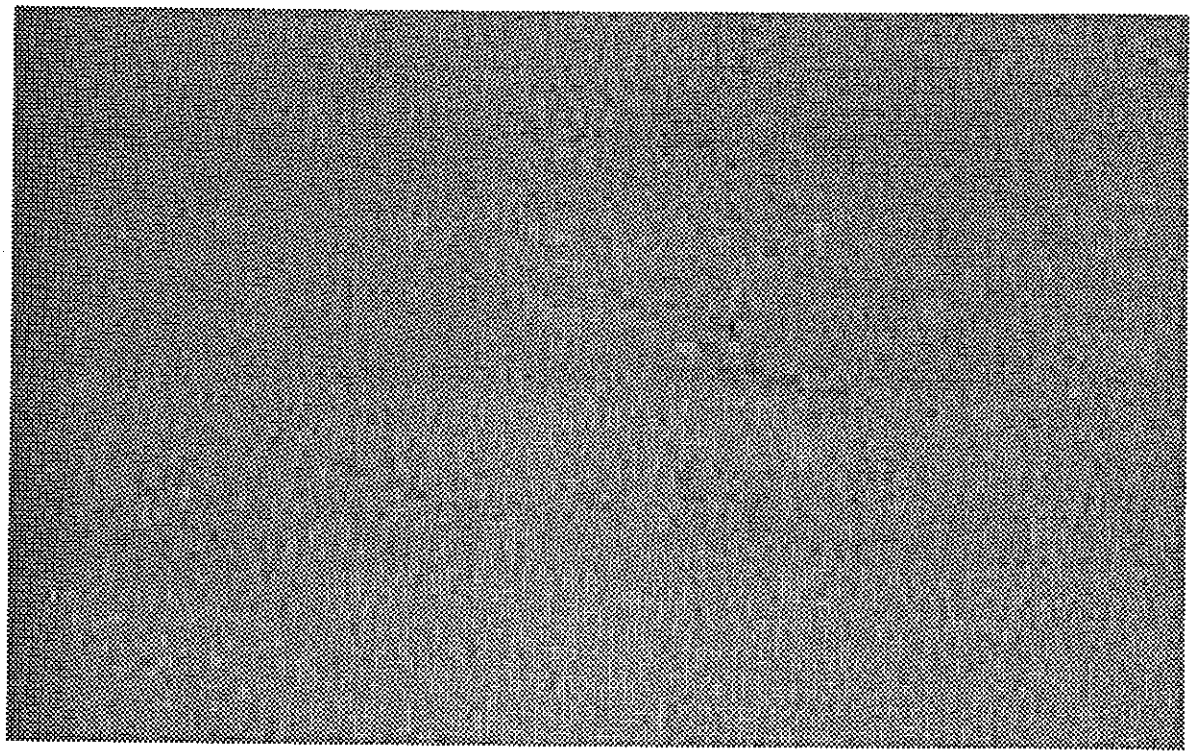
For test image 3 the temporal subsampling rate was chosen to be 4:1 i.e., T_g=4τ_{60}. Consequently, from every 2 image fields 3 motion fields are computed.

In earlier work [56] I have compared the quality of interpolated image sequences 3 and 4 compensated for motion using algorithm 1, with simple field repetition and with linear interpolation in the temporal direction. The sequences using field repetition were characterized by very annoying motion jerkiness. This effect was significantly reduced in the linearly interpolated sequences, but blur and multiple edges remained. The sequences produced using algorithm 1 were clearly superior to the above non-compensated schemes and also when compared with simple block matching algorithms.

