Estimation of Accelerated Motion and Occlusions from Time-Varying Images

by

Michel J. Chahine

B. Eng.

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

Department of Electrical Engineering

McGill University

Montréal, Canada

July 1994

Abstract

This thesis addresses the problem of modeling and computing dense 2-D velocity and acceleration fields from time-varying images and applying them to motion-compensated interpolation. Unlike in many other approaches that assume motion to be locally translational, the approach proposed here uses a quadratic motion trajectory model that incorporates both velocity and acceleration. This model corresponds better to natural image sequences especially when processing over multiple frames is considered. One of the advantages of using accelerated motion over linear trajectories is in motion-compensated processing over multiple images. This is due to the fact that over longer time frame, a quadratic motion model is capable of providing a better intensity match along trajectories than the linear model. The side effect is, however, that with more images used for estimation occlusion effects play a more dominant role. Therefore, another motion model is proposed to account for occlusions and motion discontinuities. The algorithm for the estimation of velocity, acceleration, occlusion and discontinuity fields is formulated using Gibbs-Markov models that are linked together by the *Maximum A Posteriori* (MAP) probability criterion. This is equivalent to *regularization* where the stabilizing functional is related to *a priori* motion models. The resulting multiple term cost function is optimized using deterministic relaxation implemented over a pyramid of resolutions. Numerous experimental results are presented for interlaced and progressive test images. Mean-squared error is calculated for motion estimates obtained from images with (known) synthetic motion. For sequences with natural motion temporal interpolation compensated for motion is implemented and the estimates are evaluated with respect to the image reconstruction error. It is concluded that for images containing acceleration, such as "talking heads", the quadratic motion model permits a substantial reduction of the reconstruction error when compared with the ubiquitous linear model. A further improvement, especially around motion boundaries, is observed when motion as well as occlusions are estimated. The improvements are particularly striking around the mouth and eyes of a "talking head".

Sommaire

Le présent mémoire traite le problème de modélisation et de calcul de champs denses 2-D de vélocités at d'accélérations à partir de séquences d'images dynamiques et leur applications dans un contexte de codage interpolatif avec compensation du mouvement. L'approche proposée utilise un modèle quadratique de trajectoire de mouvement, incorporant des vélocités et des accélérations, en contraste à plusieurs autres approches qui supposent un mouvement de translation. Le modèle quadratique correspond mieux aux séquences d'images naturelles surtout quand le traitement sur plusieurs trames est considéré. Un des atouts de ce modèle en comparaison avec le modèle linéaire apparait lors d'un traitement compensé par le mouvement sur plusieurs images. Ceci est dû au fait que sur une plus longue période de temps, un modèle de trajectoire quadratique est capable d'offrir un meilleur appariement d'intensités le long des trajectoires que le modèle linéaire. Par contre, les effets d'occlusions jouent un rôle dominant quand l'estimation est étendue sur plusieurs images. Un autre modèle de mouvement qui tient compte des occlusions et des discontinuités en mouvement est en conséquence proposé. L'algorithme d'estimation de champs de vélocités, d'accélérations, d'occlusions, et de discontinuités est formulé à partir de modèles de Gibbs-Markov reliés par le critère de probabilité *A postériori Maximale* (APM). Ceci est équivalent à la méthode de *régularisation* où le fonctionnel stabilizateur est relié aux modèles *à priori* de mouvement. La fonction de coût résultante, à termes multiples, est optimisée par relaxation déterministe avec un traitement hierarchique. Plusieurs résultats expérimentaux sont présentés pour des tests d'images à structures d'échantillonage progressives et entrelacées. L'erreur quadratique moyenne est calculée sur les paramètres de mouvement estimés à partir de séquences d'images ayant un mouvement synthétique (connu). Une interpolation temporelle avec compensation du mouvement est par ailleurs implémentée pour les séquences d'images naturelles. L'estimation de mouvement sur ces dernières est évaluée par rapport à l'image d'erreur reconstruite. Il sera conclu que pour les images contenant des accélérations, comme dans les "têtes parlantes", le modèle de mouvement quadratique, en comparaison avec le modèle linéaire omniprésent, permet une réduction considérable de l'erreur reconstruite. Une amélioration plus importante, surtout autour des contours en mouvement, est observée quand le mouvement aussi bien que les occlusions sont estimés. Ces améliorations sont particulièrement saillantes autour de la bouche et des yeux d'une "tête parlante".

Acknowledgements

I would like to thank my supervisor Professor Janusz Konrad for his patient guidance throughout my graduate studies. His multiple suggestions and advices during the course of my research are infinitely appreciated.

I am also grateful to the Institut National de la Recherche Scientifique, (INRS)-Télécommunications, for providing me excellent computer facilities to conduct my research, and to the National Science and Engineering Research Council (NSERC) for its financial assistance during my graduate studies.

Finally, this thesis could not have been completed without the constant love and support of my parents Joseph and Renée, my sister Joëlle, and my brother Gaby.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Video compression

The amount of data associated with visual information is very large. Typical television images, for instance, generate data rates exceeding 10 million pixels per second. Storage and/or transmission of such data require large capacity and/or bandwidth, which could be very expensive. Video compression is concerned with the reduction of number of bits required to store or transmit images under the constraint of achieving some target quality. The 2:1 line interlacing in conventional television was one of the first techniques used to provide a simple means of 2:1 bandwidth reduction without objectionable flicker or image breakup. The success of this method rests on the fact that the Human Visual System (HVS) acts as a low-pass filter and has a poor response to simultaneous high spatial and temporal frequencies.

Video compression methods fall into two common categories [23]. The first category is concerned with statistical redundancy removal such as *Huffman, run-length*, and *arithmetic coding*. Huffman coding is one of the most efficient techniques in this category that increases the average compression by assigning shorter code words to frequently encountered blocks (or symbols) and longer ones for rarely encountered blocks. This technique is not very practical for television images whose long-term histogram is approximately uniform, but it is quite useful for coding of binary data such as graphics and facsimile images, and also in *predictive and transform coding* algorithms. These algorithms fall in the second category that is concerned with perceptual irrelevancy removal. They try to exploit the low-pass response of the HVS by removing mutual redundancies in the video signal while preserving a good subjective quality. One of the most common techniques used in predictive coding is the *differential pulse code modulation* (DPCM) or *differential* PCM. Transform coding, such as the Discrete Cosine Transform (DCT), is an alternative to predictive coding. In this method, a block of data is unitarily transformed so that a large fraction of its total energy is packed in relatively few transform coefficients, which are quantized independently. The optimal transform coder is defined as the one that minimizes the mean square distortion of the reproduced data for a given number of total bits.

On the other hand, moving image compression can exploit temporal redundancy due to the high correlation of intensity along motion trajectories. For instance, motion-compensated prediction is a powerful tool, provided that motion is known, helpful in removing interimage redundancy. This technique has given rise to the currently most sophisticated realizable coder, known as the *motion-compensated hybrid* (DPCM/DCT) *coder* (Figure 1.1), used in MPEG-1, MPEG-2, and H.261

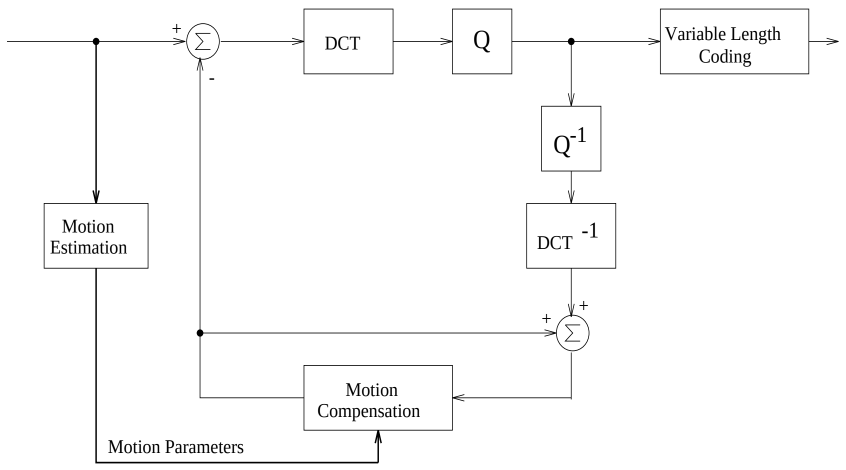


Figure 1.1: Motion-compensated hybrid (DPCM/DCT) coder.

standards. This encoder consists typically of a temporal DPCM, with a prediction based on motion parameters estimated at the encoder and transmitted to the decoder as side information. The difference between the predicted and actual fields is then compressed like a still image using DCT transform.

In motion-compensated predictive coding [8][33], the main goal of motion compensation is to minimize the variance of the prediction error; it is not necessary to obtain the true motion field since errors in motion estimates simply increase the magnitude of the prediction error. This error is quantized and transmitted at the cost of a few extra bits with little impact on picture quality. Thus highly accurate motion estimation is not crucial with DPCM coding schemes currently used. The block-oriented motion field estimates are probably adequate for this application.

1.2 Motion-compensated processing

Motion-compensated processing is another important area which exploits the high correlation along motion trajectories. *Sampling structure conversion* that relies on spatio-temporal interpolation, and *interpolative coding* are two frequently encountered applications that fall into this category.

There are two main situations where the use of motion-compensated interpolation in sampling structure conversion is advantageous over the use of fixed spatial interpolation filters. The first is when the current image field is spatially aliased, usually because a nonorthogonal spatio-temporal sampling structure (such as in interlaced sampling) has been used, so that spatial interpolation alone does not perform adequately. A use of temporal interpolation perhaps jointly with spatial interpolation as proposed in [34] can considerably increase the quality of interlaced images. An example of this application is the conversion from interlaced scanning to progressive scanning known as *deinterlacing.* The second situation arises when no input data is available at a certain time instant for which interpolation is being carried out and hence purely temporal interpolation must be used. In other words, an image field is being generated at a time for which no input image fields exists. This is the case for applications such as field rate conversion (e.g., between 50 Hz and 60 Hz), upconversion from temporally subsampled signals, and field rate increase to reduce display artifacts or for slow motion effects. Sampling structure conversion is also used in *standards conversion* (such as conversion from NTSC to PAL and vice versa), and in *spatio-temporal pyramidal* coding [37][38][39] for multiscale representation of video signals (i.e., HDTV, videophone, and video-conference). It should be noted that there is no possibility of recovering from errors during sampling structure conversion, thus it is imperative that motion estimates be of high quality, and that occlusions be properly handled. The occlusion effect is a common problem in motion estimation algorithms. It manifests itself by the fact that moving objects in a sequence of images generate occluded regions (i.e., covered or exposed) in which the estimation of motion becomes much more complex or even maybe impossible.

Motion-compensated processing can also be used in the context of interpolative coding. In this application, images are omitted in the transmitter and then reconstructed in the receiver by motion-compensated interpolation. These images correspond to the B-frames (Figure 1.2) in MPEG standards where I-frames are intra-coded frames, P-frames are predictive-coded frames, and B-frames are bidirectionally interpolated frames using motion compensation. The motion estimates and/or the

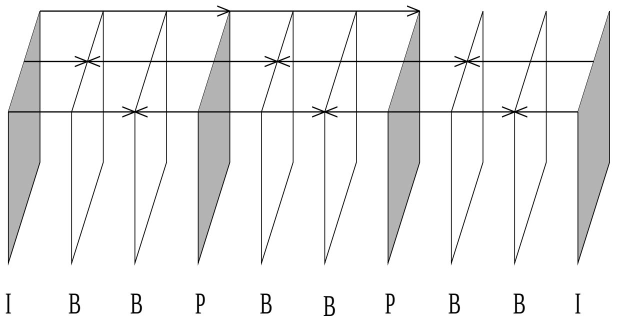


Figure 1.2: Typical MPEG motion compensation structure.

motion-compensated interpolation error (residual) of the omitted fields are encoded and transmitted. If motion estimates are not precise, then correction by the transmitted residual is possible. However, in case of no transmission of residual, motion estimates must be precise.

1.3 Motion estimation

As discussed, motion is widely used in motion-compensated processing and video compression. On the other hand, estimation of motion from dynamic images is a very difficult task due its *ill-posedness* [2]. Despite this difficulty, however, many approaches to the problem have been proposed in the last dozen years [20][18][27][7]. Most of these approaches use only 2 fields to estimate motion. In this thesis, a new motion estimation algorithm that uses multiple fields to estimate motion is addressed. For this task a quadratic model, incorporating both velocity and acceleration, is used to model motion trajectories. The side effect is, however, that with more images used for estimation occlusion effects play a dominant role. Therefore, another feature which helps in canceling this side effect, has also been added to the motion estimation algorithm. This feature consists of simultaneously detecting occlusion areas [9] and estimating motion in order to maintain high quality of motion estimates near motion discontinuities.

1.4 Organization of the thesis

The ill-posed nature of the motion estimation problem and its solution via *regularization theory* are discussed in Chapter 2. An overview of motion estimation techniques is then presented. Various approaches used to improve motion estimates, such as hierarchical processing, modeling of motion discontinuities and occlusions, and multiframe processing, are finally described. Chapter 3 is concerned with the description of the new motion estimation algorithm. The derivation of the objective function is illustrated in detail using Gibbs-Markov models linked together by the *Maximum*

A Posteriori (MAP) probability criterion. The optimization of this objective function using deterministic relaxation implemented over a pyramid of resolutions is then discussed. Experimental results for image sequences with synthetic and natural motion are also presented. In Chapter 4, the proposed motion estimation algorithm is extended to account for occlusions. Experimental results illustrating the advantages of occlusion processing in generating piecewise-continuous motion fields rather than globally-continuous are presented at the end. The conclusions and summary of the main contributions of this thesis are discussed in Chapter 5.

Chapter 2

Overview of Motion Estimation Techniques

In this chapter definition of the motion estimation problem is given and some existing methods that solve it are described. The image acquisition process along with the definition of the displacement field and some applications of motion estimation are presented in Section 2.1. Section 2.2 is concerned with the ill-posed nature of the problem and with its solution via *regularization theory*. Various approaches to motion estimation are discussed and compared in Section 2.3. Techniques used to improve motion estimates such as hierarchical processing, modeling of discontinuities and occlusion areas and multiframe processing are reviewed in Section 2.4.

2.1 Introduction

2.1.1 Apparent motion

The relative motion between objects in a scene and a camera gives rise to the apparent motion of objects in a sequence of images. This motion can be characterized by observing the apparent motion of a discrete set of features or brightness patterns in the images. Two distinct categories have been developed for the computation of motion from image sequences.

1. The first category, known as *feature matching*, requires an extensive image analysis to extract a set of relatively sparse, but highly discriminatory, 2-D features in the images (i.e., points, corners, lines). Such features are extracted from each image, and then are identified in subsequent images leading ultimately to the computation of motion parameters of different objects in the image. This category is very suitable for establishing the long range correspondence in a sequence of images.

2. The second category, characterized by pixel-based processing, consists of using pixel intensities to compute 2-D field of instantaneous velocities of pixels in the image plane. This relatively dense field is known as the *optical flow* and is usually defined for every pixel in the image. This category, on the other hand, is suitable for establishing the short range correspondence in a sequence of images.

Optical flow, or 2-D velocity field, represents "the distribution of apparent velocities of movement of brightness patterns in an image" [20]. For images sampled in the temporal direction, the concept of the velocity field is replaced by that of the displacement field which will be shortly defined. The motion field which can denote either a velocity field or a displacement field, can be used in conjunction with added constraints or information regarding the scene to compute the actual 3-D relative velocities between scene objects and camera [1]. Also, discontinuities in the motion field can help in segmenting images into regions that correspond to different objects.

2.1.2 Applications

Apparent motion estimated from a sequence of images (often called video) is used in a wide range of applications such as:

1. transmission and processing of video: motion-compensated interpolation for sampling structure conversion, motion-compensated filtering for noise reduction, motion-compensated coding for bit rate reduction,

2. biomedical applications: analysis of medical imagery for generation of diagnostics,

3. meteorology: interpretation of satellite images for prediction of atmospheric processes,

4. computer vision (robotics): structure from motion for passive navigation, 3-D motion from 2-D motion for passive navigation and object tracking.

All these applications reveal the importance of 2-D motion information, and the need of a good estimation algorithm capable of calculating motion that is close to the true underlying motion.

2.1.3 Observation process

The data from which motion is estimated is usually obtained by an image acquisition system. Thus, the observed image g is related to the true underlying image u by the observation process which can be modeled at varying degrees of sophistication. In general, this process has three main elements: a nonlinear shift-variant spatio-temporal filtering, a random perturbation, and a spatio-temporal sampling operation.

The basic model typically used is the filtering of u by a linear shift-invariant camera aperture (impulse response h) and addition of noise n

$$g_c(\mathbf{x}, t) = h(\mathbf{x}, t) * u(\mathbf{x}, t) + n(\mathbf{x}, t), \qquad \mathbf{x} \in R^2,\ t \in R \tag{2.1}$$

followed by sampling on lattice $(\Lambda_g)_t$ [10] every T seconds

$$g(\mathbf{x}, t) = g_c(\mathbf{x}, t), \qquad \mathbf{x} \in (\Lambda_g)_t,\ t = kT. \tag{2.2}$$

In this thesis, progressive and interlaced lattices are studied. Without loss of generality, the case of no filtering, i.e., $h(\mathbf{x}, t) = \delta(\mathbf{x}, t)$ with δ being the Dirac delta function, is used to simplify subsequent developments.

The degree of sophistication used in modeling the observation process undoubtedly has an impact on the estimated motion fields. It is not clear, however, whether it is more advantageous to increase the complexity of the observation process or of the models used in the estimation algorithm, especially in the view of the usual unavailability of imaging system parameters.

2.1.4 Definition of the displacement field

Most of the estimation methods proposed in the literature rely on spatio-temporal variations of the observed intensity g to estimate 2-D motion. In the context of digital coding adapted to motion information, the goal is to find such a motion field that minimizes the amount of information to be transmitted. Displacement field at time t establishes a correspondence between points from the image at time t and points from images at time t_- and t_+. The displacement field $\mathbf{d}(t)$ at time t of a sequence of images consisting of a sphere moving downward on a still background is shown in Figure 2.1. Only the displacement vectors of point $\mathbf{x}$ that belongs to the moving

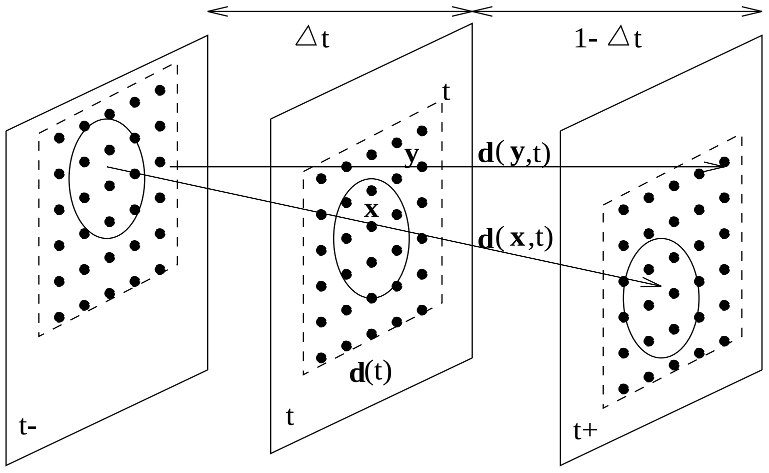


Figure 2.1: Illustration of the displacement field $\mathbf{d}_t$ at time t estimated from two image fields at t_- and t_+ (only 2 displacement vectors $\mathbf{d}(\mathbf{x},t)$ and $\mathbf{d}(\mathbf{y},t)$ are shown).

sphere, and the point $\mathbf{y}$ that belongs to the stationary background are shown. The notation $\mathbf{d}_i$ will be used to denote the displacement vector $\mathbf{d}(\mathbf{x}_i,t)$ at position $(\mathbf{x}_i,t)$.

Estimation of displacement fields usually relies on the following assumptions:

1. Image intensity remains constant along the motion trajectory.
2. Displacement fields are spatially smooth.
3. Displacement field consists of a dense set of motion vectors.
4. Effects of occlusions are negligible.
5. Motion is locally translational, i.e., linear motion trajectory model is used.

These assumptions often do not reflect real case situations; the intensity along motion trajectories can vary due to a change of illumination in the scene and hence assumption 1 is violated. The true displacement vectors at the boundaries of two objects underlying different motion trajectories are usually not smooth, and hence assumption 2 is violated at motion discontinuities. The assumption of locally translational motion (assumption 5) is violated for motion trajectories with longer temporal support. All these assumptions can however be modified in such a way to improve the motion estimates, as will be discussed in Section 2.4 which will investigate the modeling of discontinuities and occlusion areas and the use of a non-linear motion trajectory model.

2.2 Ill-posed nature of motion estimation

2.2.1 Statement of the problem

Problems encountered in *early vision* [2], such as the recovery of 3-D motion and optical flow, shape from shading, surface interpolation, and edge detection, are common in nature. They can be regarded as inverse problems that try, for instance, to recover physical properties of 3-D surfaces from their projection onto an image plane. It is clear that the data (observations) used (i.e., 2-D images) contain in general limited information about the solutions (i.e., the 3-D properties). This lack of information implies that problems of early vision are very often *ill-posed.* In the original sense of Hadamard [2], a *well-posed* problem is characterized by the following three properties:

1. Existence: There is a solution.
2. Uniqueness: The solution is unique.
3. Continuity: The solution depends in a continuous manner on the data.

Hence in an ill-posed problem, the solution may not exist, may not be unique (giving an ambiguous reconstruction), or it may not depend continuously on the data. From this definition, it is clear that motion estimation based on assumptions from the previous section is ill-posed as it may violate the above properties:

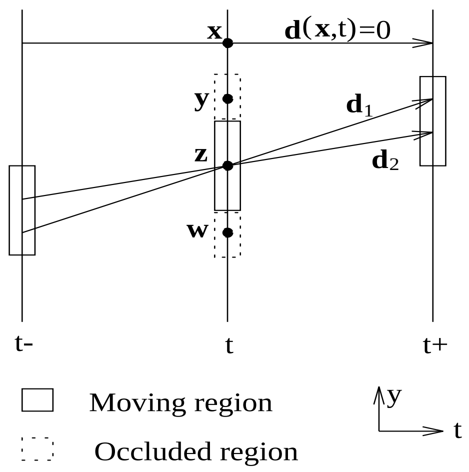


Figure 2.2: Illustration of the ill-posed nature of the motion estimation problem ($\mathbf{d_1}$ and $\mathbf{d_2}$ are two possible displacement vectors at position $(\mathbf{z}, t)$).

1. For occluded pixels there is no solution as the intensities of those pixels (i.e., the data) are not available in the next or previous frames (violation of existence). Figure 2.2 shows an example of occluded regions generated at time t when the object is moving upward from time t_- to time t_+. Hence, for pixels $(\mathbf{w}, t)$ and $(\mathbf{y}, t)$, both belonging to occluded regions at time t, displacement vector

does not exist. This is due to the fact that pixel $(\mathbf{y}, t)$ becomes covered at time t_+, whereas pixel $(\mathbf{w}, t)$ becomes exposed only at time t, and hence no correspondence can be established for these pixels along the three considered time instants.

2. There are many possible motion trajectories that satisfy the data even for some predefined motion trajectory model (violation of uniqueness). For instance, if the moving object in Figure 2.2 has constant intensity, $\mathbf{d}_1$ and $\mathbf{d}_2$ may be two possible displacement vectors at position $(\mathbf{z}, t)$ when linear motion trajectory is considered.

3. For a small local modification of intensities, there may be significant change in the estimated motion vector length and/or orientation (violation of continuity).

The need to analyze ill-posed problems such as motion estimation has given rise to *regularization theory* discussed in the next section.

2.2.2 The regularization theory

Most linear inverse problems can be formulated as follows. Suppose that functional spaces X and Y are given along with a continuous operator $\mathcal{L}$ from X into Y. The problem is then to find, by some regularization theory, a function $u \in X$ for some observation $g \in Y$ such that $g = \mathcal{L}u$. The approach proposed by Tikhonov, which is addressed in [2], attempts to solve this problem while minimizing a certain cost function. The minimization problem can be formulated as follows [2]:

$$\min_u \left\{ \|\mathcal{L}u - g\|_Y^2 + \lambda \|\mathcal{C}u\|_Z^2 \right\}, \tag{2.3}$$

with $\|\cdot\|_Y$ denoting the norm in Y, and C being a linear operator from X into the constraint space Z that expresses a certain *a priori* property of the solution such as spatial smoothness in the case of motion estimation. λ is the regularization parameter that plays a crucial role in weighting the compromise between the two terms of the cost function.

In spite of existing theory for the optimal choice of λ, such choice is probably the most difficult problem in the regularization theory as will be shown later. It is worth to note that regularization theory can provide optimal techniques to reduce the effect of noise but cannot produce new information if it is not originally available. Most of the motion estimation methods that will be discussed next can actually be viewed as a direct consequence of the regularization theory via the solution proposed by Tikhonov in (2.3).

2.3 Methods of motion field estimation

2.3.1 Transform-domain methods

The *Fourier-phase* approach [15] is the most common one used. It uses the shift property of the Fourier transform which states that a spatial shift in a signal corresponds to a shift in phase in the Fourier transform of that signal. Hence, if $G(w_x, w_y)$ denotes the Fourier transform of the image $g(x, y)$, then, if this image undergoes a uniform translation $\mathbf{d} = [d_x \;\; d_y]^T$,

$$g(x - d_x, y - d_y) \Longleftrightarrow G(w_x, w_y) \cdot e^{-j2\pi(w_x d_x + w_y d_y)}. \tag{2.4}$$

Measuring the difference in phase between the 2-D Fourier transforms of the images at t_- and t_+, one can deduct a displacement vector corresponding to a sufficiently large bloc of the image. However, the position of the obtained displacement vector is not known, and therefore has to be localized in some way.

Another approach is the *spatio-temporal frequency* method [22] that consists of calculating the orientation of the 3-D Fourier spectrum of a time-varying image $g(x, y, t)$ undergoing a translational motion with some constant velocity $\mathbf{v} = [v_x \;\; v_y]^T$. In this case, $g(x, y, t)$ can be expressed as follows:

$$g(x,y,t) = g(x - v_x t, y - v_y t, 0) = g(x,y,0) * \delta(x - v_x t, y - v_y t, 0). \tag{2.5}$$

The Fourier transform of $g(x, y, t)$ in (2.5) is then derived as follows:

$$G(w_x, w_y, w_t) = G(w_x, w_y) \cdot \delta(v_x w_x + v_y w_y + w_t). \qquad (2.6)$$

Hence, the spectrum of such an image has all its energy concentrated in a plane defined by: $v_x w_x + v_y w_y + w_t = 0$. The orientation of such a plane is uniquely defined by the components of the velocity vector $\mathbf{v}$.

These methods allow to determine a velocity vector for the whole analyzed block. They are used to determine the velocity of a single object moving on a uniform background. On the other hand, they cannot be used just by themselves on real television sequences where the motion is much more complex but can be followed by some other methods.

2.3.2 Matching algorithms

Matching algorithms associate structures in a reference image with corresponding structures in subsequent images. The best match is detected following a search that yields the optimal displacement vector for each structure. These algorithms are divided into two categories.

Feature matching

This is the only method structured to resolve the long-range correspondence problems. A number of approaches to this method is presented in [1]. Usually, features (i.e., points, lines, corners) are first identified in the images used, and then correspondence between those features is established. The task of establishing and maintaining such correspondence is, however, nontrivial. The ambiguity is also increased by occlusion effects which cause features to appear or disappear and also give rise to "false" features.

Block matching

A simpler method that does not require a search for features is block matching. This approach is based on the assumption that all pixels inside a block have the same motion (only a single vector is estimated for each block). The problem is then to estimate motion of an $M \times N$ block of pixels of the image g_t at time t with respect to the previous image at time t_-. For this purpose, the block in g_t is compared with

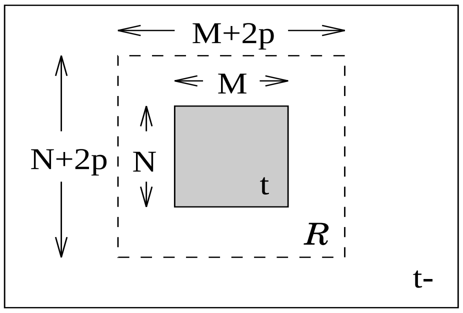


Figure 2.3: Matching of an $M \times N$ block of pixels at time t within a $(M+2p) \times (N+2p)$ search area R at t_-.

another block inside a search area R of dimensions $(M+2p) \times (N+2p)$ in g_{t-} (Figure 2.3) where p is the maximal allowed displacement. The mean distortion function between these two blocks is defined as:

$$D(i,j) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \phi\left(g_t(m,n) - g_{t-}(m+i,n+j)\right), \quad -p \leq i,j \leq p \tag{2.7}$$

where $g_t(m,n)$ denotes the brightness value at pixel position (m,n) and time t, and $\phi(x)$ is a positive ascending distortion function (i.e., the resulting criterion is MSE for $\phi(x) = x^2$, absolute minimal error for $\phi(x) = |x|$). The direction of minimal distortion is then given by (i^*, j^*) such that:

$$(i^*, j^*) = \min_{(i,j)} D(i,j), \quad -p \leq i,j \leq p \tag{2.8}$$

To speed up the search procedure many methods have been proposed such as the *2D-logarithmic search* [24], the *three-step search* [26], and the *conjugate direction search* [36].

Block matching is usually suitable for the short-range correspondence problems, while it is inappropriate for the long-range ones. This can be explained by the fact that for a larger search area, the search procedure becomes too costly, and equation (2.7) tends to be non-convex, leading to convergence to a local minimum. This method, however, works well with image sequences sampled in time at 60 Hz that do not contain violent motion.

2.3.3 Spatio-temporal gradient methods

Block matching is a simple technique to implement and relatively fast. It generates a single displacement vector for each block of pixels. But this method becomes very slow and costly when dense displacement fields are required (i.e., the bloc is reduced to a single pixel). For higher spatial resolution of the optical flow, spatio-temporal gradient methods are recommended. Three approaches to these methods are hereby introduced with an emphasis on the Horn-Schunck and Bayesian approaches.

Minimization of the DFD

This approach introduced by Netravali-Robbins [33] is based on the iterative minimization of the square of the DFD (Displaced Frame Difference), a measure of the motion-compensated prediction error. Displacement field $\mathbf{d}_t$ at time t is estimated on a pixel-by-pixel basis using the images at times t_- and t_+ with $\Delta_t = 1$ (Figure 2.1). The DFD is defined as follows:

$$DFD(\mathbf{x}, \mathbf{d}) = g(\mathbf{x}, t) - g(\mathbf{x} - \mathbf{d}, t_-), \tag{2.9}$$

where $\mathbf{d}$ is displacement vector for pixel $\mathbf{x}$ at time t, and $g(\mathbf{x}, t)$ represents the observed intensity of pixel $\mathbf{x}$ at time t. A displacement field estimate $\hat{\mathbf{d}}_t$ can be derived as follows:

$$\hat{\mathbf{d}} = \min_{\mathbf{d}} DFD^2(\mathbf{x}, \mathbf{d}), \qquad \forall \mathbf{x} \in (\Lambda_g)_t. \tag{2.10}$$

The minimization is carried out iteratively using the steepest descent algorithm. The resulting iterative update equation is expressed as:

$$\begin{aligned} \hat{\mathbf{d}}^i &= \hat{\mathbf{d}}^{i-1} - \epsilon DFD(\mathbf{x}, \hat{\mathbf{d}}^{i-1}) \nabla_{\mathbf{d}} DFD(\mathbf{x}, \hat{\mathbf{d}}^{i-1}) \\ &= \hat{\mathbf{d}}^{i-1} - \epsilon DFD(\mathbf{x}, \hat{\mathbf{d}}^{i-1}) \nabla_{\mathbf{x}} g(\mathbf{x} - \lfloor \hat{\mathbf{d}}^{i-1} \rfloor, t_-), \end{aligned} \tag{2.11}$$

where $\nabla_{\mathbf{d}}$ is the gradient with respect to $\mathbf{d}$, $\nabla_{\mathbf{x}} = [\frac{\partial}{\partial x} \;\; \frac{\partial}{\partial y}]^T$ is the spatial gradient, $\hat{\mathbf{d}}^i$ is the estimate of $\mathbf{d}$ at iteration i, and $\lfloor \hat{\mathbf{d}}^{i-1} \rfloor$ denotes the closest lower integer value of $\hat{\mathbf{d}}^{i-1}$.

The resulting estimated displacement field is more representative of the real motion when compared with the one obtained by block matching. This is due to the fact that the above algorithm overcomes, to a large extent, the problems of multiple moving objects. It also permits different parts of an object to undergo different displacements, provided the recursive algorithm in (2.11) has sufficiently rapid convergence.

The Horn-Schunck approach

Horn and Shunck [20][19] proposed to estimate the 2-D velocity field $\mathbf{v}$ at time t using the *motion constraint equation*, where $\mathbf{v}(\mathbf{x}, t) = [v_x \;\; v_y]^T$ denotes the velocity vector at position $(\mathbf{x}, t)$. If dx and dy denote the corresponding horizontal and vertical displacements of pixel (x, y, t) after a time increment dt, then the assumption of constant image intensity along motion trajectories can be expressed as follows:

$$g(x + dx, y + dy, t + dt) = g(x, y, t) \tag{2.12}$$

If the intensity varies smoothly with x, y, and t, the expansion of the left-hand side of (2.12) by the Taylor series results in the following:

$$g(x, y, t) + dx\frac{\partial g}{\partial x} + dy\frac{\partial g}{\partial y} + dt\frac{\partial g}{\partial t} + e = g(x, y, t), \tag{2.13}$$

where e contains second and higher order terms in dx, dy, and dt. Canceling $g(x, y, t)$ in (2.13), dividing through by dt, and taking the limit as $dt \rightarrow 0$, the motion constraint equation $c(\mathbf{v})$ at $(\mathbf{x}, t)$ can be obtained:

$$c(\mathbf{v}) = \mathbf{v}(\mathbf{x}, t) \cdot \nabla_{\mathbf{x}}^T g + \frac{\partial g}{\partial t} = 0. \tag{2.14}$$

Using the motion constraint equation, one can only derive the component of the velocity vector in the direction of the brightness gradient (but not the component along the isobrightness contour). This ambiguity is known as the *aperture problem.*

In order to solve this ambiguity, regularization approach proposed by Tikhonov (equation (2.3)) is used. Hence, an additional motion smoothness error $s(\mathbf{v})$ which ensures the smoothness of the calculated velocity field is added to the error based on the motion constraint equation. $s(\mathbf{v})$ is expressed as the sum of squared magnitudes of gradients of the velocity components:

$$s(\mathbf{v}) = \|\nabla_{\mathbf{x}} v_x\|^2 + \|\nabla_{\mathbf{x}} v_y\|^2 = \left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2. \tag{2.15}$$

The estimated velocity field $\hat{\mathbf{v}}$ can then be derived by solving the following continuous minimization problem:

$$\min_{\mathbf{v}} \int \left(c^2(\mathbf{v}) + \lambda \cdot s(\mathbf{v})\right) d\mathbf{x}, \tag{2.16}$$

where λ is the regularization parameter that weighs the error $c(\mathbf{v})$, relative to the smoothness error $s(\mathbf{v})$. This parameter is ideally small if the assumption expressed in (2.12) is accurate, and large otherwise. The choice of a fixed value for this parameter remains to be a very crucial problem in such motion estimation algorithms.

