

# Motion Estimation Techniques for Digital TV: A Review and a New Contribution

FRÉDÉRIC DUFAUX, MEMBER, IEEE, AND FABRICE MOSCHENI, STUDENT MEMBER, IEEE

*The key to high performance in image sequence coding lies in an efficient reduction of the temporal redundancies. For this purpose, motion estimation and compensation techniques have been successfully applied. This paper studies motion estimation algorithms in the context of first generation coding techniques commonly used in digital TV. In this framework, estimating the motion in the scene is not an intrinsic goal. Motion estimation should indeed provide good temporal prediction and simultaneously require low overhead information. More specifically, the aim is to minimize globally the bandwidth corresponding to both the prediction error information and the motion parameters. This paper first clarifies the notion of motion, reviews classical motion estimation techniques, and outlines new perspectives. Block matching techniques are shown to be the most appropriate in the framework of first generation coding. To overcome the drawbacks characteristic of most block matching techniques, this paper proposes a new locally adaptive multigrid block matching motion estimation technique. This algorithm has been designed taking into account the above aims. It leads to a robust motion field estimation, precise prediction along moving edges and a decreased amount of side information in uniform areas. Furthermore, the algorithm controls the accuracy of the motion estimation procedure in order to optimally balance the amount of information corresponding to the prediction error and to the motion parameters. Experimental results show that the technique results in greatly enhanced visual quality and significant saving in terms of bit rate when compared to classical block matching techniques.*

## I. INTRODUCTION

Recent advances in digital technology have led to new communication media in which visual information plays the key role. Digital TV, high definition TV, videoconferencing, video-telephony, medical imaging, archiving, virtual reality, and multimedia are some examples of emerging applications.

When compared to audio or text information, video signals require a huge amount of bandwidth. Despite the increase of storage capacity and the development of broadband networks, compression techniques are needed. Image compression techniques rely on two principles: the reduc-

tion of the statistical redundancies in the data and the exploitation of the human visual system. Reviews of image compression techniques can be found in [1]–[5].

In the framework of video coding, the statistical redundancies can be categorized either as spatial or temporal. Due to the different nature of the video signal along the spatial and temporal dimensions, the latter are generally processed separately. Coding techniques which reduce the spatial correlation are referred to as *intraframe* coding, whereas those which tackle the temporal correlation are called *interframe* techniques. Compared to still-image coding, the challenge of image sequence coding lies in an efficient reduction of the temporal correlation.

For the purpose of reducing temporal redundancies, motion estimation techniques have been successfully applied [3], [6]–[8]. They belong to the class of nonlinear predictive coding techniques. In a first stage, the displacement of objects between successive frames is estimated (*motion estimation*). The resulting motion information is then exploited in efficient interframe predictive coding (*motion compensation*). Consequently, the prediction error is transmitted instead of the frame itself. The motion information also has to be transmitted, unless the decoder is able to estimate the motion field. An efficient representation of the motion is thus critical in order to reach high performance in video coding.

The requirements of motion estimation techniques in the framework of video coding can now be explicitly defined. These requirements are mutually contradictory. Motion estimation techniques should on the one hand provide good prediction, but on the other hand have a low overhead information. More specifically, the tradeoff between motion information and prediction error information has to be considered. The purpose of motion estimation techniques is indeed to globally minimize the sum of these two terms. Research effort devoted to this important point is regrettably sparse. The determination of the motion is not the *intrinsic* goal. Nevertheless, a motion field representative of the true motion in the scene is desired in order to avoid artificial discontinuities in the motion compensated prediction and to reduce the transmission cost of the motion information (when it has to be transmitted).

Manuscript received July 1, 1994; revised January 19, 1995.

F. Dufaux is with the Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

F. Moscheni is with the Signal Processing Laboratory, Swiss Federal Institute of Technology, 1015-Lausanne, Switzerland.

IEEE Log Number 9411069.

This paper reviews classical motion estimation techniques and outlines new perspectives. A new technique which overcomes the drawbacks of classical methods is then proposed and described in more detail. This paper considers only the application of motion estimation for the reduction of temporal redundancies and emphasizes the framework of the first generation coding schemes adopted in recent standards for digital TV. Although in digital TV the resulting motion information can also be used for other purposes, such as format conversion or deinterlacing, those topics will not be discussed in this paper.

Before going any further, a distinction has to be made between two notions: the *2D motion field* and the *optical flow* [9]. The former is the projection on the 2D image plane of the 3D motion in the scene. The latter is the field associated with the spatiotemporal variation of intensity. In video coding, motion estimation techniques try to reduce the temporal redundancies. The goal is not to assess the motion present in a scene, but to model the changes in the spatiotemporal intensity and therefore to estimate the optical flow. In that respect, the term motion field should be understood in the following as the optical flow.

The changes in the spatiotemporal intensity derive from three main causes. The first one is called global motion or camera motion. Even though no motion may occur in the scene, the motion of the camera induces a global motion in the captured scene. The second cause is the intrinsic motions of the objects in the scene. These can be seen as local motions as they do not affect the entire image. The third cause is a variation of illumination. If the lighting conditions change while the sequence is being captured by the camera, the optical flow will be influenced. As camera motion (such as pan or zoom) can be efficiently handled if it is globally estimated, two-stage global/local motion estimation have been proposed in [10]–[12]. However, generally no distinction is made between global and local motions. The global motion is thus taken into account through local estimates of the motion. In this context, the motion estimation techniques discussed hereafter rely only on local motion estimation. Although algorithms estimating the variation of illumination have been proposed [13]–[15], as a general rule the variation of illumination is not taken into account by the motion estimation techniques. The hypothesis is that there is no change of illumination and that the variations in the spatiotemporal intensity are due only to the global and local motions.

As far as the local motion estimation techniques are concerned, a number of very different algorithms have been proposed in the literature and detailed reviews of them are given in [3], [16]–[21]. Originally, these algorithms were developed for applications such as computer vision, image sequence analysis and video coding. They can be divided into four main groups: gradient techniques [22]–[25], pel-recursive techniques [6], [26], [27], block matching techniques [7], [8], and frequency-domain techniques [28]–[30]. Among those four groups, only pel-recursive and block matching techniques have been specifically developed for image sequence coding. In particular, block matching tech-

niques are very suitable in coding schemes based on a discrete cosine transform (DCT) such as those adopted by the recent standards MPEG-1 [31], [32], MPEG-2 [33], [34], and H.261 [35]. Hence, coding applications often rely on block matching motion estimation techniques.

Despite their widespread use, block matching techniques share several common drawbacks: unreliable motion fields in the sense of the true motion in the scene, block artifacts, and poor motion compensated prediction along moving edges. In order to remove these drawbacks, a locally adaptive multigrid block matching algorithm is proposed in this paper [36]. Accurate motion compensated prediction and low overhead information constituted the guidelines for the development of this algorithm. Consequently, it meets the desired features of a motion estimation technique for video coding. The algorithm adapts to the spatial content of the scene and thus provides accurate prediction in detailed areas while requiring a reduced amount of overhead information in uniform regions. Furthermore, a criterion is used to optimally balance the motion and the prediction error information [37]. Hence, the algorithm takes into account the global minimization of the sum of the two latter components.

This paper is structured as follows. The characterization of the motion is addressed more precisely in Section II. In Section III, classical motion estimation algorithms are reviewed in a video coding perspective. Simulation results are presented in order to compare the performances of these techniques. The locally adaptive multigrid block matching motion estimation is introduced in Section IV. The method is then analyzed and simulation results are presented. Finally, Section V draws the conclusions.

## II. CHARACTERIZATION OF THE MOTION

Before discussing in more details motion estimation techniques, the notion of motion should be clarified in the framework of image sequence processing.

Formulations in terms of either instantaneous velocity or displacement are possible. Due to the discrete nature of image sequences along time, the instantaneous velocity  $\vec{v}$  of a pixel and its displacement  $\vec{d}$  are related by a constant factor  $\Delta t$  which corresponds to the temporal sampling interval. Consequently, in this case these two quantities are interchangeable and both formulations are equivalent. However, the last statement is no longer valid when the motion of a group of pixels is modeled by a set of motion parameters (parametric approaches, see Section II-A). In this case, instantaneous velocity and displacement formulations may lead to distinct models. Hereafter the formulation in terms of displacement is adopted, and the term motion vector should be understood accordingly.

In a digital image sequence, the 4D spatiotemporal continuum is projected onto a 3D discrete sample grid. A distinction is made between two entities: the 2D motion field and the optical flow [9]. The 2D motion field is defined by the projection of the 3D motion on the 2D image plane. The optical flow corresponds to the spatiotemporal variation

of intensity. In the ideal case, the optical flow corresponds to the 2D motion field. However, in practice this is not guaranteed. For instance, it may happen that a moving object gives rise to a constant brightness pattern, thus the optical flow is zero even though motion exists in the scene. Conversely, in a still scene the optical flow may be non zero due to illumination changes.

In video coding, motion estimation techniques estimate the trajectory of pixels over successive images in order to express the current image intensity from previous information. Therefore they estimate the optical flow. In what follows, no distinction will be made between these two notions, and the term motion should be understood as optical flow.

#### A. Deterministic Versus Probabilistic and Nonparametric Versus Parametric

The motion can either be identified in a deterministic framework or a probabilistic one [38]. The former approach is usually chosen and the motion is seen as a deterministic quantity which is unknown. The corresponding estimator is referred to as maximum likelihood (ML). By maximizing the probability of the observed sequence with respect to the unknown motion, it is possible to estimate the latter. All the techniques presented in this paper (i.e., gradient [22]–[25], pel-recursive [6], [26], [27]) and block matching [3], [7], [8]) rely on this deterministic approach and can thus be seen as arising from a maximum likelihood approach.

The alternative to the deterministic view of motion is the probabilistic (Bayesian) one. The latter models the motion as a random variable. The ensemble of motion vectors forms a random field which is usually modeled by a Markov random field (MRF). Based on this assumption, it has been shown that the joint distribution function characterizing the random field is a Gibbs distribution. Estimators such as maximum *a posteriori* (MAP) and minimum expected cost (MEC) can thus be derived readily. Motion estimation techniques based on this probabilistic approach have been proposed in [39], [40].

The estimation of the motion is underconstrained and results in an ill-posed problem. Therefore, all motion estimation techniques need an additional constraint. This constraint can be implicit or explicit. A distinction can be made between nonparametric and parametric motion estimation techniques which rely on nonparametric and parametric motion models respectively [41]. Nonparametric techniques rely on a dense motion field. In these techniques, an explicit constraint (e.g., smoothness or local uniformity) is introduced in order to regularize the ill-posed problem. In contrast, parametric techniques model the motion of a region whose pixels have a coherent motion with a single set of parameters, referred to as motion parameters. Hence, these motion parameters are estimated instead of the motion field itself. Consequently, such a motion model describes in a compact way the motion throughout an image sequence. The computation of the motion field is implicitly constrained by the motion model itself and an explicit constraint therefore becomes useless.

#### B. Motion Models and Support for the Motion Estimation

As far as parametric techniques are concerned, various motion models have been derived and studied. The assumption of a planar surface under a perspective projection leads to an eight-parameter model [42]. Under the further hypothesis of distant shallow plane, the perspective projection is closely approximated by the orthographic projection [42]. The latter projection leads to the following *affine motion model*, which involves six parameters and can be expressed as

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a_1 + a_2 \cdot x + a_3 \cdot y \\ a_4 + a_5 \cdot x + a_6 \cdot y \end{pmatrix} \quad (1)$$

where  $(x, y)^T$  are the spatial coordinates. Using more meaningful parameters, it can also be written as

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} s_x \cdot \cos \theta_x & -s_y \cdot \sin \theta_y \\ s_x \cdot \sin \theta_x & s_y \cdot \cos \theta_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix} \quad (2)$$

where  $d_x$  and  $d_y$  are the two components of a translation vector,  $s_x$  and  $s_y$  are the scaling ratios in the  $x$  and  $y$  directions, and  $\theta_x$  and  $\theta_y$  are the rotation angles around the  $x$  and  $y$  axis. This model is capable of describing rotation, zoom, and even some nonrigid body motions such as shear motion.

