SPATIO-TEMPORAL SEGMENTATION AND OBJECT TRACKING: AN APPLICATION TO SECOND GENERATION VIDEO CODING

THÈSE N° 1618 (1997)

PRÉSENTÉE À LA SECTION DE SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

Fabrice MOSCHENI

Ingénieur physicien diplômé EPF

originaire de Sainte-Croix (VD)

acceptée sur proposition du jury:

Prof. M. Kunt, directeur de thèse
Dr T. Ebrahimi, corapporteur
Prof. B. Girod, corapporteur
Prof. A. Lippman, corapporteur
Prof. D. Mlynek, corapporteur
Prof. T. Pun, corapporteur

Lausanne, EPFL
1997

A mes parents

Remerciements

Certains auteurs ont souligné que le but n'est pas tout, mais que le chemin pour y parvenir est également important. Cette thèse n'échappe pas à cette vérité. Elle consacre le but que je m'étais fixé, tout en ne représentant que la partie visible d'un long et enrichissant cheminement.

Je voudrais remercier le Professeur Murat Kunt pour avoir rendu cette aventure possible dans le cadre du Laboratoire de Traitement des Signaux (LTS). Sa confiance et son soutien ont joué un rôle-clé dans l'accomplissement de ce travail. Par ailleurs, l'esprit d'excellence régnant au LTS a été une source de motivation à lui seul.

Mes remerciements vont à Messieurs les Professeurs M. Hasler, B. Girod, A. Lippman, D. Mlynek, T. Pun ainsi qu'au Docteur T. Ebrahimi pour avoir fait partie de mon jury de thèse et diligemment évalué et commenté mon travail.

Finalement, mes remerciements vont à ceux que j'ai cotoyés tout au long de ce travail. Je pense aux collaborateurs du LTS et, en général, à tous ceux qui m'ont été proches durant ces années. Mes pensées vont, en particulier et dans le désordre, à Gilles, Frédéric, Anne, Erika, Jean-Luc, Pablo, Stefan, Sushil, Francesco et Mathieu. Je finirai en exprimant toute ma gratitude à mes parents dont la présence m'a apporté le soutien et l'encouragement nécessaires.

Contents

Version abrégée

L'information visuelle a pris une place prépondérante dans notre société. Les technologies numériques de l'information permettent d'entrevoir de nouvelles applications dans des domaines aussi variés que les communications, le commerce ou l'industrie du spectacle. Pour ce faire, l'information visuelle doit être représentée sous une forme qui permette à l'utilisateur d'interagir avec elle. Ceci est particulièrement crucial dans le domaine de la télévision numérique lors de la compression de séquences vidéo.

La portée et le type d'interactions possibles avec l'information visuelle dépendent de la forme sous laquelle cette dernière est représentée. Jusqu'à nos jours, une représentation canonique sous forme de *pixels* a été utilisée. Cette représentation n'est, en fait, qu'un artefact dû au processus de capture de l'image. La représentation canonique est incapable de rendre compte du contenu sémantique de l'image. Les possibilités d'interaction avec l'information visuelle offertes à l'utilisateur sont donc restreintes.

Dans cette thèse, on se propose de représenter l'information visuelle sous une forme sémantique. Dans ce but, l'image est decomposée dans l'ensemble des objets qui la forment. Ce type de représentation n'est pas assujetti au processus de capture et dérive directement du contenu sémantique de l'image. Ceci offre à l'utilisateur une grande palette d'interactions avec l'information visuelle.

L'idée directrice de ce travail est de découper une image automatiquement en l'ensemble des objets la constituant et de poursuivre ces derniers dans les images successives de la séquence vidéo. Pour ce faire, deux points majeurs ont été identifiés. Le premier est celui de définir les objets, la contrainte étant que seul l'information présente dans deux images successives de la séquence vidéo peut être utilisée. Ce problème est résolu à travers une approche "*split-and-merge*". L'image est d'abord décomposée en de petites régions qui sont spatio-temporellement homogènes. Pour ce faire, une technique "*top-down*" est présentée. Cette dernière combine l'information spatiale, temporelle et celle de changement pour affiner les régions jusqu'à ce qu'elles soient spatio-temporellement cohérentes. Ces régions sont alors utilisées pour construire les objets qui constituent la scène. A cette fin, les régions sont fusionnées itérativement à travers une technique "*bottom-up*". La propension des régions à former un objet est évaluée à travers l'utilisation de l'information temporelle et spatiale. Le deuxième point majeur identifié dans le cadre de cette thèse est celui de définir et poursuivre les objets dans les images successives de la séquence vidéo. Une technique est proposée qui assure la cohérence des segmentations consécutives grâce à la combinaison de l'information courante avec l'information provenant des images passées. De plus, une technique pour identifier les différents objets est présentée, ceci permettant de poursuivre ces derniers tout au long de la séquence vidéo. Cette procédure d'identification est basée sur les caractéristiques spatiales, temporelles et spatio-temporelles des objets.

Basé sur la représentation de l'information visuelle en terme d'objets, un schéma de codage pour séquences vidéo est décrit. Il appartient à la deuxième génération de techniques pour le codage de séquences vidéo. Il

permet une compression efficace de l'information visuelle, tout en offrant une grande palette d'interactions possibles avec cette dernière. En particulier, l'utilisateur est à même de décoder progressivement chaque objet, ainsi que de composer à sa guise le contenu de la scène décodée.

Abstract

Visual information is taking a predominant place in our society. With the advent of digital technologies, visual information will find new applications in domains ranging from communication, commerce to entertainment. New functionalities will be required that permit an extended interaction with the visual information. In particular, this is the case for digital television and the related problem of video sequence compression.

The extent of possible interactions depends on the manner in which the visual information is represented. Up to now, the canonical representation is used. Also referred to as the waveform representation, it is a technical artifact of the image capture procedure. This representation severely restricts the functionalities available to the end-user. The latter is unable to freely manipulate or customize the received visual information.

In this dissertation, we propose to describe the visual information through a semantically meaningful representation. This representation directly derives from the scene content, which is decomposed in terms of its constituting objects. Consequently, the viewing process is totally disconnected from the image capture procedure. This permits full interactivity with the visual information, leading to enhanced functionalities for the end-user.

The essence of this dissertation is to automatically define the objects forming the scene and to automatically track them through the video sequence. Two main issues are identified: the initial segmentation of the objects and their tracking. The first issue deals with segmenting the scene into its constituent objects. This has to be performed on the basis of the information available in two consecutive frames. In order to solve the problem, a "*split-and-merge*" approach is used. First, the image is segmented into small, spatio-temporally homogeneous regions. This is achieved through a top-down approach where spatial, temporal and change information is combined. These regions are used as a starting point to define the objects forming the scene. A bottom-up approach is used which iteratively merges the regions. The propensity of the regions to form an object is assessed in terms of both spatial and temporal information. The second issue deals with segmenting and tracking the objects in the successive frames. The coherence between the successive segmentations is ensured by using past and current information. Also, the different objects composing the scene are identified throughout the video sequence. This identification relies on temporal, spatial and spatio-temporal features of the objects.

The proposed representation of the visual information finds a natural application in video coding. A second generation video coding scheme is presented which combines compression efficiency with extended functionalities. In particular, scalable coding is achieved in terms of both scene content and object coding quality.

Notation and Terminology

α	significance level
$\mathcal{A}$	ensemble of the graph links
$Area(X)$	area of the region X
Δt	time interval
DFD	displaced frame difference
f_C	constraint factor in the recursive spatio-temporal similarity
f_L	luminance factor in the spatio-temporal similarity
G	graph such as $G = (\mathcal{R}, \mathcal{A})$
H_0	null hypothesis
H_1	alternative hypothesis
HVS	human visual system
$I(\vec{r}, t)$	pixel intensity value at position $\vec{r}$ and time t
λ	pondering factor in the Modified Kolmogorov-Smirnov test
$\vec{M}$	fully parametric motion model with n parameters
MAD	mean absolute difference
MKS test	Modified Kolmogorov-Smirnov test
MSE	mean square error
$\mathcal{O}$	ensemble of the objects forming the scene
O_j	j^{th} object in $\mathcal{O}$
P_{AB}	past spatio-temporal similarity between the regions A and B
$PSNR$	peak signal to noise ratio
$\mathcal{R}$	ensemble of the regions to be merged
R_j	j^{th} region in $\mathcal{R}$
$recSim(AB)$	recursive spatio-temporal similarity between the regions A and B
S_{AB}	spatial similarity between the regions A and B
$Sim(AB)$	spatio-temporal similarity between the regions A and B
t	time
T_{AB}	temporal similarity between the regions A and B
$\vec{V}$	set of affine invariants
$w(x)$	weighting function in the Modified Kolmogorov-Smirnov test

List of Figures

Chapter 1

Introduction

1.1 Statement of the problem

Visual information occupies a predominant place in our modern society. It is estimated to represent eighty percent of the total amount of information we perceive. This trend is expected to intensify with the advent of digital technologies. Even more than before, visual information will find new applications in domains ranging from communication, commerce to entertainment. One's attention is drawn to the domain of digital television and the related problem of video sequence compression. Due to its ubiquity, the visual information has a deep social impact. The form in which it is delivered constitutes a major issue. More precisely, the manner in which the information is represented is crucial. Indeed, the representation dictates the kinds of possible interactions the user may or may not have with the information.

Up to now the visual information has commonly been conveyed under its waveform representation. Each frame is portrayed as an ensemble of picture elements referred to as *pixels*. This representation directly derives from the technical constraints imposed while capturing the scene. The pixels may thus be seen as technical artifacts. They are therefore semantically meaningless with respect to the scene content. The waveform representation is the cornerstone of many techniques. The most striking examples may be found in the domain of video coding techniques with the recent standards MPEG-1 [74, 89], MPEG-2 [2, 90], H.261 [135] and H.263 [77]. Although a good compression efficiency is reached, the waveform representation severely restricts the functionalities available to the end-user. The latter is unable to freely manipulate or customize the received visual information. Most generally, the waveform representation only permits limited interaction with the visual information. As the representation is not related to the scene content, the user is unable to interact with it.

In order to allow interaction, the representation of the visual information should aim at being semantically meaningful, while being malleable. The former concept directly refers to the working principles of the human visual systems. The manner in which the human beings perceive the visual information must be taken into account. The representation should thus rely on primitives which have a semantic meaning [87]. Many types of primitives have been proposed such as edges, textures or regions. However, a full interaction between the user and the representation is reached only when the primitives are chosen to be the objects forming the scene. They permit to fully decompose the visual information into its constituent parts. Also, they are characterized by their spatial and temporal coherence. The representation of a scene in terms of its constituent objects implies a deep understanding of the scene. Based on such a representation, functionalities such as content based retrieval from libraries, database browsing, selection of appropriate coding qualities depending on the object, content based image manipulation are possible.

This dissertation deals with the development of an algorithm to detect the objects forming the scene and to track them through time. Each frame of the video sequence is decomposed into its constituent objects, which are identified as they evolve through the sequence. The resulting representation of the visual information permits a total interaction with the content of each individual frame. Also, the tracking of each object allows to interact with the content of the video sequence as a whole. The proposed representation is used as the basis for a video coding technique. This representation allows a fully scalable coding in terms of scene content and object quality.

1.2 Investigated approach

In this work, the aim is to derive, automatically, a representation of the scene at hand that is both semantically meaningful and malleable. To this end, the video sequence is decomposed in terms of objects. These are primarily defined as ensembles of connected pixels characterized by a coherent motion. The

objects may also be typified through a certain spatial coherence, although this is secondary to the temporal one. Thus, the objects may be seen as entities which are spatio-temporally coherent. Consequently, the methods which detect them are referred to as spatio-temporal segmentation techniques. In this dissertation, we propose an ensemble of algorithms which permit the segmentation and the tracking of the objects forming the scene. No *a priori* information is used nor are any constraints imposed about the scene content. In particular, no assumption is made about a static background or a predefined number of objects in the scene.

The proposed approach may be split into two successive steps. The first step is to define a procedure for spatio-temporal segmentation that only relies on two consecutive frames. The proposed procedure is a combination of a top-down algorithm with a bottom-up algorithm. The top-down algorithm defines a set of regions which are spatio-temporally homogeneous. In particular, the regions incorporate the static information of the scene, while being temporally homogeneous. This is achieved by combining spatial, temporal and change information. The regions are split until the desired characteristics of homogeneity are reached. In order to define the objects, the regions are then merged together through a bottom-up algorithm. The merging is based on the mutual spatio-temporal similarity of the regions. The spatio-temporal similarity is composed of a temporal and a spatial contribution, the emphasis being on the former. The spatio-temporal similarity measure is developed in the framework of hypothesis testing. This leads to the definition of a test statistic for each type of information contributing to the similarity. In particular, we propose a Modified Kolmogorov-Smirnov test for the temporal information. It is characterized by its efficient use of temporal information in both the residual distribution and the motion parametric representation. The region merging process is based on a weighted, directed graph. Two complementary graph clustering rules are proposed, namely, the strong rule and the weak rule. Their derivation aims at taking advantage of the natural structures existing in the graph. Also, the rules take into account the possible errors and uncertainties reported by the graph. These rules are applied successively and are embedded into a dynamic strategy for updating the graph. The spatio-temporal segmentation procedure permits to automatically partition the scene into its constituent objects. Its results may be used as the starting point for the recursive spatio-temporal segmentation and object tracking explained hereafter.

In the second step, an algorithm for recursive spatio-temporal segmentation and an algorithm for object tracking are proposed. The recursive spatio-temporal segmentation algorithm aims at deriving coherent consecutive segmentations while avoiding time-delays. It is composed of three successive steps: partition projection, region validation and constrained spatio-temporal segmentation. The recursive spatio-temporal segmentation algorithm is proposed in conjunction with an object tracking algorithm, the aim being to have a strong interaction between them. When combined, the two algorithms allow a stable partitioning of the successive frames. Furthermore, the different objects of the video sequence are identified through time.

The algorithm for recursive spatio-temporal segmentation partitions the current frame on the basis of the segmentation obtained for the previous frame. The objects forming the previous frame are tentatively projected onto the current frame. The decision whether the projection is acceptable is based on spatial and temporal criteria. Through this partition projection procedure, an initial guess of the current segmentation is defined. This procedure is performed very conservatively as only secure pieces of information are used. The region validation procedure then takes place. Among the regions obtained through partition projection, it aims at distinguishing which regions are valid objects and which are not. The non-valid regions are split through an oversegmentation procedure. This is followed by the constrained spatio-temporal segmentation procedure. It is based on the spatio-temporal segmentation algorithm proposed above for segmenting two consecutive frames. The difference lies in the assessment of the similarity between the regions. Both previous and current spatio-temporal information is fed to

the merging procedure so as to stabilize the current segmentation process. The resulting segmentation is used as a starting point for the object tracking algorithm. This last algorithm tackles the correspondence problem existing between the previously detected objects and the objects in the current segmentation. To that end, each object is characterized through temporal, spatial and spatio-temporal features. The objects in the current frame are thus identified and assigned the labels of the previously detected objects to which they correspond. This is achieved through the testing of a hierarchy of correspondence hypotheses. When judging it necessary, the tracking process is empowered to modify the current segmentation. It may merge objects when it detects that the current segmentation is oversegmented. Similarly, it may recreate objects which have been lost.

The proposed representation of the visual information is utilized in the framework of video coding. A second generation coding scheme is proposed which permits a fully content oriented coding of the scene. Namely, the end user is able to fully interact with the decoded sequence as each object is coded independently. Furthermore, the objects are coded in a fully scalable fashion. This permits a layered decoding of the bitstream related to each object.

1.3 Organization of the dissertation

The dissertation is organized as follows. **Chapter 2** presents the different types of techniques proposed to estimate the motion of a region. The region is assumed to be known. The basic concepts of motion are described. Hence, different motion models and estimation techniques are reviewed. **Chapter 3** reviews the techniques proposed in the literature for spatio-temporal segmentation and object tracking. First, the notion of spatio-temporal segmentation is introduced. A dichotomy is then made between the techniques for spatio-temporal segmentation based on only two frames and the techniques for spatio-temporal segmentation using multiple frames. The techniques in the former category are further classified as top-down techniques or bottom-up techniques. The chapter ends with the issue of object tracking. In **Chapter 4**, we present an algorithm to segment the scene into spatio-temporally homogeneous regions. The algorithm relies on a top-down approach and only relies on two consecutive frames. The regions are rendered homogeneous by using spatial, temporal and change information. Furthermore, the process is made robust against disocclusion phenomena. **Chapter 5** addresses the problem of decomposing the scene into its constituent objects. We present a spatio-temporal segmentation algorithm which uses only two consecutive frames. Relying on a bottom-up approach, the proposed technique assumes that a set of initial regions is available. These regions are iteratively merged so as to define the objects composing the scene. The merging uses both temporal and spatial information and relies on a graph representation. The segmentation is performed automatically as no *a priori* is required. In **Chapter 6**, we propose an unsupervised algorithm for recursive spatio-temporal segmentation and an unsupervised algorithm for object tracking. The two algorithms are presented together as they intimately interact. The recursive spatio-temporal segmentation algorithm combines past and current spatio-temporal information. This ensures a robust segmentation of the current frame. The tracking algorithm identifies the objects constituting the current frame and puts them into correspondence with the objects found in former frames. If necessary, the tracking algorithm is empowered to modify the current segmentation. **Chapter 7** uses the decomposition of the scene into objects to address the problem of video coding. A second generation video coding is described which permits an extended control over the decoded sequence. The functionalities are both in terms of content and coding quality. Finally, **Chapter 8** gives a summary of the main developments of this dissertation. Possible extensions and new directions of research are also given.

1.4 Main contributions

The main contributions of this work are the following:

- Definition of an algorithm for decomposing the scene into spatio-temporally homogeneous regions. Only two consecutive frames are used. The algorithm is based on a top-down approach which uses spatial, temporal and change information. The effects of disocclusion phenomena and camera motion are corrected for.
- Definition of an unsupervised region merging algorithm for spatio-temporal segmentation. Only two consecutive frames are used. The region merging is carried out in the statistical framework of hypothesis testing and has the following characteristics:
  - the region similarity integrates both temporal and spatial information into a single measure.
  - a novel test statistic is presented to compare the motion of two regions. It integrates both parametric and residual information.
  - the merging strategy is based on a weighted, directed graph representation. Two complementary merging rules are defined which take into account the characteristics of the problem at hand.
- Definition of an unsupervised algorithm for recursive spatio-temporal segmentation that permits to stabilize the spatio-temporal segmentation through time. It finds a compromise between the requirement for stability of the partition and the necessity to allow for possible changes in the current frame.
- Definition of an unsupervised object tracking algorithm to pursue each object of the scene through time. The object tracking interacts with the spatio-temporal segmentation and may modify the current segmentation. Its characteristics are:
  - each object is characterized through a set of distinctive features. These permits the identification of the object.
  - the object identification procedure is performed within the statistical framework of multiple hypothesis testing. A hierarchy of correspondence hypotheses is defined.
- Definition of an object oriented video coding scheme. The scheme provides a high level of functionalities which allow the end-user to interact with the content of the decoded scene. Also, the coding quality of each object can be chosen independently from the other objects in the scene.

Chapter 2

Segmentation based motion estimation

2.1 Introduction

In a video sequence and assuming there is no scene change, two consecutive frames show similarities in terms of the scene they describe. Both frames are composed of the same set of objects. Strong redundancies may thus be found as far as the scene content is concerned. Although similar, the two consecutive frames are nevertheless different. The objects forming the scene may have changed their relative positions. Also, the camera capturing the scene may have moved. Both phenomena entail temporal changes in the sequence. These are denoted under the generic concept of *motion*. Representative of the dynamic processes, the motion is a key information in order to correctly comprehend the scene at hand.

Due to its characteristics, motion information is the basis for many applications. Video coding techniques use it to improve significantly the data compression performance. Motion information is used to reduce the temporal redundancy existing between successive frames. An accurate temporal prediction of the current frame is obtained through motion compensation, the residual error being referred to as Displaced Frame Difference (DFD). The motion information and the DFD are transmitted to the decoder. This is referred to as *predictive coding*. Due to its coding efficiency, this principle is used as the basis of many coding standards such MPEG-1 [74, 89], MPEG-2 [2, 90], H.261 [135] and H.263 [77]. Furthermore, motion information is the cornerstone of a semantic understanding of the scene. Regions which show coherent motion are likely to correspond to the objects forming the scene. The motion information is therefore characterized by both its efficiency in data compression and its semantic meaning. According to Lippman [93], these properties make motion information the ideal criterion on which to base the image representation. Such an image representation would indeed allow for efficient storage or transmission, while being semantically meaningful.

According to Aggarwal and Nandhakumar [9], motion estimation techniques may be split into two different classes: the *feature-based* approach and the *optical flow* approach. In the former, a sparse set of discriminatory features is extracted from the image. Examples of such features are points, corners or lines. The motion is determined by estimating the displacements of these features from one frame to another. Consequently, the feature based approach assumes that the inter-frame correspondence has been already established between the features. This prerequisite represents a major drawback as no general technique for feature correspondence is readily available. The optical flow approach does not require the selection of particular features. The motion estimation relies on the changes of the image brightness. These changes are globally referred to as the *optical flow*. As the optical flow approach to motion estimation is not tributary to the correspondence issue, it has been selected in the framework of this dissertation.

This chapter reviews motion estimation techniques based on the optical flow computation. The estimation is carried out in the context of segmentation based motion estimation [42]. Consequently, we assume that the image is segmented into regions with arbitrary boundaries, the task being to estimate their motion. Section 2.2 reviews the basic concepts of motion. In Sec. 2.3, different motion models to represent the motion of a region are derived. The emphasis is on the fully parametric motion model. Section 2.4 addresses the problem of estimating the motion for fully parametric motion models. Different techniques are reviewed and discussed. The conclusions are drawn in Sec. 2.5.

2.2 The notion of motion

2.2.1 Introduction

Imagine that we observe a video sequence through a fixed pinhole. The pinhole is so small that we only perceive one image element, commonly referred to as a *pixel*. Over time, the pixel intensity is very likely to vary. Such changes derive from three main causes. The first one is called global motion or camera motion. Even though no motion may occur in the scene, the motion of the camera induces a global displacement of the captured scene. The second cause is the intrinsic motions of the objects in the scene. These can be seen as local motions as they do not affect the entire image. However, no distinction is generally made between global and local motions. The global motion is taken into account through local estimates of the motion. The third reason which may cause an intensity change is a variation of illumination. If the lighting conditions change while the sequence is being captured by the camera, the pixel intensities vary. Although algorithms estimating the variation of illumination have been proposed [102, 110, 143], as a general rule the variation of illumination is not taken into account by the motion estimation techniques. In this dissertation, the hypothesis is that there is no change of illumination and that the intensity variations are due only to the global and local motions.

2.2.2 The motion hypotheses

In a digital video sequence, the $4D$ spatio-temporal continuum is projected onto a discrete $3D$ sample grid. Corresponding to the spatio-temporal variation of intensity, the resulting $2D$ flow is referred to as the *optical flow*. Assuming that no changes of illumination occur, the optical flow uniquely arises from the motion existing in the scene. More precisely, the $2D$ *motion field* which is defined by the projection of the $3D$ motion on the $2D$ image plane should be equal to the optical flow. In what follows, no distinction is made between these two notions, and the term motion should be understood as optical flow. Furthermore, the concept of motion always refers to the $2D$ apparent motion arising from the projection of the $3D$ motion.

Many techniques have been proposed for the estimation of the motion field. Due to the difficulties of the task, hypotheses have to be made in order to render the problem workable. Among the many assumptions which may be made, three underlying hypotheses are common to most algorithms. They are namely the hypothesis of *intensity conservation*, the hypothesis of *motion spatial continuity* and the hypothesis *motion temporal continuity*.

The hypothesis of intensity conservation

The estimation of the motion existing in the scene inherently necessitates the comparison of two or more frames. The motion estimation relies on the intensity changes taking place at the pixel level. The relationship between these intensity changes and the motion is made explicit through the hypothesis of *intensity conservation*. This hypothesis states that the intensity stays constant along motion trajectories. Denoting the time interval by Δt, the hypothesis of intensity conservation is mathematically expressed as follows

$$I(\vec{r}, t) = I(\vec{r} - \vec{v}(\vec{r}) \cdot \Delta t, t - \Delta t) \ , \tag{2.1}$$

where $I(\vec{r}, t)$ denotes the image intensity at the pixel location $\vec{r}$ and time t, and $\vec{v}(\vec{r})$ is the pixel motion during the time interval Δt.