After discretization of g and $\mathbf{v}$, an estimate $\hat{\mathbf{v}}_{(i,j)}$, where (i, j) denotes the discretized spatial position $\mathbf{x}$, can be calculated directly by solving a linear system of the form: $A\mathbf{v}_{(i,j)} = b$. This linear system is obtained by using the necessary condition for optimality of the objective function in (2.16) [20]. If N_v is the number of velocity sites at time t, then standard methods such as Gauss-Jordan elimination, used in an attempt to solve simultaneously the $2N_v$ linear equations (2 for each position), are very costly. The reason for this is that the corresponding matrix of order $2N_v$ is very large and sparse. Therefore, Horn and Schunck proposed to use deterministic relaxation (Jacobi, Gauss-Seidel) to solve iteratively this linear system. The relaxation algorithm resulted in the following iterative update equation:

$$\hat{\mathbf{v}}^{n+1}_{(i,j)} = \bar{\mathbf{v}}^{n}_{(i,j)} - \frac{\bar{\mathbf{v}}^{n}_{(i,j)} \cdot \nabla^T_{(i,j)} g + g^t_{(i,j)}}{\lambda + \|\nabla_{(i,j)} g\|^2} \nabla_{(i,j)} g, \tag{2.17}$$

where $\nabla_{(i,j)} g = [g^x_{(i,j)} \;\; g^y_{(i,j)}]^T$, and $g^x_{(i,j)}$, $g^y_{(i,j)}$, $g^t_{(i,j)}$ are finite-difference approximations of the horizontal, vertical, and temporal derivatives of g respectively. Note that the new value of $\hat{\mathbf{v}}^{n+1}_{(i,j)}$ at (i,j) is set equal to the average of the surrounding vectors $\bar{\mathbf{v}}^{n}_{(i,j)}$ at the previous iteration minus an adjustment which, in velocity space, is in the direction of the brightness gradient. This algorithm results in velocities that are estimated at points lying midway between the pixels and successive frames. This is due to the fact that the first derivatives g^x, g^y, and g^t, required in the iterative scheme are estimated using finite differences in a $2 \times 2 \times 2$ cube of brightness values [20].

The Bayesian approach

This probabilistic approach consists of estimating the 2-D displacement field at time t using images at t_- and t_+ (Figure 2.1) [28]. The estimated displacement field is a Maximum *A Posteriori* Probability (MAP) estimate that represents the most likely displacement field $\mathbf{d}_t$ on the basis of the two observed fields g_{t_-} and g_{t_+}:

$$\begin{aligned} \hat{\mathbf{d}}_t &= arg \max_{\mathbf{d}_t} P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}, G_{t_+} = g_{t_+}) \\ &= arg \max_{\mathbf{d}_t} \left[P(G_{t_+} = g_{t_+} | \mathbf{D}_t = \mathbf{d}_t, G_{t_-} = g_{t_-}) \cdot P(\mathbf{D}_t = \mathbf{d}_t | G_{t_-} = g_{t_-}) \right]. \end{aligned} \tag{2.18}$$

Two models are therefore needed in the formulation: a *structural model* that models the relationship between observed images and the real displacement field, and a *displacement field model* that ensures the smoothness of the displacement field over all spatial positions (disregarding discontinuities and occlusion effects).

The structural model relies on the assumption of constant intensity along motion trajectory and is expressed by the *Gibbs* distribution (Appendix A) with a potential function U_g equal to the square of the *DPD* (Displaced Pixel Difference). The *DPD* can be regarded as a motion-compensated prediction error measure between the images at t_- and t_+, and is expressed as follows [27]:

$$DPD(\mathbf{x}_i, \mathbf{d}_i) = \tilde{g}(\mathbf{x}_i + (1 - \Delta_t)\mathbf{d}_i, t_+) - \tilde{g}(\mathbf{x}_i - \Delta_t \mathbf{d}_i, t_-), \tag{2.19}$$

where $\mathbf{d}_i$ is defined at $(\mathbf{x}_i, t)$, and $\tilde{g}(\mathbf{x}, t)$ denotes the interpolated intensity at time t and position $\mathbf{x}$ which does not necessarily belong to the sampling grid of the image (refer to Figure 2.1 for illustration of this case). The potential function $U_g(\mathbf{d})$ that describes the ill-posed matching problem of the data by the motion field, is then expressed as follows:

$$U_g(\mathbf{d}) = \sum_{i=1}^{N_d} DPD^2(\mathbf{x}_i, \mathbf{d}_i), \tag{2.20}$$

where N_d is the number of displacement vectors to estimate at time t.

The displacement field $\mathbf{d}$ at time t, on the other hand, is modeled by a 2-D VMRF (Vector Markov Random Field) expressed also by Gibbs distribution whose energy function U_d captures the smoothness of the displacement field as follows:

$$U_d(\mathbf{d}) = \sum_{i=1}^{N_d} \sum_{\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{C}_i} \|\mathbf{d}_i - \mathbf{d}_j\|^2 \tag{2.21}$$

with $\|\cdot\|$ denoting the norm in $\mathcal{R}^2$. This potential function represents the cost associated with the lack of smoothness of the motion field through a first-order neighborhood system η^1 (Figure A.1) where $\mathcal{C}_i$ denotes the ensemble of 2-element cliques [31] at position $\mathbf{x}_i$.

Using the above models, the MAP problem in (2.18) reduces to one of minimizing an energy function $U(\mathbf{d})$, having the following regularized form:

$$U(\mathbf{d}) = \lambda_g U_g(\mathbf{d}) + \lambda_d U_d(\mathbf{d}), \tag{2.22}$$

and which is, in a sense, equivalent to the objective function used by Horn and Schunck in equation (2.16) except that the *motion constraint equation* is replaced by the *DPD* measure. The ratio λ_g/λ_d plays the role of the regularization parameter λ, introduced in the Horn-Schunck approach, and weighs the confidence in the *a priori* model.

The global minimum of $U(\mathbf{d})$ in (2.22) can be calculated using simulated annealing

[27], a stochastic optimization method. However, using a deterministic relaxation with first-order neighborhood η^1, as the one used by Horn and Schunck, will only result in an approximation of the MAP estimate [30]. This is calculated from the following iterative update equation:

$$\hat{\mathbf{d}}_i^{n+1} = \bar{\mathbf{d}}_i^n - \frac{DPD(\mathbf{x}_i, t)}{4\frac{\lambda_d}{\lambda_g} + \|\nabla_d DPD(\mathbf{x}_i, t)\|^2} \nabla_d DPD(\mathbf{x}_i, t). \tag{2.23}$$

The algorithm in (2.23) is a modified version of the Horn-Schunck algorithm in (2.17) whereby:

- Algorithm (2.23) allows computation of displacement vectors for arbitrary positions unlike the Horn-Schunck algorithm. This property is crucial for motion-compensated interpolation applications.

- The *motion constraint equation* is replaced by the the DPD measure in (2.23). This modification is important because it allows intensity pattern tracking thus permitting more accurate intensity derivative computation.

- The spatial intensity derivatives are computed from a separable polynomial model instead of finite difference approximation over a cube as proposed by Horn and Schunck.

The simulations that have been carried out in [30], showed that the MAP estimation algorithm (stochastic and deterministic) resulted in a better estimation than the original Horn and Schunck algorithm which produces over-estimated motion vectors at strong edges, and under-estimated vectors in uniform areas.

2.4 Other important aspects of motion estimation

Since the purpose of this thesis is to estimate dense motion parameters from real TV image sequences, a spatio-temporal gradient approach is chosen. Such method, as shown earlier, requires the minimization of an objective function of the form:

$$U(\mathbf{p}) = U_s(\mathbf{p}) + \lambda U_p(\mathbf{p}), \tag{2.24}$$

where $\mathbf{p}$ is the field of motion parameters (in the continuous case $\mathbf{p}_i = [v_x \quad v_y]^T$ represents one velocity vector at $\mathbf{x}_i$, and in the discrete case $\mathbf{p}_i = [d_x \quad d_y]^T$ represents one displacement vector). The *regularized* objective function in (2.24) consists of a combination of two terms:

1. The structural model term $U_s(\mathbf{p})$ that represents a measure of matching error between images used in the estimation. This term consists of the *motion constraint equation* in the Horn and Schunck algorithm, and the *DPD* in the Bayesian approach. Any other error measure, such as the sample variance over multiple frames, can be used in this term. However, the choice of a particular measure depends directly on the type of motion-compensated processing applications (interpolation, coding, filtering) for which the estimated motion is intended.

2. The smoothness term $U_p(\mathbf{p})$ that measures how well motion field $\mathbf{p}$ conforms to an *a priori* model, such as spatial smoothness with the exception of isolated boundaries. Equation (2.21) is a typical expression of this term through a first-order neighborhood system η^1. A high value of $U_p(\mathbf{p})$ indicates that the motion field is not smooth.

On the other hand, the role of the regularization parameter λ in (2.24) is three-fold:

- Compensate errors due to noise present in the image. Noise in the image can render invalid the structural model. In [20] it is proposed to choose λ as the variance of the noise in the image.

- Control the propagation of motion information in low contrast regions from the neighboring regions. This behavior, however, must be inhibited in occluded areas and across discontinuities where motion is not smooth.

- Minimize the bit rate allocated to image residual and to motion information in a DPCM-like coding scheme where motion parameters are coded and transmitted. As will be seen in Chapter 3, the minimum of $U_s(\mathbf{p})$ and $U_p(\mathbf{p})$ cannot be achieved simultaneously. Hence λ must be chosen in a way to obtain the best compromise.

In this section, several factors that may help in optimizing the objective function in (2.24) and may result in estimating motion field $\mathbf{p}$ that is closest to the true underlying motion are investigated.

2.4.1 Minimization of the objective function

The minimization of $U(\mathbf{p})$ in (2.24) is very complex. For a motion field $\mathbf{p}$ of N_p vectors with each vector consisting of 2 motion parameters (i.e., the horizontal and vertical displacements), the number of variables of the problem is $2N_p$. In addition, these variables are not independent because of the smoothness term $U_p(\mathbf{p})$ that establishes a relation between neighboring motion vectors as in equation (2.21). Standard methods such as Gauss-Jordan elimination are very costly as explained earlier in the Horn and Schunck approach. For this reason the minimization problem is carried out using an iterative relaxation algorithm which from a certain approximation of the solution $\mathbf{p}^n$ is going to produce a better approximation $\mathbf{p}^{n+1}$ such that $U(\mathbf{p}^{n+1}) < U(\mathbf{p}^n)$.

To this end, the minimization problem is expressed in the form of a system of $2N_p$ linear equations of $2N_p$ unknowns. This is accomplished by using Taylor expansion to approximate the objective function $U(\mathbf{p})$ by a quadratic function of $\mathbf{p}$, and then using the necessary condition for optimality:

$$\frac{\partial U(\mathbf{p})}{\partial \mathbf{p}_i} = 0, \qquad i = 1, \cdots, N_p \tag{2.25}$$

for each of the N_p motion vectors.

The resulting system of equations can be solved using deterministic relaxation such as the Gauss-Seidel or Jacobi iterative algorithms. The difference between these two methods is that in the Gauss-Seidel method, the motion vector $\mathbf{p}_i^n$ at position i and iteration n is iterated using the updated neighbor motion vectors (usually from a first-order neighborhood system η^1) at the current iteration. The Jacobi method, however, uses the neighboring motion vectors of the previous iteration $n-1$ to update $\mathbf{p}_i^n$ at iteration n. Hence, the Gauss-Seidel method is faster in convergence than the Jacobi method as in the latter one has to wait a complete field iteration in order to use the latest updated motion vectors. The order of update in both of these methods usually follows a horizontal sequential scan of the motion field being estimated.

The iterative update equations in (2.17) for the Horn-Schunck approach, and (2.23) for the Bayesian approach are examples of relaxation algorithms solved by the Gauss-Seidel method. A detailed derivation of such an algorithm will be presented in Chapter 3 where the new algorithm is discussed.

These deterministic relaxation algorithms fail to converge to the global minimum when the objective function $U(\mathbf{p})$ is not convex. A stochastic relaxation algorithm based on *simulated annealing* has been proposed in [27] and allows to obtain the global minimum. However, the improvement in the subjective quality of the resulting motion vectors applied to motion-compensated interpolation is marginal.

2.4.2 Hierarchical processing

<u>**Motivation**</u>

The approximation, by Taylor expansion, of the objective function $U(\mathbf{p})$ by a quadratic function of $\mathbf{p}$ is weakened when the data structure is characterized by relatively high frequency content such as very sharp edges and noise. Hence, for instance, the second and higher order terms in (2.13) become no more negligible and have to be considered in order to derive the motion constraint equation. Smoothing may reduce this high frequency content, so that the data is closer to a locally linear behavior. The convergence to the global minimum will be more likely since $U(\mathbf{p})$ becomes closer to a convex function. Therefore, the minimization of $U(\mathbf{p})$ is extended over a pyramid of image resolutions. Such hierarchical processing may also allow handling rapid motion since reducing high frequency image content allows to perform matching at larger displacements [29].

The hierarchical extension

Various multiresolution methods have been proposed for motion estimation [5][13][29]. One class of such methods is based on a non-recursive multigrid (coarse-to-fine resolution) approach [29]. It consists of generating a pyramid of varying image resolutions from the lowest resolution at the top level ($k = L$) to the full resolution at the bottom level ($k = 0$) of the pyramid. Figure 2.4(a) is an example of a 4-level ($L = 4$) pyramid of 1D data resolution. A second pyramid for motion fields is then constructed as follows. The estimation starts at the top level ($k = L$) of the pyramid

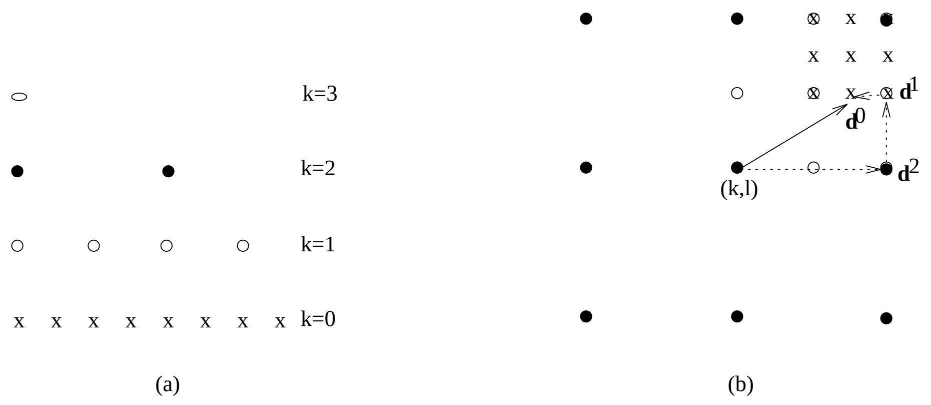


Figure 2.4: Schematic (1D) representation of a 4-level pyramid (a) for hierarchical data representation; (b) illustration of hierarchical displacement vector update over 3 image resolutions.

(coarse resolution), where the number of motion vector sites is small, and hence a minimum is located very quickly. Then, the resulting estimate from this coarse level is interpolated to the next finer resolution level ($k = L - 1$) where it is used as an initial solution for the estimation at this finer resolution level. This hierarchical process is repeated until the full resolution estimate at the bottom of the pyramid ($k = 0$) is obtained. An illustration of this hierarchical displacement vector update for $L = 2$ is shown in Figure 2.4(b). The end point of the displacement vector at position (k, l) for each resolution level k is denoted by $\mathbf{d}^k$, and the displacement update vector at each resolution level is shown by a dotted line.

Choice of the smoothing filter

Generation of each resolution level of the image pyramid consists of spatial subsampling, preceded by filtering in order to avoid aliasing effects which can significantly deteriorate the quality of motion estimates. However, the images do not have to be subsampled when moving up the pyramid, as subsampling causes data loss. This loss may affect the performance of spatio-temporal gradient methods that require the calculation of derivatives. Hence a "constant-width" pyramid for images and a regular pyramid for motion fields are sometimes used in such algorithms.

The choice of the optimal smoothing filter (low-pass or band-pass) is not yet clear. Enkelmann [13] has used the circularly symmetric Gaussian low-pass filter with radius $R = 4$ pixels and spatial variance $\sigma^2 = 2.5$. The 1D magnitude response

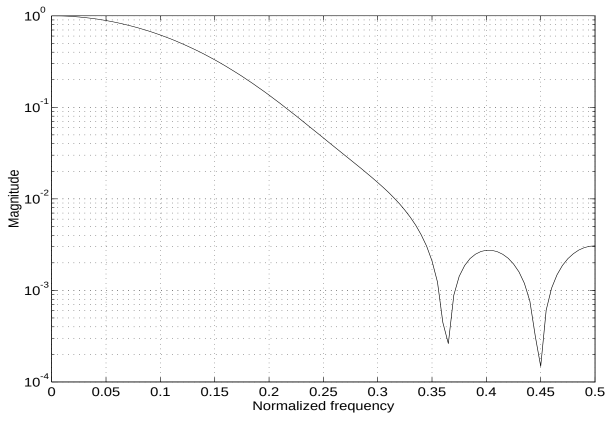


Figure 2.5: Magnitude of the frequency response of the Gaussian filter used in the generation of the pyramid of resolutions.

of such a filter is shown in Figure 2.5. Note that the large transition band in the magnitude response is essential in reducing the ringing effects near the contours.

On the other hand, the filtering may introduce unwanted artifacts, such as confusion of objects with background, which usually leads to locally unreliable estimates. To overcome this, a *multiscale* estimation (another class of coarse-to-fine algorithms) has been proposed for globally smooth linear motion (displacements) based on Markov hierarchical model [17] where instead of subsampling, the size of a block in which all estimates are kept constant varies from level to level. In this approach, Markov models at higher levels of the pyramid are rigorously derived from the full resolution model. This method has been reported to give results similar to those obtained by stochastic monoresolution techniques [28] and superior to those obtained by deterministic multiresolution methods using the smoothing filters [30].

2.4.3 Modeling of discontinuities and occlusion areas

Gradient-based motion estimation methods allow dense motion measurements, but generally suffer from severe shortcomings especially near discontinuities and in occlusion areas. Moreover, for real TV images the underlying motion is piecewise continuous rather than globally continuous. Taking into account motion discontinuities is thus important when accurate motion estimates are required.

The smoothness term $U_p(\mathbf{p})$ in (2.24) captures the smoothness of the motion field $\mathbf{p}$ to estimate. However, the smoothness constraint is violated at the boundaries of an object moving across a still background. Hence, it was proposed [27][16] to include a *line field* l in order to inhibit the smoothness constraint at certain boundaries, and therefore to allow the estimation of a piecewise continuous motion field. The typical smoothness term in (2.21) is then modified [27] as follows:

$$U_p(\mathbf{p}, l) = \sum_{i=1}^{N_d} \sum_{\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{C}_i} \|\mathbf{p}_i - \mathbf{p}_j\|^2 [1 - l(\mathbf{x}_i, \mathbf{x}_j)], \tag{2.26}$$

where $l(\mathbf{x}_i, \mathbf{x}_j)$ is the binary-valued line element defined between two pixel positions at $\mathbf{x}_i$ and $\mathbf{x}_j$ (0 and 1 represent respectively absence and presence of a discontinuity). If the line element separating motion vectors from clique $\{\mathbf{x}_i, \mathbf{x}_j\}$ is "turned on" ($l(\mathbf{x}_i, \mathbf{x}_j) = 1$), there is no cost associated with that clique, and hence the cost function

$U_p(\mathbf{p}, l)$ is not increased. However, if the line element is "turned off" $(l(\mathbf{x}_i, \mathbf{x}_j) = 0)$, there exists an associated cost if the motion vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ depart from the assumed *a priori* model.

The above extension implies simultaneous estimation of two fields: motion field and line field. It is interesting to note that if all the line elements in the line field are "turned on", then the smoothness term in (2.26) is null and therefore is at minimum. A model of the line field is therefore necessary. A binary MRF model is proposed in [27] leading to the addition of $U_l(l)$ to the objective function in (2.24). This term has the role of introducing penalties discouraging certain configurations such as the introduction of a line element on a non-edge site, or the occurrence of an isolated motion site.

On the other hand, it was shown in [16] that the detection of motion discontinuities alone is not sufficient, but that the processing of occlusion areas (along with motion discontinuities) is necessary in order to maintain a high quality of estimation near motion discontinuities.

This approach has been investigated in [9] for the estimation of a displacement field from 3 images at t_-, t, and t_+. The three-state (M, E, C) *occlusion field* o defined on the same sampling lattice as the motion field, was defined as follows: M indicates that the point is visible over the entire interval from t_- to t_+, E represents newly exposed points over the interval from t_- to t, and C indicates covered points over the interval from t to t_+. An illustration of this three-state occlusion field is depicted in Figure 2.2 where point $\mathbf{z}$ should be labeled as M, point $\mathbf{y}$ as C, and point $\mathbf{w}$ as E. Note that the number of states an occlusion tag can take, depends on the number of fields used in the estimation algorithm.

Using this definition of the occlusion field o, along with that of the line field l, an additional term $U_o(o, l)$ is used in the objective function (2.24) which can be expressed now as follows:

$$U(\mathbf{p}, o, l) = U_s(\mathbf{p}, o) + \lambda_p U_p(\mathbf{p}, l) + \lambda_o U_o(o, l) + \lambda_l U_l(l) \tag{2.27}$$

where the λ's denote weights associated with each term. Note that the structural term $U_s(\mathbf{p}, o)$ becomes dependent on the occlusion tag in the sense that if a point is labeled as occluded, the fields on which this point is invisible are discarded in the evaluation of U_s. The additional cost introduced by $U_o(o, l)$ depends on whether or not there is a discontinuity between two vector positions, and reflects the fact that occlusion tags should only appear near motion discontinuities.

The same deterministic minimization techniques, as discussed earlier, can be applied in order to minimize the modified objective function in (2.27). However, the number of variables of this minimization problem is enormous. It consists of $Card(\mathbf{p})$ (the cardinality of the motion vector $\mathbf{p}$) motion parameters, plus one or two line elements (depending whether inside or at the boundary), plus one occlusion tag per sampling point. For this reason, an interleaved optimization approach is used whereby a sequential minimization with respect to each of the three fields is carried out, while maintaining the other two fields fixed. This process is repeated until suitable convergence is achieved.

The processing of discontinuities and occlusions has shown [9] to be helpful in obtaining a more realistic estimate of the motion field especially in presence of acceleration. This is due to the fact that accelerated motion tends to generate larger occluded regions and hence the computation of these regions becomes vital for obtaining good motion estimates. The estimation of occlusion and line fields will be discussed in more detail in Chapter 4 where they will be incorporated into the new motion estimation algorithm.

2.4.4 Multiframe processing

Most of the existing motion estimation methods use two fields to estimate motion parameters (displacements, velocities). But motion estimation, and in particular the identification of occlusion areas, can be considerably improved by using multiple images as discussed in [8] where 3 fields have been used. Hence, considering a matching error measure over several images should allow a more robust pixel-matching (with a certain allowable range of illumination variation). Also, accumulating the occlusion tags over several fields should allow easier identification of occlusion regions (i.e., the newly exposed and covered points).

The only problem in this approach is that the assumption of the locally translational motion model is no more valid over multiple fields. This is illustrated in Figure 2.6 where the real motion trajectory of point $(\mathbf{x}, t)$ is shown in solid line. In this

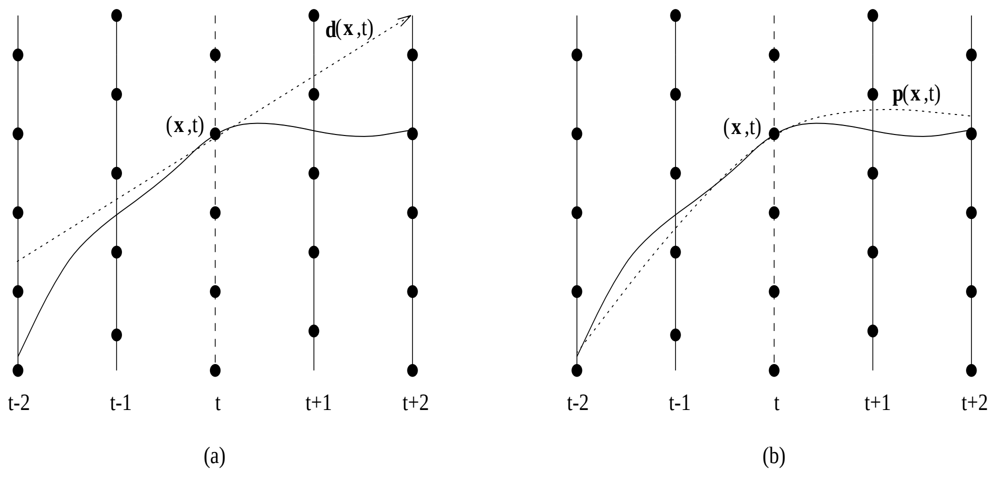


Figure 2.6: Motion estimation using 5 fields and (a) first-order motion trajectory model; (b) second-order motion trajectory model.

example, the estimation is done for field at time t using the five fields ($N = 5$) at $t-2$, $t-1$, t, $t+1$, $t+2$. The estimated displacement vector $\mathbf{d} = [d_x \;\; d_y]^T$ at $(\mathbf{x}, t)$ obtained in the case of a first-order trajectory model (i.e., linear model) is shown in dotted line in Figure 2.6(a). However, for a second-order trajectory model [12] (i.e., both velocity $\mathbf{v}$ and acceleration $\mathbf{a}$ are included), the vector $\mathbf{p} = [v_x \;\; v_y \;\; a_x \;\; a_y]^T$ of motion parameters is estimated. This results in the quadratic trajectory drawn in dotted line in Figure 2.6(b) through the point $(\mathbf{x}, t)$.

Note that the use of a higher order motion model in multiframe processing should result in a better approximation of the real motion trajectory, and hence reduction of the matching error in the structural term. This is expected to have consequence in motion-compensated interpolation applications.

Hence, for multiframe processing (i.e., $N \geq 3$), the consideration of a higher-order motion model should be an asset in motion-compensated applications. In Chapter 3, a detailed formulation of a new spatio-temporal gradient motion estimation method over multiple frames is discussed along with some simulation results. A second-order (quadratic) motion model is used to model motion trajectories over a variable number of fields (selected by the user). The proposed algorithm relies on a deterministic relaxation approach implemented over a pyramid of image resolutions. Subjective and quantitative comparison between the use of a linear and quadratic motion trajectory models with multiframe processing is also discussed. The proposed algorithm is then extended in Chapter 4 to handle occlusions and motion discontinuities.

Chapter 3

Estimation of Motion Trajectories with Acceleration

In this chapter, the estimation of trajectories for accelerated motion from image sequences is proposed. Unlike in many other approaches that assume linear trajectories, a higher order model that incorporates both velocity and acceleration is considered. This model corresponds better to real case situations especially when the estimation is carried out over several images.

One of the advantages of using accelerated over linear trajectories is in motion-compensated processing over multiple images. This is due to the fact that over longer time frame, a quadratic motion model is capable of providing a better intensity match along trajectories than the linear model. In particular, the standards conversion problem, has been addressed in the presence of accelerated motion in [35] where it has been demonstrated that by taking into account acceleration during frame rate conversion and deinterlacing, a superior result can be achieved. In the above work, however, it was assumed that the velocity and acceleration parameters in an image sequence are known *a priori*. However, in real TV sequences these motion parameters are unknown. Hence, a good estimate of the velocity and acceleration parameters is essential.

In Section 3.1 the notion of a motion trajectory is introduced and the motion estimation problem is defined. The algorithm for the estimation of dense accelerated motion fields is formulated in Section 3.2 using Gibbs-Markov models linked together by the *Maximum A Posteriori* (MAP) probability criterion. This results in a minimization of a regularized cost function. The solution that is proposed in Section 3.3 is based on deterministic relaxation implemented over a pyramid of resolutions. The test images that are used to simulate the proposed algorithm are presented in Section 3.4 along with a description of the measures that will be used to evaluate the validity of motion estimates. The parameters used in the estimation algorithm are then optimized in Section 3.5. Finally, experimental results of the proposed algorithm are presented in Section 3.6 along with a comparison between the linear and quadratic motion trajectory models.

3.1 Definition of motion trajectory

To describe motion with acceleration, the concept of motion trajectory is used [11]. The projection of each scene point traces out a trajectory in the image plane W during the time it is visible in the image. Hence, the motion in the image sequence is characterized by the collection of all such trajectories. An illustration of a typical trajectory of the center of a circle moving across the image plane is shown in Figure 3.1. The trajectory starts at the time $t_i(\mathbf{x}, t) = t_-$ when the point first becomes visible, and ends at time $t_f(\mathbf{x}, t) = t_+$ when the point disappears. The trajectory of point $(\mathbf{x}, t)$ can be specified by the function $\mathbf{c}(\tau; \mathbf{x}, t)$ which gives the spatial position at time τ of an image point located at position $\mathbf{x}$ at time t.

For $\tau \neq t$, let $V(\tau; t)$ define a subset of W at time t consisting of pixels that are visible over the entire interval between t and τ:

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

For $\tau > t$, $W - V(\tau; t)$ is the set of pixels covered or leaving the image between t and τ, while for $\tau < t$, $W - V(\tau; t)$ is the set of pixels exposed or introduced into the image between τ and t.

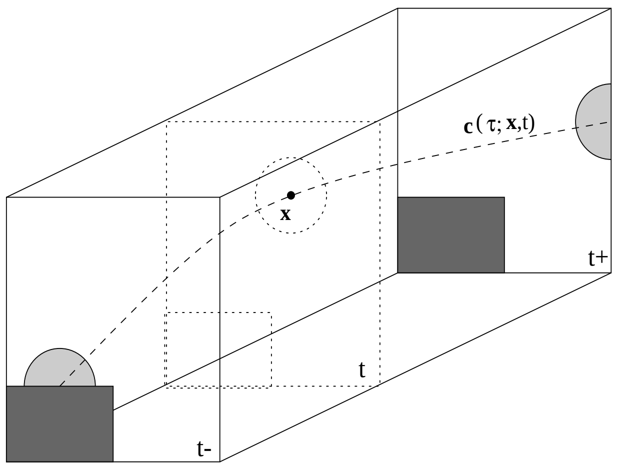


Figure 3.1: Trajectory of a projected scene point in 3-D x-y-t space.

From this mathematical description of motion in the image plane, one can derive the *optical flow* which consists of the instantaneous velocities in the image:

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

For linear trajectory model, the displacement field, defined in the previous chapter, is used. It can be described as follows:

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

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

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

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

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

To make the estimation problem tractable, we model each motion trajectory $\mathbf{c}$ by a parametric function $\mathbf{c}^{\mathbf{P}}$ of a vector of motion parameters $\mathbf{p}$ [12]. Hence for the linear trajectory model in (3.5), $\mathbf{p} = \mathbf{v}$. For quadratic trajectories based on velocity and acceleration, $\mathbf{p} = [\mathbf{v}^T \ \ \mathbf{a}^T]^T$ and

$$\mathbf{c}^{\mathbf{p}}(\tau; \mathbf{x}, t) = \mathbf{x} + \mathbf{v}(\mathbf{x}, t)(\tau - t) + \mathbf{a}(\mathbf{x}, t)(\tau - t)^2. \tag{3.6}$$

Note that $\mathbf{v}(\mathbf{x}, t)$ is the instantaneous velocity, and $\mathbf{a}(\mathbf{x}, t)$ is the constant acceleration over the length of the trajectory, both at $(\mathbf{x}, t)$. Equation (3.6) can be rewritten as follows:

$$\mathbf{c}^{\mathbf{p}}(\tau; \mathbf{x}, t) = \mathbf{x} + \Delta_\tau \mathbf{p}(\mathbf{x}, t) \tag{3.7}$$

with

$$\Delta_\tau = \begin{bmatrix} \tau - t & 0 & (\tau - t)^2 & 0 \\ 0 & \tau - t & 0 & (\tau - t)^2 \end{bmatrix}, \qquad \mathbf{p}(\mathbf{x}, t) = \begin{bmatrix} v_x(\mathbf{x}, t) \\ v_y(\mathbf{x}, t) \\ a_x(\mathbf{x}, t) \\ a_y(\mathbf{x}, t) \end{bmatrix}. \tag{3.8}$$

The goal of this chapter is to estimate the field of motion parameters $\mathbf{p}_t$ at time t defined on the 2-D dense lattice $(\Lambda_p)_t$ which corresponds to the sampling grid of the image. Let g_t be an image at time t, and let

$$\mathcal{I}_t = \{\tau : g_{t+\tau} \text{ is used in estimation of } \mathbf{p}_t\} \tag{3.9}$$

with $Card(\mathcal{I}_t) = N$. The estimation of motion at time t is carried out over images $\{g_{t+\tau}\}$ such that $\tau \in \mathcal{I}_t$. Figure 3.2 illustrates an example of quadratic trajectory over N fields characterized by the vector of motion parameters $\mathbf{p}(\mathbf{x}, t)$.

Assuming that the following is true

$$\forall \tau \in \mathcal{I}_t \Rightarrow \begin{cases} t_i(\mathbf{x}, t) \leq t + \tau \leq t_f(\mathbf{x}, t), & \text{and} \\ \mathbf{x} \in V(\tau; t) \end{cases} \tag{3.10}$$

implies that the estimation is performed only on moving or stationary (i.e., not occluded) pixels. Hence, the effects of occlusions are not considered in the estimation algorithm. For the remaining of this chapter it is assumed that (3.10) is true which is not the case in real sequences. For this reason, in Chapter 4, an occlusion model will be incorporated into the estimation algorithm, in conjunction with a motion discontinuity model, in order to obtain better estimates in occluded regions.

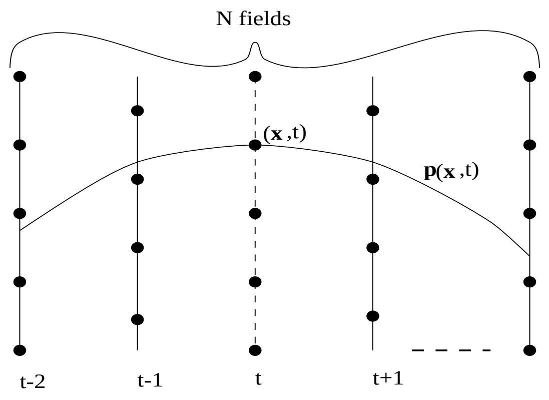


Figure 3.2: Estimated quadratic trajectory $\mathbf{c}^{\mathbf{p}}(\tau;\mathbf{x},t)$ at $(\mathbf{x},t)$.

3.2 Formulation

The estimation of accelerated motion trajectories is tackled using a spatio-temporal gradient method based on a statistical approach. This method has been overviewed in Chapter 2 for the estimation of 2-D displacement fields, and will be extended in detail in this chapter to handle the estimation of the motion field $\mathbf{p}$ defined for quadratic trajectories.

In the following sections MRFs, characterized by Gibbs distributions (GD), are used to model both the matching error along motion trajectories and the motion field $\mathbf{p}$. In the case of MAP estimation [28], it was shown that the use of GD can lead directly to a cost function of the same form as equation (2.22).

According to the MAP criterion, the estimated motion field $\mathbf{p}_t$ at time t is the most likely motion field $\mathbf{p}_t$ based on observations $\mathcal{G}_t = \{g_{t+\tau} : \tau \in \mathcal{I}_t\}$. Using the Bayes' rule, MAP estimation for $\mathbf{p}_t$ is a modified version of equation (2.18), and is expressed as follows:

$$\begin{aligned}\hat{\mathbf{p}}_t &= \arg\max_{\mathbf{p}_t} P(\mathbf{p}_t|\mathcal{G}_t) \\ &= \arg\max_{\mathbf{p}_t} [P(\mathcal{G}_t^n|\mathbf{p}_t, g_{t_n}) \cdot P(\mathbf{p}_t|g_{t_n})]\end{aligned} \tag{3.11}$$

where $t_n = t + \tau_n$ is an arbitrary chosen time instant such that $\tau_n \in \mathcal{I}_t$, and $\mathcal{G}_t^n = \{g_{t+\tau} : \tau \in \mathcal{I}_t - \{\tau_n\}\}$. It is assumed that vectors $\mathbf{p}_t$ are samples from a vector random field $\mathbf{P}_t$, and that images g_t are samples from the luminance random fields G_t. The first probability $P(\mathcal{G}_t^n|\mathbf{p}_t, g_{t_n})$ in (3.11) is determined by the structural model relating motion to the observed image, and $P(\mathbf{p}_t|g_{t_n})$ is determined by the motion trajectory model. These two models are discussed in the next two sections.

