Dense Estimation of Optical Flow in the Compressed Domain Using Robust Techniques

Konstantinos E. Rapantzikos

A Thesis submitted for the
Degree of Master of Science
in the Department of Electronic & Computer Engineering
Technical University of Crete

Accepted on the recommendation of

Prof. M. Zervakis	supervisor
Prof. E. Christodoulou	advisor
Ast. Prof. E. Petrakis	advisor

September 2002

Acknowledgments

I am greatly indebted to my supervisor, Michalis Zervakis, for his invaluable support during our long cooperation. He guides me since my first student years and i owe him grateful thanks for his warmth and patience. My strong interest for image and video processing aroused by his immerse knowledge and enthusiasm for the subject area.

My heartfelt thanks go to my parents for their love and affection over all these years. Nothing would have been possible without you…

Few people have made the last years a memorable experience. I would like to thank Vangelis for our inspired and in-depth discussions. Special thanks to Anna, Christos, Paraskevi and Petros for the pleasure of being a part of my life. I am not sure if i could make it to the end without their enjoyable, pleasant and creative friendship.

For the sincere and useful discussions we had, i would like to thank Euripides Petrakis, advisor of this thesis.

Terminology

MC	Motion Compensation	(Chapter 2)
MB	MacroBlock	(Chapter 2)
ME	Motion Estimation	(Chapter 2)
MV	Motion Vector	(Chapter 2)
DCT	Discrete Cosine Transform	(Chapter 2)
RLE	Run Length Encoding	(Chapter 2)
MSE	Mean Square Error	(Chapter 2)
MAD	Mean Absolute Distortion	(Chapter 2)
VLC	Variable-Length Code	(Chapter 2)
DPCM	Differential Pulse Coded Modulation	(Chapter 2)
OFCE	Optical Flow Constraint Equation	(Chapter 3)
DFD	Displaced-Frame-Difference	(Chapter 3)
MAP	Maximum A Posteriori	(Chapter 3)
ML	Maximum Likelihood	(Chapter 3)
PDF	Probability Density Function	(Chapter 3)
IRLS	Iterative Reweighted Least Squares	(Chapter 4)
PSM	Pseudo M- Estimator	(Chapter 4)
SOR	Simultaneous Over-Relaxation	(Chapter 4)
GNC	Graduated Non-Convexity	(Chapter 4)
LMedS	Least Median of Squares	(Chapter 4)
LMSOD	Least Median of Squares Orthogonal Distances	(Chapter 4)
WLS	Weighted Least Squares	(Chapter 4)
MAP	Maximum A Posteriori	(Chapter 4)
RRMAP	Reweighted Robust MAP	(Chapter 4)
IF	Influence Function	(Chapter 4)
I	Intra frame	(Chapter 2)
P	predicted frame	(Chapter 2)
B	Bi-directional predicted frame	(Chapter 2)

Chapter 1

1. Introduction

A good part of understanding and impression of the world is based on our sense of vision. Nevertheless, the mechanics that enable vision are not obvious even for the experienced researcher. How do we understand shape? How do we understand the objects' movement? Looking around and recognizing a face or admiring details of a landscape is an amazing accomplishment that is more difficult to achieve than e.g. the mind processing needed to play chess.

Until recently, the function of the human vision was compared to that of a photo-camera [Kandel *et al.*, 1995]: The crystalline eye lens focalize an inverse object's image on the retina, exactly as the lens of a photo-camera. This theory is easily understood to be unsatisfactory, by realizing that under this model assumption a central vision function cannot be explained, namely the depth (3-D) understanding of the surroundings. Additionally, the image projected to the retina is not informative enough to justify our ability to recognize objects under varying illumination conditions.

Even the state of the art vision systems are not able to imitate the human vision. Most of them are nearly efficient in controlled environments or require human interaction. In this thesis, we consider and discuss the concept of motion. The ability to understand the moving world is essential to our survival and is taken as granted for us. If we want to create autonomous platforms (robots) that interact with their environment in an intelligent way, then we have to make them understand motion.

This chapter outlines concepts related to motion in computer vision and gives a brief overview of our approach. We attempt to represent motion in the sequences with possible application in low/high level video segmentation. It is said that a picture is worth a thousand words. If this is correct, we can hardly imagine the worth of a sequence of images…

1.1 Motion & Computer Vision

For many users, *video* is synonymous to television and motion pictures. But, in recent years, we have witnessed the increasing popularity and use of video in and beyond the realms of entertainment, in the form of home movies, education and training, internet, interactive TV. Video sequences and their ability to represent evolution over time are also widely used in medical applications. Great research has been directed towards effective machine vision using motion as the main cue. Video databases are huge and therefore efficient retrieval techniques have become a great challenge. Video retrieval involves content analysis and feature extraction, indexing and querying. Generally, d*igital video segmentation* can be defined as the problem of automatically analyzing video content into meaningful and manageable component units, which may be individual *objects* or reasonable *scenes*.

Motion plays an important role in our visual understanding of the surrounding world. The moving objects are precisely the interesting objects that help us understand the situation. It is needless to say that knowing that "something" is moving in a particular way is often much more important than knowing what this object actually is. Obviously, motion is an informative property and should be incorporated in any machine vision application attempting to achieve a more human-like vision. Video sequences introduce a third dimension to the static world of 2D image space that can be directly related to motion, namely *time*. The translation of objects over time gives us the impression of movement. The notion of motion in image sequences and the attempt to recover is important for both low- and high- level processing. For example, low level processing is related to predictive coding that is widely used in video compression and is mainly based on the temporal redundancy inherent in the batch of available images. Additionally, a close representation of the true underlying motion can help us decompose the sequence into coherently moving objects and understand their interactions. The latter is related to high level processing.

1.1.1 Applications

As indicated before, motion is useful in many computer vision and video analysis tasks. A representative set of applications is given below:

- **Video Coding.** A sequence of pictures can occupy a vast amount of storage space when represented in digital form. For example, suppose the pictures in a sequence are digitized as discrete grids or arrays with 360 pixels per raster line and 288 lines/picture, a resolution that is typical for MPEG. Assuming the picture sequence is in color, a three-color separation (red, green, blue) can be used for each picture. If each color component in the separation is sampled at a 360×288 resolution with 8-bit precision, each picture occupies approximately 311 Kbytes. If the moving pictures are represented uncompressed at 24 frames/s, the raw data rate for the sequence is about 60Mbit/s, and a one-minute video clip occupies 448Mbytes! A very important part of many coding schemes, with MPEG being one of them, is *motion compensation*. Pixels in a region of a reference picture are used to predict pixels in a region of the current frame based on their motion pattern. Differences between the reference picture and the current picture are then coded to whatever accuracy is affordable at the desired bitrate. Therefore, determining which areas of the image are moving and their motion pattern are crucial tasks. [Mitchel *et al.*, 1997]

- **Robotic Vision.** Motion is a valuable source of information for autonomous robots to navigate and interact with their environment (obstacle avoidance, path planning etc). In cases where direct control by a human is not possible, the visual sensor and the computed motion are perhaps the main sources of information used to achieve autonomy. Although navigation is a hard task, there are several techniques developed for "constrained" environments: Robotic arms can perform specific operations on objects passing by on a conveyor belt based on video camera input [Lewis *et al.*, 1993]. Autonomous vehicles are capable of following a road based on specific features extracted from visual processing [Giachetti *et al.*, 1998; Leuven *et al.*, 2001]

- **Video Indexing.** Digital video indexing techniques are becoming increasingly important with the recent advances in very large scale integration technology (VLSI), broadband networks (ISDN, ATM) and video compression standards. The goal of video indexing is to develop techniques that provide the ability to store and retrieve

video sequences based on their content. Few of the potential applications of video indexing are: multimedia information systems [Docherty *et al.*, 1991], digital libraries [Digital libraries, 1995], remote sensing and natural resource management [Ehlers et al., 1989], movie industry and video on demand [Little *et al.*, 1993]. A video stream is composed of video elements constrained by the spatiotemporal piecewise continuity of visual cues. The normally visual motion becomes suddenly discontinuous in the event of new activities or scene changes. Hence, motion discontinuities may be used to mark the inception of a new activity or the change of a scene. [Mandal *et al.*, 1999]

- **Super Resolution.** The large overlap between successive frames and regions in the scene is used to achieve images with a higher spatial resolution. The process of reconstructing a high-resolution image from several images covering the same region in the world is called *Super Resolution*. If a good model of possible degradation is defined, the various moving regions and their approximate motions are accurately computed, a super resolution image can be reconstructed [Irani et al. 1993].

- **Medical imaging.** The interpretation of symptoms or the diagnosis of a doctor can be greatly assisted by motion analysis. It can be used, for example, to monitor the heart movement from MR imagery [Funkalea e*t al.*, 1996] or to interpret ultrasound scans [Quistgaard, 1997].

1.1.2 Compressed Domain

With rapid advances in communication and multimedia computing technologies, the mass amounts of data associated with visual information are a reality on the information highway. As the amount and complexity of video information grow, the need for efficient compression becomes obvious. Video compression is concerned with the reduction of bits required to store or transmit images under the constraint of achieving some target quality. Motion compensation plays an important role as indicated in the previous section.

MPEG compression is widely accepted as a video compression standard and it is adopted for several applications. The MPEG stream carries both motion and intensity information of the underlying scene. Motion is represented by a field of motion vectors and intensity by a set of discrete cosine transform (DCT) coefficients. More details regarding the standard will be given at chapter 2. Processing in the compressed domain reduces the amount of effort involved in full decompression and keeps the storage cost low. In our approach, we only use information that is available in the compressed stream.

1.2 Optical Flow

Dynamic image analysis has focused research on the understanding of motion analysis and representation. The previous discussion outlines and explains the apparent importance of motion in computer vision applications. Although a great amount of work on motion representation has been published, the motion characteristics of objects and how to represent them is still a challenging issue.

As a camera moves, the images of the objects move on the focal plane too. Their motion is the projection of the 3-D motion with respect to the camera coordinate system. *Optical flow*, or "projected motion", is the (perspective or orthographic) projection of this 3-D motion in real world scene on the 2-D image plane. The *motion field* [Horn, 1986] is the field in the image plane that is associated with the spatiotemporal variations of intensity pattern. For most applications, the world has enough structure and the recovered optical flow provides a good approximation to the motion field. If this were not the case, then humans would not be able to perceive motion.

In this thesis, we are dealing with two groups of motion estimation algorithms, namely the gradient-based and block matching motion estimation techniques. The main idea under the gradient-based methods is that the optical flow field can be estimated from the spatiotemporal image gradients by using an appropriate smoothness constraint [Horn *et al.*, 1981]. The optical flow is recovered by minimizing a functional on the spatiotemporal variation of data with the additional smoothness constraint. These methods are referred to as *regularization* techniques. The assumption of block-matching techniques is that the pixels of a small image block exhibit the same motion from frame to frame. Therefore, the same motion vector is assigned to all pixels within this block.

The motion vectors are chosen so that they either maximize correlation or minimize error between a block and a corresponding array of pixel values in a reference frame.

1.2.1 Constraints on Motion

Regularization techniques are based on assumptions about the objects' motion. These assumptions are usually ideal and are often violated in real scenes. They are expressed as constraints on the objective function.

Three motion constraints are often used in the literature, namely the *data*, *smoothness* and *temporal* constraints. The *data* constraint (or brightness/intensity conservation) states that the intensity measurements corresponding to a surface change slowly over time. The *smoothness* constraint (or spatial coherence) states that the surfaces have spatial extent and hence neighboring points on a surface will have similar motion. The *temporal* constraint (or temporal coherence/continuity) is based on the fact that the velocity of a surface changes gradually over time.

1.3 Problems in Motion Analysis

All motion analysis and representation algorithms face common problems that arise mainly from the 3-D to 2-D projection of the real motion field. The most important of them are briefly reviewed below.

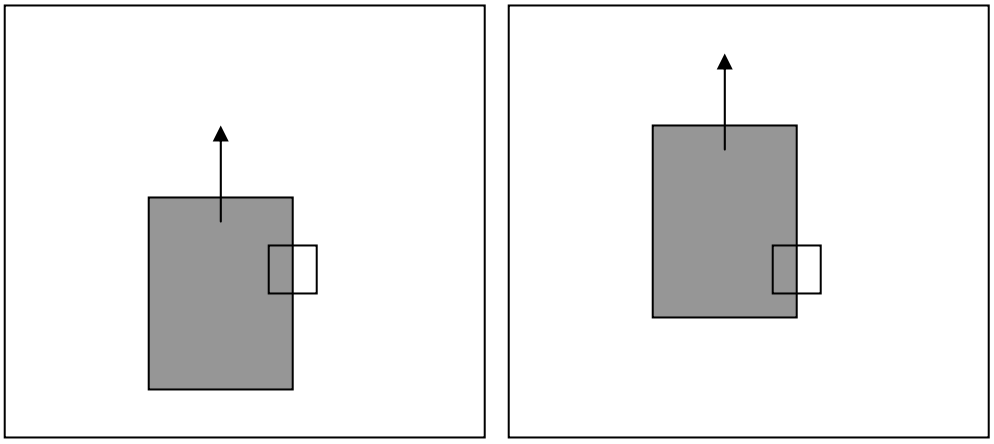


Fig. 1.1 Aperture Problem: Although the gray rectangle is moved upwards we cannot observe it through the small square (aperture)

When the gray level variation in the image does not uniquely constrain the functional to be minimized, i.e. more than one candidate motion fits the functional equally well, the *aperture problem* occurs. To illustrate this further let us study a simple example. In Fig. 1.1, a gray rectangle moves upwards and we observe it through the small square aperture. Although the gray rectangle moves upwards, we cannot recognize its motion through the small aperture. In the contrary, if we observe the movement through an aperture located at a corner, we can recognize the motion accurately. Indeed, optical flow is best determined at the corners, i.e. when there is enough gray scale variation.

Motion discontinuities form another difficulty for the motion estimation algorithms. When a depth discontinuity occurs in the scene, e.g. overlapping objects or abrupt direction change, several points in the 3-D are projected to the same point in the 2-D space. This is called a spatial motion field discontinuity. In this case, a common assumption, namely the single motion of small regions, is violated and the correct recovery of the velocity becomes difficult. Most algorithms estimate an averaged motion at these problematic regions, which results in inaccurate optical flow.

Temporal discontinuities can also occur due to *occlusions*/*disocclusions* of objects or change of scene. In these cases, there is no data correspondence between neighboring frames and therefore the optical flow is undefined. Motion has to be derived by other means in these areas, for example using a three-frame matching, making the solution difficult and perhaps not efficient.

1.4 Approach

In this thesis we propose an algorithm for optical flow estimation that is thoroughly based on information that us directly available in the compressed domain. Hence, we preserve the advantages of compressed domain processing and improve the existent MPEG velocity field in terms of accuracy and density (1 motion vector/pixel).

We address the issues of robust, incremental, dense optical flow estimation by combining information from two different velocity fields: the available MPEG motion field that is generated by a block-matching and the generated motion field by a robust regularization technique. The regularization technique is based only on information that is directly available in the compressed stream avoiding therefore the time and memory

consuming decompression. Both motion estimation techniques, namely the block-matching and gradient-based, have their own problems in regions with specific characteristics (homogenous, noisy, with motion borders etc). We attempt to develop an efficient method that combines only their advantages over these regions in order to recover the true underlying motion as correct as possible.

Our work can be seen as an extension of Black & Anandan's work [Black *et al.*, 1996] towards dense optical flow recovery in compressed video and use of additional constraints on image motion. The robust estimation framework for motion estimation was first explored by Black & Anandan in an earlier work [Black *et al.*, 1993]. We use their formulation and incorporate new terms in the objective function by using the MPEG motion field and exploiting temporal information in a different way.

1.5 Thesis Overview

The rest of this thesis is devoted to describe the previously published work, to introduce and explain the necessary mathematics and examine in detail the developed approach.

Chapter 2. The MPEG standard and its components are reviewed. The extraction and manipulation of information that is available in the MPEG video without the need of full decompression, namely the DC coefficients and the motion vectors, is considered.

Chapter 3. The most popular techniques for motion estimation with emphasis on intensity differential and block matching techniques are examined. Constraints on motion (data conservation, spatial coherency and temporal continuity) are presented under a least-squares regularization formulation. Additionally, coarse-to-fine methods designed to compensate for large displacements are discussed. Constraint violations and the aperture problem are also elaborated.

Chapter 4. Robust statistics are briefly reviewed and the robust optical flow estimation framework used by [Black *et al.*, 1996] is presented. The minimization technique is described and provided in terms of pseudocode.

Chapter 5. In this chapter, we elaborate on dense optical flow recovery using robust regularization based on the DC coefficients and the motion vectors of the MPEG stream. New constraints on the objective function are introduced and ideas regarding

current and future work are discussed. The developed approach is systematically presented.

Chapter 6. Experimental results are presented and several problematic cases are illustrated. Improvements of our approach over OFC and MPEG fields are shown and are analyzed using various examples.

Chapter 7. Evaluation and discussion of our approach is attempted along with possible future directions.

Chapter 2

2. MPEG Information Extraction & Motion Vectors Manipulation

Recent advances in multimedia compression technology, coupled with the significant increase in computer performance and the growth of the Internet, have led to the widespread use and availability of digital video. Applications such as digital libraries, distance learning, video-on-demand, digital video broadcast, interactive TV, multimedia information systems generate and use large collections of video data [Docherty *et al.*, 1991; Digital libraries, 1995; Bhatt *et al.*, 1997; Chang *et al.*, 1997. This has created the need for tools that can efficiently classify and retrieve relevant material. Automatic classification of video sequences would increase usability of these masses of data by enabling people to search quickly and efficiently multimedia databases.

There are three main sources of information in video: first the audio; secondly the individual images which can be classified by their content; thirdly, the dynamics of the image information held in the time sequence of the video. It is the last attribute that makes video classification different from image classification. The most successful approaches to video segmentation/classification are likely to use a combination of static and dynamic information.

Two forms of dynamic information can be identified: foreground and background motion. The foreground motion is related to object motion, while the background motion is related to camera motion. A complete video processing system should separate these two motion signals in order to assess the classification potential of each one individually.

Currently, a large part of video material is in compressed form due to recent advances in video compression (H.261, MPEG). For compressed video, processing typically starts with decompression. Operations on fully decompressed or uncompressed video do not permit rapid processing because of the data size. It is thus advantageous to

develop algorithms to operate directly on compressed data without having to first perform full frame decompression. In this thesis, we present a technique for recovering optical flow for MPEG sequences. Information that is available in the MPEG video, without the need of full decompression, is combined with gradient information form the DC images to achieve this task. More details will be given in the next chapters. The following sections briefly review the MPEG standard and describe the techniques for DC & Motion Vectors extraction and manipulation.

2.1 Video Compression & MPEG Coding

Video sequences contain a significant amount of data redundancy within and between frames. The ultimate goal of video source coding is the bit-rate reduction for storage and transmission by exploring the redundancies to perform encoding of only a "minimum set" of information using entropy-coding techniques. This usually results in a compression of the coded video data compared to the original source data. The performance of video compression techniques depends on the amount of redundancy contained in the image data as well as on the actual compression techniques.

The coding of the video data may be "lossless" or "lossy" depending on the application. The input to the data compressor is usually called *source* data and the output of decompression forms the *reconstructed* data [Mitchel *et al.*, 1997]. Some compression techniques are designed such that reconstructed data and source data exactly match, and these techniques are called "*lossless"*. Other techniques provide only good (hopefully) approximations to the source data, something relevant to the MPEG video standards. These techniques are called "*lossy*". The aim of lossy techniques is to optimize image quality for a specific target bit rate subject to given optimization criteria. The next subsections introduce the basic aspects of the MPEG video standard. Information is mainly drawn from [Mitchel *et al.*, 1997; **ISO**/IEC, 1993; **ISO**/IEC, 1996]

2.1.1 The MPEG Model

The key aspect of moving picture compression is the similarity between pictures in a sequence. This similarity is "expressed" by statistical redundancies in both temporal and spatial directions. The d*iscrete cosine transform* (DCT) is used to compensate for

spatial redundancy, while a *motion compensation* (MC) scheme based on a block matching method compensates for the temporal redundancies. An entropy encoder is used afterwards to code the generated symbols from the encoder model in a process that minimizes the bitstream length in a statistical sense. The basic statistical property upon which MPEG compression techniques rely is inter-pixel correlation, including the assumption of simple translatory motion between consecutive frames. Thus, it is assumed that the magnitude of a particular image pixel can be predicted from nearby pixels within the same frame (using Intra-frame coding techniques) or from pixels of a nearby frame (using Inter-frame techniques). High compression needed by MPEG applications is achieved by coding most of the pictures as differences relative to neighboring pictures (Inter-frame compression): The parts of the image that do not change significantly are simply copied from other areas or other frames. Other parts may be best predicted by parts of the image that are displaced because of motion. The latter requires the use of motion compensation to capture temporal redundancy. Intuitively it is clear that under some circumstances, i.e. during scene changes of a video sequence, the temporal correlation between pixels in nearby frames is small or even vanishes - the video scene then assembles a collection of uncorrelated still images. In this case, Intra-frame coding techniques are appropriate to explore spatial redundancy in each image in order to achieve efficient data compression.

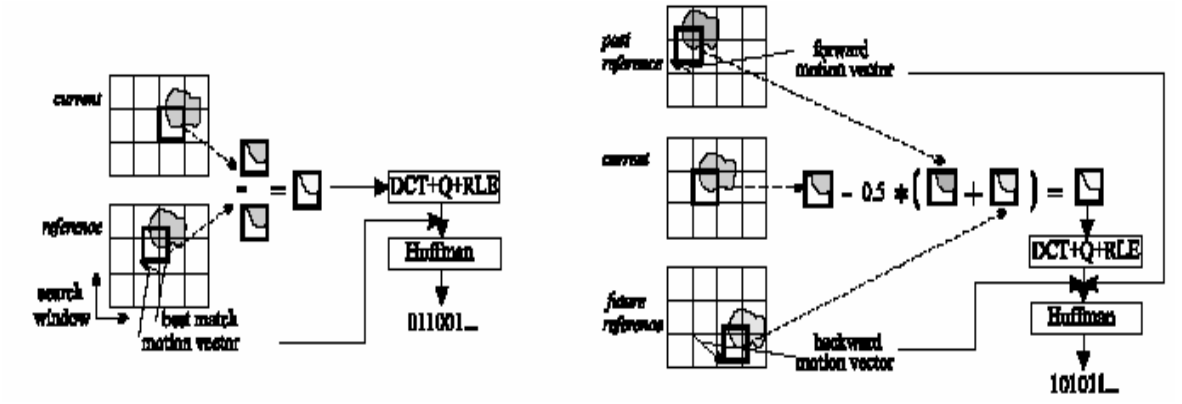
The smallest image unit of MPEG coding is a block. A block represents an 8×8 pixel group of the original image. The MPEG family algorithms employ block-based compression by applying the DCT on image blocks. In case of intra-frame compression the result is similar to JPEG compression. The basic building block of an MPEG frame is the *macroblock* (MB). The MB consists of a 16×16 array of luminance (grayscale) samples together with one 8×8 block of samples for each of two chrominance (color) components. The 16×16 sample array of luminance samples is actually composed of four 8×8 blocks of samples.

Under these guidelines, an MPEG stream consists of three types of pictures: *I*, *P* and *B*. *Intra* (I) frames provide random access points into the compressed data and are coded using only information present in the picture itself by the DCT, quantisation and Huffman entropy coding. As indicated, the DCT is responsible for reducing spatial

redundancy of the picture to be encoded. The DCT does not directly reduce the number of bits required to represent the block. The reduction in the number of bits follows from the observation that, for typical blocks from natural images, the distribution of coefficients is non-uniform. The transform tends to concentrate the energy into the low-frequency coefficients and many of the other coefficients are near zero. The bit rate reduction is achieved by not transmitting the near-zero coefficients and by quantising and coding the remaining coefficients as described below. The non-uniform coefficient distribution is a result of the spatial redundancy present in the original image block. The function of the coder is to transmit the DCT block to the decoder in a bit rate efficient manner, so that it can perform the inverse transform to reconstruct the image. It has been observed that the numerical precision of the DCT coefficients may be reduced while still maintaining good image quality at the decoder. *Quantisation* is used to reduce the number of possible values to be transmitted, reducing the required number of bits. The degree of quantisation applied to each coefficient is weighted according to the visibility of the resulting quantisation noise to a human observer. In practice, this results in the high-frequency coefficients being more coarsely quantised than the low-frequency coefficients. Note that the quantisation noise introduced by the coder is not reversible in the decoder, making the coding and decoding process 'lossy'.

The serial arrangement and coding of the quantised DCT coefficients exploits the likely clustering of energy into the low-frequency coefficients and the frequent occurrence of zero-value coefficients. Each block is scanned in a diagonal zigzag pattern starting at the DC coefficient to produce a list of quantised coefficient values, ordered according to the scan pattern. The list of values produced by scanning is entropy coded using a Variable-Length Code (VLC). Variable length codes are needed to achieve good coding efficiency, as very short codes must be used for the highly probable events. The VLC allocates code words, which have different lengths depending upon the probability with which they are expected to occur. To enable the decoder to distinguish where one code ends and the next begins, the VLC has the property that no complete code is a prefix of any other. Huffman coding is used to generate the tables of variable length codes needed for this task.

The first DCT coefficient of each block is called DC term and is 8 times the average intensity of the respective block. P (*predicted*) frames are coded with *forward* motion compensation using the nearest previous reference (I- or P-) images. *Bi-directional* (B) pictures are also motion compensated, this time with respect to both past and future reference frames. For each MB of the current frame, the encoder finds the best matching block in the respective reference frame(s), calculates and DCT-encodes the residual error and transmits one or two motion vectors, see Fig. 2.1(a), (b). During the encoding process, a test is made on each MB of P and B frame to see if it is more expensive to use MC or intra- coding. The latter occurs when the current frame does not have much in common with the reference frames. As a result, each MB of a P frame could be coded either intra or forward while for each MB of a B frame there are four possibilities: intra, forward, backward or interpolated. Interpolated motion-compensated prediction is achieved by the simultaneous use of both forward and backward motion-compensated prediction. The prediction is a simple average of the pixel values from the forward and backward motion-compensated reference pictures. A possible reason for using interpolated prediction is to average the noise in the two reference pictures, thereby improving the prediction.