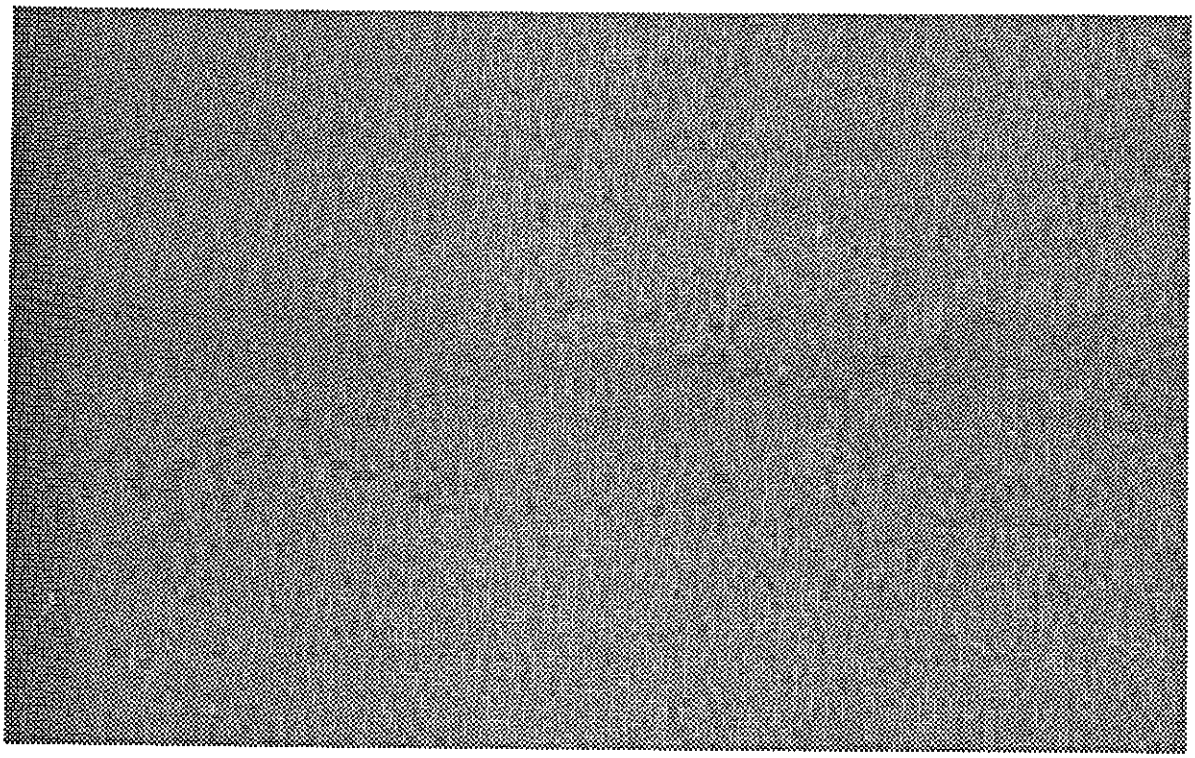
Fig. 9.2 shows the displaced field differences for both algorithms for motion fields number 1. The image energies U_g per pel again confirm significant reduction of the DPD error for algorithm 2 compared to algorithm 1. Such reduction can be also observed in the images, where the errors for algorithm 1 are more pronounced around the hand and at the face contours. In particular, there is a significant improvement in the area just above the palm of the hand, which is suggested to move according to algorithm 1 while it is almost stationary according to algorithm 2. Note also that algorithm 2 managed to avoid significant errors at the motion boundaries due to the hand and face movements.

Fig. 9.3 shows the differences between the original and interpolated fields number 1. Note that there is very little difference between the two algorithms. Moreover, algorithm 2 has even introduced few more pronounced errors at the occlusion borders of the palm of the hand. This is confirmed by a slightly higher mean squared error for algorithm 2. Subjective evaluation, however, has revealed no clear preference between the two algorithms. The interpolated sequences looked alike. In spite of some differences compared to the original sequence, both presented very good image quality. Major artifacts could be observed in the occlusion and newly exposed areas. The improvement due to the two-layer motion model, very striking when viewing the DPD sequence, had basically no impact on the quality of

the interpolated sequence. As remarked in the previous section, any improvements due to a reduction of oversmoothing are either minor or masked by nearby occlusion artifacts. Again, if the background structure were stronger such that one might hope for visible improvements, then less oversmoothing would have been present due to such a structure, and the occlusion errors would have been much more pronounced. In any event, the interpolation errors due to motion field oversmoothing are not a major source of distortion. Errors due to occlusion areas are more important, and have to be given attention if one hopes for improvements to motion-compensated interpolation.