3.2.1 Structural model and matching error

The structural model follows directly from the constant intensity assumption along motion trajectories. The conditional probability $P(\mathcal{G}_t^n|\mathbf{p}_t, g_{t_n})$ depends directly on the intensity variation along motion trajectories. This variability is assumed to be independent for each distinct trajectory on the lattice $(\Lambda_p)_t$, and similarly to [31] is modeled here by independent and identically distributed Gaussian random variables. Since Gaussian distribution is a special case of GD, the intensity variation along motion trajectories can be expressed by the GD (equation (A.3)). The conditional distribution along each trajectory $\mathbf{c}^{\mathbf{p}_i}$ at a certain pixel site $(\mathbf{x}_i, t) \in (\Lambda_p)_t$ is therefore expressed as follows:

$$P'(\mathcal{G}_t^n|\mathbf{p}_t, g_{t_n}) = \frac{1}{Z'_s} e^{-\frac{U_s^{(i)}(\mathbf{p})}{\beta'_s}}, \tag{3.12}$$

where $U_s^{(i)}(\mathbf{p})$ is an energy function that measures departure of the observations from the structural model, and hence must represent a measure of the intensity matching error between the $\mathit{Card}(\mathcal{I}_t) = N$ fields used in the estimation. Since the main interest in this chapter is to estimate motion fields for applications related to motion-compensated interpolation, the energy $U_s^{(i)}(\mathbf{p})$ at pixel site $(\mathbf{x}_i, t) \in (\Lambda_p)_t$ (refer to Figure 3.2) is defined as the sample variance:

$$U_s^{(i)}(\mathbf{p}) = \sum_{k \in \mathcal{I}_t} \left[\tilde{g}(\mathbf{x}_i^k, t+k) - \zeta(\mathbf{x}_i, t)\right]^2 \tag{3.13}$$

with

$$\zeta(\mathbf{x}_i, t) = \frac{1}{N} \sum_{\tau \in \mathcal{I}_t} \tilde{g}(\mathbf{x}^\tau, t+\tau). \tag{3.14}$$

$\tilde{g}(\mathbf{x}_i^k, t+k)$ is the interpolated intensity at time $t+k$ and position $\mathbf{x}_i^k$ defined as follows:

$$\begin{aligned} \mathbf{x}_i^k &= \mathbf{x}_i + \mathbf{v}(\mathbf{x}_i, t)k + \mathbf{a}(\mathbf{x}_i, t)k^2 \\ &= \mathbf{x}_i + \Delta_{(t+k)}\mathbf{p}_i \end{aligned} \tag{3.15}$$

with $\Delta_{(t+k)}$, and $\mathbf{p}_i = \mathbf{p}(\mathbf{x}_i, t)$ defined in (3.8).

The total conditional distribution is then a product over the entire field of N_P pixel sites in $(\Lambda_p)_t$, of the distributions $P'(\mathcal{G}_t^n|\mathbf{p}_t, g_{t_n})$ along each trajectory:

$$\begin{aligned} P(\mathcal{G}_t^n|\mathbf{p}_t, g_{t_n}) &= \prod_{i=1}^{N_P} \frac{1}{Z_s'} e^{-\frac{U_s^{(i)}(\mathbf{p})}{\beta_s'}} \\ &= \frac{1}{Z_s} e^{-\frac{U_s(\mathbf{p})}{\beta_s}}, \end{aligned} \tag{3.16}$$

where, $U_s(\mathbf{p})$ is known as the structural model term:

$$\begin{aligned} U_s(\mathbf{p}) &= \sum_{i=1}^{N_P} U_s^{(i)}(\mathbf{p}) \\ &= \sum_{i=1}^{N_P} \left\{ \sum_{k \in \mathcal{I}_t} \left[\tilde{g}(\mathbf{x}_i^k, t+k) - \zeta(\mathbf{x}_i, t) \right]^2 \right\}, \end{aligned} \tag{3.17}$$

which constitutes the main term of the objective function (2.24) discussed in Chapter 2, and whose minimization yields a MAP estimate of the motion field $\mathbf{p}$.

3.2.2 Motion trajectory model

It was assumed earlier that motion vectors $\mathbf{p}_t$ are samples from a vector random field $\mathbf{P}_t$. Defining $\mathbf{P}_t$ as a vector MRF on the lattice $(\Lambda_p)_t$, the *a priori* distribution $P(\mathbf{p}_t|g_{t_n})$ can be expressed as a Gibbs distribution:

$$P(\mathbf{p}_t|g_{t_n}) = \frac{1}{Z_p} e^{-\frac{U_p(\mathbf{p})}{\beta_p}}, \tag{3.18}$$

where $U_p(\mathbf{p})$ is the energy function that captures the desired smoothness property of the motion field through a first-order neighborhood system η^1 (Figure A.1):

$$U_p(\mathbf{p}) = \sum_{i=1}^{N_P} \sum_{\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{C}_i} (\mathbf{p}_i - \mathbf{p}_j)^T \boldsymbol{\Gamma} (\mathbf{p}_i - \mathbf{p}_j). \tag{3.19}$$

Note that $\mathbf{p}_i$ and $\mathbf{p}_j$ are parameter vectors for trajectories passing through $(\mathbf{x}_i, t)$ and $(\mathbf{x}_j, t)$, respectively, and $\boldsymbol{\Gamma}$ is a 4×4 positive definite weight matrix that was introduced in order to permit different weighting of horizontal and vertical motion parameters as well as of lower and higher order motion parameters (i.e., velocity and acceleration parameters). For this purpose, $\mathbf{\Gamma}$ was chosen to be a diagonal matrix of the form:

$$\mathbf{\Gamma} = \begin{bmatrix} w_{v_x} & 0 & 0 & 0 \\ 0 & w_{v_y} & 0 & 0 \\ 0 & 0 & w_{a_x} & 0 \\ 0 & 0 & 0 & w_{a_y} \end{bmatrix}, \tag{3.20}$$

where w_{v_x}, w_{v_y}, w_{a_x}, and w_{a_y} are the respective weights assigned to each of the four motion parameters in $\mathbf{p}$. This weight matrix $\mathbf{\Gamma}$ was discussed in [32] where its entries, not necessary on the diagonal, were chosen as functions of the observations g_{t_n} in order to allow suitable adaptation of the smoothness property to the local image structure. Note that when $\mathbf{\Gamma}$ is the identity matrix, the Euclidean norm results and hence, the smoothness term in (3.19) is reduced to the same form as equation (2.21) which was introduced earlier for the linear trajectory model.

3.2.3 Derivation of the objective function

Combining now the calculated conditional distributions of the previous sections into equation (3.11), the following results:

$$\begin{aligned} \hat{\mathbf{p}}_t &= arg \max_{\mathbf{p}_t} P(\mathbf{p}_t|\mathcal{G}_t) \\ &= arg \max_{\mathbf{p}_t} \left\{ \frac{1}{Z} e^{-U(\mathbf{p})} \right\} \\ &= arg \min_{\mathbf{p}_t} U(\mathbf{p}), \end{aligned} \tag{3.21}$$

where Z is a new normalizing constant incorporating Z_s and Z_p, and $U(\mathbf{p})$ is the new energy function defined as follows:

$$U(\mathbf{p}) = U_s(\mathbf{p}) + \lambda_p U_p(\mathbf{p}), \tag{3.22}$$

with $\lambda_p = 1/\beta_p$, and $\beta_s = 1$. Like in equation (2.24) the regularized form of the objective function $U(\mathbf{p})$ follows directly from the MAP criterion. λ_p is a regularization parameter that plays a vital role in weighting the importance of the *a priori* motion trajectory model with respect to the structural model.

3.3 Solution method

The optimization of the objective function in (3.22), which is a non-linear, non-quadratic, and non-convex function, has been briefly discussed in Section 2.4.1. Complex non-linear optimization approaches are not suitable here. Hence, a deterministic relaxation algorithm resulting in an iterative update equation of the same form as equation (2.23) is derived in this section. This algorithm is derived by approximating $U(\mathbf{p})$ by a quadratic function of $\mathbf{p}$, and then using the necessary condition for optimality expressed in (2.25) to derive a linear system of the form $\mathbf{A}\mathbf{p}_i = \mathbf{b}$ for each of the N_P motion vector positions in $(\Lambda_p)_t$. The solutions of these dependent linear systems are then calculated iteratively by using the Gauss-Seidel relaxation approach, discussed in Chapter 2.

3.3.1 Approximation of the objective function

Approximation of the objective function in (3.22) by a quadratic function of $\mathbf{p}$ is made possible by using the Taylor expansion of $\tilde{g}(\mathbf{x}_i^k, t+k) = \tilde{g}(\mathbf{x}_i + \Delta_{(t+k)}\mathbf{p}_i, t+k)$ in (3.17) about some intermediate solution $\dot{\mathbf{p}}_i$:

$$\tilde{g}(\mathbf{x}_i + \Delta_{(t+k)}\mathbf{p}_i, t+k) \approx \tilde{g}(\mathbf{x}_i + \Delta_{(t+k)}\dot{\mathbf{p}}_i, t+k) + \nabla_{\mathbf{p}}^T\tilde{g}(\mathbf{x}_i + \Delta_{(t+k)}\dot{\mathbf{p}}_i, t+k)(\mathbf{p}_i - \dot{\mathbf{p}}_i), \tag{3.23}$$

with $\nabla_{\mathbf{p}}\tilde{g}(\cdot)$ being the gradient of $\tilde{g}(\cdot)$ with respect to the motion vector $\mathbf{p}$ expressed as:

$$\begin{aligned}
\nabla_{\mathbf{p}}\tilde{g}(\cdot) &= \Delta_{(t+k)}^T . \nabla_{\mathbf{x}}\tilde{g}(\cdot) \\
&= \begin{bmatrix} k & 0 \\ 0 & k \\ k^2 & 0 \\ 0 & k^2 \end{bmatrix} . \begin{bmatrix} \frac{\partial \tilde{g}(\cdot)}{\partial x} \\ \frac{\partial \tilde{g}(\cdot)}{\partial y} \end{bmatrix} \\
&= \left[k\frac{\partial \tilde{g}(\cdot)}{\partial x} \quad k\frac{\partial \tilde{g}(\cdot)}{\partial y} \quad k^2\frac{\partial \tilde{g}(\cdot)}{\partial x} \quad k^2\frac{\partial \tilde{g}(\cdot)}{\partial y} \right]^T .
\end{aligned} \tag{3.24}$$

Substituting equation (3.23) into (3.17), $U(\mathbf{p})$ can be approximated as follows:

$$U(\mathbf{p}) \approx \sum_{i=1}^{N_P} \left\{ \sum_{k\in\mathcal{I}_t} [r_i^k(\dot{\mathbf{p}}_i) + (\mathbf{s}_i^k(\dot{\mathbf{p}}_i))^T(\mathbf{p}_i - \dot{\mathbf{p}}_i)]^2 + \lambda_p \sum_{\{\mathbf{x}_i,\mathbf{x}_j\}\in\mathcal{C}_i} (\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{\Gamma}(\mathbf{p}_i - \mathbf{p}_j) \right\}, \tag{3.25}$$

where

$$r_i^k(\dot{\mathbf{p}}_i) = \tilde{g}(\mathbf{x}_i + \Delta_{(t+k)}\dot{\mathbf{p}}_i, t+k) - \frac{1}{N}\sum_{\tau\in\mathcal{I}_t} \tilde{g}(\mathbf{x}_i + \Delta_{(t+\tau)}\dot{\mathbf{p}}_i, t+\tau), \tag{3.26}$$

and

$$(\mathbf{s}_i^k(\dot{\mathbf{p}}_i))^T = \nabla_{\mathbf{p}}^T \tilde{g}(\mathbf{x}_i + \Delta_{(t+k)}\dot{\mathbf{p}}_i, t+k) - \frac{1}{N}\sum_{\tau\in\mathcal{I}_t} \nabla_{\mathbf{p}}^T \tilde{g}(\mathbf{x}_i + \Delta_{(t+\tau)}\dot{\mathbf{p}}_i, t+\tau). \tag{3.27}$$

3.3.2 The iterative algorithm

Now that the objective function $U(\mathbf{p})$ is a quadratic function of $\mathbf{p}$, the global minimum $\mathbf{p}_{t_{min}}$ which corresponds to an approximation of the MAP estimate $\hat{\mathbf{p}}_t$ (equation (3.21)) can be obtained by establishing the necessary condition for optimality over the entire motion field at time t:

$$\frac{\partial U(\mathbf{p})}{\partial \mathbf{p}_i} = 0, \qquad i = 1, \cdots, N_P. \tag{3.28}$$

The derivative of $U(\mathbf{p})$ with respect to the motion vector $\mathbf{p}_i$ defined at pixel site $\mathbf{x}_i \in (\Lambda_p)_t$ is expressed as follows:

$$\frac{\partial U(\mathbf{p})}{\partial \mathbf{p}_i} = \sum_{k\in\mathcal{I}_t} \frac{\partial}{\partial \mathbf{p}_i}\left[r_i^k(\dot{\mathbf{p}}_i) + (\mathbf{s}_i^k(\dot{\mathbf{p}}_i))^T(\mathbf{p}_i - \dot{\mathbf{p}}_i)\right]^2 + 2\lambda_p \frac{\partial}{\partial \mathbf{p}_i} \sum_{\{\mathbf{x}_i,\mathbf{x}_j\}\in\mathcal{C}_i} (\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{\Gamma}(\mathbf{p}_i - \mathbf{p}_j). \tag{3.29}$$

After some straightforward differentiations, equation (3.28) becomes

$$\sum_{k\in\mathcal{I}_t}\left(r_i^k(\dot{\mathbf{p}}_i)\mathbf{s}_i^k(\dot{\mathbf{p}}_i)\right) + \left[\sum_{k\in\mathcal{I}_t}\left(\mathbf{s}_i^k(\dot{\mathbf{p}}_i)(\mathbf{s}_i^k(\dot{\mathbf{p}}_i))^T\right)\right]\cdot(\mathbf{p}_i - \dot{\mathbf{p}}_i) + 2\xi_i\lambda_p\cdot\mathbf{\Gamma}\cdot(\mathbf{p}_i - \bar{\mathbf{p}}_i) = 0, \tag{3.30}$$

where $\bar{\mathbf{p}}_i$ is the average motion vector at position $\mathbf{x}_i$ given by:

$$\bar{\mathbf{p}}_i = \frac{1}{\xi_i}\sum_{j\in\eta^1(i)} \mathbf{p}_j. \tag{3.31}$$

$\eta^1(i)$ is the first-order neighborhood at position $\mathbf{x}_i$ resulting in $\xi_i = 4$ everywhere except at the boundaries of the lattice $(\Lambda_p)_t$ where $\xi_i < 4$ (in particular, $\xi_i = 2$ at the four corners).

Hence, the global minimum $\mathbf{p}_{t_{min}}$ of the quadratically approximated objective function (3.25) is determined by solving (3.30) for $\mathbf{p}_i$ at each of the N_P pixel sites on the lattice $(\Lambda_p)_t$. But since each solution $\mathbf{p}_i$ depends on $\bar{\mathbf{p}}_i$ which is a function of motion vectors in the neighborhood of $\mathbf{x}_i$, then an iterative relaxation method is needed. Considering that each iteration n consists of a full scan of the field at time t, and letting $\dot{\mathbf{p}}_i^n = \bar{\mathbf{p}}_i^{n-1}$ at each iteration, then $\mathbf{p}_i^n$ is updated at iteration n by solving the following linear system:

$$\mathbf{A}_i^n \cdot \mathbf{p}_i^n = \mathbf{b}_i^n, \tag{3.32}$$

which directly follows from equation (3.30) with:

$$\mathbf{A}_i^n = \left[\sum_{k \in \mathcal{I}_t} \left(\mathbf{s}_i^k(\bar{\mathbf{p}}_i^n)(\mathbf{s}_i^k(\bar{\mathbf{p}}_i^n))^T \right) \right] + 2\xi_i \lambda_p \cdot \mathbf{\Gamma}, \tag{3.33}$$

and

$$\mathbf{b}_i^n = \left(\left[\sum_{k \in \mathcal{I}_t} \left(\mathbf{s}_i^k(\bar{\mathbf{p}}_i^n)(\mathbf{s}_i^k(\bar{\mathbf{p}}_i^n))^T \right) \right] + 2\xi_i \lambda_p \cdot \mathbf{\Gamma} \right) \cdot \bar{\mathbf{p}}_i^n - \sum_{k \in \mathcal{I}_t} \left(r_i^k(\bar{\mathbf{p}}_i^n) \mathbf{s}_i^k(\bar{\mathbf{p}}_i^n) \right). \tag{3.34}$$

The deterministic relaxation method is based on the Gauss-Seidel approach, which calculates $\bar{\mathbf{p}}_i^n$ at iteration n using the latest updated neighbor motion vectors at the current iteration. That is why for a horizontal scan at each iteration, the average motion vector $\bar{\mathbf{p}}_i^n$ makes use of the updated left and top neighbor motion vectors at iteration n, as well as of the right and bottom neighboring motion vectors at iteration $n-1$ (since the ones at iteration n are not yet available). This characterization of $\bar{\mathbf{p}}_i^n$ changes obviously with the scan mode.

The Gauss-Seidel relaxation method is used to iterate the complete motion field, and this process is repeated until a convergence is achieved. Convergence is detected by monitoring the oscillatory decreasing behavior of $U^n(\mathbf{p})$ in (3.22) after each iteration, and hence the following condition:

$$\left| \frac{U^n(\mathbf{p}) - U^{n-1}(\mathbf{p})}{U^n(\mathbf{p})} \right| < \epsilon \tag{3.35}$$

can serve as a robust stopping criterion for convergence.

3.3.3 Spatial interpolation

The motion vector $\mathbf{p}_i$ defined at pixel $\mathbf{x}_i \in (\Lambda_p)_t$ is described by continuous-valued parameters. Hence, in order to evaluate the components of the linear system in (3.32), intensities $\tilde{g}(\mathbf{x}_i^k, t+k)$ (Figure 3.2) and there respective spatial derivatives at non grid points $\mathbf{x}_i + \Delta_{(t+k)}\mathbf{p}_i \notin (\Lambda_p)_t$ are needed. Therefore, a spatial interpolation method is needed.

For this task, a C^1 cubic convolution interpolation [25] has been used. Note that by using cubic convolution instead of linear interpolation or nearest-neighbor resampling, the degree of complexity of functions that can be exactly reconstructed is increased. The 2-D cubic convolution interpolation function (used in the motion estimation algorithm) is just a separable extension of the 1-D interpolation function (Figure B.1). It is worth to note here that the order of accuracy of the cubic convolution method, introduced by Keys, is between that of the linear interpolation and that of cubic splines. However, cubic convolution is much more efficient than the method of cubic splines in terms of both storage and computation time.

3.3.4 Estimation over a hierarchy of resolutions

The approximation of the objective function $U(\mathbf{p})$ by a quadratic function of $\mathbf{p}$ in (3.25) is made possible by the use of first-order Taylor expansion. The higher order terms of this expansion are considered to be negligible in the case of small motion parameters and are therefore dropped. However, in the case of fast motion (large displacements), a first-order approximation is not sufficient as these higher order terms cannot be neglected anymore. To deal with the above problem a hierarchical approach is used in the motion estimation algorithm. This hierarchical processing that has been explained in Section 2.4.2 consists of updating the motion fields (velocity and acceleration) at each image resolution level until the full image resolution is reached (Figure 2.4). These updates become smaller at the bottom level of the image pyramid and hence less iterations are needed to reach convergence. Besides of saving computational time (since the number of motion vectors to estimate is divided by four at each lower resolution), the hierarchical method allows to reduce the risks of convergence to a local minimum especially when estimating large motion. This is due to the fact that the image at each resolution is filtered by a Gaussian filter (Figure 2.5) and hence the high frequency content in the data, that is the main reason of forcing the solution to be trapped in a local minimum, is reduced.

3.4 Test images

The motion estimation algorithm presented so far has been tested on some image sequences with synthetic and natural motion. These test images are described next along with the measures that will be used to evaluate the validity of motion estimates.

3.4.1 Image sequences with synthetic motion

In order to test the accuracy of motion estimation, natural sequences with synthetic motion are generated. Each sequence consists of a 45×38 pixel rectangle moving on a still background. The moving rectangle follows a quadratic trajectory at a certain initial velocity $\mathbf{v}_0$ and constant acceleration parameter $\mathbf{a}$. The position of the rectangle at field t is then described as follows:

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0 t + \mathbf{a}t^2, \tag{3.36}$$

where $\mathbf{x}_0$ is its initial position at field 0. The instantaneous velocity $\mathbf{v}(t)$ at field t of any pixel within that rectangle is therefore described by:

$$\mathbf{v}(t) = \mathbf{v}_0 + 2\mathbf{a}t. \tag{3.37}$$

The data in the moving rectangle was obtained from a still image different from that of the background by the following procedure. An image had been first prefiltered by a 2-D low-pass Gaussian FIR filter in order to avoid aliasing after subsampling. The data inside the rectangle was then obtained by a suitable shift of the 4 times subsampled version of the prefiltered image. The subsampling factor of 4 provides a 1/4 pixel precision of the motion parameters. Figure 3.3 illustrates the idea for a moving rectangle of 2×2 pixel. The 2×2 pixels are selected from the 8×8

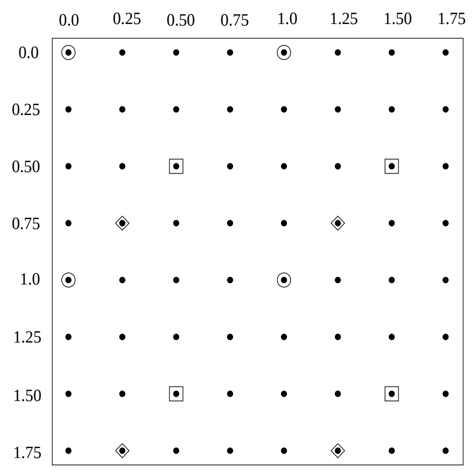


Figure 3.3: Adaptive shifting of a 2×2 pixel rectangle for 1/4 pixel precision.

pixel grid according to the real displacement of the moving rectangle. Hence, for no displacement the pixels surrounded by a circle are selected, for a displacement of $(0.50, 0.50)$ the pixels surrounded by a square are selected, and for a displacement of $(0.25, 0.75)$ the pixels surrounded by a diamond are selected. Due to this sub-pixel accuracy, the matching in the rectangle area is not perfect and hence providing a more realistic testing than pixel accuracy.

In this context, two test images with synthetic motion and different sampling structures have been generated: the test image 1 in Figure 3.4a is an interlaced test sequence, whereas the test image 2 in Figure 3.4b is a progressively sampled test sequence.

In each of these test images the white frame emphasizes the area of 72×64 pixels (the actual size of test image 1 shown in Figure 3.4a is twice larger in the vertical dimension due to its interlaced sampling structure) over which the motion

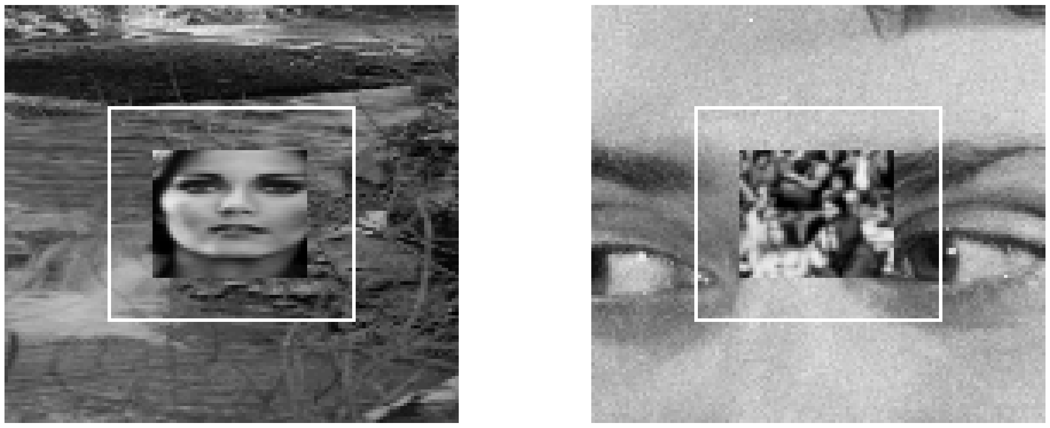

(a) (b)

Figure 3.4: Field #2 of (a) test image 1 (interlaced); field #2 of (b) test image 2 (progressive). The white frames encircle the areas used for estimation.

estimation algorithm will be tested. Each area comprises the 45×38 pixel rectangle which is moving at 1/4 pixel accuracy according to the quadratic trajectory described in (3.36).

The validity of motion estimates within the 72×64 pixel estimation area is verified by the following Mean Square Error (MSE) measure. The MSE measure is expressed as follows:

$$MSE = E\left[(\mathbf{p}(t) - \hat{\mathbf{p}}(t))^2\right] \approx \frac{1}{P_{\mathcal{R}}} \sum_{\mathbf{x}_i \in \mathcal{R}} \left[\mathbf{p}(\mathbf{x}_i, t) - \hat{\mathbf{p}}(\mathbf{x}_i, t)\right]^2, \tag{3.38}$$

where $\mathbf{p}(t)$ and $\hat{\mathbf{p}}(t)$ are the real and estimated motion fields respectively in the region $\mathcal{R}$ of $P_{\mathcal{R}}$ pixel sites. $\mathcal{R}$ can either be the full estimation area ($\mathcal{R}_0$), or the area of the moving rectangle ($\mathcal{R}_1$), or the area $\mathcal{R}_2 = \mathcal{R}_0 - \mathcal{R}_1$. In order to eliminate boundary effects of the moving rectangle, the areas labeled $\mathcal{R}_1'$ and $\mathcal{R}_2'$ will also be used to represent the areas $\mathcal{R}_1$ and $\mathcal{R}_2$ respectively deprived of a narrow strip of 5 pixels that contains the boundaries of the moving rectangle. $\mathcal{R}_1'$, for instance, represents the area inside the moving rectangle.

3.4.2 Natural sequences

The test image 3 is an interlaced 256×212 pixel sequence "femme et arbre" whose field #23 (from a total of 120 fields) is displayed in Figure 3.5.

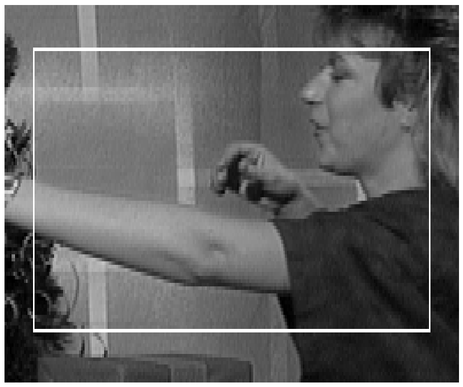


Figure 3.5: Field #23 of test image 3: "femme et arbre". The white frame encircles the area used for estimation.

It contains complex motion, primarily of the hand and the arm. The second natural sequence "Miss America" which is labeled test image 4 (displayed in Figure 3.6) is a typical progressively sampled 360×288 pixel CIF video conference sequence.

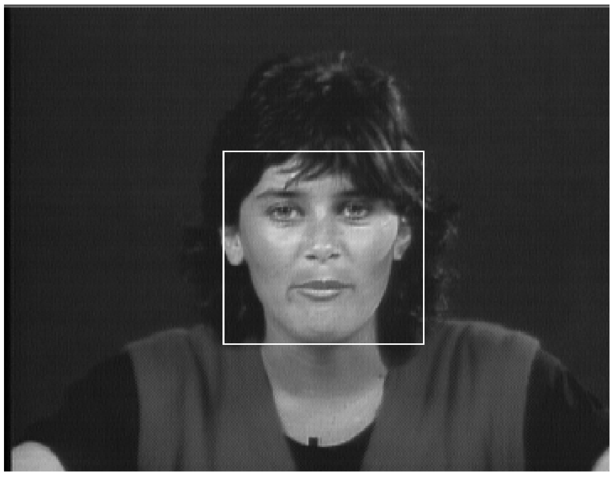


Figure 3.6: Field #16 of test image 4: "Miss America". The white frame encircles the area used for estimation.

These natural sequences have been obtained with a video camera. No filtering or any other processing has been applied to them after their acquisition. Nevertheless, some aliasing is present in the data due to insufficient filtering before sampling.

The validity of motion estimates for these natural sequences can be verified by using these estimates in a motion-compensated interpolation scheme as described next.

3.4.3 Motion-compensated interpolation application

In various video coding schemes temporal subsampling of image sequences is often used to assure high compression ratios needed. Then, at the receiver the missing images are reconstructed *via* motion-compensated interpolation using transmitted motion parameters or motion parameters computed at the receiver from the transmitted images. In Figure 3.7 the first scenario for the case of 4:1 subsampling is presented.

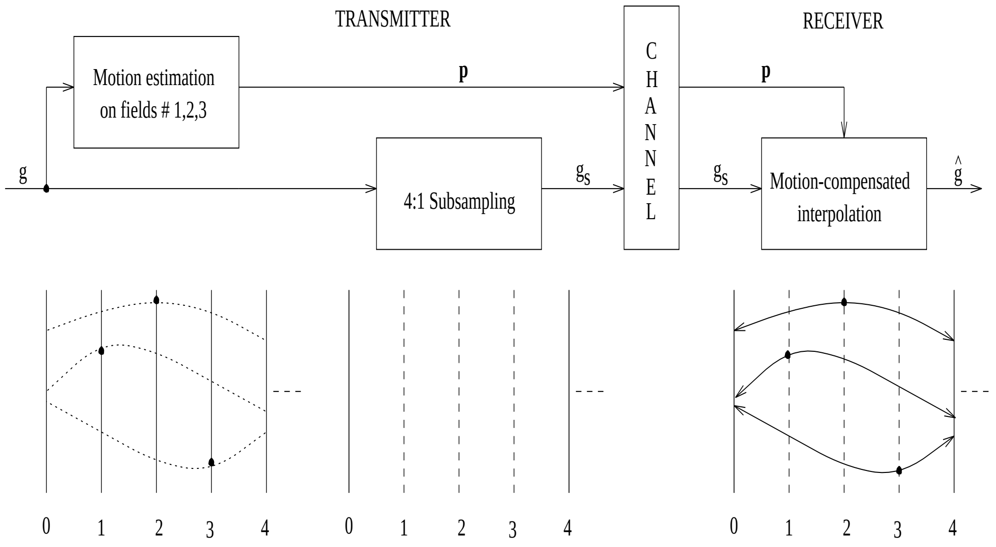


Figure 3.7: Illustration of a typical motion-compensated interpolation scheme used in video coding.

Motion fields $\mathbf{p}_t$ for the images to be omitted at the transmitter (images #1,2,3) are estimated from images #0,1,2,3,4, i.e., $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ for estimation at field #2 (Table 3.1).

Let $\mathcal{J}_t = \{\tau : g_{t+\tau}$ is used in motion-compensated interpolation$\}$. Then, images g_s #0 and 4 are transmitted, and jointly with motion estimates $\hat{\mathbf{p}}_t$ are used to reconstruct images #1,2,3 at the receiver as follows:

$$\hat{g}(\mathbf{x}_i^\tau, t+\tau) = \sum_{k \in \mathcal{J}_t} \gamma_k \tilde{g}(\mathbf{x}_i^k, t+k), \qquad i = 1, \cdots, N_P, \;\; \tau \in \mathcal{K}_t \tag{3.39}$$

where $\sum_{k \in \mathcal{J}_t} \gamma_k = 1$ and $\mathcal{I}_t = \mathcal{J}_t \cup \mathcal{K}_t$.

t	$\mathcal{I}_t$	$\mathcal{J}_t$	$\gamma_j,\ j \in \mathcal{J}_t$
1	{-1,0,1,2,3}	{-1,3}	0.75, 0.25
2	{-2,-1,0,1,2}	{-2,2}	0.50, 0.50
3	{-3,-2,-1,0,1}	{-3,1}	0.25, 0.75

Table 3.1: Configurations of $\mathcal{I}_t$ and $\mathcal{J}_t$ as well as the weights γ for each of the 3 omitted fields for the motion-compensated interpolation scheme described in Figure 3.7.

Since for each $\tau \in \mathcal{K}_t$ we know the original image $g_{t+\tau}$,

$$e_{t+\tau} = \hat{g}_{t+\tau} - g_{t+\tau}, \qquad \tau \in \mathcal{K}_t \tag{3.40}$$

is the reconstructed error at $t+\tau$ that can be used to evaluate the quality of motion estimates $\hat{\mathbf{p}}_{t+\tau}$. Note that in order that $e_{t+\tau}$ be small, the trajectories $\mathbf{c}_{t+\tau}^{\hat{\mathbf{p}}}$ must be close to the true motion trajectories in the image. To describe quantitatively the quality of motion estimates $\hat{\mathbf{p}}_{t+\tau}$, the Peak Signal to Noise Ratio ($PSNR$) is calculated as follows:

$$PSNR = 10 log_{10} \frac{255^2}{var(e_{t+\tau})}, \qquad \tau \in \mathcal{K}_t \tag{3.41}$$

where var denotes variance.

The motion-compensated interpolation scheme described above will be used to illustrate the advantages of using a quadratic trajectory model over a linear one. The estimated quadratic and linear trajectories over $N = 5$ images will be compared. Another scenario for the case of 4:1 subsampling is to estimate linear motion at the receiver using the 2 received images g_s #0 and 4 (i.e., $N = 2$). In this case, the set $\mathcal{I}_t$ will be equal to the set $\mathcal{J}_t$ defined in Table 3.1 for each of the 3 omitted fields. This scenario seems more practical than the linear trajectory model over 5 images since no motion estimates need to be transmitted.

3.5 Selection of parameters

The motion estimation algorithm, discussed in this chapter, is quite flexible from the point of view of a possible choice of parameters. The primary parameters of the algorithm are:

1. The number of resolution levels L in the hierarchical processing.

2. The maximum number of iterations at each resolution level of the image pyramid.

3. The set $\mathcal{I}_t$ of time instants of the test image used during the estimation algorithm, e.g., $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ indicates that the estimation of motion field $\mathbf{p}_t$ at time t is carried out using 5 fields at $t-2$, $t-1$, t, $t+1$, and $t+2$.

4. The choice between linear and quadratic trajectory models. This will be useful in comparing the two approaches by quantifying the gain/loss achieved in a motion-compensated interpolation application. Note that in all simulations, the quadratic trajectory model is used by default unless otherwise stated.

5. The regularization parameter λ_p which plays a crucial role in weighting the importance between the smoothness term and the structural term.

6. The matrix $\mathbf{\Gamma}$ that permits different weighing of individual motion parameters.

The maximum number of iterations at each image resolution level was set to 50. However, the algorithm was allowed to stop or switch to the next higher resolution level if the condition expressed by (3.35) is satisfied for a sufficiently small ϵ. When the algorithm stops, the motion field $\mathbf{p}$ of the best converged energy $U^n(\mathbf{p})$ at iteration n selected among the last 10 iterations is stored as a final result.

On the other hand, the diagonal elements of the weight matrix in (3.20) were chosen as follows: $[w_{v_x} \quad w_{v_y} \quad w_{a_x} \quad w_{a_y}] = [1 \quad 1 \quad 2 \quad 2]$. In this way, more weight is given to the acceleration parameters in the smoothness term (3.19), and hence more smoothness is enforced on the acceleration field than on the velocity field. This seems reasonable especially if more than 2 fields (i.e., $N > 2$) are used in the estimation since a small deviation in the estimated acceleration causes a significant deviation of the two end points of the estimated trajectory (dependence on $(\tau - t)^2$ in (3.6)). In the following sections the selection of the rest of the parameters is discussed.