(a) (b)

Fig. 2.1 (a) Forward prediction for P frames; (b) Interpolated prediction for B frames; figure from [Koprinska *et al.*, 2001]

2.1.2 Motion Compensation

Motion compensated prediction is a powerful tool to reduce temporal redundancies between frames and is used extensively in MPEG video coding standards. The concept of motion compensation is based on the estimation of motion between frames, i.e. if all elements in a video scene are approximately spatially displaced, the motion between frames can be described by a limited number of motion parameters (i.e. by motion vectors for translatory motion of pixels). In this simple example, the best prediction of an actual pixel is given by a motion compensated prediction pixel from previously coded frames. Usually both, prediction error and motion vectors, are transmitted to the receiver. A trade-off must be made between the accuracy in predicting complex motion in the image and the expense of transmitting the motion vectors. Smaller regions require more complex estimation incorporating techniques such as noise smoothing. To this end, images under the MPEG format are separated into disjoint macroblocks (MB) of pixels (i.e. 16x16 pixels in MPEG-1 and MPEG-2 standards) and only one motion vector is

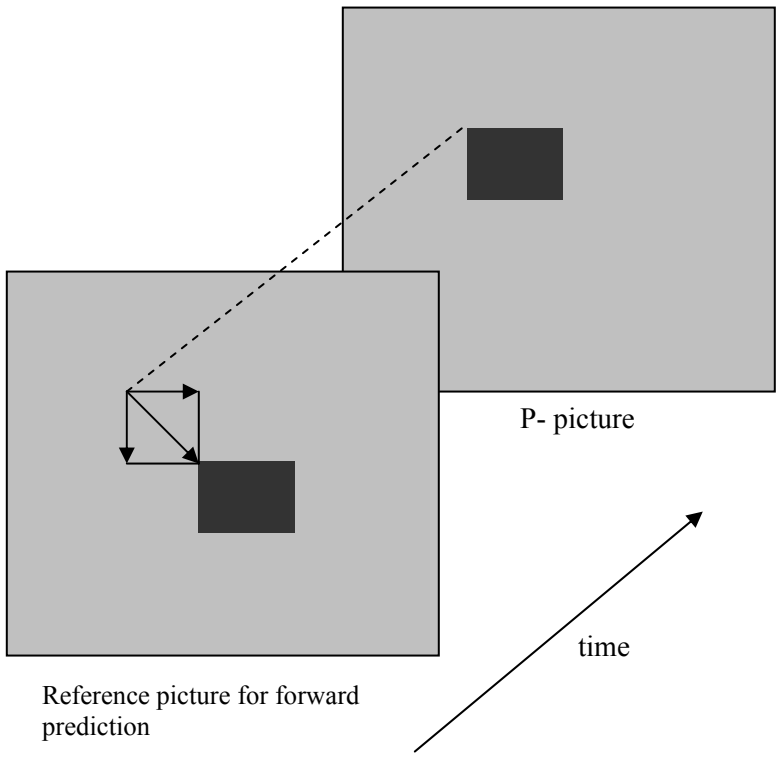


Fig. 2.2 P- picture motion vector displacements. Positive displacements are to the right and down, relative to the macroblock being coded. [Mitchell et al., 1997]

estimated, coded and transmitted for each of these macroblocks.

The P- frames use forward motion-compensated prediction, so named because the predicted pixels are projected forward in time from earlier I- or P- frames in the sequence (Fig.2.2). Each MB has one MV associated with it.

B-frames may use either forward or backward motion-compensated prediction or both (Fig. 2.3). In backward motion-compensated prediction the reference picture occurs later in the sequence. Forward motion-compensated prediction is done much the same as in P- frames.

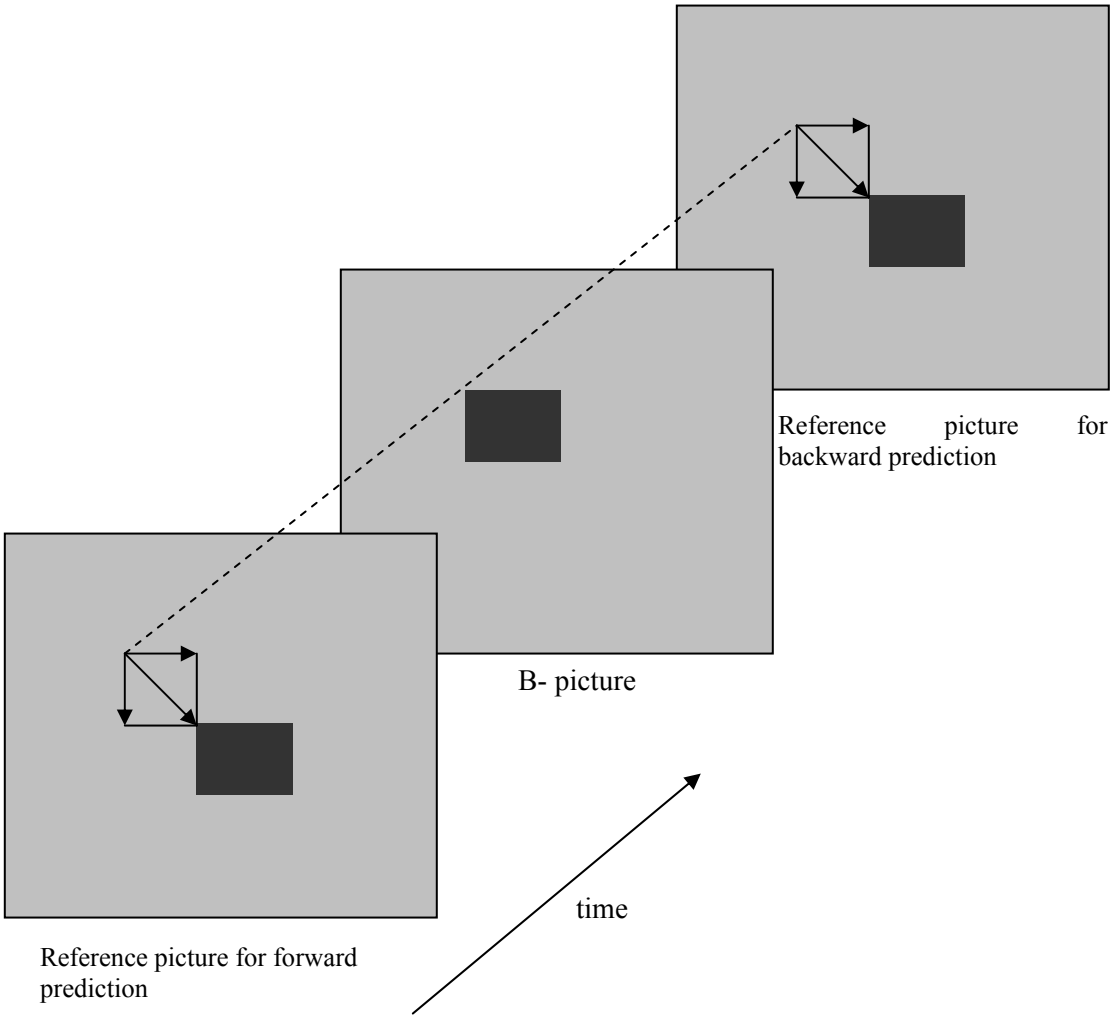


Fig. 2.3 B- picture motion vector displacements. Positive displacements are to the right and down, relative to the macroblock being coded. [Mitchell et al., 1997]

2.2 DC Extraction

The Discrete Cosine Transform (DCT) is an essential part of the MPEG standard. It is used as the basis of compression for I frames and of the residue images from P and B frames. The MPEG uses a 2-D 8x8 DCT for each block. The forward and reverse DCT are defined as:

$$F(u,v) = \frac{C(u)}{2}\frac{C(v)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} f(x,y)\cos[(2x+1)u\,\pi/16]\cos[(2y+1)v\,\pi/16] \quad (2.1)$$

$$f(x,y) = \sum_{y=0}^{7}\sum_{x=0}^{7}\frac{C(u)}{2}\frac{C(v)}{2}F(u,v)\cos[(2x+1)u\pi/16]\cos[(2y+1)v\,\pi/16] \quad (2.2)$$

where *u*, *v* are the horizontal and vertical frequency indices, respectively, and the constants, *C(u)*, *C(v)*, are given by:

$$C(u) = {}^{1}\!\!\diagup\!\!_{\sqrt{2}}, \text{ if u=0}$$

$$C(u) = 1, \qquad \text{if u>0}$$

DCT coefficients are ordered using a zigzag traversal pattern, run length and then Huffman coded [Mitchell *et al.*, 1997].

Given an image *f* of size $N \times M$ we define a *DC image* f_{DC} to be a reduced one of size $\frac{N}{2^3} \times \frac{M}{2^3}$. Therefore, the image f_{DC} is reduced 8 times in each direction. This corresponds to using one pixel to represent an 8×8 block. Every pixel of the derived image equals the DC coefficient, *F(0,0)*, of each block. A *DC sequence* is a sequence of such DC images. Although much smaller than *f,* the f_{DC} and consequently the DC sequence retain a significant amount of global information present in the scene.

The DC is coded in a different way than the rest DCT coefficients in order to be more easily accessible. Therefore, after the DC coefficient of a block has been quantised to 8 bits, it is coded losslessly by a differential pulse coded modulation (DPCM) technique. In this coding technique, a difference is calculated between each pixel and a prediction calculated from neighboring pixel values already transmitted. At the decoder, the original quantised DC values are exactly recovered by following the inverse procedure.

DCT coefficients are readily accessible for I frames, but they must be estimated for P and B frames. To calculate them, the DCT coefficients of the 16×16 area of the reference frame that the current block was predicted from need to be calculated. Since the DCT is a linear transform, the DCT coefficients of this reference MB in the reference

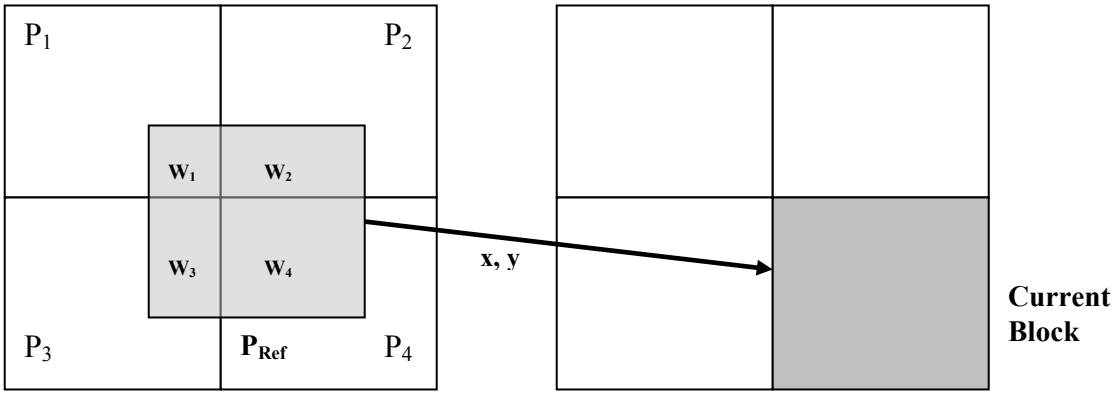


Fig. 2.4 Reference block (P_{Ref}), motion vector and original blocks.

frame can be calculated from the DCT coefficients of the four MBs that overlap this reference MB as shown in Fig. 2.4, albeit with substantial computational expense. Yeo & Liu proposed a technique for calculating reasonable approximations to the DC coefficients of a MB of a P or B frame. We adopt their method and provide a short description of their work. More details can be found in [Yeo *et al.*, 1995a].

To reduce the computation of reconstruction of DC values they propose two methods, to be called *zero-order* and *first-order* approximations. For the zero-order approximation, they take the DC value from the block, which has the most overlap with block P_{ref} (see Fig. 2.4). For the first order approximations they weigh the contributions from the 4 neighboring DC values with the ratio of the overlaps of the block P_{ref} with each of the block $P_1,\ldots,P_4$:

$$DC(P_{\mathrm{Re}f}) = \sum_{i=1}^{4} w_i \times DC(P_i), \tag{2.3}$$

where w_i is the ratio of the area of the shaded region of P_i to its total area, as shown in Fig. 2.4. If a MB of a B frame is interpolated from two reference MBs, its DC coefficient is approximated by an average of the estimated DC coefficients of each of these two MBs. Fig 2.5 and Fig. 2.6 shows images at their original resolution and the corresponding

first order DC images. Additionally, Fig. 2.6 shows a DC image spatially scaled to the uncompressed frame's dimensions, in order to get a visual grip of the information loss in such an image.

Fig 2.5 Original image and first-order DC image

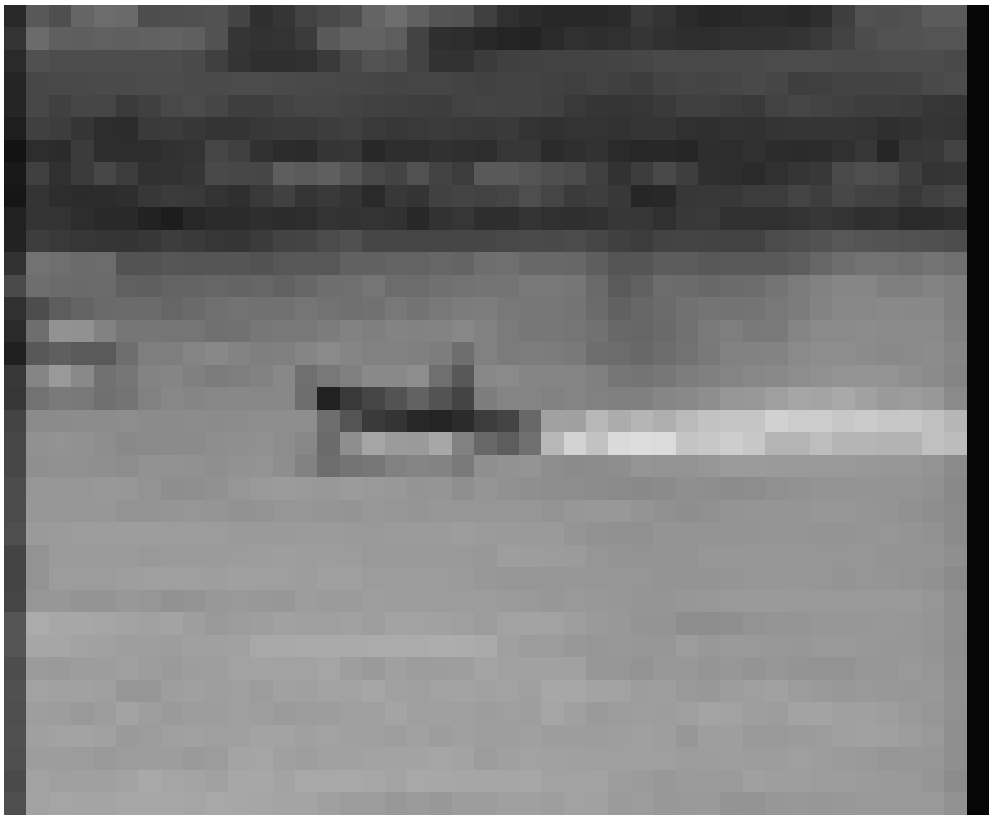


Fig 2.6 Original image; first-order DC image; DC image spatially scaled to uncompressed frames dimensions

2.3 Motion Vector Extraction & Manipulation

As indicated in section 2.1, the frames of a MPEG video stream may be of different types, i.e. I-, P or B-, and can occur in a variety of GOP (Group Of Pictures) patterns. An I frame has no motion vectors assigned to it in contrast to P (one MV max for every MB) or B (two MVs max for every MB) frames.

We adopt the approach of Kobla *et al.* [Kobla *et al.*, 1997] to produce a unified set of motion vectors that is independent of the frame type and the direction of prediction. Their method represents each motion vector as a backward predicted vector with respect to the next frame, independently of frame type. In other words, the motion vectors of each frame represent the direction of motion of each MB with respect to the next frame.

The method recovers the flow between every pair of frames by treating every pair according to its type. There are seven possible types, namely IP, which is an I frame followed by a P and PP, IB, PB, BI, BP, BB combination. For IP or PP the flow is derived as follows: The flow for the first frame is simply the set of forward predicted motion vectors of the following P frame after inversion. In other words, if a MB in the P frame is displaced by a motion vector (x, y) with respect to a MB in the I or P frame, then it is logical to conjecture that the latter MB is displaced by a motion vector ($-x$, $-y$) with respect to the MB in the P frame. For clips containing B frames let us consider two consecutive reference frames, R_i and R_j. If the B frames between them are denoted by $B_1,\ldots,B_n$, where n is the number of B frames between the reference ones, then we have the following options:

- $\boldsymbol{R_iB_1}$: The flow for R_i is derived by using the inverse forward predicted motion vectors of B_1.
- $\boldsymbol{B_nR_j}$: The flow for R_j is derived by using the backward predicted motion vectors of B_n. There is no need to invert he motion vectors here.
- If a MB in a B frame does not have a forward or backward motion vector we look at successive/preceding B frames till we find a corresponding MB in frame B_k with valid forward/backward motion vector. Since this vector is predicted from k frames earlier/after, we scale it down by k.
- ***BB***: Obviously, there is no direct interaction between consecutive B frames. Flow between successive B frames is derived by analyzing corresponding MBs in

those B frames and their motion vectors with respect to their reference frames. Since each MB in each B frame can be any of three types, namely, forward-predicted (F), backward-predicted (B) or bidirectionally-predicted (D), there exist nine possible combinations; FF, FB, FD, BF, BB, BD, DF, DB and DD. Each of these nine combinations is considered individually.

More details about the uniform motion vector representation can be found in [Kobla *et al.*, 1997].

Chapter 3

3. Estimating Optical Flow

Optical flow is a 2D motion measure, which has a wide range of applications in computer vision, video coding and computer graphics. Its efficient estimation is a hard task and has been studied widely during the last decades. A host of different motion estimation algorithms have been proposed based on different ideas. They fall broadly into three categories: 1) Gradient-based methods that use spatial and temporal derivatives at each point in the image and impose some further constraint to uniquely identify the motion. 2) Region-matching methods, in which the *displaced-frame-difference* (DFD) or some similar error criterion is minimized over a set of local regions by employing an appropriate form of search mechanism. 3) Pel-recursive methods that are iteratively refining motion estimation for individual pixels by gradient techniques. They involve more computational complexity and less regularity.

Gradient-based estimation has become very popular in computer vision applications. The main reasons are that it can be computationally efficient and produces a dense (one MV/pixel) motion field estimate. Such methods require motion constraints that can be classified as global or local in nature. Therefore, the optical flow estimation is often expressed as a global optimization problem that involves local or global constraints, where the issue is to find a global minimum of a cost function (or *energy*) involving the data and the "hidden" variables of interest to be extracted from the data. Usually, a first part of the energy expresses the interaction between the unknown variables and the data (*observation*), while the second one captures some kind of prior knowledge (*prior*) about the unknown motion field. The essential role of the second part is to regularize and constrain the first one. Such approaches give rise to the so-called *regularization techniques* that have received a great deal of attention.

Three motion constraints are often used in the literature, namely the *data*, *smoothness* and *temporal* constraints. The *data* constraint (or brightness/intensity

conservation) states that the intensity measurements corresponding to a surface change slowly over time. The *smoothness* constraint (or spatial coherence) states that the surfaces have spatial extent and hence neighboring points on a surface express similar motion. The *temporal* constraint (or temporal coherence/continuity) is based on the fact that the velocity of a surface changes gradually over time.

The following sections examine the most popular techniques for motion estimation with emphasis on intensity differential and block matching techniques. Additionally, coarse-to-fine methods designed to compensate for large displacements are discussed in section 3.1.5. The latter topic will be explored in more detail at the next chapter. Constraint violations and the aperture problem are also elaborated in section 3.3. Ad hoc techniques using a robust estimation framework are examined at the next chapter.

3.1 Intensity-based differential methods

Differential techniques compute image velocity from spatiotemporal derivatives of image intensities. The image domain is therefore assumed to be differentiable in space and time. The most popular methods use additional information like smoothness regularization or temporal coherence terms to increase solution accuracy and computational efficiency. Such constraints are often violated in real sequences. A robust, in terms of efficient results, technique should take into account these violations and compensate for them in order to produce a "good" motion field estimate.

The following three sections provide information regarding the nature of the constraints used in optical flow estimation. Section 3.2.4 reviews the techniques used so far.

3.1.1 Optical Flow Constraint Equation (OFCE)

The usual starting point for velocity estimation is to assume that the intensities are shifted (locally translated) from one frame to the next, and that the shifted intensity values are conserved, i.e.

$$I(x, y, t) \approx I(x + u\delta t, y + v\delta t, t + \delta t), \tag{3.1}$$

where[1] u, v denote the horizontal and vertical optical flow vector components and δt is small. This constraint implies that the intensity of a moving point in the image plane remains constant along the trajectory of the point in time as shown in Fig. 3.1. It is needless to say that such an assumption is only approximately true in practice. Methods

Fig. 3.1 Data Conservation: Although it has moved, the highlighted region on the right looks roughly the same with the region on the left.

making direct use of this constraint in areas of the image are called matching-based. Due to computational difficulties, they can yield unsatisfactory accuracy [Barron *et al.*, 1994]. For this reason gradient-based methods have become popular. They use the Taylor Series approximation of (3.1) and yield:

$$I(x+u\delta t, y+v\delta t, t+\delta t) = I(x,y,t) + \nabla I(x,y,t)\cdot \mathbf{u}\delta t + \delta t I_t(x,y,t) + O^2, \quad (3.2)$$

where in $\nabla I = (I_x, I_y)$, the subscripts indicate partial derivatives of the brightness function with respect to x and y. I_t indicates its partial derivative over time and O^2 stands for the 2nd and higher order terms. Dropping the terms above first order, the data conservation constraint gives the standard *Optical Flow Constraint Equation* (OFCE):

$$I_x u + I_y v + I_t = 0, \quad (3.3)$$

[1] The mathematical symbols are in correspondence with those from Black *et al.* [Black et al., 1996a] throughout this thesis

It is obvious that it is impossible to recover velocity, given just the gradient constraint at a single position, since Eq. (3.2) provides a single equation with two unknowns, i.e. u, v. As shown in Fig. 3.2, motion vectors satisfying Eq. (3.2) are constrained along a line in (u, v) space. Only the component of velocity in the gradient direction that is *normal* to the spatial image orientation is determined [Horn, 1986]. This is referred to as the *aperture* problem and may be understood by considering an edge of an object moving below a small aperture. More details are given in section 3.3.2.

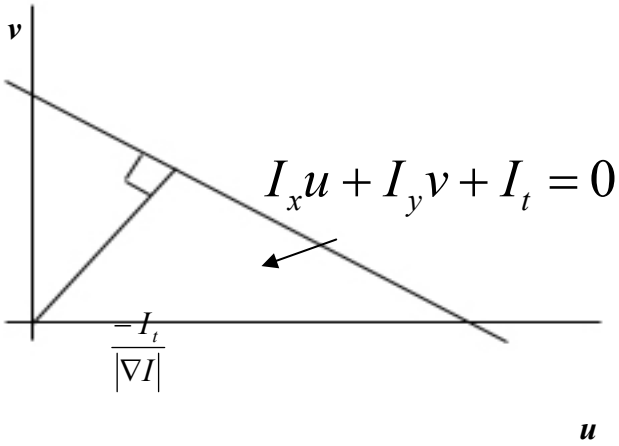


Fig. 3.2 The data constraint equation constrain velocity to lie somewhere on a line in 2d velocity space

To recover an estimate of the optical flow at a point, one should simply minimize the following term by taking into account a small neighborhood:

$$E_D(u,v) = \sum_{(x,y)\in\Re} (I_x(x,y,t)u + I_y(x,y,t)v + I_t(x,y,t))^2 , \tag{3.4}$$

which is actually the standard least-squares estimate as described by Horn & Shunck [Horn *et al.*, 1981]. This regression formulation assumes that a single motion exists in this small neighborhood and can be considered as a local motion estimation method.

3.1.2 Bayesian Framework & Regularization Techniques

The ill-posed nature of the OFCE leads to the unavoidable need for further constraints to achieve a unique solution. This consideration fits well within the *maximum a posteriori*

(MAP) framework that provides a way of incorporating prior information (constraints) in the solution recovery procedure. To illustrate this, let $\mathbf{x} = \{x_1,...,x_T\}$ denote a given set of *T observation* vectors, where $x_1,...,x_T$ are either independent and identically distributed or are drawn from a probabilistic function. The difference between MAP and *maximum likelihood* (ML) estimation lies in the assumption of an appropriate prior distribution of the parameters to be estimated. If **θ**, assumed to be a random vector taking values in the space Θ, is the parameter vector to be estimated from the sample **x** with probability density function (PDF) $f(\cdot \mid \boldsymbol{\theta})$, and *g* is the *prior* PDF of θ, then the MAP estimate, θ_{MAP}, is defined as the mode of the posteriori PDF of θ denoted as $g(\cdot \mid \mathbf{x})$, i.e.

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\theta} g(\boldsymbol{\theta} \mid \mathbf{x}) = \arg\max_{\theta} f(\mathbf{x} \mid \boldsymbol{\theta})g(\boldsymbol{\theta}) \tag{3.5}$$

If **θ** is assumed to be fixed but unknown, then there is no knowledge about **θ**, which is equivalent to assuming a non-informative prior or an improper prior, i.e. *g(**θ**)=constant*. Under such an assumption, Eq. 3.5 reduces to the familiar ML formulation. In general, the *observation* is only related to the current available information, while the *a priori* constraint is related to the fact that we know something about the data without even observing them.

In terms of optical flow, the *observation* part could be related with the probability of a prediction error-image and the *a priory* part could be formulated by quantifying prior expectations on the estimation. Widely used prior "expectations" are discussed at subsequent sections, while new ideas regarding such motion constraints are discussed in chapter 5.

Various low-level vision problems can be solved by regularization, a powerful tool that reaches a solution by approximating given observations. The problem with this approach lies on the oversmoothed solutions at discontinuities. Many researchers have proposed methods to alleviate this artifact. Section 3.1.6 reviews these methods. Generally, the regularization converts ill-posed problems into well-posed ones by constraining the solution with *a priori* assumption; exactly as in the MAP formulation. The energy function $J(\theta, \alpha)$ in Tikhonov's regularization framework is defined by [Tikhonov et al., 1977; Bertero et al., 1988; Sim *et al.*, 1998]

$$J(\boldsymbol{\theta},\alpha) = \mathrm{D} + \alpha \mathrm{S} = \sum_{\mathrm{x_i} \in \mathbf{x}} \left\| A_i \vec{\theta}_{\mathrm{i}} - \mathrm{x_i} \right\|^2 + \alpha \mathrm{S},$$

where S and D represent smoothness (see Section 3.1.3) and data terms, respectively. The smoothness on the solution depends on α and can be adaptively determined according to *a priori* knowledge. By minimizing this energy function, the solution $\boldsymbol{\theta}_{\mathrm{reg}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta} \mid \alpha)$ is obtained.