The model defined by (2) can be simplified to four parameters by restricting  $s_x = s_y = s$  and  $\theta_x = \theta_y = \theta$

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} s \cdot \cos \theta & -s \cdot \sin \theta \\ s \cdot \sin \theta & s \cdot \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}. \quad (3)$$

Finally, the simpler model is the *translational motion model* which defines the motion of an entity by a translation vector

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}. \quad (4)$$

This model is motivated by the fact that a complex motion can be approximated, under certain hypotheses, by a sum of infinitesimal translations. However, this model has some limits and cannot cope with complex scenes.

Fig. 1 illustrates some examples of motion field which can be represented with the two, four, and six parameters models above. Although motion models involving more than six parameters have also been considered and studied, they are more commonly used in image analysis and vision applications.

In parametric motion estimation techniques, the motion model and the support for the estimation are obviously closely related. A complex model results in a better description of the motion in a sequence, and thus allows representing efficiently the motion of a larger region of the image. Conversely, a simple model is sufficient to represent the motion of a small region of the image.

In coding applications the motion parameters usually have to be sent to the decoder as overhead. Therefore,

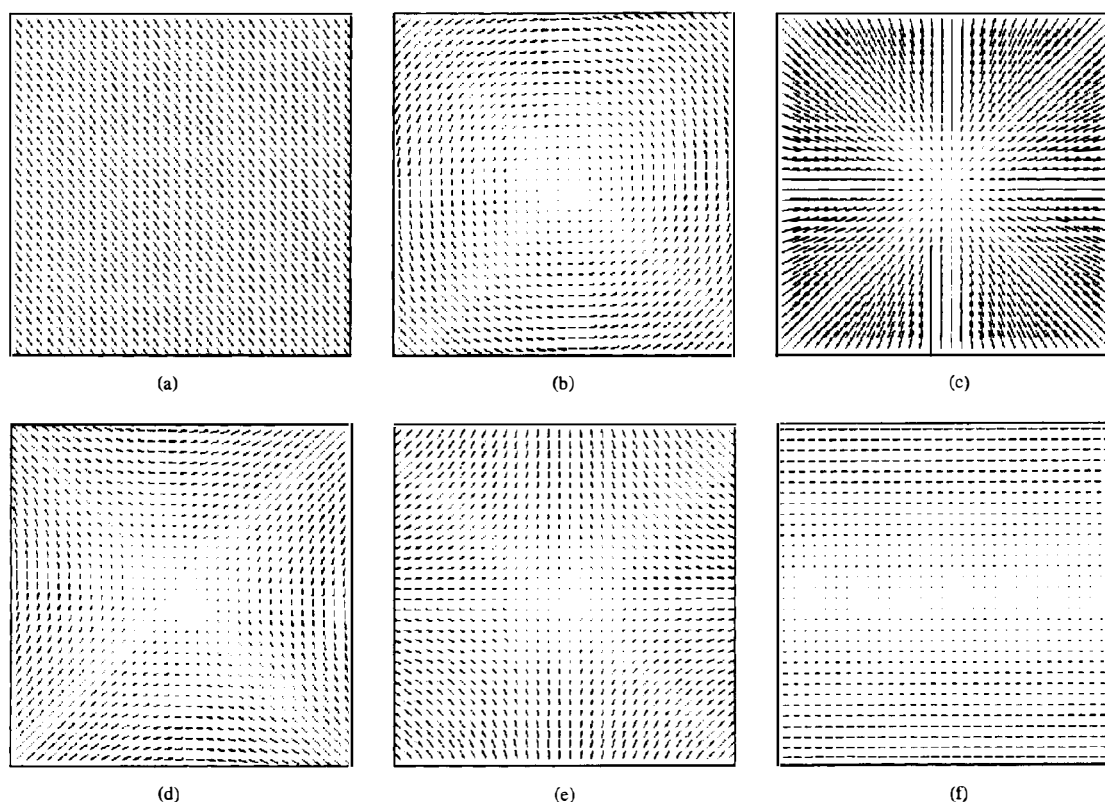


Fig. 1. Examples of motion fields when using 2, 4 and 6 parameter models: (a) 2 parameters in (4), (b), (c) 4 parameters in (3), and (d), (e), (f) 6 parameters in (2).

an optimal tradeoff has to be found on the motion model complexity and the support for the estimation in order to optimally balance the accuracy to describe the motion in the sequence and the amount of motion parameters to be transmitted. Finally, a complex model leads to a greater difficulty in parameter estimation and a higher computational complexity. In particular, the process becomes more sensitive to noise.

In [43] a block-based motion estimation with a hierarchy of models is introduced. The algorithm selects for each block a model depending on the scene content. Of course, the difficulty of this approach is to select the optimal model and to estimate the corresponding parameters.

In most of the recent coding schemes, pels or small blocks of pels are coded separately (first generation coding). For instance, the current standards MPEG-1 [31], [32], MPEG-2 [33], [34], and H.261 [35] are based on transform coding in which the image is partitioned into small blocks which are then coded separately. In this context, motion estimation techniques that work on a pel-by-pel or block-by-block basis are the most suitable. The former case corresponds to nonparametric approaches (e.g., optical flow gradient techniques [22]–[25] or pel-recursive techniques [6], [26], [27]). In the latter case, as the region described by one set of parameters is small, a simple model is sufficient. For this

reason, block matching motion estimation techniques [3], [7], [8] relying on the simple translational motion model are the most commonly used. These techniques can be considered as a trivial case of parametric techniques. As this paper emphasizes first generation coding techniques, gradient, pel-recursive and block matching techniques will be described in more details in Section III.

In an object-based coding scheme, such as those proposed in [44], [45], the scene is represented in terms of objects (second generation coding [4]). Therefore, in this case an object-based parametric motion estimation is more meaningful. Although this subject has already been intensively studied in the field of image analysis, it is still new in the field of coding. However, the domain is promising and algorithms have been proposed in [46]–[51]. As the motion estimation is performed on large regions of the image a more complex motion model is required, thus the affine model is often chosen.

### III. REVIEW OF MOTION ESTIMATION TECHNIQUES

A number of very different motion estimation algorithms have been proposed in the literature. Detailed reviews are given by Musmann [3], Nagel [16], Aggarwal and Nandhakumar [17], Singh [18], Sezan and Lagendijk [19], Barron *et al.* [20], and Tziritis and Labit [21]. These algo-

gorithms have been developed for very different applications such as image sequence analysis, machine vision, robotics, image sequence restoration or image sequence coding.

Even though [16]–[20] provide detailed reviews of motion estimation algorithms, they address the problem in an image sequence analysis perspective. As stated before, the purpose of motion estimation in the field of coding is significantly different. More specifically, the determination of the motion is not an intrinsic goal; indeed motion estimation techniques aim at minimizing the bandwidth corresponding both to the prediction error information and to the motion overhead. Therefore, in the remainder of this section some of the classical motion estimation algorithms are discussed and compared from an image sequence coding point of view.

Motion estimation techniques can be divided into four main groups:

- gradient techniques [22]–[25],
- pel-recursive techniques [6], [26], [27],
- block matching techniques [7], [8],
- frequency-domain techniques [28]–[30].

Gradient techniques have been developed for image sequence analysis applications. They solve the optical flow and results in a dense motion field. Both pel-recursive and block matching techniques have been developed in the framework of image sequence coding. Pel-recursive techniques can be considered as a subset of gradient techniques. However as they constitute an important contribution in the field of coding, we consider them as a separate group. Block matching techniques are based on the minimization of a disparity measure. They are the most widely used in coding applications. Finally, frequency-domain techniques are based on the relationship between transformed coefficients (e.g., Fourier or Gabor transform) of shifted images. However, they lack a widespread use, especially in the field of image sequence coding, and therefore they will not be further discussed in this paper.

In the following, the image intensity at pixel location  $\vec{r} = (x, y)^T$  and at time  $t$  is denoted by  $I(\vec{r}, t)$ , and  $\vec{d} = (d_x, d_y)^T$  is the displacement during the interval  $\Delta t$ . All the techniques presented hereafter rely on the hypothesis that a change in the image intensity  $I(\vec{r}, t)$  is due only to the displacement  $\vec{d}$ . It is expressed by

$$I(\vec{r}, t) = I(\vec{r} - \vec{d}, t - \Delta t) \quad (5)$$

and the displaced frame difference (DFD) is defined as

$$\text{DFD}(\vec{r}, t, \vec{d}) = I(\vec{r}, t) - I(\vec{r} - \vec{d}, t - \Delta t). \quad (6)$$

#### A. Gradient Techniques

Gradient techniques rely on the hypothesis that the image luminance is invariant along motion trajectories. The Taylor series expansion of the right hand side in (5) gives

$$I(\vec{r} - \vec{d}, t - \Delta t) = I(\vec{r}, t) - \vec{d} \cdot \vec{\nabla} I(\vec{r}, t) - \Delta t \frac{\partial I(\vec{r}, t)}{\partial t} + \text{higher order terms} \quad (7)$$

where  $\vec{\nabla} = [(\partial/\partial x), (\partial/\partial y)]^T$  is the gradient operator. Neglecting the higher order terms (first order approximation), assuming the limit  $\Delta t \rightarrow 0$ , and defining the motion vector  $\vec{v} = (v_x, v_y)^T = \vec{d}/\Delta t$  we obtain

$$\vec{v} \cdot \vec{\nabla} I(\vec{r}, t) + \frac{\partial I(\vec{r}, t)}{\partial t} = 0. \quad (8)$$

The latter equation is known as the *spatio-temporal constraint equation* or the *optical flow constraint equation* [22].

As the image intensity change at a point due to motion gives only one constraint (8), while the motion vector at the same point has two components, the motion field (actually the optical flow) cannot be computed without an additional constraint. In fact, only the projection of  $\vec{v}$  on  $\vec{\nabla} I$ , in other words the component of  $\vec{v}$  parallel to the intensity gradient, can be determined from (8). This problem is known as the *aperture problem*. Therefore an additional constraint must be introduced to regularize the ill-posed problem and to solve the optical flow.

In [22], Horn–Schunck introduce a smoothness constraint, that is to minimize the square of the optical flow gradient magnitude

$$\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 \quad \text{and} \quad \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2. \quad (9)$$

Consequently, the optical flow is obtained by minimizing the following error term

$$\iint \left\{ \left( \vec{v} \cdot \vec{\nabla} I + \frac{\partial I}{\partial t} \right)^2 + \alpha^2 \left[ \left( \frac{\partial v_x}{\partial x} \right)^2 + \left( \frac{\partial v_x}{\partial y} \right)^2 + \left( \frac{\partial v_y}{\partial x} \right)^2 + \left( \frac{\partial v_y}{\partial y} \right)^2 \right] \right\} dx dy \quad (10)$$

where  $\alpha^2$  is a weighting factor. This minimization problem is solved by the variational calculus and an iterative Gauss–Seidel procedure.

Many variations of the above algorithm have been proposed. Instead of the smoothness constraint, an assumption of local uniformity is made by Lucas and Kanade in [23]. In [24], Nagel develops the Taylor series of (7) up to the second-order terms. In [25], Nagel introduces an oriented smoothness constraint, which takes occluding edges into consideration, instead of the isotropic smoothness constraint used in (10). More detailed discussions on the many variations of the above Horn–Schunck algorithms can be found in [16], [17], [20]. The above approach deals with images at a single resolution scale, hierarchical schemes based on the spatiotemporal constraint equation have been developed in [52], [53].