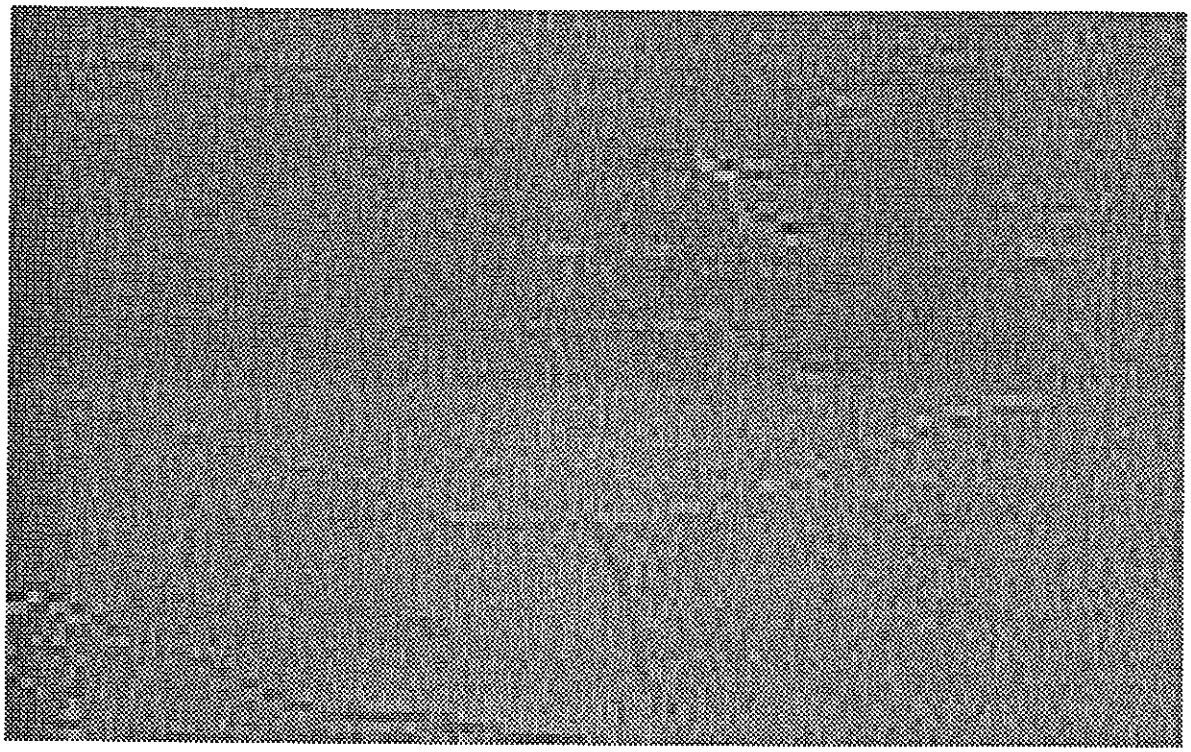


(a) algorithm 1 (U_g=3.4043/pel)

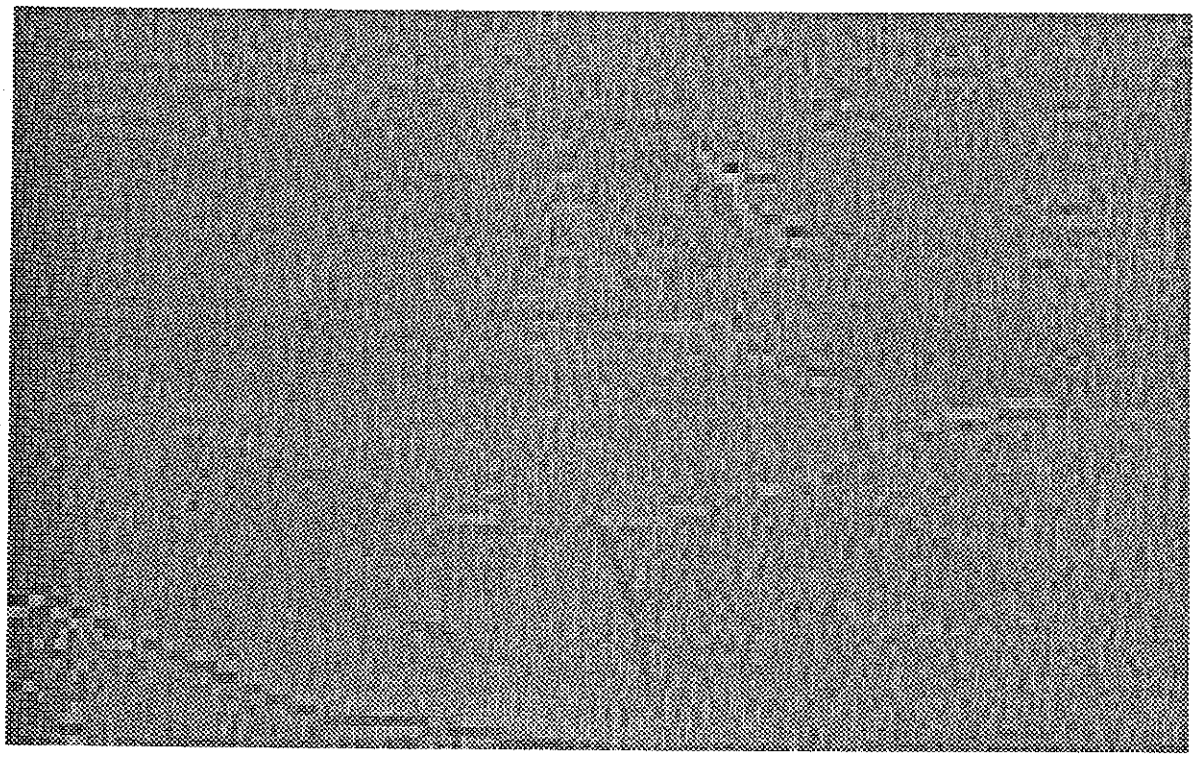


(b) algorithm 2 (U_g=2.8129/pel)

Fig. 9.2 Displaced field difference for test image 3 (field number 1).



(a) algorithm 1 (MSE=8.5397/pel)



(b) algorithm 2 (MSE=8.7770/pel)

Fig. 9.3 Original/interpolated difference for test image 3 (field number 1).