It becomes obvious that MAP estimation formally looks very similar to regularization. Their fundamental difference should therefore be emphasized. In regularization theory, the chosen smoothness constraint must assure that optimization of the resulting criterion is a *well-posed* problem. In the Bayesian framework, both terms cannot be chosen arbitrarily, but must reflect probability distributions [Stiller, 1997]. In this thesis, we follow a regularization approach emphasizing on the smoothness constraint(s).

3.1.3 Spatial Coherence Constraint

One way to constrain (regularize) the solutions derived by the data constraint equation is to invoke a smoothness assumption. Local gradient methods constrain only partially the solution, as indicated before, and are very sensitive to noise deriving bad solutions in cases of areas with little variation in texture. In the original sense of Hadamard, [Revalski, 1997], 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, considering 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.

The introduction of the spatial coherence constraint makes the optical flow estimation problem *well-posed* by implying that the OFCE holds locally within some neighborhood. It assumes that the flow within a neighborhood changes gradually or, in other words, neighboring points in the scene typically belong to the same surface and therefore have similar velocities as shown in Fig. 3.3. This further assumes that there is only a single motion within a confined range. However, optical flow is not totally continuous but is

Fig. 3.3 Smoothness Constraint: Constant motion is assumed for a small neighborhood

only piecewise smooth since depth boundaries in the scene may give rise to discontinuities in the flow. This assumption is commonly violated at the borders of a moving object. If this assumption is falsely imposed, then motion estimation error can occur at object boundaries (Fig. 3.3).

To recover an estimate of the optical flow at a certain point, one should simply minimize Eq. (3.3), in the least-squares sense, with the addition of the regularizing term E_S:

$$E(\mathbf{u}) = \lambda_D E_D(\mathbf{u}) + \lambda_S E_S(\mathbf{u}) = \lambda_D (I_x u_s + I_y v_s + I_t)^2 + \lambda_S E_S(\mathbf{u}), \qquad (3.5)$$

where λ_D and λ_S control the relative importance of the data conservation and spatial coherence terms. Eq. (3.5) is the mathematical expression of our expectation that the optical flow will minimize any violations of the OFCE , and at the same time, will minimize the magnitude of velocity changes between neighboring pixels.

The most common formulation of E_S is *the first-order*, or *membrane*, model [Horn et al., 1981]

$$E_s(u,v) = u_x^2 + u_y^2 + v_x^2 + v_y^2,$$

where the subscripts indicate partial derivatives in the x or y direction. For an image of size $n \times n$ pixels, we define a grid of sites,

$$S = \left\{ s_1, s_1, ..., s_{n^2} \mid \forall 1 \le w \le n^2, 0 \le i(s_w), j(s_w) \le n-1 \right\},$$

where *(i(s),j(s))* denotes the pixel coordinates of site *s*. The first-order constraint can then be discretized as:

$$E_S(\mathbf{u}) = \sum_{s \in S} \left[\frac{1}{8} \sum_{n \in G_s} \left[(u_s - u_n)^2 + (v_s - v_n)^2 \right] \right],$$

where the subscripts *s* and *n* indicate sites in *S* and where G_s is the set of 4-connected neighbors of *s* in the grid.

3.1.4 Temporal Continuity

Another assumption used for regularizing optical flow is the temporal persistence of

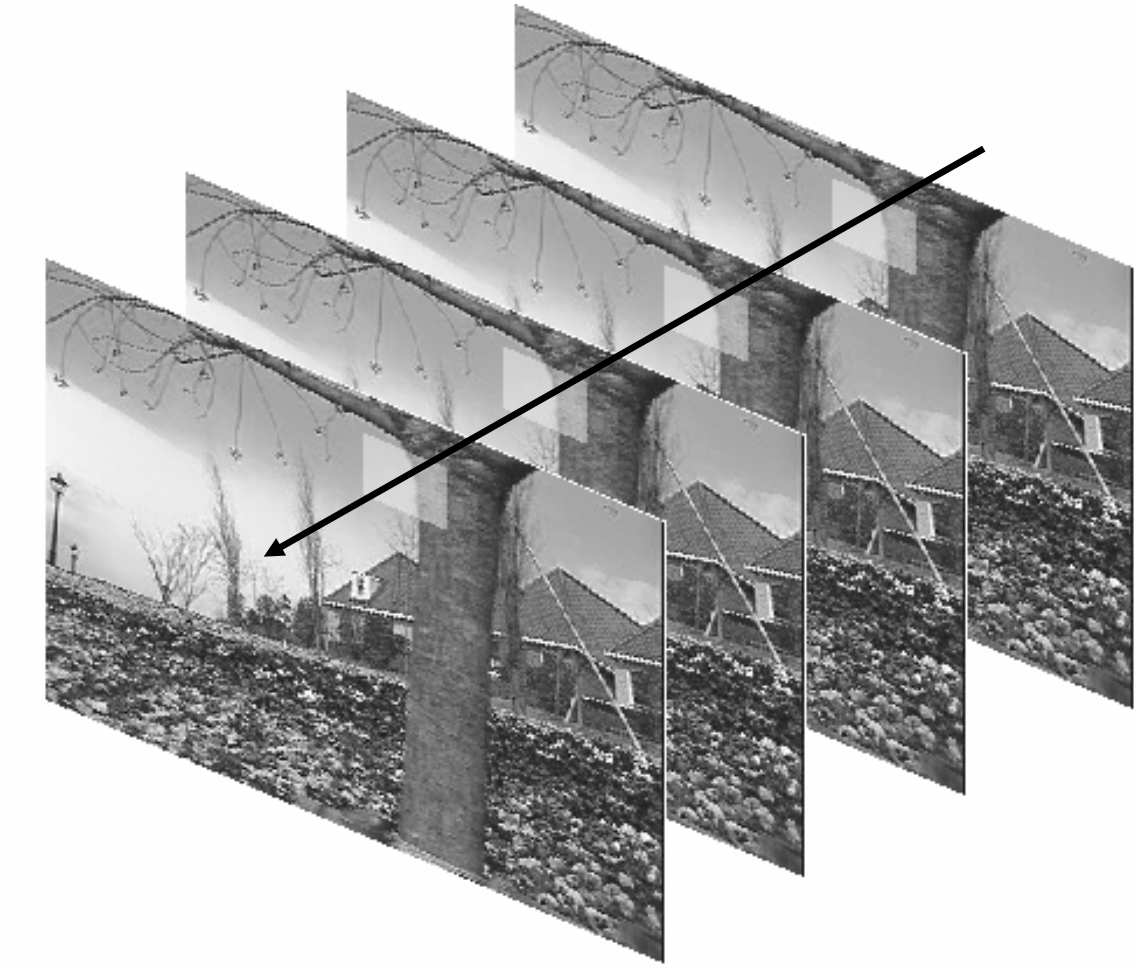


Fig. 3.4 Temporal Continuity: A small neighborhood is assumed to have constant velocity or acceleration over time.

surfaces. It assumes that the motion of a surface changes gradually over time. Black & Anandan [Black *et al.*, 1990; Black *et al.*, 1991; Black, 1994] first exploited the power of this constraint to integrate motion information over time in an incremental estimation framework, where motion vectors are projected from one to the next instant and incrementally adjusted based on frame differences.

Incremental approaches have gained strong interest during the last decade, because they are more suited to the dynamic nature of motion estimation. Towards this direction, Black [Black, 1994] uses a temporal continuity constraint under a general incremental minimization framework to obtain more accurate information about the motion in the scene over time. He uses the simple assumption that the acceleration of a surface is constant over time (Fig. 3.4). From a mathematical point of view, this assumption states that we can predict the flow at the next instant, t, from *t-1* as follows:

$$u^{-}(x,y,t) = u(x-u\delta t, y-v\delta t, t-\delta t) + \frac{\partial}{\partial t}u(x-u\delta t, y-v\delta t, t-\delta t)\delta t\,, \tag{3.6}$$

where u^{-} is the predicted flow field and the acceleration is approximated by

$$\frac{\partial}{\partial t}u(x,y,t) \approx \frac{u(x,y,t)-u^{-}(x,y,t)}{\delta t}\,. \tag{3.7}$$

To recover an estimate of the optical flow that consistently evolves over time one should simply minimize Eq. (3.5) with the addition of the temporal continuity term E_T:

$$E(\mathbf{u}) = \lambda_D E_D(\mathbf{u}) + \lambda_S E_S(\mathbf{u}) + \lambda_T E_T(\mathbf{u}, \mathbf{u}^{-})\,, \tag{3.8}$$

The last term force the solution to be close to the prediction. This constraint, like the previous two, is often violated in real scenes.

3.1.5 Coarse-to-fine Processing

A common problem in optical flow estimation is temporal aliasing. Video imagery is typically sampled below the Nyquist rate in time. Although, newer technology and more sophisticated hardware can circumvent this inherent shortcoming of frame acquisition, the problem persists when, for example, an object undergoes a large motion. In this case the OFCE, Eq. (3.3), becomes inappropriate. A number of authors have developed coarse-to-fine processing strategies for handling the temporal aliasing in the context of motion estimation [Anandan, 1989; Enkelmann *et al.*, 1988; Lucas *et al.*, 1981]. These

algorithms are efficiently implemented by using image pyramids. The image pyramids are generated by Gaussian or Laplacian methods [Anandan, 1989; Battiti *et al.*, 1991; Enkelmann, 1988; Glazer, 1987; Burt *et al.*, 1983]. Because of the low frequency representation at coarse resolution, the OFCE becomes applicable in the case of small image motions at the coarsest resolution [Kearny *et al.*, 1987]. The basic concept in these approaches is that the aliasing affects only the high frequency component of the input image. Thus, one can accurately estimate image velocities on a spatially lowpass-filtered version of the input image (coarse level). These estimates may then be used as initial guesses for initializing the motion at the next (finer) levels. The process is repeated recursively until we reach the finest level (initial resolution).

For efficiently overcoming the effect of temporal aliasing in the estimation of the optical flow field at time t (from frame t to $t+1$) the frame t at coarse resolution is projected to the next level with the appropriate pixel displacement $(u\delta t, v\delta t)$ estimated at the current level l.

$$I^l(x,y,t) \xrightarrow{projection} \tilde{I}^{l+1}(x+u\delta t, y+v\delta t, t)$$

Thus, from one level to the next, the current frame is motion aligned (compensated) with the next frame so that the OFCE does not engage the entire motion field from t to $t+1$, but only its correction $(\delta u^l, \delta v^l)$ that is revealed at the current level of resolution. Fig. 3.6 summarizes graphically the procedure.

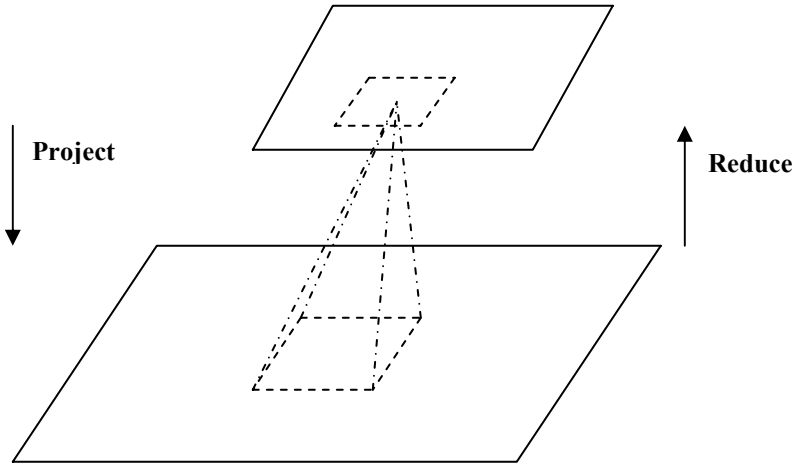


Fig.3.5 Image Pyramid

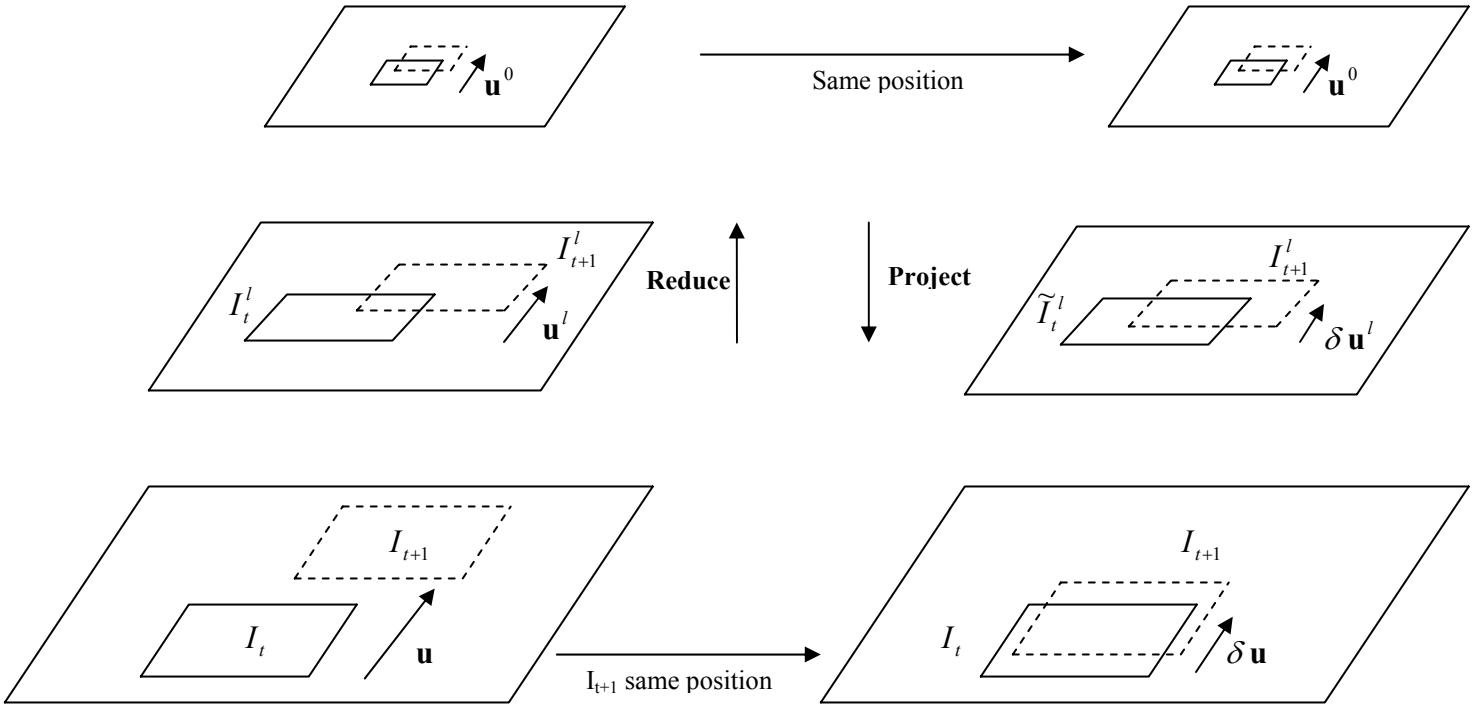


Fig. 3.6 $\widetilde{I}_t$ moved/compensated from motion vectors at previous levels (see text)

3.1.6 Literature Review

Regularization by requiring a slow varying optical flow field was first introduced by Horn & Shunck [Horn *et al.*, 1981]. The solution for *u*, *v* are given as a set of Gauss-Seidel equations that are solved iteratively. Lucas & Kanade [Lucas *et al.,* 1981] use a local constant velocity model and solve it with a weighted least squares fit of local first-order constraints by minimizing

$$\sum_{(x,y)\in\Re} W^2(\nabla I(x,y,t)\cdot u + I_t(x,y,t))^2 \,, \qquad (3.9)$$

where $\mathbf{u} = [u,v]^T$. Simoncelli *et al.* [Simoncelli *et al.*, 1991] present a Bayesian perspective of Eq. (3.9). They model the OFCE, Eq. (3.3), using Gaussian distributed errors on gradient measurements and a Gaussian distributed prior on velocity vector **u**. The resulting solutions are not more accurate than those from Lucas & Kanade, but their technique provides a confidence measure on unreliable estimates [Barron *et al.*, 1994].

Nagel was one of the first to use second-order derivatives to measure optical flow [Nagel, 1983; Nagel, 1987; Nagel *et al.*, 1989]. As an alternative to Eq. (3.5), Nagel

suggested an oriented-smoothness constraint in which smoothness is not imposed across edges in an attempt to handle occlusion.

Several researchers use line processes [Geman *et al.*, 1984; Black *et al.*, 1996b] to explicitly model the motion discontinuities. Konrad *et al.* [Konrad *et al.*, 1992] model the motion and the discontinuity fields as a pair of coupled MRFs and minimize the resulting energy function by means of stochastic relaxation. Identifying the need to exploit intensity discontinuities to detect motion discontinuities, they propose a potential function for the line field that depends on the local image gradient.

Stiller [Stiller, 1997] develops a stochastic image sequence model and use it to unsupervised Bayesian estimation of dense motion fields and their segmentation. The stochastic image sequence model includes two main components. First, the prediction error and second the *a priori* distribution of the motion field. The first component's distribution is modeled by a white generalized Gaussian. The second component's distribution is modeled by a compound MRF accounting for small spatial bindings as well as for bindings along motion trajectories. Based on this model, the MAP criterion is formulated as an objective function.

3.2 Region Level Motion Estimation (region-based matching)

Accurate numerical differentiation may be impractical because of noise. The natural alternative is region-based matching [Anandan, 1989; Little *et al.*, 1989]. Block matching, a special case of area-based matching, is one of the most popular estimation schemes used in video coding. As indicated in chapter 2, it is the central part of the motion compensation technique used in MPEG standards.

When determining the optimal motion displacement of the prediction, a full search is guaranteed to produce the best possible value. This assumes, however, that the criterion for optimality is known and that the computational resources needed for a full search are available. Generally, displacements are chosen that either maximize correlation or minimize error between a macroblock and a corresponding array of pixel values in the reference frame. Correlation calculations are computationally expensive and therefore error measures such as mean square error (MSE) and mean absolute distortion (MAD) are more commonly used. MAD is perhaps the simplest and most accepted

measure of best match [Musmann *et al.*, 1985]. MAD for a 16×16 pixel macroblock (MPEG) is defined as:

$$MAD(x,y)=\frac{1}{256}\sum_{i=0}^{15}\sum_{j=0}^{15}\left|V_n(x+i,y+j)-V_m(x+dx+i,y+dy+j)\right|, \qquad (3.1)$$

where $V_n(x+i,y+j)$ is the current pixel array at macroblock position *(x ,y)* and $V_m(x+dx+i,y+dy+j)$ is the corresponding array of pixels in the next frame at macroblock position *(x+dx, y+dy)*. The 16×16 array in frame *m* is displaced horizontally by *dx* and vertically by *dy*. By convention, *x*, *y* refer to the upper left corner of the macroblock, indices *i*, *j* refer to values to the right and down and displacements *dx*, *dy* are positive when to the right or down. To speed up the search procedure many methods have been proposed, such as the *2D-logarithmic search* [Jain et al., 1981], the *three-step search* [Koga *et al.*, 1981] and the *conjugate direction search* [Srinivasan *et al.*, 1985].

While block-matching proves to be very successful for most macroblocks, there are cases when the search fails. At long edges of the image and in uncovered smooth background areas, there is often no unique prediction, which means that the estimated motion is not necessarily representative of the true motion. The latter consideration is closely related to the aperture problem discussed in section 3.3.2. If the motion is greater than the search range, the search will also fail. By inference, one can "trust" the motion vectors obtained by such an algorithm, but has to pay attention to certain cases.

3.2.1 Literature Review

Anandan introduced a technique based on a Laplacian pyramid and a coarse-to-fine SSD (Sum of Squared Differences) based matching strategy [Anandan, 1989]. The Laplacian pyramid [Burt *et al.*, 1983] allows the computation of large displacements between frames and helps to enhance image structure, such as edges. The mathematical expression of SSD is as follows:

$$SSD_{1,2}(\mathbf{x};\mathbf{d})=\sum_{i=-n}^{n}\sum_{i=-n}^{n}W(i,j)[I_1(\mathbf{x}+(i,j))-I_2(\mathbf{x}+\mathbf{d}+(i,j))]^2=W(\mathbf{x})*[I_1(\mathbf{x})-I_2(\mathbf{x}+\mathbf{d})]^2$$

where *W* denotes a discrete 2-D window function, ***x****=[x, y]* and ***d****=(dx, dy)* take on integer values. Under such approaches, the velocity vector is defined as the shift ***d*** that yields the best fit between image regions at different times.

Their algorithm is briefly described in [Barron *et al.*, 1994]. They begin at the coarsest level, where displacements are assumed to be 1 pixel/frame or less. SSD minima are first located to pixel accuracy by computing SSD values in 3×3 search space using a 5×5 Gaussian for *W(**x**)*. Subpixel displacements are then computed by finding the minimum of a quadratic approximation to the SSD surface (about the minimum SSD value found with integer displacements). A smoothness constraint is also applied on the velocity estimates. Matching and smoothing are performed at each level of the Laplacian pyramid. When moving from coarser to finer levels they use an overlapped projection scheme. The initial 3×3 SSD search area is determined by projecting the coarser level estimate at each pixel to all pixels in a 4×4 region at the next finer level so that each pixel at the finer level has four initial guesses.

A simple area-matching algorithm that uses an exhaustive search over a small neighborhood, across the previous *n* frames is proposed by Camus in [Camus, 1997]. They determine the correct motion of a patch of pixels by simulating the motion of the patch for each possible displacement and consider a match strength for each displacement. If φ represents a matching function, which returns a value proportional to the match of two given features (such as the absolute difference between two pixels' intensity values), then the match strength *M* for a point *(x, y)* is calculated as:

$$M(x,y)=\sum_{x,y}\phi\ \left(\left\|I_t(x,y)-I_{t+\delta t}(x+\delta \mathrm{x},\mathrm{y}+\delta y)\right\|\right).$$

The actual motion of the pixel is taken to be that of the particular displacement, out of (2n+1)*(2n+1) possible displacements, with the maximum neighborhood match strength.; thus it is called a "winner-take-all" algorithm.

3.3 Constraint Violations & Aperture Problem

3.3.1 Constraint Violations

The short description of commonly used constraints for optical flow estimation reveals that such constraints are often violated in real scenes. A robust motion estimation method should cope with such problems in order to properly recover the optical flow.

The interpretation of intensity variation as pure relative motion as expressed by the OFCE is problematic, because velocity is a geometric quantity independent from illumination conditions. Regarding the data conservation constraint one can come to following conclusions [Memin *et al.*, 1998; Black et al., 1996a]:

- The first order Taylor series expansion used, assumes locally constant translational motion
- The assumption of brightness constancy is commonly violated in cases of occlusion, transparency, specular reflection, change of illumination (e.g. shadows), non-rigid movement etc.
- When multiple motions exist within a region, the constraint does not hold. The single motion assumption is violated.

Realistically, the OFCE conditions are never entirely satisfied in scenery. The degree to which these conditions are satisfied partly determines the accuracy with which optical flow approximates image motion. Several authors have addressed the problems arising from non-uniform illumination with more or less success [Bergen *et al.*, 1992; Fleet *et al.*, 1990; Jepson *et al.*, 1993; Mukawa, 1990; Tull *et al.*, 1996].

The most obvious violation of the spatial coherence constraint is at motion discontinuities. The implicit assumption that the velocities within a region are constant is often not even true for apparently smooth areas. Most of the real velocity fields exhibit such motion discontinuities that tend to be ignored and smoothed out by the quadratic estimate (Eq. (3.4)).

The motion of objects in the surrounding world is often not so predictable. They abruptly change direction, rotate, stop etc. Consequently, the temporal continuity constraint is often invalid. In addition, the constraint cannot cope with

occlusion/disocclusion situations. In these cases, an object suddenly appears or disappears leading to discontinuities of the natural optical flow.

3.3.2 Aperture Problem

As identified in section 3.1.1 the optical flow is not uniquely determined by the OFCE as shown in Fig 3.1. The fundamental reason for this is what is called the *aperture*

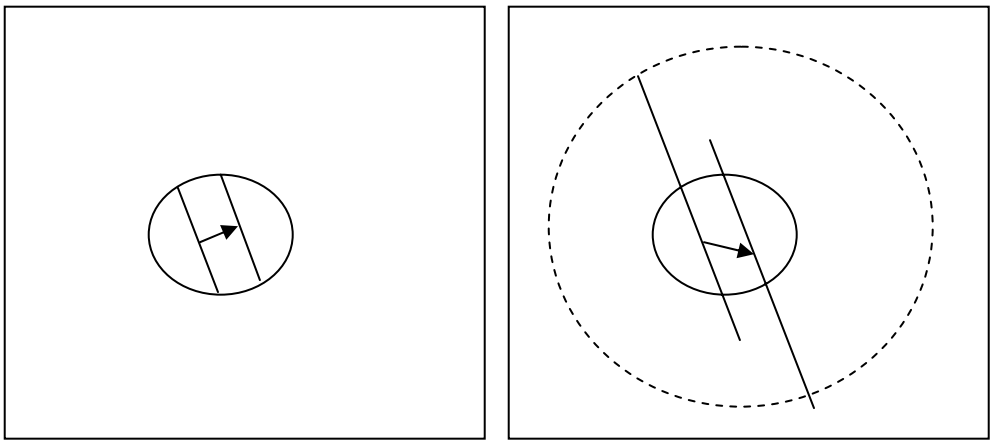


Fig. 3.7 Aperture Problem: If we look through the small aperture we see a line moving to the indicated direction. If we expand the aperture, though, we observe a different direction.

problem. Consider one point in the image. We are computing the gradient in a small window around this point, the aperture. Within this small window, the intensity varies in the direction of the gradient, but not in the direction perpendicular to the gradient. In terms of edges: the intensity varies across the edge but not along the edge. As a result, a motion that is parallel to the edge can never be recovered. In chapter 1, we show a simple example with the moving gray rectangle that illustrates the aperture problem. To illustrate this further let us study another example. In Fig. 3.7, we observe a moving line through a small circular aperture. It appears as it has moved along the indicated direction (arrow) in the left picture. Only if we "open" our eyes a little further we can observe and determine the correct movement as in the right picture.

By considering the aperture problem, a great trade-off problem arises: Estimation of optical flow involves the pooling of constraints over some spatial neighborhood. Since the image is prone to noise, the region must be sufficiently large enough to robustly and

accurately estimate the solution. However, a larger region of integration is more likely to contain multiple motions. That is, the region must be small to avoid violating assumptions such as single motion for the region. This is the so-called *generalized aperture problem*. Therefore, deciding how large a chosen region should be remains a very hard problem.