3.5.1 Selection of L

The number of resolution levels L has been varied from 1 to 4 in the estimation algorithm applied to field #2 of test image 2. A quadratic motion trajectory at pixel accuracy has been selected with the actual motion parameters $\mathbf{p}(2) = [1 \quad 2 \quad 1 \quad 1]^T$ at field #2. Note that the velocity parameters have been calculated directly from equation (3.37). The behavior of the objective function $U^n(\mathbf{p})$ as a function of the iteration number n for each case is shown in Figure 3.8 for $\lambda_p = 30$, $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$, and $\mathbf{\Gamma} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

The discontinuities in Figures 3.8b, 3.8c, and 3.8d correspond to switching of the estimation algorithm to the next higher resolution level which is determined when the inequality in (3.35) is satisfied for certain ϵ. It is worth to notice here the advantage of using a multiresolution approach: the rate of convergence to a minimum is improved. Hence, without a multiresolution approach, i.e., $L = 1$, the algorithm failed to converge to the optimum, and instead converged to a local one at $U^n(\mathbf{p}) = 165$ (Figure 3.8a). However, for $L > 1$, the optimum $U^n(\mathbf{p}) = 20$ has been detected successfully with a lower computational burden. Thus, for $L = 2$ the global optimum is detected at $n \approx 60$ (Figure 3.8b), for $L = 3$ the global optimum is detected at $n \approx 50$ (Figure 3.8c), and for $L = 4$ the global optimum is detected at

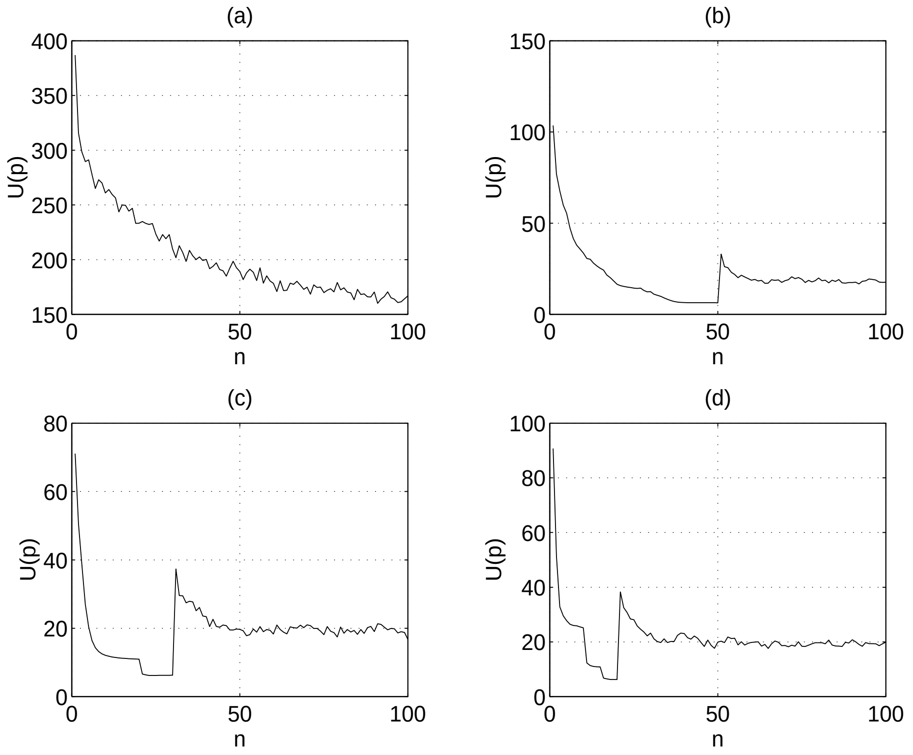
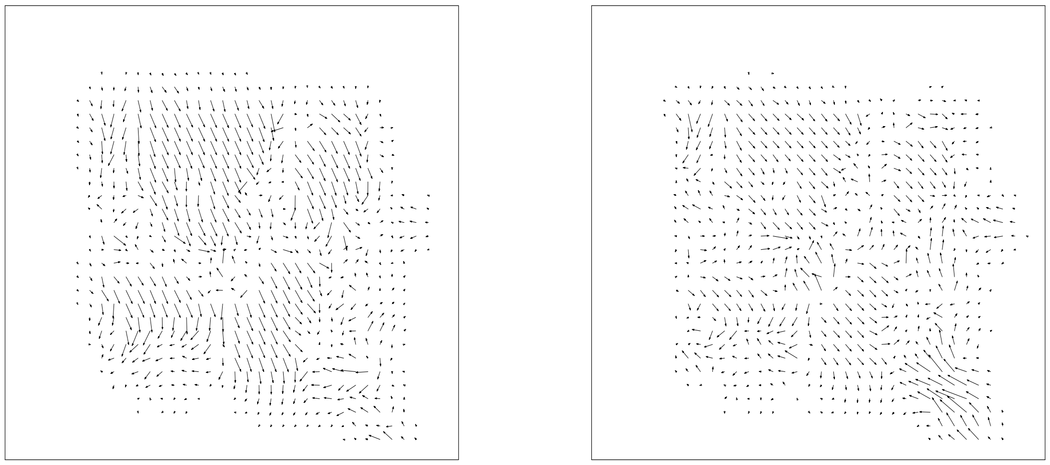


Figure 3.8: Behavior of $U^n(\mathbf{p})$ as a function of the iteration number n for field #2 of test image 2 with $\lambda_p = 30$, $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$, and $\mathbf{\Gamma} = \mathbf{I}$, for (a) $L = 1$; (b) $L = 2$; (c) $L = 3$; (d) $L = 4$.

$n \approx 40$ (Figure 3.8d). The estimated motion fields $\hat{\mathbf{p}}(2)$ for $L = 1$, and $L = 4$ are shown in Figures 3.9 and 3.10, respectively.

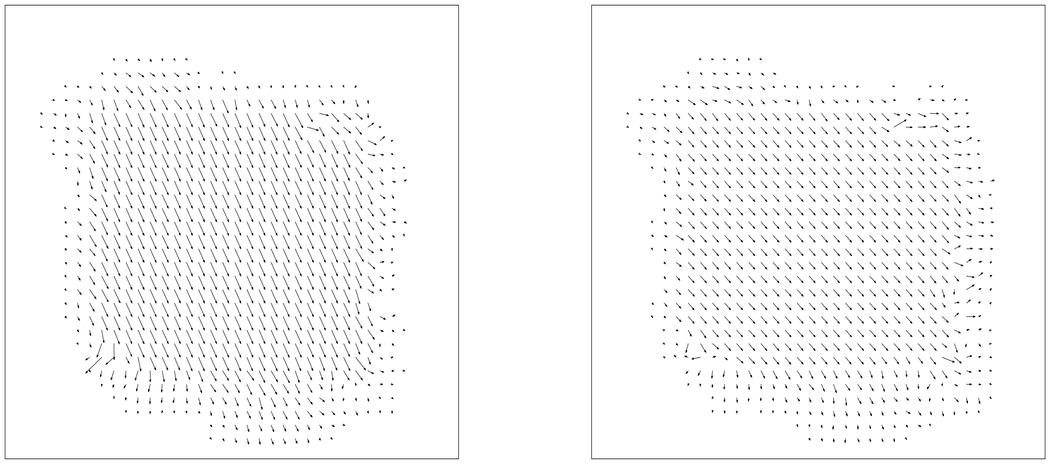
The estimated motion parameters for $L = 1$ in Figure 3.9 correspond to a local minimum, that is why the estimation inside the moving rectangle is far from the true motion parameters. For $L = 4$ the estimate in Figure 3.10 is consistent with the true motion except at the boundaries of the moving rectangle where the motion of the rectangle has smeared outside the boundaries due to the desired smoothness property of motion introduced by the smoothness term. Hence $L = 4$ will be used in the estimation algorithm for the rest of the simulations in this chapter.



(a

(b)

Figure 3.9: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ for $L = 1$ and field #2 of test image 2 with $\mathbf{p}(2) = [1 \;\; 2 \;\; 1 \;\; 1]^T$.



(a)

(b)

Figure 3.10: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ for $L = 4$ and field #2 of test image 2 with $\mathbf{p}(2) = [1 \;\; 2 \;\; 1 \;\; 1]^T$.

3.5.2 Selection of λ_p

The test image 2 has been used with the same set of synthetic motion parameters but now the regularization parameter λ_p is varied from 10 to 200 with a step of 10. The results are shown in Figure 3.11.

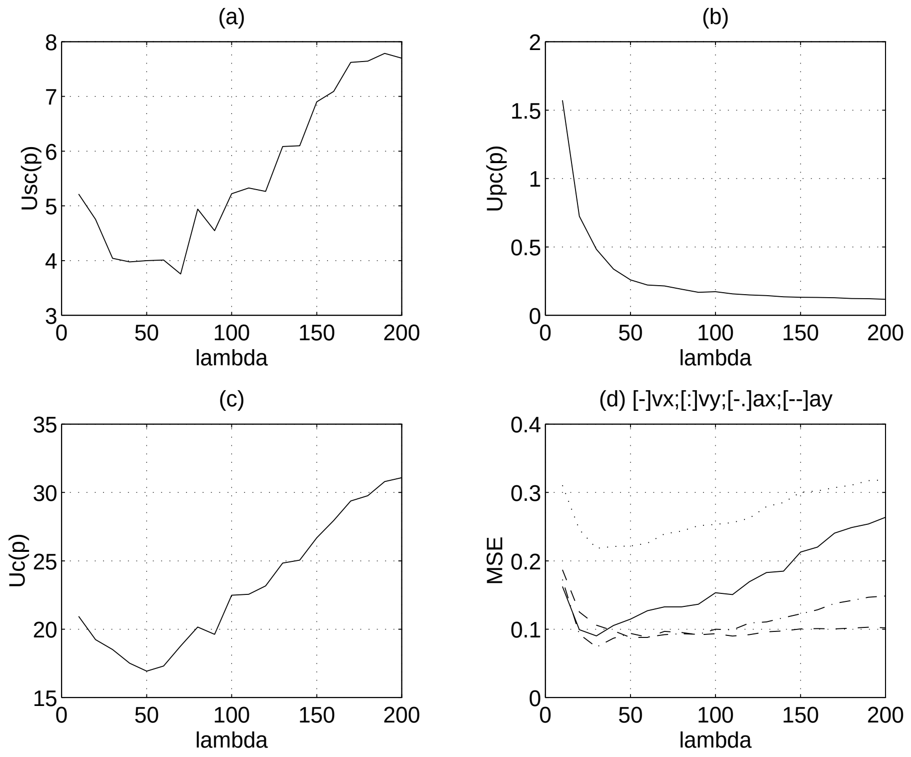
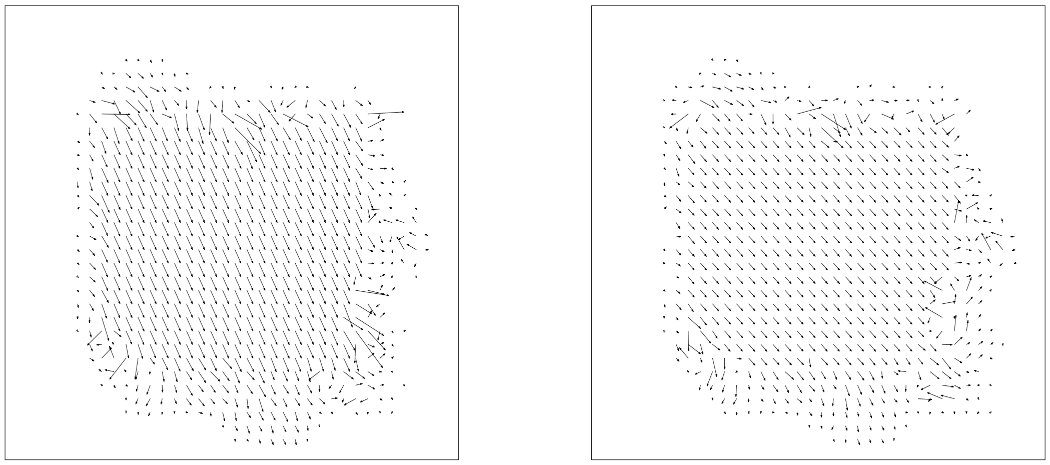


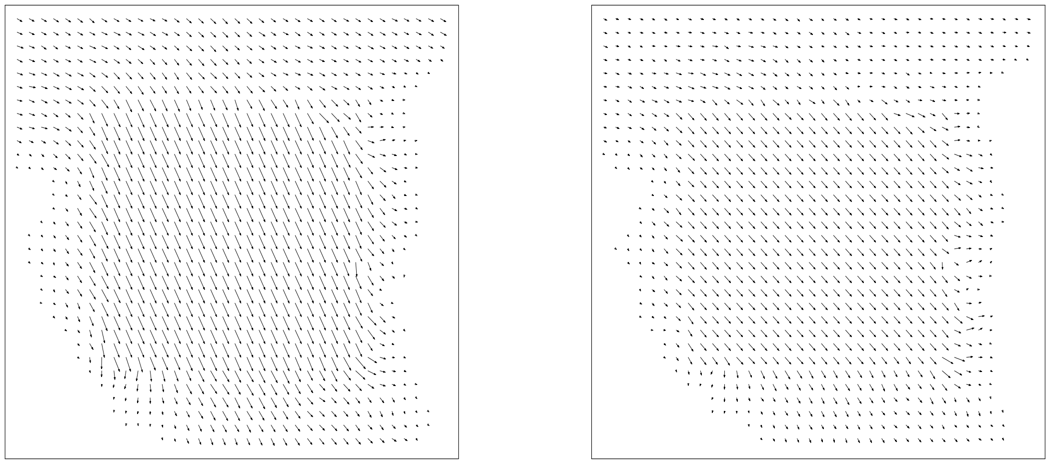
Figure 3.11: Energies: (a) $U_{sc}(\mathbf{p})$; (b) $U_{pc}(\mathbf{p})$; (c) $U_c(\mathbf{p})$ after convergence and (d) the MSE of motion estimates in $\mathcal{R}_0$ as a function of λ_p for field #2 of test image 2 with $\mathbf{p}(2) = [1 \;\; 2 \;\; 1 \;\; 1]^T$, $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$, and $\mathbf{\Gamma} = \mathbf{I}$.

Figures 3.11a, 3.11b, and 3.11c illustrate the behavior of the following energies: $U_{sc}(\mathbf{p})$, $U_{pc}(\mathbf{p})$, $U_c(\mathbf{p}) = U_{sc}(\mathbf{p}) + \lambda_p U_{pc}(\mathbf{p})$, respectively, as functions of λ_p (the additional subscript 'c' is used to denote converged energy). Figure 3.11d, on the other hand, illustrates the behavior of the MSE as a function of λ_p for each of the 4 motion parameters estimated over $\mathcal{R}_0$. Note that the graphs of Figures 3.11c, and 3.11d both show a minimum around $\lambda_p = 30$. The estimated motion fields for $\lambda_p = 10$ and $\lambda_p = 200$ are shown in Figures 3.12 and 3.13 respectively.



(a) (b)

Figure 3.12: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ for $\lambda_p = 10$ and field #2 of test image 2 with $\mathbf{p}(2) = [1 \;\; 2 \;\; 1 \;\; 1]^T$.



(a) (b)

Figure 3.13: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ for $\lambda_p = 200$ and field #2 of test image 2 with $\mathbf{p}(2) = [1 \;\; 2 \;\; 1 \;\; 1]^T$.

For $\lambda_p = 10$, the smoothing at the boundaries of the moving rectangle is not sufficient causing the estimates at the boundaries to deviate noticeably from the true motion vectors. On the other hand, for $\lambda_p = 200$ the high importance of the smoothness term has caused the estimates within the moving rectangle to propagate far away from its boundaries (Figure 3.13). This illustrates the convex property of the

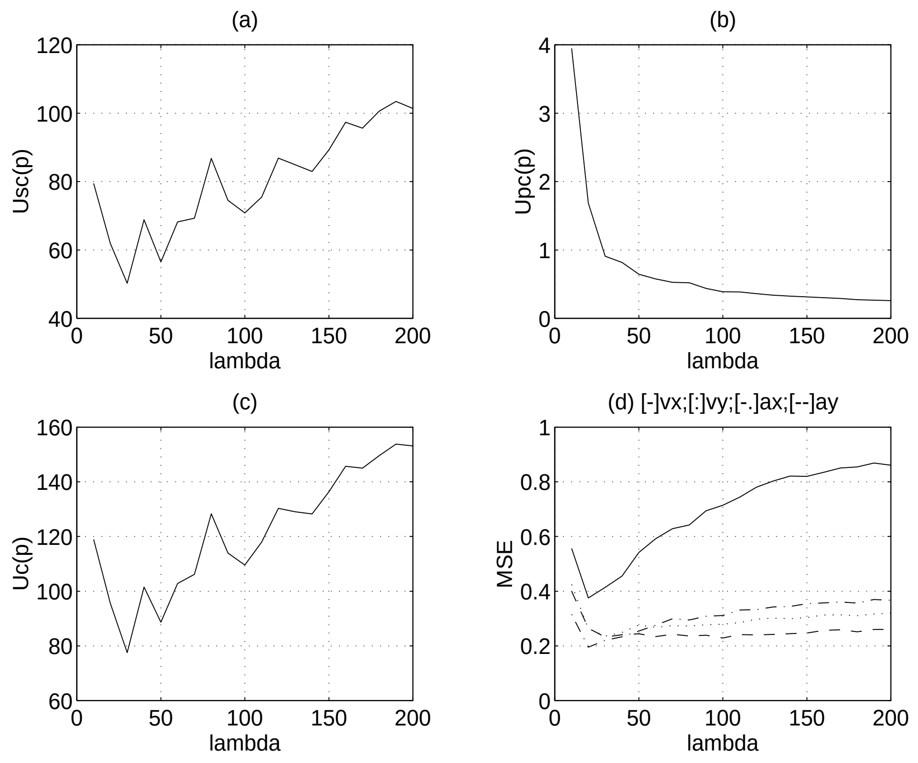


Figure 3.14: Energies: (a) $U_{sc}(\mathbf{p})$; (b) $U_{pc}(\mathbf{p})$; (c) $U_c(\mathbf{p})$ after convergence and (d) the MSE of motion estimates in $\mathcal{R}_0$ as a function of λ_p for field #2 of test image 1 with $\mathbf{p}(2) = [1.75 \;\; 1.5 \;\; 1 \;\; 1.5]^T$, $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$, and $\mathbf{\Gamma} = \mathbf{I}$.

MSE curves drawn in Figure 3.11d. A value of $\lambda_p = 30$ (Figure 3.10) seems to give a reasonable trade-off between under- and over-smoothing, and also coincides with the minimum of $U_c(\mathbf{p})$ (Figure 3.11c). The same test has been performed for test image 1 but with sub-pixel accuracy of motion parameters $\mathbf{p}(2) = [1.75 \;\; 1.5 \;\; 1 \;\; 1.5]^T$. The same kind of behavior can be noticed in the results shown in Figure 3.14.

The selection of λ_p on natural sequences has been performed on field #30 of test image 3 with $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ and $\mathbf{\Gamma} = \mathbf{I}$. Two evaluation measures are plotted as a function of λ_p in Figure 3.15.

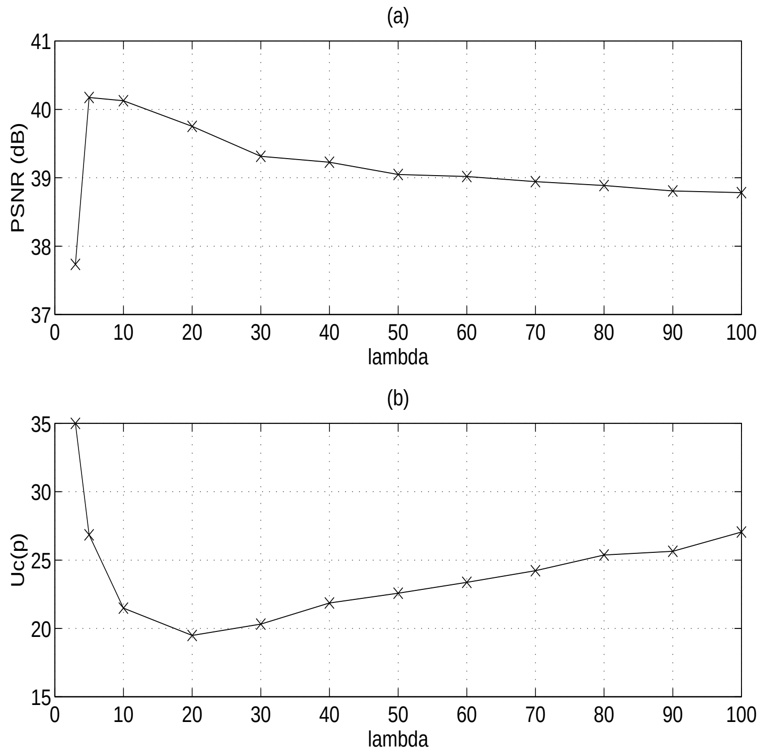


Figure 3.15: Behavior of the $PSNR$ (a) and the converged energy $U_c(\mathbf{p})$ (b) as a function of λ_p resulting from motion estimation performed on field #30 of test image 3 with $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$, and $\mathcal{J}_t = \{-1, 1\}$.

The first one (Figure 3.15a) is the $PSNR$ in dB of the reconstructed field #30 using interpolation. The reconstruction is done using the estimated motion vectors and a simple bilinear interpolation of the motion-compensated pixels between the previous (#29) and the next (#31) fields. Hence, the case $\mathcal{J}_t = \{-1, 1\}$ is used here. The second measure (Figure 3.15b) represents the best converged value $U_c(\mathbf{p})$ of the objective function $U(\mathbf{p})$ during the estimation algorithm. $\lambda_p = 20$ seems to result in a good $PSNR$ and the best converged energy. The estimated motion fields at $\lambda_p = 20$ are shown in Figure 3.16. These fields illustrate the motion of the hand that is moving downward with some deceleration (note the acceleration vectors on the hand that are

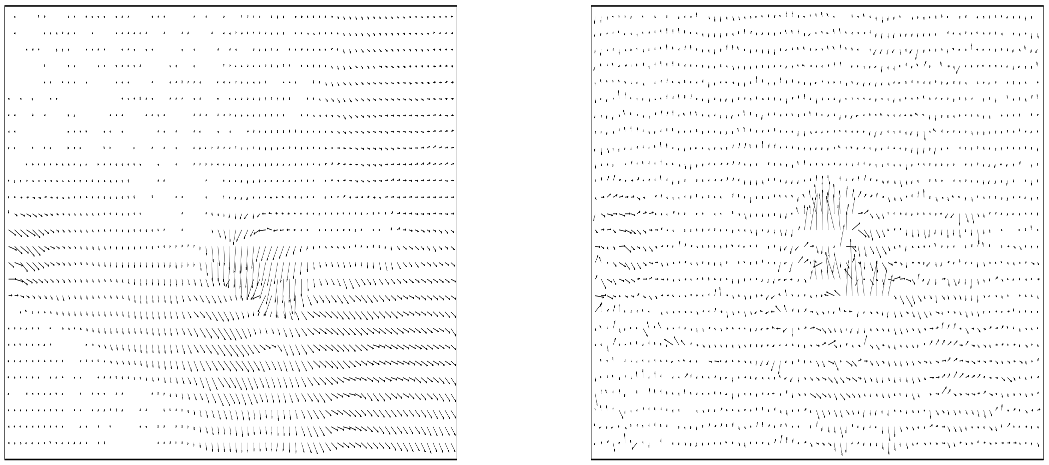

(a) (b)

Figure 3.16: Estimated: (a) velocity $\hat{\mathbf{v}}(30)$, and (b) acceleration $\hat{\mathbf{a}}$ (scaled by 4) for $\lambda_p = 20$ and field #30 of test image 3.

pointing upward).

To conclude, one can say that it is difficult to choose a fixed value of λ_p for all the sequences. However, a range of $20 < \lambda_p < 30$ seems to yield to best compromise for the tests that were performed on test images 1, 2, and 3. The value of λ_p has been divided by 2 between a higher and a lower resolution level. This is due to the fact that the distance between two consecutive motion vectors at the next lower resolution level is multiplied by two and hence the contribution of the smoothness should be reduced.

3.5.3 Selection of $\mathcal{I}_t$

The number of fields $N = Card(\mathcal{I}_t)$ used in the estimation algorithm has been varied by selecting different configurations of the set $\mathcal{I}_t$. This was done for $\lambda_p = 25$, and field #2 of test image 1 with the rectangle moving at subpixel accuracy along a quadratic trajectory characterized by the set $\mathbf{p}(2) = [1.5 \;\; 1.5 \;\; 0.5 \;\; 1]^T$.

The MSE of motion estimates inside the moving rectangle (i.e., region $\mathcal{R}'_1$) for each case is shown in Table 3.2. Note that the overall MSE measure (of the 4

$\mathcal{I}_t$	$v_x(2)$	$v_y(2)$	a_x	a_y
$\{-1,1\}$	0.035082	0.054667	0.249625	0.838163
$\{-1,0,1\}$	0.056641	0.068800	0.010226	0.910025
$\{-2,-1,0,1\}$	0.081383	0.146087	0.025569	0.073454
$\{-1,0,1,2\}$	0.057416	0.048389	0.043047	0.055020
$\{-2,-1,0,1,2\}$	0.045844	0.028816	0.027701	0.034414
$\{-2,-1,0,1,2,3\}$	0.040133	0.029073	0.020734	0.020564

Table 3.2: MSE of motion estimates in region $\mathcal{R}_1'$ at field #2 of test image 1 with $\mathbf{p}(2) = [1.5 \;\; 1.5 \;\; 0.5 \;\; 1.0]^T$ for different sets $\mathcal{I}_t$, and $\lambda_p = 25$.

parameters) decreases when N increases.

This is illustrated in Figure 3.17 that displays the real and estimated trajectories at position $\mathbf{x}(2) = [37 \;\; 30]^T$ (a point inside the moving rectangle chosen at random) for field #2 of test image 1 for each of the 6 combinations of the set $\mathcal{I}_t$.

The estimated and real trajectories are drawn in dotted and full lines respectively around the selected motion site $\mathbf{x}(2)$ at field #2 indicated by a 'o'. The positions indicated by an 'x' represent the tracked (on the dotted line) or real (on the full line) positions of pixel $\mathbf{x}(2)$ at fields #0,1,3,4,5 of test image 1.

It is clear from Figure 3.17 that a better tracking of the real trajectory is achieved for larger temporal support N which seems to be reasonable. However, for much larger temporal support (i.e., $N \geq 5$), occluded areas begin to play a role in the estimation algorithm, especially on natural TV sequences. Since occlusions are not considered in this chapter, a choice of $\mathcal{I}_t = \{-2,-1,0,1,2\}$ i.e., $N = 5$ seems to be reasonable (Figure 3.17e) and will be used for the rest of the simulations in this chapter.

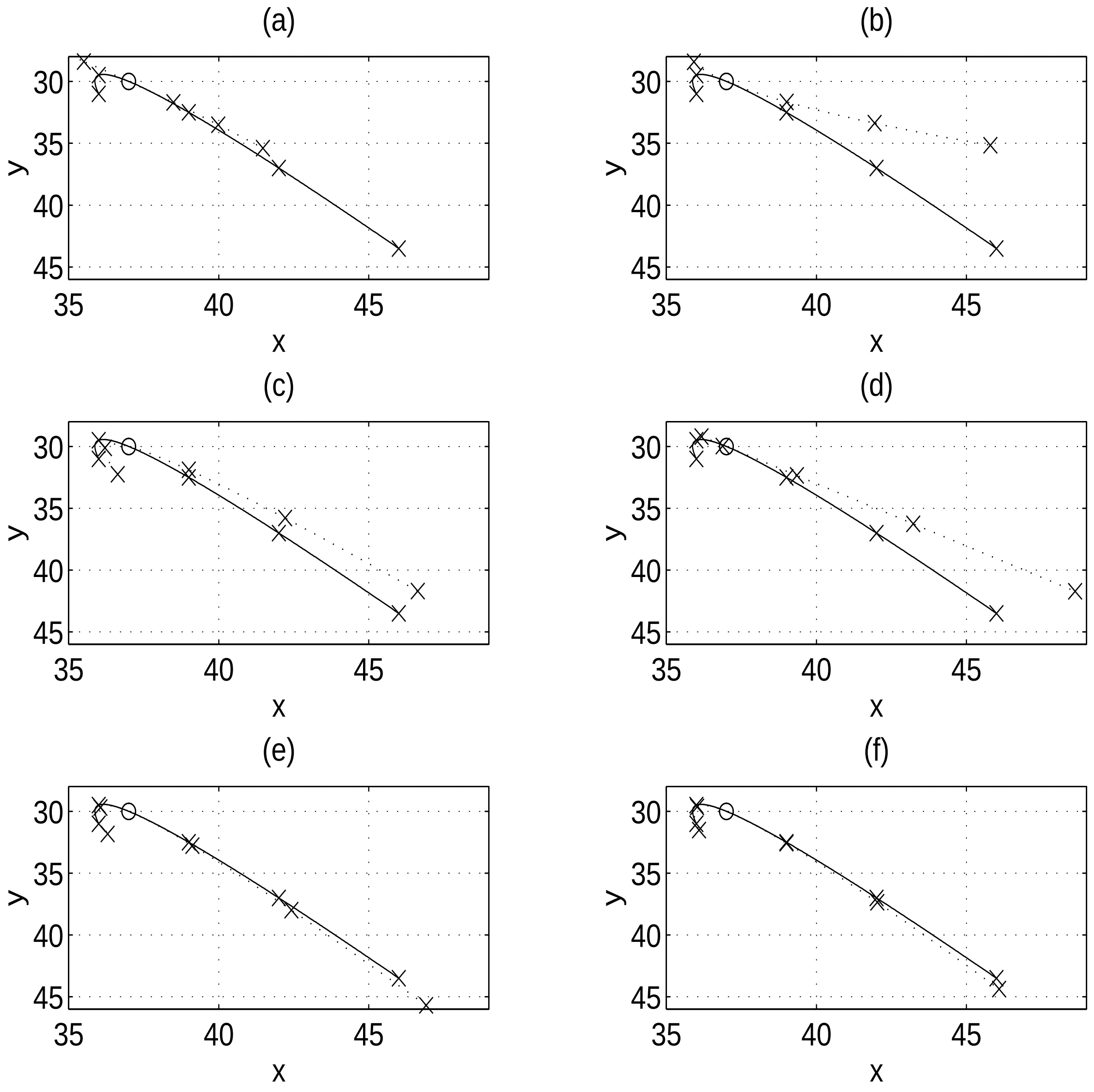


Figure 3.17: Estimated (dotted line) and real trajectories (full line) at $\mathbf{x}(2) = [37 \quad 30]^T$ for field #2 of test image 1 with $\mathbf{p}(2) = [1.5 \quad 1.5 \quad 0.5 \quad 1.0 \quad]^T$ for (a) $\mathcal{I}_t = \{-1, 1\}$; (b) $\mathcal{I}_t = \{-1, 0, 1\}$; (c) $\mathcal{I}_t = \{-2, -1, 0, 1\}$; (d) $\mathcal{I}_t = \{-1, 0, 1, 2\}$; (e) $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$; (f) $\mathcal{I}_t = \{-2, -1, 0, 1, 2, 3\}$.

3.6 Simulation results

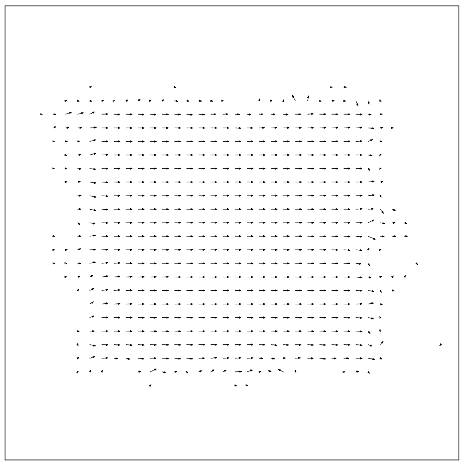
3.6.1 Results for synthetic sequences

The test image 1 has been generated using 4 different sets of synthetic motion parameters $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$, where $\mathbf{p}_1$ and $\mathbf{p}_2$ are motion fields generated at pixel accuracy, while $\mathbf{p}_3$ and $\mathbf{p}_4$ are motion fields generated at sub-pixel accuracy.

	$v_x(2)$	$v_y(2)$	a_x	a_y
$\mathbf{p}_1(2)$	1.000000	0.000000	0.000000	1.000000
MSE in $\mathcal{R}_0$	0.036494	0.005016	0.014221	0.044516
MSE in $\mathcal{R}_1$	0.006976	0.001729	0.001633	0.025963
MSE in $\mathcal{R}'_1$	0.002572	0.000982	0.001027	0.018506
MSE in $\mathcal{R}_2$	0.053912	0.006955	0.021648	0.055463
MSE in $\mathcal{R}'_2$	0.003176	0.000346	0.015453	0.029455
$\mathbf{p}_2(2)$	2.000000	2.000000	1.000000	1.000000
MSE in $\mathcal{R}_0$	0.316888	0.217085	0.114484	0.091598
MSE in $\mathcal{R}'_1$	0.008364	0.020709	0.003950	0.003291
$\mathbf{p}_3(2)$	1.500000	1.500000	0.500000	1.000000
MSE in $\mathcal{R}_0$	0.163142	0.103574	0.057372	0.073342
MSE in $\mathcal{R}'_1$	0.045844	0.028816	0.027701	0.034414
$\mathbf{p}_4(2)$	1.250000	1.750000	0.250000	0.750000
MSE in $\mathcal{R}_0$	0.173114	0.173606	0.069072	0.082292
MSE in $\mathcal{R}'_1$	0.096116	0.054574	0.081085	0.063015

Table 3.3: MSE of motion estimates in various regions at field #2 of test image 1 for 4 different sets of synthetic motion parameters.

In each case the motion fields at field #2 are estimated with $L = 4$, $\lambda_p = 25$, $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$, and $[w_{v_x} \;\; w_{v_y} \;\; w_{a_x} \;\; w_{a_y}] = [1 \;\; 1 \;\; 2 \;\; 2]$. The MSE in $\mathcal{R}_0$ & $\mathcal{R}'_1$ along with the true velocity and acceleration parameters at field #2 for each case are


(a)

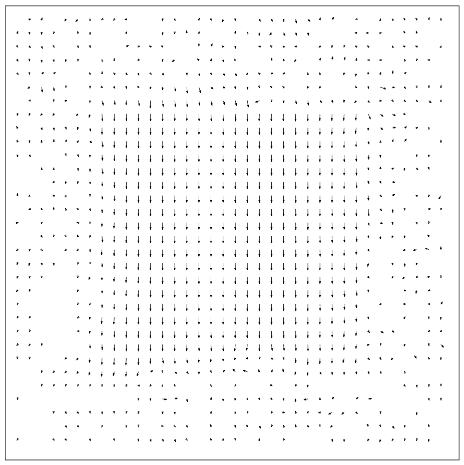
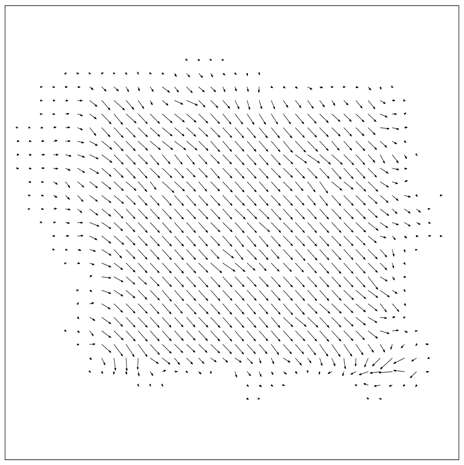

(b)

Figure 3.18: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ for field #2 of test image 1 with $\mathbf{p}(2) = \mathbf{p}_1(2)$ (due to interlacing, the actual distance between two motion sites in the vertical direction is twice larger than it appears).


(a)

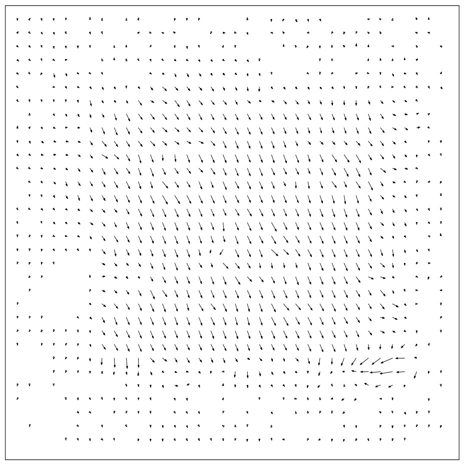

(b)

Figure 3.19: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ for field #2 of test image 1 with $\mathbf{p}(2) = \mathbf{p}_3(2)$.

shown in Table 3.3.