All these techniques result in a dense motion field (i.e., a motion vector per pixel). This is qualitatively interesting for motion analysis applications. However, from an image sequence coding point of view, these techniques suffer from two serious drawbacks. First the smoothness con-

straint leads to an increased energy of the prediction error, especially on moving objects boundaries. Second, the dense motion field requires much overhead information. In [54], this problem is tackled through the use of a motion vector field coded at different resolutions. At each level, only the motion vectors from which the motion compensation benefits most are coded.

### B. Pel-Recursive Techniques

Pel-recursive techniques recursively minimize the prediction error or DFD in (6). The recursion is usually carried out on a pel-by-pel basis, leading to a dense motion vector field. These methods are among the first motion estimation techniques developed for image sequence coding applications and have been designed with a constraint of a very low hardware complexity. Pel-recursive techniques can be considered as a subset of the gradient techniques in which the spatiotemporal constraint is minimized recursively.

The first pel-recursive algorithm was proposed by Nevtrali-Robbins in [6]. In this algorithm, the DFD<sup>2</sup> is iteratively minimized by the steepest descent technique,

$$\vec{d}^{\vec{k}+1} = \vec{d}^{\vec{k}} - \frac{\epsilon}{2} \nabla_{\vec{d}} \text{DFD}^2(\vec{r}, t, \vec{d}^{\vec{k}}) \quad (11)$$

with a constant gain  $\epsilon > 0$ , and  $k$  denotes the iteration index. From the definition of the DFD, (6), we have

$$\begin{aligned} \nabla_{\vec{d}} \text{DFD}^2(\vec{r}, t, \vec{d}) &= 2 \text{DFD}(\vec{r}, t, \vec{d}) \\ &\cdot \nabla_{\vec{r}} I(\vec{r} - \vec{d}, t - \Delta t). \end{aligned} \quad (12)$$

Substituting (12) in (11), the displacement vector update becomes

$$\begin{aligned} \vec{d}^{\vec{k}+1} &= \vec{d}^{\vec{k}} - \epsilon \text{DFD}(\vec{r}, t, \vec{d}^{\vec{k}}) \\ &\cdot \nabla_{\vec{r}} I(\vec{r} - \vec{d}^{\vec{k}}, t - \Delta t). \end{aligned} \quad (13)$$

The performance of the pel-recursive techniques depends strongly on the way to compute the update term in (13). In the algorithm introduced in [6], the iteration from  $k$  to  $k+1$  is carried out either on one pel location, or from one pel to its consecutive neighbor. To smooth out the effect of noise, the algorithm can be extended by calculating and averaging the update term on several pels.

Improved algorithms based on the same principle have been proposed for instance by Cafforio and Rocca [26], and Walker and Rao [27]. Compared to [6], the gain  $\epsilon$  is substituted by variables in order to achieve a better adaptation to the local image statistic and consequently a better convergence. A more detailed description of various pel-recursive algorithms, as well as some comparative results, is given in [3].

Special care has been taken to enhance the tracking capability and the stability of the pel-recursive algorithms. In [55] a multiple frames model-based approach is presented, whereas in [56] a multiple mask regularization technique is proposed. The algorithm introduced in [57] shares

characteristics of both pel-recursive and block matching approaches by combining motion information sent from the encoder and motion information recursively estimated at the decoder.

Provided that the recursion has a sufficiently rapid convergence (i.e., it can handle motion discontinuities), such pel-recursive algorithms may overcome the problem of multiple moving objects. Furthermore, when the update of the displacement vector is based only on previously transmitted data (causality), the decoder is able to estimate the same motion field than the encoder. In this case, no overhead motion information is required, which is of course a further advantage of these methods.

However, the causality constrains these algorithms and reduces their prediction capability relative to noncausal methods. Furthermore, the pel-recursive motion estimation technique (with recursion on pels) is not compatible with transform coding of the DFD, as in this case the decoder is unable to reconstruct the motion vectors. Moreover, the algorithm implies an increased computational complexity at the decoder, as the latter should also estimate the motion field. Pel-recursive algorithms suffer from two further drawbacks. First, as the error function to be minimized contains generally many local minima, the iterative procedure may converge to a local minimum rather than to the global one. In particular, these algorithms are very sensitive to noise. Second, large displacements and discontinuities in the motion field cannot be efficiently handled.

### C. Block Matching Techniques

Block matching algorithms are based on the matching of blocks between two images, the aim being to minimize a disparity measure [3], [7], [8]. Specifically developed for image sequence coding, they are widely used in this field.

In block matching motion estimation, the image is partitioned into blocks and the same displacement vector is assigned to all pixels within a block. The motion model usually assumes that an image is composed of rigid objects in translational motion. Although this model is clearly restrictive, it is justified by the fact that complex motion can be decomposed as a sum of translational components. Consider the problem of predictive coding, the aim of motion estimation is to find the displacement vector  $\vec{d}$  which allows predicting  $I(\vec{r}, t)$  from  $I(\vec{r}, t - \Delta t)$  in (5). For each block, the displacement vector is evaluated by matching the information content of a measurement window  $\mathcal{W}$  with that of a corresponding measurement window within a search area  $\mathcal{S}$ , placed in the previous frame, and by searching the spatial location minimizing the matching criterion,

$$\vec{d} = \arg \min_{\vec{d} \in \mathcal{S}} \sum_{\vec{r} \in \mathcal{W}} \|I(\vec{r}, t) - I(\vec{r} - \vec{d}, t - \Delta t)\| \quad (14)$$

where the most widely used distance measures are the quadratic norm  $\|x\| = x^2$  and the absolute value  $\|x\| = |x|$ .

Finding an absolute minimum for the matching criterion can only be guaranteed by performing an exhaustive search

of a series of discrete candidate displacements within a maximum displacement range, this technique is called full-search block matching. Despite the heavy computations it requires, it is widely used in video coding, due to its simplicity and ease of hardware implementation. In order to decrease the computational load of the full-search algorithm, fast search techniques have been proposed [7], [8], [58], [59]. These fast search techniques afford significantly reduced computation times compared to the full-search algorithm. However, convergence toward the global minimum is guaranteed only when the matching criterion is a monotonic function of  $\vec{d}$ .

Block matching algorithms have been designed initially to estimate displacements with a precision of one pixel, however a sub-pixel accuracy can be obtained. For this purpose, the image intensity has to be interpolated at fractional pixel locations. In practice, this stage is implemented in post-processing where the one pixel accuracy displacement vectors are refined to a fractional pixel precision. Notably, this post-processing significantly increases the computational complexity.

The above described algorithms deal with images at a single resolution scale. In order to reduce computational complexity and to take into account the multi-scale characteristic of the motion in a scene, hierarchical [60], [61] and multigrid [36] algorithms have been proposed.

In the standard block matching technique, the motion is restricted to translation. However, block matching based motion estimation algorithms that relax this constraint have been investigated. In [62] an affine model for image matching is proposed, where each block undergoes an affine transform instead of a translation in the standard technique. Similarly, a generalized block matching algorithm is proposed in [63] which includes complex motions such as rotation or nonlinear deformation. Of course, a more complex motion model leads to a more accurate motion estimation. However, it also introduces the two following drawbacks. First, the computational complexity is greatly increased and an accurate estimation of the motion parameters becomes difficult. Second, for coding applications, a more complex motion model means a higher amount of overhead information. If this additional overhead is not counterbalanced by the gain due to more accurate motion vectors, then the more complex models may in fact lead to a globally lower performance.

To conclude, as the block matching methods directly minimize the DFD, they are well suited for image sequence coding. Furthermore, due to the block-based nature of these techniques, they require only little overhead motion information. For these reasons, block matching motion estimation techniques are the most widely used in image sequence coding. Recent standards such as MPEG-1 [31], [32], MPEG-2 [33], [34], and H.261 [35] are based on them, even though the algorithm to estimate the motion vectors is not specified explicitly.

However, despite their successful applications, block matching motion estimation techniques suffer from several drawbacks. Among the major ones are: unreliable motion

fields in the sense of the true motion in the scene, block artifacts, and poor motion compensated prediction along moving edges.

#### *D. Comparison Between Gradient, Pel-Recursive and Block Matching Motion Estimation Techniques*

Experiments were carried out to assess the performance of the above motion estimation techniques in an image sequence coding perspective. Three of the best-known algorithms were chosen for comparison, namely the Horn-Schunck gradient technique [22], the Netravali-Robbins pel-recursive technique [6], and the full-search block matching technique. As different results would be obtained when using different algorithms, the conclusions inferred from the following experiments should be carefully considered. However, these experiments give a faithful insight about these three different approaches to motion estimation and most of the following observations are generally representative of them.

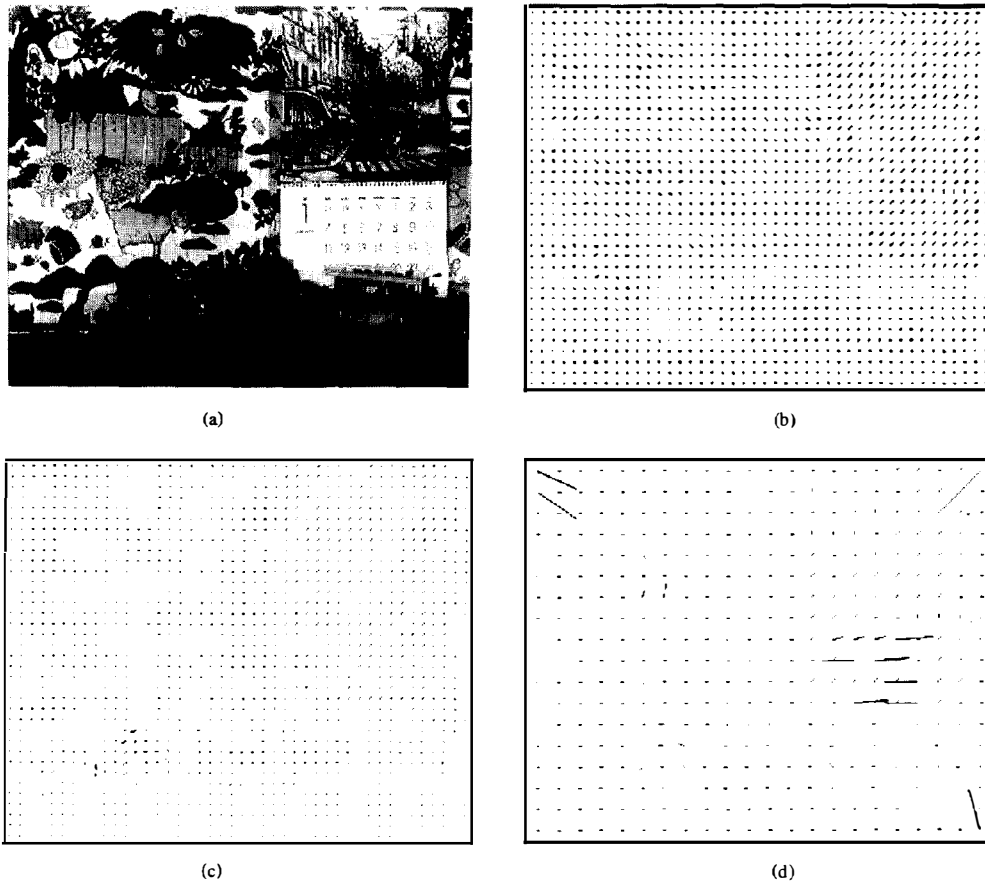
Simulations were performed on the luminance component of the three sequences "Mobile Calendar," "Table Tennis" and "Flower Garden" in CIF format (288 × 352 pixels, 8 b/pixel, 25 Hz). These sequences were chosen for their difficult motion and their different characteristics. In particular, "Mobile Calendar" and "Flower Garden" include panning and highly detailed moving areas, while "Table Tennis" contains large displacements and zooming.