3.4 Comparison of Motion Estimation Methods

In an attempt to briefly present the advantages and disadvantages of the motion estimation methods discussed before, namely the differential and block-matching in MPEG, we make the following comments:

- Both suffer from the aperture problem, but differential techniques give smoother vector fields and do not allow sharp MV changes in smooth areas.
- Block matching proves better in edges at the expense of possibly drastic MV changes on edge, when aperture problem is severe.
- Differential techniques perform equally well with block-matching at large motions (within the search range of block-matching).
- The dynamic nature of differential methods makes them more flexible, because of the continuously improving estimate they recover.
- The sparseness of the MPEG vector field renders it almost inappropriate for accurate motion segmentation.

Chapter 4

4. Robust Estimation Framework & Optical Flow

4.1 Robust Statistics

The field of robust statistics [Hampel *et al.*, 1986; Huber, 1981] has been developed to address the fact that strict distributional model assumptions are not supposed to be exactly true. As Huber indicates, "*such assumptions are mathematically convenient rationalizations of an often fuzzy knowledge or belief. As in every other branch of applied mathematics, such rationalizations or simplifications are vital, and one justifies their use by appealing to a vague continuity or stability principle: a minor error in the mathematical model should cause only a small error in the final conclusions*."

Unfortunately, this does not always hold. For example, the typically assumed normality of errors in a parametric formulation provides access to standard statistical tools for drawing conclusions about parameters of interest, but there may not be any guarantee that such a regularity assumption is tenable in a given context. The study on the effects of departures from model assumptions led to the development of robust statistics.

Various definitions of greater or lesser mathematical rigor are possible for the term "robustness". In general, referring to a statistical estimator, it means "*insensitive to small departures from the idealized assumption for which the estimator is optimized*"[Huber, 1981]. The word "small" can have two different interpretations, both important: either fractionally small departures for all data points, or else fractionally large departures for a small number of data points. It is the latter interpretation, leading to the notion of *outlier* points, which is generally the most stressful for statistical procedures. The idealized distribution model is usually the Gaussian, since it is the most important case and the best understood one.

As identified by Huber [Huber, 1981] the main goals of a robust procedure are:

1. It should have a reasonably good (optimal or nearly optimal) efficiency at the assumed model.
2. It should be robust in the sense that small deviations from the model assumptions should impair the performance only slightly, that is, the latter should be close to the nominal value calculated at the model.
3. Somewhat larger deviations from the model should not cause a catastrophe.

4.1.1 Measures of robustness

The first step in describing robust estimators is to state clearly what is meant by robustness. Several measures of robustness are used in the literature. Most common is the *breakdown point* [Rousseeuw *et al., 1987*] —the minimum fraction of outlying data that can cause an estimate to diverge arbitrarily far from the true estimate. For example, the breakdown point of least squares is 0 because one bad point can be used to move the least squares fit arbitrarily far from the true fit as shown in Fig. 4.1. The theoretical maximum breakdown point is 0.5 because when more than half the data are outliers they can be arranged so that a fit through them will minimize the estimator objective function.

A second measure of robustness is the *influence function* [Hampel *et al.*, 1986; Huber, 1981], which, intuitively, is the change in an estimate caused by insertion of outlying data as a function of the distance of the data from the (uncorrupted) estimate. For example, the influence function of the least-squares estimator is simply proportional to the distance of the point from the estimate. To achieve robustness, the influence function should tend to 0 with increasing distance.

Finally, although not a measure of robustness, the *efficiency* of a robust estimator is also significant. This is the ratio of the minimum possible variance in an estimate to the actual variance of a (robust) estimate [48], with the minimum possible variance being determined by a target distribution such as the normal (Gaussian) distribution. Efficiency clearly has an upper bound of 1.0. *Asymptotic efficiency* is the limit in efficiency as the number of data points tends to infinity. Robust estimators having a high breakdown point

tend to have low efficiency, so that the estimates are highly variable and many data points are required to obtain precise estimates.

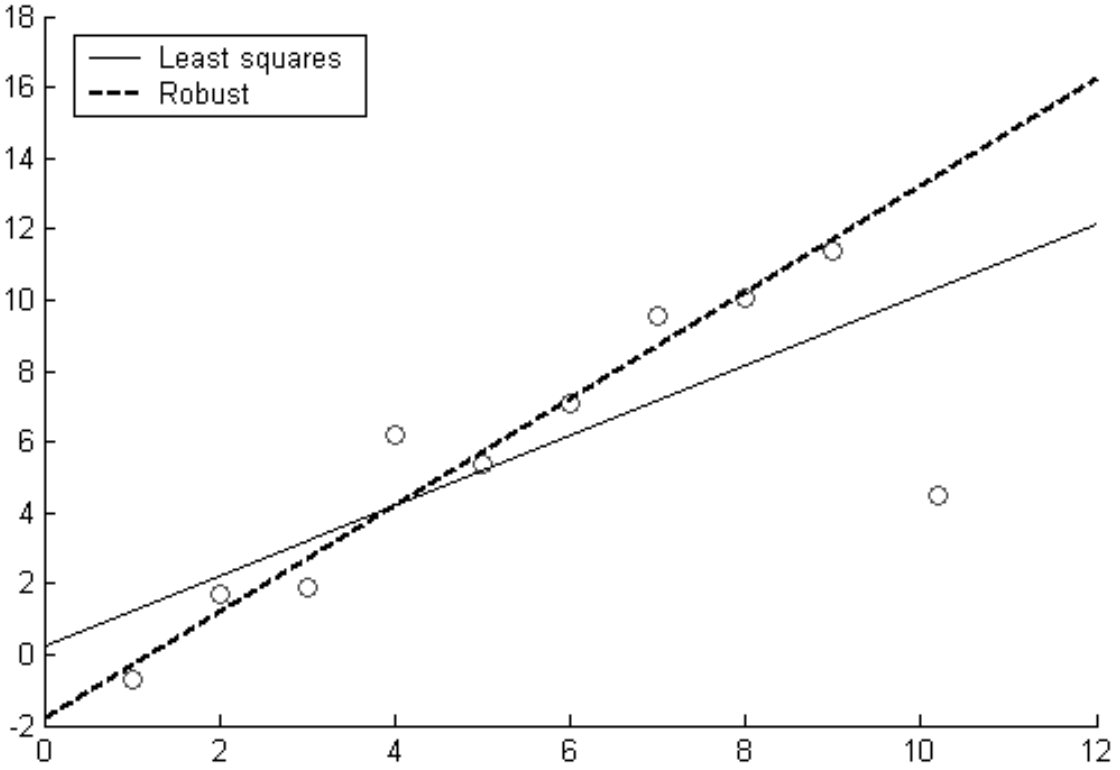


Fig. 4.1 A 2D distribution fitted to a straight line; non-robust techniques such as least-squares fitting can have undesired sensitivity even to a single outlier.

4.1.2 Mathematical Framework & Robust Estimators

To state the issue more concretely, robust statistics addresses the problem of finding the values for the parameters, $\mathbf{a} = [a_{0,\ldots,}a_n]$, that best fit a model, $\mathbf{u}(s;\mathbf{a})$, to a set of data measurements, $\mathbf{d} = \{d_0, d_1, \ldots, d_s\}$, in cases where the data differs statistically from the model assumptions. In fitting a model the goal is to find the values for the parameters, **a**, that minimize the size of the *residual errors* $r = (d_s - \mathbf{u}(s;\mathbf{a}), \sigma_s)$:

$$\min_{\mathbf{a}} \sum_{s \in S} \rho(r_s, \mathbf{a}), \qquad (4.1)$$

where σ_s is a scale parameter, which may or may not be present, and ρ is our *estimator*. When the errors in the measurements are normally distributed, the optimal estimator is the quadratic:

$$\rho(r_s, a) = \frac{((r_s, \mathbf{a}))^2}{2\sigma_s^2}, \qquad (4.2)$$

which gives rise to the standard least-squares estimation problem. The function ρ is called an *M-estimator* since it corresponds to the Maximum-likelihood estimate.

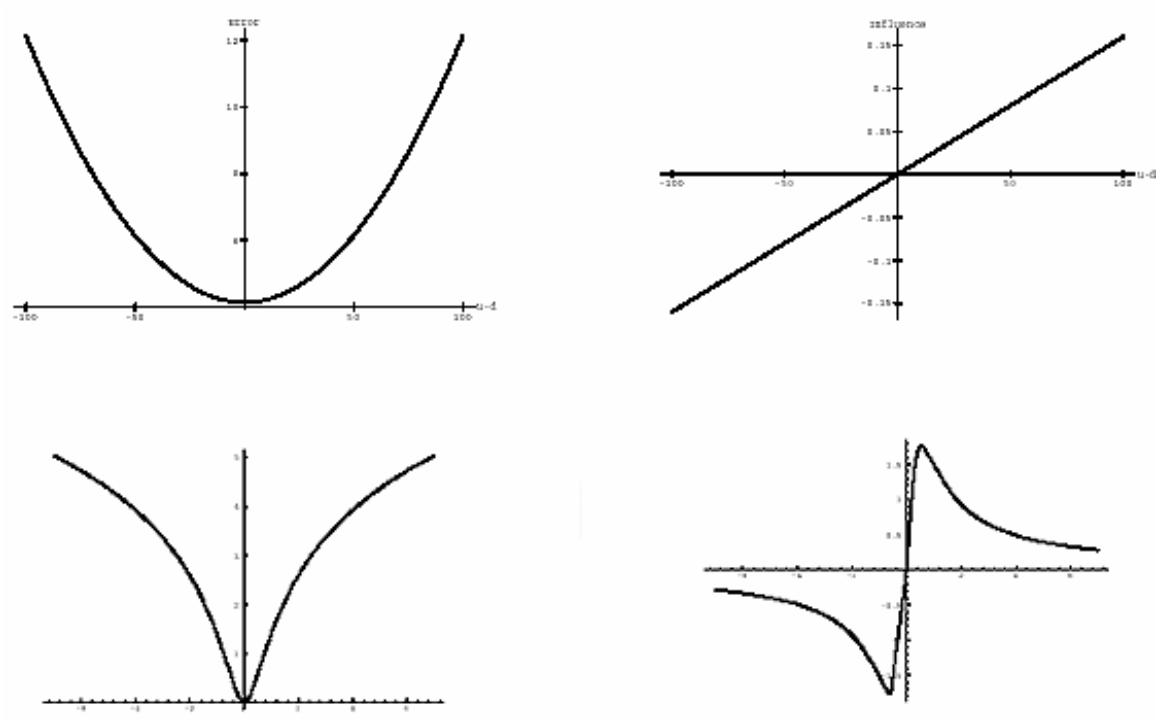


Fig. 4.2 (α) Quadratic estimator and ψ- function; (b) Lorentzian estimator and ψ- function

To analyze the behavior of an estimator we will look at the "influence function" (IF), [Hampel *et al.*, 1986], approach. The IFs measure the effect of a single data point on parameter estimates. The influence function of estimator ρ is defined as its derivative $\psi(x) = \frac{\vartheta\rho}{\vartheta x}$. For the quadratic estimator, we have $\psi(r) = \frac{r}{\sigma^2}$: note that this grows without bound as *r* increases (Fig. 4.2). Therefore, any single measurement can have an arbitrarily large effect on the estimate. For example: for least-squares, the mean is estimated as

$$\bar{x} = \frac{1}{N}\sum_{1}^{N} x_i \,.$$

If we add one more point, x_{N+1}, we have

$$\bar{x}' = \frac{1}{N+1}\sum_{1}^{N+1} x_i = \frac{1}{N+1}x_{N+1} + \frac{N}{N+1}\bar{x}\,.$$

Since x_{N+1} can take on any value, it can have an arbitrarily large effect on the estimate of the mean.

To increase robustness we will consider a *redescending* estimator for which the influence of outliers tends to zero, namely the *Lorentzian*:

$$\rho(r) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right), \quad \psi(r) = \frac{2r}{2\sigma^2 + r^2}\,. \tag{4.3}$$

The Lorentzian is continuously differentiable, its ψ function is roughly linear for r small compared to σ, like the quadratic estimator, and redescends for $r \approx \sigma$, reducing the effect of outliers. Both the least-squares and Lorentzian estimators are shown in Fig. 4.2.

4.2 Optical Flow & Robustness (Framework & Literature Review)

Many existing techniques based on robust statistics are able to cope adequately with the hard problem of optical flow recovery when their assumptions hold. The challenge is to achieve high robustness against strong assumption violations commonly met in real sequences. In this thesis, we will mainly use and extend the regularization techniques presented at the previous chapter (sections 3.1.2, 3.1.3). Several authors propose an extension of these techniques using robust statistics [Black *et al.*, 1991; Black et al., 1996a; Black, 1994; Memin *et al.*, 1998; Ye *et al.*, 2001; Ye *et al.*, 2002]. We mainly focus on the extension of the work of [Black et al., 1996a] towards dense optical flow recovery in compressed video and use of additional constraints on image motion.

4.2.1 Robust Formulation of Regularization Techniques

In Chapter 3, we formulated the objective function including OFCE and its constraints and recovered the optical flow estimate in the least-squares sense. We also indicated the need for robustness due to the sensitivity of least-squares approaches at constraints' violations. The reformulation of the objective function to include the robust statistics

tools described above is almost straightforward. Black & Anandan, [Black et al., 1996a], use an M-estimator in the Horn & Shunck method [Horn *et al.*, 1981]. They simply take the standard least-squares formulation of optical flow and treat them in terms of robust estimation in an attempt to overcome the problems of oversmoothing and noise sensitivity. The standard least-squares estimate, Eq. (3.4), is simply reformulated as:

$$E_D(\mathbf{u}) = \sum_{(x,y)\in\Re} \rho(I_x u + I_y v + I_t, \sigma_D),$$

and the smoothness constraint, Eq. (3.5), becomes:

$$E_S(\mathbf{u}_s) = \lambda_S \sum_{n\in G_s} \left[\rho(u_s - u_n \sigma_S) + \rho(v_s - v_n, \sigma_S)\right], \text{ (rob. Form)}$$

where ρ is a robust estimator and σ_D and σ_S are the scale parameters for the robust estimators used. Similarly, the temporal continuity constraint becomes:

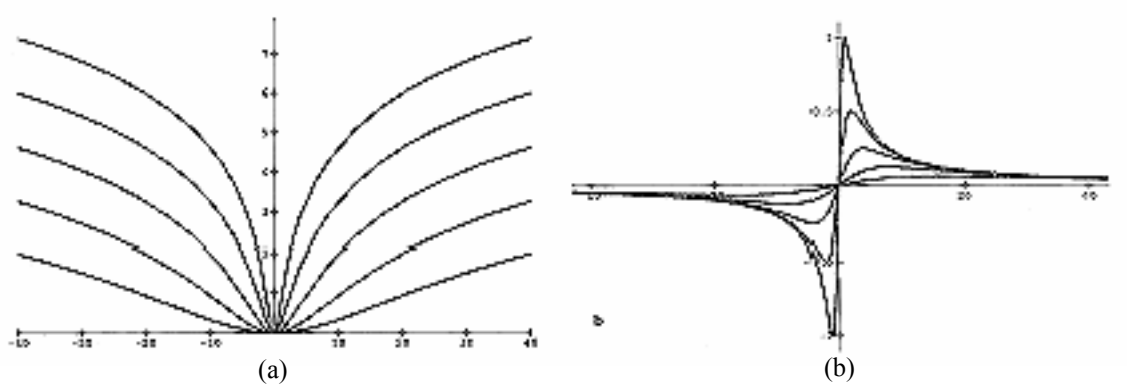
$$E_T(\mathbf{u}) = \lambda_T \rho(u - u^-, \sigma_T) + \rho(v - v^-, \sigma_T)$$

This robust formulation is adopted by many authors in the field of optical flow computation. Section 4.3 gives a brief overview of existing robust estimation regularization methods.

4.2.2 Minimization

Given the above robust formulation, many optimization techniques can be employed to recover the motion estimates. In general, the robust formulations do not admit closed form solutions, and often result in an objective function that is non-convex. Black & Anandan explored the use of stochastic minimization techniques such as simulated annealing [Black *et al.*, 1991] but found deterministic continuation methods to be more efficient and practical [Black *et al.*, 1993]. Under this consideration, they use SOR (Simultaneous Over-Relaxation), [Press *et al.*, 1988], to find the local minima and GNC (Graduated Non-Convexity) to find a globally optimal solution [Blake *et al.*, 1987]. The general idea is to take the non-convex objective function and construct a convex approximation. In the case of the Lorentzian estimator, this can be achieved by making the scale $(\sigma_D, \sigma_S, \sigma_T)$ parameters sufficiently large. This approximation is then minimized using a coarse-to-fine SOR technique. Successively better approximations of the true objective function are then constructed by altering the σ values, and minimized

starting from the solution of the previous approximation. Fig. 4.3 shows the Lorentzian estimator and its ψ-function for various values of σ. A coarse-to-fine strategy is employed to cope with large motions.



(a) (b)

Fig. 4.3 Lorentzian objective function plotted for thresholds $\sigma \in \{16,8,4,2,1\}/\sqrt{2}$. (a) Lorentzian Error measure; (b) Influence function, $\psi(x,\sigma)$ [Black *et al.*, 1996]

The coarse-to-fine approach is employed as described in section 3.1.5. A pyramid of spatially filtered and sub-sampled images is constructed using Gaussian kernels. Beginning at the lowest spatial resolution with the flow **u** starting from the initial

$\mathbf{u}, \mathbf{u}^- \leftarrow [0,0]$
$\sigma_{D,}\sigma_{S,}\sigma_T \leftarrow$ initial value at every site
$n \leftarrow$ fixed number of iterations
for each image
    // Perform c*oarse – to – fine* SOR
    $\mathbf{u} \leftarrow P\textit{yramid\_SOR}(\text{Image}_1, \text{Image}_2, \text{max\_level}, \text{min\_level}, \text{n}, \mathbf{u}, \mathbf{u}^-)$
    // Update control parameters
    $\sigma_i(x,y) \leftarrow f(\sigma_i(x,y)), \quad i \in \{D,S,T\}$
    // Retain previous solution
    $\mathbf{u}^- \leftarrow \mathbf{u}$
end

Fig. 4.4 GNC pseudocode

estimates, the change in the flow δ**u** is computed. The new flow field, **u**+δ**u**, is then projected to the next level in the pyramid (scaled as appropriate) and the first image at that level is warped towards the latter image using the flow information. The warped image is then used to compute the δ**u** at this level. The process is repeated until the flow has been computed at the full resolution. The GNC and SOR algorithms, which are used to recover the solution, are stated below.

The overall structure of GNC, coarse-to-fine SOR and SOR in terms of pseudocode is shown at Fig. 4.4, 4.5, 4.6, respectively. At any instant in time, the algorithm has a current estimate of the flow field **u**. When the projected image is

P*yramid _ SOR*$(\text{Image}_{t\text{-}1}, \text{Image}_t, \text{max\_level}, \text{min\_level}, \text{n}, \mathbf{u}, \mathbf{u}^-)$

if $\text{max\_level} \le \text{min\_level}$ *then*

$\delta\mathbf{u} \leftarrow [0,0]$

$\mathbf{u} \leftarrow SOR(\text{Image}_{t\text{-}1}, \text{Image}_t, \delta\mathbf{u}, \mathbf{u}^{\text{p}}, n)$

else

// recursively call Pyramid_SOR

$\mathbf{p\_u} \leftarrow reduce(\mathbf{u}) / 2$

$\mathbf{p\_u}^- \leftarrow reduce(\mathbf{u}^-) / 2$

$\text{p\_Image}_{t\text{-}1} \leftarrow reduce(\text{Image}_{t\text{-}1})$

$\text{p\_Image}_t \leftarrow reduce(\text{Image}_t)$

$\mathbf{u}^{p-1} \leftarrow Pyramid\_SOR(\text{p\_Image}_{t\text{-}1}, \text{p\_Image}_t, \text{max\_level}-1, \text{min\_level}, \text{n}, \mathbf{p\_u}, \mathbf{p\_u}^-)$

$\mathbf{u}^p \leftarrow project(\mathbf{u}^{p-1}, \text{max\_level}-1)$

$when \left|u - u^p\right| > 0.5 \quad OR \quad \left|v - v^p\right| > 0.5$

$\mathbf{u} \leftarrow \mathbf{u}^p$

// warp $\text{Image}_{t\text{-}1}$ by (u,v)

$\text{Image}_{t\text{-}1} \leftarrow \text{Image}_{t\text{-}1}(x-u, y-v)$

$\delta\mathbf{u} \leftarrow [0,0]$

$\mathbf{u} \leftarrow SOR(I_{t-1}, I_t, \delta\mathbf{u}, \mathbf{u}, \mathbf{u}^-, iterations)$

end

end

Fig. 4.5 Pyramid_SOR pseudocode

processed, the constraints are applied to yield a new objective function E and the estimate is refined, beginning with the prediction $\mathbf{u}^-$ as an initial estimate, by performing a fixed number of iterations of a continuation method, where an iteration corresponds to updating all flow vectors in the image.

$SOR(\text{Image}_{t-1}, \text{Image}_t, \delta\mathbf{u}, \mathbf{u}^p, iterations)$

// Compute derivatives from images

$I_x \leftarrow x\ Gradient$

$I_y \leftarrow y\ Gradient$

$I_t \leftarrow t\ Gradient\ (I_t - I_{t-1})$

// Compute bounds on second derivatives of E

$$T(u) \leftarrow \frac{\lambda_D I_x^2}{\sigma_D^2} + \frac{4\lambda_S}{\sigma_S^2} + \frac{\lambda_T}{\sigma_T^2}$$

$$T(v) \leftarrow \frac{\lambda_D I_y^2}{\sigma_D^2} + \frac{4\lambda_S}{\sigma_S^2} + \frac{\lambda_T}{\sigma_T^2}$$

for iterations

for all pixels do

$\mathbf{u} \leftarrow \mathbf{u}^p + \delta\mathbf{u}$

$$\delta u \leftarrow \delta u - \omega \frac{1}{T(u)} \frac{\partial E}{\partial u}$$

$$\delta v \leftarrow \delta v - \omega \frac{1}{T(v)} \frac{\partial E}{\partial v}$$

end for

end for

$\mathbf{u} \leftarrow \mathbf{u} + \delta\mathbf{u}$

end

Fig. 4.6 SOR pseudocode

As indicated before, the continuation method used is a coarse-to-fine SOR technique. The iterative update equations for minimizing E at step $n+1$ are simply [Blake *et al.*, 1987]. $u_s^{n+1} = u_s^n - \omega \frac{1}{T(u_s)} \frac{\partial E}{\partial u_s}$

$$v_s^{n+1} = v_s^n - \omega \frac{1}{T(v_s)} \frac{\partial E}{\partial v_s},$$

where

$$\frac{\partial E}{\partial u_s} = \sum_{s \in S} \left[\lambda_D I_x \psi(I_x u_s + I_y v_s + I_t, \sigma_D) + \lambda_S \sum_{n \in G_S} \psi(u_s - u_n, \sigma_S) \right],$$

$$\frac{\partial E}{\partial v_s} = \sum_{s \in S} \left[\lambda_D I_y \psi(I_x u_s + I_y v_s + I_t, \sigma_D) + \lambda_S \sum_{n \in G_S} \psi(v_s - v_n, \sigma_S) \right]$$

Therefore, the SOR method is summarized as:

where $\delta\mathbf{u}$ is the velocity vector refinement at each iteration and $\mathbf{u}^p$ the projected velocity vector from the previous level.

All parameters (σ_D, σ_S, λ_D, λ_S) are manually tuned and are constant for a number of experiments. Although, the authors claim that there is no need for exact parameter tuning due to the stability of the approach, there is still a need for automatic parameter evaluation to tackle the different video content scenarios.

4.3 Robust Estimation Literature Review

Odobez & Bouthemy [Odobez *et al.*, 1995] describe two robust multiresolution algorithms to solve the M- estimation problem, namely IRLS (Iterative Reweighted Least Squares) and PSM (Pseudo M- Estimator), and compare both of them with a multiresolution least-mean squares method.

Memin & Perez [Memin *et al.*, 1998] present a *multiresolution*/*multigrid* framework for optical flow estimation and object-based motion segmentation. The minimization of the energy function is processed through a multigrid algorithm, which consists in imposing weaker and weaker constraints on the searched estimates. This method leads to a multigrid iteratively reweighted least squares minimization of the objective function. The associated parameters are manually tuned and held constant for the whole procedure.

Sim & Park, [Sim *et al.*, 1998], propose an algorithm that is constructed by embedding the least median of squares (LMedS) of robust statistics into the Maximum A

Posteriori (MAP) estimator. They call it Reweighted Robust MAP (RRMAP). They describe it for a pair of images and extend it for multiple frame cases.

Bab-Hadiashar & Suter, [Bab-Hadiashar *et al.*, 1998], introduce and study two new robust optical flow recovery methods. For the first method, they modify the LMedS and use it to find an initial estimate. This initial estimate is then used to classify each pixel into two groups: "*inliers*" and "*outliers*". Finally, the inlier group is solved using the least squares technique. The resulting technique is called Weighted Least Squares (WLS). The second presented method is called Weighted Total Least Squares (WTLS). The weights for this method are computed using a new robust statistical method named the Least Median of Squares Orthogonal Distances (LMSOD).

Ye *et al.* [Ye *et al.*, 2002] propose a formulation based on three-frame matching and global optimization allowing local variation. Specifically, they begin with a robust local gradient method for initial flow and variance estimates, then refine the results using a global gradient descent method, and finally minimize the original energy by fastest descent. The optimization technique they adopt, namely *graduate optimization*, bears a certain similarity with GNC in that they start from an initial estimate and progressively minimize a series of finer approximations to the original energy.

Chapter 5

5. Robust Optical Flow Recovery From Compressed Video

In chapter 2, we discussed the advantages of processing compressed video without the need for full decompression. We also described the information that can be made available from an MPEG stream under the latter requirement, namely the motion vectors and the DC coefficients. In this chapter, we will elaborate on dense optical flow recovery using this information and will develop our approach step-by-step. We defined the term "dense" to characterize the generation of a motion vector for each single pixel in the image. In our approach, each pixel corresponds to the DC value of an MPEG block. Hence, under our framework, dense optical flow means the recovery of a single motion vector for each MPEG block or DC coefficient. New constraints on OFCE will be introduced and ideas regarding current and future work will be discussed.