The estimated motion fields for the sets $\mathbf{p}_1$ and $\mathbf{p}_3$ of synthetic motion parameters are shown in Figures 3.18 and 3.19, respectively. Note that the estimated motion and acceleration fields correspond well to the motion of the rectangle (the smoothing at boundaries is due to lack of a motion boundary model). This is confirmed by the MSE in $\mathcal{R}'_1$ (no boundary effects) that is in general much smaller than the error in $\mathcal{R}_0$ (Table 3.3).

	$v_x(2)$	$v_y(2)$	a_x	a_y
$\mathbf{p}_5(2)$	2.000000	1.000000	0.000000	0.000000
MSE in $\mathcal{R}_0$	0.229595	0.068679	0.012156	0.009899
MSE in $\mathcal{R}'_1$	0.005424	0.007288	0.000961	0.003390
$\mathbf{p}_6(2)$	1.000000	2.000000	1.000000	1.000000
MSE in $\mathcal{R}_0$	0.126479	0.250304	0.100246	0.096698
MSE in $\mathcal{R}'_1$	0.001685	0.008511	0.001823	0.002857
$\mathbf{p}_7(2)$	1.750000	1.500000	1.000000	1.500000
MSE in $\mathcal{R}_0$	0.690936	0.279143	0.312324	0.275153
MSE in $\mathcal{R}'_1$	0.106735	0.069062	0.057062	0.036957
$\mathbf{p}_8(2)$	1.750000	2.250000	0.750000	1.250000
MSE in $\mathcal{R}_0$	0.585331	0.613616	0.227904	0.268670
MSE in $\mathcal{R}'_1$	0.145411	0.231920	0.120366	0.188574

Table 3.4: MSE of motion estimates in regions $\mathcal{R}_0$ & $\mathcal{R}'_1$ at field #2 of test image 2 for 4 different sets of synthetic motion parameters.

The MSE in various regions of the motion field estimate for the $\mathbf{p}_1$ set of synthetic motion parameters is also shown in Table 3.3. Note that the small values of the MSE in regions $\mathcal{R}'_1$ and $\mathcal{R}'_2$ (no boundary effect) demonstrate well the ability of the motion estimation algorithm to detect accurately the quadratic trajectory specified by motion parameters at pixel accuracy.

The same simulation has been carried out for test image 2 using another set of synthetic motion parameters $\mathbf{p}_5$, $\mathbf{p}_6$, $\mathbf{p}_7$, and $\mathbf{p}_8$, where $\mathbf{p}_5$ and $\mathbf{p}_6$ are motion fields generated at pixel accuracy, while $\mathbf{p}_7$ and $\mathbf{p}_8$ are motion fields generated at sub-pixel accuracy. The MSE results in $\mathcal{R}_0$ and $\mathcal{R}_1'$ for each case are shown in Table 3.4.

To illustrate how well the motion parameters with sub-pixel accuracy are detected, a comparison between the estimated and real trajectories at a certain point inside the moving rectangle is considered.

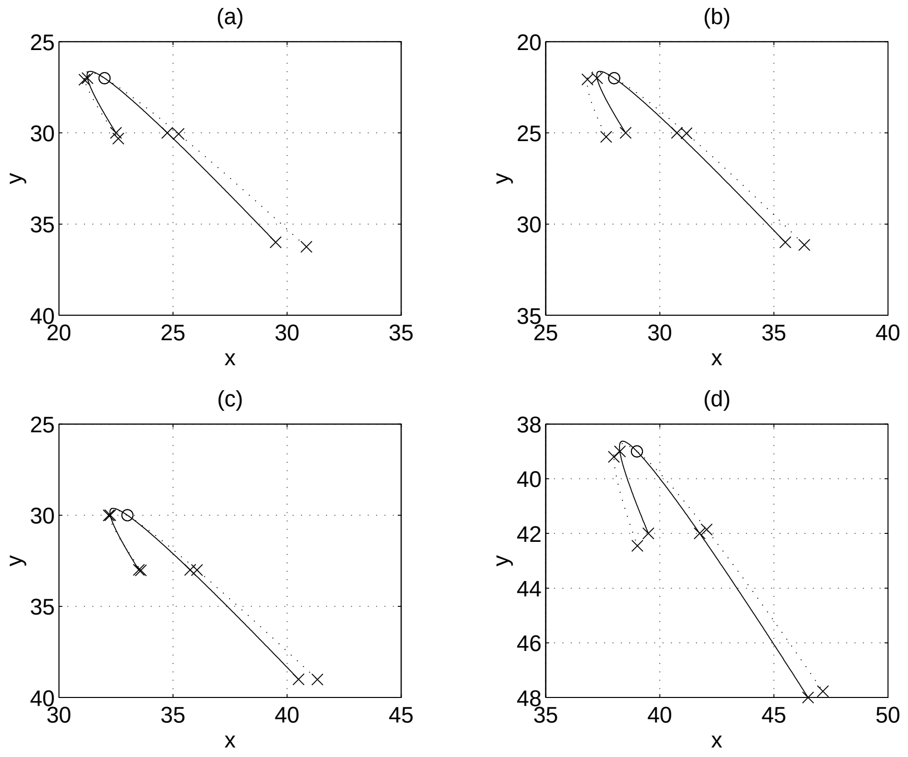


Figure 3.20: Estimated trajectories at: (a) $\mathbf{x}(2) = [22 \ \ 27]^T$; (b) $\mathbf{x}(2) = [28 \ \ 22]^T$; (c) $\mathbf{x}(2) = [33 \ \ 30]^T$; (d) $\mathbf{x}(2) = [39 \ \ 39]^T$ inside the moving rectangle of test image 2 with $\mathbf{p}(2) = \mathbf{p}_7(2) = [1.75 \ \ 1.5 \ \ 1 \ \ 1.5]^T$, and $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$.

Figure 3.20 illustrates the estimated trajectories at 4 different positions (chosen at random) inside the moving rectangle of test image 2 for the set $\mathbf{p}_7$. Note that the deviation from the real trajectory is more noticeable at the furthest fields, i.e., fields #0 and 5, however, in general, the quadratic trajectory is very well followed.

This trajectory representation is useful to illustrate the difference in tracking between linear and quadratic motion trajectory models. The estimation was performed using both models on test image 1 with the set $\mathbf{p}_3$ of synthetic motion parameters generated at sub-pixel accuracy.

The resulting velocity field estimates $\mathbf{v}(2)$ obtained using the linear and quadratic motion trajectory models are compared in Figure 3.21. The discrepancy between the two models can be seen in Table 3.5 that shows the calculated MSE in $\mathcal{R}_0$ and $\mathcal{R}_1'$ for the two considered models.

	Model	$v_x(2)$	$v_y(2)$	a_x	a_y
MSE in $\mathcal{R}_0$	quadratic	0.163142	0.103574	0.057372	0.073342
	linear	0.811878	0.512640	-	-
MSE in $\mathcal{R}_1'$	quadratic	0.045844	0.028816	0.027701	0.034414
	linear	2.230566	0.932370	-	-

Table 3.5: Comparison of the MSE of motion estimates in $\mathcal{R}_0$ and $\mathcal{R}_1'$ resulting from quadratic and linear trajectory models at field #2 of test image 1 with $\mathbf{p}(2) = \mathbf{p}_3(2) = [1.5 \;\; 1.5 \;\; 0.5 \;\; 1]^T$, and $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$.

From Figure 3.21, and Table 3.5 one can deduce that the use of a linear trajectory model is not enough to track the real instantaneous velocities on a quadratic trajectory, and hence the use of the quadratic trajectory model is advantageous in comparison with the linear one.

This difference between the two models is also illustrated graphically in Figure 3.22 in which the real trajectory at a certain point $(\mathbf{x}, 2)$ (chosen at random inside the moving rectangle) is drawn in full line. Trajectory resulting from the use of the quadratic motion trajectory model is shown in dotted line, and trajectory resulting from the use of the linear model is shown in dashed line. It is worth to note how well the real trajectory is represented by using the quadratic trajectory model. However, in the case of the linear model, the algorithm failed to track the real trajectory because of the restricted degree of freedom of this model.

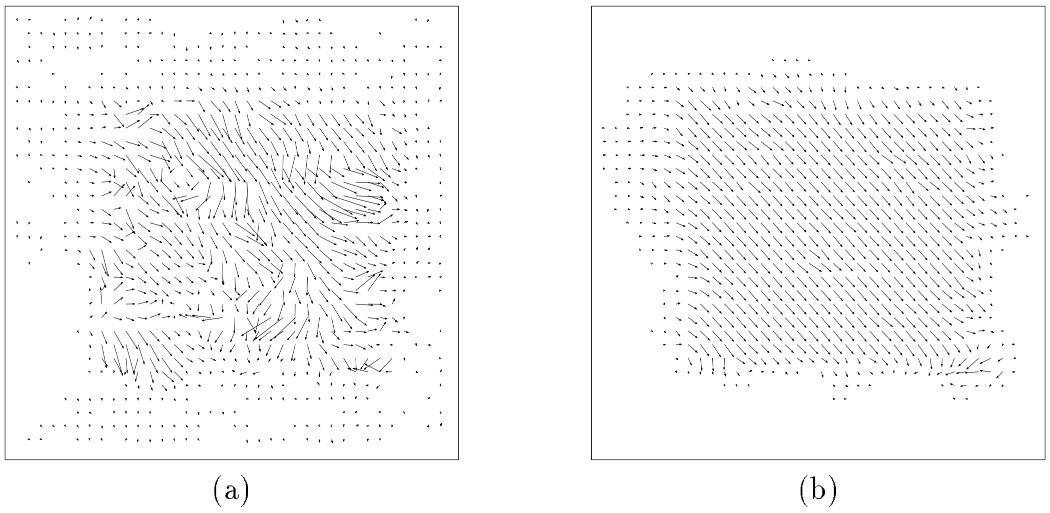


Figure 3.21: Estimated velocity $\hat{\mathbf{v}}(2)$ at field #2 of test image 1 with $\mathbf{p}(2) = \mathbf{p}_3(2)$ using (a) linear and (b) quadratic motion trajectory models.

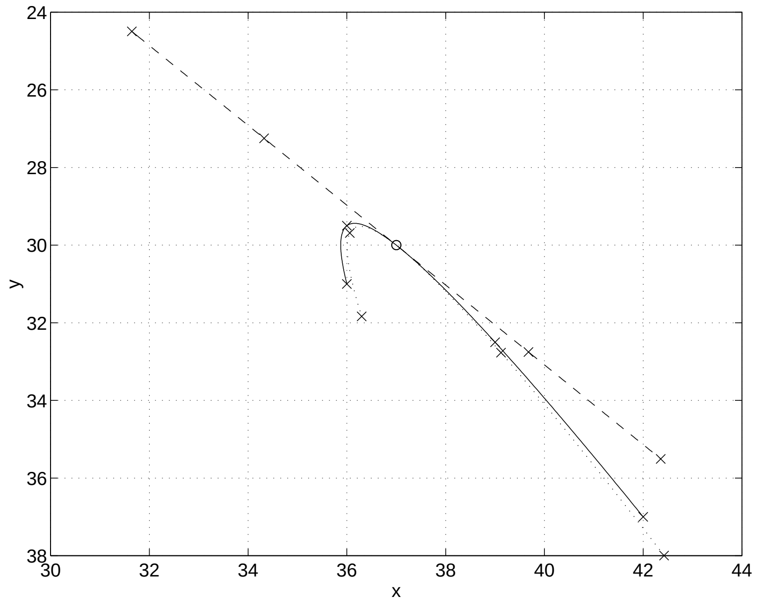


Figure 3.22: Estimated linear (dashed line) and quadratic (dotted line) trajectories at the point $\mathbf{x}(2) = [37 \quad 30]^T$ inside the moving rectangle of test image 1 with $\mathbf{p}(2) = \mathbf{p}_3(2) = [1.5 \;\; 1.5 \;\; 0.5 \;\; 1]^T$, and $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$.

3.6.2 Results for natural sequences

In order to simulate the motion-compensated interpolation application described in Figure 3.7, the estimation algorithm was applied to the first 33 fields of test image 3 (over the estimation area defined in Figure 3.5) with $L = 4$, $\lambda_p = 20$, and the sets $\mathcal{I}_t$ and $\mathcal{J}_t$ defined as in Table 3.1. This experiment was performed three times: using first a linear trajectory model with $N = 5$, then a quadratic trajectory model with $N = 5$, and finally a linear trajectory model with $N = 2$. In the last case the set $\mathcal{I}_t$ is equal to the set $\mathcal{J}_t$, as described in Section 3.4.3. The $PSNR$ curves for all the processed fields are plotted in Figure 3.23.

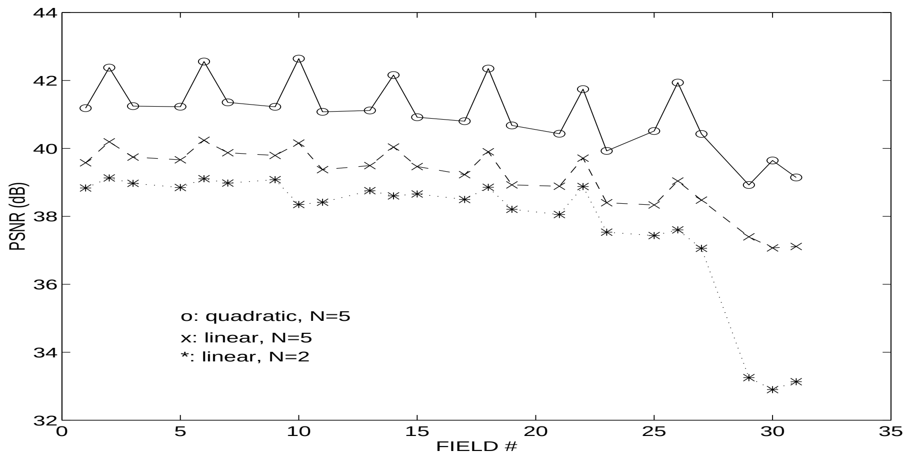


Figure 3.23: Comparison of the $PSNR$ for reconstructed fields from test image 3 using linear (dashed line) and quadratic (full line) motion trajectory models with $N = 5$ and linear trajectory model with $N = 2$ (dotted line).

The resulting means for $PSNR$ ($\overline{PSNR}$) obtained from the estimation algorithm are shown in Table 3.6 for the linear and quadratic trajectory models.

The use of the quadratic trajectory model resulted in an average gain of +1.89 dB with respect to the linear trajectory model with $N = 5$, and +3.27 dB with respect to the linear trajectory model with $N = 2$. These gains are due to better tracking of the real motion trajectories that causes the structural term $U_s(\mathbf{p})$ in (3.17) to decrease substantially.

Model	$\overline{PSNR}$
linear ($N = 2$)	37.79
linear ($N = 5$)	39.17
quadratic ($N = 5$)	41.06

Table 3.6: Means for $PSNR$ evaluated over 24 fields of test image 3 using linear and quadratic motion trajectory models.

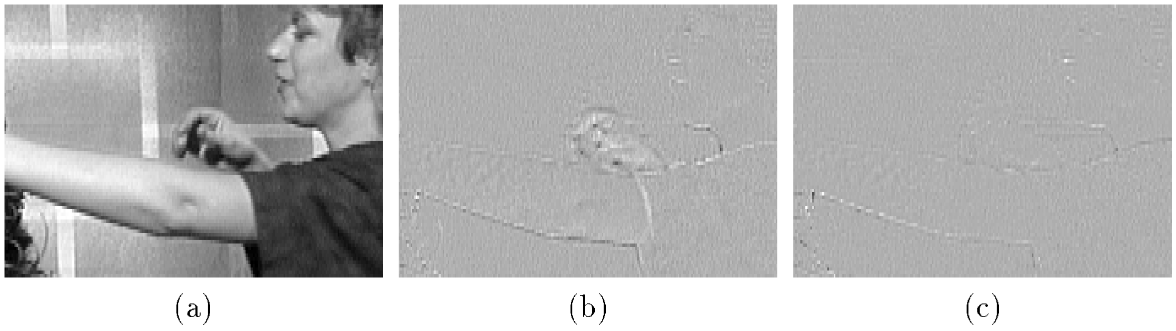


Figure 3.24: Estimation area (a) of field #26 of test image 3; interpolation error (magnified by 2) using: (b) linear trajectory model ($PSNR = 39.03$ dB); (c) quadratic trajectory model ($PSNR = 41.93$ dB) with $N = 5$.

The error images (between the reconstructed and the original ones) for the linear and quadratic trajectory models with $N = 5$ are shown in Figures 3.24, 3.25, and 3.26 for fields #26, 80, and 105 of test image 3, respectively. Also the estimated velocity and acceleration fields for fields #80 and 105 are shown in Figures 3.27 and 3.28, respectively.

It is evident from the motion field plots, and the interpolation error images that in the case of the linear trajectory model the errors are more concentrated in the regions that have accelerated motion (i.e., the hand and the arm). This explains the degradation in the $PSNR$ for the reconstructed fields when linear trajectory model is used in the estimation algorithm.

The same experiment was run on 33 fields of the test image 4 (over the estimation

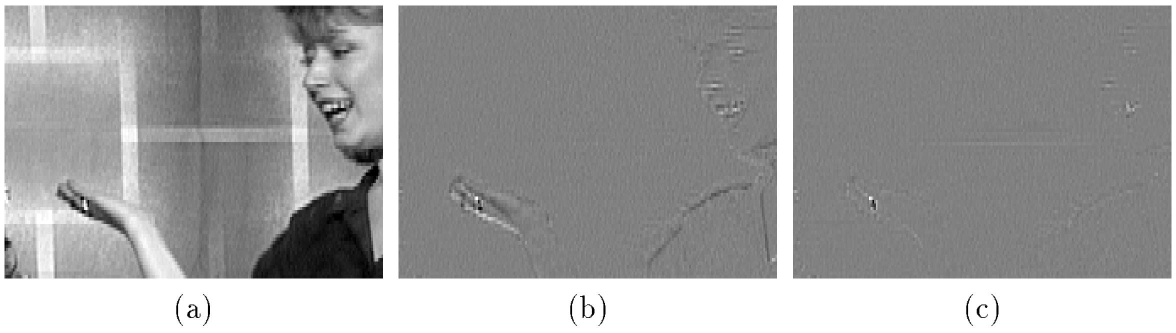


Figure 3.25: Estimation area (a) of field #80 of test image 3; interpolation error (magnified by 2) using: (b) linear trajectory model ($PSNR = 37.76$ dB); (c) quadratic trajectory model ($PSNR = 41.61$ dB) with $N = 5$.

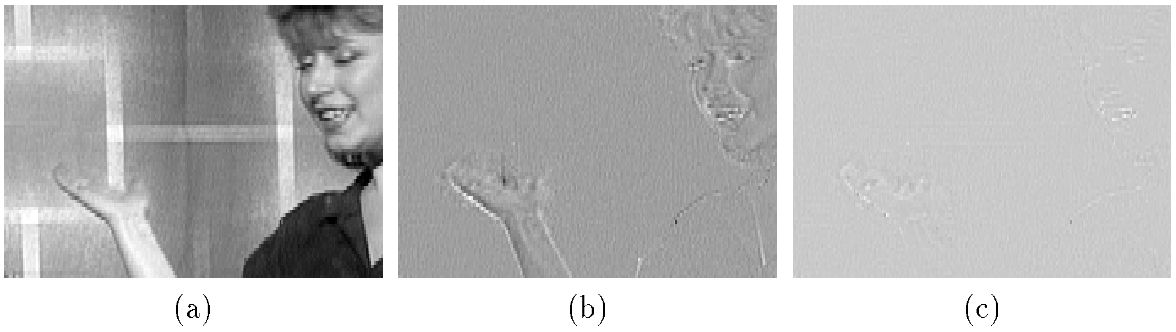


Figure 3.26: Estimation area (a) of field #105 of test image 3; interpolation error (magnified by 2) using: (b) linear trajectory model ($PSNR = 37.68$ dB); (c) quadratic trajectory model ($PSNR = 41.80$ dB) with $N = 5$.

area defined in Figure 3.6) with $L = 4$, $\lambda_p = 20$ and the sets $\mathcal{I}_t$ and $\mathcal{J}_t$ defined as in 3.1. The $PSNR$ curves for all the processed fields are plotted in Figure 3.29. The resulting means for $PSNR$ ($\overline{PSNR}$) obtained from the estimation algorithm are shown in Table 3.7 for the linear and quadratic trajectory models. Note that the use of the quadratic trajectory model resulted in an average gain of +4.39 dB with respect to the linear trajectory model with $N = 5$, and +5.72 dB with respect to the linear trajectory model with $N = 2$. These results are consistent with those in Table 3.6 for test image 3. However, higher gains have been achieved by using the quadratic

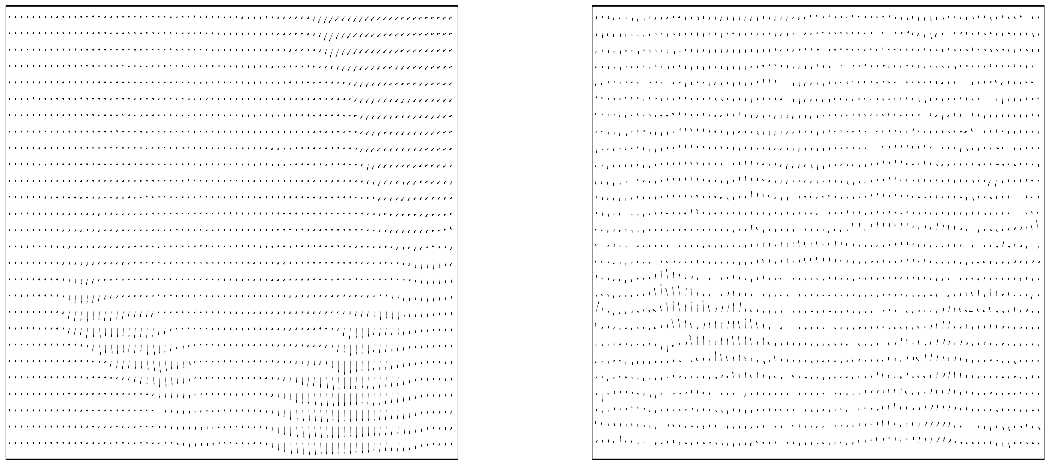

(a) (b)

Figure 3.27: Estimated: (a) velocity $\hat{\mathbf{v}}(80)$, and (b) acceleration $\hat{\mathbf{a}}$ (scaled by 4) at field #80 of test image 3.

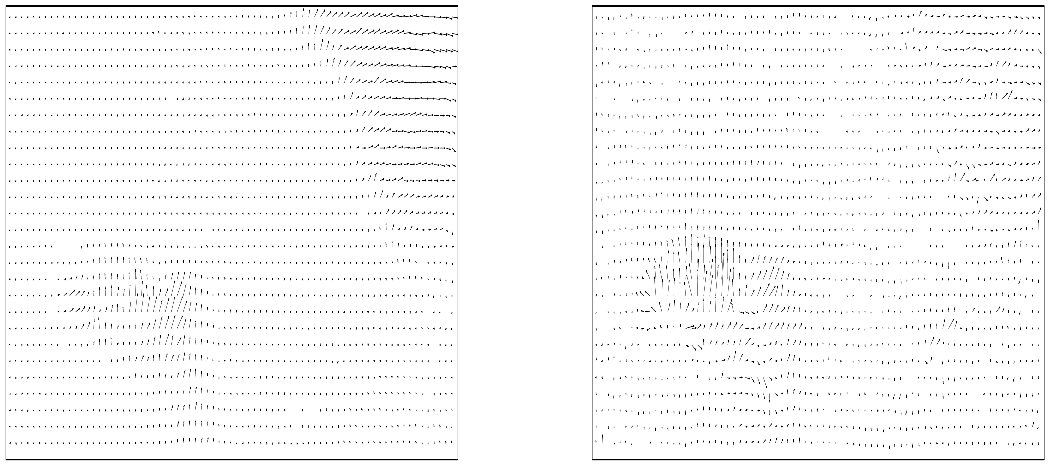

(a) (b)

Figure 3.28: Estimated: (a) velocity $\hat{\mathbf{v}}(105)$, and (b) acceleration $\hat{\mathbf{a}}$ (scaled by 4) at field #105 of test image 3.

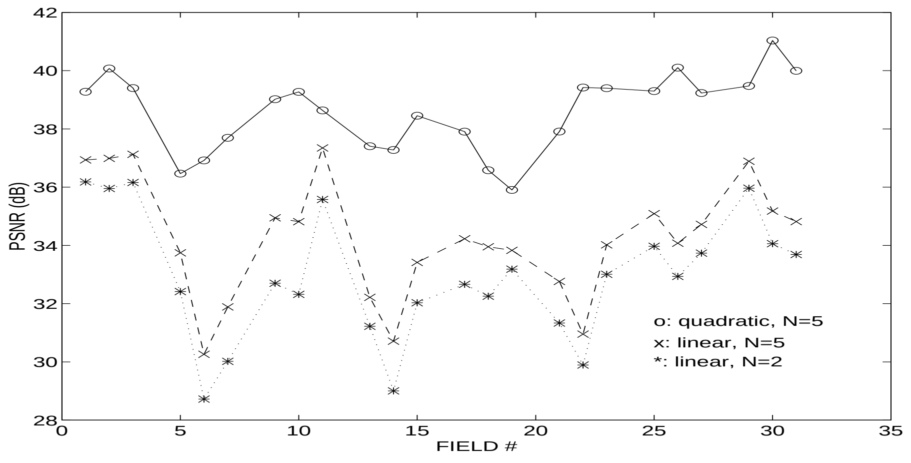


Figure 3.29: Comparison of the $PSNR$ for reconstructed fields from test image 4 using linear (dashed line) and quadratic (full line) motion trajectory models with $N = 5$ and linear trajectory model with $N = 2$ (dotted line).

Model	$\overline{PSNR}$ (dB)
linear ($N = 2$)	32.87
linear ($N = 5$)	34.20
quadratic ($N = 5$)	38.59

Table 3.7: Means for $PSNR$ evaluated over 24 fields of test image 4 using linear and quadratic motion trajectory models.

trajectory model. This is due to motion present in the sequence that is closer to the quadratic trajectory model than to the linear model. This hypothesis seems to be reasonable as test image 4 is a typical video conference sequence in which the movements of the mouth and the eyes of the speaker exhibit substantial acceleration.

The reconstructed images as well as the interpolation error for the linear and quadratic trajectory models with $N = 5$ are shown in Figures 3.30, 3.31, and 3.32 for fields #6, 14, and 22 of test image 4, respectively. Also, the estimated velocity and acceleration fields for these same fields are shown in Figures 3.33, 3.34, and 3.35.

It is worth to note here that most of the velocity and acceleration vectors are concentrated in the regions of the eyes and the mouth. These are basically the regions

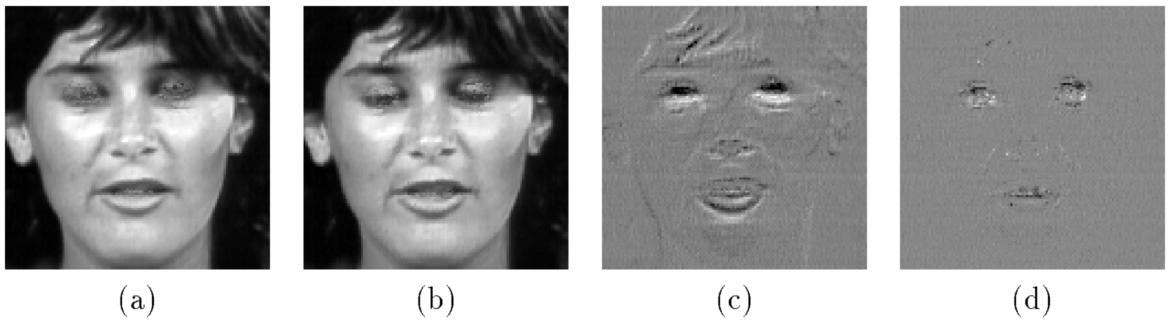


Figure 3.30: Reconstructed field #6 of test image 4 using: (a) linear ($PSNR = 30.26$ dB); (b) quadratic trajectory model ($PSNR = 36.91$ dB) with $N = 5$ and their respective error images (magnified by 2) in (c) and (d).

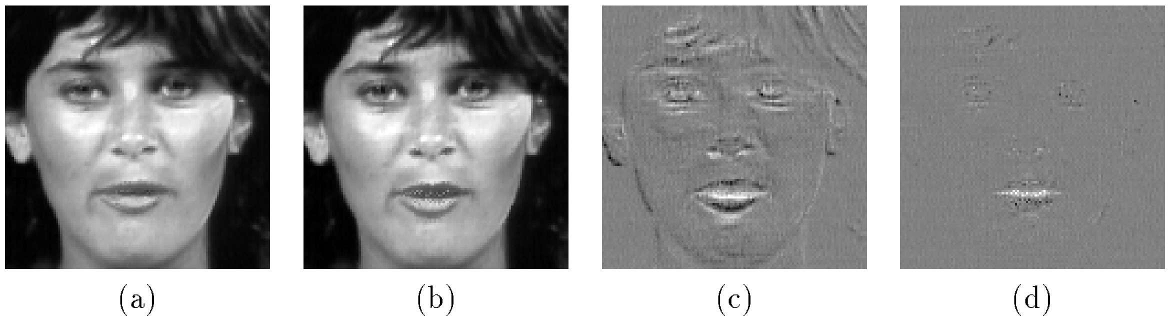


Figure 3.31: Reconstructed field #14 of test image 4 using: (a) linear ($PSNR = 30.71$ dB); (b) quadratic trajectory model ($PSNR = 37.27$ dB) with $N = 5$ and their respective error images (magnified by 2) in (c) and (d).

where the interpolation errors are concentrated and where the use of a linear or quadratic motion trajectory model makes the difference. The motion fields displayed in Figures 3.34 and 3.35, for instance, reflect well the opening and closure of the mouth, respectively. The estimated acceleration vectors in the region of the mouth result in a substantial decrease of the interpolation errors when a quadratic trajectory model is used. This is also reflected in the interpolated images where, in particular, the mouth in Figure 3.32a appears to be less open than in Figure 3.32b.

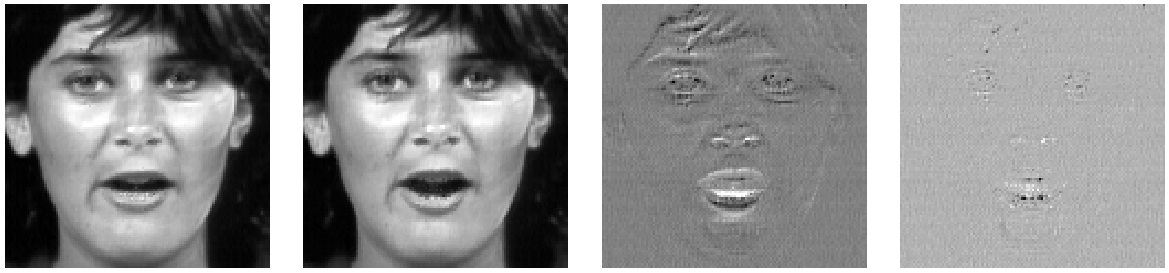


Figure 3.32: Reconstructed field #22 of test image 4 using: (a) linear ($PSNR = 30.95$ dB); (b) quadratic trajectory model ($PSNR = 39.42$ dB) with $N = 5$ and their respective error images (magnified by 2) in (c) and (d).

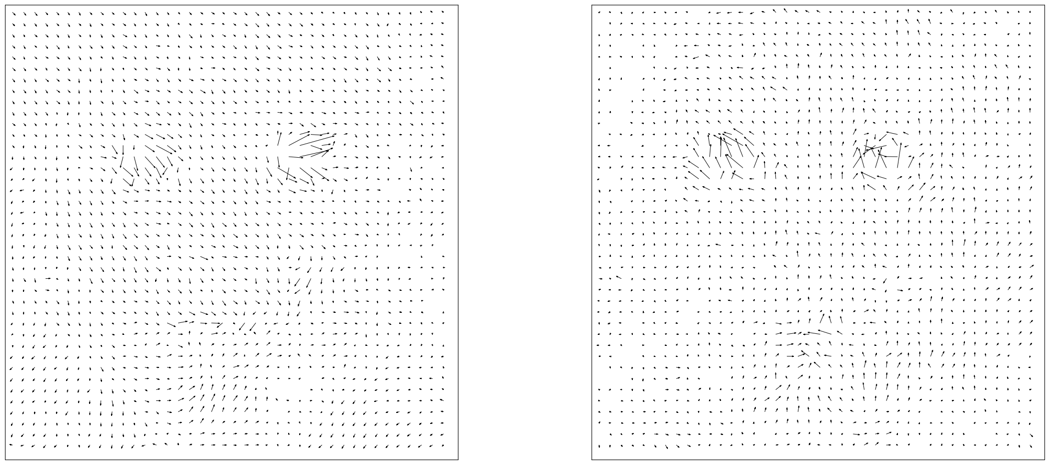
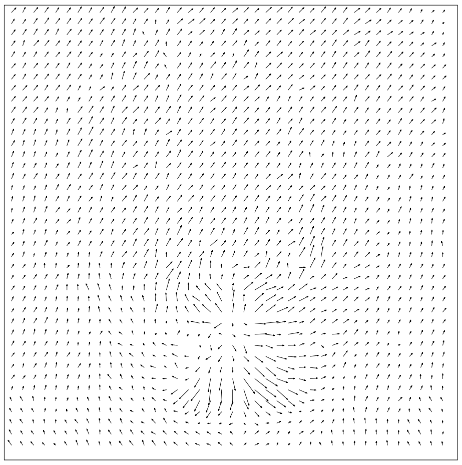


Figure 3.33: Estimated: (a) velocity $\hat{\mathbf{v}}(6)$ (scaled by 2); (b) acceleration $\hat{\mathbf{a}}$ (scaled by 4) at field #6 of test image 4.


(a)

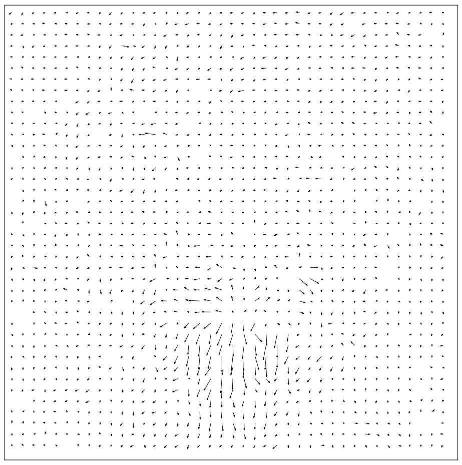
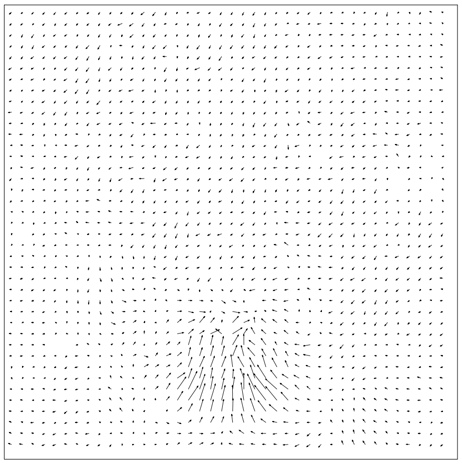

(b)

Figure 3.34: Estimated: (a) velocity $\hat{\mathbf{v}}(14)$ (scaled by 2); (b) acceleration $\hat{\mathbf{a}}$ (scaled by 4) at field #14 of test image 4.


(a)

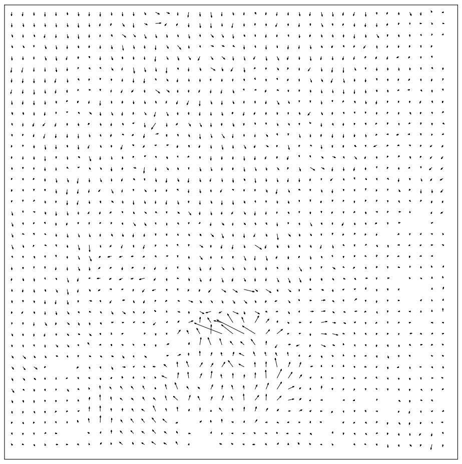

(b)

Figure 3.35: Estimated: (a) velocity $\hat{\mathbf{v}}(22)$ (scaled by 2); (b) acceleration $\hat{\mathbf{a}}$ (scaled by 4) at field #22 of test image 4.