To assess the performances of the different motion estimation techniques, the following comparisons were made. First, the subjective quality of the estimated motion field was evaluated, showing the capability of the algorithm to estimate the true motion in the scene. In particular, smooth motion fields are desired in coding in order to prevent artificial discontinuities in the DFD and to reduce the overhead to transmit the motion information. Second, the DFD error energy was measured, giving insight about the quality of the prediction which is a key feature in coding.

Based on simulation results, the following implementation choices were made. As far as the Horn-Schunck algorithm is concerned, the parameter  $\alpha^2$  in (10) was set to  $\alpha^2 = 100$ , and the results were considered after 100 iterations of the Gauss-Seidel procedure. Furthermore, in order to improve the approximation of the spatial gradient by first-order differences, a Gaussian spatial low-pass prefilter was applied to smooth out the effect of noise as suggested in [20].

Concerning the Netravali-Robbins algorithm, the parameter  $\epsilon$  in (13) was set to  $\epsilon = 1/1024$ , and the update term, i.e., the second term on the right hand side of (13) was averaged on an area of  $5 \times 5$  pels and clipped to a maximum magnitude of  $1/16$  pel/frame. The algorithm performed one iteration per pel. These choices correspond to the original algorithm proposed in [6]. However, it should be underlined that the performance of the pel-recursive algorithm depends heavily on the way the update term is computed.

Finally, the full-search block matching was applied on a block size of  $16 \times 16$  pixels with a maximum displacement of  $\pm 15$  pixels. In [36], it is shown that half-pixel accuracy



**Fig. 2.** Motion field: (a) frame of "Mobile Calendar" and motion vectors obtained when using (b) Netravali-Robbins, (c) Horn-Schunck, and (d) full-search block matching.

motion vectors leads to a very significant improvement when compared to one pixel accuracy, whereas a higher precision results in negligible changes. This result is in agreement with the results obtained in [64], [65]. Therefore, an half-pixel accuracy was chosen in our simulations.

Figs. 2-4 depict motion field needle diagrams obtained with the three different motion estimation algorithms. The motion vectors (subsamped for pel-recursive and gradient techniques) estimated between the frames 1 and 2 are shown for the sequences "Mobile Calendar" and "Flower Garden," and between frames 8 and 9, i.e., corresponding to the zoom, for the sequence "Table Tennis." The limitations of the pel-recursive technique proposed in [6] appear clearly. The method estimates moderate motion as in "Mobile Calendar" or on the field of flowers in "Flower Garden," but is unable to cope with large displacements as in "Table Tennis" or on the tree in "Flower Garden." This is not surprising as this approach to pel-recursion was developed for tame video-conferencing sequences. Besides, in the regions where the DFD remains small the update term is negligible [see (13)], and therefore the displacement estimate does not change. As a consequence, in the uniform areas, motion vector estimates propagate along lines due to the recursion. It sometimes results in wrong motion

vectors, however it does not have an impact on the coding performance, as the DFD is small anyway in these areas. With regards to the gradient and block matching techniques, both lead globally to motion fields close to the true motion in the scene. The motion fields estimated by the gradient technique tend to be very smooth due to the smoothness constraint. In contrast, the ones obtained by the block matching technique exhibit a few wrong motion vectors in the sense of the motion in the scene (but optimal in minimizing the DFD energy). A significant difference between the two techniques occurs for the portion of zooming in the sequence "Table Tennis." Although a motion model constraint to blockwise translation is often considered to be a major drawback in the presence of zoom, the block matching technique is able to estimate closely the true zooming motion. In contrast, the gradient technique without constraint on the motion model is surprisingly unable to do so. It can be concluded that, despite its simple motion model, the block matching motion estimation results globally in motion fields more representative of the true motion in the scene.

Fig. 5 shows the DFD energy corresponding to the above results. It appears clearly that the implemented pel-recursive technique is significantly less efficient than the two other



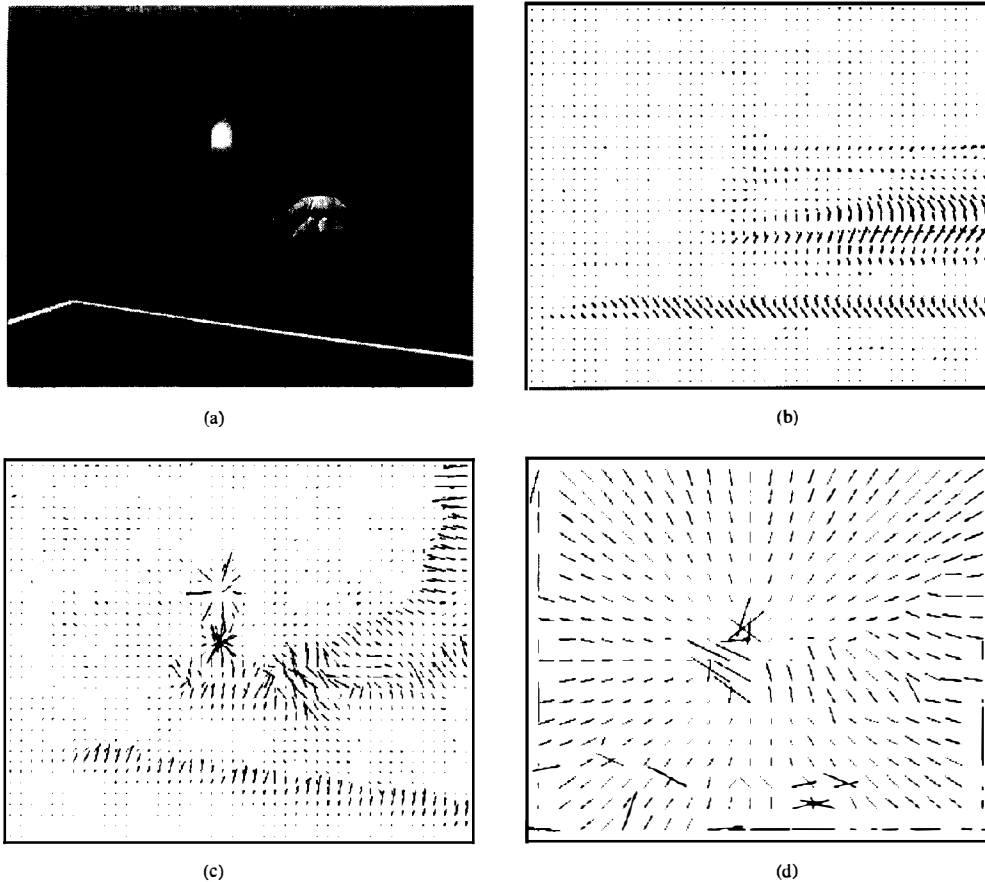


Fig. 3. Motion field: (a) frame of "Table Tennis" and motion vectors obtained when using (b) Netravali-Robbins, (c) Horn-Schunck, and (d) full-search block matching.

methods. It is mainly due to the difficulty of the algorithm to cope with large displacements and discontinuities in the motion field. As it provides a higher density of motion vectors (one vector per pixel instead of one per block), the gradient technique would be expected to outperform the block matching technique. However, it is observed that both perform similarly for the sequence "Mobile Calendar" and "Flower Garden," whereas the block matching technique is superior for the sequence "Table Tennis." In particular it is very interesting to observe that the block matching technique outperforms the gradient one in the portion of the sequence which contains a zoom, even though the method limits the motion to simple blockwise constant translations. The relatively poor performances of the gradient technique are explained by three limitations: the undesirable effect of the smoothness constraint, the error in the gradient estimation, and the limitation due to the modeling of the luminance by a Taylor expansion.

From the above simulation results, the following conclusions can be drawn. First, in the Horn-Schunck gradient technique, obtaining a dense motion field does not lead to an improved prediction capability, whereas it does induce a high amount of overhead information. Therefore, the

method is more interesting from an analysis rather than coding point of view. Second, the Netravali-Robbins pel-recursive technique, even though it provides a dense motion field, performs poorly in estimating the motion. The savings in overhead information is not sufficient to compensate for this poor prediction. Therefore, despite its drawbacks, the block matching technique, which directly minimizes the DFD energy and requires only one motion vector per block, appears to be the most suitable for coding purposes.

The above considerations illustrate the specificity of motion estimation for coding applications. Indeed, the aim is to minimize globally the bit rate corresponding both to the prediction error and to the motion parameters, rather than to estimate the true motion. Hence, an optimal tradeoff on the motion estimation accuracy has to be found in order to optimally balance the bandwidth corresponding to these two components. The three algorithms studied in this section correspond to three very different allocations of the bandwidth between the DFD and the motion parameters. More precisely, the Horn-Schunck technique leads to an accurate motion estimation thanks to a dense motion field, but it requires a very high amount of overhead information. Conversely, the Netravali-Robbins technique

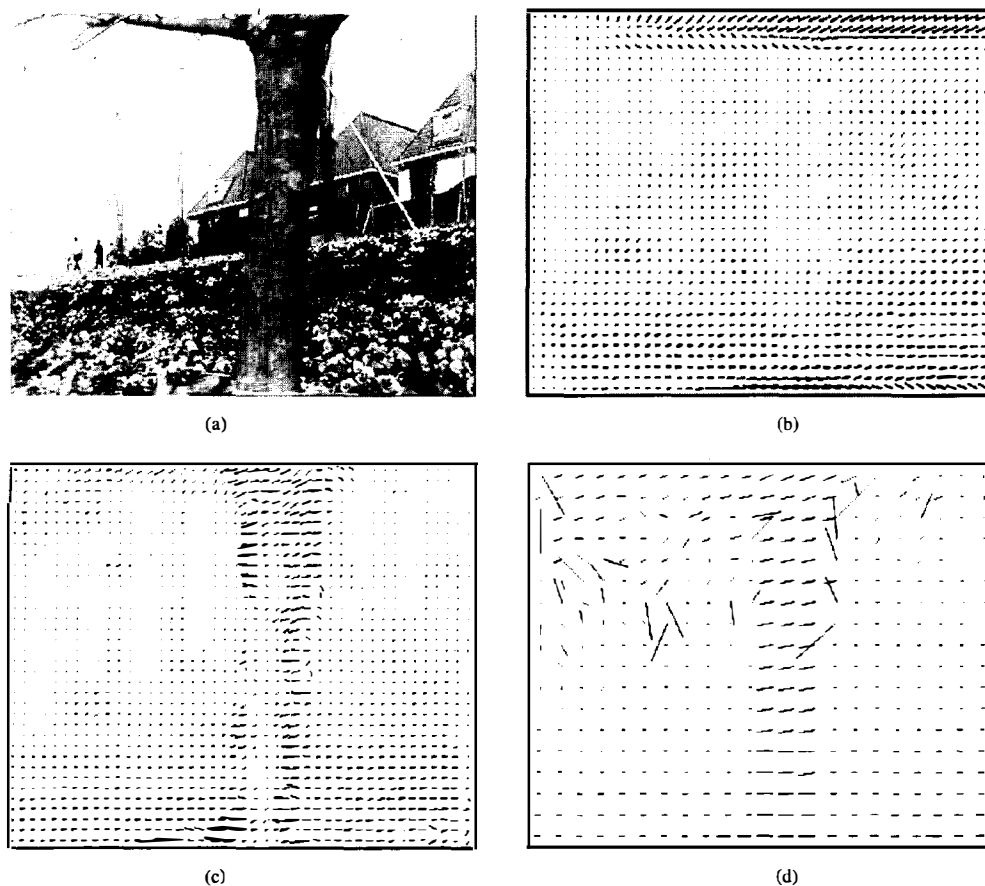


Fig. 4. Motion field: (a) frame of "Flower Garden" and motion vectors obtained when using (b) Netravali-Robbins, (c) Horn-Schunck, and (d) full-search block matching.

does not require the transmission of the motion vectors, but it does entail a poor motion estimation. In between these two extremes, the block matching technique relies on a simple motion model which leads simultaneously to a precise motion estimation and a low overhead. Therefore, it achieves a good allocation of the bandwidth between DFD and motion parameters, and in this respect outperforms the gradient and pel-recursive techniques. Furthermore, in the current standards MPEG-1 [31], [32], MPEG-2 [33], [34], and H.261 [35] which are based on a DCT transform coding, block matching motion estimation techniques are clearly the most appropriate.