Chapter 10 CONCLUSIONS

10.1 SUMMARY

In this thesis I have presented a stochastic approach to the estimation of 2-D motion from a sequence of spatio-temporally sampled time-varying images. The Maximum *A Posteriori* Probability (MAP) and the Minimum Expected Cost (MEC) criteria have been used to formulate the estimation problem. Both criteria require the knowledge of the *a posteriori* probability $P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+})$ to specify a cost functional for optimization. This probability can be factored using Bayes rule into terms which are representative of the models used. The observation model, relating the underlying and the observed images, and expressed as additive Gaussian noise, is combined with the structural model assuming constant image intensity along motion trajectories. As the displacement field model a vector Markov random field has been used, resulting in a Gibbs *a priori* distribution. Combined via Bayes rule, these distributions provided a regularized cost functional to be minimized. The minimization problem, involving several thousands of unknowns, has been solved using simulated annealing. Two stochastic relaxation algorithms: the Metropolis algorithm and the Gibbs sampler, have been used to generate Markov random field samples according to the *a posteriori* probability. It was concluded that the Gibbs sampler had provided slightly better performance, hence it has been used subsequently. Also the spatial interpolation of images to provide samples at non-integer locations has been investigated. The importance of C^1-continuous image models, especially for spatio-temporal gradient techniques has been demonstrated.

First, the MAP estimation algorithm for vectors defined over a discrete state-space has been derived. This algorithm was an extension of methods proposed for image restoration. Within the motion estimation context, it was identified as a pel matching algorithm with motion smoothness constraint. Then, the possibility of solving the MAP estimation over a continuous state-space has been investigated. Linearization of the displaced pel difference $\tilde{r}$ resulted in a Gaussian distribution driving the Gibbs sampler, and thus provided a much more efficient spatio-temporal gradient algorithm. This algorithm proved to be a stochastic extension of Horn-Schunck-type methods. Both the discrete and the continuous state-space stochastic MAP estimation resulted in very good motion estimates.

The MEC estimation problem has been solved by applying the *Law of Large Numbers for Markov chains*. In practice, an iteration-wise averaging has been performed without the need for an annealing schedule, which is an important ingredient of simulated annealing. The performance of MEC estimation has been shown to be similar to that of the MAP estimation. In a noisy environment, however, the MEC estimator did not provide better performance than the MAP estimator, as found by Marroquin [62]. The reasons could be twofold: 1. the state-spaces used here were much larger than in Marroquin's experiments, and thus might have needed many more iterations to show differences between the two algorithms, 2. the SNR value was much higher here.

To perform efficient estimation of large displacements, a hierarchical non-recursive approach has been proposed for the MAP estimation criterion. An important theorem for 1-D shift-invariant signal matching has been proved, which relates the filtering operations on the data and on the cost functional. Extrapolated to 2-D motion estimation, this theorem shows why low-pass prefiltering of images improves reliability of motion estimation. Two even pyramids have been used in the hierarchical approach: a constant-width image pyramid, which increases precision of interpolated intensities at a cost of increased memory size, and a standard variable-width displacement field pyramid. Simple repetition as well as bilinear interpolation have been tried to interpolate displacement fields between subsequent resolution levels with no difference in performance. The hierarchical approach has been shown to give good quality motion estimates on some test images, however it disclosed substantial problems on the other. In general, it can be concluded that the occlusion

and exposure effects present in an image are enhanced by the filtering operation. Even if a good matching (explicit or via spatio-temporal gradient) can be performed at the full resolution level, this may not be the case at higher levels of the image pyramid. Violation of the assumption of constant image intensity along motion trajectories is spread by the filtering, thus disallowing correct estimation, which may not be recoverable at lower levels of the pyramid. Another problem characteristic of hierarchical methods is motion field oversmoothing across the motion boundaries.

The problem of estimation across motion boundaries has been tackled by introducing a two-layer motion model. The original vector MRF model for motion fields has been augmented with a coupled binary MRF modeling motion discontinuities. Another MAP estimation criterion has been proposed, and a three-term regularized cost functional derived. The motion discontinuity model has been based on four-, two- and one-element line cliques to encourage formation of smooth, continuous boundaries at locations characterized by a substantial spatial gradient. This model has been shown to improve the subjective quality of motion fields, as well as to reduce the mean squared error (this criterion applies only to the test sequences with synthetically generated motion).

Throughout the research I have observed that other cues (than intensity) may be useful in motion estimation. The MAP criterion has been extended to also include the colour. Both the one-layer and the two-layer motion models using colour have shown substantial improvements in performance compared to the estimates based on intensity only.