Chapter 4

Estimation of Occlusions

The motion estimation algorithm that was proposed in Chapter 3 gives accurate motion estimates in moving or stationary regions. However erroneous motion estimates result in occluded regions (i.e., near motion discontinuities) due to the lack of occlusion processing in the motion estimation algorithm. This problem is reflected in motion-compensated interpolated images where the ability to reconstruct clearly moving boundaries fails in general. The need to process occluded regions becomes even more critical in the presence of accelerated motion which generally produces larger occluded areas. Also, the use of multiframe processing contributes to the need of processing the occluded regions. In this chapter, the motion estimation algorithm is modified in order to take occlusion effects into consideration. The modeling of occlusions and motion discontinuities is discussed in Section 4.1 along with the derivation of a new multiple-term objective function. Section 4.2 describes an optimization method that allows to minimize this objective function and compute piecewise smooth motion fields along with the corresponding occlusion and motion discontinuity fields. Experimental results for sequences with synthetic motion and for natural sequences are presented in Section 4.3.

4.1 Extension of the motion estimation algorithm

4.1.1 Definitions and reformulation of the problem

The estimation algorithm presented in Chapter 3 is made complete by taking into account occlusion effects present in dynamic images. This is possible, as explained in Section 2.4.3, by defining an *occlusion field* o and a *motion discontinuity* field l, often called line field or *line process* [14]. The occlusion field has its samples defined on the lattice Λ_p with sampling periods (T_p^h, T_p^v, T_p) [10]. Every occlusion tag o can take one of several possible states, e.g., moving/stationary (visible), exposed, or covered. The number of such states is finite and depends on the cardinality of $\mathcal{I}_t$. On the other hand, the line field is defined over a union of shifted lattices $\Psi_l = \psi_h \cup \psi_v$, where $\psi_h = \Lambda_p + [0 \;\; T_p^v/2 \;\; 0]^T$, and $\psi_v = \Lambda_p + [T_p^h/2 \;\; 0 \;\; 0]^T$ are orthorhombic cosets [10] specifying positions of horizontal and vertical discontinuities, respectively. Hence, each line element is defined between two pixel positions. The notation $l(\mathbf{x}_i, \mathbf{x}_j)$ will be used to denote the absence ($l(\mathbf{x}_i, \mathbf{x}_j) = 0$) or presence ($l(\mathbf{x}_i, \mathbf{x}_j) = 1$) of a motion discontinuity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$.

The algorithm is hence extended to determine the *most likely* triplet $(\mathbf{p}_t, o_t, l_t)$ corresponding to the true underlying image u based on observations $\mathcal{G}_t = \{g_{t+\tau} : \tau \in \mathcal{I}_t\}$. Assuming that occlusion fields o_t and line fields l_t are samples from scalar random fields O_t and L_t, respectively, the MAP estimate is obtained by extending equation 3.11 as follows:

$$\begin{aligned}
(\hat{\mathbf{p}}_t, \hat{o}_t, \hat{l}_t) &= \arg\max_{(\mathbf{p}_t, o_t, l_t)} \; P(\mathbf{p}_t, O_t = o_t, L_t = l_t | \mathcal{G}_t) \\
&= \arg\max_{(\mathbf{p}_t, o_t, l_t)} \; [\, P(\mathcal{G}_t^n | \mathbf{p}_t, o_t, l_t, g_{t_n}) \cdot P(\mathbf{p}_t | o_t, l_t, g_{t_n}) \cdot \\
&\qquad P(O_t = o_t | l_t, g_{t_n}) \cdot P(L_t = l_t | g_{t_n}) \,] \, .
\end{aligned} \tag{4.1}$$

In the following sections, models that allow to specify the constituent probabilities in (4.1) are investigated in order to derive the new objective function $U(\mathbf{p})$ (equation (3.22)).

4.1.2 Structural model

In the formulation of the structural model term (Section 3.2.1) it was assumed that the trajectory through a point $(\mathbf{x}, t)$ extends through the whole time interval defined by $\mathcal{I}_t$ (Figure 3.2). But this is only true if the necessary condition stated in (3.10) is satisfied, i.e., the point $(\mathbf{x}, t)$ is visible over the N fields defined by $\mathcal{I}_t$. Hence, at each spatio-temporal position $(\mathbf{x}, t)$ a subset of $\mathcal{I}_t$ can be defined as follows:

$$\mathcal{I}_t^{\mathbf{x}} = \{\tau \in \mathcal{I}_t : \mathbf{x} \in \mathcal{L}_t\}, \tag{4.2}$$

where $\mathcal{L}_t = (\Lambda_p)_t \cap V(\tau; t)$ is the set of all pixels on lattice Λ_p at time t visible in the image sequence between t and τ. $\mathcal{I}_t^{\mathbf{x}}$ is called the *visibility set* [12] and contains time instants from $\mathcal{I}_t$ at which pixel $(\mathbf{x}, t)$ is visible. This set can be directly derived from the occlusion state $o(\mathbf{x}, t)$ at $(\mathbf{x}, t)$ as illustrated in Table 4.1 for 3- and 5-image estimation. Only the most likely visibility/non-visibility combinations are taken into account for $N = 5$.

$\mathcal{I}_t$	$o(\mathbf{x}, t)$	Description	$\mathcal{I}_t^{\mathbf{x}}$
	M	moving/stationary	$\{-1, 0, 1\}$
$\{-1, 0, 1\}$	E	exposed	$\{0, 1\}$
	C	covered	$\{-1, 0\}$
	M	moving/stationary	$\{-2, -1, 0, 1, 2\}$
	E	exposed in $(t-1, t)$	$\{0, 1, 2\}$
$\{-2, -1, 0, 1, 2\}$	E_{-1}	exposed in $(t-2, t-1)$	$\{-1, 0, 1, 2\}$
	C	covered in $(t, t+1)$	$\{-2, -1, 0\}$
	C_{+1}	covered in $(t+1, t+2)$	$\{-2, -1, 0, 1\}$

Table 4.1: Table of occlusion states and visibility sets for $\mathcal{I}_t = \{-1, 0, 1\}$ and $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$.

The structural model term in (3.17), which relies on the sample variance measure (equation (3.13)), is then modified as follows:

$$U_s(\mathbf{p}, o) = \sum_{i=1}^{N_P} \left\{ \sum_{k \in \mathcal{I}_t^{\mathbf{x}_i}} \left[\tilde{g}(\mathbf{x}_i^k, t+k) - \zeta(\mathbf{x}_i, t) \right]^2 \right\}, \tag{4.3}$$

with

$$\zeta(\mathbf{x}_i, t) = \frac{1}{Card(\mathcal{I}_t^{\mathbf{x}_i})} \sum_{\tau \in \mathcal{I}_t^{\mathbf{x}_i}} \tilde{g}(\mathbf{x}_i^\tau, t+\tau) \tag{4.4}$$

and the position $\mathbf{x}_i^k$ defined as in (3.15). Hence, the new structural term $U_s(\mathbf{p}, o)$ becomes dependent on the occlusion field o in the sense that only the fields referenced by the visibility set $\mathcal{I}_t^{\mathbf{x}_i}$ at position $(\mathbf{x}_i, t)$ will contribute to the evaluation of the sample variance. However, the dependence of $U_s(\mathbf{p}, o)$ on the line field l has been omitted since the information about motion discontinuities will be conveyed through the motion trajectory model.

It is worth also to mention that $U_s(\mathbf{p}, o)$ is the energy function of Gibbs distribution (equation (3.16)) that models the conditional distribution $P(\mathcal{G}_t^n | \mathbf{p}_t, o_t, l_t, g_{t_n})$ in (4.1).

4.1.3 Motion trajectory model

Defining motion vectors $\mathbf{p}_t$ as samples from a vector MRF $\mathbf{P}_t$, as discussed in Section 3.2.2, the conditional distribution $P(\mathbf{p}_t | o_t, l_t, g_{t_n})$ in (4.1) can be expressed by a Gibbs distribution whose energy function (equation (3.19)) is modified as follows:

$$U_p(\mathbf{p}, l) = \sum_{i=1}^{N_P} \sum_{\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{C}_i} (\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{\Gamma} (\mathbf{p}_i - \mathbf{p}_j)[1 - l(\mathbf{x}_i, \mathbf{x}_j)]. \tag{4.5}$$

This energy captures the desired smoothness property of the motion field for the first-order neighborhood system η^1 (Figure 4.1) only in the absence of motion discontinuities. The dependence of the smoothness term $U_p(\mathbf{p}, l)$ on the line field l by the multiplicative term $[1 - l(\mathbf{x}_i, \mathbf{x}_j)]$ has been investigated in Section 2.4.3 whereby a jump in motion parameters is not penalized if a motion discontinuity had been detected. However, the dependence on the occlusion field o is not necessary here, since

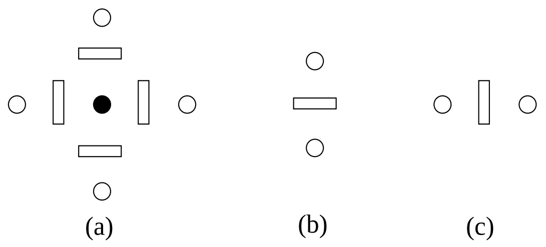


Figure 4.1: First-order neighborhood system η^1 (a) for motion field $\mathbf{p}$ defined over Λ_p with motion discontinuities l defined over Ψ_l; (b) vertical clique; (c) horizontal clique (empty circle: motion position; filled circle: central motion position; rectangle: position of a line element).

the information about occlusion state transition is considered to be passed by a line element.

4.1.4 Occlusion model

In [7] the occlusion field o_t was modeled for the case of estimation from 3 images (i.e., $\mathcal{I}_t = \{-1, 0, 1\}$). A similar approach is used here for the case of 5-image estimation (i.e., $N = 5$). The five possible states of an occlusion label $o(\mathbf{x}, t)$ are shown in Table 4.1 for $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$. The occlusion field o_t is thus modeled by a discrete-valued scalar MRF described by the following Gibbs distribution

$$P(O_t = o_t | l_t, g_{t_n}) = \frac{1}{Z_o} e^{-\frac{U_o(o,l)}{\beta_o}}, \tag{4.6}$$

with Z_o and β_o being constants. The energy function $U_o(o, l)$ is defined as follows:

$$U_o(o,l) = \sum_{i=1}^{N_P} \left\{ V_{o_1}\left(o(\mathbf{x}_i, t)\right) + \sum_{\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{C}_i} V_{o_2}\left(o(\mathbf{x}_i, t), o(\mathbf{x}_j, t), l(\mathbf{x}_i, \mathbf{x}_j)\right) \right\}, \tag{4.7}$$

where V_{o_1} and V_{o_2} are potential functions associated with single- and two-element cliques respectively. These cliques are chosen from the first-order neighborhood system η^1 shown in Figure 4.1. It is expected that a typical occlusion field consists mostly of patches of pixels labeled as visible, and some smaller clusters of pixels labeled as occluded. Therefore, the potential function V_{o_1} provides a penalty associated with the introduction of an occlusion state, whereas the potential function V_{o_2} favors the creation of continuous occlusion regions near motion discontinuities only. To achieve this goal, the dependence of V_{o_2} on the line field l is utilized. For instance, whenever $(\mathbf{x}_i, t)$ and $(\mathbf{x}_j, t)$ have the same occlusion state, $V_{o_2}(o(\mathbf{x}_i, t), o(\mathbf{x}_j, t), l(\mathbf{x}_i, \mathbf{x}_j))$ is set to 0 (high probability) if the two positions are not separated by a motion discontinuity (i.e., $l(\mathbf{x}_i, \mathbf{x}_j) = 0$) and to a high value (low probability) if they are separated by a motion discontinuity (i.e., $l(\mathbf{x}_i, \mathbf{x}_j) = 1$).

The assigned costs associated with all possible configurations of single- and two-element cliques in a 5-state occlusion field are shown in Figure 4.2. These costs are

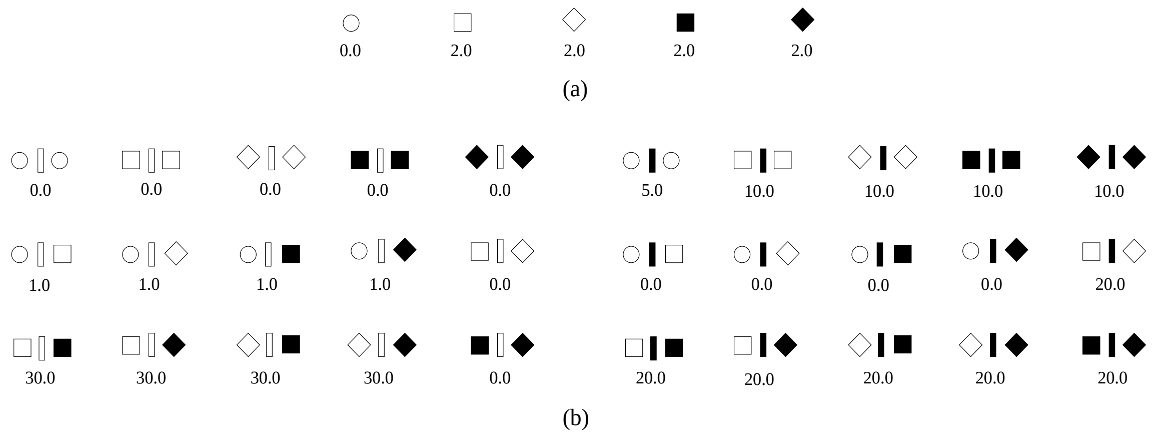


Figure 4.2: Costs assigned to: (a) V_{o_1}; (b) V_{o_2} for various configurations (up to rotation and permutation) of occlusion cliques for $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ (occlusion states: circle (M); empty square (E); empty diamond (E_{-1}); filled square (C); filled diamond (C_{+1}), line element states: empty rectangle ("off"); filled rectangle ("on")).

chosen experimentally, and therefore are not optimal in any way. Basically, the costs associated with 2-element cliques are chosen in a way to discourage the occurrence of an incompatible combination of neighboring occlusion tags (i.e., E and C), and to favor the creation of clusters of occluded pixels near motion discontinuities.

4.1.5 Motion discontinuity model

In order to model continuity of motion boundaries, the motion discontinuity field l_t is modeled by a binary MRF L_t [12] described by the Gibbs distribution

$$P(L_t = l_t | g_{t_n}) = \frac{1}{Z_l} e^{-\frac{U_l(l,g)}{\beta_l}}, \tag{4.8}$$

with Z_l and β_l being the usual constants. The energy function $U_l(l)$ is defined as follows:

$$U_l(l,g) = \sum_{i=1}^{N_l} \left\{ V_{l_1}(l_i, e_i) + \sum_{c_i \in \mathcal{C}_{l_2}} V_{l_2}(l_t, c_i) + \sum_{c_i \in \mathcal{C}_{l_3}} V_{l_3}(l_t, c_i) \right\}, \tag{4.9}$$

where c_i is a clique, N_l is the number of horizontal and vertical line elements in the motion discontinuity field l_t, l_i denotes a horizontal or vertical line element at position $\mathbf{x}_i \in (\Psi_l)_t$, and e_i denotes presence ($e_i = 1$) or absence ($e_i = 0$) of an intensity edge at $\mathbf{x}_i$. A line element l_i is said to be turned "on" (respectively "off") at $\mathbf{x}_i$ if $l_i = 1$ (respectively $l_i = 0$) i.e., a motion discontinuity is present (respectively absent) at $\mathbf{x}_i$. V_{l_1}, V_{l_2}, and V_{l_3} are potential functions associated with single-element, four-element square-shaped, and four-element cross-shaped cliques, respectively (Figure 4.3). These cliques are chosen from a sufficiently large neighborhood system η_l [31]

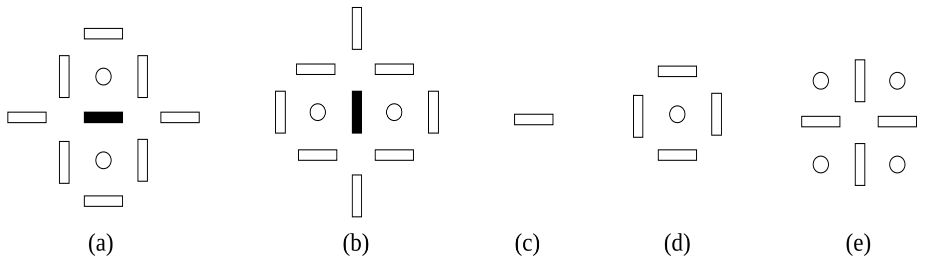


Figure 4.3: Neighborhood system η_l for motion discontinuity field l defined over Ψ_l (a) for horizontal discontinuity defined over ψ_h; (b) for vertical discontinuity defined over ψ_v; (c) single-element clique; (d) four-element square-shaped clique; (e) four-element cross-shaped clique (empty rectangle: positions of a line element; filled rectangle: position of the central line element; circle: pixel position).

defined over Ψ_l at a horizontal or vertical discontinuity (Figure 4.3(a) and 4.3(b)).

The set of four-element square-shaped cliques is denoted by $\mathcal{C}_{l_2}$ while the set of four-element cross-shaped cliques is denoted by $\mathcal{C}_{l_3}$.

It is assumed that, in general, the introduction of a motion discontinuity should coincide with an intensity edge. This is enforced by the potential function V_{l_1} which uses single-element cliques to associate a high penalty whenever a motion discontinuity does not match an intensity edge [21]. V_{l_1} can therefore be formulated as follows:

$$V_{l_1}(l_i, e_i) = 10 \cdot (1.1 - e_i) \cdot l_i, \tag{4.10}$$

with l_i and e_i denoting the values of the line element and the intensity edge, respectively, at position $\mathbf{x}_i$. Hence, the introduction of a motion discontinuity ($l_i = 1$) is penalized by 11 in the absence of an intensity edge ($e_i = 0$) and by 1 if such edge is present ($e_i = 1$). This latter value assures a penalty associated with the introduction of a line element since otherwise such elements could be introduced everywhere on $(\Psi_l)_t$ to bring the energy (4.5) to zero.

The field of intensity edges e_t at t is calculated *a priori* by the application of Canny [6] edge detector $\mathcal{E}$ to the observation field g_t, i.e., $e_t = \mathcal{E}(g_t)$. This operator consists of finding zero-crossings (i.e., intensity edges) of a smoothed version of g_t at positions $\mathbf{x}_i \in (\Psi_l)_t$, along direction $\mathbf{n}$, as follows:

$$\mathcal{E}: \quad \frac{\partial}{\partial \mathbf{n}}\left(\frac{\partial}{\partial \mathbf{n}}\left(h(\mathbf{x}_i) * g(\mathbf{x}_i, t)\right)\right) = 0, \qquad i = 1, \cdots, N_l \tag{4.11}$$

where h is a 2-D Gaussian

$$h(\mathbf{x}) = e^{\frac{-|\mathbf{x}|^2}{2\sigma^2}}, \tag{4.12}$$

and $\frac{\partial}{\partial \mathbf{n}}$ is the directional derivative with respect to $\mathbf{n}$ which represents the direction normal to the intensity edge e_i at $\mathbf{x}_i \in (\Psi_l)_t$. Since only horizontal and vertical intensity edges are needed, then

$$\mathbf{n} = \begin{cases} [0 \quad 1]^T & \text{if } \mathbf{x}_i \in (\psi_h)_t, \\ [1 \quad 0]^T & \text{if } \mathbf{x}_i \in (\psi_v)_t. \end{cases} \tag{4.13}$$

The operator $\mathcal{E}$ in (4.11) can be equivalently characterized by

$$\mathcal{E}: \quad \mathbf{n}^T \cdot \nabla^2_{\mathbf{x}}\left(h(\mathbf{x}_i) * g(\mathbf{x}_i, t)\right) \cdot \mathbf{n} = 0, \qquad i = 1, \cdots, N_l, \tag{4.14}$$

with $\nabla^2_{\mathbf{x}}$ being the spatial Hessian matrix. This explains the dependence of the *a priori* probability of the line process (4.8) on the observation g_{t_n}.

The control over straight lines, corners and intersections is achieved by the penalty functions V_{l_2} and V_{l_3} using the four-element cliques in Figure 4.3(d) and 4.3(e), respectively. V_{l_2}, in particular, discourages formation of double lines and also inhibits the generation of isolated trajectories while V_{l_3} discourages the creation of unended and intersecting segments.

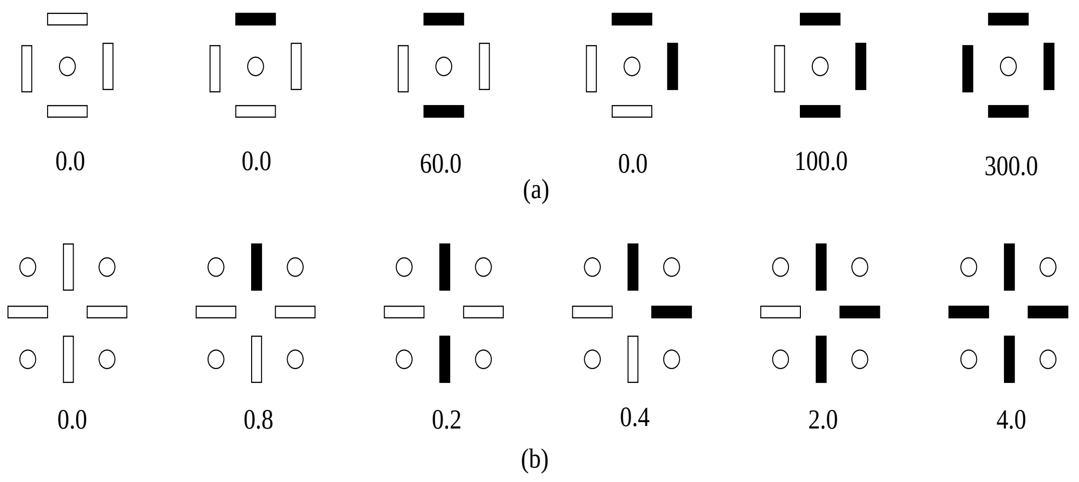


Figure 4.4: Costs assigned to: (a) V_{l_2}; (b) V_{l_3} for various configurations (up to rotation) of line cliques (filled rectangle: line element "on", empty rectangle: line element "off', circle: pixel position).

Figure 4.4 shows the proposed costs (inspired by [7]) associated with different configurations (up to a rotation) of four-element square- and cross-shaped cliques.

4.1.6 New objective function

Substituting the conditional probabilities in (4.1), the following optimization problem results

$$(\hat{\mathbf{p}}_t, \hat{o}_t, \hat{l}_t) = \min_{\{\mathbf{p}_t, o_t, l_t\}} U(\mathbf{p}_t, o_t, l_t), \tag{4.15}$$

where $U(\mathbf{p}, o, l)$ is the new multiple-term objective function expressed as follows:

$$U(\mathbf{p}, o, l) = U_s(\mathbf{p}, o) + \lambda_p U_p(\mathbf{p}, l) + \lambda_o U_o(o, l) + \lambda_l U_l(l, g). \tag{4.16}$$

The constituent energies in (4.16) are defined in (4.3), (4.5), (4.7), and (4.9) respectively, and $\lambda_p = 1/\beta_p$, $\lambda_o = 1/\beta_o$, $\lambda_l = 1/\beta_l$ denote their respective associated weights.

4.2 Optimization method

Since trajectories are described by continuous-valued parameters while occlusions and motion discontinuities are described by finite discrete state spaces, different optimization methods must be used to estimate $\mathbf{p}$, o, and l. This can be accomplished by solving the minimization problem in (4.15) in an interleaved fashion, i.e., while one field is iteratively updated, the others are kept constant. Hence, at each iteration (full scan of a field) n, the three fields $\mathbf{p}_t$, o_t, and l_t are updated consecutively by performing one iteration of the following minimization problems

$$\begin{aligned}
&\text{(a)} \quad \mathbf{p}_t^n = \arg\min_{\{\mathbf{p}_t\}}\{U_s(\mathbf{p}_t, o_t) + \lambda_p U_p(\mathbf{p}_t, l_t)\} \quad \text{with} \begin{cases} o_t = o_t^{(n-1)} \\ l_t = l_t^{(n-1)} \end{cases} \\
&\text{(b)} \quad l_t^n = \arg\min_{\{l_t\}}\{\lambda_p U_p(\mathbf{p}_t, l_t) + \lambda_o U_o(o_t, l_t) + \lambda_l U_l(l_t, g_t)\} \quad \text{with} \begin{cases} \mathbf{p}_t = \mathbf{p}_t^{(n-1)} \\ o_t = o_t^{(n-1)} \end{cases} \\
&\text{(c)} \quad o_t^n = \arg\min_{\{o_t\}}\{U_s(\mathbf{p}_t, o_t) + \lambda_o U_o(o_t, l_t)\} \quad \text{with} \begin{cases} \mathbf{p}_t = \mathbf{p}_t^{(n-1)} \\ l_t = l_t^{(n-1)} \end{cases}
\end{aligned} \tag{4.17}$$

respectively. Once all three fields have been updated, the process is repeated until a suitable convergence of $U(\mathbf{p}, o, l)$ is achieved.

4.2.1 Optimization of the motion field

Optimization of the motion field $\mathbf{p}_t$ in equation (4.17a) is carried out using the deterministic relaxation algorithm discussed in Chapter 3. The same iterative algorithm, as derived in Section 3.3.2, results with some minor modifications. Hence, the updating of a motion vector $\mathbf{p}_i^n$ at iteration n is accomplished by solving the linear system: $\mathbf{A}_i^n \cdot \mathbf{p}_i^n = \mathbf{b}_i^n$. The matrix $\mathbf{A}_i^n$ and vector $\mathbf{b}_i^n$ are defined in (3.33) and (3.34) respectively with the set $\mathcal{I}_t$ replaced by the visibility set $\mathcal{I}_t^{\mathbf{x}_i}$, and the average motion vector $\bar{\mathbf{p}}_i$ at position $\mathbf{x}_i$ modified as follows:

$$\bar{\mathbf{p}}_i = \frac{1}{\xi_i} \sum_{j \in \eta^1(i)} \mathbf{p}_j [1 - l(\mathbf{x}_i, \mathbf{x}_j)], \tag{4.18}$$

with

$$\xi_i = \sum_{j \in \eta^1(i)} [1 - l(\mathbf{x}_i, \mathbf{x}_j)]. \tag{4.19}$$

4.2.2 Optimization of the line and occlusion fields

Optimization of the line and occlusion fields is carried out by solving the minimization problems in equations (4.17b) and (4.17c) respectively using Besag's *Iterated Conditional Modes* (ICM) method [4]. Each iteration of this method consists of two scans. During the first scan some selected positions are visited and the line element/occlusion state at each position is updated using an exhaustive search over all possible states (2 states for the line field, 5 states for the occlusion field). The line element/occlusion state that yields the lowest energy is chosen as the new state. The remaining positions are then visited during the second scan. Such a procedure has been chosen in order to break the dependence of line/occlusion state from neighboring states, and thus to allow quick convergence.

4.3 Simulation results

4.3.1 Definition of parameters

The modified estimation algorithm has been simulated on some test images with the main parameters chosen as follows:

1. The number of resolution levels L in the image pyramid is set to 4. Hence, at each resolution level k, coarse motion fields are estimated along with the corresponding occlusion and line fields before switching to the next finer resolution level $k-1$ (Figure 2.4). An intensity edge field e_t for each resolution level is

therefore needed in the estimation algorithm. These intensity edges are prepared *a priori* by application of the Canny edge detection operator $\mathcal{E}$, described in (4.14), at each resolution level.

2. The quadratic motion trajectory model is used in the subsequent simulations. Hence, both velocity and acceleration fields are estimated.

3. The regularization parameters (λ_o, λ_l) associated respectively with the occlusion term $U_o(o, l)$ and the line term $U_l(l, g)$ of the new objective function $U(\mathbf{p}, o, l)$ (4.16) are chosen experimentally for each image. On the other hand, the regularization parameter associated with the smoothness term $U_p(\mathbf{p}, l)$ was set to $\lambda_p = 20$.

4. $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ is selected in the motion estimation algorithm. However, $\mathcal{I}_t = \{-1, 0, 1\}$ will be used sometimes to compare the estimated occlusion and motion discontinuity fields for $N = 5$ and $N = 3$.

5. The set of time instants used in the motion-compensated interpolation is set to $\mathcal{J}_t = \mathcal{I}_t^{\mathbf{x}} \cap \{-2, 2\}$. Hence the set $\mathcal{J}_t$ is now adapted to occlusion labels. Also, a comparison with the interpolated sequences generated using motion estimates from the previous algorithm (Chapter 3) will be possible.

In the following, algorithm $\mathcal{A}$ is used to represent the motion estimation algorithm discussed in Chapter 3 (without processing of occlusion areas and motion discontinuities), whereas algorithm $\mathcal{B}$ represents the new modified algorithm that estimates piecewise-continuous motion along with occlusion areas and motion discontinuities.

4.3.2 Results for synthetic sequences

In order to verify the accuracy of the occlusion and motion discontinuity estimates, test image 5 with synthetic motion has been generated using the procedure from Section 3.4.1. The test image 5, shown in Figure 4.5a, differs from test images 1 and 2 by the fact that it is a highly detailed sequence thus making it more challenging for motion estimation. The ability to isolate moving contours of the rectangle from

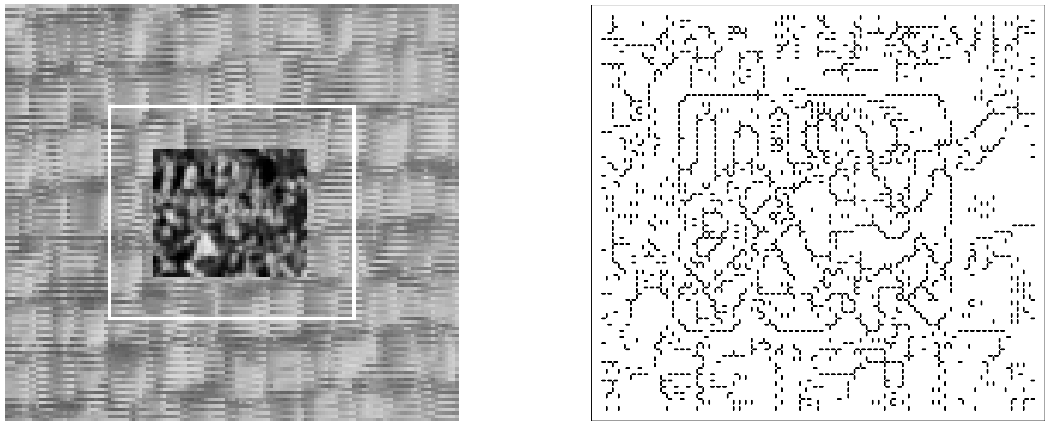

(a) (b)

Figure 4.5: Field #2 of (a) test image 5 (the white frame encircles the area used for estimation); (b) intensity edges detected by the edge detection operator $\mathcal{E}$ applied to the encircled area.

the rest of the intensity edges will demonstrate the validity of motion discontinuity estimates. Therefore, test image 5 should be a good test sequence for the modified estimation algorithm $\mathcal{B}$. Intensity edges computed within the estimation area by the Canny operator $\mathcal{E}$ are shown in Figure 4.5b.

The algorithm has been tested on field #2 of test image 5 with $(\lambda_o, \lambda_l) = (7, 5)$, and $\mathbf{p}(2) = [2 \;\; 0 \;\; 0 \;\; 0]^T$ being the real motion parameters of the moving rectangle at field #2. The resulting motion field estimates for algorithms $\mathcal{A}$ and $\mathcal{B}$ are shown in Figures 4.6 and 4.7, respectively.

Note that the estimates in Figure 4.7 obtained by algorithm $\mathcal{B}$ are more accurate around the moving rectangle than those obtained by algorithm $\mathcal{A}$. The reason for this is that the successful estimation of motion discontinuities (Figure 4.8b) in algorithm $\mathcal{B}$ was helpful in disabling the motion smoothness constraint around these discontinuities and hence providing a piecewise-constant estimate. This corresponds better to the true underlying motion and leads to a substantial decrease in the MSE of the

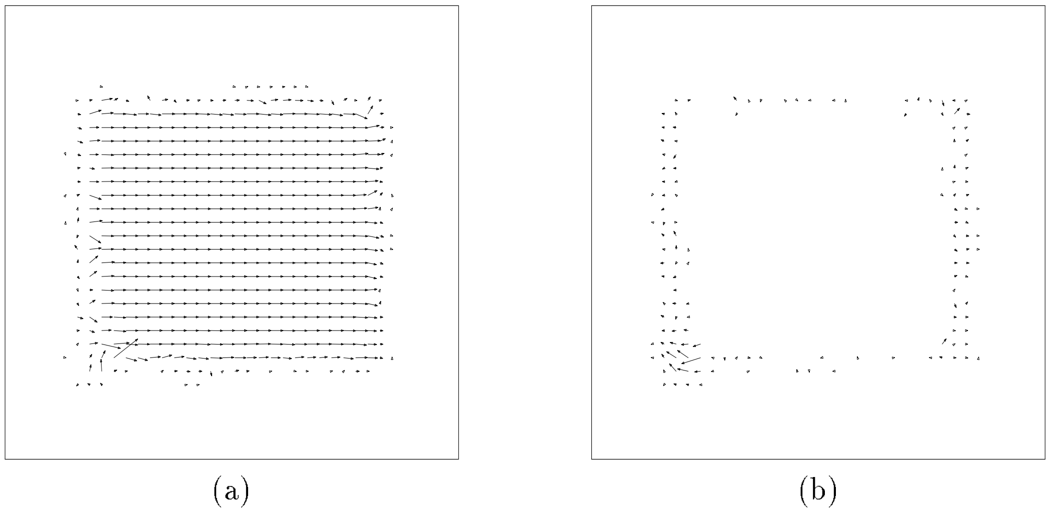


Figure 4.6: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ using algorithm $\mathcal{A}$ at field #2 of test image 5 with $\mathbf{p}(2) = [2 \;\; 0 \;\; 0 \;\; 0]^T$.

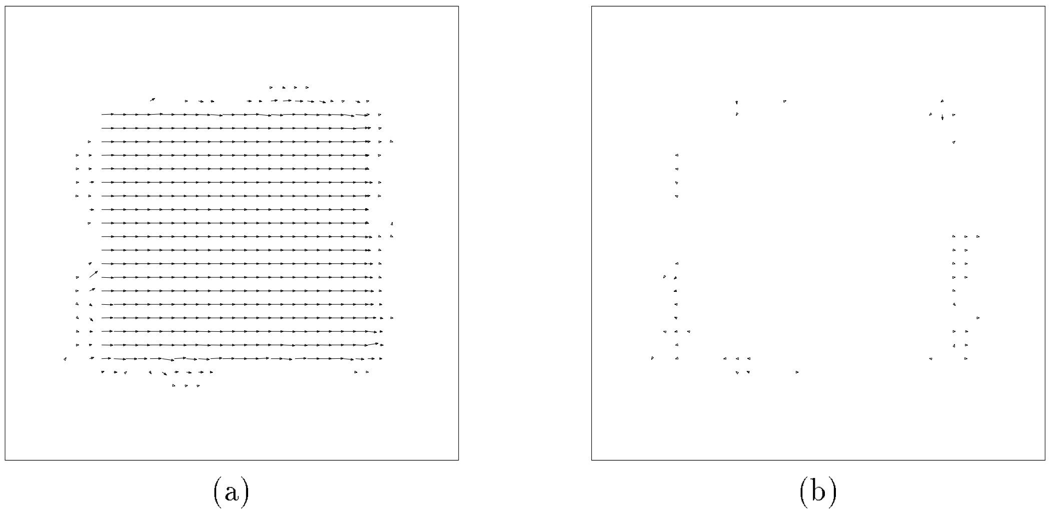


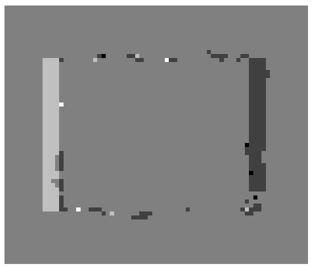
Figure 4.7: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$, and (b) acceleration $\hat{\mathbf{a}}$ using algorithm $\mathcal{B}$ at field #2 of test image 5 with $\mathbf{p}(2) = [2 \;\; 0 \;\; 0 \;\; 0]^T$.

estimates (Table 4.2) in the overall estimation area $\mathcal{R}_0$ and in the moving rectangle area $\mathcal{R}_1$ (with its boundaries). The estimated occlusion $\hat{o}(2)$ and motion discontinuity

	Algorithm	$v_x(2)$	$v_y(2)$	a_x	a_y
MSE in $\mathcal{R}_0$	$\mathcal{A}$	0.085129	0.021696	0.029210	0.011074
	$\mathcal{B}$	0.024777	0.002168	0.001252	0.002068
MSE in $\mathcal{R}_1$	$\mathcal{A}$	0.026600	0.023800	0.027487	0.014547
	$\mathcal{B}$	0.005794	0.002501	0.000659	0.004616

Table 4.2: Comparison of the MSE of motion estimates in $\mathcal{R}_0$ and $\mathcal{R}_1$ resulting from application of algorithms $\mathcal{A}$ and $\mathcal{B}$ at field #2 of test image 5 with $\mathbf{p}(2) = [2\ \ 0\ \ 0\ \ 0]^T$.

fields $\hat{l}(2)$ are shown in Figure 4.8 for $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ ($N = 5$) and in Figure 4.9 for $\mathcal{I}_t = \{-1, 0, 1\}$ ($N = 3$). The assigned intensity level to each possible occlusion state in the occlusion field is: $M = 128$, $E = 192$, $E_{-1} = 255$, $C = 64$, and $C_{+1} = 0$.