Despite its simplicity, Eq. (2.1) is non-linear. A linear approximation may prove useful when searching for analytical solutions of the motion estimation problem. To that end, the first order Taylor series expansion of the right-hand term in Eq. (2.1) is derived. This leads to the *spatio-temporal constraint equation* or the *optical flow constraint equation*

$$\vec{v}(\vec{r}) \cdot \vec{\nabla} I(\vec{r}, t) + \frac{\partial I(\vec{r}, t)}{\partial t} = 0 \,, \tag{2.2}$$

where $\vec{\nabla}$ is the spatial gradient operator. Note that this approximation is only valid for small velocities.

Nevertheless, some situations are not encompassed by the hypothesis of intensity conservation. Broadly speaking, they correspond to the cases when the optical flow and the $2D$ motion flow differ [146]. For instance, it may happen that a moving object is spatially defined through a constant brightness pattern. The optical flow is thus null even though the object is moving. Similarly, a still scene may produce a non-zero optical flow due to illumination changes. Although no motion occurs, the pixel intensities change. Another example where the intensity is not conserved deals with the disocclusion phenomenon. In this case, new pixels appear in the current frame, that have no corresponding pixels in the previous frame.

The hypothesis of motion spatial continuity

Under the hypothesis of intensity conservation, the intensity change due to motion is modeled by Eq. (2.1). However, the constraint imposed by Eq. (2.1) is not sufficient to determine the motion. For instance, let us consider the simplest situation where only one pixel is examined. As its motion is translational, the motion estimation involves the determination of two unknowns. However, only one constraint is available. This ambiguity is known as the *aperture problem* [68, 109]. Therefore additional constraints must be introduced to regularize this ill-posed problem and to allow to solve the optical flow.

The regularization of the motion estimation relies on the hypothesis of *motion spatial continuity*. This hypothesis states that the motion of neighboring pixels is very similar. The hypothesis of spatial continuity may be implemented through an explicit constraint. For instance, Horn and Schunck [68] propose a smoothness constraint, and Lucas and Kanade make the assumption that the motion is uniform and translational in a small region [94]. Such techniques provide a dense motion field as each pixel is assigned a motion vector. The hypothesis of motion spatial continuity may also be made implicitly. This is achieved by grouping adjacent pixels into a region which is assumed to be temporally coherent. This implies that all these pixels move according to the same motion model [11]. The motion $\vec{v}(\vec{r})$ becomes a function of the motion parameters $\vec{M}$, i.e. $\vec{v}(\vec{r})$ is replaced by $\vec{v}(\vec{r}, \vec{M})$ in Eq. (2.1). In other words, the motion of an ensemble of pixels is defined by a limited set of parameters $\vec{M}$. To estimate the motion, an equation reflecting the hypothesis of intensity conservation is formulated for each pixel of the region. The ensemble of these equations provides the constraints which implicitly solve the aperture problem.

The hypothesis of motion spatial continuity is required in order to solve the aperture problem. However, the hypothesis may be put into fault, giving rise to inaccurate and erroneous motion estimations. This typically occurs when crossing the border between two different objects of the scene. In that case, neighboring pixels may have very different motions. The break-down of the hypothesis of motion spatial continuity is encompassed in the *generalized aperture problem* [82, 24]. On one hand, there is the need for large regions so as to constrain and stabilize the motion estimation. On the other hand, larger regions are likely to embrace multiple motions. The solution to this dilemma is closely linked with the goal of the spatio-temporal segmentation techniques.

The hypothesis of motion temporal continuity

As already explained, the $2D$ motion arises from the movements of the camera and the objects of the scene. These motions are expected to undergo smooth changes as time passes. Indeed, the objects composing the scene generally follow predictable trajectories [17], meaning that their position and motion at time $t - \Delta t$ give a good hint of their position and motion at time t. At the pixel level, such smooth changes of the object motions are described by the hypothesis of *motion temporal continuity*. It states that the motion of each pixel varies in a continuous manner over time [23]. This implies that the temporal evolution of the motion may be modeled through a kinematic model [17]. In turn, this permits the modeling of the trajectories of the pixels and, hence, those of the objects to which the pixels belong. The hypothesis of motion temporal continuity finds an important application in techniques for analyzing multiple consecutive frames. It indeed justifies the integration of the information present in many frames in order to carry out a robust analysis of the scene. When segmenting the scene in terms of moving objects, the notion of predictable trajectories permits to foresee a tracking procedure. Each object undergoes a pursuit through time, enabling a thorough semantic understanding of the scene.

The motion temporal continuity is however not always verified. Brisk pixel motions may occur between two consecutive frames. This phenomenon is referred to as a *maneuver*. A sudden external force completely changes the motion of the pixel. It thus does not follow a smooth evolution and no kinematic model of the pixel evolution may be derived. The resulting trajectory invalidates the hypothesis of motion temporal continuity. Further, the hypothesis is not valid whenever disocclusion phenomena occur. Pixels are indeed unable to find any corresponding pixels in the previous frames.

2.2.3 Global vs local motion

The $2D$ motion flow has been described as arising from the ensemble of motions existing in the scene. Two types of motion may occur, being the global and the local motion. When estimating the motion, one has therefore to deal with a composite motion. These two motions are very different in nature. The global motion is in essence artificial as it arises from the displacement of the camera with respect to the scene. This phenomenon induces a motion for each pixel composing the scene, even when no object is moving. Conversely, the local motions arise from the displacements of the objects composing the scene. Depending on the object to which the pixel belongs, the corresponding local motion differs.

The distinction between global and local motions is usually made within a two-stage global/local motion estimation process [6, 103]. Consider two consecutive frames, at times $t - \Delta t$ and t, respectively. The two-stage procedure starts by estimating, for every pixel, the global motion $\vec{v}_G(\vec{r})$. The effect of the global motion is then removed from the scene by global motion compensation as follows

$$I_G(\vec{r}, t) = I(\vec{r} - \vec{v}_G(\vec{r}) \cdot \Delta t, t - \Delta t) \ , \tag{2.3}$$

where $I_G(\vec{r}, t)$ is the image intensity at the pixel location $\vec{r}$ and time t, after the global motion has been removed.

In the second phase, the local motions are estimated. This estimation is performed between the pixels, $I_G(\vec{r}, t)$, of the globally motion compensated frame and the pixels, $I(\vec{r}, t)$, of the frame at time t. The effect of the global motion having been removed, the only remaining motions are the local ones. Therefore, this results in a more precise subsequent local motion estimation.

In many applications, the distinction between global and local motions is of importance. For instance, video coding techniques separating global from local motions have shown improved performances [144, 6,

103]. The amount of side information required for the motion vectors is reduced. This results in a better compression ratio. Furthermore, a distinction between background and foreground may also be made based on the global and local motions. In that respect, the coding strategy may be optimized so as to increase the bit allocation into important areas of the scene (i.e. the foreground). Finally, the distinction between global and local motions allows to build a background memory [43]. This permits to tackle the problem of uncovered background areas, leading to improved coding performances.

As far as dynamic scene analysis is concerned, the distinction between global and local motion helps at deriving a semantic understanding of the scene. The global motion compensation allows to accentuate the importance of the local motions. This may be used to detect the moving objects present in the scene [32, 72, 105]. The global motion estimation may also be used to obtain a mosaic representation [71, 130]. Also referred to as *salient still*, these techniques align the images in the sequence by canceling the contribution of camera motion. The mosaic is built by temporal integration of the aligned images. In this way, the mosaic encapsulates the information existing in multiple frames of a video sequence.

2.3 Motion models

2.3.1 Introduction

Two critical choices have to be made before estimating the motion of a region. The first choice involves the type of motion model. The model reflects the characteristics that the region motion is expected to show. The second choice is related to the procedure to estimate the parameters of the motion model. In this section, we will present the different alternatives as far as the choice of the motion model is concerned. In the remaining of the section, different motion models are first derived by the projection of the $3D$ motion in the scene onto the $2D$ image plane. Then, the issue of choosing the appropriate motion model for the problem at hand is dealt with.

2.3.2 Rigid body motion

The relation defining the $2D$ motion in the image plane induced by the $3D$ motion in the scene is derived as follows [42]. If we assume a pinhole camera, an image is formed by a perspective projection of the real world scene onto the focal plane [67]. Let $\vec{R} = (X, Y, Z)^T$ be the camera-centered Cartesian coordinates system, and $\vec{r} = (x, y)^T$ be the coordinates in the image plane (i.e. Z is perpendicular to the image plane and defines the optical axis), as illustrated in Fig. 2.1. Under perspective projection, $\vec{r}$ is related to $\vec{R}$ by

$$x = \frac{X}{Z} \text{ and } y = \frac{Y}{Z} , \tag{2.4}$$

where the focal length has been normalized to 1 without any loss of generality.

For simplicity, the scene is usually supposed to be composed of rigid bodies. This hypothesis is motivated by the difficulty to cope with a non-rigid body motion, although algorithms to estimate non-rigid body motion have been proposed [120]. Furthermore, it is justified by the fact that a rigid body motion closely approximates a non-rigid body motion when the temporal sampling frequency is sufficiently high and the motion between two frames is sufficiently small. This assumption is therefore made throughout the dissertation.

With regard to the $3D$ rigid body motion, two formulations are possible, one in terms of instantaneous velocities and the other in terms of displacements. In the first case, the motion of a $3D$ rigid body is

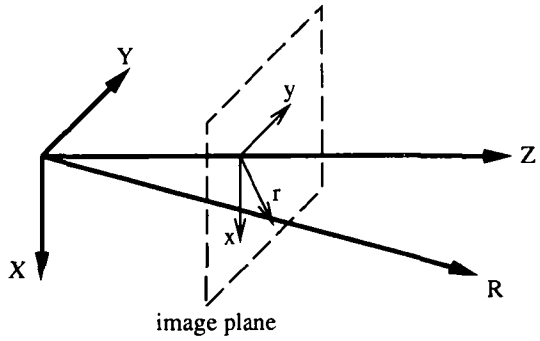

Figure 2.1: $(X, Y, Z)^T$ are the Cartesian coordinates fixed with the camera and $(x, y)^T$ are the coordinates in the image plane.

expressed as [67]

$$\dot{\vec{R}} = \vec{\omega} \times \vec{R} + \vec{T} \ , \tag{2.5}$$

where $\dot{\vec{R}}$ represents the temporal derivative, $\vec{\omega} = (\omega_x, \omega_y, \omega_z)^T$ is the angular velocity and $\vec{T} = (T_x, T_y, T_z)^T$ is the translational motion. Therefore, in this formulation $\vec{\omega}$ and $\vec{T}$ correspond to instantaneous velocities. Introducing Eq. (2.4) in Eq. (2.5) leads to

$$\dot{\vec{r}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\omega_x \ xy + \omega_y(x^2+1) - \omega_z \ y + \frac{T_x - T_z x}{Z} \\ -\omega_x(y^2+1) + \omega_y \ xy - \omega_z \ x + \frac{T_y - T_z y}{Z} \end{pmatrix} . \tag{2.6}$$

In the formulation in terms of displacements, the $3D$ rigid body motion equation is written as

$$\vec{R}' = \Omega \vec{R} + \vec{D} \ , \tag{2.7}$$

where $\vec{R}$ and $\vec{R}'$ are the positions at time t and t' respectively, Ω denotes the rotation matrix and $\vec{D} = (D_x, D_y, D_z)^T$ is the translational displacement. Therefore, Ω and $\vec{D}$ denote differences in orientation and position over a time interval. Similarly, by introducing Eq. (2.4) in Eq. (2.7), it follows that

$$\vec{r}' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{x - \Omega_z y + \Omega_y + D_x/Z}{1 - \Omega_y \ x + \Omega_x \ y + D_z/Z} \\ \frac{\Omega_z x + y - \Omega_x + D_y/Z}{1 - \Omega_y \ x + \Omega_x \ y + D_z/Z} \end{pmatrix} , \tag{2.8}$$

where the rotation matrix Ω has been approximated, assuming small rotation angles, by

$$\Omega = \begin{pmatrix} 1 & -\Omega_z & \Omega_y \\ \Omega_z & 1 & -\Omega_x \\ -\Omega_y & \Omega_x & 1 \end{pmatrix} . \tag{2.9}$$

Equations (2.6) and (2.8) define the $2D$ motion in the image plane induced by the $3D$ motion for the formulation in terms of instantaneous velocities and displacements, respectively. Under some realistic assumptions, it can be shown that both formulations in terms of instantaneous velocities and displacements are equivalent [4, 42].

2.3.3 Planar surface motion

The dependency on the depth map, Z, in Eqs. (2.6) and (2.8) causes two problems. First it is difficult to accurately estimate the depth map. Second, in the context of coding, the depth map represents a large

amount of overhead information to be transmitted to the decoder. In order to overcome these problems, the model of the scene is restricted to a patchwork of planar surfaces which closely approximate $3D$ rigid bodies.

A planar surface is defined as

$$k_x X + k_y Y + k_z Z = 1 \ , \tag{2.10}$$

or equivalently as

$$k_x x + k_y y + k_z = \frac{1}{Z} \ . \tag{2.11}$$

For the formulation in terms of instantaneous velocities, introducing Eq. (2.11) in Eq. (2.6) leads to

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \\ a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \end{pmatrix} \ , \tag{2.12}$$

where the coefficients $a_1, ..., a_8$ are given in Appendix B. The 8 parameter model defined by Eq. (2.12) specifies the instantaneous velocity in the image plane due to a moving planar surface under perspective projection.

Similarly, for the formulation in terms of displacements, the introduction of Eq. (2.11) in Eq. (2.8) gives

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{a_1 + a_2 x + a_3 y}{a_7 x + a_8 y + a_9} \\ \frac{a_4 + a_5 x + a_6 y}{a_7 x + a_8 y + a_9} \end{pmatrix} \ , \tag{2.13}$$

where the coefficients $a_1, ..., a_9$ are given in Appendix B. Equation (2.13) is referred to as the perspective model. It defines the displacement in the image plane due to a moving planar surface under perspective projection. Usually, the parameters are normalized by a_9, leading to an 8 parameter model.

When the distance to the scene is much larger than the variation in distance among objects in the scene, perspective projection is closely approximated by orthographic projection [67]. In other words, this simplification is valid when the depth map of the scene is small relative to the distance from the camera. Rather than modeling rays of light passing through the origin (pinhole camera) as in the perspective projection, the orthographic projection supposes that the rays of light are parallel to the optical axis.

Under an orthographic projection, the second order terms in Eq. (2.12) can be neglected as $x = X/Z \ll 1$ and $y = Y/Z \ll 1$. This simplification leads to the *affine model*

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix} \ . \tag{2.14}$$

The model defined by Eq. (2.14) allows for the representation of the motion of a planar surface under orthographic projection. For the formulation in terms of displacements, under the same hypothesis of orthographic projection, a similar affine model is obtained from a first order Taylor series expansion of Eq. (2.13). It is useful to note that the affine model can equivalently be expressed in terms of a translation vector, two scaling factors, and two rotation angles [152, 56].

The affine model may be simplified further. For instance, the simple and widely used translational model with 2 parameters results from setting $a_2 = a_3 = a_5 = a_6 = 0$, and

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_4 \end{pmatrix} \ . \tag{2.15}$$

2.3.4 Choice of an appropriate model

According to Anandan *et al.* [11], the motion models may be classified in three categories: non-parametric, quasi-parametric and fully parametric. The non-parametric models rely on a dense local motion field obtained through techniques such as [68, 94]. Each pixel is assigned a $2D$ displacement vector. The parametric models represent the motion of a large region by a single set of parameters, referred to as motion parameters. For instance, Eqs. (2.12) - (2.15) correspond to fully parametric models. A further distinction is made between quasi-parametric models which represent the motion by a set of parameters as well as a local field (e.g. the depth map or local optical flow) and fully parametric models which completely specifies the motion by a set of parameters. Examples of quasi-parametric models are given by Eqs. (2.6) and (2.8). In the following, the terms non-parametric, quasi-parametric and fully parametric motion estimation techniques should be understood as techniques which rely on a non-parametric, a quasi-parametric and a fully parametric motion model, respectively.

In the framework of segmentation based motion estimation, different types of motion models may be used. Although it implicitly requires the region to be planar, the fully parametric model is the most natural choice. Indeed, such a model is the best compromise between modeling efficiency and model complexity. Furthermore, when estimating the motion of a region, one implicitly makes the assumption that its pixels show coherent motion. Typically, the region represents an object or part of it. The temporal coherence between the pixels is perfectly reflected by the fully parametric motion model. The region pixels have their respective motions modeled by a common set of parameters. Hence, these motion parameters are estimated instead of the motion field itself. As all the pixels composing the object contribute to the motion estimation, robustness and high accuracy can be expected. The fully parametric model has the additional advantage of characterizing the motion of the region completely. The pixel motion is no longer described by itself, but is encompassed in the context of the region motion. This region oriented motion description is in sharp contrast with non-parametric and quasi-parametric models which remain at the pixel level. This has important implications when the motions of different regions are compared. The fully parametric model permits to address the problem in terms of regions and no longer in terms of pixels. Furthermore, the fully parametric model provides a compact representation of the motion. This is a valuable asset when addressing video coding applications. For instance, the current video coding standards, MPEG-1 [74, 89], MPEG-2 [2, 90], H.261 [135] and H.263 [77], partition the image into small blocks and rely on a translational fully parametric motion model to represent their motions.

With regard to the choice the type of fully parametric model (e.g. Eqs. (2.12) - (2.15)), the choice depends on the type of region. A simple translational model is sufficient for current video standards as the regions are small blocks. However, the region may also be a significant part of an object or an object itself. This type of regions usually correspond to a large area of the image. In this context, a simple translational model appears to be inadequate and more complex fully parametric models are needed (e.g. perspective models Eqs. (2.12) and (2.13), or affine model Eq. (2.14)). In particular, the affine model shows a good compromise between complexity and modeling capabilities. It has been adopted in [26, 148, 44, 117].

2.4 Segmentation based fully parametric motion estimation

2.4.1 Introduction

In this section, the problem of motion estimation for a given region is addressed [42]. We assume that the image is segmented into regions with arbitrary boundaries and that, for each region, a fully parametric

motion model has been chosen. In order to estimate the parameters of the motion model, two different approaches have been investigated: the *indirect* and the *direct* approaches. As already underlined in Sec. 2.2.2, the concept of motion should be understood as the apparent $2D$ motion.

The indirect techniques are composed of two steps: the computation of a dense optical flow using a non-parametric technique, followed by the modeling of the motion vectors by a set of motion parameters [4, 148, 117, 144]. As the motion parameters are not computed from the luminance signal itself, this approach is referred to as an indirect or regression technique. Its drawback is that its performance largely depends on the efficiency of the non-parametric motion estimation technique.

The second type of techniques directly estimates the parameters of the motion model as in [11, 66, 132, 44, 152, 103]. The motion parameters are obtained by minimizing an error norm computed from the luminance signal. As the estimation is performed on the luminance signal itself, these techniques are said to be *direct*. In this framework, differential techniques, which are based on a Taylor series expansion of the luminance signal, are the most widely used [11, 66, 132, 152]. An alternative approach are the matching techniques [44, 103]. In contrast to the differential techniques, the matching techniques do not rely on a model of the luminance. Therefore, they are characterized by their robustness and their resilience to noise.

In the parametric motion estimation techniques above, the estimation procedure may be spoilt through outliers. These outliers are due to noise, or a badly defined support of the motion estimation which does not correspond to an area characterized by a coherent motion. To overcome these problems, robust estimators, less sensitive to outliers, are used [124, 98].

2.4.2 Indirect parametric motion estimation

The indirect parametric motion estimation techniques [4, 148, 117, 144] estimate the motion parameters of a parametric model in two steps. A dense optical flow is first estimated using a non-parametric technique. In [4, 148], this initial optical flow is estimated by a differential technique, whereas [117] uses a block matching algorithm (i.e. in fact a trivial parametric technique). The method described in [144] is similar to the one in [117], except that it is applied to estimate the camera motion. The motion vectors obtained are then modeled by a set of motion parameters, using a Least Mean Square (LMS) technique [4, 148, 117, 144]. These methods can be seen as indirect, as they compute the motion parameters from a dense motion field rather than from the luminance signal itself. Their drawback is that they depend greatly on the accuracy of the non-parametric motion estimation technique. Errors in the initial optical flow may indeed lead to incorrect motion parameters. The initial optical flow estimation and the modeling of the motion vectors by motion parameters are now described in greater detail.

Non-parametric optical flow estimation

In the first step, a dense optical flow field is estimated, usually via non-parametric differential techniques [68, 94]. These techniques are based on Eq. (2.2) and require explicit constraints.

A smoothness constraint has been introduced by Horn and Schunck [68] for this very purpose. With this constraint, the optical flow is obtained through the minimization of the following expression

$$\iint \left[\left(\vec{v} \cdot \vec{\nabla} I + \frac{\partial I}{\partial t} \right)^2 + \alpha^2 \left(\left(\frac{\partial v_x}{\partial x} \right)^2 + \left(\frac{\partial v_x}{\partial y} \right)^2 + \left(\frac{\partial v_y}{\partial x} \right)^2 + \left(\frac{\partial v_y}{\partial y} \right)^2 \right) \right] dxdy \ , \tag{2.16}$$

where α^2 is a weighting factor. This minimization problem is solved by variational calculus and an iterative Gauss-Seidel procedure [68].

The assumption that motion is uniform and translational in a small region is made by Lucas and Kanade [94]. In this case, the optical flow is computed by a weighted LMS minimizing

$$\sum_{\vec{r}\in R} W^2(\vec{r}) \left[\vec{v}\cdot\vec{\nabla}I + \frac{\partial I}{\partial t}\right]^2 , \tag{2.17}$$

where W denotes a window function and R the local neighborhood.

The differential optical flow techniques proposed above suffer from three major drawbacks. First, the smoothness constraint or the local uniformity constraint leads to a poor motion estimation at the boundaries of the moving objects. In order to improve motion estimation in these areas, an oriented smoothness constraint has been introduced in [108]. Second, the estimation of the gradient is subject to errors, and in particular is very sensitive to noise. Third, the modeling of the luminance by a first order Taylor series expansion relies on the hypothesis of small velocities. However, such a hypothesis is not always valid. To overcome this problem, hierarchical schemes have been proposed in [61, 52].

Instead of the differential optical flow techniques, a block matching technique is used in [117, 144]. As the block matching technique does not rely on a model of the luminance signal, it is characterized by its robustness when compared to differential techniques [41]. However, the block-based nature of the technique may result in block artifacts, especially along moving edges. In order to overcome this drawback, only those blocks included within the predefined region of interest are used for block matching in [117].

Modeling of the motion vectors by motion parameters

Based on the above estimated optical flow and assuming a predefined segmented region, the motion parameters of this region can be computed. For this purpose, an LMS technique is used in [4, 148, 117, 144].

Using LMS, the motion parameters $\vec{M}$ over the region R are chosen so as to fit the optical flow optimally in the LMS sense. Let $v_x(\vec{r})$ and $v_y(\vec{r})$ be the x and y components of the optical flow, and $\hat{v}_x(\vec{r},\vec{M})$ and $\hat{v}_y(\vec{r},\vec{M})$ the fully parameterized optical flow. The motion parameters are estimated by

$$\min_{\vec{M}} \sum_{\vec{r}\in R} \left[v_x(\vec{r}) - \hat{v}_x(\vec{r},\vec{M}))\right]^2 , \tag{2.18}$$

$$\min_{\vec{M}} \sum_{\vec{r}\in R} \left[v_y(\vec{r}) - \hat{v}_y(\vec{r},\vec{M}))\right]^2 . \tag{2.19}$$

By setting to zero the partial derivatives with respect to M_i, $i = 1,...,n$ (where n is the number of parameters of the model), the motion parameters are straightforwardly computed.

The above LMS modeling relies only on the optical flow to estimate the motion parameters of a region. In particular, the LMS estimation is very sensitive to incorrect input motion vectors. For instance, errors in the computation of the optical flow at the early stage lead to incorrect motion parameters.