The advantages of using stochastic instead of deterministic relaxation have been also investigated. The MAP estimation criterion over a discrete state-space has been optimized by exhaustive search for the maximum of the local marginal conditional probabilities. This method has been shown to be inferior to the stochastic optimization. In the case of a continuous state-space, the Gauss-Newton optimization has been used. By performing instantaneous freezing to obtain a deterministic algorithm, the single-resolution method has been demonstrated to be a modification to the Horn-Schunck method [41], while a hierarchical approach has been shown to be related to the methods proposed by Glazer [31] and Enkelmann [22]. It has been also found that the deterministic approximation with the

two-layer model is very similar to the method proposed by Hutchinson *et al.* [44]. Note that all these methods are special cases of a general stochastic MAP formulation and solution.

In numerous experiments it has been observed that the stochastic solution methods provide superior performance at one resolution level. The difference in performance is significantly diminished, however, once a hierarchy of resolutions is used. This observation is in agreement with the theorem from Chapter 5 extrapolated to 2-D space-variant motion estimation. Since low-pass filtering smooths-out the objective function, with sufficient amount of filtering this function may become nearly unimodal. In such a case any optimization technique will attain the local (and hence the global) minimum at this resolution level. Repetitive estimation at subsequent resolution levels, starting from the previous-level estimate should easily locate the global minimum at the full-resolution level. No stochastic search should be needed. These remarks are consonant with observations of Simchony *et al.* [81]. They used the GNC algorithm for the image restoration problem, and obtained similar results to those produced by simulated annealing. As described earlier, hierarchical methods can be viewed as examples of the GNC algorithm.

Finally, two estimation algorithms proposed in this thesis have been applied to motion-compensated interpolation. Both (hierarchical) algorithms have performed very well producing good quality interpolated sequences. The stochastic algorithm (2) incorporating the two-layer motion model has performed substantially better in terms of the displaced field difference error, both subjectively and objectively, than the deterministic algorithm (1) with globally smooth motion model. The reduction of oversmoothing error in the interpolated sequences has been minor, however, compared to the occlusion artifacts. Since the error due to motion oversmoothing is never very severe, and since it is always located close to occlusion borders, it tends to be masked by occlusion effects and is not a major source of distortion. Concluding, the two-layer model did not provide improvements to the quality of motion-compensated interpolation. Not everything is lost, however. The line elements are usually formed at the motion boundaries, and as such frequently surround the occlusion areas, or even partition them into smaller sub-areas. The vectors in such areas are distinctly different from their neighbours, which is allowed by the line elements. Since there is no correct match in the occlusion area anyway, these vectors will usually provide substantial

DPD error. Thus, if a vector belongs to an isolated (by line elements) area, and if its DPD is substantial, it is very likely that this vector is trying to find a match for a pel which simply disappeared. Knowing that, different processing can be performed at the interpolation stage. For example, a sought after pel can be taken from the preceding or the following image at appropriate spatial location, rather than averaged from the two images. Identification of the motion boundaries thus can be helpful in motion-compensated interpolation, as well as in robotics or computer vision. Note also that due to a significantly reduced DPD error such an algorithm can further reduce the bit rate in motion-compensated predictive coding schemes.

10.2 CONTRIBUTIONS

This thesis has contributed to the theory of 2-D motion estimation, as well as to practical implementation of the algorithms. The major contributions of this work can be summarized as follows:

1. 2-D motion estimation has been formulated as a Bayesian estimation problem. The formulation has been extended to incorporate a hierarchy of resolutions. The basic motion model has been augmented with another layer to include motion discontinuities.
2. The influence of various parameters on the behaviour of the proposed stochastic relaxation algorithms has been identified through numerous experiments.
3. Various existing motion estimation methods have been shown to be special cases of the stochastic solution when instantaneous freezing is applied. Consequently, the stochastic formulation and solution can be viewed as a generalization of various deterministic methods.
4. It has been shown for 1-D shift-invariant signal matching, that low-pass filtering aids in obtaining unimodality of the objective function, and thus helps in reliable shift estimation.
5. It has been demonstrated that at single resolution level the stochastic methods offer substantially better performance than the corresponding deterministic algorithms. This improvement is reduced, however, when a hierarchy of resolutions is used. It has been shown that the stochastic and the hierarchical approaches are two effective but conceptually different methods of finding the global optimum of a multimodal function.

6. It has been shown that the colour information in an image should not be disregarded as a cue for motion estimation. Also the importance of using a C^1-continuous luminance and chrominance models has been indicated.
7. It has been demonstrated that image matching error due to motion oversmoothing across motion boundaries can be substantially reduced by using a piecewise smooth motion model. It has been also concluded that such oversmoothing is not a dominant source of artifacts in motion-compensated interpolation. The most severe errors are due to interpolation across occlusion borders. It has been suggested to use line elements in identification of such borders.

10.3 OPEN QUESTIONS

10.3.1 Regularization parameters

The constants λ_g, λ_{d} and λ_l have been chosen here empirically. It may be important to choose these constants optimally. It is not clear how to estimate the DPD model variance necessary to compute λ_g. Also the other λ's are not easily computable. Training data, extensively used in texture segmentation, does not apply since both motion and motion boundaries are not observable ! Estimation of the λ's seems to be a challenging task.