5.1 Initial Formulation

Processing of uncompressed video (or moving image sequences) is the main topic for several research areas during the last two decades. Algorithms for generation of dense optical flow and intensity/motion segmentation have been proposed and integrated to complete computer vision systems, e.g. robotic vision & navigation. Many of them have proved promising and have undergone further research. The common problems they share are the complexity and memory/time consumption that render most of them inappropriate for several applications. The wide use of compressed video and the advantages it offers, naturally leads to the idea of transferring expertise from the uncompressed to the compressed domain. Hence, many researchers applied well-established uncompressed domain techniques to compressed video streams, mainly, for

scene or shot segmentation [Yeo *et al.*, 1995b; Ardizzone *et al.*, 1996; Ardizzone *et al.*, 1998; Xiong *et al.*, 1998; Mandal *et al.*, 1999; Bonzanini *et al.*, 2000].

To our knowledge, there is not serious work related to the estimation of dense optical flow field for compressed video. The motion field that is already available in an MPEG stream, generated by the block-matching technique (Chapter 2), is by no means representative of true motion and therefore not appropriate for possible use in accurate robotic vision, or motion segmentation and representation. Its main disadvantages are the inaccuracy in homogenous regions and the sparseness, since one MV for each MB is provided.

We start the presentation of our approach by realizing that most velocity estimation algorithms suffer from the initial value problem. Most optimization techniques fail, if the initial estimate of the solution is far from optimal. Several authors address this problem and encounter it by generating a crude initial motion field [Ye *et al.*, 2002] or by setting the motion field equal to zero and expecting the algorithm to converge after a number of iterations [Black et al., 1996a]. The MPEG standard provides us with a valuable tool that can be used to provide a "good" initial solution, namely the MPEG motion vectors. Although not accurate, especially at the borders and homogenous regions, these motion vectors are something "more" than a crude initial motion field. In our approach, we combine information from MPEG and robustly estimated motion vectors in order to recover a more accurate and dense optical flow field. In order to avoid full decompression of the MPEG stream we apply our robust estimation technique to the approximate DC images discussed in chapter 2. The intensity information they bear is enough for achieving efficient motion estimation.

5.2 Approach (step-by-step)

In the following sections, we describe our approach and explain the developed methodologies step-by-step. We discuss the algorithm for selecting the initial and final values of the scales used in incremental minimization of the objective function and the way they affect the final solution. The minimization framework is the same with that of [Black et al., 1996a], described analytically in chapter 4, and therefore no extended information is given. The main topic is the new motion constraints we introduce and the

way we treat the previous frame's motion field to recover the current frame's optical flow by balancing the individual constraint factors (λ).

5.2.1 Available Information & Construction of the Objective Function

An advantage of the compressed video over uncompressed is the available information included in the compressed stream. Such information can form the basis of many algorithms for motion representation, scene change/cut detection, intensity/motion segmentation etc. As explained in chapter 2, the directly (without full decompression) available information is the MPEG motion field and the DC intensity field. These two fields should ideally be spatially related, in the form that intensity changes due to object movement or scene change should be accurately captured by the MPEG motion field. Unfortunately, this relation breaks down at special locations in an uncontrolled manner.

In an attempt to generate a more representative velocity field of the true underlying motion, we look at two different ways to compute dense optical flow: The dense MPEG motion field (details are given in section 5.2.3) and the dense motion field resulting from the OFCE minimization. Their advantages and disadvantages have been already discussed in previous chapters. Our aim is to fuse the information they carry in a combinatorial and/or a selective manner. Motion constraints like the temporal continuity and the MPEG consistency will help us to effectively constrain the solution and generate a consistent optical flow.

As a consequence of the previous discussion, a logical question comes to mind: Can we relate the computation of the two dense motion fields in order to benefit from their inter-relations and incorporate the use of prior information by means of constraints? Under the framework of our approach, we view the formulation and solution of OFCE in relation to constraints provided by the available dense MPEG field. We formulate the objective function to be minimized as the combination of the *observed* and *a priori* information. The data term of the objective function, which involves the computation of derivatives, can be considered as the *observation*, while the smoothness (constraint on the spatial distribution of MVs), temporal and MPEG constraints (constraints on the motion field itself) can be regarded as the *a priori* part. The temporal continuity and MPEG consistency constraints can be further classified as algorithm dependent and algorithm

independent information respectively. The MPEG constraint involves motion vectors generated by the block-matching technique during the encoding, while the temporal constraint involves the last frame's optical flow that is generated during the algorithm's evolution. Hence, the objective function along a regularization framework is formulated as

$$E(\mathbf{u}) = \lambda_D E_D(\mathbf{u}) + \lambda_S E_S(\mathbf{u}) + \lambda_T E_T(\mathbf{u}, \mathbf{u}^-) + \lambda_M E_M(\mathbf{u}, \mathbf{u}_M),$$

initialized at the dense MPEG field, where λ_i with $i = \{D, S, T, M\}$ are the weight factors and *D,S,T,M* stand for *data*, *smoothness*, *temporal* and *MPEG* respectively . The individual constraints can be adaptively tuned, through the individual weights, within the extent of computation. Our efforts are mainly focused on tuning them through scalar parameters λ_i (combinatorial manner) or through on-off combination (selective manner), by keeping the λ_i ratios fixed throughout the data field. The latter combination leads to computationally faster results, while retaining an adequate accuracy.

In essence, our aim is to combine efficiently the available information and achieve the following targets:

1. Relative high accuracy of the recovered flow field (limited by the DC image resolution).
2. Improved detection of motion discontinuities.
3. Fast processing to retain the advantages of compressed domain processing.

5.2.2 Initial Velocity Estimation & MPEG Constraint

The original MPEG motion field cannot be used directly in conjunction with a pixel-based technique, like the robust estimation scheme we adopt, for two reasons:

- The MPEG motion vectors may refer to previous or past images that are multiple frames apart.
- The MPEG motion field associates one MV with a region of 16×16 pixels.

In chapter 2, we review the approach of Kobla *et al.* [Kobla *et al.*, 1997] that produces a reorganized set of motion vectors, which is independent of the frame type and the direction of prediction. We use this technique to compensate for the first problem. Therefore, we generate a new motion field for each frame that represents the direction of

motion of each MB with respect to the previous frame. Afterwards, we generate a dense velocity field by assigning the MV to all pixels inside the corresponding MB as illustrated in Fig. 5.1. We call this field the "uniform MPEG motion field".

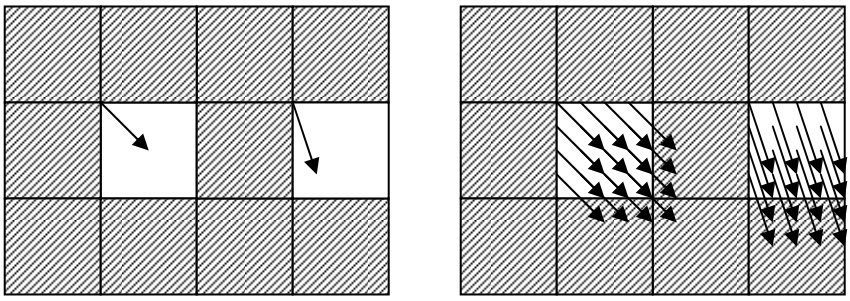


Fig 5.1 (a) Two MBs with the corresponding MV; (b) Same MV assigned to each pixel of the MB to generate a dense motion field

Summarizing the process at this point, we start with two different inputs, namely the intensity and motion information. The DC images provide the intensity information that will be used for the computation of the derivatives in the OFCE. These images are reduced 8 times in each direction with respect to the original uncompressed image and are derived by the technique presented in chapter 2. Each pixel of this small image represents a block of a MB and is associated with the corresponding MV from the generated dense MPEG field that provides us with initial motion information. The use of the dense MPEG velocities as initial solution for the objective function is considered reasonable, since we expect that the block-matching technique used in MPEG encoding does not deviate much from the true block motion field, at least as an overall distribution over the image. Hence, we have an initial estimate, which may exhibit errors, but it is overall sufficiently precise in representing the underlying motion.

The initialization to the MPEG field does not necessarily express our expectation that the final estimate will not deviate much from the initial MPEG solution. We actually want the final solution to deviate from the MPEG field wherever the MPEG vectors are erroneously estimated. On the other hand, we should encounter for inaccuracies in the OFC estimation process. Indeed, the OFCE can be influenced by approximation errors in the computation of derivatives that may falsely move the solution away from the MPEG field. Moreover, the smoothness constraint that compensates for large fluctuations in the data term induces excessive smoothing in the areas of motion boundaries. Therefore, in

order to further enhance the effects of the MPEG field in the final solution, we introduce an additional new constraint that is based on the difference between the current estimate and the initial solution. Mathematically, the latter constraint is robustly formulated as:

$$E_M(\mathbf{u}_s) = \lambda_M \rho(\mathbf{u}_s - \mathbf{u}_M, \sigma_M),$$

where $\mathbf{u}_M = [u_M, v_M]$ is the velocity vector of the dense MPEG MV, λ_M weights the relative importance of the constraint and σ_M is the scale of the robust estimator used. From now on, we will refer to this constraint as the "MPEG consistency constraint". Its robust form allows for large deviations from the MPEG field wherever appropriate, i.e. wherever there exist large inconsistencies between the MPEG field and the OFC formulation.

Frames from another well-known sequence are shown in Fig. 5.2. The "garden"

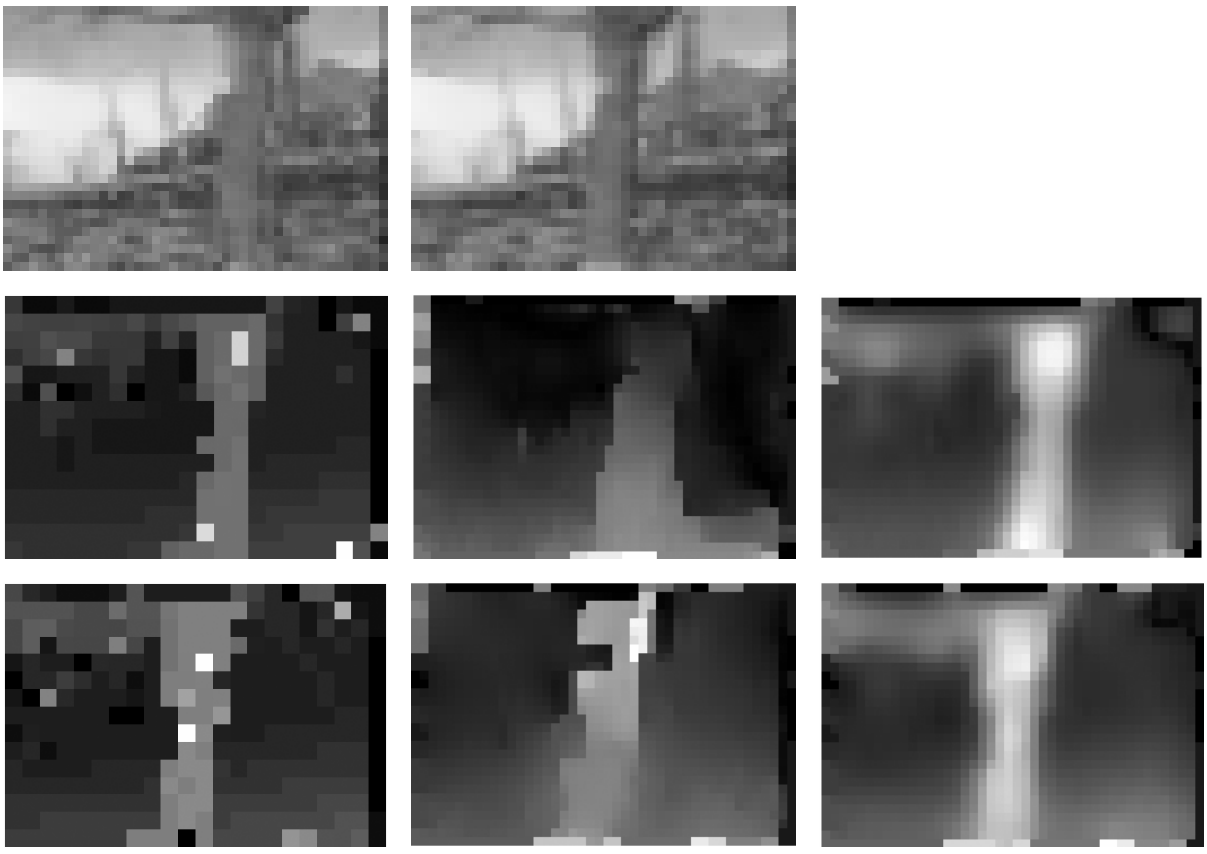


Fig. 5.2 Velocity magnitudes for frames 3 and 9 of the "garden" sequence. The first row shows the original DC images. The second & third rows show the initial MPEG field for frame 2 and 8, the results using OFCE and the results using OFCE+MPEG constraint. Both trials use the uniform MPEG MVs as initial solutions.

sequence was shot by a camera placed on a driving car, and the image motion is related to the distance from the camera. Thus, the tree that is closest to the camera moves faster than the background. The first row shows the obtained DC-images from frame 2 and 8 respectively. The second row presents the initial MPEG velocity field, the results obtained by using only the OFCE, in terms of the velocity magnitude ($\sqrt{u^2 + v^2}$) and the results using the MPEG constraint along with the OFCE. Both algorithms use the dense MPEG MVs, which are shown at the first column of Fig. 5.2, as initial solutions and the same values for σ_i and λ_i as in [Black et al., 1996].

The MPEG motion field provides crisp edges that indicate motion discontinuities, but the strong variations at homogenous areas, like the trunk of the tree or the branches at the upper left corner, are undesirable. As expected, the solution recovered using OFCE is inaccurate and oversmoothed at motion edges. Regions belonging to the tree cannot be easily distinguished because they are intermixed with the background. Additionally, artifacts of the intensity image strongly influence the solution and lead to false optical flow; the upper part of the tree has a small intensity "gap" due to lossy compression, which is retained in the recovered optical flow. In homogenous areas however, the OFCE solution is smoother than the MPEG one. The combination of two constraints, referring to optical flow and the MPEG motion field, along with the smoothness constraint, takes advantage of the competing nature of these two different motion fields in the overall criterion and improves the solution at the areas of their mismatch. Generally, the distinction between foreground (tree) and background (house, garden) is clearer on the combined solution, which also retains smooth structure within each individual region.

5.2.3 Initial Velocity Estimation & Temporal Constraint

We have not yet considered the temporal continuity constraint that we discussed in chapter 3. Obviously, we can add it to the objective function, as proposed by [Black, 1994], by assuming that we expect similar motion from previous to the next frame. The goal is to incrementally integrate motion information from new images with previous optical flow estimates to obtain more accurate information about the motion in the sequence over time.

Obviously, the temporal continuity constraint brings prior information into the computation of the current motion field, by means of the projected velocity field of the previous frame (equals zero for the first frame). We now have two different initial motion fields, namely the uniform MPEG that is algorithm independent and the previous frame's that is algorithm dependent. These two fields are by no means uncorrelated and it may be reasonable to claim that they are complementary in nature. We expect that the dense MPEG MVs are accurate at motion discontinuities and perhaps noisy at homogenous regions. On the contrary, we expect that the last recovered optical flow may be less accurate at motion borders, mainly due to the influence of smoothing, but more correct at uniform intensity regions. Hence, an efficient method should combine both of effects and advantages in order to recover, as correctly as possible, the true underlying motion.

Naturally, the first attempt to combine the temporal and MPEG constraints is to add both of them to the objective function, yielding therefore an objective function of the following form:

$$E(\mathbf{u}) = \lambda_D E_D(\mathbf{u}) + \lambda_S E_S(\mathbf{u}) + \lambda_T E_T(\mathbf{u}, \mathbf{u}^-) + \lambda_M E_M(\mathbf{u}, \mathbf{u}_M),$$

where

$$E_D(\mathbf{u}) = \lambda_D \sum_{(x,y)\in\Re} \rho(I_x u + I_y v + I_t, \sigma_D)$$

$$E_S(\mathbf{u}) = \lambda_S \sum_{n\in G_s} \left[\rho(u - u_n \sigma_S) + \rho(v - v_n, \sigma_S)\right]$$

$$E_T(\mathbf{u}, \mathbf{u}^-) = \lambda_T \rho(u - u^-, \sigma_T) + \rho(v - v^-, \sigma_T)$$

$$E_M(\mathbf{u}, \mathbf{u}_M) = \lambda_M \rho(u - u_M, \sigma_M) + \rho(v - v_M, \sigma_M)$$

Fig. 5.3 illustrates the results and shows the improvement gained by the incorporation of both constraints. The parameters are the same as before with σ_T varying within the σ_M range. The relative weights λ_T and λ_M are set to 0.5. The improved quality of the solution can be optically determined by the more compact optical flow (first row) and the more crispy motion edges (second row).

An in-depth look at the dense MPEG and temporal motion fields reveals an intrinsic redundancy at most frame regions. Both fields refer to the same motion and therefore we expect them to have similar content. This expectation leads us naturally to

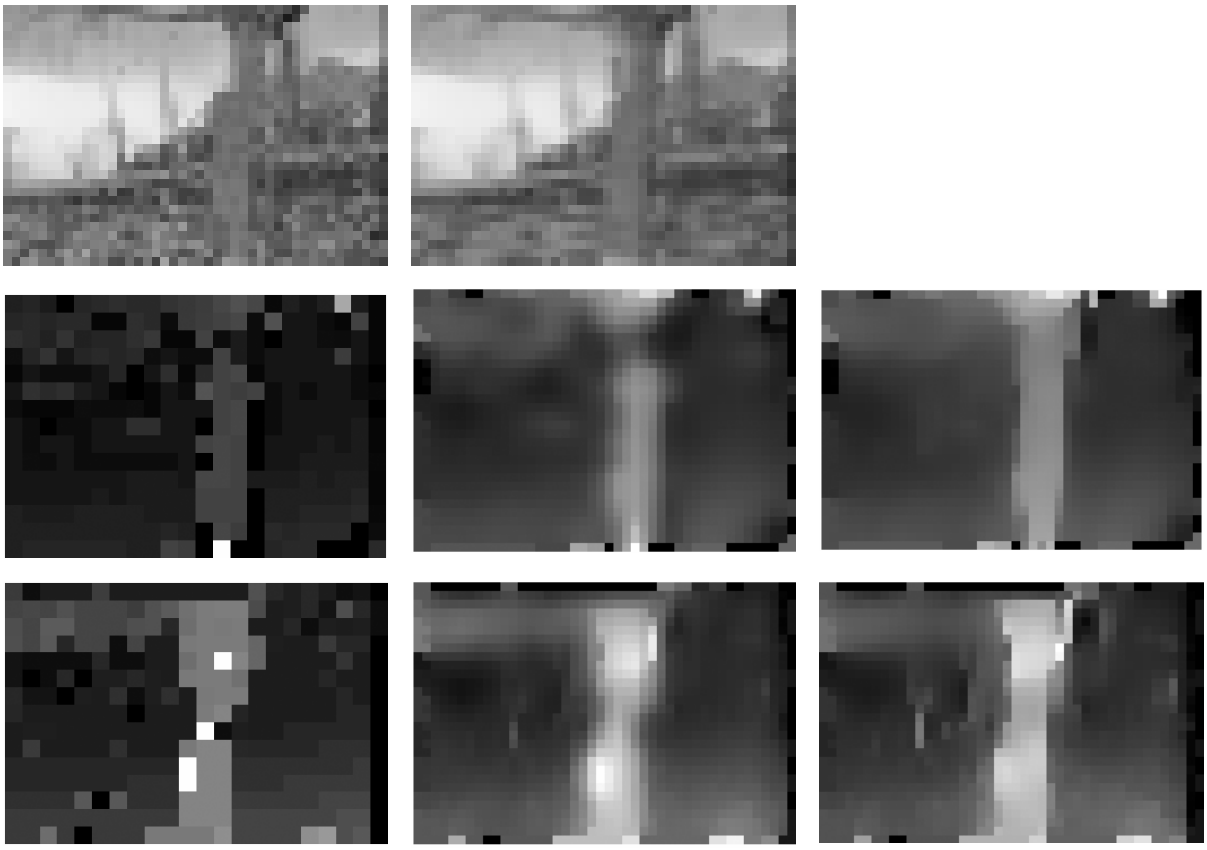


Fig. 5.3 Velocity magnitudes for frames 4 and 6 of the "garden" sequence. The first row shows the original DC images. The second & third rows show the initial MPEG field for frame 3 and 5, the results using OFCE +MPEG and the results using OFCE+MPEG+TEMPORAL constraints. Both trials use the uniform MPEG MVs as initial solutions.

find a way to use them in a distinct rather than a combinatorial form whenever they carry similar or consistent information. We reduce redundancy by keeping one of them whenever agreeing and combine them to take advantage of their complementary nature when we face mismatch. If we expect the dense MPEG field to be more accurate in a region, then we can use it alone; if the previous solution seems reliable then we can use it independently from the MPEG MVs. Although not yet fully explained, the last scheme introduces several advantages:

1. We can affirm the reliability of the uniform MPEG motion field with the additional information from the last recovered optical flow.

2. If we conclude that the uniform MPEG field is "trustworthy", we can avoid the minimization of the OFCE by retaining the initial solution for the current frame. A great amount of computational time can be saved.
3. Either we always have the choice to use the previous or the uniform MPEG MVs as initial velocity solutions with possible advantages this can offer.
4. Hard to face cases, like occlusion/disocclusion, illumination shading, appearance of new object etc may be handled more efficiently with appropriate MPEG-Temporal information fusion. For example, region spatial deviations of MPEG/temporal fields may help us to identify the previous cases.
5. Scene changes may be correctly located at frames where the temporal field relating the previous to the current frame, deviates as an overall distribution from that of the MPEG field, which relates the current to the next frame.

5.2.4 Scales estimation

As indicated in section 4.2.2, the minimization of the objective function for recovering optical flow begins with a convex approximation and the resulting estimate contains no outliers. In this sense, it is very much like the least-squares estimate. Outliers are gradually taken care of by lowering the value of each scale and repeating the minimization.

Black & Anandan [Black et al., 1996a] set the outlier threshold for the Lorentzian at $|x| \geq \pm\sqrt{2}\sigma$, where σ is the Lorentzian scale (Eq. 4.3). The value $\tau = \pm\sqrt{2}\sigma$ is actually the point at which the second derivative of the Lorentzian equals zero. If the maximum expected residual is τ, then choosing $\sigma = \tau/\sqrt{2}$ will result in a convex optimization problem. Black *et al.* define manually the initial and final values of the scales. They start with applying the coarse-to-fine strategy to this convex approximation. Then they lower the scales' values according to an annealing schedule and repeat the whole procedure. The scales' initial and final values are manually set.

We use an automatic method for selecting the initial and final scales for the robust data conservation and smoothness constraints. As indicated before, the initial solution to our algorithm is the available MPEG motion field, which in a Bayesian framework can be

viewed as *a priori* information. Assuming that most of these vectors are correct, i.e. fit

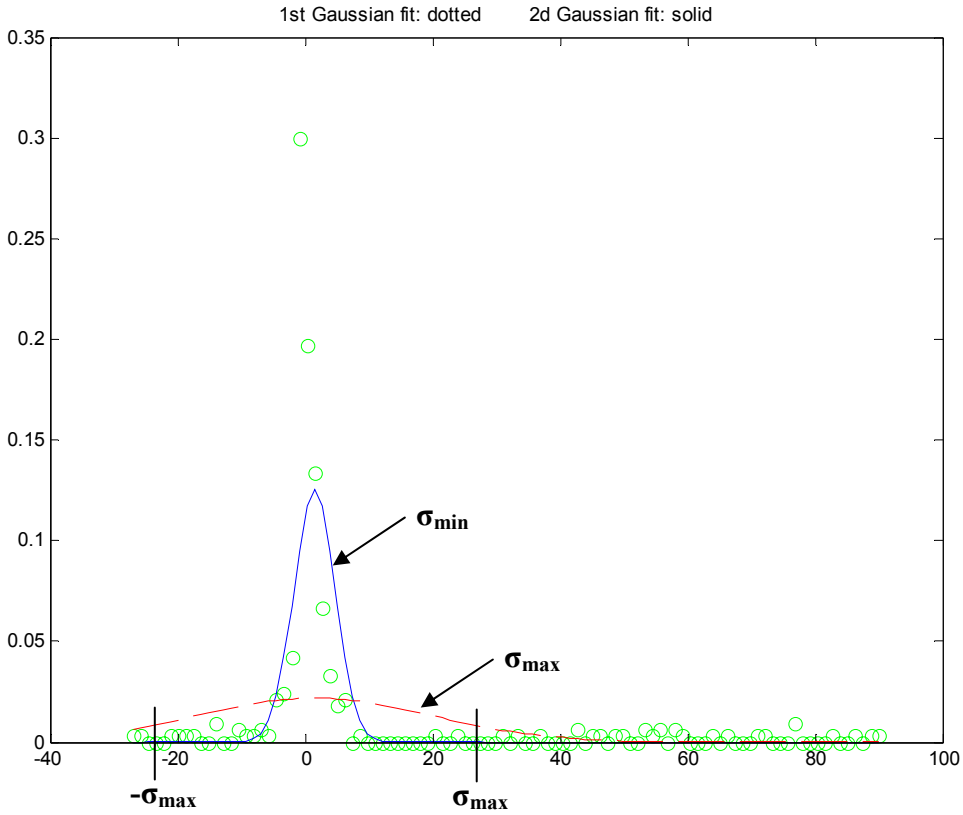


Fig 5.4 The first Gaussian fit is sketched with the dotted line, while the second one with the solid one. The circles represent the data constraint residuals for a frame of the "garden" sequence.

well the data conservation term, we are based on them to obtain scale estimates. We initialize the OFCE with the MPEG motion vectors' components (u, v) and obtain a value for each pixel in the frame. We repeat this procedure for the smoothness constraint equation. The resulting distributions are assumed ε-contaminated Gaussian with means μ_i and standard deviations σ_i. This Gaussian assumption holds well for small residuals, which are located around the mean, but fails for large residuals forming the long tails of the distribution, which are due to wrong estimates of the MPEG vectors or to large inconsistencies between the matching and the OFC criteria. Therefore, we calculate an initial global scale σ_1 by fitting a Gaussian distribution, having in mind that this scale estimate can be "crude" and only used to generate the convex approximation of the

objective function. Afterwards, we make a second fit on the residuals inside the interval $[-\sigma_1, +\sigma_1]$ and obtain a more accurate Gaussian fit with standard deviation σ_2. The second step rejects outliers that may influence negatively the fit. The probability that the residuals will fall within this confidence interval is 0.9544997 [Papoulis, 1984]. To enhance the robustness of the fit we use the median of the distribution as the estimated mean. We then relate the σ_1 and σ_2 with the *max* and *min* scales of the Lorentzian functions used to model the observation and smoothness terms in section 4.1, in order to obtain an upper (convex approximation) and lower (min outlier) threshold for the estimated scales. The attempted correspondence between Gaussian and Lorentzian scales can be justified by realizing that at small deviations the Lorentzian distribution approximates well the Gaussian. Fig. 5.4 illustrates the automatic selection of scales.