2.4.3 Direct parametric motion estimation

The techniques described in [11, 66, 132, 44, 152, 103] directly estimate the parameters of the motion model. The parametric motion model implicitly constrains the computation of the motion field, and no additional explicit constraint is therefore required. In contrast to the indirect techniques (see Sec. 2.4.2), the motion parameters are estimated from the luminance signal itself. Consequently, these techniques can be seen as direct.

Various techniques have been proposed for direct parametric motion estimation. They can be divided into two main groups: differential techniques and matching techniques. The differential techniques are the most widely used [11, 66, 132, 152]. They are based on the optical flow constraint equation and therefore utilize a Taylor series expansion of the luminance signal. Matching techniques are proposed in [44, 103]. The motion parameters are computed by minimizing a disparity measure. As the matching techniques do not rely on a model of the luminance, they are characterized by their robustness. The differential and matching motion estimation techniques are now described in greater detail.

Differential techniques

The differential techniques [11, 66, 132, 152] solve the optical flow constraint equation (Eq. (2.2)). They assume a Taylor series expansion of the luminance signal. However, in contrast to the non-parametric motion estimation which is underconstrained, the parametric motion estimation is implicitly constrained by the motion model. Hence, additional constraints such as smoothness or local uniformity becomes redundant.

Assuming a first order Taylor series expansion, the problem is to solve the optical flow constraint equation, where $\vec{v}(\vec{r})$ is replaced by the optical flow $\vec{v}(\vec{r}, \vec{M})$ parameterized by $\vec{M}$. The parameters $\vec{M}$ of the model over the region R are obtained by minimizing [11]

$$\sum_{\vec{r} \in R} \left[\vec{v}(\vec{r}, \vec{M}) \cdot \vec{\nabla} I(\vec{r}, t) + \frac{\partial I(\vec{r}, t)}{\partial t} \right]^2 . \tag{2.20}$$

The motion parameters can be straightforwardly computed by setting to zero the partial derivatives with respect to M_i, $i = 1, ..., n$ (where n is the number of parameters of the model).

The techniques in [66, 152] are based on a second order Taylor series approximation instead of the above first order series expansion.

These differential techniques have drawbacks similar to the ones of the differential non-parametric optical flow techniques (Sec. 2.4.2). Namely, the estimation of the gradient is very sensitive to noise, and the modeling of the luminance by a first order Taylor series expansion implicitly assumes small velocities. To reduce the effect of these drawbacks, the algorithms in [11, 66, 132, 152] are based on hierarchical schemes. The solution is iteratively refined on a multiresolution data representation. The resulting iterations are equivalent to refining the parameter estimates by means of a Gauss-Newton minimization algorithm.

Matching techniques

Direct parametric motion estimation by means of a matching technique has been proposed in [44, 103]. This algorithms are a generalization of the classical block matching technique [81, 85, 107, 41] widely

used in video coding. The motion parameters are computed by directly minimizing the disparity existing in Eq. (2.1).

Using the same notation as before, we obtain the motion parameters, $\vec{M}$, by minimizing a disparity measure between the region R in the current frame and the mapped region in the previous frame

$$\min_{\vec{M}} \sum_{\vec{r} \in R} \| I(\vec{r}, t) - I(\vec{r} - \vec{v}(\vec{r}, \vec{M}) \cdot \Delta t, t - \Delta t) \| , \tag{2.21}$$

where the most commonly used distance measures are the L_2 quadratic norm $\| x \| = x^2$ and the L_1 absolute value $\| x \| = | x |$. The expression $\vec{r} - \vec{v}(\vec{r}, \vec{M}) \cdot \Delta t$ specifies the location of the pixel to be compared with the pixel currently positioned in $\vec{r}$.

In order to compute the motion parameters, a search is carried out in the n-dimensional parameter space (where n is the number of parameters of the model). For this purpose, the parameter space is discretized, and upper and lower bounds are set for each parameter. The absolute minimum in Eq. (2.21) is reached with certainty only if an exhaustive search of the n-dimensional discretized parameters space is performed. However, depending on the number of parameters, this may require a too high computational complexity. To reduce the computational complexity, a fast non-exhaustive search can be carried out. In [44, 103], a generalization of the 3-step search [85] is used. Furthermore, the algorithm is carried out on a multiresolution structure based on a Gaussian pyramid [31]. The final motion parameters at one level propagate as initial estimates on the next level. A deterministic relaxation scheme is applied during the propagation stage. It compares the motion parameters obtained for neighboring regions and selects the one corresponding to the lowest prediction error. This multiresolution and relaxation scheme allows for the reduction of the computational load, as well as the prevention of local minima due to the non-exhaustive search.

2.4.4 Robust estimation

In the above parametric motion estimation, the estimation process is spoiled in the presence of samples whose value is far from the prevailing tendency. Those samples are referred to as outliers. They may be the result of impulse noise, a clutter environment, or a badly defined support of the motion estimation which does not correspond to an area characterized by a coherent motion. To overcome these problems, robust estimators can be used [124, 98]. They are less sensitive to outliers and permit to robustly estimate the motion of the region.

A robust estimator should possess the following qualities [12, 98]. It has to be reliable in the presence of various type of noise as well as be unaffected by the outliers present in the estimation support. The robust estimator should also have a high breakdown point [98]. The breakdown point is defined as the smallest fraction of outliers which results in a biased estimation. It indicates therefore the robustness of the estimator. Moreover, the computational complexity should stay within reasonable limits, a fair guess being $O(N^2)$ where N is the number of data.

Estimators can be grouped in three categories: L-estimators, R-estimators and M-estimators [98]. L-estimators are linear combinations of order statistics. Two examples are the α-trimmed-mean and the median. As the median remains reliable when half of the data samples at the most are contaminated, it yields the maximum breakdown point of 0.5. R-estimators are based on rank tests. Finally, M-estimators minimize $\sum_i \rho(r_i)$ where r_i are the residuals (i.e. the differences between the data and the modeled value in Eq. (2.1)), and ρ is a symmetric positive-definite function with a unique minimum at $r_i = 0$. Least-squares method is an example of such M-estimators where $\rho(r_i) = r_i^2$. Several functions ρ have

been proposed to reduce the effect of large residuals, i.e. outliers. One typical example is the following robust estimator, also known as Geman and McLure estimator,

$$\rho(r_i) = \frac{\frac{r_i^2}{s^2}}{1 + \frac{r_i^2}{s^2}} , \tag{2.22}$$

where $s = 1.4826 \text{ median}\{| r_i |\}$. The breakdown point of this estimator is less than $1/(n+1)$ where n is the number of parameters to be estimated.

2.5 Conclusions

Motion information plays a key role when analyzing or compressing a video sequence. In this chapter, the notion of motion and the related hypotheses are reviewed. Then, the most popular models for embodying the motion information are derived under the constraint of rigid body motion. The different techniques for estimating the model parameters are discussed for the case of a fully parametric motion model. The assumption is that the delimitation of the region of support for the estimation is known.

Chapter 3

Spatio-temporal segmentation - State of the art

3.1 Introduction

The previous chapter reviewed techniques for estimating the motion of a region assuming its boundaries known. In this chapter, the problem of simultaneously estimating the motion and defining its support is considered. This issue is referred to as spatio-temporal segmentation whose main concepts are presented in Sec 3.2. Such a segmentation decomposes the scene at hand in terms of objects. Both temporal and spatial information may be used. However, the emphasis should be put on temporal information as an object is primarily defined as an ensemble of pixels having a coherent motion. Fundamentally, the spatio-temporal segmentation remains an image processing technique as no explicit analysis of the scene is performed [78]. However, it serves as the basis for a wide range of applications in the domains of dynamic scene analysis and video coding.

In Sec. 3.3, the challenges faced by the spatio-temporal segmentation techniques are reviewed. First, the spatio-temporal segmentation concept is shown to be a "chicken & egg" problem. The motion of the object indeed derives from the object segmentation, while, reciprocally, the segmentation depends on the motion of the object. Another challenge faced by spatio-temporal segmentation techniques relates to the coherence of the segmentation through time. Assuming temporal continuity in the video sequence, the segmentations of successive frames should be coherent with one another. In particular, the objects forming the scene should robustly be segmented and retrieved at successive times, this enabling a tracking procedure to take place.

Further, a classification of the algorithmic approaches for spatio-temporal segmentation is presented. The analysis is focused on the methods which provide a complete image partition, without requiring any *a priori* knowledge nor user's input. The dichotomy is made between respectively top-down methods which only use two consecutive frames, bottom-up methods which only use two consecutive frames and methods using many frames. The top-down methods are described in Sec. 3.4. They sequentially extract the objects present in the scene on the basis of the outlier detection/rejection paradigm. Section 3.5 deals with bottom-up techniques. Starting from a set of initial regions, bottom-up techniques merge the regions in order to define the objects. The merging is based on the spatio-temporal similarity existing between the regions. The techniques which use many frames are presented in Sec. 3.6. These techniques aim at tackling the problem of the stability of the successive spatio-temporal partitions through time. They are intimately related to the concept of tracking. Section 3.7 presents the conclusions.

3.2 The spatio-temporal segmentation

3.2.1 Introduction

In the framework of computer vision, the spatio-temporal segmentation procedure plays a central role. It aims at identifying in the input image, features or components that are relevant for the problem at hand. In particular, the emphasis of this dissertation is on the complete decomposition of the scene. In other words, the segmentation must yield a complete partition of the scene into its constituent components. These are referred to as objects. The resulting image representation may be seen as an approximation of the one inferred by the human visual system. The image representation derived from the spatio-temporal segmentation finds applications in various domains. For instance, it is very well suited for dynamic scene analysis as it implicitly involves a semantic understanding of the scene. Furthermore, it is very appealing in the context of second generation video coding.

3.2.2 The notion of spatio-temporal segmentation

The notion of spatio-temporal segmentation has several interpretations. A first interpretation is closely linked to the notion of target tracking [17]. The image is analyzed so as to determine whether a given feature of interest is present in the scene. The feature may be a certain object, particular edges, corners or lines. In the case where the feature is detected, it is extracted for further analysis. For instance, military applications aim at extracting from video sequences typical objects such as airplanes, roads or bunkers. In the medical domain, the goal is to segment a particular organ in a video sequence so as to help the physician in her/his diagnostic. In particular, the segmentation of the heart is the subject of intensive research. Other examples of features of interest include cars in case of traffic surveillance and human beings for intruder detection.

In the dissertation, we define the spatio-temporal segmentation as a means of deriving a full partition of the scene [38, 26, 4, 119]. The scene is segmented into an ensemble of regions whose union exactly forms the scene. The value of the segmentation directly depends on the characteristics of the regions. In particular, the regions should carry a semantic meaning [25]. According to the Gestalt "law of common fate", such meaningfulness is attained if the regions are defined on the basis of temporal coherence. Indeed, this Gestalt law states that if a collection of pixels move in unison, one tends to see them as a single figure. Consequently, the resulting regions may be identified to the moving objects composing the scene [44]. Such a segmentation provides an alternative to the waveform representation of the visual information. In contrast to the latter representation which directly derives from the image capture procedure, the segmentation in terms of objects describes the content of the scene. Not only it is independent of the capture procedure, but it is also semantically meaningful.

It has to be stressed that the spatio-temporal segmentation is not, in itself, an image analysis procedure [78, 79]. Although the segmented objects may be semantically meaningful, the spatio-temporal segmentation does not provide any analysis nor characterization of them. As such, the spatio-temporal segmentation procedure has to be seen as an image processing technique. Nevertheless, the information it provides is essential when one tries to comprehend the scene at hand. The spatio-temporal segmentation forms the backbone of schemes aiming at recognizing and classifying objects, or tracking them. Tracking aims at following the objects throughout the video sequence, defining a trajectory for each of them. Object tracking is of particular interest in this dissertation due to its close link to the problem of the spatio-temporal segmentation stability.

When a spatio-temporal segmentation of the scene is desired, one should decide what will the underlying quality of the objects be. In other words, a criterion of coherence for assigning pixels to a particular objects has to be chosen. As already pointed out, the temporal information should form the backbone of the criterion. However, diverging coherence criteria have been proposed in the literature. A first type of coherence criteria only uses spatial information [79, 151, 128]. The objects are formed through texture properties or edge information. A second type of coherence criteria is based on temporal information [112, 4, 119]. Pixels sharing a common motion are grouped into a single object. Finally, both spatial and temporal information may be used [42, 139, 38]. Depending on the coherence criterion which is chosen, the obtained spatio-temporal partition may vary greatly. However, it is important to notice that the coherence criterion has to be adapted to the application at hand [95]. The visually relevant objects composing the scene may be represented by different partitions depending on the application constraints. Nevertheless and as a general comment, the mere fact of defining objects in the context of a video sequence advocates for the use of temporal information. Only this last information may ensure that the defined objects are stable throughout the sequence.

Finally, a distinction has to be made between spatio-temporal methods and change detection methods.

While spatio-temporal segmentation algorithms aim at finding the set of objects defining the scene, change detection algorithms aim at localizing the parts of the scene which have undergone a temporal intensity change [3, 150, 27, 69]. Algorithms in the latter category typically involve no motion estimation and only deal with the grey value difference image between two successive frames. The lack of motion information makes the distinction between different moving objects impossible. Change detection methods may however be useful as a pre-processing tool towards the derivation of a spatio-temporal segmentation.

3.2.3 Applications of the spatio-temporal segmentation

Dynamic scene analysis

The spatio-temporal segmentation finds a wide a range of applications. Among these, the domain of dynamic scene analysis may cited as a whole [70, 145, 134, 38, 116, 26]. Typical examples are scene understanding and robot vision. In the former, the spatio-temporal segmentation is used to comprehend the scene at hand. For instance, scene understanding may address the issue of tracking objects over time. The objects undergo a pursuit procedure that identifies them in successive frames. This permits to resolve practical problems such as the "time-to-collision" problem [100]. It is defined as the time that will elapse before a given object and the image sensor are in contact. As far as robot vision is concerned, the aim is to enable a robot to navigate in an unknown environment. The segmentation based representation makes it possible to teach the robot how to react and act in the real world [55]. Another domain of application is the entertainment world. Mosaicking techniques may be applied to obtain a time-integrated view of the scene background [43]. This enables the user to play with the image content and change it as required.

Video coding

Nowadays, most video compression techniques belong to the first generation [57, 41]. They are based on the waveform representation of the scene, commonly referred to as the pixel representation. Well known examples relying on this representation are the video coding schemes based on the Discrete Cosine Transform (DCT) such as the recent standards MPEG-1 [74, 89], MPEG-2 [2, 90], H.261 [135] and H.263 [77]. Although the above first generation coding techniques have reached good performances, they suffer from a severe drawback: the image is represented in terms of pixels or blocks of pixels which do not carry any semantic meaning. Consequently, visually annoying artifacts occur at high compression ratios, such as block artifacts in DCT-based schemes. No major improvement in terms of compression performances can be foreseen in this direction. Furthermore, the lack of information about the content of the scene limits the functionalities available to the end-user at the decoder side. For instance, desired features such as the ability to manipulate and personalize the received visual information are not available.

In contrast to the coding techniques of the first generation, the techniques of the second generation introduced by Kunt *et al.* [87, 86] rely on a semantic representation of the scene. Each frame is coded by decomposing it into psycho-visually meaningful primitives. The primitives are transmitted and permit to synthesize the frame at the decoder side. By taking into account the semantic meaning of the scene, second generation video coding techniques stay in sharp contrast to first generation techniques where the aim is a mere reduction of the spatial and temporal redundancy existing in the whole frame.

A spatio-temporal partition is perfectly suited to the requirements of second generation video coding. The partition is both semantically meaningful and provides with a complete decomposition of the scene. The resulting image representation is psycho-visually meaningful, while allowing for very high compression ratios and visually less annoying artifacts [148, 128, 106, 59, 46]. Typically, for each object of the

partition, the motion, the shape and the texture are transmitted to the decoder. Furthermore, the objects being semantically meaningful, new functionalities are possible at the decoder side [76]. The end-user can interact with the decoded sequence so as to customize it. Objects particularly interesting to the viewer may be coded with more accuracy than the rest of the scene content. Furthermore, a dynamic control over the content of the decoded image is possible. These characteristics make the segmentation based representation an ideal candidate to satisfy the requirements of the second generation video coding techniques promoted in the MPEG-4 framework [76].

3.3 The challenges of spatio-temporal segmentation

When speaking about spatio-temporal segmentation, the semantic meaningfulness of the partition has to be assessed [60]. This is usually done by comparing the partition with what is perceived by our eyes. According to Wilson and Spann [151], the human visual system has prodigious performances when segmenting a scene. The segmentation is mediated by processes which are largely subconscious [96]. Despite ambiguous information, the visual system constantly infers the properties of the objects we perceive [147]. Essential to the vision, this inference procedure is however hidden from our conscious awareness. Segmentation algorithms are thus confronted with a very hard task when put into comparison with the visual system. Nevertheless, this comparison is required when the task at hand is to derive a semantically meaningful segmentation.

By its very nature, the definition of the objects forming the scene is a "chicken & egg" problem [21]. There is indeed a very strong interaction between the estimation of the moving objects partition and their motions. On one hand, as the motion estimation depends on the region of support, a good segmentation is needed in order to precisely estimate the motion. On the other hand, as the moving objects are defined by a coherent motion, an accurate estimate of the motion is required to obtain a good segmentation.

Furthermore, the object definition may not only involve the motion information, but also spatial characteristics. Despite their very different natures, both types of information are indeed needed for a precise definition of the objects. In particular, the spatial information provides important hints about object boundaries. However, the best strategy to combine both sources of information remains an open issue. In that respect, Wandell [147] states that the neuron's sensitivity jointly depends on the space and time. This non-separability may be an helpful guideline for the derivation an efficient procedure to extract the objects. Nevertheless, one should not forget that an object is primarily defined through its coherent motion. To put it in other terms, the emphasis should remain on the temporal information.

Another challenge facing spatio-temporal segmentation methods lies in the coherence of the partition through time. The aim being to extract the objects present in the scene, the partitions of the consecutive frames should not undergo random fluctuation and be stable. This is rendered even more difficult to achieve as inherent uncertainties may exist in the scene [151]. Going further on the route towards semantic understanding, the requirement for spatio-temporal stability is the starting point for solving the correspondence problem [7]. In terms of moving objects, the tackling of the correspondence problem boils down to defining the trajectories of the objects. This is formally achieved by performing a temporal linkage between successive spatio-temporal partitions. Each object is identified in the successive frames of the sequence and is tracked throughout the video sequence [99]. Such a tracking procedure enables to obtain a layered representation for each object which evolves dynamically through time.

Faced with the above challenges of the spatio-temporal segmentation, some techniques not only rely on the information available in the image, but they also require the scene to satisfy a given set of constraints. These may be interpreted as *a priori* knowledge which is used to perform the segmentation procedure.

For instance, the number of objects existing in the scene may be predefined [19, 82]. Other techniques require the supervision of the user so as to initialize the segmentation process [35]. In [34], it is claimed that supervised methods generally perform better than unsupervised algorithms. However, the mere fact of using *a priori* information in the segmentation process make the algorithm less general. The segmentation process can either no longer be performed automatically or only addresses a certain type of scenes. This loss of generality may be quite crucial in applications such as video coding.

3.4 Spatio-temporal segmentation: the top-down approach

Due to the implicit relationship between motion and segmentation, the spatio-temporal segmentation procedure is a "chicken & egg" problem. In order to tackle this issue, the top-down methods use the outlier detection/rejection paradigm. The objects are sequentially extracted by determining iteratively the successive dominant motions [119, 73, 24, 106, 32, 153, 38, 113, 66, 16]. Pixels complying with the current dominant motion are assumed to comprise an object and are classified as inliers. The other pixels are seen as outliers. These are considered in the next iteration so as to estimate the subsequent dominant motion and the corresponding inliers.

The top-down algorithmic approach to spatio-temporal segmentation is confronted with two main issues. They are namely the generalized aperture problem [82, 24] (see Sec. 2.2.2) and the outlier detection/rejection procedure. The former issue relates to the dominant motion estimation. The dominant motion must indeed be estimated correctly despite the fact that multiple motions may coexist in the motion estimation support. In the presence of such outliers, the dominant motion estimation procedure may be misled and, ultimately, may provide an inaccurate or even erroneous motion. The latter issue deals with the extraction of the object corresponding to the dominant motion. Assuming that the motion estimation has been correctly performed, the object segmentation relies on the outlier detection/rejection paradigm. In that respect, the type of information used to define the outlier has to be predefined. The outlier selection must take into account the inherent ambiguities or errors linked with the type of information used.

With regard to the dominant motion estimation process, Burt [33] advocates that the procedure is rendered more robust by using a pyramidal decomposition. Such a decomposition separates the different motions existing in the scene and permits to estimate each of them independently of the others. In the presence of a dominant motion, the estimation procedure "locks on" this motion, while ignoring the inputs from others. The pyramidal approach has been used by the top-down techniques proposed in [32, 20, 72, 73, 119, 66]. In order to further improve the estimation procedure, Rousseeuw and Leroy [124] advocate for the use of resistant estimators. These are commonly referred to as robust estimators (see Sec. 2.4.4). They have the ability to derive the dominant motion without being influenced by the other motions [14, 113, 132].

The second issue to be addressed by the top-down methods is the outlier detection/rejection paradigm. This process is responsible for determining which areas are moving in line with the estimated dominant motion and which are not. The pixels moving accordingly to the dominant motion are grouped into a single object. The challenge lies in determining which pixels possess this characteristic. To that end, most techniques rely uniquely on the temporal information to perform this task [119, 72, 14, 113]. The previous frame is warped using the estimated dominant motion and the result is compared with the current frame. The outliers are defined as the areas corresponding to large prediction errors. However, the use of temporal information may turn out to be unreliable at the pixel level. This is especially the case in low gradient regions. To overcome these problems, some techniques integrate the residual

measurements over large spatio-temporal regions so as to make the outlier detection more robust [15]. Furthermore, the use of spatial information permits to precisely define the object boundaries [132, 40, 62].

As a summary, the top-down approach has the following advantages. It permits the user to stop the segmentation procedure when a predefined number of objects have been extracted. This may be used in applications where the number of objects composing the scene is known or where only the most important motions (and the corresponding objects) are of interest. A typical example for the latter case is given by techniques aiming at extracting the camera motion [72]. An other advantage of the top-down approach is that the algorithms are generally characterized by their simplicity and low computational load. However, several disadvantages are inherent to the top-down approach. The segmentation may be spoilt by the underlying hypotheses and algorithmic constraints of the top-down approach. For instance, it may be put into fault when the hypothesis of a dominant motion in the scene is wrong. In that case, the motion estimation is unable to correctly estimate the motions and, consequently, the object definition through outlier detection/rejection is affected. On the algorithmic side, the top-down methods do not treat the different motions in an equivalent manner. Based on the outlier detection/rejection paradigm, each motion is defined on the basis of the previously extracted dominant motions. This sequential approach leads to a hierarchy among the different motions and, hence, among the different objects. Finally, a complete partition is not ensured. The successive extraction of the dominant motions may lead to a situation where the remaining outlier is not representative of any object. As a consequence, no image representation based on the spatio-temporal segmentation may be properly obtained.

3.5 Spatio-temporal segmentation: the bottom-up approach

3.5.1 Introduction

In order to resolve the intricate relationship existing between motion and segmentation, the bottom-up approach relies region merging strategy. In the first phase, a set of initial regions is derived. They function as building elements of the objects. Starting from these regions, the bottom-up approach indeed merges them so as to obtain the moving objects. The merging is performed in accordance with a predefined spatio-temporal similarity measure. The hypothesis is that each object may be naturally defined through the spatio-temporal relationships between the regions composing it. The bottom-up approach allows to carry out the extraction of the different objects in a simultaneous manner. This is in sharp contrast with the top-down approach where the objects are sequentially extracted.

The bottom-up approach may be decomposed into three steps. They are respectively the choice of the set of initial regions, the definition of the region similarity measure and, how this measure is used to merge the regions into objects. These three steps are now reviewed in greater detail.