Although it is clear that different results would be obtained when using different gradient, pel-recursive and block matching algorithms, the above comments are generally true of these three groups of motion estimation techniques.

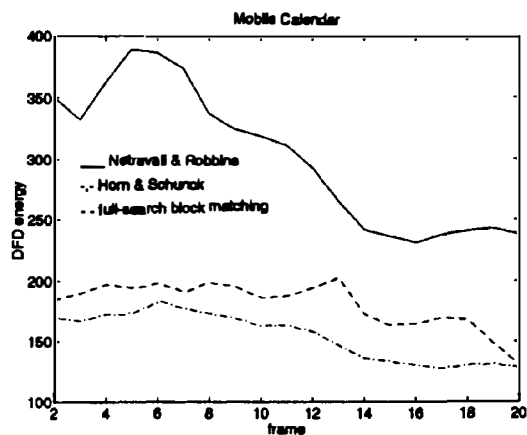
#### IV. LOCALLY ADAPTIVE MULTIGRID BLOCK MATCHING MOTION ESTIMATION

For the reasons discussed previously, block matching motion estimation techniques are the most widely used in video coding applications. However, they have several

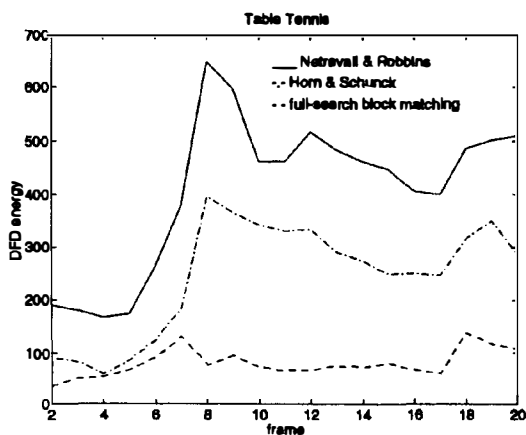
serious drawbacks: unreliable motion fields in the sense of the true motion in the scene, block artifacts, and poor motion compensated prediction along moving edges. Several techniques have been proposed to overcome these drawbacks, in order to improve performance.

In order to obtain more reliable motion fields, hierarchical block matching algorithms have been proposed [60], [61]. They are based on a multiresolution representation: A Laplacian pyramid in [60] and a low-pass filter by local average in [61]. A coarse but robust estimation of the motion field is obtained at the lowest resolution level, and is iteratively refined on the high resolution levels.

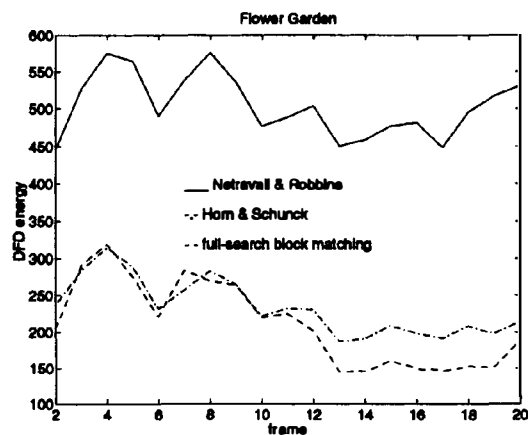
In order to overcome the problem of block artifacts in the motion compensated frame, due to the hypothesis that each pixel within a block has the same motion, different techniques have been investigated. A simple method is to use overlapped windows [66]–[68]. A very different technique, based on control grid interpolation, has been proposed in [69]. First, spatial displacements are specified for a small number of points in an image, named control points and normally chosen as vertices of a rectangular grid. Next, the displacement of the other points is determined by interpolating between the control points. Block matching algorithms can be considered as a trivial special case of



(a)



(c)



(c)

Fig. 5. DFD energy: Comparison between Netravali-Robbins, Horn-Schunck, and full-search block matching, (a) "Mobile Calendar," (b) "Table Tennis," and (c) "Flower Garden."

control grid interpolation in which interpolation is performed by nearest-neighbor, block artifacts being a result of the latter operation. In [70], a very similar algorithm is proposed. The grid is composed of triangular patches and an affine transform is used to represent the transformation of these patches.

A more promising approach to solve the problem of block artifacts and to provide more accurate prediction along moving edges is to segment the motion field. A locally variable block size block matching algorithm has been introduced [71], in which a binary-tree segmentation of the motion field is carried out. In [72], blocks corresponding to moving edges are segmented, taking into account the information of the previous frame. Similarly, another approach which segments the block-based motion field by means of vector quantization (VQ) has been proposed [73], [74]. In this method, blocks which contain several objects moving in different directions are segmented.

In this section, a locally adaptive multigrid block matching motion estimation technique is introduced [36], [75].

It is based on the multigrid theory developed in the field of mathematics [76]. The observation that natural scenes frequently contain motion at different scales motivates the introduction of multi-level algorithms. Large range displacements are robustly estimated on large-scale structures and short range displacements are accurately estimated on small-scale structures. Besides, finer structures are important in detailed areas, whereas coarser structures are sufficient in uniform regions. Locally adaptive multigrid block matching takes into consideration these remarks. It combines a multigrid structure and a quad-tree decomposition. Due to the multigrid structure, smooth, robust and accurate motion fields are obtained. Furthermore, the computational complexity is greatly reduced. Due to the adaptive structure, more precise motion vectors are estimated on moving edges and the side information is decreased in uniform areas.

Another problem specific to motion estimation for video coding is determining the appropriate precision of the motion estimate. Clearly, more precise motion vectors lead

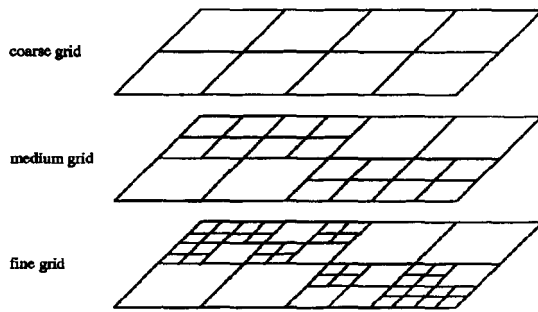


Fig. 6. Example of a three-grid adaptive multigrid structure.

to an improved prediction, but require a higher coding cost, and conversely a less accurate motion estimation needs a lower amount of side information but provides a poorer prediction. Very little research effort has been devoted to this topic in the literature. In this section, a criterion, the so-called entropy criterion, is proposed in order to optimally balance the amount of information corresponding to the prediction error and the representation of the motion [37]. The optimal tradeoff is reached by evaluating the transmission cost relative to both the prediction error and the motion information, and by minimizing the sum. Hence, a global minimization of the total bit rate is achieved. This criterion is general and can be applied to optimize most of motion estimation algorithms. In locally adaptive multigrid block matching motion estimation, it is applied to control the quad-tree segmentation.

In the remainder of this section, locally adaptive multigrid block matching, including the entropy criterion, is described in more detail.

#### A. Locally Adaptive Multigrid Block Matching

A new multi-level motion estimation technique, locally adaptive multigrid block matching, is discussed now. The formalism of the multigrid theory [76] as well as the local mesh refinement [77], provides a mathematical description of the algorithm. The multilevel structure is built on a set of grids with different sizes (multigrid structure). Therefore it is not a multiresolution approach in the classical sense. Furthermore, the multigrid structure is made locally adaptive to take into account the spatial content of the scene.

Fig. 6 illustrates the locally adaptive multigrid block matching motion estimation. The segmentation is carried out by a quad-tree decomposition. The algorithm starts by estimating the motion field on the coarsest level. Then, the grid is split only in the regions where the current solution accuracy is judged unsatisfactory. The corresponding motion vectors are down-projected to the finer grid and are refined. This procedure is iterated until a satisfactory accurate solution is obtained or a minimum block size is reached. The algorithm is thus composed of three major components which are the motion estimation at each level, the segmentation decision rule and the down-projection operator to map the motion field between two grids.

Obviously, the segmentation information, namely the quad-tree, should be sent to the decoder as side information. Nevertheless, as one bit per node of the tree is sufficient to completely define the segmentation, it represents a very low amount of information.

Due to the multigrid structure, large displacements are estimated robustly on the coarse grids with large matching windows, and small displacements are found accurately on the fine grids with small matching windows. Therefore, the method leads to motion fields which are simultaneously robust in the sense of the true motion in the scene and accurate in the sense of minimizing the prediction error. Furthermore, the local adaptation results in small grid sizes in areas containing detail and large grid sizes in the uniform ones. Thus the block artifacts are greatly reduced and very accurate predictions are obtained in important areas such as moving edges, while the amount of side information is kept low.

The proposed algorithm shares some similarities with the hierarchical block matching techniques proposed in [60], [61] and the variable size block matching algorithm in [71]. However, it includes new features which distinguish it. First, the algorithm relies on a multigrid structure and not on a multiresolution representation. Second, it combines advantageously the benefits of the multigrid approach with the advantages of the locally varying block size. Finally, it includes a criterion to control the segmentation and to achieve an optimal bandwidth allocation between the DFD information on the one hand and the motion parameters on the other hand.

The main components of the algorithm, namely the motion estimation at each level, the segmentation decision rule and the down-projection operator, play a key role on the performances of the algorithm. They are described in more detail below.

#### B. Motion Estimation at Each Level

As shown in Section III, block matching techniques are well-suited for coding applications. Consequently, block matching motion estimation is performed at each level. In order to reduce the computational complexity of the algorithm, a fast search technique, the modified  $n$ -step search, is applied. Fig. 7 illustrates the modified three-step search. At the first step, the nine locations defined by the set  $(0, \pm 2^{n-1})$  are evaluated. The best estimate is the initial point for the next step, and at the  $i$ th step, the eight locations defined by the set  $(0, \pm 2^{n-i})$  around the initial point are evaluated. Consequently, the resulting maximum displacement of the  $n$ -step search is  $2^{n-1}$ .

As the coarse grids provide robust estimates and the fine grids accurate ones, large (respectively small) maximum displacements are allowed on the coarse (respectively fine) grids. The fast search technique combined with the multigrid structure allow estimating large displacements with a very low computational complexity, when compared to a monoresolution full-search block matching algorithm. Indeed, in [36] it is shown that for a CCIR-601 format the number of match positions to be evaluated is decreased

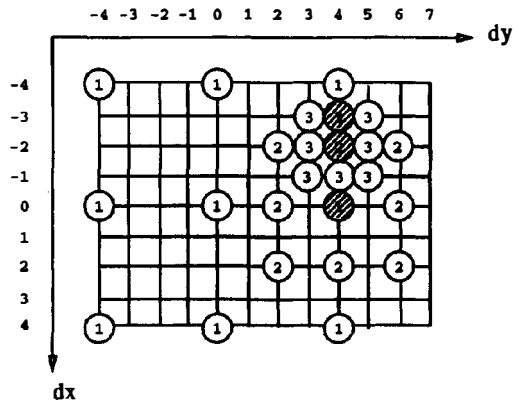


Fig. 7. Modified three-step search: example for a displacement  $dx = -3$  and  $dy = 4$ , and 25 search positions (the numbers  $i = 1, \dots, 3$  in circle indicate the search points at step  $i$ , the shaded ones indicate the displacement after step  $i$ ).

by approximately two orders of magnitude. Although this result is extremely important for software simulations, it cannot be deduced straightforwardly that the hardware implementation will be more efficient. However, a study of an earlier version of the proposed multigrid algorithm has been carried out in [78], [79] and has shown the feasibility of an efficient hardware implementation.