Consequently, the initial/final scales estimation procedure and the annealing schedule is summarized as follows:

1. Initialize the data & smoothness constraint equations with the MPEG motion vectors.
2. Fit Gaussians to the overall distributions and calculate the standard deviation $\sigma_{D\max}$ and $\sigma_{S\max}$.
3. Fit Gaussians to the residuals within $[-\sigma_{D\max}/-\sigma_{S\max}, \sigma_{D\max}/\sigma_{S\max}]$ and calculate the standard deviations $\sigma_{D\min}$ and $\sigma_{S\min}$.
4. Set $\sigma_{D\max}/\sigma_{S\max}$ value as initial and $\sigma_{D\min}/\sigma_{S\min}$ as the final values for the minimization algorithm. Initially, the estimate considers all motion vectors as inliers. Outliers are gradually introduced by lowering $\sigma_{D\max}/\sigma_{S\max}$ until the minimum allowed values $\sigma_{D\min}/\sigma_{S\min}$ are reached.

The MPEG consistency and temporal continuity constraints impose a requirement to the derived motion field for being smoothly varying around the MPEG and temporal fields, respectively. The robust form of these constraints allows for large deviations wherever large inconsistencies with the overall criterion are detected. Thus, the utilizing and consequently the structure of these constraints should resemble those of the smoothness constraint. Based on this reasoning, we use the same scales for the smoothness, MPEG and temporal constraints.

To illustrate the behavior of the automatic scale estimation we consider a sequence containing a car moving on a highway viewed from a static camera [WSSAE, "road surveillance 1" sequence] (Fig. 5.5). The first row presents the uncompressed fifth frame and the corresponding DC image, which is appropriately scaled for illustrative purposes.

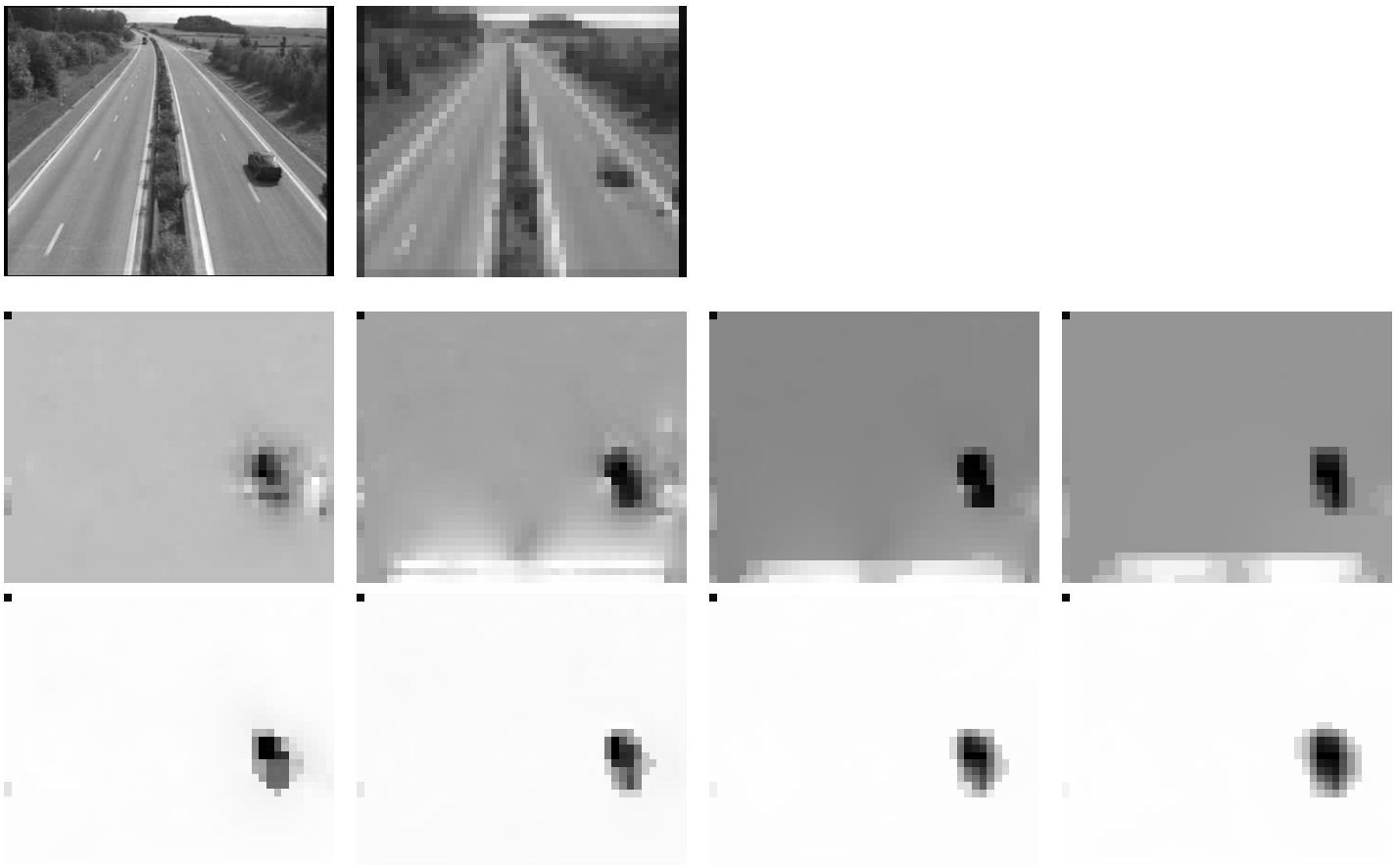


Fig. 5.5 ***1st row*: original frame from "road1" sequence and DC image scaled as appropriate for illustrative purposes; *2d row*: stages 1-4 using manually set values for σ (see text); *3 d row*: stages 1-4 using automatically calculated values for σ.**

The second row shows the estimated vertical velocities (each value specifies a gray intensity level) resulting from the annealing schedule using manually set scales for the OFCE and the smoothness constraint. Each image is the result of a stage of constant scale, where the scale is reduced at subsequent stage. We use the parameter values indicated by [Black et al., 1996a] in a one-level pyramid and a four-stage scheme ($\sigma_D \leftarrow {}^{18}\!/_{\sqrt{2}} \ldots {}^{5}\!/_{\sqrt{2}}$, $\sigma_S \leftarrow {}^{3}\!/_{\sqrt{2}} \ldots {}^{0.03}\!/_{\sqrt{2}}$, $\lambda_D = 5$, $\lambda_S = 1$). The third row shows the results using our method for estimating the scales. The first two stages prove that we are

already close to a "good" solution and therefore we can avoid a scheme with many stages that is time consuming. Some artifacts, like the bright regions at the bottom of the image of the second row, are also eliminated.

A more complex sequence is considered in Fig. 5.6. The well-known "coast guard" sequence contains two boats moving in opposite directions viewed by a moving camera that follows the small boat at first and the larger boat afterwards. Fig. 5.6 shows the second frame, where only the small boat is visible. The manual scales are set the same as before. The horizontal velocity estimates of our approach become obviously more exact around the object of interest (small boat).

5.2.5 Constraint-Weight Selection

The derived objective function consists of the data, smoothness, temporal and MPEG

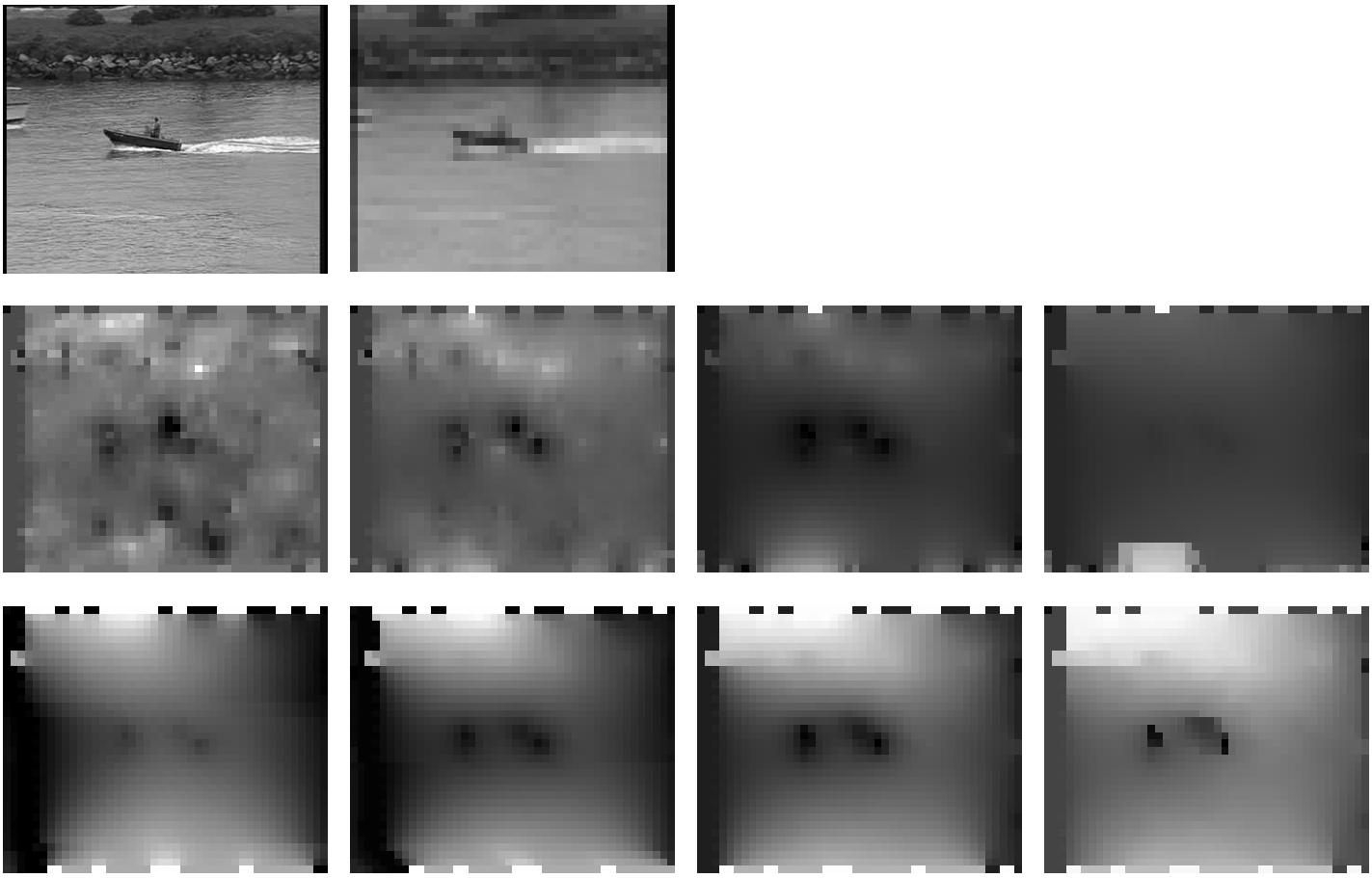


Fig. 5.6 ***1st row*: original frame from "road1" sequence and DC image scaled as appropriate for illustrative purposes; *2d row*: stages 1-4 using manually set values for σ (see text); *3 d row*: stages 1-4 using automatically calculated values for σ.**

constraints with each of them having its own relative weight λ_i. We undertook several experiments with different values for λ_i and concluded that the data-smoothness and MPEG-temporal ratios should be held constant as in the work of Black *et al.* [Black *et al.*, 1996].

The balancing problem turns to be the design of an efficient way, in terms of reliable results and time consumption, to turn on or off the individual groups of constraints. We use the term "group" to distinct between the data-smoothness and MPEG-temporal pairs. We incorporate this distinction, since the data should always be used in addition with a non-zero weighted smoothness term in order to make the solution recovery well-posed. The MPEG-temporal terms are grouped together because of the complementary nature discussed above.

In section 5.2.4 we discussed the obvious way to combine the constraints; just add them to the objective function and start the minimization procedure. This is done by incorporating the $\frac{\lambda_D}{\lambda_S} = \frac{5}{1}$ ratio for the data-smoothness pair, as used in Black & Anandan [Black *et al.*, 1996]. The weights λ_{T} and λ_{M} for the temporal and MPEG constraints, respectively are set to 0.5. In order to employ their complementary nature and the advantages it offers, we design a different method to balance the objective function's terms.

Let us state once more the notion of *outlier* and incorporate it in our approach. Huber, [Huber, 1981], refers to a robust estimator as an estimator *"insensitive to small departures from the idealized assumption for which the estimator is optimized"*. Fractionally large departures for a small number of data points lead to the notion of *outlier* points. Outliers are detected where the final values of the data coherence and spatial smoothness terms are greater than the outlier thresholds τ_D and τ_S ($\sqrt{2}\sigma_{\mathrm{D}}$ and $\sqrt{2}\sigma_{\mathrm{S}}$ for the Lorentzian). The data and smoothness outliers are treated in a different way, since they provide different information regarding the present motion between two frames. Motion discontinuities are simply outliers with respect to spatial smoothness [Black *et al.*, 1996]. We use the data outliers as a criterion to test the initial velocity correctness. Hence, we initialize the data constraint with the dense MPEG motion vectors

that form the initial solution to our estimation approach. If the result is an inlier, then we may expect that the MPEG MV repesents relative well the underlying motion, which means that it almost satisfies the OFCE ($I_x u + I_y v + I_t = 0$). In the contrary, if the result is an outlier, then we can claim that the initial solution is not a good starting point for the minimization procedure that follows. Keeping in mind the previous thoughts, we provide the proposed algorithm in pseudocode form in Fig. 5.7 and explain few aspects below. The terms "spatialMPEG" and "spatialTEMPORAL" found in the pseudocode represent the result of the smoothness constraint when initialized with the dense MPEG MVs and the estimated velocities of the previous frame respectively. The term "dataMPEG" represents the result of the data constraint when initialized with the dense MPEG MVs. Reading carefully the proposed scheme, one can reach the following conclusions:

1. We use both the spatialMPEG and spatialTEMPORAL outlier check to ensure the presence of motion discontinuities (lines 1, 14)
2. We always check for MPEG MV correctness using the dataMPEG outlier check (lines 3, 16)
3. Generally, we trust the MPEG MV. Hence, if spatialMPEG is an outlier and spatialTEMPORAL an inlier, we adopt the MPEG velocity as the final solution (lines 27-35)
4. We do not use the previous solution as a constraint in the objective function (temporal constraint)!

The first three conclusions arise easily from the previous discussion. The last one seems strange, because it leaves out a constraint that was described as a powerful one. To explain our choice let us consider two consecutive frames, *i-1* and *i*, of a video sequence. The temporal field projected to *i* from *i-1* frame results from a previous combination of the two motion estimation methods, namely the dense MPEG (block-matching) and robust optical flow (pixel-based). Hence, the current temporal field comes with the influence of previous spatial relations and interactions between the two motion fields that may change at this particular instant (frame *i*). To decouple the current computation from previous assumptions regarding the interdependence of motion fields, we use the temporal field, wherever appropriate, rather that as a strong constraint on the current motion field.

```
1.  If ( spatialMPEG > τ_S) AND ( spatialTEMPORAL > τ_S) )      % if both outliers
2.  {
3.          if (dataMPEG< τ_D )      % if MPEG is data inlier
4.          {
5.                  MPEG MV is "correct". Use it as the solution of the current frame (avoid
6.                  minimization of the objective function)
7.          }
8.          else                     % if MPEG is data outlier
9.          {
10.                 minimize the OFCE starting from the previous solution}
11.         }
12. else
13. {
14.         If ( spatialMPEG < τ_S) AND ( spatialTEMPORAL < τ_S) )      % if both inliers
15.         {
16.                 if (dataMPEG < τ_D )     % if MPEG is data inlier
17.                 {
18.                         MPEG MV is "correct". Use it as the solution of the current frame
19.                         (avoid minimization of the objective function)
20.                 }
21.         else                     % if MPEG is data outlier
22.         {
23.                 minimize the OFCE starting from the previous solution
24.         }
25.         else
26.         {
27.                 if ( spatialMPEG < τ_S)          % if MPEG is spatial  inlier
28.                 {
29.                         minimize the OFCE starting from the previous solution
30.                 }
31.                 else                             % if MPEG is spatial outlier
32.                 {
33.                         MPEG MV is "correct". Use it as the solution of the current frame
34.                         (avoid minimization of the objective function)
35.                 }
36.         }
37. }
```

Fig. 5.7 Pseudocode for automatic balancing of objective function's constraints (see text)

Chapter 6

6. Experimental Results

This chapter shows the results of our optical flow recovery approach on real image sequences. Video sequences that are commonly used in the literature are considered and the results are analyzed and explained thoroughly. We present the recovered optical flow for videos with different motion scenarios imposing different challenges.

The results were carefully studied in order to present advantages and disadvantages of the proposed method. Hence, we present cases where our algorithm succeeds or fails. The latter cases are explained and ways to overcome potential problems are discussed. Due to lack of ground truth optical flow, the analysis is based on qualitative rather than quantitative criteria.

6.1 Experimental Framework

We process several video sequences to test the validity of our approach. The most representative of them are shown and analyzed below. Most of them are downloaded in an uncompressed form from the "working site for sequences and algorithms exchange" [WSSAE] that serves as a repository for the call of comparison initiated by the Cost-211 group. All of them were MPEG compressed using the TMPGEnc encoder that is publicly available. For uniformity reasons, they share common characteristics, namely IBBPBB GOP structure, normal precision motion search (TMPGEnc option), CIF format (352×288 pixels) and 30 frames/s rate.

As indicated in previous chapters, our approach is an extension of Black & Anandan's work [Black *et al.*, 1996]. We used their software that is publicly available in C, [Black, software], as a starting point for software development of our algorithm. For the presentation of results, we implemented several programs in MATLAB because of the powerful visualization tools it offers.

The first illustrated sequence is the *"flower garden"* (Fig. 6.1). In this sequence the tree, flowerbed and row of houses move towards the left due to camera pan, but at different velocities. Regions of the scene closer to the camera move faster than the regions near the row of houses in the background. This sequence contains large depth

Fig. 6.1 Consecutive frames from the "Flower Garden" sequence illustrating the large depth disparities and the occlusions (areas behind the tree)

disparities and hence, significant perspective distortions. It also contains prominent motion field spatial discontinuities and occlusions.

The *"table tennis"* sequence, shown in Fig. 6.2, presents a whole range of situations that makes it a challenging stream. Many of the motions of regions of interest are discontinuous and rapidly changing (the motion of the ball exceeds 20 pixels between frames). The limited intensity variation from frame to frame and the zooming process after the 23^{d} frame (approximately) pose additional difficulties.

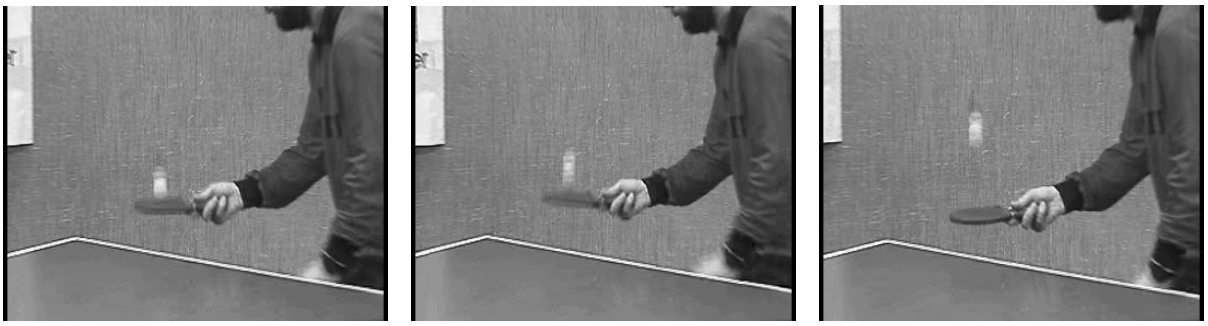


Fig. 6.2 Consecutive frames from the "Table Tennis" sequence illustrating the rapid movement of the ball that yields a discontinuous motion field and occluded areas below and above.

The *"coast guard"* sequence, Fig. 6.3, shows a complex scene with different objects present. Two boats are moving in opposite directions in a river, while the camera pans, following the smaller boat initially and then the larger one. The coast, covered by

trees and rocks, and the wavy river are present throughout the sequence. The objects' motions are small in magnitude and make the distinction between them rather difficult.

Fig. 6.3 Consecutive frames from the "Coast Guard" sequence illustrating the quite small motion present in the scene.

6.2 Results & Discussion

In chapter 5, we attempted to make an initial presentation of the results by applying our algorithm to the "flower garden" sequence. Improvements of the optical flow using the MPEG and temporal constraints in both a linear and selective way were presented. Hence, in this chapter we make a brief illustration of optical flow results and continue with problems rising from the other sequences.

6.2.1 *"Flower Garden"* Sequence

The first row of Fig. 6.4 shows the uncompressed third frame of the video sequence and the corresponding DC image. The second row shows the recovered optical flow field by the OFCE and the third illustrates the dense MPEG field. The additional information to the figures of the previous chapter is the zoomed regions beside them. The thin black box encloses a region, where occlusion takes place: The tree is moving to the left in the foreground covering or uncovering regions of the background, namely the house, the trees and the flower bed. The occlusion in the specified region occurs due to the covered tree in the background (its branches can be seen in the uncompressed frame shown in Fig. 6.4). The OFCE minimization produces false MVs as shown by the different directions of neighboring vectors. On the contrary, the MPEG motion field is correct at the occluding region. Fig. 6.5 illustrates the solution recovered by the additive and selective

combination of the temporal and MPEG constraints. The direction of the velocity vectors is the expected at both sides of the motion discontinuity. In general, the selective combination provides better results in terms of velocity magnitude and direction as illustrated by the optical flow field overlaid on the DC image.

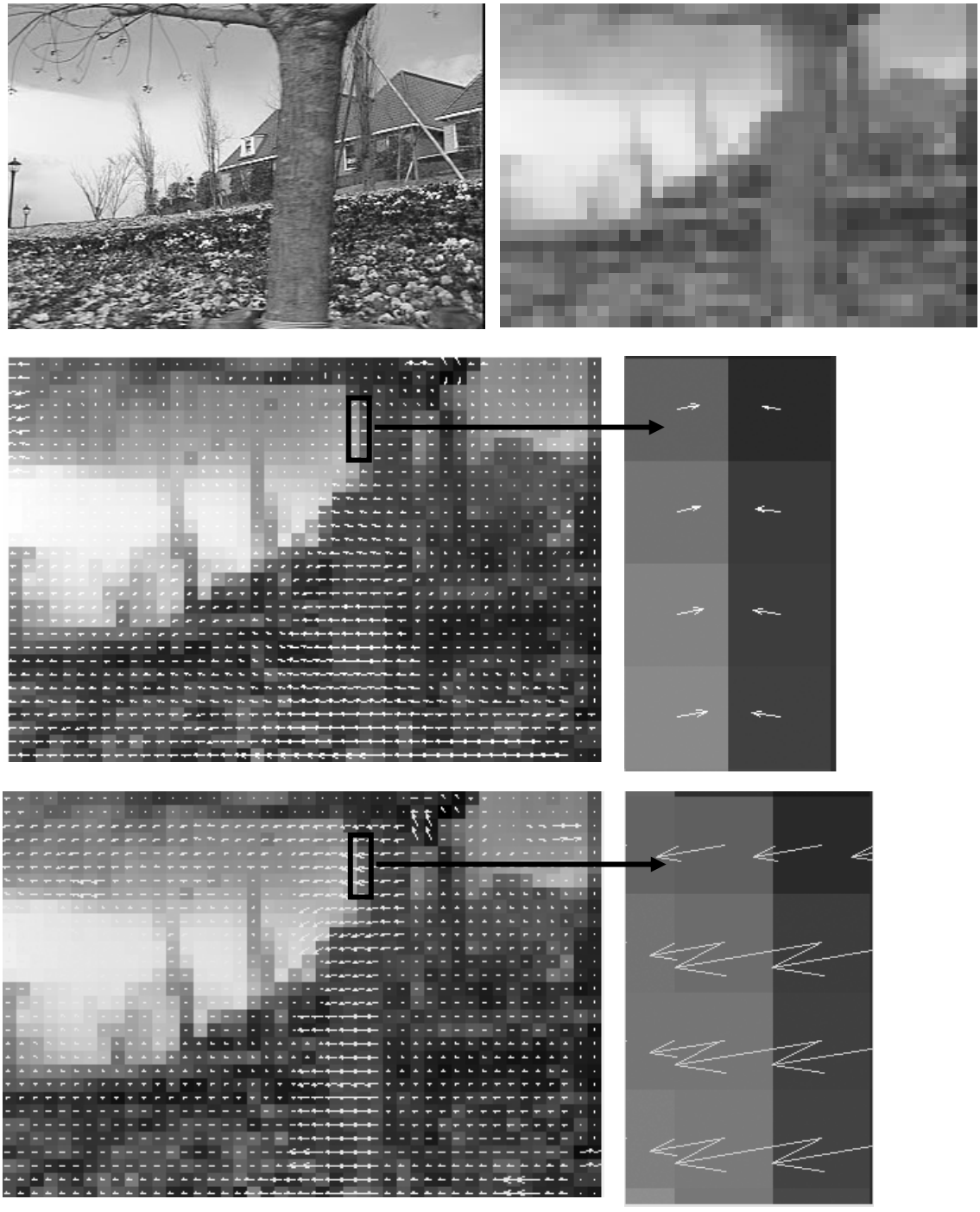


Fig. 6.4 First row: original frame 3 and corresponding DC image; second row: optical flow by OFCE and zoom on occlusion; third row: optical flow by dense MPEG and zoom on occlusion (see text)

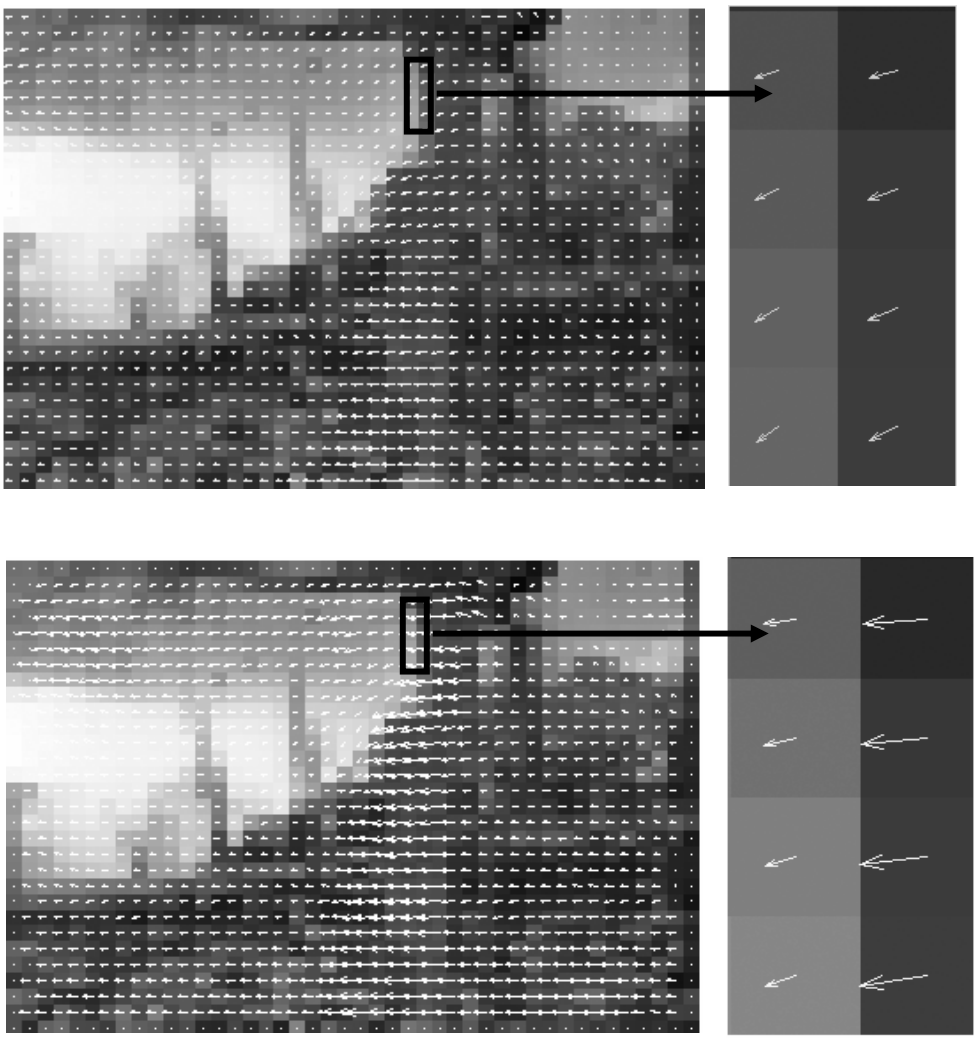


Fig. 6.5 First row: optical flow by TEMPORAL+MPEG and zoom on occlusion; second row: optical flow by LAMDA and zoom on occlusion (see text)

Video streams contain hundreds of frames and it is therefore difficult to present overall results using images. Hence, in an attempt to provide a qualitative assessment of the results, we could say that the algorithm works well for most frames of the sequence under investigation. Two main types of motion are present: the foreground moving tree (local motion) and the camera pan (global motion). In most cases, these two motions are easily distinguishable. The MPEG motion field is more accurate at the motion edges, as illustrated by the above example, while the OFCE field recovers velocity information on smooth areas, like the flowerbed. The selective combination of the constraints, which was analyzed in the previous chapter, seems to work better in efficiently combining both advantages of the MPEG and OFCE fields.