3.5.2 Generation of the set of initial regions

The manner in which the set of initial regions is selected greatly varies. The regions may be arbitrarily defined as arising, for example, from a quadtree segmentation. The image is split into blocks which form the set of initial regions [155]. However, these regions do not embody the content of the scene at hand. In particular, their spatial delimitation does not take into account the spatial information present in the scene. When merging such regions, the resulting objects have a hard time to be spatio-temporally coherent as their building parts, i.e. the initial regions, do not possess this characteristic themselves. Consequently, the initial regions should be, as much as possible, spatio-temporally homogeneous. To

that end, two main approaches have been investigated. The first takes the limiting case of defining each pixel as an initial region [26, 13, 4]. As a pixel represents the smallest image element, it is by definition spatio-temporally coherent. However, pixels are an artifact arising from the image scanning procedure and intrinsically bear no meaning. Furthermore, the set of initial regions should exploit the object connectivity property [78]. This states that each object is formed by a set of areas whose pixels are naturally related to one another. When starting from pixels, one incurs the risk of seeing any isolated pixel or small area being classified as an object.

In contrast to the techniques using pixels as initial regions, techniques have been proposed to define initial regions which are both spatio-temporally homogeneous and visually meaningful. The latter characteristic implies that these regions are bound to be larger than one pixel [151]. As far as the spatio-temporal homogeneity is concerned, it may rely on different criteria. In [8, 128], the regions are generated on the basis of their spatial homogeneity, through static segmentation. The initial set is characterized by its homogeneity of texture or contours. Temporal information may also be used. For instance, the technique proposed by Wang *et al.* [149] starts from an arbitrary quadtree segmentation and relies on the motion information to refine it so as to derive the set of initial regions. In [44], the initial set is generated on the basis of both spatial and temporal information. Starting from an initial static segmentation of the image, the regions which are not well motion compensated are split further. This last procedure is performed according to a static segmentation.

A general remark may be made when considering the techniques proposed to derive initial regions larger than a pixel. Broadly speaking, the extraction of the regions relies on a top-down approach [28] (see Sec. 3.4). Starting from a first estimate of the regions, these are iteratively split if they do not comply with the criterion of coherence. The resulting regions define an oversegmentation of the scene from which the objects are built. This is the reason for denoting the bottom-up approach a "*split-and-merge*" approach.

3.5.3 Region similarity measure

The region similarity measure defines the degree of spatio-temporal affinity existing between two or more regions. According to it, the likelihood that regions are part of a single objects can be assessed. The similarity measure provides the information on which the region merging procedure is based.

The definition of the similarity measure greatly depends on the type of information which is used. For instance, the similarity measure may uniquely rely on spatial information [128, 115, 127]. The resulting objects showing spatial coherence, they are also likely to be homogeneous as far as motion is concerned [95]. However, within a same object, regions with very different spatial characteristics may exist. For instance, let us consider a person's head. The skin of the face is spatially very different from the hair, although they may move in a similar way. Based on this example, it is clear that a similarity measure relying uniquely on spatial information results in an oversegmentation of the objects of the scene.

To remedy the shortcomings of the above approach, the region similarity should use temporal information. This requires a motion representation which, for a given region, enables a comparison between different motions. In this context, the fully parametric approach [11] finds a natural application (see Sec. 2.3.4). The motion information is represented by a set of parameters. Their number and characteristics depend on the model chosen. Such a model can range from the translational model to the perspective one. A common choice is the affine parametric model which consists of six parameters.

When assessing the region similarities, the parametric motion representation has been used in many ways. Two classes of approaches can however be singled out. The first one only uses the temporal information

available in the motion parameter space. The methods of Dufaux *et al.* [44], and Wang and Adelson [149] define the region similarity measure to be the distance existing between the sets of motion parameters in the motion parameter space. This approach is obviously sensitive to error in motion estimation and to the distance measure. A more severe problem is the following. Depending on the scene and the motion model chosen, similar optical flows may be represented by very different sets of motion parameters [5]. In other words, the parameterization of the motion may not have a unique solution and the hypothesis that the motion parameters represent the entire motion information may indeed be wrong. This implies that two regions which are moving in a similar way may turn out to have very different motion parameters.

The alternative way of using temporal parametric information is based on statistics of the residual distributions obtained by motion compensation. When comparing region A with region B, the motion of the latter is applied to the former. The resulting residual distribution gives a measure of how region A moves in the same way as region B. By its very nature, the resulting residual distribution is representative of the mismatch existing between the parametric motion and the real optical flow. It can thus be considered as encompassing the parameters complementary motion information. The information in the residual distribution is used in the form of statistics. For instance, the similarity measure proposed by Adiv [4] is the variance of the residual distribution. Two regions are merged if the variance for the newly formed region is similar to those of the individual regions before merging. However, the use of a single statistic is too drastic a reduction of distribution dimensionality. It may induce wrong merging decisions, since all the information present in the distribution is not examined.

In [155], the temporal information present in both the residual distribution and the parametric representation is used. This is achieved in the framework of the Minimum Description Length (MDL) paradigm [123]. This paradigm directly derives from Occam's words in the 14th century, "*Entities should not be multiplied beyond necessity*". The similarity measure between two regions is defined as the ideal coding reduction. In other words, the similarity represents the number of bits which would be saved if the two regions were merged together.

The definition of the region similarity is a challenging issue. As a general comment, one may say that all the available information should be put to work so as to robustly define the objects present in the scene. The similarity measure should exploit both spatial and temporal information [139, 4, 64]. In particular, Thompson proposes to merge the regions on the basis of a contrast criterion modified through temporal information [139]. However, the two types of information are not combined into a single similarity measure. They are used separately, the emphasis being on the spatial information. Finally, let us underline that the segmentation purpose is to obtain a semantic representation of the scene. To that end, our understanding of the HVS may be of great help. In particular, the fact that the sensitivity of the neurons depends jointly on space and time may give helpful hints [147].

3.5.4 Region merging strategy

The region similarity measure allows the assessment of whether two or more regions are part of the same object. In order to carry out the actual merging procedure, a region merging strategy has however to be defined. It is responsible for making the best use of the information provided by the similarity measure. This strategy should also be robust against error or ambiguities which may occur in the definition of the region similarities. Finally, the computational load implied should remain reasonable. Two types of strategies may be defined. The first type assumes a two-step process [148, 13]. The motions describing the scene are first determined, then the objects are derived by merging the regions corresponding to the each motion. The second type of merging strategy simultaneously determines the motions and the objects [26].

The first type of merging strategy is carried out in two steps. In the first stage, the number of the motions describing the scene are determined. In the second step, these are used to determine which regions should be merged together. In [148], each region is assigned affine motion parameters. These are computed by LMS regression (see Sec. 2.4.2). A k-means clustering algorithm [84] is then applied in the motion parameters space. Hence, this clustering provides a set of motion models representative of the motion in the scene. This set is used to merge the regions which cover the same moving object. Another representative example of a two-step strategy is proposed by Ayer and Sawhney [13]. They propose a MDL criterion which aims at minimizing the cost of the ensemble of the different motions. The criterion is formed of two parts, the coding cost of the motion parameters and the coding cost of the corresponding DFD. Based on this, an iterative Expectation-Maximization (EM) algorithm derives the optimum set of motions. After having obtained these, the regions are merged accordingly. This is performed on the basis of a maximum a posteriori (MAP) estimation.

In contrast to the two-steps techniques described above, Bouthemy and François [26] use a merging strategy which simultaneously determines the set of motions and merges regions. The segmentation is expressed as a relaxation problem based on a Markov Random Field (MRF) modeling. Due to the equivalence between MRF and Gibbs distribution [58], the model reduces to minimizing energy functions over local neighborhoods. Hence, the probability p that the system is in a particular state is given by

$$p = \frac{1}{Z} e^{-U/T} , \tag{3.1}$$

where Z is a normalizing constant (also known as partition function), T is a constant (corresponding to the temperature in statistical mechanics), and U is an energy function (or objective function). The energy U is expressed as a sum of potentials, defined on so-called cliques, that specify how mutual neighbors contribute to the probability of a particular state. The energy function U is minimized by a deterministic relaxation scheme. The labels, which are local, and the motion parameters, which are global, are alternatively and iteratively estimated. A supplementary label is considered in order to determine newly appearing regions. Convergence is achieved when the number of sites whose label has been changed fall below a preset threshold.

3.5.5 Advantages and drawbacks

In comparison to the top-down approach, the bottom-up approach has the following advantages. First, the objects composing the scene are extracted in parallel. In other words, the extraction of a given object does not rely on previously extracted objects. The bottom-up approach defines all the objects simultaneously. Furthermore, the bottom-up approach ensures that a complete decomposition of the scene is obtained. This is essential in application where the segmentation is used as an alternative to the pixel based image representation. However, the bottom-up approach is also characterized by its algorithmic complexity. For instance, the generation of the set of initial regions is often performed using a top-down technique, while the region merging strategy usually involves a complex iterative process. Moreover, the resulting segmentation heavily depends on the region similarity measure. Its definition plays a crucial role and thus requires much attention and efforts.

3.6 Multiframe-based spatio-temporal segmentation

3.6.1 Introduction

The spatio-temporal segmentation is generally carried out between two consecutive frames and, as such, faces many pitfalls. In particular, estimation inaccuracies, noise, as well as the lack of decisive information may alter the interpretation of the scene. The resulting spatio-temporal segmentation may not only be incomplete, but even erroneous. Another problem lies in the stabilization of the spatio-temporal segmentation procedure over time. Assuming that no brisk scene changes occur in the sequence, the successive frames are composed of the same set of objects. The corresponding partitions should thus show similarities with one another. They must be coherent and must define partitions composed of the same set of objects. This stability is a prerequisite to defining the temporal evolution throughout the sequence of the different objects composing the scene.

In order to make the spatio-temporal segmentation more robust and coherent through time, techniques which use many frames have been proposed. Among these techniques, a dichotomy may be made between the batch and the recursive approaches. Finally, the spatio-temporal of consecutive frames directly relates to the problem of tracking the objects through the sequence. These issues are now discussed in greater detail.

3.6.2 Batch vs recursive spatio-temporal segmentation

In order to stabilize the spatio-temporal segmentation procedure, the amount of available information should be increased. The extra information permits to overcome inherent ambiguities as well as problems occuring in a cluttered environment. This is achieved by increasing the temporal support used for analysis [10, 16, 137, 18]. The segmentation procedure can thus take into account the information present in multiple frames. Furthermore, the expansion of the time support also tackles the issue of the coherence of successive spatio-temporal segmentations. Two types of algorithmic approaches have been investigated, respectively the *batch* and the *recursive* approaches.

In the batch approach, the whole sequence is first captured. The current segmentation benefits from information contained in "past" frames as well as in "future" frames [92, 111]. The batch approach works thus off-line and inherently implies a time delay. Batch techniques rely on the notion of $3D$ block. It is formed by the volume defined by the ensemble of successive frames. The $3D$ block is then segmented into $3D$ regions. This procedure may rely on different type of information. In [115, 127], a $3D$ segmentation based on luminance information is performed. Other batch techniques use multiple frames to stabilize the extraction of the motion information. Most of these methods assume motion to be constant in time [54], this assumption being quite restrictive in the case of natural sequences. In [16, 40], this constraint is relaxed as the motion is assumed to vary over time and is modeled using polynomials. The batch approach has interesting properties of information integration. Nevertheless, its inherent time-delay is a serious limitation. This is most problematic in applications such as video coding or real-time dynamic scene analysis.

The alternative to the batch approach is referred to as the recursive approach. The current segmentation only benefits from past information. Thus, it can be performed on-line. The recursive approach is based on the assumption that the current partition is likely to be very similar to the previous one. Similar to the batch approach, the recursive approach tends to ensure coherence between successive partitions. The challenge lies in the manner in which past information influences the current spatio-temporal segmentation

process. Broadly speaking, two classes of recursive techniques may be defined. They differ in the extent to which past information is used in the current segmentation procedure.

The first class of recursive techniques uses past information only to initialize the current segmentation procedure [126, 63, 95, 148, 26]. The previous partition is projected onto the current frame. To perform the projection, the motion of the previous partition is used. The assumption is that the motion does not change significantly in one time interval Δt. The resulting partition defines the starting point of the current segmentation procedure. The interaction between past and current information is limited to the projection step. It indeed stops after the initial guess for the partition has been provided.

The second class of recursive techniques aims at using the past information to a fuller extent than the methods belonging to the first class. In a first step, the techniques in the second class initialize the current segmentation through the motion projection of the previous partition. In a second step, the current segmentation procedure is constrained by past information. The current segmentation procedure may be penalized whenever the current segmentation tries to deviate from the previous one [112, 22, 29]. The modification is accepted only if the gain in terms of spatio-temporal coherence compensates for the lost stability with respect to the previous segmentation. Another way of constraining the current segmentation procedure with previous information is presented in [72]. The dominant object is pursued through a technique using temporal integration. An integrated image is built by accumulating information about the dominant object. By estimating the motion on the basis of the integrated image, the segmentation procedure is implicitly constrained to retrieve the corresponding object.

3.6.3 Object tracking

The tracking concept finds its roots in the biological activity of the human eye [39]. Confronted with a scene containing moving objects, the visual system first detects the object of interest and focalizes on it. Afterwards, the object is followed as it moves so as to keep it at the center of attention. This phenomenon is referred to as *tracking*.

In the framework of image analysis, the tracking process has been formally described as the *correspondence* problem [7]. Given a feature of interest, the issue is to define its position in successive frames. The process allows to identify the feature at successive times, this ensemble of positions permitting the derivation of the feature trajectory [42, 17]. The tracking procedure is a key concept to understand the dynamic scene at hand. Many types of features may undergo a tracking procedure. They range from corners [37], edges [64], a particular object [91] or the ensemble of objects forming the scene [128]. As a general rule, the success of the tracking algorithm depends heavily on the intrinsic meaning of the chosen features. In other words, the tracked features should carry a natural meaning in order to allow for a tracking procedure to be successful.

Due to their intrinsic meaning, the objects obtained through spatio-temporal segmentation are very well suited for tracking. They are semantically meaningful and are homogeneous in terms of motion. They are thus very likely to be found in successive frames of the video sequence. Technically speaking, the object tracking task boils down to assigning the same label to the pixels corresponding to the same object in the successive frames. Two types of object tracking techniques have been proposed: the template based tracking methods and the region based tracking methods. Template-based techniques rely on a deformable contour model [91, 141]. The underlying idea is to model the boundary of the object of interest. The model is expressed in terms of snakes [91] or a mesh [141]. The tracking procedure consists in retrieving in successive frames the contour model. In a first step, the contour model of the former frame is used as an initialization for the current one. To that end, a motion projection is generally used, even though some

techniques simply use the contour model of the former frame without any motion projection [91]. The second step consists in refining the contour model so as to make it fit the object in the current frame. The contour tracking approach is generally implemented when a single object has to be tracked. Furthermore, its tracking ability heavily relies on the hypothesis of motion temporal continuity (see Sec. 2.2.2). If the object performs a maneuver, the tracking algorithm has trouble following it. Finally, the contour tracking approach requires that the first contour model is delineated. This necessitates *a priori* information or inputs from the user.

When facing a complete decomposition of the scene in terms of objects, the region based approach to tracking is very appealing. It relies on global deformation models which allow to follow the object through time [100, 131, 88]. The underlying idea is to use the characteristics of the ensemble of pixels forming the object so as to track it. For instance, Meyer and Bouthemy [100] propose an object tracking algorithm. They use a single affine motion model for each object and assign a second-order temporal trajectory to each affine model parameter. In [88], the objects are modeled through affine invariants. For a given object in the former frame, the object tracking consists in determining which object of the current frame shows the most similar invariants.

In the framework of the region based tracking approach, the object tracking procedure is closely related to the spatio-temporal segmentation step. The task of solving the correspondence problem indeed implies segmenting the objects at successive times. Also, we look for spatio-temporal segmentations which are coherent through time. This requirement intrinsically implies that objects in past segmentation should find their counterparts in the current segmentation, this being the definition of the tracking task. Due to this close relationship, most techniques which address the object tracking problem do not distinguish between the segmentation and the tracking processes [128, 112, 95, 27, 22, 29]. The correspondence problem is solved by labeling the objects in the process of defining them. Such labeling relies on the hypothesis of motion temporal continuity and is simply carried out by projecting the previous partition labels onto the current frame.

Although correlated, object segmentation and object tracking are not similar tasks. The tracking permits the identification of the objects as they move, while the spatio-temporal segmentations provide instantaneous measurements of spatio-temporally coherent regions in each frame [100]. In other terms, segmentation is an image processing technique. It models the image in terms of a set of objects which have given characteristics, but it does not provide any understanding nor analysis of the scene [25]. In contrast, tracking is an image understanding procedure. Although it is intimately linked with the segmentation process, tracking involves a higher level of image analysis. To that end, it requires an object identification procedure which enables to uniquely characterize each object and follow it through time. The fundamental difference existing between the two procedures is highlighted with the example of an object which undergoes total occlusion. When disocclusion takes place, the spatio-temporal segmentation procedure creates a new object. In contrast, the tracking procedure recognizes that the newly appearing object is the same as the one that was occluded several frames ago. The tracking thus requires the ability of identifying an object. In turn, this capacity allows to obtain an object memory.

3.7 Conclusions

This chapter deals with segmenting a video sequence into spatio-temporally coherent entities. First, the notion of spatio-temporal segmentation and the related challenges are presented. The benefits of deriving a spatio-temporal segmentation are discussed along with applications. In a second step, the techniques for spatio-temporal segmentation are reviewed. A dichotomy between the techniques using a top-approach

and those using a bottom-up approach is made. The distinction between techniques relying uniquely on two consecutive frames and those using more frames is also made. Finally, the notion of object tracking is presented.

Chapter 4

Defining spatio-temporally homogeneous regions

4.1 Introduction

This chapter addresses the problem of segmenting the scene into a set $\mathcal{R}$ of homogeneous regions. In order to encompass the visually important features of the scene, the regions have to be well defined both temporally and spatially. This is equivalent to saying that the regions have to be spatio-temporally homogeneous. Generally, the set $\mathcal{R}$ represents an oversegmentation of the scene. Thus, the objects forming the scene may be constructed by merging selectively the regions.

In the literature, two main approaches to generate spatio-temporally homogeneous regions have been proposed. The first one simply defines each pixel as being a region [26, 13, 4]. It is trivial to see that the requirement that the regions be spatio-temporally homogeneous is fulfilled. However, these pixel based regions bear no meaning and are not related to the scene content. The second approach to generate the set $\mathcal{R}$ aims at defining regions larger than one pixel. This approach is in accordance with the working principles of the HVS [151]. Indeed, human visual systems do not rely on a pixel-by-pixel classification, but on some form of local average. The techniques belonging to the second approach group the pixels into regions on the basis of an homogeneity criterion. For instance, the regions may be defined through the spatial homogeneity. This corresponds to a static segmentation of the scene [128]. The criterion may also rely on temporal information. Starting from arbitrary regions, the regions are split until each resulting region is temporally homogeneous [45]. Dufaux *et al.* [44] propose to combine both types of information in order to have temporally homogeneous regions with precisely defined contours.

In this chapter, a novel technique to segment the image into a set $\mathcal{R}$ of spatio-temporally homogeneous regions is presented. The regions are generated through a top-down approach combining spatial, temporal and change information. An initial estimate of the regions is obtained through spatial segmentation. Then, the regions are refined on the basis of motion information and change information. The final set $\mathcal{R}$ contains regions that are characterized by their spatial and temporal homogeneity. An overview of the proposed technique is given is Sec. 4.2. Section 4.3 presents the way in which spatial information is used. The incorporation of temporal information into the region generation process is described in Sec. 4.4. In particular, the technique to correct for the effects of disocclusion phenomena is presented. The refinement of the regions based on change information is proposed in Sec. 4.5. Section 4.6 presents simulation results on different types of video sequences, while the conclusions are drawn in Sec. 4.7.

4.2 Overview of the proposed technique

The method for defining spatio-temporally homogeneous regions follows a top-down approach. Starting from an initial estimate of the regions, these are iteratively split until they satisfy the desired homogeneity characteristics. The first input of the method are the Red, Green and Blue (RGB) components of the frame to be segmented. In the remaining, this frame is referred to as the current frame. The first step of the method relies on spatial information. A static segmentation is performed on each color component of the current frame. These segmentations are then superimposed in order to define a first approximation of the set $\mathcal{R}$. In the second step, the temporal homogeneity of the regions is checked, requiring a motion estimation. In order to precisely estimate the motion of the regions, the effects of the camera motion are first removed. A local motion estimation then takes place between the current frame and the globally motion compensated previous frame. The regions corresponding to a failure of the motion estimation are further split through the above static segmentation procedure. The phenomenon of disocclusion is taken into account. This permits a robust evaluation of whether a given region is temporally homogeneous. Finally, a change detection procedure is carried out in order to detect regions which have a low contrast.

The change information results from the comparison of the current frame with its prediction through motion compensation. The output of the method is the final set $\mathcal{R}$ of spatio-temporally homogeneous regions. The regions are characterized by their temporal homogeneity and their precise boundaries. They represent all visually important details of the image. In Fig. 4.1, an overview of the proposed method is given.

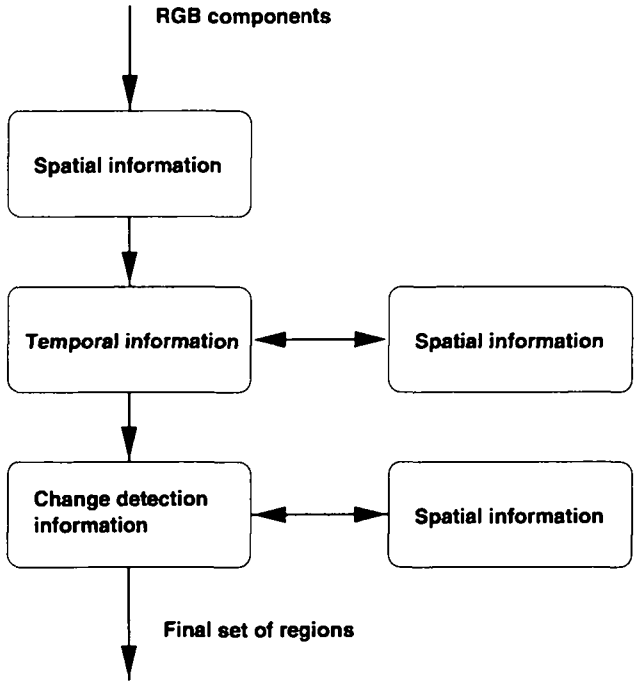

Figure 4.1: Overview of the proposed top-down technique to generate the set of regions $\mathcal{R}$. Initially, a spatial segmentation is performed using the information present in the RGB components of the frame. This segmentation is successively refined through the use of temporal and change information.

4.3 Using spatial information

Spatial information encompasses static characteristics of the scene. Due to its visual significance, it constitutes an ideal starting point for the generation of the regions. In particular, it permits to precisely define the region boundaries. In order to exploit the spatial information, a static segmentation procedure is performed. According to [122], static segmentation techniques can be categorized into two principal classes which respectively rely on

- boundary or contour formation, or,
- region formation.

The focus of the former methods is upon differences or discontinuities whereas it is upon similarities for the latter. Contour-based segmentation techniques precisely determine the spatial position of region boundaries but do not guarantee regions with closed contours. A complete partition of the image in terms of independent regions is not ensured. Contour-based segmentation techniques are therefore not well suited for complete image decompositions and have not been considered any further in this dissertation. Region formation methods aim at segmenting a given image into areas which are homogeneous with respect to

some spatial characteristic. As the region formation approach provides with a complete decomposition of the image, it is used in this dissertation to exploit the spatial information.

When forming the regions, different criteria should be available. For instance, criteria of size and contrast are very appropriate. Moreover, a greater stability of the segmentation is obtained if the region definition is carried out in a hierarchical way. For these reasons, we have chosen to perform the static segmentation through a purely top-down splitting method which is based on mathematical morphology. A detailed description may be found in [95, 128, 126]. It is characterized by the small number of control parameters it requires. In this technique, the segmentation process at each level of the hierarchy is divided into four steps: modeling, simplification, marker extraction and decision.