10.3.2 Hierarchical approach

Three major problems are related to the hierarchical approach. Firstly, it is not clear what kind of filtering should be applied to construct the image pyramid (efficiency issues aside). In this thesis Nyquist-like filters have been proposed to minimize aliasing errors after spatial subsampling. They provided better performance for this particular application than the Gaussian filters which tend to unnecessarily oversmooth the data. Following the comments on the theorem from Chapter 5, it may be asked whether there exists an optimal filter for a given estimation method, such that in a K_l-level image pyramid the lowest-resolution cost functional will be unimodal.

Secondly, the filtering operation should be adaptive to avoid spreading of occlusion and exposure effects at the top of the image pyramid. Fixed filtering used so far enhances the

occlusion effects by contributing to violation of the constant image intensity along motion trajectories in the vicinity of the occlusion areas.

Thirdly, the hierarchical approach seems to be related to the GNC algorithm proposed by Blake and Zisserman [12]. It may be profitable to use GNC-type algorithms for motion estimation.

10.3.3 Piecewise smooth motion model

The motion boundary model investigated here was very simple. Multi-level instead of binary line cliques could be implemented to identify directions of line elements for better rendition of motion boundaries. Also the single-element line clique relating the motion boundary to an intensity edge could be improved, especially for temporal positions different from image positions.

10.3.4 Motion-compensated interpolation

The very important problem of occluded and newly exposed areas in image sequences remains unresolved. It seems, however, that the piecewise smooth motion model may be helpful in identification of such areas. Vectors unable to attain low DPD errors and isolated by line contours from other vector patches where the DPD error is small, are very likely to belong to occluded or newly exposed areas. If they can be identified reliably, then appropriate processing can be applied to reduce errors due to averaging across occlusion borders.

10.3.5 Multiple-frame processing

This thesis has addressed the problem of motion estimation and motion-compensated interpolation based on two images only. Motion estimation, and in particular identification of occlusion areas, can be done much more reliably by using multiple image frames (fields). Accumulating the history of objects over several frames should allow easier identification of what is new in the image and what disappeared. As remarked earlier this kind of information is of crucial importance for occlusion-adaptive processing.

10.3.6 Other structural models

Here, only the constant image intensity structural model with respect to luminance and two chrominances has been developed. It should be possible to extend this approach to constant intensity gradient (or even higher-order derivative of the intensity). Also other cues than used here could be investigated, for example: contrast. It would be also interesting to compare Y-C1-C2 and R-G-B formats with respect to their applicability to motion estimation.

REFERENCES

[1] J.K. Aggarwal and R.O. Duda, "Computer analysis of moving polygonal images," *IEEE Trans. Comput.*, vol. C-24, pp. 966–976, October 1975.

[2] N. Ahuja and A. Rosenfeld, "Image models," in *Handbook of Statistics: Classification, Pattern Recognition and Reduction of Dimensionality*, P.R. Krishnaiah and L.N. Kanal Eds. Amsterdam: North-Holland Publishing Company, 1982, pp. 383–397.

[3] P. Anandan, "Computing dense displacement fields with confidence measures in scenes containing occlusion," in *Proc. SPIE (Intelligent Robots and Computer Vision)*, 1984, pp. 184–194.

[4] P. Anandan and R. Weiss, "Introducing a smoothness constraint in a matching approach for the computation of optical flow fields," in *Proc. Workshop Comp. Vision: Represenation and Control*, 1985, pp. 186–194.

[5] P. Anandan, "A unified perspective on computational techniques for the measurement of visual motion," in *Proc. IEEE Int. Conf. Computer Vision ICCV'87*, 1987, pp. 219–230.

[6] P. Anandan, "Measuring visual motion from image sequences," Ph.D. Thesis, Univ. of Massachusetts, 1987.

[7] S.T. Barnard and W.T. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-2, pp. 333–340, July 1980.

[8] H.C Bergmann, "Displacement estimation based on the correlation of image sequences," in *IEE Int. Conf. on Electronic Image Processing*, 1982, pp. 215–219.

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

[10] J.E. Besag, "Nearest-neighbour systems and the auto-logistic model for binary data," *J. Royal Stat. Soc.*, vol. B 34, pp. 75–83, 1972.

[11] J. Besag, "On the statistical analysis of dirty pictures," *J. R. Statist. Soc.*, vol. 48, B, pp. 259–279, 1986.

[12] A. Blake and A. Zisserman, *Visual Reconstruction*, Cambridge, Massachusetts: MIT Press, 1987.

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

[14] P.J. Burt, C. Yen and X. Xu, "Local correlation measures for motion analysis, a comparative study," in *Proc. IEEE Conf. on Pattern Recognition and Image Process.*, 1982, pp. 269–274.

[15] P.J. Burt, "Pyramid-based extraction of local image features with applications to motion and texture analysis," in *Proc. SPIE (Robots and Industrial Inspection)*, 1982, pp. 114–124.