6.2.2 *"Table Tennis"* Sequence

Considering the first frames of this sequence, we investigate the motion of objects of interest when the camera has a fixed focus depth, focusing on the player's arms, and when zooming out. Fig. 6.6 shows four different optical flow fields superimposed on the 11^{th} DC image of the sequence. As illustrated by the two consequent DC images, the ball moves very fast and therefore undertakes a large motion. The arm is moving upwards performing a relative small motion.

The optical flow recovered by the OFCE seems a little bit "messy", especially around the ball. The motion blurring due to the large movement of the ball in conjunction with the homogenous background gives spurious derivatives and produces many false motion vectors. The distinction between the ball and the bat is visually impossible. As expected, the dense MPEG field is more accurate at the motion edges, but provides inconsistent information for the moving homogenous regions of the arm. The complementary combination of the OFC, MPEG and TEMPORAL constraints produces a better field than the OFCE and dense MPEG individually in terms of vector direction and homogenous regions' velocity assignment. The selective combination of constraints generates the best field by achieving an improvement on the velocities' direction and magnitude. The small motion of the bat is clear and can be distinguished from the ball's motion. The latter remark is better illustrated by zooming on the region containing the ball and the bat, as illustrated in Fig. 6.8.

An interesting part of the sequence is the zooming out effect performed approximately after the 23^{d} frame. The camera zooms out, but remains focused on a region between the ball and the bat. The challenge is to distinguish the movements of the ball and the arm without being confused by the global change of the motion field. Frames 36 and 37 of the sequence are shown in Fig. 6.7. The camera zooms out, the ball is moving downwards and the left part of the bat makes a slightly larger movement than the arm. Fig. 6.6 and Fig. 6.8 illustrate the optical flow obtained by OFCE, dense MPEG, the linear the selective combination respectively in a row-wise manner. Considering the global velocity distributions shown in Fig. 6.7, we confirm previous stated remarks. The OFCE recovers almost correctly the zoom pattern in homogenous regions, like the wall behind the player, but fails to distinguish the objects' movements. The dense MPEG field

is more accurate at the motion edges and recovers partially the zoom pattern in homogenous regions. The movement of the objects of interest is well recovered. The table edge that can be considered as a moving edge, because of the zoom out, produces artifacts in the resulting optical flow as can be seen by the messy motion vectors in terms of magnitude and direction around this region (severe effect of aperture problem on edges). The linear combination of the constraints generates the worse optical flow. The zoom pattern is hardly observed, while no distinction between moving objects exists. The careful examination of the results after the 23^{d} frame, where zooming out begins, reveals the inability of the linear combination to give a good solution. The temporal constraint is possibly responsible for this. The constraint it imposes, namely the similarity between the previous and the current solution, cannot handle the abrupt change between static and change of focus. This irregularity produces undesirable results on the final optical flow. The selective constraint combination generates optical flow that is smoother than the MPEG and more accurate in representing motion than the OFC field. The artifacts around the table edge, present in the MPEG field, are almost smoothed out and many “gaps” in homogenous regions are filled in with motion vectors that are in accordance with the global motion pattern. The zoomed region of interest in Fig. 6.9 illustrates the previous facts. The point of focus, the center of the zoom pattern, is relative clear at the last case. The upward movement of the bat is not clearly represented, but can still be distinguished.

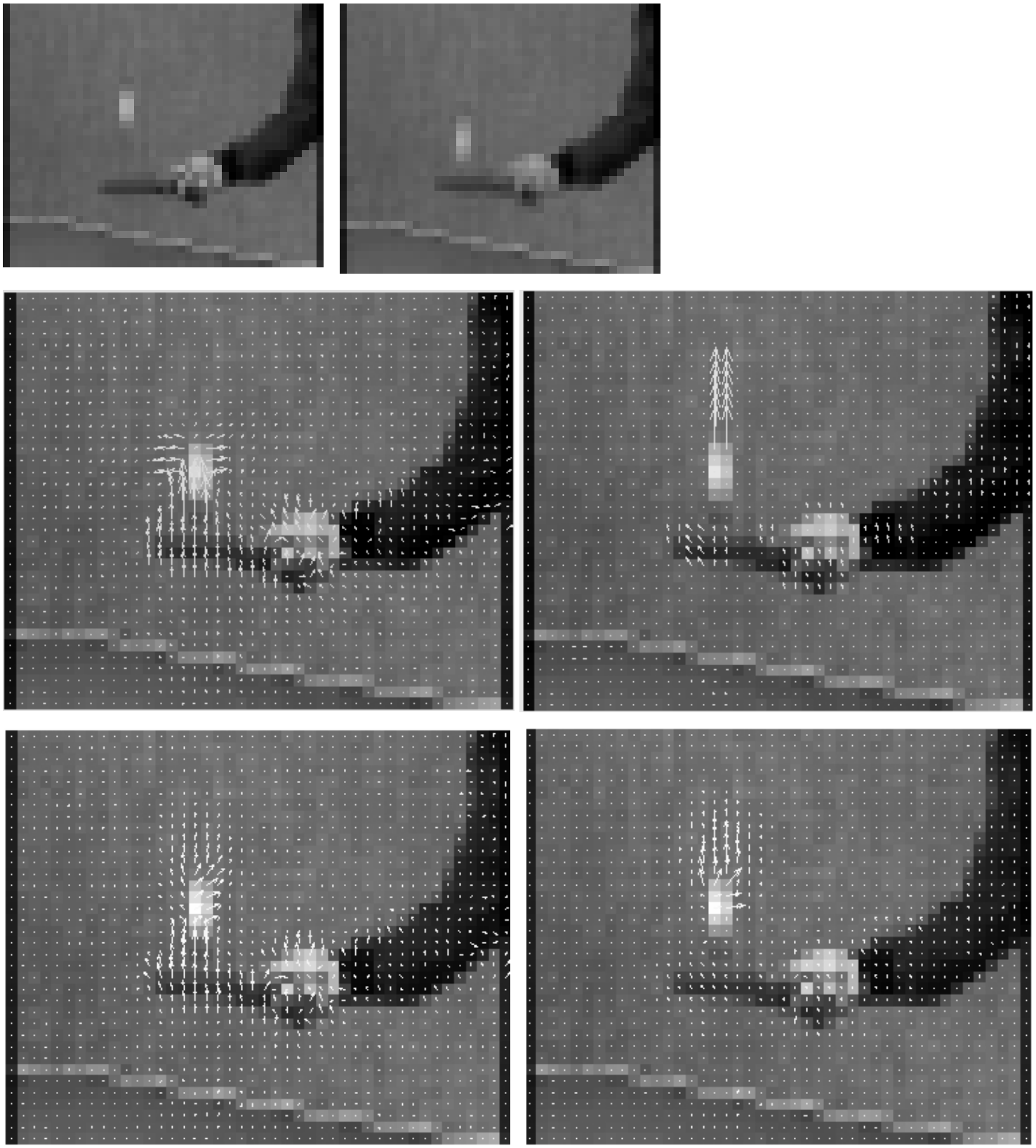


Fig. 6.6 10th & 11th DC images (scaled) of the "Table Tennis" sequence. Optical flow field for the 11th frame (scaled). Row-wise: OFCE, dense MPEG, linear combination and selective combination

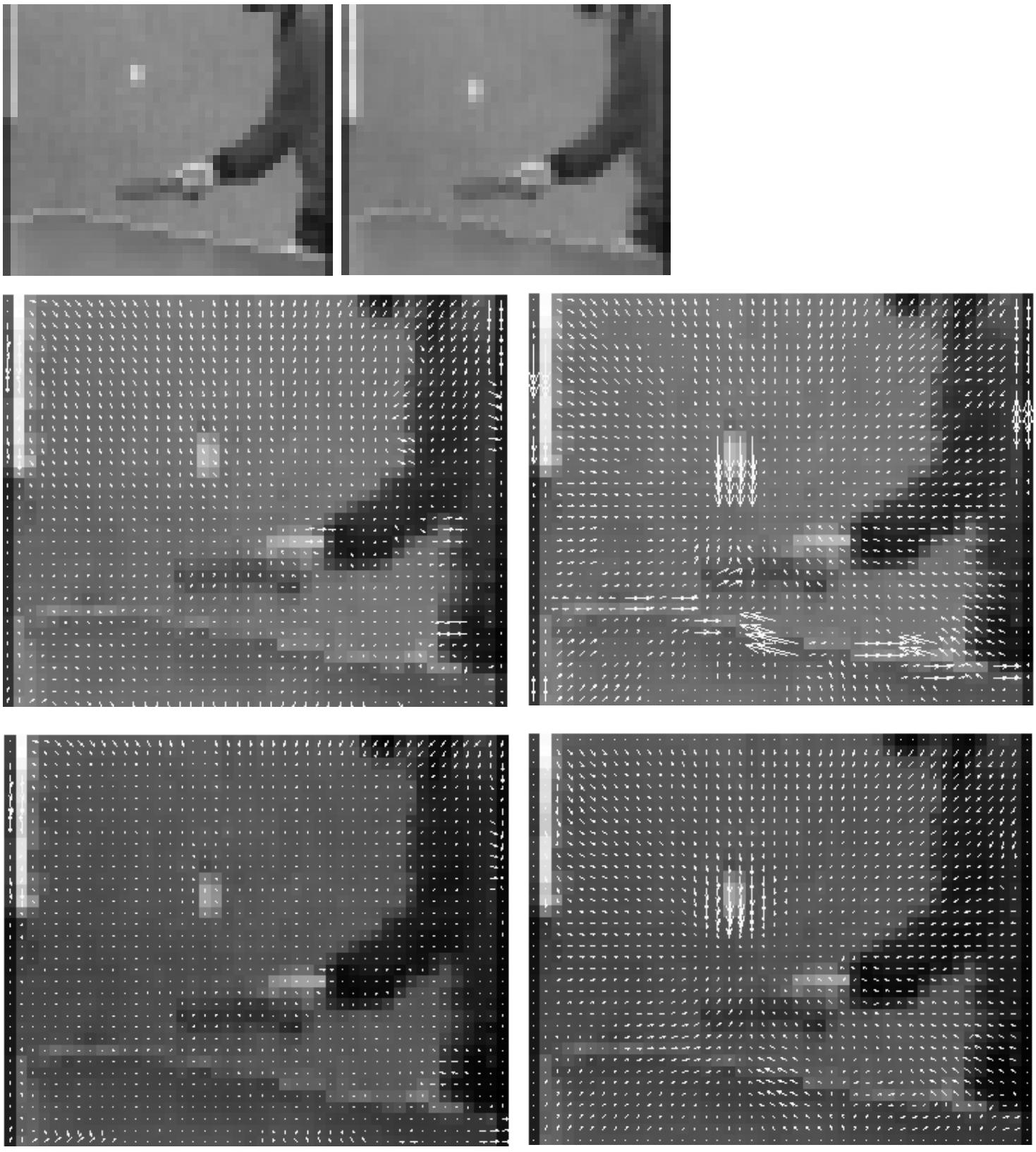


Fig. 6.7 36th & 37th DC images (scaled) of the "Table Tennis" sequence. Optical flow field for the 37th frame (scaled). Row-wise: OFCE, dense MPEG, linear combination and selective combination

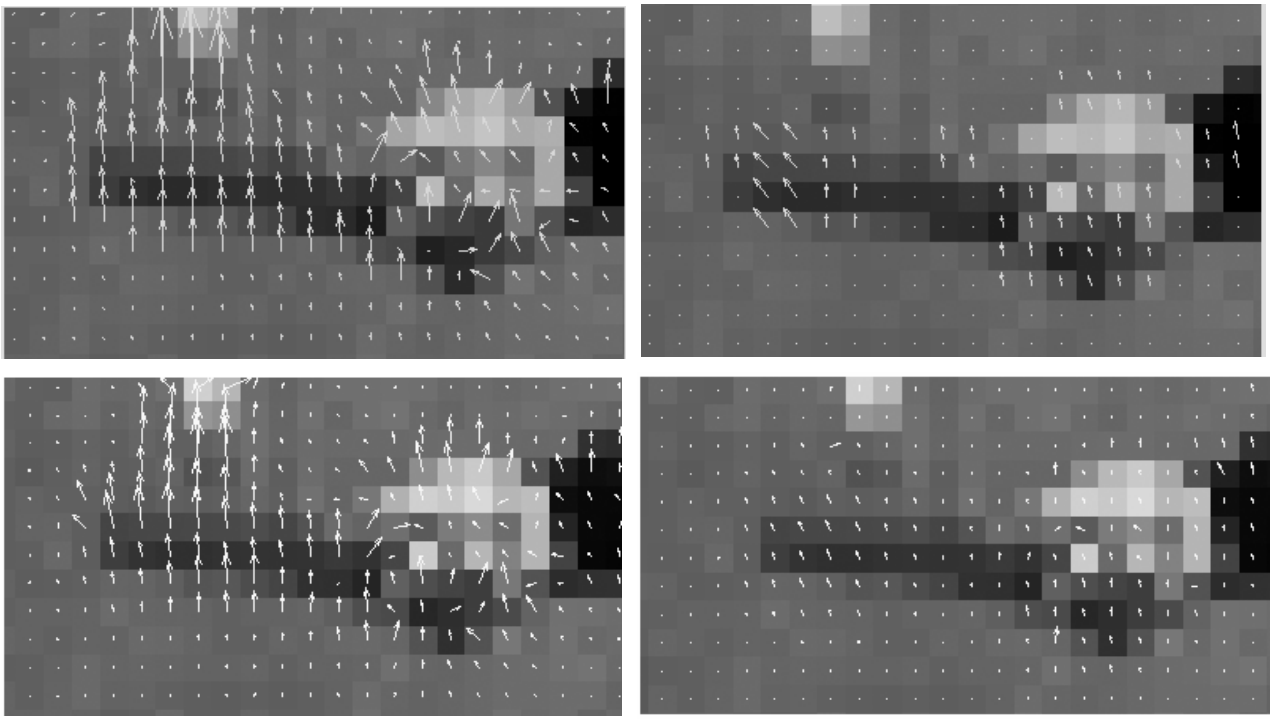


Fig. 6.8 Frame 11. row-wise: OFCE, dense MPEG, linear combination and selective combination. The distinction between the ball and the bat is better at the latter case (see text)

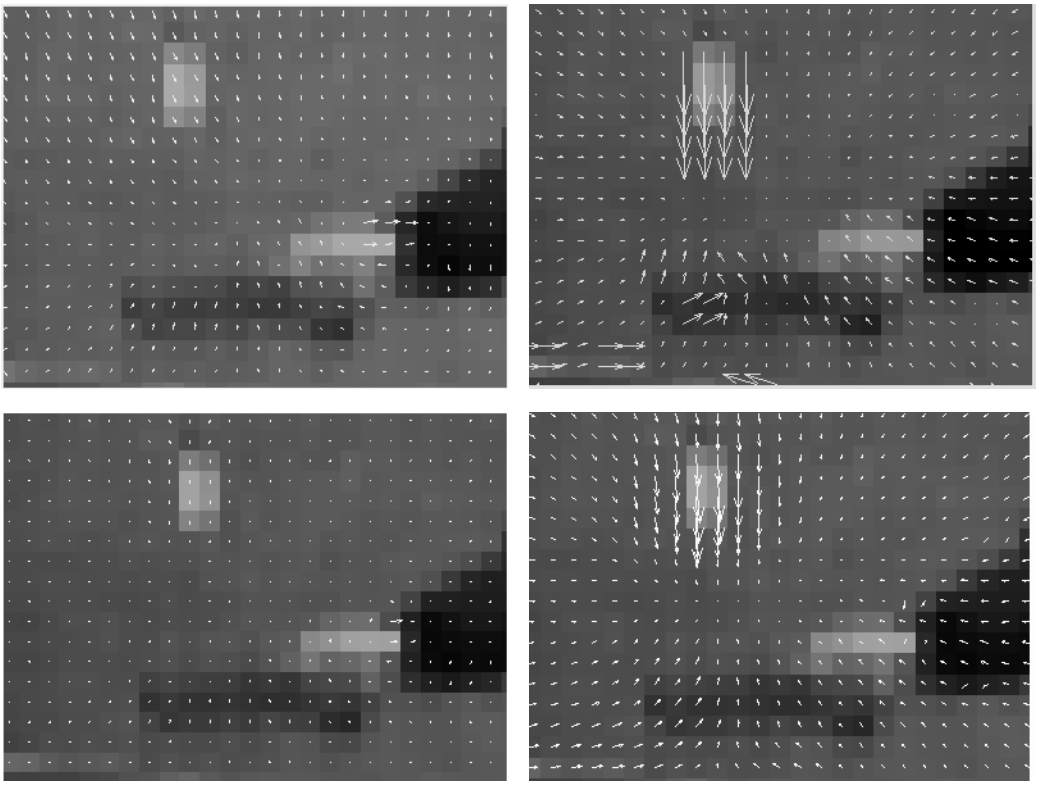


Fig. 6.9 Frame 37. row-wise: OFCE, dense MPEG, linear combination and selective combination (see text)

6.2.3 *"Coast guard"* Sequence

The moving objects of interest in this sequence are the two boats. During the first frames the camera pans, following the smaller boat. Hence, the relative motion of the small boat is expected to be very small in magnitude.

The global optical flow of this sequence cannot be clearly illustrated in paper due to the small amplitude of the motion vectors. To make the presentation easier, in Fig. 6.11 and Fig. 6.12 we present the same zoomed region for frames 35 and 52, showing the boats and the superimposed motion vectors, and the global optical flow field in terms of magnitude. Both frames are shown in Fig. 6.10. The challenge here is to distinguish the

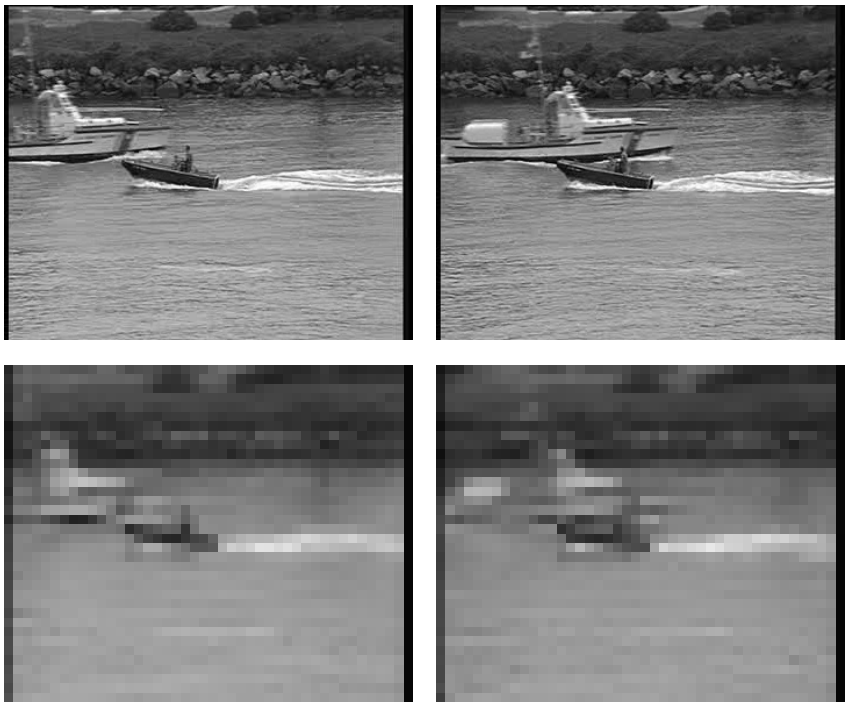


Fig. 6.10 Frames 35 & 52 of the "coast guard" sequence. First row: uncompressed frames. Second row: DC images

boats without being affected by the global motion pattern (camera pan).

The optical flow obtained by the OFCE is overall smooth, as expected due to the camera pan. The motion vectors assigned to the two moving boats do not represent the correct motion, as can be seen by the spurious directions in the images at the first row of Fig. 6.10. The reason is probably the non-informative temporal derivative computation, because of the small temporal variation from frame to frame (short objects' motion). The

dense MPEG field provides a more distinct representation of the three motions present in the scene. It assigns almost zero velocities to the small boat so that one can distinguish it from the global pan. The motion of bigger boat is correctly recovered in terms of magnitude and direction. As expected, the MPEG field suffers from motion artifacts in homogenous regions, as illustrated by "gaps" and intensity discontinuities in the second row of Fig. 6.11. The two approaches we propose provide almost similar results with the selective combination being a little bit more accurate. The global motion pattern is correctly assigned to the background, as shown by the smooth magnitude on the background.

The previous remarks hold the same after seventeen frames, namely in frame 52 of the sequence. Fig. 6.12 illustrates the results. The OFCE adequately recovers the background motion, while the dense MPEG field represents well the motion edges. Both of our approaches combine efficiently the previous characteristics and generate velocity fields closer to the true ones. The selective combination is again more accurate at motion discontinuities, while the additive one tends to oversmooth the moving edges.

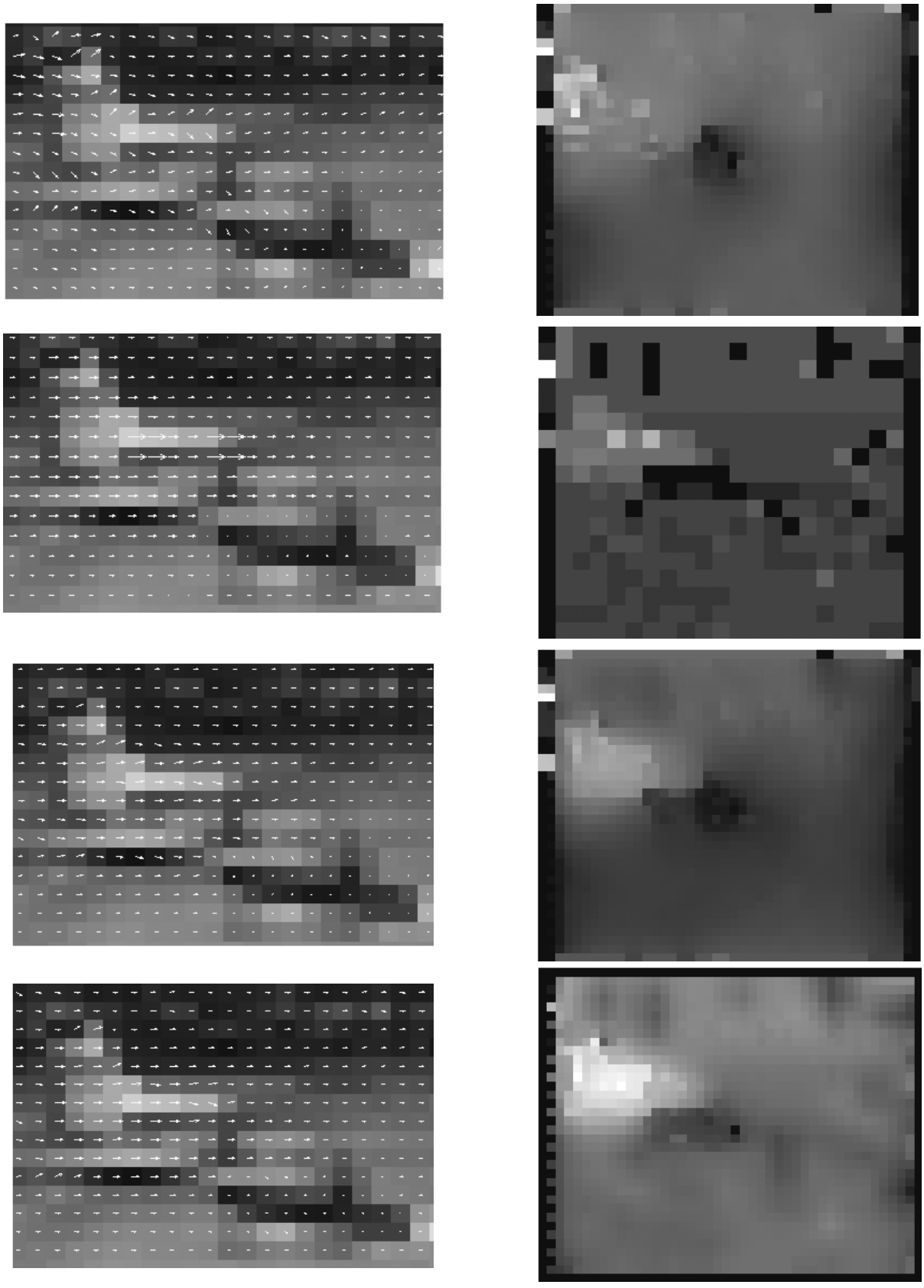


Fig. 6.11 Frame 35. first column: zoomed region with superimposed motion vectors for OFCE, dense MPEG, linear combination and selective combination respectively. Second row: corresponding velocity magnitudes of overall optical flow.