In the matching criterion, both the mean absolute error (MAE) and the mean square error (MSE) measures can be used. In [36], simulation results show that the MAE and the MSE matching criteria perform similarly, motivating the choice of the former due to its easier hardware implementation.

The above fast search technique leads to motion vectors with one pixel accuracy. It has been shown that sub-pixel accuracy leads to higher performances [36]. In this case, a post-processing which refines the one pixel accuracy motion vectors to a sub-pixel precision is performed after the locally adaptive multigrid block matching motion estimation. This operation requires the image intensity to be interpolated at fractional pixel locations. In the proposed algorithm, this interpolation is performed by bilinear interpolation.

### C. Segmentation Decision Rule

The above quad-tree decomposition requires evaluating the accuracy of the current motion vectors. More precisely, it requires a rule to decide whether to split a block. The segmentation decision rule influences greatly the overall performances of the algorithm. Depending on this rule, the algorithm can lead either to more accurate motion vectors which require a higher amount of side information (many blocks are split) or to poorer motion vectors which need a reduced amount of overhead information (few blocks are split). Even though it is very easy to gain on one of the two terms, prediction error or side information, the challenge is to achieve an overall gain.

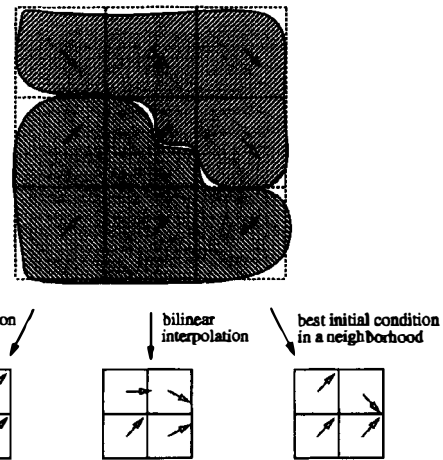


Fig. 8. Down projection: Comparison for the central block of duplication, bilinear interpolation, and best initial condition in a neighborhood.

A simple criterion to decide whether to split a block and applied in [71], [75], [80] is the following:

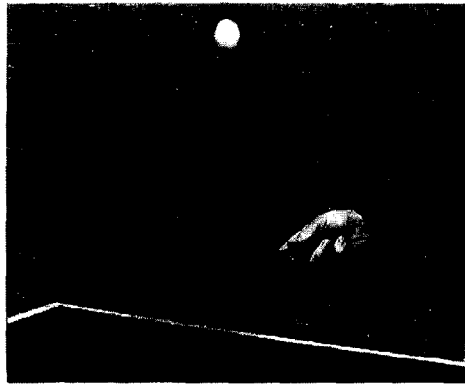
- If the MAE (or another error measure) of the motion compensated block is above a preset threshold  $T$ , the block is split.

$$MAE_{nosplit} > T \Rightarrow \text{split.} \quad (15)$$

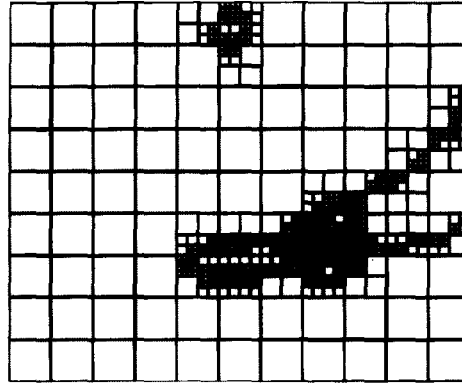
However, the above criterion does not guarantee that the extra-cost to send more motion parameters is worth the gain of decreasing the DFD energy. Furthermore, it requires to determine an appropriate value of the threshold  $T$ .

The entropy criterion aims at the control of the segmentation in order to reach the optimal bit allocation between motion parameters and DFD information [36], [37]. This criterion compares the extra-cost to send additional motion parameters with the gain obtained on the DFD side to decide whether to split a block. Moreover, it overcomes the problem of setting a threshold.

The entropy criterion demands an estimation of the transmission cost for both the DFD and the motion components. As far as the DFD is concerned and assuming an entropy coding, its coding cost is given by its entropy. In order to estimate the latter, a source model is needed and ideally a high order statistical model should be used. In this case one can compute the *total* entropy (i.e., high order entropy) which indicates the DFD coding cost. An important remark is that this last statement is always valid, even when a transform coding technique is applied to the DFD, as in MPEG-1 [31], [32], MPEG-2 [33], [34], and H.261 [35]. However, observations have shown that the correlation is very low in DFD [36], [81], [82]. Consequently, the DFD can be represented by a memoryless source model and the 0th order entropy provides a good approximation of the total entropy. Furthermore, by the very nature of the prediction process, the DFD exhibits a characteristic distribution which allows its modeling as a Laplacian probability density function (PDF) [5], [36]. Hence an



(a)



(b)

Fig. 9. "Table tennis." (a) a frame and (b) the corresponding final grid.

analytical expression to estimate the entropy of the DFD, namely its coding cost, can be derived [36], [37]. With regard to motion information, its cost is most of the time straightforward and computationally easy to estimate.

By minimizing the sum of the coding costs corresponding to the two components, DFD and motion information, an optimal tradeoff is reached. This minimization defines the entropy criterion. For the locally adaptive multigrid block matching algorithm, it can be written as follows:

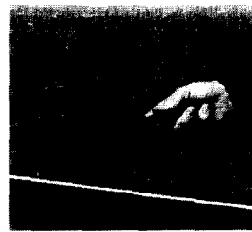
- If the extra-cost to send additional motion parameters is worth the gain obtained on the DFD side, then the block is split:

$$n \cdot (H_{\text{DFD}_{\text{nosplit}}} - H_{\text{DFD}_{\text{split}}}) > 4 \cdot H_{\vec{v}_{\text{split}}} - H_{\vec{v}_{\text{nosplit}}} \Rightarrow \text{split} \quad (16)$$

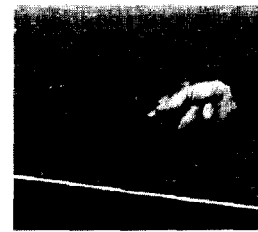
where  $n$  is the number of pixels in the block,  $H_{\text{DFD}_{\text{split}}}$  and  $H_{\text{DFD}_{\text{nosplit}}}$  are their entropy with/without split respectively, and  $H_{\vec{v}_{\text{split}}}$  and  $H_{\vec{v}_{\text{nosplit}}}$  the entropy of the motion vectors with/without split, respectively.

In the algorithm, the amount of information to transmit the segmentation information, i.e., the quad-tree, is negligible. Therefore, the extra cost is only represented by an increased number of motion vectors. The factor 4 is due to the fact that, in case of splitting, four motion vectors are transmitted for the block (quad-tree segmentation) instead of one.

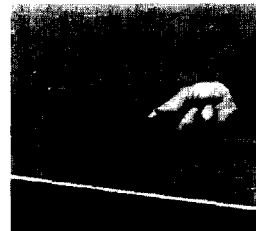
When compared to the criterion based on a threshold (15), the entropy criterion (16) avoids the setting of a threshold and guarantees a minimization of the total bit rate. The entropy criterion is therefore clearly superior as shown in [36], [37]. As far as the visual quality is concerned, the following remarks can be done. First, the motion estimation improvement is likely to lead to a large gain on moving edges which correspond to regions difficult to predict accurately. Therefore, the criterion is accepting more precise motion parameters in these visually important regions. Second, the bit rate saving due to the criterion can



(a)



(b)



(c)

Fig. 10. "Table tennis," zoom on the bat and player's arm: (a) original, (b) motion compensated prediction using full-search block matching, and (c) motion compensated prediction using the locally adaptive multigrid algorithm.

be used to enhance the visual quality of the reconstructed sequence (e.g., by decreasing the quantization step size in the DFD coding).

#### D. Down-Projection Operator

An operator is required to map the motion field between two grids in the coarse-to-fine iterative refinement process. This operator should avoid the propagation into fine levels of block artifacts due to the use of large block sizes on the coarser levels. It should also prevent wrong motion vectors estimates, due to local minima of the matching criterion, from propagating throughout the multigrid levels. Finally, it should guarantee smooth and robust motion field. To fulfill these requirements, the down-projection operator should incorporate a spatial consistency of the motion field.

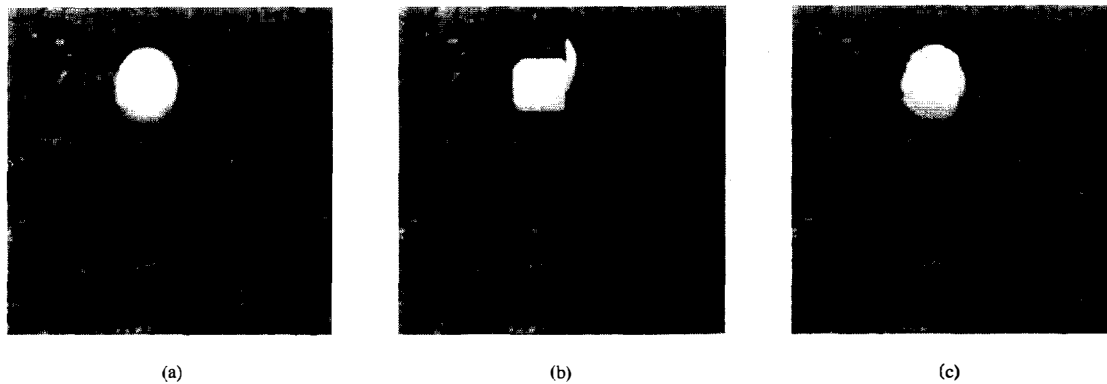


Fig. 11. "Table Tennis," zoom on the ball: (a) original, (b) motion compensated prediction using full-search block matching, and (c) motion compensated prediction using the locally adaptive multigrid algorithm.

According to the dyadic structure, four children subblocks are generated on the fine grid from the corresponding parent block on the coarser grid. Each of these subblocks needs an initial motion vector, obtained by the down-projection of the motion field estimated on the coarser grid. The simplest method duplicates four times the motion vector of the parent block. A bilinear interpolation is also possible. A more efficient operator consists in selecting for each child subblock the best initial condition among four parent blocks. The selection is based on the matching criterion, and the four parent blocks are the ones which are the closest to the considered child subblock. This down-projection is referred to as the best initial condition in a neighborhood.

Fig. 8 illustrates the down-projection and compares the duplication, the bilinear interpolation and the selection of the best initial condition in a neighborhood. In this example, the central block, which contains two different objects (indicated with stripes) moving in different directions (indicated with arrows), is shown after down-projection. In this case, the down-projection by duplication assigns an irrelevant estimate to the upper-right subblock. The bilinear interpolation leads to more effective estimates; nevertheless it does not fully overcome the above drawback. By exploiting the spatial consistency of the motion field, the method selecting the best initial condition in a neighborhood is able to assign to the upper-right subblock a motion vector issued from a neighboring block. Therefore, it fully overcomes the above problem. The example presented in Fig. 8 shows clearly the advantage of the down-projection by selecting the best initial condition in a neighborhood compared to the duplication and the bilinear interpolation. This observation has been confirmed by simulation results [36]. The down-projection by the best initial condition in the neighborhood fulfills the above desired requirements and is adopted in the algorithm.

#### E. Simulation Results

Simulation results are presented in this section to assess the performances of the proposed locally adaptive

Table 1 The Locally Adaptive Multigrid Algorithm

grid	block size	$n$ -step search	max. displ.
0	$4 \times 4$	2-step	$\pm 3$
1	$8 \times 8$	2-step	$\pm 3$
2	$16 \times 16$	3-step	$\pm 7$
3	$32 \times 32$	4-step	$\pm 15$

multigrid block matching motion estimation. The reference is the classical full-search block matching with the same parameters as in Section III.