[16] P.J. Burt, C. Yen and X. Xu, "Multi-resolution flow-through motion analysis," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 1983, pp. 246–252.

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

[18] N. Cornelius and T. Kanade, "Adapting optical-flow to measure object motion in reflectance and X-ray image sequences," in *Proc. ACM SIGGRAPH/SIGART Interdiscipl. Workshop on Motion: Representation and Perception*, 1983, pp. 145–153.

[19] G.R. Cross and A.K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-5, pp. 25–38, January 1983.

[20] J.L. Crowley and R.M. Stern, "Fast computation of the difference of low-pass transform," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-6, pp. 212–222, March 1984.

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

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

[23] C.L. Fennema and W.B. Thompson, "Velocity determination in scenes containing several moving objects," *Comput. Graphics Image Process.*, vol. 9, pp. 301–315, 1979.

[24] H. Gafni and Y.Y. Zeevi, "A model for separation of spatial and temporal information in the visual system," *Biological Cybernetics*, vol. 28, pp. 73–82, 1977.

[25] H. Gafni and Y.Y. Zeevi, "A model for processing of movement in the visual system," *Biological Cybernetics*, vol. 32, pp. 165–173, 1979.

[26] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-6, pp. 721–741, November 1984.

[27] D. Geman, "Stochastic model for boundary detection," *Image & Vision Computing*, vol. 5, pp. 61–65, May 1987.

[28] M. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," MIT Artificial Intelligence Laboratory A.I. Memo 975, 1987.

[29] G. Gidas, "A renormalization group approach to image processing problems," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-11, pp. 164–180, February 1989.

[30] F. Glazer, G. Reynolds and P. Anandan, "Scene matching by hierarchical correlation," in *Proc. Conf. Comp. Vision Patt. Rec. CVPR'83*, 1983, pp. 432–441.

[31] F.C. Glazer, "Hierarchical motion detection," Ph.D. Thesis, Univ. of Massachusetts Dept. of Comp. and Info. Science, 1987.

[32] W.E.L. Grimson, "A computational theory of visual surface interpolation," MIT Artificial Intelligence Laboratory A.I. Memo 613, 1981.

[33] W.E.L. Grimson, "An implementation of a computational theory of visual surface interpolation," *Comput. Vision, Graphics Image Process.*, vol. 22, pp. 39–69, 1983.

[34] W.E.L. Grimson, "Surface consistency constraints in vision," *Comput. Vision, Graphics Image Process.*, vol. 24, pp. 28–51, 1983.

[35] A. Habibi, "Two-dimensional Bayesian estimate of images," *Proc. IEEE*, vol. 60, pp. 878–883, July 1972.

[36] J.M. Hammersley and D.C. Handscomb, *Monte Carlo Methods*, London: Chapman and Hall, 1964.

[37] B.G. Haskell, "Frame-to-frame coding of television pictures using two-dimensional Fourier transforms," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 119–120, 1974.

[38] M. Hassner and J. Sklansky, "The use of Markov random fields as models for texture," in *Image Modeling*, A. Rosenfeld, Ed. Academic Press. Inc., 1981, pp. 185–198.

[39] E.C. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intelligence*, vol. 23, pp. 309–354, 1984.

[40] E.C. Hildreth, "Edge detection," MIT Artificial Intelligence Laboratory A.I. Memo 858, 1985.

[41] B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[42] B.K.P. Horn, *Robot Vision*, Cambridge, Massachusetts: The MIT Press, 1986.

[43] T.S. Huang and R.Y. Tsai, "Image sequence analysis: Motion estimation," in *Image Sequence Analysis*, T.S. Huang, Ed. Berlin, Germany: Springer-Verlag, 1981, pp. 1–18.

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

[45] K. Ikeuchi and B.K.P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, vol. 17, pp. 141–184, 1981.

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

[47] A.K. Jain, "A semicausal model for recursive filtering of two-dimensional images," *IEEE Trans. Comput.*, vol. C-26, pp. 343–350, April 1977.

[48] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1808, December 1981.

[49] S. Karlin and H.M. Taylor, *A First Course in Stochastic Processes*, New York: Academic Press, 1975.

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

[51] R. Kindermann and J.L. Snell, *Markov Random Fields and their Applications*, Providence, RI: Amer. Math. Soc., 1980.

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

[53] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Conf. Rec., Nat. Telecomm. Conf.*, 1981, pp. G5.3.1–G5.3.5.

[54] J. Konrad and E. Dubois, "Estimation of image motion fields: Bayesian formulation and stochastic solution," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Process. ICASSP'88*, 1988, pp. 1072–1074.

[55] J. Konrad and E. Dubois, "Multigrid Bayesian estimation of image motion fields using stochastic relaxation," in *Proc. IEEE Int. Conf. Computer Vision ICCV'88*, 1988, pp. 354–362.