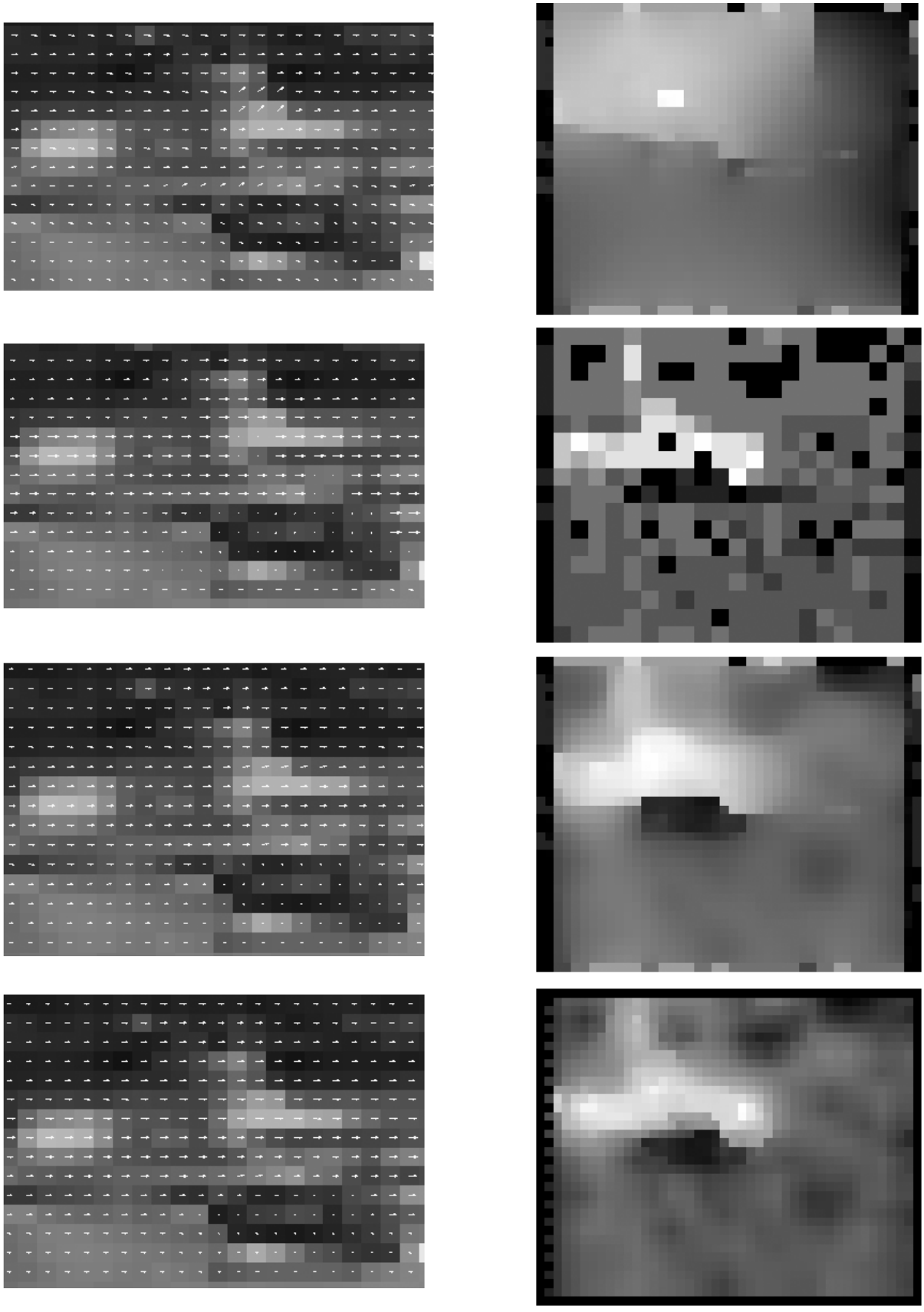


Fig. 6.12 Frame 52. first column: zoomed region with superimposed motion vectors for OFCE, dense MPEG, linear combination and selective combination respectively. Second row: corresponding velocity magnitudes of overall optical flow.

Chapter 7

7. Discussion & Further Work

In the first three chapters, we focus on an introduction of the problem and a review of related work in order to get a grip of the current trends in optical flow estimation. In the last three chapters, we introduce the robust estimation framework, incorporate it in our work and propose our approach to dense optical flow estimation in the compressed domain. In this chapter, we give an evaluation of our ideas and attempt to extend them in order to serve as a starting point for future research.

7.1 Robust Estimation Framework

For many years, the Gaussian assumption serves as a valuable tool for many scientific problems, but the evolution of technology is followed by the need for increasingly complex models to represent underlying distributions. The Gaussian model is still a very important tool for mathematical analysis, but we should be able to cope efficiently with possible deviations from it, as is generally the case. Hence, we adopted the robust estimation framework for recovering optical flow that gives us the ability to introduce outliers and use them in an efficient way to represent motion.

Many researchers use robust statistics tools in order to develop algorithms that remain insensitive to violations of motion assumptions occurring mainly at motion discontinuities. We adopted the framework proposed by Black & Anandan [Black *et al.*, 1996], since it provides a direct way to improve the performance of standard least-squares estimation, reduces smoothing across motion boundaries, detects outliers (violations of motion constraints) and copes with multiple motions.

7.2 OFCE & Additional Motion Constraints

The standard regularization techniques minimize an objective function consisting of a data and a regularizing smoothing term. We introduce two additional constraints, namely the temporal continuity and the MPEG consistency. The temporal constraint has been studied in the literature and is a powerful tool for incremental motion estimation. We incorporate it in to two different ways: as a mathematical constraint in the objective function and as check for the presence of motion discontinuity. The MPEG motion field is used both as initial solution field and as a constraint in the objective function. In this way, we combine the two different motion estimation approaches, namely the pixel-based regularization and the MPEG block matching techniques into a single approach.

We examined several experiments using the constraints in a linear and selective way. A part of them is illustrated in chapters 5 and 6. Most of the recovered optical flow fields inherent the advantages of both the MPEG and OFC motion estimation methods. The improvement achieved by the proposed formulation was mainly illustrated in chapter 5 for sequences exhibiting different motion patterns. The lack of ground truth leads us to evaluate the results in a qualitative way by looking carefully the recovered optical flow and the evolution over time. It is the incremental nature and the velocity fields over a whole sequence that cannot be thoroughly presented in the form of static images.

Generally, the results are sufficiently good for the cases we tested. Nevertheless, our approach does not recover optical flow equally well for all frames of a video sequence. Occlusions/disocclusions, camera zoom/pan etc, state difficulties that may make the algorithm corrupt under specific situations. Such problems are introduced to all video segmentation or analysis approaches. The limits to our approach come mainly from the initial information we use, namely the DC images and the MPEG motion field. The use of DC images saves computational time due to the small spatial extend, but introduces strong smoothness on the intensity image. Hence, relative small moving objects may disappear or intermix with the background making the correct motion recovery impossible. Similar problems exist even for large moving objects. If a moving edge lies inside an MPEG block (8×8 pixels) then the DC averaging may introduce large inaccuracies in the final intensity field, which influence negatively the OFC motion

estimation. On the other hand, the possible poor quality of MPEG velocity vectors and their sparse distribution pose additional difficulties to the accurate solution recovery.

Our method provides a framework for combining information from different sources available to produce spatiotemporal consistent optical flow field that represents the true motion field as accurate as possible at this level of resolution (DC resolution). It combines efficiently both motion estimation methods, compensates for their possible artifacts, and generates an improved MPEG optical flow field. Motion segmentation can be seen as a two-stage problem: the first stage is the low level processing involving the extraction of motion information; the second level classifies or clusters previous information in an intelligent way. Hence, our algorithm may provide the platform (low level processing) for more efficient and accurate segmentation or analysis (high level processing) of the velocity field that is generated directly from compressed domain features.

7.3 Further Work

The work presented in this thesis is concerned with dense motion estimation (at the level of blocks) from compressed domain data. Although good results are obtained, some open issues remain. These are outlined below with possible indications on how they might be solved.

Accuracy on motion edges is a hard problem for many motion estimation techniques. The combination of gradient and block matching methods we attempt alleviates moving edge artifacts, but further improvement is always desirable. One possible approach worth of further investigation is the incorporation of intensity information to generate an explicit boundary map. Intensity can be easily imported in our robust estimation framework in a form of a functional biasing the objective function. Additionally, we could use intensity information in the selective combination of constraints to ensure the presence of a moving edge and provide a more efficient way to tune the developed algorithm.

The spatial information as used in the algorithm requires the computation of partial derivatives. Derivative calculation has seldom received proper attention in optical flow estimation. Crude derivative estimators are widely used. Consequently, OFC

methods may break down near motion boundaries due to non-robust derivative estimation. Pointing out this limitation, we could use a more robust method to calculate high-quality derivatives like the one of Ye & Haralick. In [Ye *et al.*, 2000] they calculate derivatives from an explicit 3D facet model and discuss on the achieved improvements.

A natural extension of our method would be towards the implementation of robust regression using affine models. The goal is to recover the affine parameters of a motion model that minimize a criterion similar to that used for the robust regularization approach we adopt. Black & Anandan in [Black *et al.*, 1996] propose such an approach and present several results. We could easily extent their work by incorporating the constraints and the ideas developed in the previous chapters and apply them in the compressed domain.

The combination of MPEG and temporal motion fields seems promising and is worth of further analysis. Weaknesses of our approach may be eliminated by a finer combination. Hard to face cases, like occlusion/disocclusion, illumination shading, appearance of new object etc. may be handled more efficiently with appropriate MPEG-temporal information fusion. Additionally, scene changes may be correctly located at frames where the temporal field relating the previous to the current frame deviates as an overall distribution from that of the MPEG field that relates the current to the next frame.

Our approach was shown to perform well over a range of different motion scenarios. Nevertheless, a complete evaluation can only come from extended testing and application of the algorithm. The basis of our work, namely the robust motion estimation framework in a multiresolution scheme, has been widely studied and established in the literature and seems to achieve good results. Its application in the compressed domain has not been fully explored yet. Hence, we hope that our work can serve as a basis to develop further the proposed ideas and incorporate new and powerful motion constraints in the compressed domain. As major areas of application, we see (a) the direct motion parameter estimation from compressed video and (b) the video segmentation in the compressed domain.

REFERENCES

[Anandan, 1989] P. Anandan. *A computational framework and an algorithm for the measurement of visual motion*. Int. Journal Comp. Vision 2, pp. 283-310,1989.

[Ardizzone *et al.*, 1996] E. Ardizzone, M. La Cascia. *Video indexing using optical flow field*. ICIP'96, Vol. 3, pp. 831-834, 1996.

[Ardizzone *et al.*, 1998] E. Ardizzone, M. La Cascia. *Video indexing using MPEG motion compensationvectors*. Proc. IEEE ICMCS'98, Jun1999.

[Bab-Hadiashar *et al.*, 1998] A. Bab-Hadiashar, D. Suter. *Robust Optic Flow Computation.* IJCV, Vol. 29, No. 1, pp. 59-77, 1998.

[Barron *et al.*, 1994] J.L. Barron, S.S. Beauchemin, D.J. Fleet. *Performance of optical flow techniques.* IJCV, Vol. 12, No 1, pp. 43-77, 1994

[Battiti *et al.*, 1991] R. Battiti, E. Amaldi, C. Koch. *Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy*. IJCV, Vol. 6, pp. 133-145, 1991.

[Bergen *et al.*, 1992] J.R. Bergen, P.J. Burt, R. Hingorani, S. Peleg. *Three-frame algorithm for estimating two-component image motion*. IEEE PAMI, Vol. 14, No. 9, pp. 886-896, 1992

[Bertero et al., 1988] M. Bertero, T.A. Poggio, V. Torre. *Il-posed problems in early vison*. Proc. IEEE, Vol. 76, No. 8, pp. 869-889, Aug 1988.

[Bhatt *et al.*, 1997] B. Bhatt, D. Birks, D. Hermreck. *Digital television: making it work.* IEEE Spectrum, Vol. 34, No. 10, pp. 19-28, Dec 1997.

[Biemond *et al.*, 1987] J. Biemond, L. Looijenga, D. Boekee, R. Plompen. *A Pel-Recursive Wiener-based Displacement Estimation Algorithm*. Signal Processing, Vol. 13, pp. 399-412, 1987.

[Black *et al.*, 1990] M.J. Black, P. Anandan. *A model for the detection of motion over time*. In Proc. ICCV-90, pp. 33–37, Dec 1990.

[Black *et al.*, 1991] M.J. Black, P. Anandan. *Robust dynamic motion estimation over time*. In Proc. CVPR-91, pp. 296–302, Jun 1991.

[Black *et al.*, 1993] M.J. Black, P. Anandan. *A framework for the robust estimation of optical flow*. ICCV'93, pp. 231-236, May 1993.

[Black *et al.*, 1996a] M.J. Black, P. Anandan. *The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields*, Computer Vision and Image Understanding, Vol. 63, No. 1, pp. 75-104, January 1996.

[Black *et al.*, 1996b] M.J. Black, A. *Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision*. International Journal on Computer Vision, Vol. 19, No. 1, pp. 57-92, Jul 1996.

[Black, 1994] M.J. Black. *Recursive non-linear estimation of discontinuous flow fields*. ECCV'94, Springer-Verlag, Vol. 800, pp. 138-145, 1994

[Black, software] M.J. Black. http://www.cs.brown.edu/people/black/code.html.

[Blake *et al.*, 1987] A. Blake, A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, USA, 1987.

[Bonzanini *et al.*, 2000] A. Bonzanini, R. Leonardi, P. Migliorati. *Semantic video indexing using MPEG motion vectors*. EUSIPCO'00, 2000

[Burt *et al.*, 1983] P.J. Burt, E.H. Adelson. *The Laplacian pyramidas a compact image code*. IEEE Trans. On Communications 31, pp.532-540, 1983.

[Camus, 1997] T. Camus. *Real-time quantized optical flow*. Real-Time Imaging (special issue on Real-Time Motion Analysis), Vol. 3, pp.71–86, 1997.

[Chang *et al.*, 1997] S.-F. Chang, J. Smith, M. Beigi, A. Benitez. *Visual information retrieval from large distributed online repositories*. CACM, Vol. 40, No. 12, pp. 63-71, Dec 1997.

[Digital libraries, 1995] Digital libraries, Special Issue of Communications of the ACM, Vol. 38, No. 4, Apr 1995.

[Docherty *et al.*, 1991] M. O'Docherty, C. Daskalakis. *Multimedia information systems: the management and semantic retrieval of all electronic datatypes*. The Computer Journal, Vol. 34, No. 3, pp. 225-238, Apr 1991

[Ehlers et al., 1989] M. Ehlers, G. Edwards, Y. Bedard. *Integration of remote sensing with geographic information systems: a necessary evolution*. Photogrammetric Engineering and Remote Sensing, Vol. 55, No. 11, pp. 1619-1627, 1989.

[Enkelmann *et al.*, 1988] W. Enkelamm, H. Nagel. *Investigation of multigrid algorithms for estimation of optical flow fields in image sequences*. CVGIP, Vol. 43, pp. 150-177, 1988.

[Fleet *et al.*, 1990] D.J. Fleet, A.D. Jepson. *Computation of component image velocity from local phase information*. IJCV, Vol. 5, No. 1, pp.77-104, 1990.

[Funkalea e*t al.*, 1996] G. Funkalea, A. Gupta. *The use of hybrid models to recover cardiac wall-motion from MR_images*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.625-630, 1996.

[Geman *et al.*, 1984] S. Geman, D. Geman. *Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images*. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol 6, pp. 721-741, Nov 1984.

[Giachetti *et al.*, 1998] A. Giachetti, M. Campani, and V. Torre. *The use of optical flow for road navigation*. IEEE Trans. on Robotics and Automation, Vol. 14, No. 1, pp. 34-48, Feb 1998.

[Glazer, 1987] F.C. Glazer. *Computation of optical flow by multilevel relaxation*. Technical Report COINS-TR-87-64, Univer. Of Mass., 1987.

[Hampel *et al.*, 1986] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. Robust Statistics: The Approach Based on Influence Functions. John Wiley and Sons, New York, NY, 1986.

[Horn *et al.*, 1981] B.K.P Horn, B.G. Shunck. *Determining optical flow.* Artificial Intelligence, 17(1-3), pp. 185-203,Aug. 1981

[Horn, 1986] B.K.P. Horn. *Robot Vision*. The MIT Press, Cambridge, Massachusetts, 1986.

[Huang *et al.*, 1994] J. Huang, R. Merserau. *Multi-Frame Pel-Recursive Motion Estimation for Video Image Interpolation*. Proc. ICIP'94, Vol. 2, pp.267-271, 1994.

[Huber, 1981] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, NY, 1981.

[Irani et al. 1993] M. Irani, S. Peleg. *Motion analysis for image enhancement: Resolution, occlusion and transparency*. Journal of Video Communication and Image Representation, Vol. 4, pp. 324-335, Dec 1993.

[**ISO**/IEC, 1993] Joint Technical Committee **ISO**/IEC JTC 1.ISO/IEC 11172-1:1993. *Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s***.** 1993

[**ISO**/IEC, 1993] Joint Technical Committee **ISO**/IEC JTC 1.ISO/IEC 13818-2:1996. *Information technology—Generic coding of moving pictures and associated audio information: Video*. May 1996.

[Jain *et al.*, 1981] J.R. Jain, A.K. Jain. *Displacement measurement and its application in Interframe Image Coding*. IEEE Transactions on Communications, Vol. COM-29, pp. 1799-1808,Dec 1981.

[Jepson et al., 1993] A.D. Jepson, M.j. Black. *Mixture Models for optical flow computation.* In IEEE Proc. Of CVPR, pp. 760-761, New York, Jun 1993.

[Kandel *et al.*, 1995] E.R. Kandel, J.H. Schwartz, T.M. Jessell. *Essentials of Neural Science and Behavior.* Appleton & Lange.

[Kearny *et al.*, 1987] J.K. Kearny, W.B. Thompson, D.L. Boley. *Optical flow estimation: An error analysis of gradient-based methods with local optimization*. IEEE PAMI, Vol. 9, pp. 229-244, 1987.

[Kobla *et al.*,1997] V. Kobla, D. Doermann, C. Faloutsos. *Compressed domain video indexing techniques using DCT and motion vector information in MPEG video*. In Proc. of the SPIE, volume 3022, pp. 200-210, 1997.

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

[Konrad *et al.*, 1992] J. Konrad, E. Dubois. *Bayesian estimation of motion vector fields*. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 910-927, Sep 1992.

[Koprinska *et al.*, 2001] I. Koprinska, S. Carrato. *Temporal Video Segmentation: A Survey.* Signal Processing: Image Communication, Vol. 16, pp. 477-500, 2001.

[Leuven *et al.*, 2001] J. van Leuven, M.B. van Leeuwen, F.C.A. Groen. *Real-Time Vehicle Tracking in Image Sequences*. In: *Proc. IEEE Instrumentation and Measurement Conference* , Budapest, Hungary, May 21-23, 2001, pp.2049-2054

[Lewis *et al.*, 1993] F. Lewis, C. Abdallah, D. Dawson. *Control of robot* manipulators. Macmillan, 1993

[Lindgren, 1993] B.W. Lindgren. *Statistical Theory*, Chapman and Hall, London, 1993.

[Little *et al.*, 1989] J.J. Little, A. Veri. *Analysis of differential and matching methods for optical flow*. IEEE Workshop on Visual Motion, Irvine CA, pp. 173-180, 1989.

[Little *et al.*, 1993] T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, F.W. Reeve, D.H. Schelleng, D. Venkatesh. *A digital video-on-demand service supporting content-based queries*. Proc. of 1st International Conference on Multimedia, pp. 427-436, Aug. 1993.

[Lucas *e t al.*, 1981] B.D. Lucas, T. Kanade. *An iterative image registration technique with an application to stereo vision*. In Proc. 7th IJCAI, Vancouver,B. C., Canada, pp. 674-479, 1981

[Mandal *et al.*, 1999] M.K. Mandal, F. Idris, S. Panchanathan. *A critical evaluation of image and video indexing techniques in the compressed domain*. Image and Vision Computing Journal, Vol. 17, No. 7, pp. 513-529, May 1999.

[Memin *et al.*, 1998] E. Memin, P. Perez. *Dense estimation and object-based segmentation of the optical flow with robust techniques*. IEEE Image Processing, Vol. 7, No. 5, pp. 703-719, May 1998

[Mitchel *et al.*, 1997] J.L. Mitchell, W.B. Pennebaker, Fogg C.E., LeGall D.J.. *MPEG Video Compression Standard*. Chapman & Hall, New York, 1997.

[Mukawa, 1990] N. Mukawa. *Estimation of shape, reflection coefficients and illuminant direction from image sequences*. In Proc. Of ICCV, pp. 507-512, Osaka, Japan, Dec 1990.

[Musmann *et al.*, 1985] H.G. Musmann, P. Pirsch, H.J. Grallert. *Advances in pitcure coding*. Proc. IEEE, Vol. 73, No. 4, pp. 523-548, Apr 1985.

[Nagel *et al.*, 1989] H.H. Nagel, W. Enkelmann. *An investigation of smoothness constraints for the estimation of the displacement vector fields from image sequences*. IEEE Trans. PAMI 8, pp.565-593, 1989.

[Nagel, 1983] H.H. Nagel. *Displacement vectors derived form second-order intensity variations in image sequences*. CGIP, pp.85-117, 1983.

[Nagel, 1987] H.H. Nagel. *On the estimation of optical flow: Relationsbetween different approaches and some new results*. AI 33, pp. 299-324, 1987.

[Odobez *et al.*, 1995] J.M. Odobez, P. Bouthemy. *Robust multiresolution estimation of parametric motion models*. Journal of Visual Communication and Image Representation, Vol. 6, No. 4, pp. 348-365, 1995.

[Papoulis, 1984] A. Papoulis. *Probability, random variables and stochastic processes, 2^d ed.* New York: McGraw-Hill, 1984.

[Press *et al.*, 1988] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling. *Numerical recipes in C: The art of scientific computing*. Cambridge Univ. Press, Cambridge, 1988.

[Quistgaard, 1997] J.U. Quistgaard. *Signal acquisition and processing in medical diagnostic ultra-sound*. IEEE Signal Processing Magazine, Vol. 14, No. 1, pp. 67-74, 1997.

[Revalski, 1997] J.P. Revalski. *Hadamard and strong well-posedness for convex programs*. Society for Industrial and Applied Mathematics, Vol. 7,No. 2, pp. 519-526, May 1997

[Rousseeuw *et al., 1987*] P. J. Rousseeuw, A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley, New York, 1987.

[Sim *et al.*, 1998] D.-G. Sim, R.-H. Park. *Robust reweighted MAP motion estimation*. IEEE Trans on PAMI, Vol. 20, No. 4, Apr 1998.

[Simoncelli *et al.*, 1991] E.P. Simoncelli, E.H. Adelson, D.J. Heeger. *Probability distribution of optical flow*. In IEEE Proc. Of CVPR, pp. 310-315, California, Jun 1991.

[Srinivasan *et al.*, 1985] R. Srinivasan, K.R. Rao. *Predictive Coding Based on Efficient Motion Estimation.* IEEE Transactions on Communications, Vol. COM-33, pp. 888-896, Aug 1985.

[Stiller, 1997] C. Stiller. *Object-based estimation of dense motion fields*. IEEE Trans. Image Processing. Vol. 6, No. 2, pp. 234-250, Feb 1997.

[Tikhonov et al., 1977] A. Tikhonov, V. Arsenin. *Solutions of ill-posed problems*. Winston and Sons, 1977.

[Tull *et al.*, 1996] D.L. Tull and A.K. Katsaggelos. *Regularized Estimation of Occluded Displacement Vector Fields.* IEEE International Symposium on Circuits and Systems, Atlanta, GA, May 1996.

[WSSAE] *The Working Site for Sequences and Algorithm Exchange*. www.tele.ucl.ac.be/EXCHANGE/.

[Xiong *et al.*, 1998] W. Xiong, J.C.-M. Lee. *Efficient scene change detection and camera motion annotation for video classification*. Computer Vision and Image Understanding, Vo. 71, No. 2, pp. 166-181, Aug 1998

[Ye *et al.*, 2000] Ye Ming, R.M. Haralick. *Optical flow from a least-trimmed squares based adaptive approach*. Proc. ICPR, Vol. 3, pp. 1052-1055,2000.

[Ye *et al.*, 2001] Ye Ming, R.M. Haralick. *Local gradient, global matching, piecewise-smooth optical flow*. CVPR01, pp. 712-717, 2001

[Ye *et al.*, 2002] Ye Ming, R.M. Haralick, L.G. Shapiro. *Estimating optical flow using a global matching formulation and graduated optimization*. ICIP, 2002.

[Yeo *et al.*, 1995a] B.L. Yeo, B. Liu. *On the extraction of DC sequence from MPEG compressed video*. ICIP'95, Vol. 2, pp.260-263, 1995.

[Yeo *et al.*, 1995b] B.L. Yeo, B. Liu. *Rapid scene analysis on compressed video*. IEEE Trans. On Circuits ans Systems for Video Technology., Vol. 5, No. 6, pp. 533-544, Dec 1995.