(a)

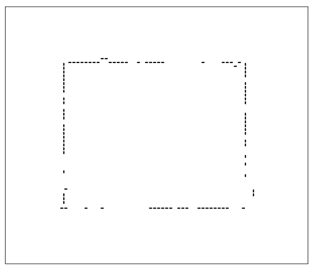

(b)

Figure 4.8: Estimated: (a) occlusion field $\hat{o}(2)$, and (b) line field $\hat{l}(2)$ at field #2 of test image 5 with $\mathbf{p}(2) = [2\ \ 0\ \ 0\ \ 0]^T$ and $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$.

Note that the detected occluded regions in Figure 4.8a for $N = 5$ are consistent with the horizontal motion (from left to right) of the rectangle moving at a constant velocity of 2 pixels per field. The dark area represents the area that is going to be covered within the next 2 time intervals whereas the bright area represent the area that has been exposed within the previous 2 time intervals. A comparison with the detected occlusion regions for $N = 3$ [9] (Figure 4.9a) shows that multiframe

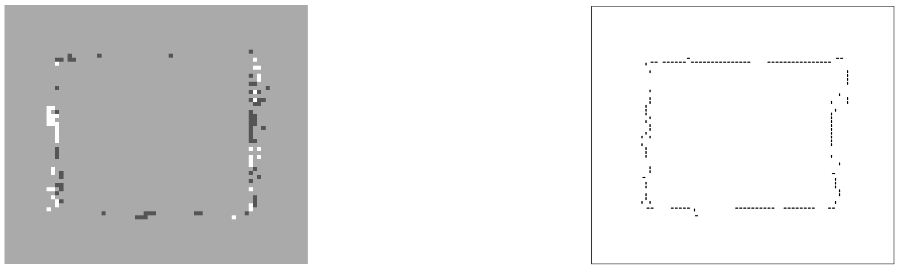

(a) (b)

Figure 4.9: Estimated: (a) occlusion field $\hat{o}(2)$, and (b) line field $\hat{l}(2)$ at field #2 of test image 5 with $\mathbf{p}(2) = [2 \;\; 0 \;\; 0 \;\; 0]^T$ and $\mathcal{I}_t = \{-1, 0, 1\}$.

processing (i.e., $N = 5$) helps in a better identification of occluded regions. This is also true for the identification of motion discontinuities (Figures 4.8b and 4.9b).

So far, Algorithm $\mathcal{B}$ has been tested on test image 5 with the rectangle moving horizontally and no acceleration. In a second experiment, algorithm $\mathcal{B}$ has been tested on field #2 of test image 5 with $(\lambda_o, \lambda_l) = (7, 9)$, and $\mathbf{p}(2) = [4 \;\; 4 \;\; 1 \;\; 1]^T$. In this case, the rectangle is moving diagonally (from top-left to bottom-right) with an acceleration of $\mathbf{a} = [1 \;\; 1]^T$ per field.

The resulting motion field estimates for algorithms $\mathcal{A}$ and $\mathcal{B}$ are shown in Figures 4.10 and 4.11, respectively, and the MSE of the resulting motion estimates in regions $\mathcal{R}_0$ and $\mathcal{R}_1$ is reported in Table 4.3.

	Algorithm	$v_x(2)$	$v_y(2)$	a_x	a_y
MSE in $\mathcal{R}_0$	$\mathcal{A}$	0.964976	1.140209	0.243863	0.197281
	$\mathcal{B}$	0.616461	0.858218	0.129602	0.096950
MSE in $\mathcal{R}_1$	$\mathcal{A}$	0.232956	0.211468	0.096320	0.072566
	$\mathcal{B}$	0.019926	0.024808	0.003422	0.004236

Table 4.3: Comparison of the MSE of motion estimates in $\mathcal{R}_0$ and $\mathcal{R}_1$ resulting from application of algorithms $\mathcal{A}$ and $\mathcal{B}$ at field #2 of test image 5 with $\mathbf{p}(2) = [4 \;\; 4 \;\; 1 \;\; 1]^T$.

The estimated occlusion $\hat{o}(2)$ and motion discontinuity fields $\hat{l}(2)$ are shown in

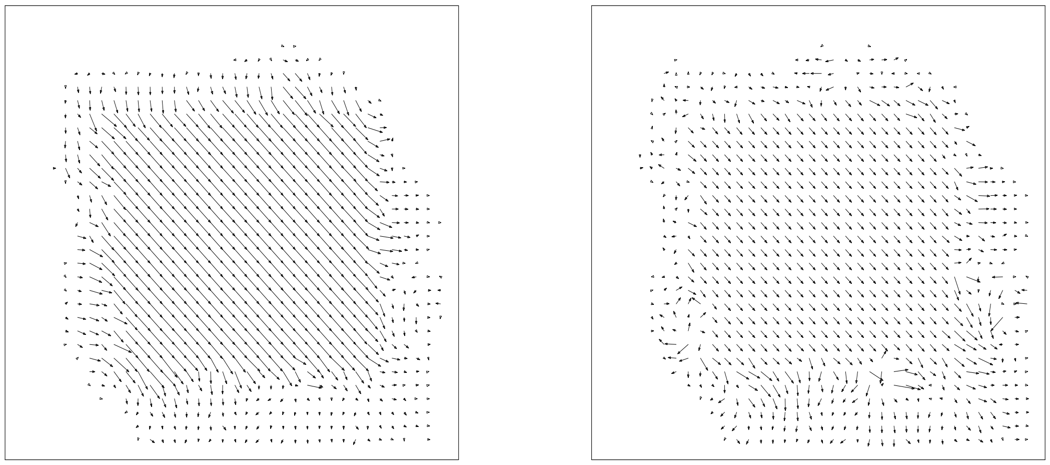

(a) (b)

Figure 4.10: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$ (scaled by 0.5), and (b) acceleration $\hat{\mathbf{a}}$ using algorithm $\mathcal{A}$ at field #2 of test image 5 with $\mathbf{p}(2) = [4 \;\; 4 \;\; 1 \;\; 1]^T$.

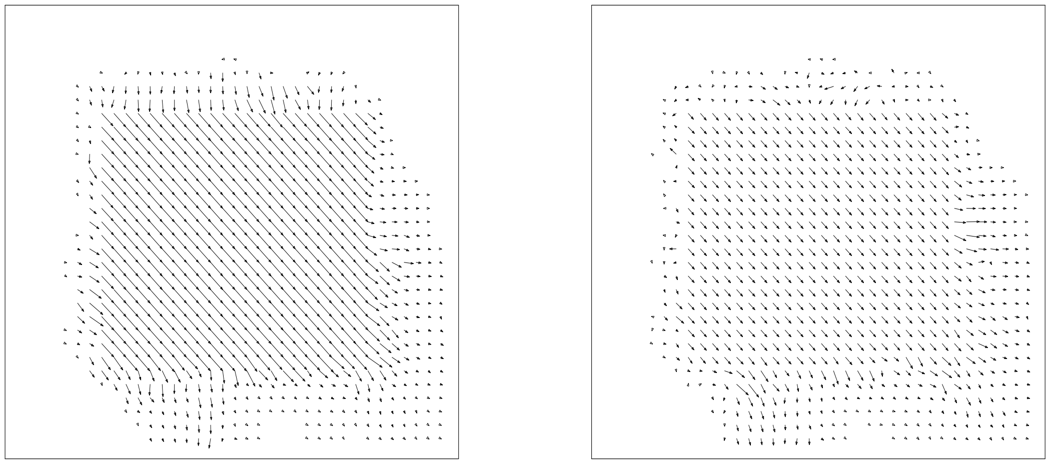

(a) (b)

Figure 4.11: Estimated: (a) velocity $\hat{\mathbf{v}}(2)$ (scaled by 0.5), and (b) acceleration $\hat{\mathbf{a}}$ using algorithm $\mathcal{B}$ at field #2 of test image 5 with $\mathbf{p}(2) = [4 \;\; 4 \;\; 1 \;\; 1]^T$.

Figure 4.12 for $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$ ($N = 5$) and in Figure 4.13 for $\mathcal{I}_t = \{-1, 0, 1\}$ ($N = 3$).

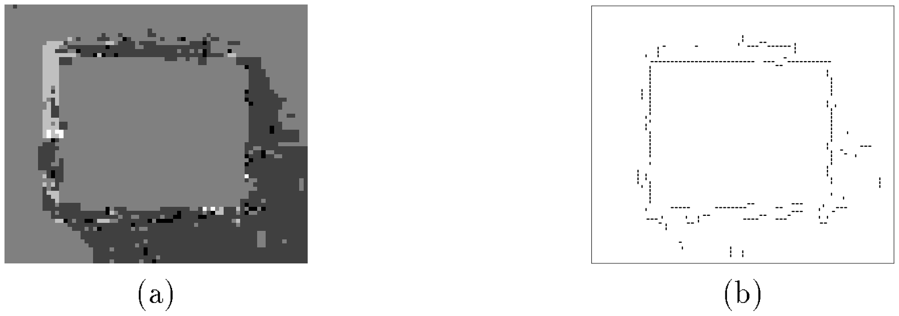


Figure 4.12: Estimated: (a) occlusion field $\hat{o}(2)$, and (b) line field $\hat{l}(2)$ at field #2 of test image 5 with $\mathbf{p}(2) = [4 \;\; 4 \;\; 1 \;\; 1]^T$ and $\mathcal{I}_t = \{-2, -1, 0, 1, 2\}$.

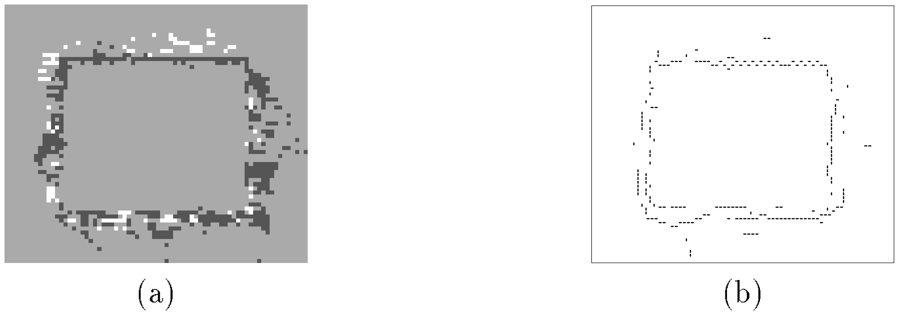


Figure 4.13: Estimated: (a) occlusion field $\hat{o}(2)$, and (b) line field $\hat{l}(2)$ at field #2 of test image 5 with $\mathbf{p}(2) = [4 \;\; 4 \;\; 1 \;\; 1]^T$ and $\mathcal{I}_t = \{-1, 0, 1\}$.

Similar observations as before can be made; processing of occlusion areas and motion discontinuities helps to decrease the MSE of motion estimates, especially around the border of the moving rectangle (Table 4.3). Also, occluded areas are better tracked with $N = 5$ then with $N = 3$. Note that the covered area in Figure 4.12a is larger than the exposed area due to the presence of acceleration in the diagonal direction, and hence the detected occluded areas are consistent with the motion of the moving rectangle.

4.3.3 Results for natural sequences

Algorithm $\mathcal{B}$ has been also tested on test images 3 and 4 (Figures 3.5 and 3.6) with $(\lambda_o, \lambda_l) = (1, 2)$. The results are illustrated in this section along with a comparison with the results obtained by algorithm $\mathcal{A}$.

The computed intensity edges at fields #26, 42, and 70 of test image 3 are shown in Figure 4.14. These fields are used in algorithm $\mathcal{B}$ to penalize the introduction of line elements at non-edge sites.

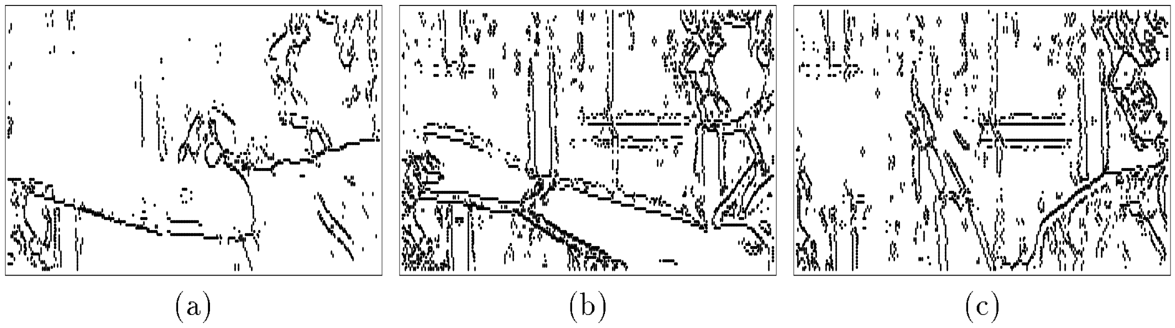


Figure 4.14: Computed intensity edges: (a) $e(26)$; (b) $e(42)$; (c) $e(70)$ at fields #30, 42, and 70 of test image 3, respectively.

Figure 4.15 illustrates the $PSNR$ of 28 reconstructed fields from test image 3 using motion obtained from algorithm $\mathcal{A}$ (full line) and motion and occlusions obtained from algorithm $\mathcal{B}$ (dashed line).

An average increase of +1.27 dB in the $PSNR$ has been achieved by using algorithm $\mathcal{B}$ instead of algorithm $\mathcal{A}$. The $PSNR$ boost is most apparent in fields containing accelerated motion of the hand or arm. This can be easily explained by the fact that accelerated motion generates more occluded regions than linear motion, and hence the detection of these regions (in Algorithm $\mathcal{B}$) in the presence of acceleration is helpful in obtaining better motion estimates especially around motion discontinuities.

The estimated piecewise-continuous velocities at fields #26 and 42 are shown in Figures 4.16a and 4.17a, respectively, along with the corresponding estimated line fields. The same fields derived by algorithm $\mathcal{A}$ are shown in Figures 4.16b and 4.17b. Note that the estimation of motion discontinuities has allowed to disable the motion

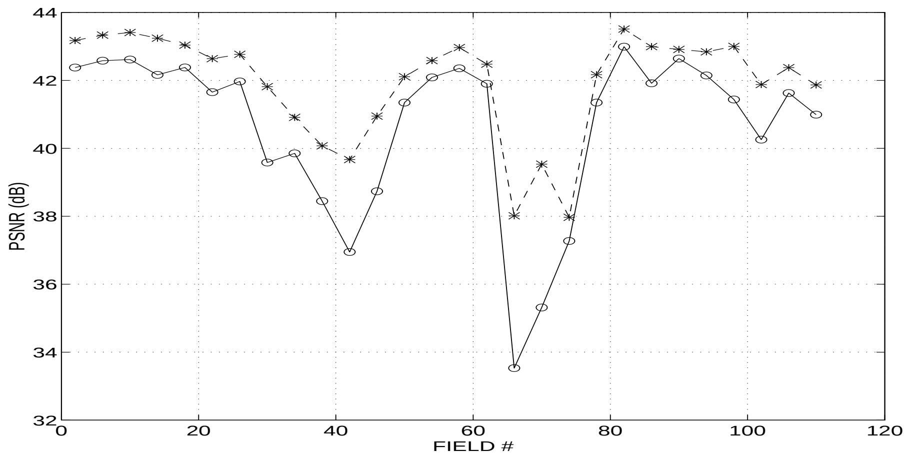


Figure 4.15: Comparison of the $PSNR$ for reconstructed fields from test image 3 using algorithm $\mathcal{A}$ (full line) and algorithm $\mathcal{B}$ (dashed line).

smoothness constraint around these discontinuities (i.e., between moving/stationary and occluded regions), and hence to produce piecewise-continuous motion estimates that are closer to the true underlying motion.

The occlusion and motion discontinuity fields along with the reconstructed and error images (using algorithms $\mathcal{A}$ and $\mathcal{B}$) for fields #26, 42, and 70 are shown in Figures 4.18, 4.19, and 4.20 respectively.

Note that the occlusion and motion discontinuity estimates seem consistent with the motion in the sequence. This can be confirmed by the reconstructed fields and their respective error images in Figures 4.18, 4.19, and 4.20. For instance, one can notice that the interpolated field #40 obtained using the motion and occlusion estimates obtained by algorithm $\mathcal{B}$ (Figure 4.19e) is closer to the true field (Figure 4.19a) than the interpolated field obtained using only the motion estimate obtained by algorithm $\mathcal{A}$ (Figure 4.19d). This difference is most noticeable in the region of the left arm. The same kind of behavior can be noticed in Figure 4.20 where the contours of the moving right hand are better reconstructed when algorithm $\mathcal{B}$ is used.

The same experiments have been run for test image 4. The computed intensity edges at fields #6, 14, and 19 of test image 4 are shown in Figure 4.21.

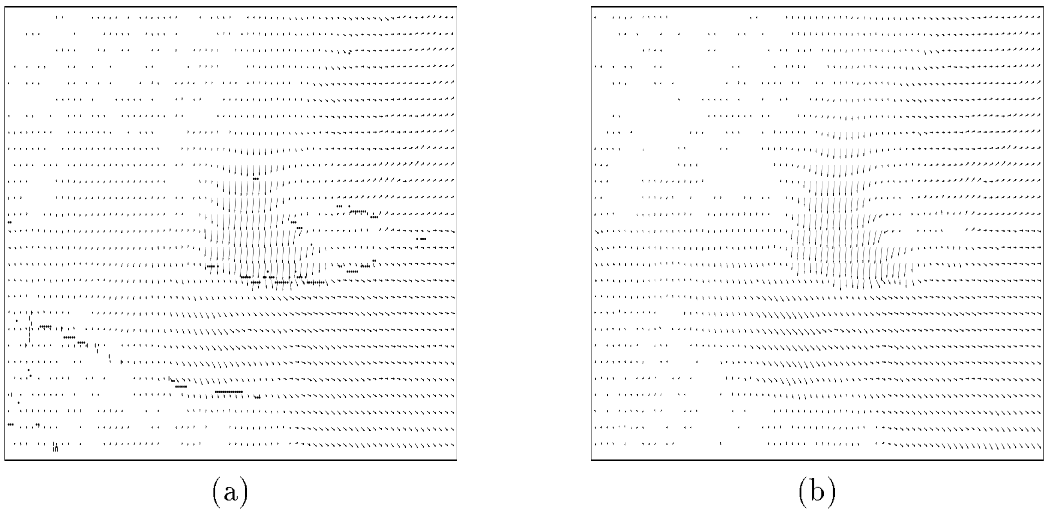


Figure 4.16: Estimated (a) velocity $\hat{\mathbf{v}}(26)$ and line fields $\hat{l}(26)$ using algorithm $\mathcal{B}$; (b) velocity $\hat{\mathbf{v}}(26)$ using algorithm $\mathcal{A}$ at field #26 of test image 3.

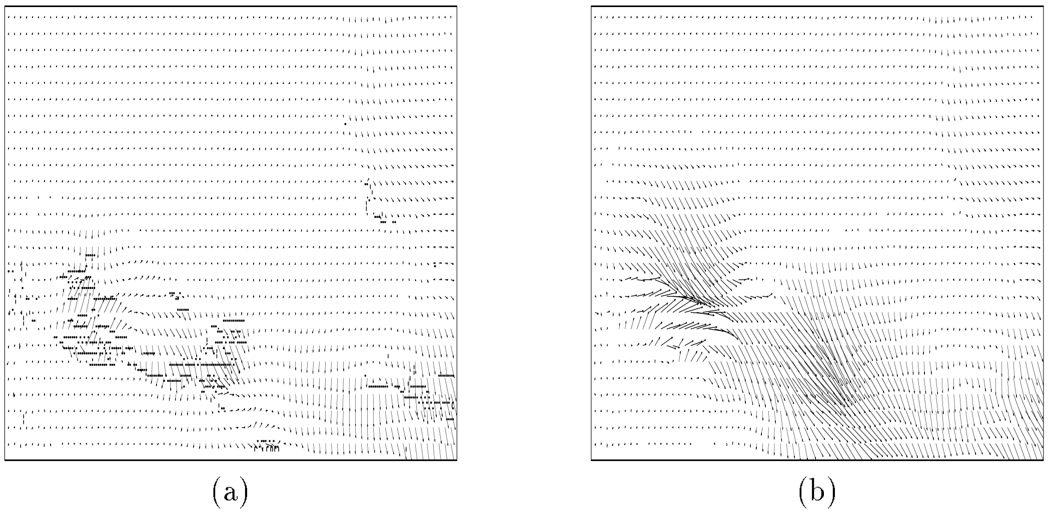


Figure 4.17: Estimated: (a) velocity $\hat{\mathbf{v}}(42)$ and line fields $\hat{l}(42)$ using algorithm $\mathcal{B}$; (b) velocity $\hat{\mathbf{v}}(42)$ using algorithm $\mathcal{A}$ at field #42 of test image 3.

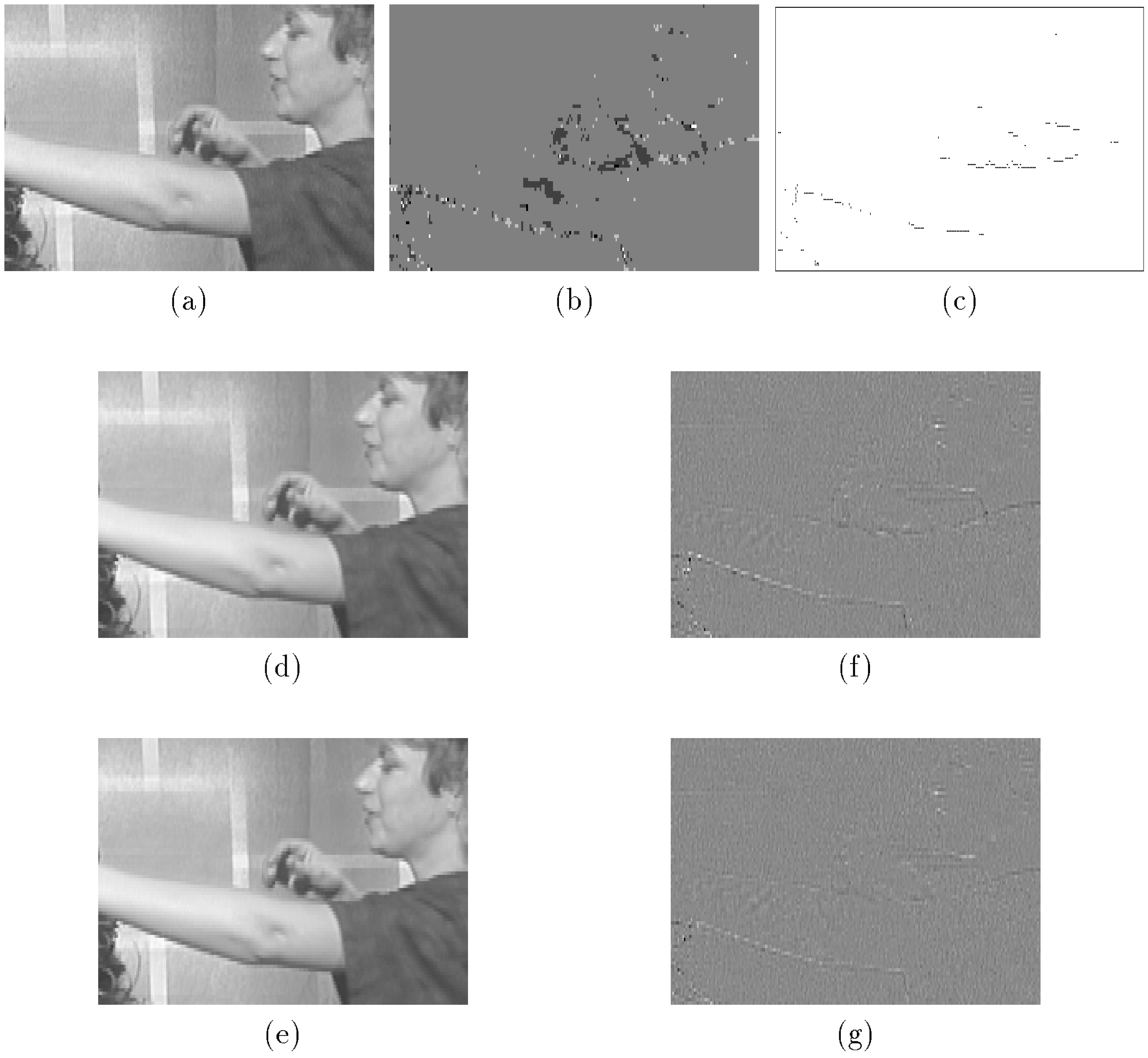


Figure 4.18: Estimation (a) area of field #26 of test image 3; (b) occlusion field $\hat{o}(26)$; (c) line field $\hat{l}(26)$; reconstructed field using: (d) algorithm $\mathcal{A}$ ($PSNR = 41.96$ dB); (e) algorithm $\mathcal{B}$ ($PSNR = 42.76$ dB) with their respective error images (magnified by 2) in (f) and (g).

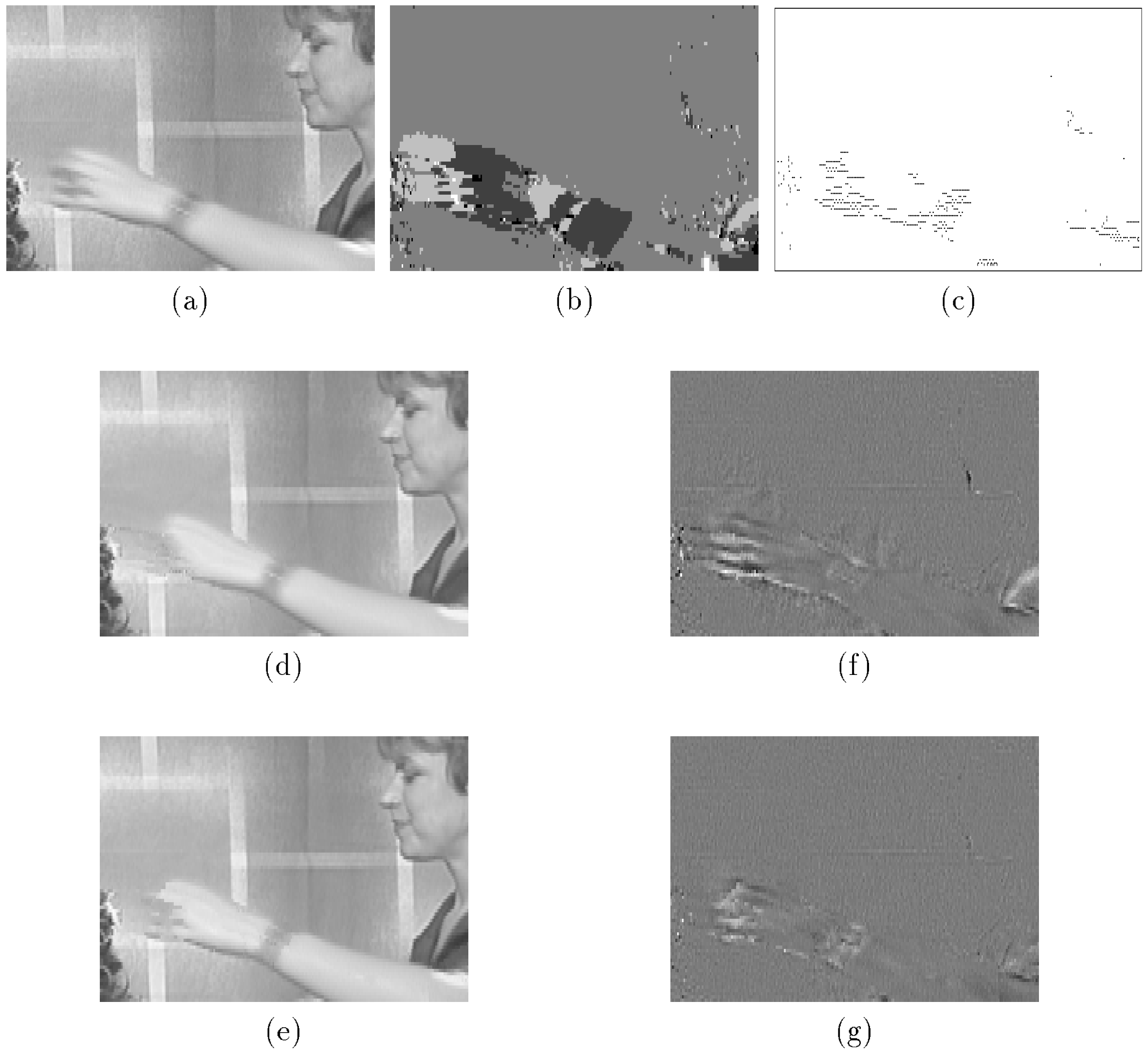


Figure 4.19: Estimation (a) area of field #42 of test image 3; (b) occlusion field $\hat{o}(42)$; (c) line field $\hat{l}(42)$; reconstructed field using: (d) algorithm $\mathcal{A}$ ($PSNR = 36.94$ dB); (e) algorithm $\mathcal{B}$ ($PSNR = 39.67$ dB) with their respective error images (magnified by 2) in (f) and (g).

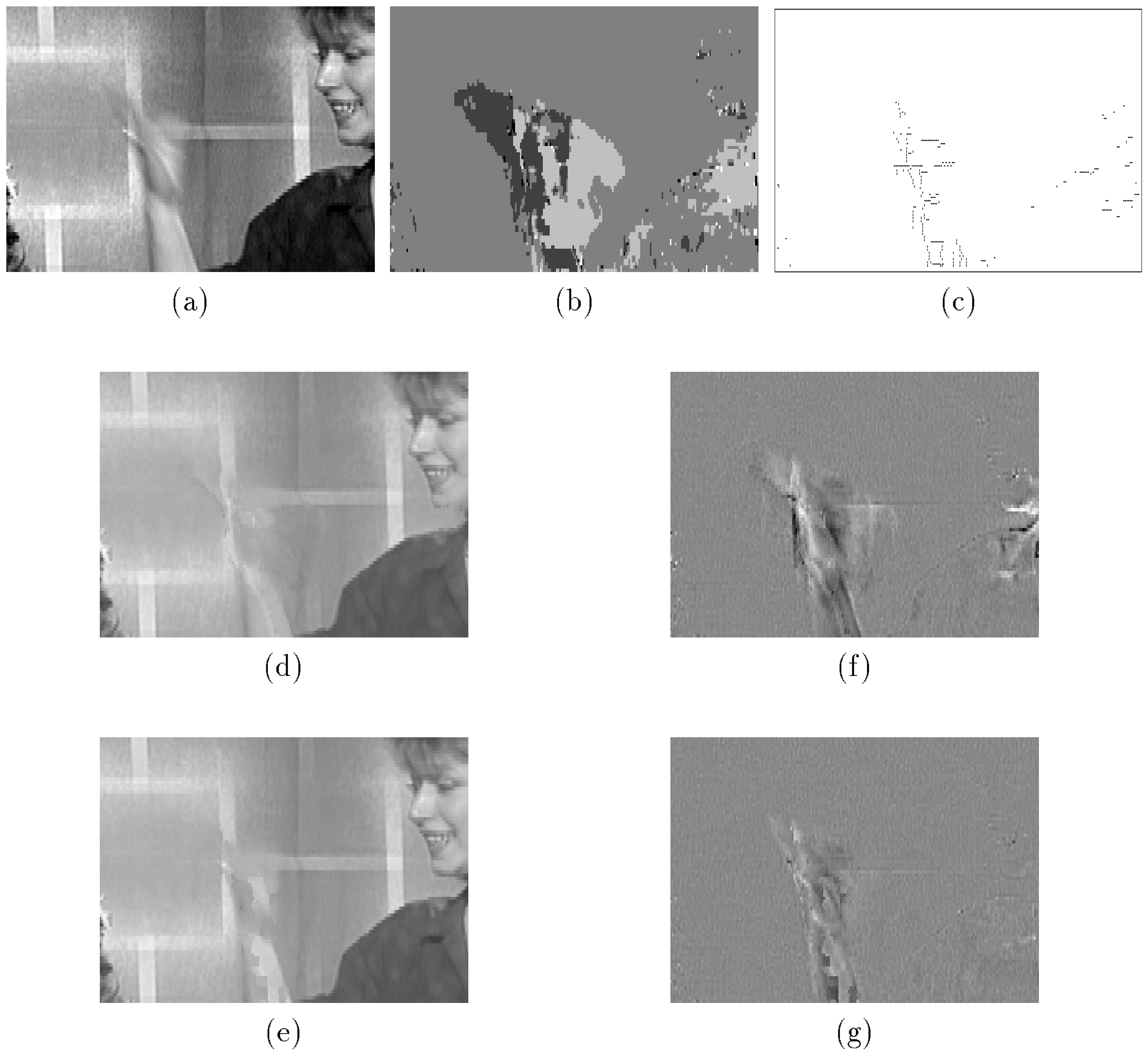


Figure 4.20: Estimation (a) area of field #70 of test image 3; (b) occlusion field $\hat{o}(70)$; (c) line field $\hat{l}(70)$; reconstructed field using: (d) algorithm $\mathcal{A}$ ($PSNR = 35.31$ dB); (e) algorithm $\mathcal{B}$ ($PSNR = 39.53$ dB) with their respective error images (magnified by 2) in (f) and (g).

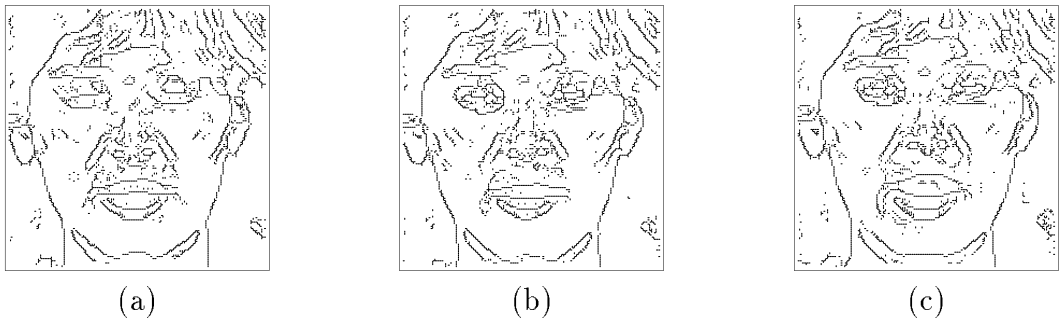


Figure 4.21: Computed intensity edges: (a) $e(6)$; (b) $e(14)$; (c) $e(19)$ at fields #6, 14, and 19 of test image 4, respectively.