Table 1 summarizes the choices made concerning the locally adaptive multigrid algorithm. The multigrid structure consisted of four levels, the smaller block size being  $4 \times 4$  pixels. With these parameters, the maximum displacement was  $\pm 28$  pixels. The resulting motion vectors were refined to half-pixel accuracy.

Fig. 9 depicts an example of the final segmentation obtained on the sequence "Table Tennis" while using the locally adaptive multigrid algorithm. The algorithm clearly generates large blocks in uniform areas and small blocks on the player's arm, the bat, and the ball.

From the above result, an enhanced visual quality is expected while using the locally adaptive multigrid algorithm. Figs. 10 and 11 show two original portions of the sequence "Table Tennis," as well as the motion compensated predictions obtained when using the full-search and the locally adaptive multigrid block matching motion estimation techniques. Whereas Figs. 10(b) and 11(b) exhibit strong block artifacts (especially on the bat, the hand and the ball) in Figs. 10(c) and 11(c) these block artifacts have been removed and boundaries are smooth and sharp, providing a more accurate prediction of the original frame. Consequently a greatly higher visual quality of the reconstructed sequence is obtained.

The performances within a coding scheme are now evaluated. This is very important to completely assess the performances of a motion estimation algorithm in the framework of coding. In particular, it allows taking into consideration the quality of the interframe coding as well as the amount of overhead information required to transmit

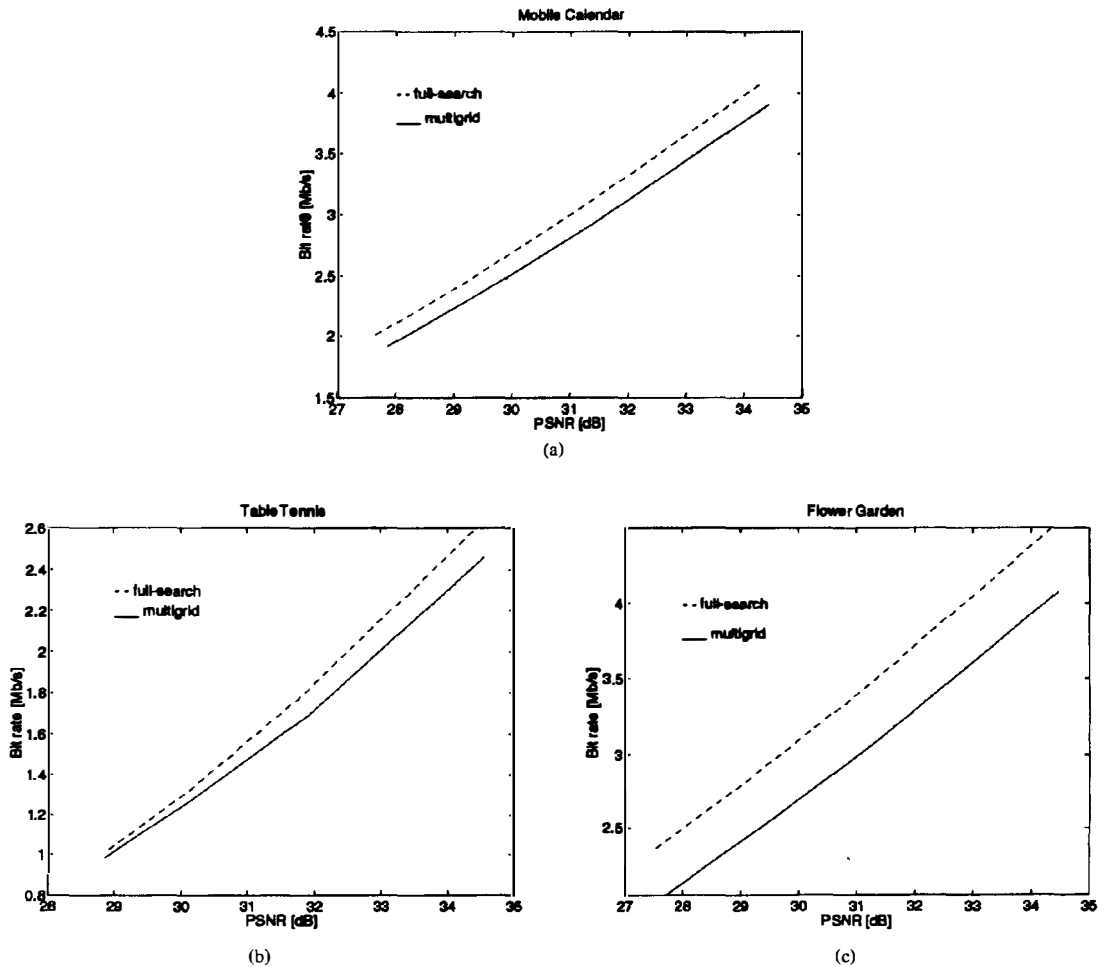


Fig. 12. Bit rate versus PSNR: comparison between Netravali–Robbins, Horn–Schunck, and full-search block matching, (a) “Mobile Calendar,” (b) “Table Tennis,” and (c) “Flower Garden.”

motion parameters. These results are expressed in terms of bit rate and peak signal-to-noise ratio (PSNR). However, it should be underlined that PSNR is a poor measure of the visual quality. Indeed, edges are perceptually very important [83]. Nevertheless, as they constitute only a small portion of the entire image, their importance is not reflected by PSNR. Even though it is a poor measure, the PSNR is widely used due to the lack of perceptually reliable visual quality measures.

The coding scheme used for comparison purpose was based on an interframe differential pulse code modulation (DPCM). The first frame was intraframe coded by a wavelet technique as proposed in [84], the following ones were motion compensated predicted. The resulting DFD’s were uniformly quantized and entropy coded by the adaptive arithmetic coder introduced in [85]. As far as the motion vectors are concerned, they have to be quantized, coded and transmitted to the decoder. With the block matching technique, the quantization of the motion vectors is intrinsic to the algorithm (in our case 1/2 pixel accuracy). The

quantized motion vectors were then differentially coded using the same adaptive arithmetic coder [85].

Fig. 12 shows the coding results obtained for “Mobile Calendar,” “Table Tennis” and “Flower Garden,” in terms of bit rate and PSNR. For the three sequences, the locally adaptive multigrid algorithm outperforms the full-search block matching. The saving in terms of bit rate is approximately 0.5 Mb/s (10% to 20%) for “Flower Garden,” 0.25 Mb/s (6% to 12%) for “Mobile Calendar,” and up to 0.2 Mb/s (up to 8%) for “Table Tennis.” It corresponds to a gain in terms of PSNR ranging from 0.5 to 1.5 dB. Moreover, despite the low gain in terms of PSNR, the visual quality of the reconstructed sequence is greatly enhanced by the locally adaptive multigrid algorithm. In particular moving edges are much sharper. Consequently, the efficiency of the proposed motion estimation algorithm is clearly shown.

## V. CONCLUSIONS

This paper addressed the problem of motion estimation for image sequence coding applications and more



precisely for first generation coding techniques such as those commonly adopted in digital TV applications. In this framework, the aims of motion estimation are to provide good temporal prediction while requiring little overhead information. Therefore, the determination of the motion is not an intrinsic goal. Motion estimation aims indeed at minimizing the amount of information corresponding to both the prediction error and the motion parameters.

After clarifying the notion of motion, classical motion estimation techniques have been reviewed in a perspective of coding applications. Despite a simple motion model, block matching techniques have been shown to be the most appropriate in the framework of first generation coding.

Finally, a locally adaptive multigrid block matching motion estimation has been introduced. It overcomes the drawbacks of classical block matching techniques, namely unreliable motion fields in the sense of the true motion, block artifacts and poor prediction along moving edges. Due to the multigrid structure, robust and accurate motion fields are obtained, whereas the computational complexity is greatly reduced. By adapting to the scene content, the method provides more accurate prediction along moving edges and a decreased amount of overhead information in uniform areas. Furthermore, the algorithm is controlled by an entropy criterion so that it achieves an optimal bandwidth allocation between the DFD information and the motion parameters. Simulation results have shown that the method results in a greatly enhanced visual quality of the reconstructed sequence and a significant saving in terms of bit rate.

#### ACKNOWLEDGMENT

The authors would like to thank A. Lippman, R. Ker-mode, K. Popat, and the anonymous reviewers for providing valuable comments to improve this paper.

#### REFERENCES

- [1] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. PROC-68, pp. 366-406, Mar. 1980.
- [2] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. PROC-69, pp. 349-389, Mar. 1981.
- [3] H. G. Musmann, P. Pirsch, and H. J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. PROC-73, pp. 523-548, Apr. 1985.
- [4] M. Kunt, A. Ikononopoulos, and M. Kocher, "Second generation image coding techniques," *Proc. IEEE*, vol. PROC-73, pp. 549-575, Apr. 1985.
- [5] A. N. Netravali and B. G. Haskell, *Digital Pictures Representation and Compression*. New York: Plenum, 1991.
- [6] A. Netravali and J. D. Robbins, "Motion compensated television coding part I," *Bell Syst. Tech. J.*, vol. 58, no. 3, pp. 629-668, Apr. 1979.
- [7] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1808, Dec. 1981.
- [8] T. Koga *et al.*, "Motion compensated interframe coding of video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Dec. 1981, pp. G5.3.1-G5.3.5.
- [9] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," MIT Tech. Rep. A.I. 917, Dec. 1986.
- [10] Y. T. Tse and R. L. Baker, "Global zoom/pan estimation and compensation for video compression," in *IEEE Proc. ICASSP'91*, Toronto, Canada, May 1991, vol. 4, pp. 2725-2728.
- [11] D. Adolph and R. Buschmann, "1.15 Mbit/s coding of video signals including global motion compensation," *Signal Process.: Image Commun.*, vol. 3, nos. 2-3, pp. 259-274, June 1991.
- [12] F. Moscheni, F. Dufaux, and M. Kunt, "A new two-stage global/local motion estimation based on a background/foreground segmentation," in *IEEE Proc. ICASSP'95*, Detroit, MI, May 1995.
- [13] C. R. Moloney and E. Dubois, "Estimation of motion fields from image sequences with illumination variation," in *IEEE Proc. ICASSP'91*, Toronto, Canada, May 1991, vol. 4, pp. 2425-2428.
- [14] S. Negahdaripour and C. H. Yu, "A generalized brightness change model for computing optical flow," in *IEEE Proc. Int. Conf. on Computer Vision*, Berlin, Germany, May 1993, pp. 2-11.
- [15] P. Treves and J. Konrad, "Motion estimation and compensation under varying illumination," in *IEEE Proc. ICIP'94*, Austin, TX, Nov. 1994, vol. 1, pp. 373-377.
- [16] H.-H. Nagel, "Image sequence—ten (octal) years—from phenomenology toward a theoretical foundation," in *Proc. Int. Joint Conf. on Patt. Recog.*, Paris, France, Oct. 1986, pp. 1174-1185.
- [17] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917-935, Aug. 1988.
- [18] A. Singh, *Optical Flow Computation: A Unified Perspective*. Los Alamitos, CA: IEEE Computer Soc. Press, 1991.
- [19] M. I. Sezan and R. L. Lagendijk, Eds., *Motion Analysis and Image Sequence Processing*. Boston, MA: Kluwer, 1993.
- [20] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Computer Vision*, vol. 12, no. 1, pp. 43-77, 1994.
- [21] G. Tziritas and C. Labit, *Motion Analysis for Image Sequence Coding*. Amsterdam: Elsevier, 1994.
- [22] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185-203, 1981.
- [23] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Image Understanding Workshop*, pp. 121-130, 1981.
- [24] H.-H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Computer Graphics and Image Process.*, vol. 21, pp. 85-117, 1983.
- [25] ———, "Constraints for the estimation of displacement vector fields from image sequences," in *Proc. Int. Joint Conf. on Artif. Intell.*, Karlsruhe, Germany, Aug. 1983, pp. 945-951.
- [26] C. Cafforio and F. Rocca, "The differential method for motion estimation," in *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed. New York: Springer-Verlag, pp. 104-124.
- [27] D. R. Walker and K. R. Rao, "Improved pel-recursive motion compensation," *IEEE Trans. Commun.*, vol. COM-32, pp. 1128-1134, Oct. 1984.
- [28] 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, Jan. 1974.
- [29] D. J. Heeger, "Optical flow using spatio-temporal filters," *Int. J. Computer Vision*, vol. 1, no. 4, pp. 279-302, Jan. 1988.
- [30] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *Int. J. Computer Vision*, vol. 5, pp. 77-104, 1990.
- [31] ISO/IEC JTC1 CD 11172, "Information technology—coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s—Part 2: Coding of moving pictures information," Tech. Rep., 1991.
- [32] D. LeGall, "MPEG: A video compression standard for multimedia," *Commun. ACM*, vol. 34, no. 4, pp. 47-58, Apr. 1991.
- [33] ISO/IEC JTC1/SC29/WG11 Moving Picture Experts Group. MPEG2 test model 4, 1993.
- [34] D. LeGall, "The MPEG video compression algorithm," *Signal Processing: Image Commun.*, vol. 4, no. 2, pp. 129-140, Apr. 1992.
- [35] CCITT SG XV, "Recommendation H.261—Video codec for audiovisual services at p\*64 kbit/s," Tech. Rep. COM XV-R37-E, Aug. 1990.
- [36] F. Dufaux, "Multigrid block matching motion estimation for generic video coding," Ph.D. dissertation, Swiss Fed. Institute of Tech., Lausanne, Switzerland, 1994.
- [37] F. Moscheni, F. Dufaux, and H. Nicolas, "Entropy criterion for optimal bit allocation between motion and prediction error information," in *SPIE Proc. Visual Commun. and Image Process. '93*, Cambridge, MA, Nov. 1993, vol. 2094, pp. 235-242.