[56] J. Konrad, "Motion-compensated interpolation for TV frame-rate conversion," INRS-Télécommunications Tech. Rep. 88–26, 1988.

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

[58] R. Kories and G. Zimmermann, "Motion detection in image sequences: an evaluation of feature detectors," in *Proc. IEEE Int. Conf. Pattern Recognition*, 1984, pp. 778–780.

[59] E.A. Krause, "Motion estimation for frame-rate conversion," Ph.D. Thesis, MIT Dept. of Electr. Eng. and Comp. Science, 1987.

[60] D. Lee and T. Pavlidis, "One-dimensional regularization with discontinuities," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-10, pp. 822–829, November 1988.

[61] J.O. Limb and J.A. Murphy, "Estimating the velocity of moving images in television signals," *Comput. Graphics Image Process.*, vol. 4, pp. 311–327, 1975.

[62] J.L. Marroquin, "Probabilistic solution of inverse problems," Ph.D. Thesis, MIT Dept. of Electr. Eng. and Comp. Science, 1985.

[63] J. Marroquin, S. Mitter and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Am. Stat. Soc.*, vol. 82, pp. 76–89, March 1987.

[64] D.S. Martinez, "Model-based motion estimation and its application to restoration and interpolation of motion pictures," Ph.D. Thesis, MIT Dept. of Electr. Eng. and Comp. Science, 1986.

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

[66] A. Mitiche, Y.F. Wang and J.K. Aggarwal, "Experiments in computing optical flow with the gradient-based, multiconstraint method," *Pattern Recognition*, vol. 20, pp. 173–179, 1987.

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

[68] H.-H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Comput. Vision, Graphics Image Process.*, vol. 21, pp. 85–117, 1983.

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

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

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

[72] Y. Ninomiya and Y. Ohtsuka, "A motion-compensated interframe coding scheme for television pictures," *IEEE Trans. Commun.*, vol. COM-30, pp. 201–211, January 1982.

[73] R. Paquin and E. Dubois, "A spatio-temporal gradient method for estimating the displacement field in time-varying imagery," *Comput. Vision, Graphics Image Process.*, vol. 21, pp. 205–221, 1983.

[74] T. Poggio and V. Torre, "Ill-posed problems and regularization analysis in early vision," MIT Artificial Intelligence Laboratory A.I. Memo 773, 1984.

[75] T. Poggio, H. Voorhees and A. Yuille, "A regularized solution to edge detection," MIT Artificial Intelligence Laboratory A.I. Memo 833, 1985.

[76] J.L. Potter, "Velocity as a cue to segmentation," *IEEE Trans. Syst., Man and Cybern.*, vol. SMC-5, pp. 390–394, May 1975.

[77] J.L. Potter, "Scene segmentation using motion information," *Comput. Graphics Image Process.*, vol. 6, pp. 558–581, 1977.

[78] A. Rosenfeld and G.J. Vanderburg, "Coarse-fine template matching," *IEEE Trans. Syst., Man and Cybern.*, vol. SMC-7, pp. 104–107, February 1977.

[79] S.M. Ross, *Applied Probability Models with Optimization Applications*, San Francisco: Holden-Day, 1970.

[80] B.G. Schunck, "Motion segmentation and estimation," PhD Thesis, MIT, Dept. of Electr. Eng. and Comput. Sci., 1983.

[81] T. Simchony, R. Chellappa and Z. Lichtenstein, "Graduated nonconvexity algorithm for image estimation using compound Gauss Markov field models," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Process. ICASSP'89*, 1989, pp. 1417–1420.

[82] R. Srinivasan and K.R. Rao, "Predictive coding based on efficient motion estimation," in *Conf. Rec., Int. Conf. Commun.*, 1984, pp. 521–526.

[83] D. Terzopoulos, "Multi-level reconstruction of visual surfaces," MIT Artificial Intelligence Laboratory A.I. Memo 671, 1982.

[84] D. Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-8, pp. 129–139, March 1986.

[85] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-8, pp. 413–424, July 1986.

[86] W.B. Thompson and T.-C. Pong, "Detecting moving objects," in *Proc. IEEE Int. Conf. Computer Vision ICCV'87*, 1987, pp. 201–208.

[87] V. Torre and T.A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-8, pp. 147–163, March 1986.

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

[89] P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, Dordrecht: D. Reidel Publishing Company, 1987.

[90] K. Wohn, L. S. Davis and P. Thrift, "Motion estimation based on multiple local constraints and nonlinear smoothing," *Pattern Recognition*, vol. 16, pp. 563–570, 1983.

[91] R.Y. Wong and E.L. Hall, "Sequential hierarchical scene matching," *IEEE Trans. Comput.*, vol. C-27, pp. 359–366, April 1978.

[92] J.W. Woods, "Two-dimensional discrete Markovian fields," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 232–240, March 1972.