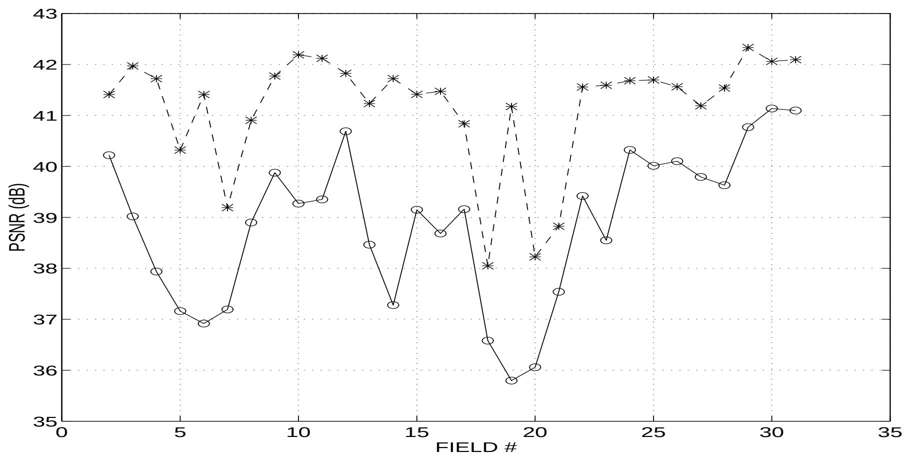


Figure 4.22: Comparison of the $PSNR$ for reconstructed fields from test image 4 using algorithm $\mathcal{A}$ (full line) and algorithm $\mathcal{B}$ (dashed line).

The resulting $PSNR$ curves are reported in Figure 4.22 where an average increase of +2.3 dB in the $PSNR$ has been achieved by using algorithm $\mathcal{B}$ instead of algorithm $\mathcal{A}$.

The occlusion and motion discontinuity fields obtained for fields #5, 6, 8, 14, 19, and 26 are shown in Figures 4.23, 4.24, 4.25, 4.26, 4.27, and 4.28 respectively. Note that most of the occluded regions are concentrated in the area of the eyes and the mouth. Also, the estimated motion discontinuities match well the moving contours of the eyes and the mouth.

The reconstructed fields and their respective error images (using algorithms $\mathcal{A}$

and $\mathcal{B}$) at fields #6, 14, and 19 are shown in Figures 4.24d-g, 4.26d-g, and 4.27d-g respectively. From these results, one can conclude that the processing of occlusions and motion discontinuities has helped to eliminate most of the interpolation errors that were present in the regions of the eyes and the mouth prior to occlusion processing (i.e., algorithm $\mathcal{A}$). This improvement is also visible in the reconstructed fields, especially in Figure 4.27, where a comparison of the two reconstructed fields #19 using algorithms $\mathcal{A}$ and $\mathcal{B}$ (shown in Figures 4.27d and 4.27e, respectively) with the original field (Figure 4.27a) illustrates the difference, mainly in the region of the mouth.

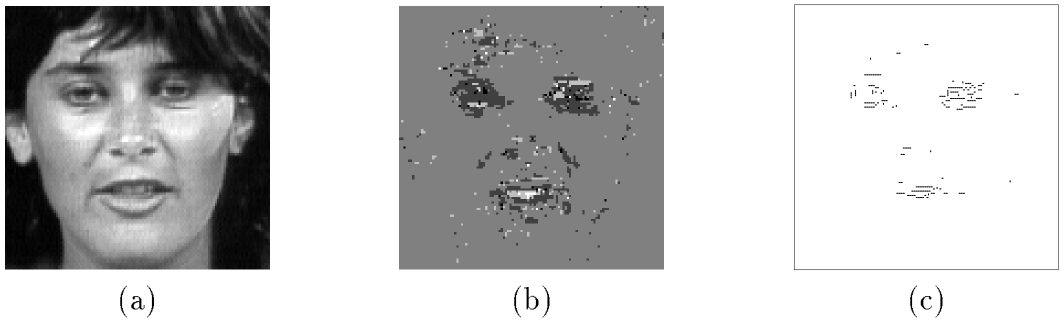


Figure 4.23: Estimation (a) area of field #5 of test image 4; (b) occlusion field $\hat{o}(5)$; (c) line field $\hat{l}(5)$.

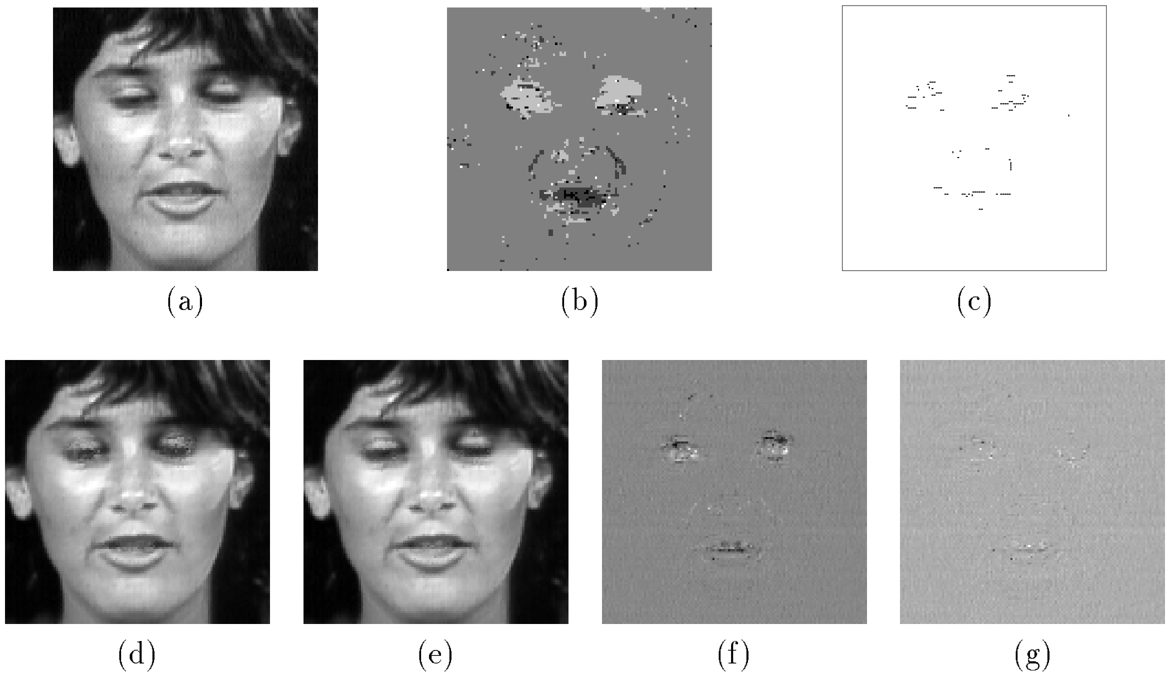


Figure 4.24: Estimation (a) area of field #6 of test image 4; (b) occlusion field $\hat{o}(6)$; (c) line field $\hat{l}(6)$; reconstructed field using: (d) algorithm $\mathcal{A}$ ($PSNR = 36.91$ dB); (e) algorithm $\mathcal{B}$ ($PSNR = 41.41$ dB) with their respective error images (magnified by 2) in (f) and (g).

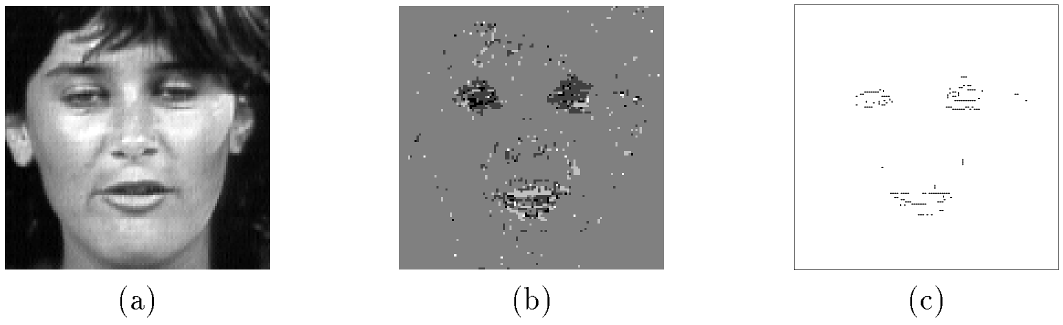


Figure 4.25: Estimation (a) area of field #8 of test image 4; (b) occlusion field $\hat{o}(8)$; (c) line field $\hat{l}(8)$.

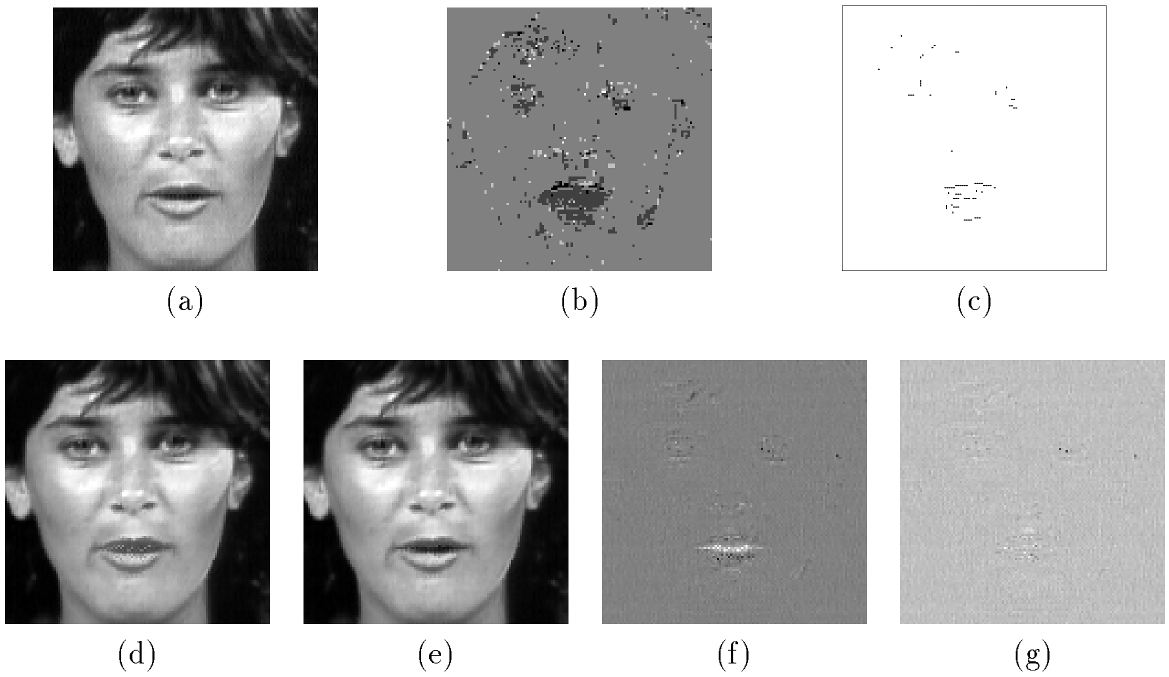


Figure 4.26: Estimation (a) area of field #14 of test image 4; (b) occlusion field $\hat{o}(14)$; (c) line field $\hat{l}(14)$; reconstructed field using: (d) algorithm $\mathcal{A}$ ($PSNR = 37.27$ dB); (e) algorithm $\mathcal{B}$ ($PSNR = 41.72$ dB) with their respective error images (magnified by 2) in (f) and (g).

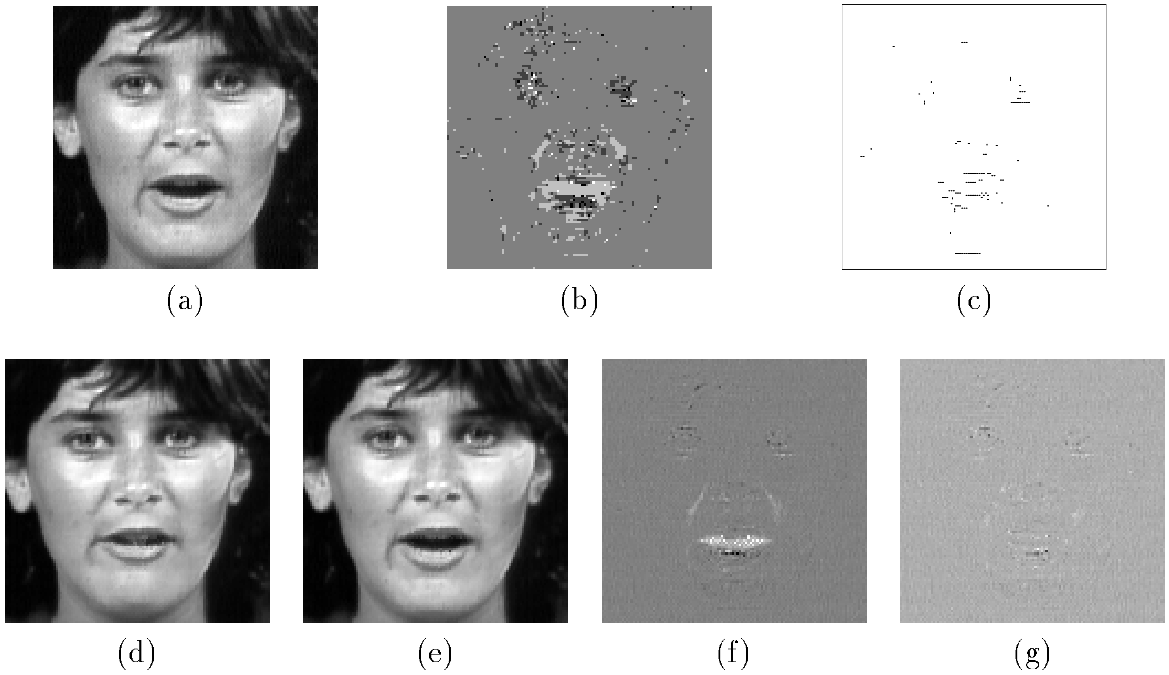


Figure 4.27: Estimation (a) area of field #19 of test image 4; (b) occlusion field $\hat{o}(19)$; (c) line field $\hat{l}(19)$; reconstructed field using: (d) algorithm $\mathcal{A}$ ($PSNR = 35.79$ dB); (e) algorithm $\mathcal{B}$ ($PSNR = 41.17$ dB) with their respective error images (magnified by 2) in (f) and (g).

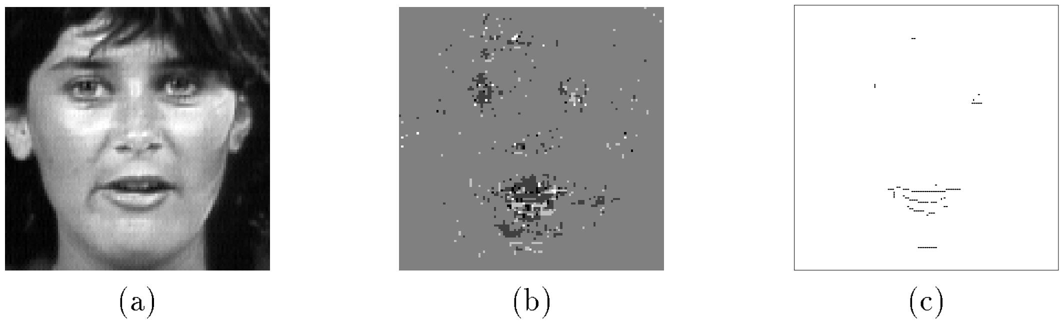


Figure 4.28: Estimation (a) area of field #26 of test image 4; (b) occlusion field $\hat{o}(26)$; (c) line field $\hat{l}(26)$.

Chapter 5

Conclusions

5.1 Summary

Different types of motion-compensated processing of time-varying images, such as predictive coding and standards conversion, require the availability of 2-D motion estimates. In this thesis, estimation of dense motion trajectories with acceleration has been investigated. Unlike in most of the existing motion estimation algorithms that assume a linear trajectory model over two fields in an image sequence, the proposed method assumes a quadratic trajectory model defined over longer temporal support. Hence, two motion field estimates of instantaneous velocities and accelerations are generated to describe quadratic trajectories instead of one displacement field that describes linear trajectories in an image.

Due to the ill-posed nature of motion estimation, the algorithm for the estimation of dense accelerated motion fields has been formulated using *regularization.* The objective function has been derived using Gibbs-Markov models linked together by the *Maximum A Posteriori* (MAP) probability criterion. It consists of a structural model that follows directly from the constant intensity assumption along motion trajectories, and of an *a priori* motion trajectory model that captures the desired smoothness property of motion fields. Energies resulting from these models have been combined linearly using regularization parameter λ_p that plays a vital role in weighting the importance of the two models. Optimization of the objective function has been carried out using a deterministic relaxation algorithm implemented over a pyramid of resolutions. The importance of multiresolution methods in the estimation of fast motion and efficient localization of a near global optimum has also been addressed.

The motion estimation algorithm has been tested successfully on progressive and interlaced sequences with synthetic motion parameters of 1/4 pixel accuracy. The MSE measure has been used to measure the validity of motion estimates and to select certain parameters in the algorithm such as the regularization parameter λ_p, and the number of images used in the estimation. Also, plots of some estimated trajectories have been compared with their respective true trajectories in order to illustrate the difference in trajectory tracking when the quadratic trajectory model is used instead of the linear model.

The usefulness of motion trajectories with acceleration for motion-based processing has been investigated on natural sequences. The estimated trajectories have been applied to a motion-compensated interpolation scheme for the case of 4:1 subsampling. A comparison of the $PSNR$ for the reconstructed images using linear and quadratic motion trajectory models over 5 fields and linear trajectory model over 2 fields was carried out. Similarly to [35], it was concluded that in images containing acceleration, the knowledge of this acceleration permits a substantial reduction of the reconstruction error. Also, subjectively the quadratic motion trajectory model has resulted in a remarkable improvement of the reconstructed image quality. This observation is particularly true for image sequences containing "talking heads" where eyes and mouth do exhibit acceleration. Occasionally, the difference between the two models has amounted to the mouth being closed, whereas in the original image it was open. From the transmission point of view, this improvement comes at the cost of additional bit rate allocated to acceleration parameters. It is not clear at this point whether this increase can be compensated by the reduced prediction residual.

The motion estimation algorithm was finally extended in order to detect occlusion areas while estimating motion parameters. This feature is vital in motion-compensated interpolation applications, where it is imperative that estimated motion parameters near motion discontinuities be of high quality and that occlusions be properly handled. A new multiple-term objective function including an occlusion model and a motion discontinuity model (in addition to the structural model and *a priori* motion model) has been derived. The occlusion model, in the case of estimation from 5 images, has been presented. This model favors the creation of clusters of occlusion tags near motion discontinuities. On the other hand, the motion discontinuity model assigns a high penalty whenever a motion discontinuity does not match an intensity edge. It also controls the formation of straight lines, corners, and intersections. The minimization of the new objective function, performed in an interleaved fashion, results in piecewise-continuous motion fields that correspond better to real TV images than the globally-continuous motion fields generated in absence of the occlusion model.

The use of multiframe processing in the proposed motion estimation algorithm has been expected to be beneficial from the point of view of improved identification of occlusion areas and motion discontinuities. This was confirmed by comparing occlusion and line fields estimated using 3 and 5 images from sequences with synthetic motion. However, the estimation of occlusion fields in the motion estimation algorithm requires more computational time. Also, the occlusion information has to be transmitted with the motion information in interpolative coding schemes resulting in some extra bits to be transmitted. These disadvantages, however, have to be weighted against a significant increase in the quality of reconstructed sequences at the receiver as was discussed in the section on experimental results for natural sequences.

The work reported in this thesis is of exploratory nature. We were interested in finding out what possible improvements could the computation of acceleration and occlusions bring. The proposed algorithm is very complex computationally due to its iterative nature and due to the calculation of derivatives. It is not intended for real-time implementation. However, we hope that the demonstrated improvements will eventually find their way to real-time implementations through some simplified algorithms.

5.2 Contributions

This thesis has contributed to the theory of 2-D motion estimation. The major contributions of this work can be summarized as follows:

1. Modeling of motion trajectories with acceleration.

2. Estimation of motion and occlusions over multiple images.

3. Application of motion trajectories with acceleration to motion-compensated temporal interpolation in a multiple-frame scenario. It has been shown that for images containing acceleration, such as "talking heads", the quadratic motion model permits a substantial reduction of the reconstructed error when compared with the ubiquitous linear model.

4. Application of occlusion processing in the context of motion-compensated temporal interpolation. It has been demonstrated that a further improvement, especially around motion discontinuities, is observed in reconstructed images when occlusions are accounted for.

5.3 Open questions

5.3.1 Regularization parameters

The regularization parameters λ_p, λ_o, and λ_l have been chosen empirically. Optimal estimation of these parameters remains to be a challenging task, especially when estimating unobservables such as motion.

5.3.2 Hierarchical processing

In the hierarchical processing, a “constant-width” pyramid for images and a regular pyramid for motion fields have been considered. The likelihood of convergence to the global optimum may be improved by considering also a regular pyramid for images. This will allow to spread the displacement vector updates proportionally over image resolutions and may result eventually in a better optimum. However, the use of regular image pyramids will result in data loss (due to subsampling), and hence may affect the performance of the motion estimation algorithm (because of the derivative computation). Moreover, the smoothing by Gaussian filters destroys the contours in the image, and hence results in erroneous occlusion estimation at the lower resolution levels of the pyramid. Nyquist-like filters that do not unnecessarily oversmooth the data may be worth considering in the generation of the pyramid of image resolutions.

5.3.3 Rate-constrained motion estimation

This thesis has demonstrated the importance of the estimation of accelerated motion and occlusions in reducing the reconstruction error in an interpolative coding scheme. With such an improvement, the motion-compensated interpolation error is very small. This error may be transmitted or not, depending on the target quality. It remains to be studied whether the reduction of the transmitted residual (reconstruction error) or the improvement in quality of reconstructed images compensate for the increased bit rate needed to transmit the acceleration and/or occlusion information. This problem, called “rate-constrained motion estimation”, has not been studied in-depth yet, except for very simple cases. Perhaps, acceleration is worth considering for post-processing in video conferencing and videophone applications where temporal subsampling is often used.

Appendix A

Markov Random Fields and the Gibbs Distribution

The main focus here are 2-D random fields defined over a finite $N_1 \times N_2$ rectangular lattice of points (pixels) defined as: $\Lambda = \{(i,j) : 1 \leq i \leq N_1, 1 \leq j \leq N_2\}$. The concepts of neighborhood and cliques are essential in the definition of the Gibbs distribution. A neighborhood system on lattice Λ is defined as follows:

Definition 1: A collection of subsets of Λ described as:

$$\eta = \{\eta_{ij} : (i,j) \in \Lambda, \eta_{ij} \subseteq \Lambda\} \tag{A.1}$$

is a neighborhood system on Λ if and only if

1. $(i,j) \notin \eta_{ij}$, and
2. if $(k,l) \in \eta_{ij} \Rightarrow (i,j) \in \eta_{kl} \ \forall \ (i,j) \in \Lambda$.

A Markov Random Field (MRF) with respect to the neighborhood system η defined over the lattice Λ is then defined as follows:

Definition 2: Let η be a neighborhood system defined over lattice Λ. A random field $X = \{X_{ij}\}$ defined over lattice Λ is a *Markov random field* with respect to the neighborhood system η if and only if:

$$P\left(X_{ij}=x_{ij} \mid X_{kl}=x_{kl},(k,l)\in\Lambda,(k,l)\neq(i,j)\right)=P\left(X_{ij}=x_{ij} \mid X_{kl}=x_{kl},(k,l)\in\eta_{ij}\right) \tag{A.2}$$

for all $(i,j)\in\Lambda$, and $P(X=x)>0\ \forall x$.
Note that capital letters are used to denote random variables and random fields, and lower case letters to denote specific realizations.

The first-order neighborhood system η^1 is the most commonly used in image modeling. It consists of the closest four neighbors of each pixel, and is known as the nearest-neighbor model. The second-order neighborhood system $\eta^2 = \{\eta^2_{ij}\}$ is such that η^2_{ij} consists of the eight pixels neighboring (i,j). In general, the the m^{th} order neighborhood system η^m contains all sites of systems of order up to $m-1$ (i.e., $\eta^m = \{\eta^k : k < m\}$). The "cliques" associated with a lattice-neighborhood pair (Λ,η) are defined as follows [14]:
Definition 3: A *clique* of the pair (Λ,η), denoted by c, is a subset of Λ such that:

1. c consists of a single pixel, or
2. for $(i,j)\neq(k,l)$, $(i,j)\in c$, and $(k,l)\in c \Rightarrow (i,j)\in\eta_{kl}$.

The collection of all cliques of (Λ,η) is denoted by $\mathcal{C}$. The types of cliques associated with η^1 and η^2 are shown in Figure A.1.

It is known that the usual characterization of a MRF through initial and transitional probabilities is complex. On the other hand, from the *Hammersley-Clifford theorem* [3] it is known that a random field has Markovian properties *if and only if* it is governed by a Gibbs distribution (GD). The GD is defined in the following manner[14]:
Definition 4: Let η be a neighborhood system defined over the finite lattice Λ. A random field $X=\{X_{ij}\}$ defined on Λ has a Gibbs distribution or equivalently is a Gibbs Random Field (GRF) with respect to η if and only if its joint distribution is

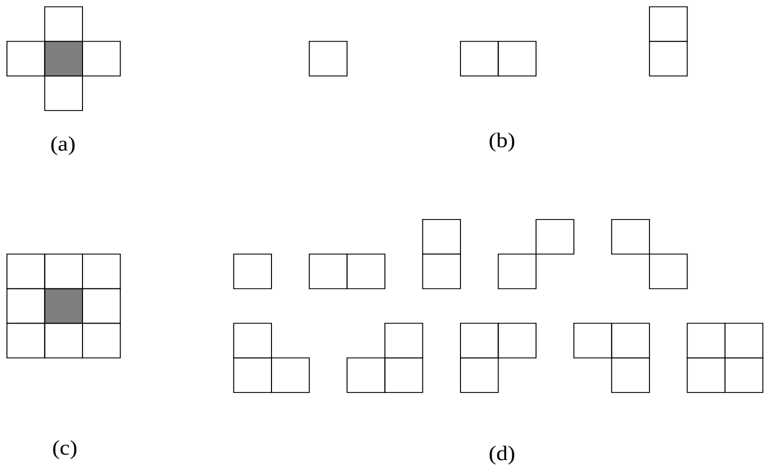


Figure A.1: Neighborhood system η^1 (a) and associated cliques (b); Neighborhood system η^2 (c), and associated cliques (d).

of the form:

$$P(X = x) = \frac{1}{Z} e^{-\frac{U(x)}{\beta}} \tag{A.3}$$

where

$$U(x) = \sum_{c \in \mathcal{C}} V_c(x) \tag{A.4}$$

is the energy function and $V_c(x)$ is the potential associated with clique c. The *partition function* $Z = \sum_x e^{-U(x)}$ is simply a normalizing constant, and β is another constant called the *natural temperature.* The only condition on the otherwise totally arbitrary clique potential $V_c(x)$ is that it depends only on the pixel values in clique c. The joint distribution in (A.3) has a physical interpretation: the smaller $U(x)$, the energy of the realization x, the more likely that realization.

The GD is basically an exponential distribution. However, by choosing the clique potential function $V_c(x)$ properly, a wide variety of distributions, both for discrete and continuous random fields, can be formulated as GD (i.e. binomial, Poisson, and Gaussian random fileds). Unlike the MRF characterization, the GD characterization is free from consistency problems and in some applications provides a more workable spatial model.

Appendix B

Cubic Convolution Interpolation

Assuming 1-D notation, let w be an input signal defined over a lattice Λ. The interpolated signal $\tilde{w}$ defined over $\mathcal{R}$ can be obtained by the following convolution:

$$\tilde{w}(x) = \sum_{y \in \Lambda} w(y) u(x-y), \qquad x \in \mathcal{R}, \tag{B.1}$$

where u is the impulse response of a low pass filter, known as the interpolation kernel to be defined. Note that also, due to the linearity of the convolution, the derivative of $\tilde{w}$ can be obtained as follows:

$$\frac{\partial \tilde{w}(x)}{\partial x} = \sum_{y \in \Lambda} w(y) \frac{\partial u(x-y)}{\partial x}, \qquad x \in \mathcal{R}. \tag{B.2}$$

Keys [25] has proposed a cubic convolution kernel $u(x)$ for one-dimensional problem which converts the discrete data w into a continuous function $\tilde{w}$ by the convolution operation in (B.1).

The cubic convolution algorithm, normally requires that the interpolation kernel be continuous, and possess a continuous first-order derivative. Otherwise the interpolated function will have sharp edges at sampling points which is an undesirable effect, especially when interpolating intensities in an image. The cubic convolution kernel introduced by Keys:

$$u(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1 & 0 < |x| < 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2 & 1 < |x| < 2 \\ 0 & 2 < |x| \end{cases} \tag{B.3}$$

is symmetric, continuous, and has a continuous first derivative as shown in Figure B.1. Moreover, it is zero for all non-zero integers, and one when its argument is

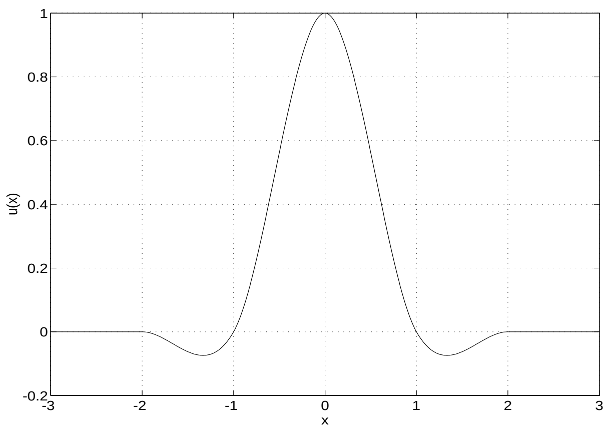


Figure B.1: Impulse response $u(x)$ of cubic interpolator proposed by Keys.

zero (this condition has an important computational significance, namely, that the interpolation coefficients become simply the sampled data points).

Bibliography

[1] J. K. Aggarwal and N. Nandhakumar, "On the Computation of Motion from Sequences of Images -A review," *Proceedings of the IEEE*, vol. 76, pp. 917–935, August 1988.

[2] M. J. Bertero, T. A. Poggio, and V. Torre, "Ill-Posed Problem in Early Vision," *Proceedings of the IEEE*, vol. 76, pp. 869–889, August 1988.

[3] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. Roy. Statist. Soc.*, vol. B 36, pp. 192–236, 1974.

[4] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. R. Statist. Soc.*, vol. 48, pp. 259–279, 1986.

[5] P. J. Burt and *et al.*, "Multi-Resolution Flow-Through Motion Analysis," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 246–252, June 1983.

[6] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, November 1986.

[7] R. Depommier, "Estimation du Mouvement Considérant les Phénomènes d'Occlusion pour le Codage Interpolatif des Séquences d'Images," Master's thesis, INRS-Télécommunications, December 1990.

[8] R. Depommier and E. Dubois, "Motion Compensated Temporal Prediction for Interlaced Image Sequences," *IEEE Workshop on Visual Signal Processing and Communication*, pp. 264–269, Sept. 1992.

[9] R. Depommier and E. Dubois, "Motion Estimation with Detection of Occlusion Areas," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 269–272, March 1992.

[10] E. Dubois, "The Sampling and Reconstruction of Time-Varying Imagery with Application in Video Systems," *Proceedings of the IEEE*, vol. 73, pp. 502–522, 1985.

[11] E. Dubois, "Motion-Compensated Filtering of Time-Varying Images," *Multidimensional Systems and Signal Processing*, vol. 3, pp. 211–239, 1992.

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

[13] W. Enkelman, "Investigations of Multigrid Algorithms for the Estimation of Optical Flow Fields in Image Sequences," *Computer Vision, Graphics, and Image Processing*, vol. 43, pp. 150–177, 1988.

[14] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 721–741, November 1984.

[15] 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.

[16] F. Heitz and P. Bouthemy, "Motion Estimation and Segmentation Using a Global Bayesian Approach," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2305–2308, 1990.

[17] F. Heitz, P. Perez, and P. Bouthemy, "Parallel visual motion analysis Using Multiscale Markov Random Fields," *Proc. IEEE Workshop on Visual Motion*, pp. 30–35, October 1991.

[18] E. C. Hildreth, "Computations Underlying the Measurement of Visual Motion," *Artificial Intelligence*, vol. 23, pp. 309–354, 1984.

[19] B. K. P. Horn, *Robot Vision*, ch. 12-Motion Field and Optical Flow, pp. 278–298. The MIT Press, 1986.

[20] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[21] J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing Motion Using Analog and Binary Resistive Networks," *Computer*, vol. 21, pp. 52–63, March 1988.

[22] L. Jacobson and H. Wechsler, "Derivation of Optical Flow Using a Spatiotemporal-Frequency Approach," *Computer Vision, Graphics, and Image Processing*, vol. 38, pp. 29–65, 1987.

[23] A. K. Jain, *Fundamentals of Digital Image Processing.* Pentice Hall, 1989.

[24] J. R. Jain and A. K. Jain, "Displacement measurement and its application in Interframe Image Coding," *IEEE Transactions on Communications*, vol. COM-29, pp. 1799–1808, December 1981.

[25] R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE trans. Acoustic, Speech, and Signal Processing*, vol. ASSP-29, pp. 1153–1160, December 1981.

[26] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding For Video Conferencing," *Conf. Rec., Nat. Telecomm. Conf.*, pp. G5.3.1–G5.3.5, 1981.

[27] J. Konrad, *Bayesian Estimation of Motion Fields from Image Sequences.* PhD thesis, Department of Electrical Engineering, McGill University, 1989.

[28] J. Konrad and E. Dubois, "Estimation of Image Motion Fields: Bayesian Formulation and Stochastic Solution," *Proc. IEEE Int. Conf. on Acoustic, Speech, and Signal Processing*, pp. 1072–1074, 1988.

[29] J. Konrad and E. Dubois, “Multigrid Bayesian Estimation of Image Motion Field Using Stochastic Relaxation,” *Proc. IEEE Int. Conf. Computer Vision*, pp. 354–362, Dec. 1988.

[30] J. Konrad and E. Dubois, “Comparison of Stochastic and Deterministic Solution Methods in Bayesian Estimation of 2-D Motion,” *Image and Vision Computing*, vol. 9, pp. 215–228, Aug. 1991.

[31] J. Konrad and E. Dubois, “Bayesian Estimation of Motion Vector Fields,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-14, pp. 910–927, Sept. 1992.

[32] H. Nagel and W. Enkelman, “An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 565–593, September 1986.

[33] A. N. Netravali and J. D. Robbins, “Motion-Compensated Television Coding: Part I,” *The Bell System Technical Journal*, vol. 58, pp. 631–670, March 1979.

[34] A. Nguyen and E. Dubois, “Adaptive Interlaced to Progressive Image Conversion,” in *Proc. Int. Workshop on HDTV*, pp. 749–756, Nov. 1992.

[35] A. J. Patti, M. I. Sezan, and A. M. Tekalp, “Digital Video Standards Conversion in the Presence of Accelerated Motion,” *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 225–229, April 1994.

[36] R. Srinivasan and K. R. Rao, “Predictive Coding Based on Efficient Motion Estimation,” *IEEE Transactions on Communications*, vol. COM-33, pp. 888–896, August 1985.

[37] K. M. Uz, M. Vetterli, and D. LeGall, “A Multiresolution Approach to Motion Estimation and Interpolation with Application to Coding of Digital HDTV,” *Proc. ISCAS-90, New Orleans*, pp. 1298–1301, May 1990.

[38] K. M. Uz, M. Vetterli, and D. LeGall, “Interpolative Multiresolution Coding of Advanced Television with Compatible Subchannels,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, pp. 86–99, March 1991.

[39] M. Vetterli and K. M. Uz, “Multiresolution Coding Techniques for Digital Television: A Review,” *Multidimensional Systems and Signal Processing*, vol. 3, pp. 161–187, 1992.