- [38] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.
- [39] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 14, pp. 910-927, Sept. 1992.
- [40] P. Boutheymy and E. Francois, "Motion segmentation and qualitative dynamic scene analysis from an image sequence," *Int. J. Computer Vision*, vol. 10, no. 2, pp. 157-182, 1993.
- [41] P. Anandan, J. R. Bergen, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Boston: Kluwer, 1993, pp. 1-22.
- [42] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
- [43] H. Nicolas, "Hiérarchie de modèles de mouvement et méthodes d'estimation associées. Application au codage de séquences d'images," Ph.D. dissertation, Univ. Rennes I, France, 1992.
- [44] H. G. Musmann, M. Hoetter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Process.: Image Commun.*, vol. 1, no. 2, pp. 117-138, Oct. 1989.
- [45] M. Hoetter, "Object-oriented analysis-synthesis coding based on moving two-dimensional objects," *Signal Process.: Image Commun.*, vol. 2, no. 4, pp. 409-428, Dec. 1990.
- [46] M. Hoetter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation," *Signal Process.*, vol. 15, no. 3, pp. 315-334, Oct. 1988.
- [47] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Process.: Image Commun.*, vol. 3, no. 1, pp. 23-56, Feb. 1991.
- [48] P. Wagner and B. Girod, "Region-based motion field estimation," in *Picture Coding Symp. '93*, Lausanne, Switzerland, Mar. 1993, p. 4.5.
- [49] H. Sanson, "Region based motion estimation and compensation for digital TV sequence coding," in *Picture Coding Symp. '93*, Lausanne, Switzerland, Mar. 1993, p. 4.4.
- [50] J. Y. A. Wang, E. H. Adelson, and U. Desai, "Applying mid-level vision techniques for video data compression and manipulation," in *SPIE Proc. Digital Video Compression on Personal Computers: Algor. and Tech.*, San Jose, CA, Feb. 1994, vol. 2187.
- [51] M. Pardas, P. Salembier, and B. Gonzalez, "Motion and region overlapping motion estimation for segmentation-based video coding," in *IEEE Proc. ICIP'94*, Austin, TX, Nov. 1994, vol. 2, pp. 428-432.
- [52] F. Glazer, "Multilevel relaxation in low-level computer vision," in *Multiresolution Image Processing Analysis*, A. Rosenfeld, Ed. New York: Springer-Verlag, 1984.
- [53] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Computer Graphics and Image Process.*, vol. 43, pp. 150-177, Aug. 1988.
- [54] N. Haddadi and C. C. Kuo, "Multiple bit-rate video compression via progressive motion field coding," in *SPIE Proc. Image and Video Compression*, San Jose, CA, Feb. 1994, vol. 2186, pp. 233-244.
- [55] S. N. Efstratiadis and A. K. Katsaggelos, "A model-based pel-recursive motion estimation algorithm," in *IEEE Proc. ICASSP'90*, Albuquerque, NM, Apr. 1990, vol. 4, pp. 1973-1976.
- [56] —, "An adaptive regularized recursive displacement estimation algorithm," *IEEE Trans. Image Process.*, vol. 2, pp. 341-352, July 1993.
- [57] M. Orchard, "New pel-recursive motion estimation algorithms based on novel interpolation kernels," in *SPIE Proc. Visual Commun. and Image Process. '92*, Boston, MA, Nov. 1992, vol. 1818, pp. 85-96.
- [58] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. COM-33, pp. 888-896, Aug. 1985.
- [59] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. CSVT-3, no. 2, pp. 148-157, Apr. 1993.
- [60] P. Anandan, "A unified perspective on computational techniques for the measurement of visual motion," in *IEEE Proc. Int. Conf. Computer Vision*, London, UK, 1987, pp. 219-230.
- [61] M. Bierling, "Displacement estimation by hierarchical block matching," in *SPIE Proc. Visual Commun. and Image Process. '88*, Cambridge, MA, Nov. 1988, vol. 1001, pp. 942-951.
- [62] C. S. Fuh and P. Maragos, "Affine models for image matching and motion detection," in *IEEE Proc. ICASSP'91*, Toronto, Canada, May 1991, vol. IV, pp. 2409-2412.
- [63] V. Seferidis and M. Ghanbari, "General approach to block-matching motion estimation," *Opt. Eng.*, vol. 32, no. 7, pp. 1464-1474, July 1993.
- [64] G. Madec, "Half pixel accuracy in block matching," in *Picture Coding Symp. '90*, Cambridge, MA, Mar. 1990.
- [65] S. L. Iu, "Comparison of motion compensation using different degrees of sub-pixel accuracy for interfield/interframe hybrid coding of HDTV image sequences," in *IEEE Proc. ICASSP'92*, San Francisco, CA, Mar. 1992, vol. 3, pp. 465-468.
- [66] H. Watanabe and S. Singhal, "Windowed motion compensation," in *SPIE Proc. Visual Commun. and Image Process. '91*, Boston, MA, Nov. 1991, vol. 1605, pp. 582-589.
- [67] S. Nogaki and M. Ohta, "An overlapped block motion compensation for high quality motion picture coding," in *IEEE Proc. Int. Symp. on Circ. and Syst.*, San Diego, CA, May 1992, pp. 184-187.
- [68] C. Auyeung, J. Kosmach, M. Orchard, and T. Kalafatis, "Overlapped block motion compensation," in *SPIE Proc. Visual Commun. and Image Process. '92*, Boston, MA, Nov. 1992, vol. 1818, pp. 561-571.
- [69] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *IEEE Proc. ICASSP'91*, Toronto, Canada, May 1991, vol. 4, pp. 2713-2716.
- [70] Y. Nakaya and H. Harashima, "An iterative motion estimation method using triangular patches for motion compensation," in *SPIE Proc. Visual Commun. and Image Process. '91*, Boston, MA, Nov. 1991, vol. 1605, pp. 546-557.
- [71] M. H. Chan, Y. B. Yu, and A. G. Constantinides, "Variable size block matching motion compensation with applications to video coding," *IEE Proc.*, vol. 137, no. 4, pp. 205-212, Aug. 1990.
- [72] M. T. Orchard, "Predictive motion field segmentation for image sequence coding," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 2, pp. 54-70, Feb. 1993.
- [73] I. Moccagatta, F. Moscheni, M. Schutz, and F. Dufaux, "A motion field segmentation to improve moving edges reconstruction in video coding," in *IEEE Proc. ICIP'94*, Austin, TX, Nov. 1994, vol. 3, pp. 751-755.
- [74] F. Dufaux, I. Moccagatta, F. Moscheni, and H. Nicolas, "Vector quantization based motion field segmentation under the entropy criterion," *J. Visual Commun. and Image Rep.*, vol. 5, no. 4, pp. 356-369, Dec. 1994.
- [75] F. Dufaux and M. Kunt, "Multigrid block matching motion estimation with an adaptive local mesh refinement," in *SPIE Proc. Visual Commun. and Image Process. '92*, Boston, MA, Nov. 1992, vol. 1818, pp. 97-109.
- [76] W. Hackbusch and U. Trottenberg, Eds., *Multigrid Methods*. New York: Springer-Verlag, 1982.
- [77] R. E. Bank, "A-posteriori error estimates, adaptive local mesh refinement and multigrid iteration," in *Proc. 2nd Europe. Conf. on Multigrid Methods*, Cologne, Germany, Oct. 1985, pp. 7-22.
- [78] R. Hervigo, J. Kowalczyk, and D. Mlynek, "A multiprocessors architecture for HDTV motion estimation system," *IEEE Trans. Consumer Electron.*, vol. 38, pp. 690-697, Aug. 1992.
- [79] J. Kowalczyk et al., "VLSI implementation for a new video codec," in *Proc. EUSIPCO '92*, Brussels, Belgium, Aug. 1992, pp. 1517-1520.
- [80] C. Labit and H. Nicolas, "Compact motion representation based on global features for semantic image sequence coding," in *SPIE Proc. Visual Commun. and Image Process. '91*, Boston, MA, Nov. 1991, vol. 1605, pp. 697-709.
- [81] M. Gilge, "A high quality videophone coder using hierarchical motion estimation and structure coding of the prediction error," in *SPIE Proc. Visual Commun. and Image Process. '88*, Cambridge, MA, Nov. 1988, vol. 1001, pp. 864-874.
- [82] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.*, vol. 38, pp. 477-486, Apr. 1990.
- [83] D. H. Hubel and T. Weisel, "Brain mechanisms of vision," *Scientific Amer.*, pp. 130-146, Sept. 1979.
- [84] T. Ebrahimi, "Perceptually derived localized linear operators: Application to image sequence compression," Ph.D. thesis, Swiss Federal Inst. Tech., Lausanne, Switzerland, 1992.
- [85] R. M. Witten, I. H. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520-540, June 1987.



**Frédéric Dufaux** (Member, IEEE) received the M.Sc. in physics and the Ph.D. in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 1990 and 1994, respectively.

From 1990 to 1994, he was a Research Assistant at the Signal Processing Laboratory at EPFL. During the summer of 1992, he was a Visiting Researcher at the Advanced Video Technology Department of AT&T Bell Laboratories, Murray Hill, NJ. He is currently a postdoctoral fellow at the Media Laboratory of the Massachusetts Institute of Technology, Cambridge, MA. His research interests include image processing, video coding, motion estimation, and image sequence analysis. He is the author or co-author of several research publications.

Dr. Dufaux is a member of EURASIP.



**Fabrice Moscheni** (Student Member, IEEE) received the M.Sc. degree in physics from the Swiss Federal Institute of Technology in Lausanne (EPFL) in 1991.

During 1991–1992 he was a Visiting Researcher at Fujitsu Laboratories Ltd. in Atsugi, Japan. During the summer of 1994 he was with Logitech Inc., Fremont, CA, where he worked on a new digital camera for video conferences. He is currently with the Signal Processing Laboratory in the EPFL where he is working on the Ph.D. His research interests are in region-based image sequence processing and coding, motion estimation, and signal detection.