First, the texture of each region is modeled by means of its luminance values. The modeling residue, i.e. the difference between the approximated and the original image, thus indicates areas in which the current image segmentation does not provide a sufficiently accurate partition of the original image into regions of homogeneous luminance. Therefore, this modeling error rather than the original image is segmented in the subsequent steps. Second, the simplification step controls the nature and the amount of information that is kept for segmentation. As visually important segmentation criteria, the size and the contrast of regions have been identified. The investigated segmentation tool uses a combination of both. Morphological tools are employed to remove small and poorly contrasted regions without corrupting the contour information. In the third step, the pixels which belong to the interior of a region with certainty are correspondingly labeled. In order to identify homogeneous areas in the simplified image, an adaptive thresholding of the morphological gradient image is employed. The marker extraction fixes the number and the interior of the regions to be segmented. Some pixels, however, remain unassigned. They correspond to areas of uncertainty which are mainly located at the region boundaries. The decision about the cluster-membership of the unmarked pixels is carried out by a watershed algorithm. Since the original watershed approach relies on morphological gradients, contour information is partially lost. A modified version which operates directly on the pixel values is therefore applied. This renders it equivalent to a region growing technique.

As the static segmentation is carried out in a hierarchical fashion, it results in a progressive improvement of the segmentation. Starting from a coarse partition with a small number of regions, the segmentation is refined on each level of the hierarchy by introducing more regions. Typically, three or four levels are used. The first two or three employ a size criterion and the last one a contrast criterion.

Since an extraction of visually important regions from the scene is desired, the derivation of the set of regions $\mathcal{R}$ should rely on a similar input data representation as the human visual system. Therefore, the spatial information is exploited in the RGB color coordinate system. The above technique for spatial segmentation is applied on each of the three color components. Consequently, the term "luminance" in the above technique refers to either to the red, or the green or the blue color component of the frame to be segmented. The segmentations obtained on the RGB components are combined into a single final segmentation. This is achieved by considering each distinct combination of color segmentation labels as an individual region. In order to avoid an oversegmentation, small regions which result from this combination are removed. To that end, the pixels in the small regions are, first, marked as unassigned and, then, attributed to neighboring regions through the modified version of the watershed algorithm. An example of the proposed procedure to generate the set of regions $\mathcal{R}$ is given in Fig. 4.2. The segmentations obtained on the different color components are seen to be mutually complementary. When combined, these segmentations define a set of regions $\mathcal{R}$ which defines a good starting point on the route to segmenting the scene into spatio-temporally homogeneous regions.

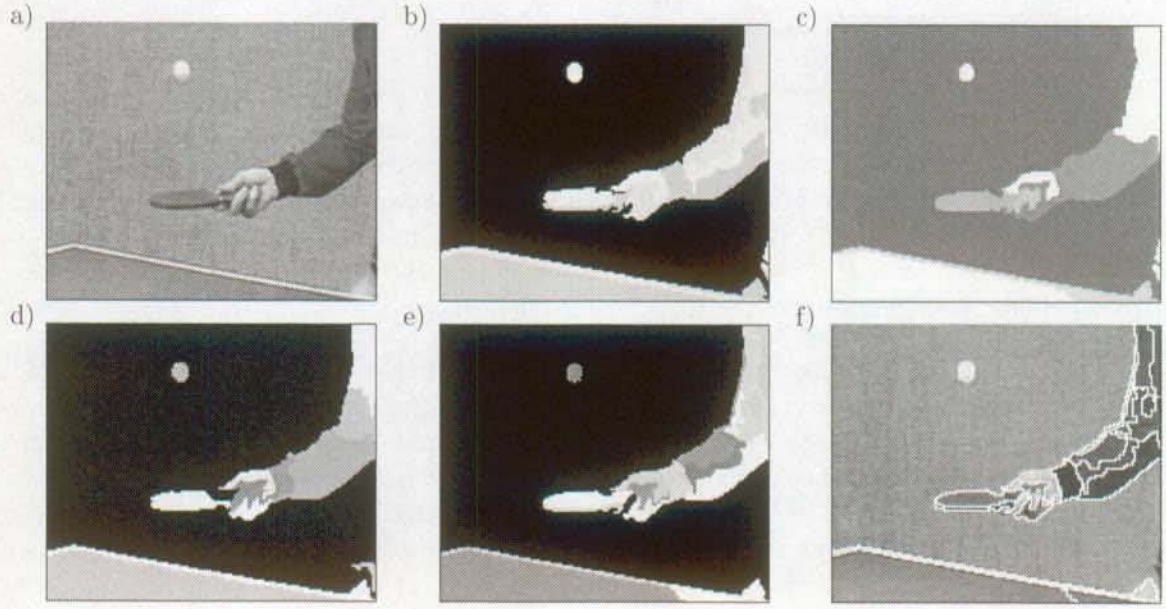

Figure 4.2: "Table Tennis": Spatial segmentation using the RGB components. **(a)** Current frame, **(b)**, **(c)** and **(d)** spatial segmentation on respectively the red, the green and the blue component of the current frame, **(e)** final spatial segmentation and, **(f)** final spatial segmentation superimposed onto the current frame.

4.4 Using motion information

4.4.1 Introduction

When using the spatial information, the resulting regions are characterized by the homogeneity of their texture. However, they are not necessarily temporally homogeneous. Two regions of similar texture may be moving differently. They can be differentiated only on the basis of temporal information. Thus, the incorporation of the temporal information into the generation of the set of regions aims at refining the spatial segmentation for those regions whose support is not characterized by an uniform movement. Thereby, the partition is enhanced in accordance to the requirement of having regions which are both spatially and temporally homogeneous.

When using the temporal information, we distinguish between the motion due to the camera from the motions inherent to the scene. The former is referred to as global motion, while the latter are referred to as local motions. Furthermore, the effects due to disocclusion phenomena are removed. This permits a robust evaluation of whether a given region is temporally homogeneous.

4.4.2 Global and local motion estimation

The motion arising in a scene can be decomposed into a global motion due to the camera and local motions due to the displacements of the objects in the scene (see Sec. 2.2.3). In order to efficiently handle camera motion, this has to be distinguished from the motions inherent to the scene. To that end, a global motion estimation is first carried out [103]. This permits the removal of the apparent movement of the whole scene caused by the camera. Thereby, the subsequent local motion estimation yields more precise

results.

For both global and local motions, the affine model is used to characterize the motion. It is indeed able to model the motion of a rigid body satisfactorily, while still remaining computationally tractable [26]. The affine model represents a good compromise between model efficiency and model complexity. When estimating the parameters of affine model, pixels within a region are assumed to undergo a coherent motion. Errors may occur when the support of the estimation is not well defined and the latter hypothesis does not hold. As explained is Sec. 2.4.4, this problem is tackled through the use of a robust estimator. In the simulations, the motion estimator is the Mean Absolute Difference (MAD).

When estimating the camera motion, the process may be corrupted by the presence of local motions. In order to overcome this problem, a background-foreground mask is defined. This mask demarcates the background area. Then, the global motion is estimated on the background area. This restriction to the background permits the removal of outliers arising from the foreground regions, resulting in more precise estimation of the camera motion. The camera motion is used to obtain a first approximation of the current frame by removing the effects due to the camera motion. This is achieved by globally motion compensating the previous frame. The remaining differences between the latter and the current frame are mainly due to local motions. A local motion estimation is performed for every region in the set $\mathcal{R}$ in order to assess its temporal homogeneity.

The distinction between the background and the foreground relies on the technique for spatio-temporal segmentation presented in Chapter 5. First, the current frame is partitioned through a quadtree segmentation. Starting from this partition, the spatio-temporal segmentation procedure then defines the objects present in the scene (see Sec. 5.5.2). The object that covers most of the border area of the current frame is classified as background. All others are considered as foreground objects. The separation between the background and foreground areas for a frame of the sequence "Table Tennis" is illustrated in Fig. 4.3. The ball, the arm and the hand with the bat are detected as foreground, while the rest of the frame is seen as background.

The global motion estimation, which takes as support uniquely the background, is performed through a matching technique (see Sec. 2.4.3). To decrease the computational complexity and to allow a non-exhaustive search while avoiding local minima, a Gaussian pyramidal structure is used to represent the input images. The final motion parameters at one level propagate as initial estimates on the next level. The obtained affine parameters represent the estimated camera movement. This is removed from the scene through global motion compensation and, subsequently, local motions are estimated. Similar to global motion estimation, the estimation of the local motions is performed through a region matching technique embedded into a Gaussian pyramid structure. Furthermore, a deterministic relaxation scheme is applied during the motion estimation procedure. It compares the sets of motion parameters obtained for neighboring regions and selects, for each region, the one providing the lowest prediction error. This deterministic relaxation scheme allows to avoid local minima. In the remaining, the notion of previous frame should be understood as the previous frame which has been compensated for global motion.

4.4.3 Detection of the disocclusion effects

After having estimated the motion of the different regions, one can predict the current frame from the previous one. By subtracting this prediction from the actual image, the Displaced Frame Difference (DFD) is obtained. It allows an assessment of the accuracy of the segmentation by indicating areas for which the current partition is potentially inadequate. An examination and suitable splitting of these areas allows to refine the segmentation.

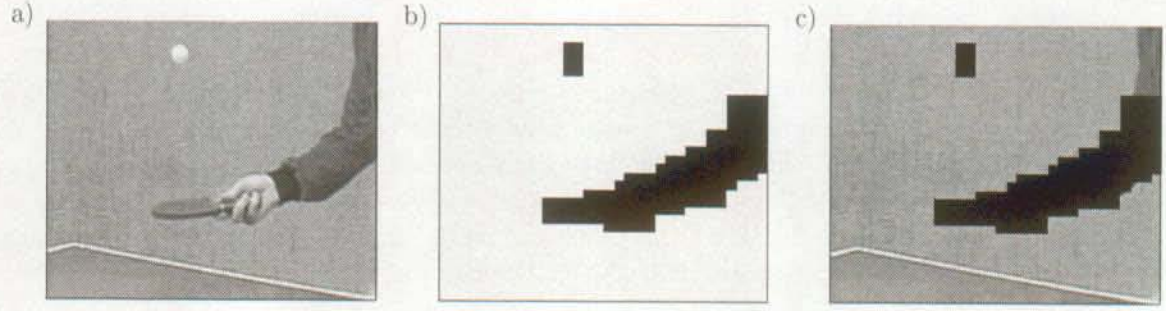

Figure 4.3: "Table Tennis": Background-foreground distinction. **(a)** Current frame, **(b)** background-foreground distinction through spatio-temporal segmentation and, **(c)** resulting background area.

The current frame is predicted from the previous frame. The two frames are generally very similar as they represent successive frames of a video sequence. However, there may be regions in the current frame which cannot be predicted. These regions correspond to areas that are uncovered as the objects move. This phenomenon is referred to as *disocclusion*. The inverse phenomenon is called *occlusion*. It refers to areas that are present in the previous frame, but are covered in the current frame. The phenomenon of disocclusion is illustrated in Fig. 4.4. The prediction of the current frame fails in image areas which are disoccluded. This is especially apparent for the ball. It appears twice in the predicted image, once at its new position and once at its old position in the background region.

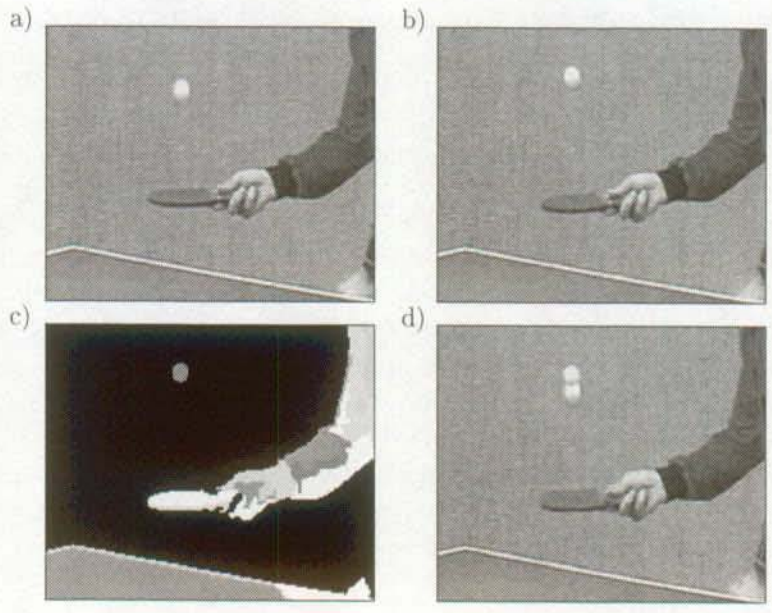

Figure 4.4: "Table Tennis": Example of disocclusion phenomena. **(a)** Previous frame, **(b)** current frame, **(c)** current set of regions $\mathcal{R}$ and, **(d)** prediction of the current frame.

When assessing the temporal homogeneity of a region, the errors resulting from disoccluded areas should not be taken into account. They indeed correspond to a failure of the prediction procedure and do not reflect the temporal homogeneity of the region. To that end, a method is proposed to detect the disoccluded areas. The proposed method proceeds according to the following two steps:

1. Each region is backward motion compensated onto the previous frame.
2. A mask of the disoccluded areas is built. For each region, the current position and the backward projected position are compared. All the pixels which are in the latter but not in the former are seen as disocclusion areas.

The proposed approach is demonstrated in Fig. 4.5. Let two independently moving regions denoted I and II be present in the scene in front of a static background III. In Fig. 4.5(a), they are represented with their respective movement from one frame to the next. The disocclusion mask marks the areas where disocclusion is taking place as shown in Fig. 4.5(b).

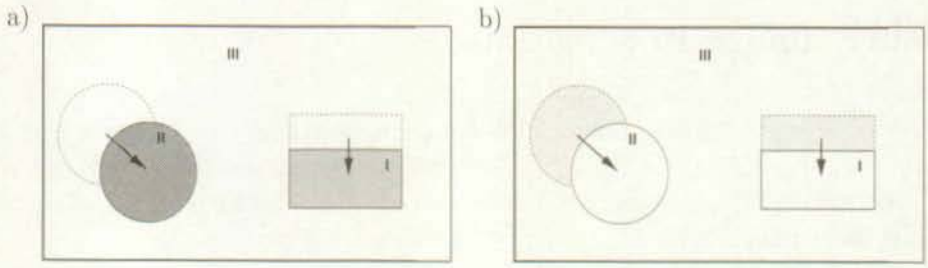

Figure 4.5: Illustration of the algorithm for detecting the disoccluded areas. **(a)** Example of a scene with two independently moving regions denoted I and II in front of a static background III, **(b)** corresponding disocclusion mask (disoccluded areas are represented in gray).

An example of the detection of the disoccluded areas in presented in Fig. 4.6. The prediction yielded by motion compensation is corrected with the luminance values of the current frame in those parts which belong to the disocclusion mask. The predicted frame now shows only minor deviations from the current image in Fig. 4.4(b). Obviously, the detection of the disoccluded areas permits a precise assessment of whether the regions of the set $\mathcal{R}$ are temporally homogeneous.

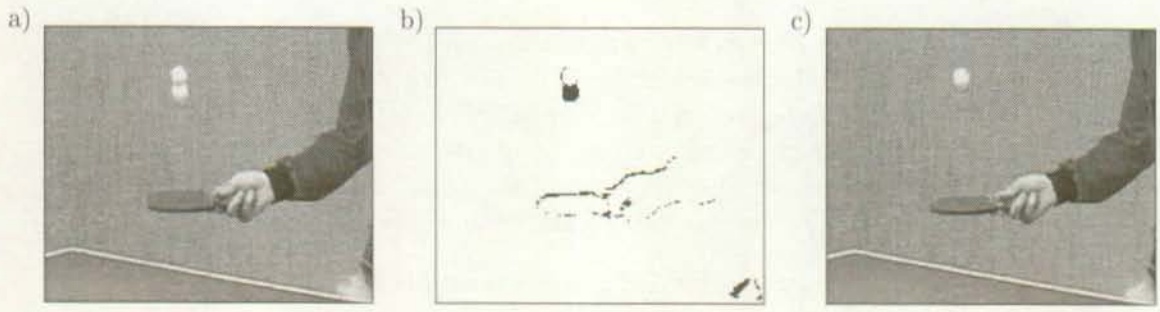

Figure 4.6: "Table Tennis": Detection of the disocclusion effects. **(a)** Prediction of the current frame without correcting for the disoccluded areas, **(b)** disocclusion mask, and **(c)** prediction of the current frame after correction of the disoccluded areas.

4.4.4 Temporally homogeneous regions

In order to detect the regions whose support is not characterized by a coherent movement, the Mean Square Error (MSE) of the prediction error of each region is calculated. The MSE statistic is used as

it gives a predominant importance to outliers. It is thus very well suited for detecting regions which encompass multiple motions. The computation of the MSE does not take into account the contributions arising from areas detected as disoccluded. If the MSE nevertheless exceeds a predefined threshold, the corresponding region is split. For this purpose, a spatial oversegmentation of the image is used. It is provided by the same algorithm that has been used to generate the initial spatial segmentation. For each region to be split, the temporal homogeneity of the subregions provided by the oversegmentation is examined. If the MSE of a subregion exceeds the MSE of the region by a preset percentage, the subregion is retained as a refinement of the region. Otherwise, the subregion is not seen as a valid refinement. This improvement of the static segmentation allows to recover areas with significant distinct motion and, which can therefore be assumed to be visually important.

4.5 Using change information

The objective of change detection is to distinguish temporal variations caused by noise from those resulting from motion or lighting. Traditionally, change detectors are employed to separate moving image areas and static background [97]. They usually rely on a global thresholding which yields a binary change mask, followed by its refinement.

In the process of generating the set of regions $\mathcal{R}$, the change information may provides useful refinements. Indeed, it is complementary to the spatial and temporal information which were used so far. In particular, those are unable to distinguish between parts of objects and the background if their luminance is similar. Furthermore, the refinement of the regions on the basis of the temporal information is highly dependent on the splitting threshold. If it is too high, no refinement takes place. If it is too low, virtually all the regions are subdivided. The choice of a suitable threshold is aggravated by the presence of noise in the scene. The distinction between systematic prediction errors in smaller parts of a region support and random prediction errors distributed over the whole support becomes difficult if only the resulting MSE is considered. The change information permits a precise spatial localization of the areas with high systematic prediction errors. It is thus an useful complementary information to refine the set of regions $\mathcal{R}$.

Change detectors have been applied so far to successive, original frames of video sequences. The results obtained by this approach deteriorate considerably in the presence of motion in the scene. For instance, a global motion causes the change detection mask to comprise the whole frame since no static areas remain. The change detection thereby becomes meaningless. Local motions result in a blurring of the mask which becomes especially evident and annoying for fast moving regions. Therefore, the successful application of current change detection algorithm has been limited to sequences with hardly any motion.

However, the drawbacks of the change detection mask are avoidable if the technique for prediction presented in Sec. 4.4 is employed. Instead of two successive original frames, one original frame and its prediction from the preceding frame are used as input to the change detection algorithm. Furthermore, since the disocclusion effects have been corrected for, they do not mislead the change detection procedure. In this dissertation, this approach has been applied using the change detection technique proposed by Aach *et al.* [3]. This change detection technique relies on the principle of hypothesis testing to decide whether a pixel has to be considered as changed. Furthermore, Aach *et al.* refine the change detection mask through a technique based on a Markov random field. The refinement precisely locates the boundaries between changed and unchanged areas.

The information arising from the change detection mask is exploited to refine the set of regions $\mathcal{R}$. However, the detection mask tends to be too large. Besides, some of the areas detected as changed do not

require any modification of the current set of regions $\mathcal{R}$. These phenomena arise from inaccuracies in the motion prediction of the current frame as well as in the correction for disocclusion effects. Furthermore, illumination changes may occur in the scene. In order to take into account these inaccuracies, an adaptation of the mask to the spatial information is performed as follows. First, the change detection mask is partitioned according to the current set of regions $\mathcal{R}$. Second, the border areas of the mask are determined by comparing the original mask with an eroded version. The pixels contained only in the original mask but not in the eroded one are classified as border pixels and marked as unassigned in the label image. The decision as to which region they should be added to is taken by the modified watershed algorithm introduced in Sec. 4.3. In a subsequent step, small regions are removed on the basis of temporal information. A local motion estimation is carried out for all regions that are kept. The small regions are then merged with the adjacent large region whose motion yields the minimal prediction error for it. Again, the decision is taken in terms of MSE.

Figure 4.7 gives an example of the use of the change information for a frame taken from the sequence "Table Tennis". Compared to the initial segmentation, the final set of regions $\mathcal{R}$ shows significant improvements. In particular, the finger of the right hand placed under the bat is recovered and the left hand is separated from the table to which it was connected.

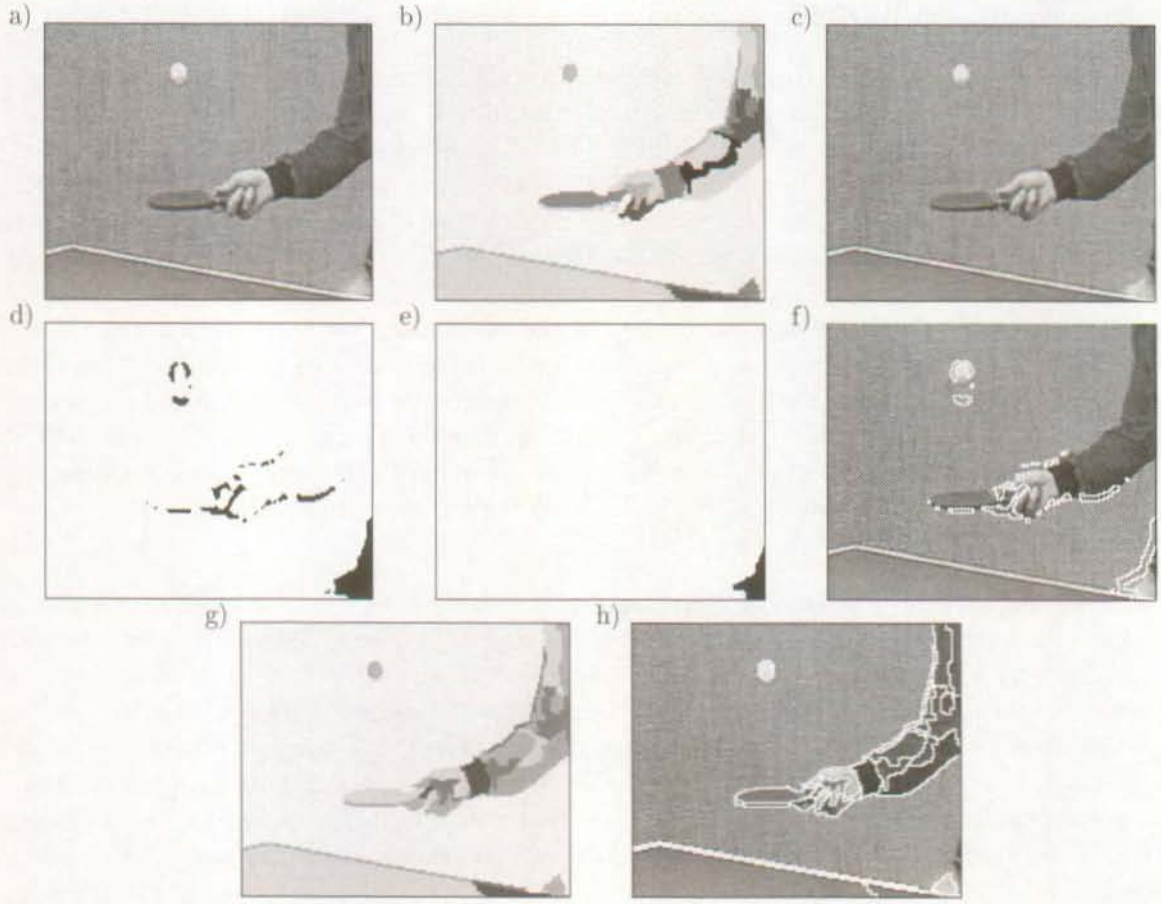

Figure 4.7: "Table Tennis": Change detection mask. **(a)** Current frame, **(b)** set of regions $\mathcal{R}$ obtained after using spatial and temporal information, **(c)** prediction of the current frame with correction for the disocclusion effects, **(d)** change detection mask, **(e)** eroded change detection mask, **(f)** border areas of the change detection mask are shown onto the current frame, **(g)** final set of regions $\mathcal{R}$ and, **(h)** final set of regions $\mathcal{R}$ superimposed onto the current frame.

4.6 Simulation results

The proposed algorithm to segment the scene into spatio-temporally homogeneous regions has been evaluated on different types of sequences. Simulation results are presented for three video sequences: "Akiyo", "Table Tennis" and "Foreman". More details about these video sequences are given in Appendix A.

Figure 4.8 presents the different stages of the proposed method for a typical frame of the video sequence "Akiyo". When performing the static segmentation, the resulting set of regions $\mathcal{R}$ includes most of the visually meaningful regions composing the scene. However, some regions such as the hair are not detected, while the whole face is considered as a single region. The set $\mathcal{R}$ is refined using temporal information. Even when disocclusion effects are removed, a high level of temporal inhomogeneity is detected at the level of the face. The algorithm refines the set $\mathcal{R}$ by segmenting the face further. Regions corresponding to the right eye and to the mouth are isolated. Notice that no region corresponding to the left eye is created. The reason is that the MSE of the prediction error in this area is not high enough for accepting this refinement (see Sec. 4.4.4). The final refinement of the set $\mathcal{R}$ uses the change detection mask. At this stage, regions with low contrast are detected. In particular, the hair is considered as a necessary refinement and the corresponding region is isolated. The same is the case for two regions in the left shoulder. The final set $\mathcal{R}$ comprises the ensemble of regions which are visually important in the scene. Furthermore, the spatial definition of the regions matches precisely the scene content.

The generation of the set of regions $\mathcal{R}$ for a typical frame of the video sequence "Table Tennis" is illustrated in Fig. 4.9. The set of regions $\mathcal{R}$ derived through spatial information is quite accurate. However, flaws may be seen at the level of the hand holding the bat. Some fingers are indeed mistakenly put in the background region. Furthermore, the left hand is connected to a region delimiting the border of the table. The use of temporal information brings no refinement. All the regions are seen as being temporally homogeneous. Note that, in case the disocclusion effects were not removed before testing the temporal homogeneity, the background region would be split into smaller regions. This is due to the very high error occuring in the area disoccluded by the ball and the left hand. Finally, the information provided by the change detection mask permits to solve the inaccuracies existing in the set $\mathcal{R}$. The fingers that were not detected by the static segmentation are recovered. In particular, the finger which is placed under the bat is perfectly isolated. Furthermore, the left hand is separated from the table. The upper part of the left hand is also detected by the change detection mask. However, no corresponding regions appear in the final set of regions $\mathcal{R}$. This results from the fact that these regions are discarded on the basis of their small size.

The last example of the generation of spatio-temporally homogeneous regions is based on a typical frame of the video sequence "Foreman". It is illustrated in Fig. 4.10. The static segmentation is able to detect most of the important regions. In particular, the combinations of the RGB components permits the detection of the helmet. Flaws are however present. These are noticeable around the ear and part of the right shoulder which are combined with regions of the background. The temporal information is not very useful. All the regions are seen as temporally homogeneous. The set $\mathcal{R}$ is however refined through the change information. The ear is separated from the background region and is created as a single region. This also occurs for the parts of the shoulder which were erroneously considered part of the background. The final set of regions $\mathcal{R}$ decomposes the scene into the most visually important regions. However, the segmentation fails for a region at the back of the helmet. This region features a low contrast combined with a slow motion. Neither spatial, nor temporal nor change information is able to detect it. The fact that the human viewer sees this region as distinct from the background relies on a semantic understanding of the object "helmet". From past experience, the human viewer indeed expects an helmet to have a certain shape. This knowledge is the basis for an inference procedure which permits to separate the region from the background.

Figure 4.8: "Akiyo": Definition of spatio-temporally homogeneous regions. **(a)** Set of regions $\mathcal{R}$ after the spatial segmentation, **(b)** boundaries of the set of regions $\mathcal{R}$ obtained after the spatial segmentation superimposed onto the current frame, **(c)** DFD before correcting for the disocclusion effects, **(d)** DFD after correcting for the disocclusion effects, **(e)** set of regions $\mathcal{R}$ after the use of temporal information, **(f)** boundaries of the set of regions $\mathcal{R}$ obtained after the use of temporal information superimposed onto the current frame, **(g)** change detection mask, **(h)** border areas of the change detection mask are shown onto the current frame, **(i)** final set of regions $\mathcal{R}$ after the use of change information, **(j)** boundaries of the final set of regions $\mathcal{R}$ obtained after the use of change information superimposed onto the current frame.

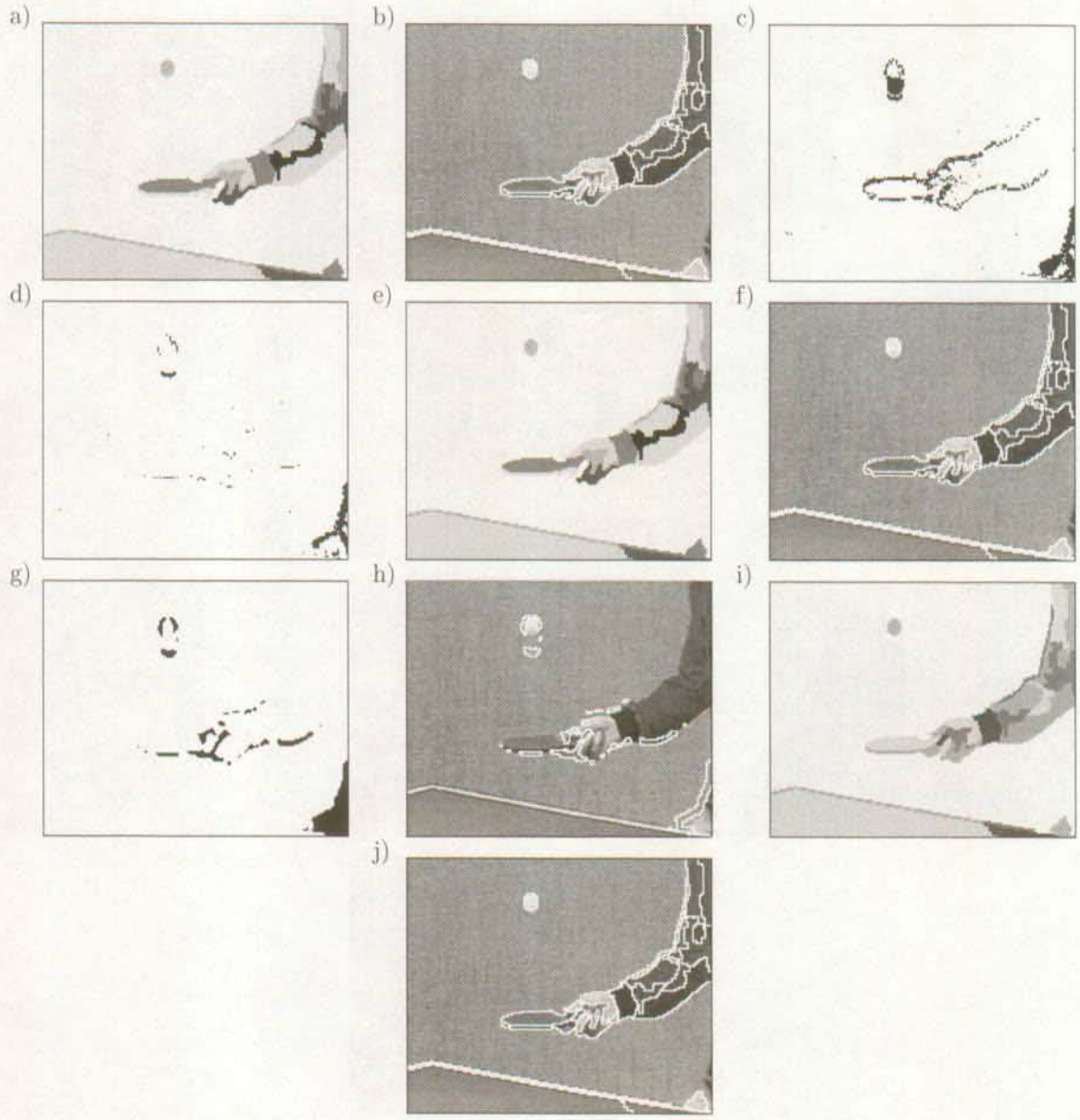

Figure 4.9: "Table Tennis": Definition of spatio-temporally homogeneous regions. **(a)** Set of regions $\mathcal{R}$ after the spatial segmentation, **(b)** boundaries of the set of regions $\mathcal{R}$ obtained after the spatial segmentation superimposed onto the current frame, **(c)** DFD before correcting for the disocclusion effects, **(d)** DFD after correcting for the disocclusion effects, **(e)** set of regions $\mathcal{R}$ after the use of temporal information, **(f)** boundaries of the set of regions $\mathcal{R}$ obtained after the use of temporal information superimposed onto the current frame, **(g)** change detection mask, **(h)** border areas of the change detection mask are shown onto the current frame, **(i)** final set of regions $\mathcal{R}$ after the use of change information, **(j)** boundaries of the final set of regions $\mathcal{R}$ obtained after the use of change information superimposed onto the current frame.

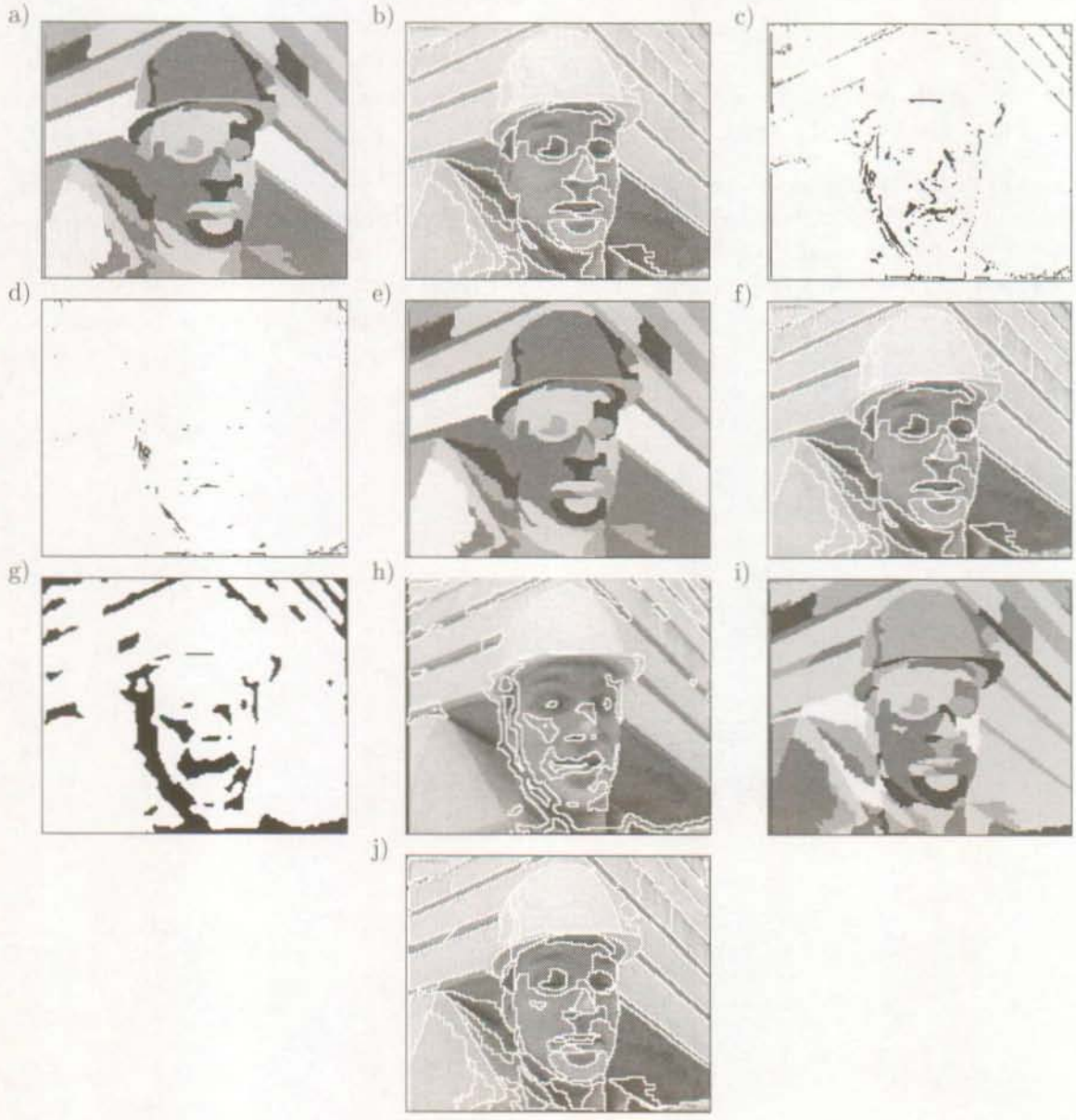

Figure 4.10: "Foreman": Definition of spatio-temporally homogeneous regions. **(a)** Set of regions $\mathcal{R}$ after the spatial segmentation, **(b)** boundaries of the set of regions $\mathcal{R}$ obtained after the spatial segmentation superimposed onto the current frame, **(c)** DFD before correcting for the disocclusion effects, **(d)** DFD after correcting for the disocclusion effects, **(e)** set of regions $\mathcal{R}$ after the use of temporal information, **(f)** boundaries of the set of regions $\mathcal{R}$ obtained after the use of temporal information superimposed onto the current frame, **(g)** change detection mask, **(h)** border areas of the change detection mask are shown onto the current frame, **(i)** final set of regions $\mathcal{R}$ after the use of change information, **(j)** boundaries of the final set of regions $\mathcal{R}$ obtained after the use of change information superimposed onto the current frame.

4.7 Conclusions

In this chapter, a method to segment an image into spatio-temporally homogeneous regions has been described. The regions are generated through a top-down approach. Starting from a static segmentation based on the RGB color components, the set $\mathcal{R}$ is successively refined by exploiting temporal and change information. The procedure is rendered more robust through the correction for the effect of disocclusion phenomena. Furthermore, the estimation of the motion distinguishes between the camera motion and the motion inherent to the scene.

The proposed technique has been evaluated on several video sequences. The combination of spatial, temporal and change information proves to be very efficient. Applied successively, these complementary types of information permit a gradual improvement of the set of regions $\mathcal{R}$. The final set of regions $\mathcal{R}$ comprises all the visually meaningful regions, all of these being characterized by their spatial and temporal homogeneity. Finally, the necessity of correcting for disocclusion effects is clearly demonstrated.

Chapter 5

Region merging for spatio-temporal segmentation

5.1 Introduction

This chapter considers the problem of segmenting a video sequence in terms of multiple moving objects. Each frame of the sequence is segmented using information available in the current frame and in the previous frame. No other information is taken into account. In particular, the number of objects forming the scene is unknown and the scene may include a camera motion. The segmentation must fully partition the scene. A complete segmentation is a collection $\mathcal{O}$ of N_o objects, $\mathcal{O} = \{O_1, O_2, ..., O_{N_o}\}$, such as the union of the ensemble of objects, i.e. $\bigcup_{k=1}^{N} O_k$, forms exactly the scene. Further, we assume that objects do not overlap, $O_i \bigcap O_j = \emptyset$, $i, j \in [1, ..., N_o], i \neq j$, [30]. The resulting segmentation constitutes an alternative representation to the waveform representation. Being semantically meaningful, it is perfectly suited to applications where the scene is analyzed in terms of its content. Examples of applications are object tracking [104] and structure from motion [4]. Furthermore, this semantic approach is very appealing for coding applications in the context of second generation video coding [142].

As discussed in Chapter 3, the techniques proposed for spatio-temporal segmentation may be classified as either top-down or bottom-up approaches. Techniques in the former class extract the objects by estimating the successive dominant motions present in the scene. Given a dominant motion, the corresponding object is defined on the basis of the outlier detection/rejection paradigm. The top-down approach is confronted with several problems. Its sequential character introduces a sort of ranking between the extracted objects. Furthermore, the hypothesis that a dominant motion exists at each object extraction may be put into fault. In turn, this adversely influences the procedure of outlier detection/rejection which is already very sensitive by nature. Finally, a complete partition of the scene is not ensured.

The bottom-up approach addresses the problem of spatio-temporal segmentation as a region merging problem. The hypothesis is that each object is formed by an ensemble of regions which show strong spatio-temporal similarity with one another. In the remaining, the notions of bottom-up and region merging are used interchangeably. In contrast to the top-down approach, the bottom-up approach permits a simultaneous definition of the different objects forming the scene. They are indeed built in parallel as the different regions are assigned to them through the merging process. Furthermore, the bottom-up approach ensures that a complete partition of the scene is obtained. Three major issues are to be examined when using the bottom-up approach. The first issue is the definition of the set of initial regions. Each initial region should ideally be part of an object or an object itself. The second issue deals with the definition of the spatio-temporal similarity. The similarity permits to compare the regions with one another and to determine which belong to the same object. The definition of the similarity opens questions about which data to choose, in which framework to use them and, if many types of data are exploited, how to combine them. The third issue of the bottom-up approach is the strategy chosen to exploit the information given by the spatio-temporal similarity.

In this chapter, a region merging technique for spatio-temporal segmentation is presented. Starting from the set $\mathcal{R}$ of N_r initial regions, $\mathcal{R} = \{R_1, R_2, ..., R_{N_r}\}$, it iteratively merges them in order to automatically build the set $\mathcal{O}$ of the N_o different connected objects forming the scene, $\mathcal{O} = \{O_1, O_2, ..., O_{N_o}\}$. No *a priori* information such as the number of objects N_o, the type of scene or motion is used. An overview of the proposed technique is given in Sec. 5.2. In Sec. 5.3, the spatio-temporal similarity is derived. Based on both spatial and temporal information, it robustly integrates them into a single measure in the statistical context of hypothesis testing. The region merging strategy is presented in Sec. 5.4. The information about spatio-temporal similarities between regions is represented as a graph. The region merging is carried out in accordance to two graph-clustering rules in a hierarchical manner. The graph is dynamically updated as the merging proceeds. Section 5.5 presents simulation results on different types of video sequences, while the conclusions are drawn in Sec. 5.6.

5.2 Overview of the proposed technique

In this chapter, we propose a method for spatio-temporal segmentation. The method is based on a region merging procedure, and hence, is a bottom-up approach. This method assumes the set $\mathcal{R}$ of N_r initial regions to be available (see Sec. 3.5.2). The manner in which they are obtained is not addressed here. The reader may turn to Sec. 3.5.2 for details on this subject. Starting from the set $\mathcal{R}$, the objects constituting the scene are automatically built. The proposed spatio-temporal segmentation is carried out in three steps. First, the spatio-temporal similarities existing between the regions are computed. To that end, the framework of hypothesis testing is adopted. In the second step, the spatio-temporal similarities are used to build a graph which synthesizes all the information available about the regions and their relationships. The graph is then used to decide which regions should be merged. These regions are merged to form new regions whose characteristics are used to update the graph. The process is iterated until no further merging occurs. The final step of the proposed method may be seen as post-processing. Objects which are too small are merged with larger objects. This is performed on the basis of temporal information using the technique described in Sec. 4.5. Figure 5.1 shows the flowchart of the proposed method.

The proposed spatio-temporal similarity aims at using both spatial and temporal information. When assessing the possibility of merging regions, the resemblances between their motions and their spatial characteristics are utilized. The spatio-temporal similarity is expressed in the statistical framework of hypothesis testing, the assumption to be tested being whether two regions are spatio-temporally similar. The significance level of this assumption defines the proposed spatio-temporal similarity. It is obtained by examining two hypotheses, one relying on spatial information and the other relying on temporal information. The former hypothesis states that two regions are spatially coherent. The significance level of this hypothesis defines their spatial similarity. The likelihood ratio test is used as the test statistic. The testing procedure is rendered less sensitive to noise and false alarms through the use of robust data. The hypothesis based on temporal information says that the two regions have a similar motion. The significance level of this hypothesis defines their temporal similarity. The hypothesis is tested through a nonparametric test statistic referred to as the *Modified Kolmogorov-Smirnov* (MKS) test. It combines the motion information present in the motion parameters with the one present in the whole residual distributions. The test also takes into account the presence of outliers. The spatio-temporal similarity is defined as a combination of both the temporal and spatial similarities. It is built as a single quantity integrating both types of information. This results in an inter-dependence of spatial and temporal information which is coherent with our understanding of the HVS [145, 147]. However, more emphasis is put on the temporal similarity. This is in line with the notion that an object is primarily defined by its temporal coherence.

The proposed region merging strategy aims to exploit fully the information provided by the spatio-temporal similarity. To that end, the similarities among regions are summarized in the form of a weighted, directed graph. The region merging procedure then boils down to a graph clustering problem. The proposed clustering strategy relies on two clustering rules, referred to as the *strong rule* and the *weak rule*. These complementary rules are derived in order to exploit the natural structures present in the graph. They also take into account the shortcomings and likely errors of the information represented in the graph. Typical examples of such shortcomings or errors are erroneous motion information or badly defined regions. The clustering procedure is carried out by first applying the strong rule. This is performed in a hierarchical fashion where the clustering requirements decrease at every step. The merging procedure is embedded into a dynamic graph update strategy which refreshes after each merging the information reported in the graph. After the strong rule, the resulting graph is clustered through the weak rule. Similar to the strong rule, a hierarchical approach is used which is embedded in a dynamic graph update strategy. The whole procedure stops when no merging any longer occurs. The ensemble of

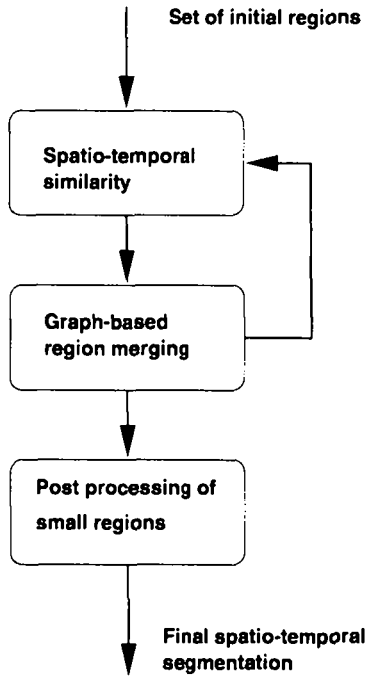

Figure 5.1: Overview of the proposed algorithm for spatio-temporal segmentation. The set of initial regions is assumed known. These regions are iteratively merged in order to build the objects of the scene. This procedure has two phases, being the computation of spatio-temporal similarities and the graph-based decision of which regions to merge. After the completion of the merging procedure, a post-processing step removes the regions that are too small to represent valid objects.

remaining regions are defined as the different objects forming the scene. The small regions are removed by merging them with the adjacent region which possesses the best-suited motion. The decision is taken in terms of the MSE of the prediction error.

5.3 The spatio-temporal similarity

5.3.1 Introduction

When assessing whether two or more regions belong to a single object, a measure of their mutual similarity should be utilized. Based on this information, a merging strategy may be applied in order to decompose the scene into its constituting objects. The region similarity measure proposed here exploits both spatial and temporal information. However, we place more emphasis on temporal information as we primarily define objects as coherently moving entities. The spatio-temporal similarity is derived in the context of hypothesis testing. For each region, temporal and spatial information is extracted. The spatio-temporal similarity, $Sim(AB)$, between two regions, A and B, is defined as a combination of the result, T_{AB}, of a test statistic on the temporal information and the result, S_{AB}, of a test statistic on the spatial information. The spatio-temporal similarity $Sim(AB)$ integrates into a single value both types of information.

5.3.2 Hypothesis testing

In the presence of an unknown phenomenon, one has to make a hypothesis about its origin. Relying on this hypothesis, a model of the phenomenon is derived which permits to predict its evolution. As no certainty exists about the phenomenon, the model is usually statistical. A fundamental step in the modeling procedure is to check whether the underlying hypothesis is correct. This process of model verification is termed *hypothesis testing* [114]. Relying on collected data, it determines whether the evidence supports the hypothesis. Hypothesis testing methods have been applied in almost every domain of scientific research, from the detection of signals, Mendel's theory about heredity, to the question of whether salt is bad for old people.

In order to explain the different notions involved in hypothesis testing, we take the example of the tossing of a coin. The hypothesis to be tested is that, when tossed, the coin is as likely to show *head* as *tail*. In other words, we want to check whether the coin is fair. This hypothesis is formally referred to as the *null hypothesis*, H_0, and may be written as $p(heads) = 0.5$, where $p(heads)$ is the probability that the coin shows head. The successive results of the coin tossing experiment may be seen as a set of random variables X_i, $i \in [1, 2, ...]$, where X_i is set to x_i of value one if the i^{th} experiment results in head and to x_i of value zero if the result is tail. The null hypothesis is tested against the *alternative hypothesis*, H_1, which states that the coin is not fair, i.e. $p(heads) \neq 0.5$. In summary

$$\begin{cases} H_0 : & p(heads) = 0.5, \quad (\text{ the coin is fair}) , \\ H_1 : & p(heads) \neq 0.5, \quad (\text{ the coin is not fair}) . \end{cases} \tag{5.1}$$

The hypothesis testing procedure aims at defining what is the probability that H_0 is true. To that end, a *test statistic*, Q, has to be chosen. The test statistic is a function of the experimental results X_i, $i \in [1, 2, ...]$, and may be written as $Q = g(X_1, X_2, ...)$. Being a function of random variables, the test statistic Q is itself a random variable. In the coin example, Q may be chosen as being the frequency with which the coin shows head. The hypothesis testing may be rewritten as

$$\begin{cases} H_0 : & Q = 0.5 , \\ H_1 : & Q \neq 0.5 . \end{cases} \tag{5.2}$$

The actual hypothesis testing is carried out in two successive steps. In the first step, the samples x_i, $i \in [1, 2, ...]$, are collected. They permit to compute the value q, $q = g(x_1, x_2, ...)$, which is the realization of the test statistic Q. In the second step, the actual hypothesis testing is performed. In our example, it is written as follows

$$\begin{cases} \text{if } |q - 0.5| \leq T, & \text{accept } H_0 , \\ \text{if } |q - 0.5| > T, & \text{reject } H_0 , \end{cases} \tag{5.3}$$

where T is a predefined threshold.

When checking a hypothesis, the result strongly depends on the choice of the threshold T. As may be seen from Eq. (5.3), T indeed defines the zone within which the hypothesis is accepted. However, the performance of the test hypothesis should be monitored through more tangible quantities. This is achieved by analyzing the two types of errors that may occur in the context of hypothesis testing, the ultimate goal being to make them as small as possible. The first type of error is denoted the *Type I* error. It occurs when the hypothesis H_0 is rejected even though it is true. The probability of this error is called the *significance level* and is denoted α. In our example of the coin tossing, we have

$$\alpha = P\left(|Q - 0.5| \geq T | H_0 \right) , \tag{5.4}$$

where the notation $P\left(... | H_0 \right)$ indicates that the hypothesis H_0 is true.

The second type of error is denoted the *Type II* error. It occurs when the hypothesis H_0 is accepted even though it is false. The probability of this error is a function of Q. It is usually written as $\beta(Q)$ and is called the *operating characteristic* of the test. In our example of the coin tossing, we have

$$\beta(Q) = P\left(|Q - 0.5| \leq T | H_1\right) , \tag{5.5}$$

where the notation $P(...|H_1)$ indicates that the hypothesis H_1 is true.

The optimization of the hypothesis testing involves reducing both the *Type I* and *Type II* errors. However, this optimization may be complex as a reduction of the *Type I* errors results in an increase of the *Type II* errors. Furthermore, it may happen that only an analytical expression for the significance level α may be derived, while this is impossible for the operating characteristic $\beta(Q)$. The solution to these problems is to focus the attention on the significance level. The performance of the hypothesis testing is completely defined in terms of the *Type I* error. By setting the significance level α, the corresponding threshold $T(\alpha)$ (see Eq. (5.3)) is determined. It is indeed the α percentile of the random variable Q. Consequently, the hypothesis is accepted only if the measure q is smaller or equal to $T(\alpha)$. This is equivalent to saying that q should imply a higher or equal probability of *Type I* errors than the significance level. Such a procedure is coherent with the requirements of most applications where the significance level is the important factor to be monitored [114].

Hypothesis testing is a powerful tool. Relying on a statistical approach, it checks whether a hypothesis is valid. In practice, the procedure involves three fundamental choices. First, the null hypothesis, H_0, and the alternative hypothesis, H_1, must be clearly stated. Second, one must select which data $\{X_1, X_2, ...\}$ are used to check the hypothesis. A statistical model for the data must also be chosen. Third, the test statistic Q, $Q = g(X_1, X_2, ...)$, must be defined.

5.3.3 The spatial similarity

As an object is primarily defined through its temporal coherence, the definition of the spatio-temporal similarity should mostly rely on the temporal information. However, one typically expects the regions forming an object to share some spatial characteristics. The spatial information may thus be used as an auxiliary source of information. When deciding whether two regions belong to the same object, it is an important complement to the temporal information.

Spatial information has usually been used through the property of *adjacency* [4, 155, 140, 118]. It states that two regions may be merged only if they are neighbors. This constraint is very natural as most objects are spatially connected. Furthermore, the adjacency requirement simplifies the merging procedure. It indeed diminishes the amount of possible region mergings to be considered. Nevertheless, the adjacency constraint exploits available spatial information poorly. In addition to the adjacency constraint, Thompson [139] also makes use of luminance information. The merging of two regions depends on the luminance contrast existing at their common border.

In this section, a measure of spatial similarity existing between two regions is proposed. Consider two regions, A and B, and denote their spatial similarity by S_{AB}. The notion of spatial similarity should be understood as follows. It is the likelihood that the regions A and B belong to the same object as far as the spatial information is concerned. The spatial similarity ranges from 100% to 0%. The former value corresponds to a perfect spatial match between the two regions, while the latter involves a complete spatial mismatch. The proposed spatial similarity imposes the adjacency constraint. It is set to 0% if the adjacency requirement is not satisfied.

The spatial similarity is derived within the framework of hypothesis testing (see Sec. 5.3.2). Let us

assume that the regions A and B are adjacent. The null hypothesis, H_0, is that, according to the spatial information, regions A and B belong to the same object. The alternative hypothesis, H_1, states that regions A and B do not belong to the same object. The spatial similarity S_{AB} is defined as the significance level of the null hypothesis. The higher the significance level, the higher the likelihood of rejecting H_0 although it is true. This implies that the validity of H_0 is directly proportional to its significance level. It is thus very well suited to serve as definition of the spatial similarity S_{AB}. Formally, the hypothesis testing is written as

$$\begin{cases} H_0: & \text{regions A and B are spatially similar ,} \\ H_1: & \text{regions A and B are spatially not similar .} \end{cases} \tag{5.6}$$

Many types of spatial data may be used to test the hypothesis described by Eq. (5.6). They range from shape information, color, contour length, to texture information. In our case, we base the test on spatial data that are representative of the luminance contrast. The premise is that if a strong contrast exists between two regions, the regions are less likely to belong to the same object. Moreover, the data have to be robust in the presence of noise. In order to fulfill these requirements, the spatial information present on the common border between the regions A and B is used as data. More precisely, the data are the respective medians, l_{AB} and l_{BA}, of their luminance values along their common border. The median is used in order to ensure that the measurements are robust against noise. The measurements l_{AB} and l_{BA} are seen as trials of two random variables L_{AB} and L_{BA}, respectively. Both these random variables are assumed to be modeled by the same Gaussian distribution. The distribution is characterized by a mean μ and a standard deviation σ, i.e. $L_{AB} \sim N(\mu, \sigma)$ and $L_{BA} \sim N(\mu, \sigma)$. In practice, the values of μ and σ are estimated over the ensemble of luminance medians for all the combinations of adjacent regions in the scene. This allows us to introduce the spatial activity existing in the whole image into the evaluation of S_{AB}.

In order to carry out the hypothesis testing, a test statistic, Q_s, is required. We choose Q_s to be the likelihood ratio test. This test is simple and robust. It is defined as the difference between L_{AB} and L_{BA}, $Q_s = L_{AB} - L_{BA}$. This implies that Q_s is a Gaussian random variable with zero mean and a standard deviation equal to $\sqrt{2}\sigma$, i.e. $Q_s \sim N(0, \sqrt{2}\sigma)$. The hypothesis testing defined by Eq. (5.6) is rewritten as

$$\begin{cases} H_0: & Q_s = 0 , \\ H_1: & Q_s \neq 0 . \end{cases} \tag{5.7}$$

The hypothesis testing thus reduces to checking whether the mean of Q_s is zero. Denoting q_s the realization of the test statistic Q_s, $q_s = l_{AB} - l_{BA}$, the spatial similarity S_{AB} is found readily. Defined as the significance level of the hypothesis testing, it is written as

$$S_{AB} = 1.0 - \frac{1}{\sqrt{2\pi}\sigma} \int_{-q_s}^{q_s} e^{-\frac{1}{2\sigma^2} x^2} dx , \tag{5.8}$$

where the fact that $Q_s \sim N(0, \sqrt{2}\sigma)$ is taken into account.

Figure 5.2 gives S_{AB} as a function of the normalized value of the test statistic, $\frac{q_s}{\sigma}$. The spatial similarity sharply falls as soon as the test statistic is different from zero. No threshold phenomenon is present. The spatial similarity becomes negligible when the test statistic reaches approximatively three times the standard deviation σ. This behavior is characteristics of a Gaussian distribution which is assumed as the statistical model.

As the median estimator is used to obtain the measurements, the spatial similarity S_{AB} is characterized by its robustness. Moreover, the value of S_{AB} takes into account the spatial activity existing in the entire scene. This is due to the fact that the standard deviation of Q_s incorporates data from the whole image. Two fairly dissimilar regions X and Y may result in a high value of S_{XY} if their mutual spatial

discrepancy is very small when compared to the rest of the regions in the scene. The spatial similarity is symmetric, $S_{AB} = S_{BA}$, and reflexive, $S_{AA} = 100\%$.

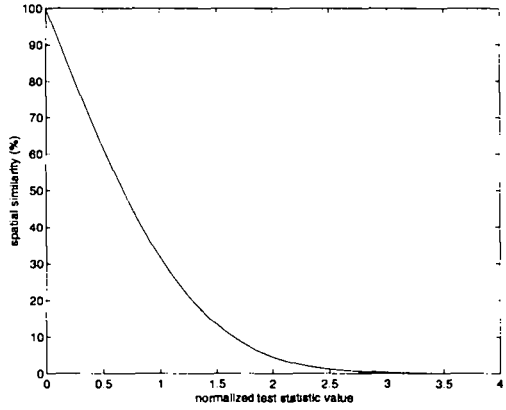

Figure 5.2: The spatial similarity as a function of $\frac{q_s}{\sigma}$.

5.3.4 The temporal similarity

When decomposing a dynamic scene into its constituent objects, the temporal information plays a fundamental role. A moving object is indeed primarily defined through its temporal coherence. This assumption may be negated if the object undergoes deformations [120]. In this case, different temporal evolutions coexist within the object. This situation is however not addressed in the dissertation. As already explained in Sec. 2.3.2, we assume that the object undergoes a rigid body motion. This implies that the ensemble of regions forming an object share the same motion. Thus, the temporal information forms a reliable basis for deciding whether two regions belong to the same object.

In the context of spatio-temporal region merging, the temporal information is naturally represented in the form of a fully parametric motion model $\vec{M}$. In this representation, the motion of a region is modeled by a set of n parameters, n depending on the motion model $\vec{M}$ being chosen. An analysis of the choice of motion model is given in Sec. 2.3.4. Given two regions, their respective parametric motions permit to determine to which extent they share the same motion. This property is referred to as their *temporal similarity*. In the literature, two different approaches may be identified when defining the temporal similarity. One works directly in the motion parameter space, while the other is based on statistics of the residual distributions obtained by motion compensation.

The first class of techniques [44, 149] defines the temporal similarity in the motion parameter space. Given two regions, their similarity is inversely proportional to the distance existing between their respective parametric motion representations. However, this approach is sensitive to noisy motion data. Inaccuracies in the motion estimation may alter the definition of the clusters. Moreover, it suffers from the inherent indeterminacy of the motion parameters [5]. Contingent on the motion model and the scene, similar optical flows may indeed be represented by very different sets of motion parameters.

The second class of techniques uses the parametric motion information in its residual distribution form. The temporal similarity between two regions is obtained as follows. First, the residual distribution resulting from motion compensating the region with its own motion is computed. Second, the residual distribution resulting from motion compensating the region with the motion of the other region is also computed. These two distributions are compared in order to assess the temporal similarity. The similarity is assumed to be proportional to the degree of resemblance between the two distributions. The comparison

is made through a statistic over the residual distributions [4, 133, 118]. However, the use of a single statistic is too drastic a reduction of dimensionality. It may induce wrong merging decisions as all the information present in the distribution is not examined.

In this section, a measure of temporal similarity existing between two regions is proposed. The temporal similarity of the region A with respect to the region B, is denoted T_{AB}. This quantity represents the likelihood that the region A belongs to the same object as the region B, as far as the temporal information is concerned. The temporal similarity typically ranges from 100% to 0%. The former value corresponds to a perfect temporal match, while the latter involves a complete temporal mismatch. Similar to the spatial similarity (see Sec. 5.3.3), the property of adjacency is also imposed. The temporal similarity is only computed for neighboring regions. It is set to 0% if the adjacency requirement is not satisfied.

The temporal similarity is derived within the framework of hypothesis testing. Assume that the regions A and B are adjacent. The null hypothesis, H_0, is that the region A moves in the same way as the region B. Clearly, the alternative hypothesis, H_1, states that the region A does not undergo the same motion as the region B. The temporal similarity T_{AB} is defined as the significance level of the null hypothesis. The higher the significance level, the higher the likelihood of rejecting H_0 although it is true. This implies that the validity of H_0 is directly proportional to its significance level. It is thus very well suited for defining the temporal similarity T_{AB}. Formally, the hypothesis testing is written as

$$\begin{cases} H_0: & \text{the region A is temporally similar with the region B ,} \\ H_1: & \text{the region A is temporally not similar with the region B .} \end{cases} \tag{5.9}$$

In order to carry out the hypothesis testing described by Eq. (5.9), all the available temporal information should be exploited. Therefore, the hypothesis testing should use the information existing in both the motion parametric representation and the residual distributions. Although the two types of motion information derive from the motion parameters, they are very different in nature and thus mutually complementary. The residual distribution is indeed representative of the mismatch existing between the parametric motion and the real optical flow. Consequently, it encompasses the motion information complementary to that available in the motion parametric representation. Moreover, the hypothesis testing has to take into account the likely outliers. Based on this observations, the test statistic denoted as the Modified Kolmogorov-Smirnov (MKS) test statistic, Q_{MKS}, is proposed.

The test statistic Q_{MKS} uses both motion parameters and the residual distributions as data. However, the former are seen as less reliable as the latter. The indetermination in the motion parameters definition may dangerously alter the hypothesis testing. Conversely, the residual distribution directly arises from the discrepancies existing between the estimated region motion and its real optical flow. Assuming that a given optical flow is modeled by two very different sets of motion parameters, the comparison of the residual distributions nevertheless detects that they approximate the same motion. The residual distribution therefore forms the backbone the test statistic Q_{MKS}.

Before explicitly formulating Q_{MKS}, some definitions are necessary. Consider the regions A and B with their respective motion parameters $\vec{M}_A$ and $\vec{M}_B$. The residual distribution h_1 is obtained by applying $\vec{M}_A$ on A, while the residual distribution h_2 is obtained when A is motion compensated with the motion parameters $\vec{M}_B$. Based on these definitions, the realization q_{MKS} of the test statistic Q_{MKS} is expressed as

$$q_{MKS} = \lambda \max_x |F_1(x) - F_2(x)| \ , \tag{5.10}$$

where

$$\begin{cases} F_1(x) &= \int_{-\infty}^{x} w(x) h_1(x) dx \ , \\ F_2(x) &= \int_{-\infty}^{x} w(x) h_2(x) dx \ , \end{cases} \tag{5.11}$$

are the weighted cumulative distributions, respectively, for the residual distributions h_1 and h_2. The *weighting function* $w(x)$ is a positive $\mathcal{L}_2$ function such as $w(x) \in [0,1]$, $\forall x$. The *pondering factor* λ is positive and real-valued, $\lambda \in [0,1]$.

Deriving from the well-known Kolmogorov-Smirnov test [114], the Q_{MKS} is a nonparametric test statistic which uses the residual distribution as a whole to carry out the hypothesis testing. The underlying idea is to measure the maximum discrepancy existing between the two cumulative distributions $F_1(x)$ and $F_2(x)$. Due to its nonparametric nature, the test statistic Q_{MKS} does not require any predefined model of the distributions, thus offering large flexibility and robustness. It simply tries to decide whether the two distributions it compares are drawn from the same (unknown) random process [101]. This is in sharp contrast to techniques which reduce the distribution information to a single statistic. Not only does the use of a single statistic restrain the amount of information being exploited in the testing procedure, but it also runs the risk of spoiling the testing through the choice of an inappropriate statistic.

Compared to the usual Kolmogorov-Smirnov test, the proposed test statistic Q_{MKS} differs in two ways. The modifications aim at rendering the nonparametric hypothesis testing more robust through efficient use of all the available information. The first modification is expressed by the *weighting function* $w(x)$. It aims at tackling the problems of outliers. More precisely, it takes into account the use of robust motion estimators. We will discuss it in greater detail in Sec. 5.3.5. The second modification corresponds to the *pondering factor* λ. It supplies to the test the motion information existing in the motion parametric representation. This factor directly affects the discrepancy found between the cumulative distributions $F_1(x)$ and $F_2(x)$. It is presented in Sec. 5.3.6.

The test statistic Q_{MKS} permits to rewrite the hypothesis testing on the temporal similarity. Equation (5.9) is reformulated as

$$\begin{cases} H_0: & Q_{MKS} = 0\,, \\ H_1: & Q_{MKS} > 0\,. \end{cases} \tag{5.12}$$

As already explained, the temporal similarity T_{AB} is defined as the significance level of the hypothesis testing. It is computed using the relationship derived for the standard Kolmogorov-Smirnov test. The temporal similarity T_{AB} is a function of both the test statistic value, q_{MKS}, and $Area(A)$, the area of the region A. According to [129, 121], it is given by

$$T_{AB} = 2\sum_{j=1}^{\infty}\left((-1)^{j-1}e^{-2j^2\,\delta^2}\right)\,, \tag{5.13}$$

where $\delta = \left(\sqrt{Area(A)} + 0.12 + \frac{0.11}{\sqrt{Area(A)}}\right) q_{MKS}$.

Figure 5.3 shows examples of the temporal similarity T_{AB} as a function of the test statistic value q_{MKS} for different $Area(A)$. The typical curve features a sharp transition which moves further away from the origin as the value of $Area(A)$ decreases. For a given q_{MKS} value, the larger the value of $Area(A)$, the lower the temporal similarity T_{AB} . This behavior reflects the statistical property that two distributions, if equal, should see their discrepancies diminish as the number of trials (i.e. $Area(A)$) increases. Furthermore, it must be underlined that the temporal similarity as defined by Eq. (5.13) is not symmetric. In other words, $T_{AB} \neq T_{BA}$. This property of directionality proves fundamental when carrying out the region merging procedure. Finally, the similarity is reflexive, i.e. $T_{AA} = 100\%$.

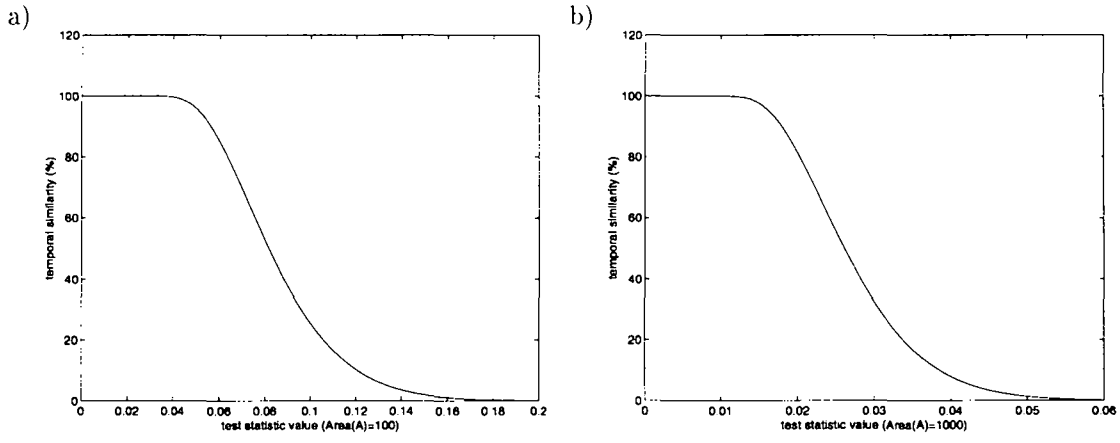

Figure 5.3: The temporal similarity as a function of the q_{MKS} test statistic value with respectively **(a)** $Area(A) = 100$ and **(b)** $Area(A) = 1000$. Note that the scales of the two plots are different.

5.3.5 The weighting function $w(x)$

When performing the hypothesis testing about the temporal similarity, the characteristics of the motion estimation procedure have to be considered. In particular, the test statistic Q_{MKS} has to be robust against outliers and has to take into account the use of robust motion estimators (see Sec. 2.4.4). This task is accomplished through the weighting function $w(x)$ (see Eq. (5.11)). Its purpose is to give a different importance to discrepancies between the distributions depending on the value x of the residue. In particular, the weighting function $w(x)$ allows to render the hypothesis testing procedure coherent with a robust motion estimation procedure. Indeed, robust motion estimators do not attribute the same importance to the different residues. The residues close to zero are typically given more weight than larger residues in order to decrease the influence of outliers in the estimation process. Consider the case where a M-estimator $\rho(x)$ is used. The relative importance of the residue, x, is determined by the value of the weight function $\frac{\dot{\rho}(x)}{x}$ [124]. The weighting function, $w(x)$, is the normalized version, in the range [0,1], of $\frac{\dot{\rho}(x)}{x}$. For instance, in case the motion estimation is performed with the Geman-McClure estimator [42] $\rho_{GM}(x)$, where

$$\rho_{GM}(x) = \frac{\frac{x^2}{\sigma^2}}{1+\frac{x^2}{\sigma^2}}, \tag{5.14}$$

with σ denoting the scale factor. The corresponding weighting function $w_{GM}(x)$ is

$$w_{GM}(x) = \frac{1}{(1+\frac{x^2}{\sigma^2})^2}. \tag{5.15}$$

Clearly, the function $w_{GM}(x)$ favors low residues, reaching the maximum weight of unity for the null residue. As the residue becomes larger, its influence gradually decreases until it becomes totally insignificant with a weight close to zero.

5.3.6 The pondering factor λ

The second difference between the usual Kolmogorov-Smirnov test and Q_{MKS} lies in the use of the pondering factor λ (see Eq. (5.10)). The pondering factor λ allows to gather the motion information

existing in the motion parametric representation and to supply it to the Q_{MKS} test statistic. Its role is to ponder the discrepancy found between the two weighted cumulative distributions, $F_1(x)$ and $F_2(x)$. In doing so, all the available motion information is used to robustly check the hypothesis of temporal similarity.

The definition of the pondering factor λ relies on the distance existing between the n motion parameters of the region A, $\vec{M}_A$, and the ones of region B, $\vec{M}_B$. More precisely, λ is derived from the significance level α_{param} of the hypothesis that the parameter vectors $\vec{M}_A$ and $\vec{M}_B$ are the same. For a given α_{param}, the factor λ is intuitively defined as

$$\lambda = 1 - \alpha_{param} \ , \tag{5.16}$$

with $\alpha_{param} \in [0, 1]$.

Depending on the value of α_{param}, the influence of the pondering factor λ varies. Let us examine the two limiting situations. In case $\vec{M}_A$ is similar to $\vec{M}_B$, α_{param} is set close to unity, entailing that λ has a value close to zero. According to Eq. (5.10), the resulting value q_{MKS} of the test statistic Q_{MKS} is decreased significantly, whatever the discrepancies present between the residual distributions. Thus, the temporal similarity T_{AB} increases considerably. In case $\vec{M}_A$ and $\vec{M}_B$ are very dissimilar, α_{param} is set close to zero. The pondering factor λ being close to unity, it has nearly no influence on the value q_{MKS} of the test statistic Q_{MKS}. The determination of temporal similarity T_{AB} completely relies on the information existing in the residual distributions.

With regard to the derivation of α_{param}, we use a procedure similar to the hypothesis testing proposed in Sec. 5.3.3. Each motion parameter $\vec{M}(i)$, $i = \{1, ..., n\}$, is modeled as a Gaussian random variable with mean $\mu(i)$ and standard deviation $\sigma(i)$. The corresponding test statistic $Q(i)$ is chosen to be the likelihood ratio test. It is defined as the difference between $\vec{M}_A(i)$ and $\vec{M}_B(i)$. The test statistic $Q(i)$ is a random variable such as $Q(i) \sim N(0, \sqrt{2}\sigma(i))$. Consequently, the significance level α_i is

$$\alpha_i = 1.0 - \frac{1}{\sqrt{2\pi}\sigma(i)} \int_{-q(i)}^{q(i)} e^{-\frac{1}{2\sigma(i)^2} x^2} dx \ , \tag{5.17}$$

where $q(i)$ is set to $\vec{M}_A(i) - \vec{M}_B(i)$. In practice, the standard deviation, $\sigma(i)$, is estimated over the population composed of the parameters, $\vec{M}(i)$, of all the regions in the frame.

Starting from the different α_i, $i = \{1, ..., n\}$, the significance level α_{param} is defined. It aims at robustly extracting the information existing in the n parameters of the motion model $\vec{M}$. To that end, a very conservative and cautious approach is adopted. The significance level α_{param} is expressed as

$$\alpha_{param} = \min_i \{\alpha_i, i = 1, ..., n\} \ . \tag{5.18}$$

The term α_{param} is chosen as being the minimum among the ensemble of α_i, $i = \{1, ..., n\}$. In doing so, the hypothesis that $\vec{M}_A(i)$ and $\vec{M}_B(i)$ are equal is conservatively checked.

The pondering factor λ allows to gather the motion information existing in the motion parametric representation and to inject it in the Q_{MKS} test statistic. The definition of the pondering factor λ being very conservative, the parametric information is only utilized when it indisputably shows strong evidence that it is reliable.

5.3.7 Defining the spatio-temporal similarity

The moving objects forming the scene are characterized through the temporal coherence and, to a lesser point, through their spatial homogeneity. In spite of sharing a similar motion, the regions constituting

an object may indeed be very different spatially. Bottom-up techniques should exploit all the available information to decide which regions should be merged together. To that end, a measure of the spatio-temporal similarity existing between the regions is defined. The way it is defined as well as the information it incorporates are now explained.

Many techniques have been proposed for segmenting out the objects in a video sequence. Some techniques only rely on one type of information, either temporal [44, 155] or spatial [128, 115]. However, a robust definition of the similarity necessitates the use of all the available information, i.e. both spatial and temporal information. The best way of combining both types of information remains however an open question. In this dissertation, the spatio-temporal similarity, $Sim(AB)$, of two regions A and B is specified as a combination of the temporal similarity T_{AB} and the spatial similarity S_{AB}. The former is defined in Sec. 5.3.4, while the latter is presented in Sec. 5.3.3. The spatio-temporal similarity $Sim(AB)$ must be understood as the likelihood that the region A belongs to the same object as the region B. The likelihood derives from both temporal and spatial information. Recall, however, that an object is likely to be composed of regions having different spatial characteristics. Thus, $Sim(AB)$ must mainly rely on the temporal information,(i.e. T_{AB}). For any region X, let Υ_X be the ensemble of its adjacent regions. The proposed spatio-temporal similarity measure $Sim(AB)$ is written as follows

$$Sim(AB) = T_{AB} - f_L\, T_{AB}\ (Max\ -\ S_{AB})\ , \tag{5.19}$$

with $f_L \in [0, 1]$, and,

$$\begin{aligned} Max &= \max(S_{AE}, S_{EF})\ , \\ S_{AE} &= \max_{I \in \Upsilon_A}(S_{AI})\ , \\ S_{EF} &= \max_{I \in \Upsilon_E}(S_{EI})\ . \end{aligned}$$

Equation (5.19) reflects the fact that T_{AB} is the most significant term in the spatio-temporal similarity $Sim(AB)$. S_{AB} is just used as a corrective factor. The level of the correction is controlled through the factor f_L, referred to as the *luminance factor*. Let us consider the limiting cases. If the luminance factor f_L is set to zero, the spatial information S_{AB} is not used. The spatio-temporal similarity $Sim(AB)$ is equal to the temporal similarity T_{AB}. Conversely, the correction induced by the spatial information S_{AB} attains its maximum value when the luminance factor f_L is set to unity. At most, the spatial information S_{AB} may completely wipe out the temporal information T_{AB}, resulting in $Sim(AB)$ being set to zero. However, the magnitude of this maximum correction is a function of the factor Max. It conveys information about the general spatial coherence of region A with its neighboring regions. The factor Max is closely linked to the spatial activity existing around the region A. Figure 5.4 illustrates the derivation of the factor Max. In the first step, we determine the region E adjacent to A, $E \in \Upsilon_A$, which is spatially the most similar to the region A. Next, we determine the region F adjacent to E, $F \in \Upsilon_E$, which is spatially the most similar to the region E. The factor Max is defined as the maximum between the spatial similarities S_{AE} and S_{EF}. In case the factor Max is small (e.g. close to zero), the region A is by definition lying in an area of high contrast. This implies that the spatial information is not very useful and it should be allowed to only slightly correct T_{AB}. In the case where Max is large (e.g. close to unity), the region A lies in an area where high spatial similarity exists. Consequently, the spatial information S_{AB} should be allowed to play a stronger role in the definition of $Sim(AB)$. This is achieved through the large value of the factor Max as it represents the maximum correction the spatial similarity may make to T_{AB}.

By construction, the spatio-temporal similarity is a robust estimation of how likely the region A is to be merged with region B. This robustness derives from the definition of both T_{AB} and S_{AB}, as well as the definition of $Sim(AB)$ itself. Furthermore, the spatio-temporal similarity is non symmetric, i.e. $Sim(AB) \neq Sim(BA)$. It is also reflexive as $Sim(AA) = 100\%$.

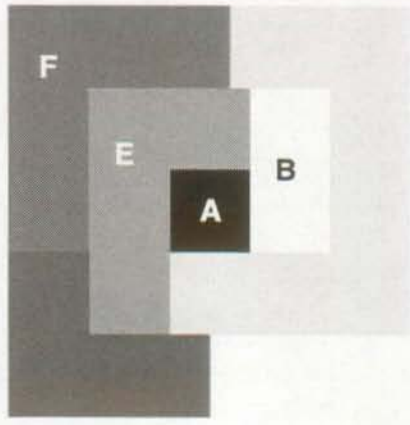

Figure 5.4: Derivation of the factor Max. The figure shows several regions around the region A. The spatial coherence of the region A with its neighborhood is checked. The factor Max reflects the importance of the spatial coherence in the vicinity of the region A. In this example, the region E, which is spatially most similar to the region A, happens to be on the left of region A. In turn, the region F, which is defined to be spatially most similar to the region E, happens to lie on the top-left of region E. In this example and according to Eq.(5.19), the factor Max is taken to be S_{EF}.

5.4 The region merging strategy

5.4.1 Introduction

Spatio-temporal region merging can be seen as a clustering problem: the regions that have similar spatio-temporal characteristics should be clustered together. After defining the spatio-temporal similarity, we must define a strategy to use this information in order to carry out the clustering procedure. In the literature, two types of strategies have been proposed. The first class starts by defining the set of motions which characterized the scene. Then, the ensemble of regions which correspond to each motion is merged to form an object [148, 13, 44]. The second type of merging strategy estimates the set of motions and merges the regions simultaneously. For each region, tentative mergings are typically carried out with its adjacent regions. The merging are accepted in case the region similarity satisfies a predefined criterion [26, 4, 155]. The simultaneous character of the motion determination and the merging procedure allows for a robust definition of the objects. This second type of merging strategy has been chosen in this dissertation.

5.4.2 Graph-based region clustering

When simultaneously determining the motions and the corresponding objects, the information provided by the region similarities should be exploited to its fullest extent. In particular, the region merging issue should be represented in a form which permits to apprehend the natural relationships existing between different groups of regions. To that end, the region merging process may be concisely formulated as a graph-based clustering problem. The graph, G, is built by having each vertex represent one the N_r regions of the ensemble $\mathcal{R} = \{R_1, R_2, ..., R_{N_r}\}$. Every link represents the spatio-temporal similarity between the two vertices it connects. The ensemble of links is denoted $\mathcal{A}$. See [80] for a good introduction to the subject of graph theory. The merging strategy must be defined so as to exploit the structures revealed by the graph representation.

Several strategies have been proposed for graph clustering. They differ mainly in the way the region

to be merged in a given iteration are selected. However, most of the theory for graph-based clustering has been developed under the assumption of a symmetrical similarity measure. That is, the similarity of vertex A to vertex B is assumed to be the same as the similarity of B to A. This implies that the graph has undirected links and each pair of vertices, if connected, are attached by a single link. Thus, in the context of spatio-temporal region merging, most techniques using a graph representation rely on a symmetric graph [155, 118, 133]. Furthermore, their merging strategy is generally limited to the Greedy Merging Algorithm (GMA). This algorithm iteratively merges the two regions showing the strongest similarity until a stop criterion is reached.

The proposed region merging strategy aims at exploiting the graph representation to a fuller extent. Furthermore, the similarity measure defined in Sec. 5.3.7 is not symmetric: $Sim(A,B) \neq Sim(B,A)$. Clearly, the graph is weighted as well as directed. If connected, each pair of vertices of the graph is connected by two directed links. An example of a typical graph G is depicted in Fig. 5.5. The ensemble of regions $\mathcal{R}$ comprises nine regions, $\{R_1, ..., R_9\}$. The links between them define their respective spatio-temporal similarities and are given in percents. More precisely, the spatio-temporal similarity of the region A with the region B is represented with an arrow going from the vertex B to the vertex A. For instance, the spatio-temporal similarity of the region R_2 with the region R_1 is 81%. Recall that the similarity is set to 0% for regions which are not adjacent. These null links as well as the links related to the reflexive property are not represented.

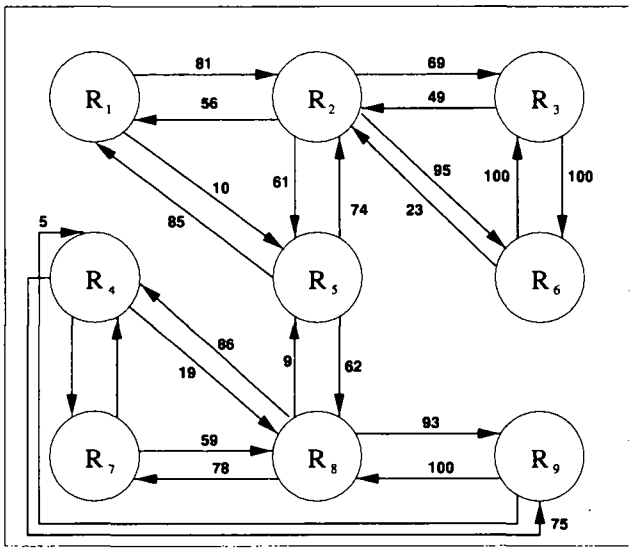

Figure 5.5: Graph representation of the region similarities. The nodes are the regions, while the links represent the spatio-temporal similarities between the regions. The similarities are expressed as a percentage.

The graph G contains all the necessary information to carry out the merging process. In terms of graph theory, such a merging corresponds to extracting the clusters existing in the graph. For this last procedure to take place, graph clustering rules have to be defined. Their derivation is closely linked to the graph at hand and to the problems which may have occured when building it. In that respect, the weighted nature and the directionality of the graph G have important implications. Their combination defines natural structures in G which are directly related to the objects forming the scene. In the example shown in Fig. 5.5, three natural structures may identified at first glance. They are respectively $\{R_1, R_2, R_5\}$, $\{R_3, R_6\}$ and $\{R_4, R_7, R_8, R_9\}$. The graph clustering rules should aim at exploiting this structural information. Furthermore, the derivation of the rules must also take into account the inaccuracies and errors reported in the graph. In our case, a first type of problems stems from regions in $\mathcal{R}$ which have no natural signification. They can either be seen as originating from noise or due to a strong oversegmentation.

Their motion being not well-defined, they may corrupt the set $\mathcal{A}$ of links putting to risk the clustering process. A second type of problems arises from erroneous motion parameters. Faulty links are produced which may lead to a mistaken cluster extraction.

Based on the above considerations, an unsupervised cluster extraction procedure, using two different rules, is proposed. The two rules are respectively denoted as the *strong rule* and the *weak rule*. They are mutually complementary and designed to address the ensemble of different situations which may occur during the merging process. The strong rule focuses on exploiting the natural structures present in the graph. The weak rule assumes that among the regions of $\mathcal{R}$, some are very well-defined, both spatially and temporally. The weak rule uses them as seeds to determine which regions should be merged together. In order to decrease the complexity of the task, both rely on a thresholded graph. It is obtained by deciding whether to actually accept or reject the hypothesis of spatio-temporal similarity among the different regions. Given a threshold of acceptance t, the hypothesis that the region A is spatio-temporally similar to the region B is accepted if and only if

$$Sim(A, B) \geq t \ . \tag{5.20}$$

In case of acceptance and using the arrow notation as in Fig. 5.5, we write $(B \rightarrow A)$. The notation $(B \rightarrow A)$ should be understood as a binary link relationship between regions A and B, i.e. *true* (or 1) if the link exists, and *false* (or 0) otherwise. Figure 5.6 gives an example of a thresholded graph. This has been generated by setting $t = 60\%$ on the weighted graph given in Fig. 5.5.

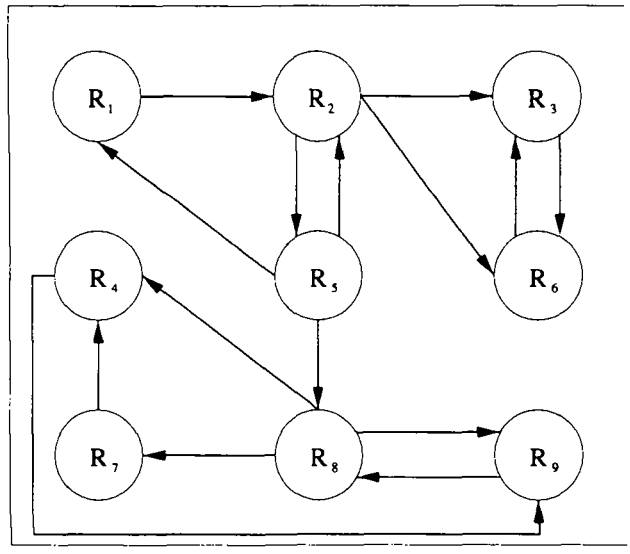

Figure 5.6: Example of a thresholded graph. The relationships between the nodes are binary valued. Either they exist or they do not.

The strong rule and the weak rule are now explained. In the remaining, the notation is as follows. N_r denotes the number of regions R_i in $\mathcal{R}$, $i \in [1, ..., N_r]$ to be merged. Through clustering, the regions are distributed in clusters C_i, $i \in [1, ..., N_c]$, $N_c \leq N_r$. The number of clusters N_c is automatically defined by the clustering rules.

5.4.3 The strong rule

The graph G is built from the ensembles $\mathcal{R}$ and $\mathcal{A}$. This implies that the presence of inaccuracies or errors in these ensembles may severely influence a graph based region merging approach. The graph based

clustering procedure should thus concentrate on the secure pieces of information. In particular, natural structures existing in the graph should be used. The strong rule exploits such structural properties in the graph. For each region in the cluster, there must be a closed path, or *cycle* via some other regions in the cluster. More precisely, the strong rule clusters regions together only if they belong to the same cycle.

The strong rule operates on a binary graph. In the case of a weighted graph, this is thresholded with a predefined threshold, t_{sr}. The cycle condition imposed by the strong rule defines the clusters C_i, $i \in [1, ..., N_c]$ as follows

$$C_i = \{R_j \mid \exists (R_k, R_l \in C_i, k \neq j, l \neq j) \text{ such that } (R_j \rightarrow R_l \text{ and } R_k \rightarrow R_j)\} , \quad (5.21)$$

with $R_j, R_k, R_l \in \mathcal{R}$. The notation defined in Sec. 5.4.2 for a thresholded graph is used.

By looking for cycles in the graph, the strong rule focuses on the secure information, thus avoiding the problems linked with doubtful regions or erroneous motion parameters. The cycle ensures that each region in the cluster is spatio-temporally well-suited with respect to at least one other region of the cluster. This discards the regions for which the motion parameters have been erroneously estimated. Also, the cycle requirement excludes the doubtful regions for which the motion is not well defined. In addition, the cycle condition imposes that, for each region in the cluster, an other region in the cluster which is spatio-temporally well-suited with respect to the region of interest is found. In this way, only regions with similar motions are put in the same cluster. If a region is not assigned to any of the clusters defined by Eq. (5.21), it is regarded as a separate cluster.

The strong rule imposes severe constraints on the clustering process. Only regions whose motion is well defined as well as correctly estimated are considered. Also, the clusters are only formed by regions which are very likely to belong to the same object. The severity of this constraint results in having two types of cluster after applying the strong rule. The first type consists of clusters which group several very well-defined regions. These clusters foreshadow the objects forming the scene. The second type of clusters are typically formed of single regions which are either oddly defined or for which the motion was erroneously estimated.

5.4.4 The weak rule

The weak rule generalizes the greedy merging algorithm [118]. First, it aims at exploiting the strongest links existing in the graph. Second, it uses a hierarchical approach to cluster the regions. This is achieved by thresholding the graph with successively lower thresholds until the lowest predefined threshold t_{wr} is reached. At each iteration, the current clustering relies on the clusters found in the previous iteration. The weak rule is very well suited to cases where badly defined or small regions coexist with regions having a strong semantic significance. The weak rule permits to merge the former regions with the most appropriate of the latter ones.

The weak rule carries out the clustering process in an iterative way. Starting with clusters each containing exactly one region, these clusters are then iteratively merged together. To that end, the weak rule relies on a hierarchical thresholding of the graph G. The threshold starts at 100% and decreases, by a fixed predetermined step, to its lowest allowed value, t_{wr}. At each iteration, the clusters previously defined serve as the basis for the current clustering process. This implies that the number of clusters N_c, initially equal to the number of regions N_r, decreases progressively as the hierarchical clustering progresses. The first advantage of this hierarchical approach is that the merging procedure deals with clusters of regions. This permits to use structural properties of the graph. This is in contrast with the greedy merging algorithm which only deals with single regions. The second advantage of the hierarchical approach is

that it permits to robustly carry out a non-dynamic merging. Although clusters of regions are created, there is no need to actually merge them and update the graph.

At each iteration of the hierarchical approach, the weak rule determines which cluster should be merged together. This is performed in two successive steps. For each cluster C_m, $m \in [1, ..., N_c]$, the weak rule first determines the ensemble Ω of clusters C_i, $i \in [1, ..., N_c]$ and $i \neq m$, with which C_m could be merged. This selection is made using the directionality of the graph. For each cluster C_i, the total number of its links heading towards C_m are counted. Then, this total number is compared to the number of regions present in C_m. The ensemble Ω is thus defined by

$$\Omega = \left\{ C_i, i \neq m \;\middle|\; \sum_{R_k \in C_i} \sum_{R_l \in C_m} (R_k \rightarrow R_l) \geq \mathrm{Card}(C_m) \right\} , \tag{5.22}$$

where $\mathrm{Card}(C_m)$ denotes the cardinality of the cluster C_m, and $R_k, R_l \in \mathcal{R}$. The notation defined in Sec. 5.4.2 for a thresholded graph is used.

Three cases are possible.

1. The set Ω may be empty. In this case the cluster C_m is left untouched.
2. The set Ω may contain a single element C_i. In this case, this element is merged with C_m.
3. In case the set Ω has several elements, further tests are required in order to select the cluster $C_s \in \Omega$ with which the cluster C_m should be merged. This selection is carried out according to the following rule

$$C_s = \{\max_{\mu}(\max_{\pi}(\max_{\chi}(C_i \in \Omega\ )))\} , \tag{5.23}$$

where

$$\begin{aligned} \chi &= \sum_{R_k \in C_i} \sum_{R_l \in C_m} (R_k \rightarrow R_l) , \\ \pi &= Area(C_i) , \\ \mu &= \sum_{R_k \in C_i} \sum_{R_l \notin C_i} (R_k \rightarrow R_l) , \end{aligned}$$

where $Area(C_i)$ is the area of the cluster C_i.

According to Eq. (5.23), the selection of the cluster C_s is made in several stages. First, the total number of links, χ, to the cluster C_m is computed for each cluster $C_i \in \Omega$. If one cluster, C_s, has the largest χ value, it is merged with C_m. The cluster C_s is indeed the cluster whose regions are the most spatio-temporally coherent with the regions of the cluster C_m. If two or more elements of Ω obtain the same total number of links χ, a further selection is made on the basis of the cluster area, π. The selected cluster C_s is the one with the largest area. This selection is based on the observation that the likelihood of having erroneous motion estimation decreases with the area. For the same value of χ, the cluster with the largest area should therefore be chosen as the cluster C_s. Finally, two or more clusters in Ω may have the same χ and π values. This problem is solved by examining the total number of links to the outside, μ, arising from the clusters, the selected cluster C_s being the one with the largest value of μ. The motivation for such a choice comes from a very simple reason. As a cluster receives new regions, it tends to isolate itself from the other clusters. In other words, when the representative cluster of an object has gathered all the regions forming this object, the cluster mostly has internal links, with only a few external erroneous links

remaining. Conversely, a cluster, which has not yet collected all the regions corresponding to the object, tends to have many links pointing to other clusters. Such clusters are favored by the selection based on the μ value.

5.4.5 The region merging strategy

In order to carry out the region merging procedure, a strategy has to be defined. In particular, the way in which the strong and the weak rules are combined must be determined. Furthermore, one must decide which strategy to adopt with regard to the generation of the successive thresholding of the graph. In addition, the graph must be updated so as to take advantage of new information coming from the merging procedure.

An overview of the adopted region merging strategy is given in Fig. 5.7. The strong rule is applied first. Indeed, we consider as prime information the natural structures existing in the graph. Most of the regions resulting from the strong rule are reasonable first approximations of the objects present in the scene. Besides them, the other regions are usually the regions whose motions have been erroneously estimated and the regions whose motions are not well defined. After the strong rule, we need a rule which merges these last regions with the regions which foreshadow the different objects. This is a perfect role for the weak rule. It relaxes the merging conditions imposed by the strong rule. Regions whose motion is either badly or erroneously defined have the opportunity to be merged with other regions. For the graph shown in Fig. 5.6, the two clustering rules act as follows. The strong rule defines the three clusters $\{R_1, R_2, R_5\}$, $\{R_3, R_6\}$ and $\{R_4, R_7, R_8, R_9\}$. Using the weak rule, the clustering procedure results in the clusters $\{R_1, R_2, R_3, R_5, R_6\}$ and $\{R_4, R_7, R_8, R_9\}$. The strong rule and the weak rule thus perform very different, complementary tasks. Their combination ensures a stable and robust merging of the regions in order to build the objects constituting the scene.

For both the strong and the weak rules, the merging process is carried out iteratively in the context of a dynamic graph updating strategy. At each iteration, the graph is first thresholded, then the region merging takes place followed by the graph update. Initially set to 100%, the threshold value for the strong rule, t_{sr}, is recomputed after each iteration. To that end, the maximum value E_s that would still allow the strong rule to carry out a merging, is determined among the links in the graph. The threshold t_{sr} is thus defined as

$$t_{sr} = Int(E_s) - D_{sr} , \tag{5.24}$$

where D_{sr} is a predefined step-size. The iteration stops when t_{sr} is less than the predefined lowest threshold threshold t_{lsr}. After each merging, the graph representing the relationships among regions is updated by recomputing the temporal and spatial characteristics of the newly created regions, and then recomputing the similarities among the current set of regions

The same dynamic merging strategy is used when applying the weak rule. The graph is updated after each iteration. Similar to t_{sr}, the threshold t_{wr} to threshold the graph is recomputed at every iteration as follows

$$t_{wr} = Int(E_w) - D_{wr} , \tag{5.25}$$

where D_{wr} is a predefined step-size for lowering t_{wr}, and E_w is the maximum value among the links in the graph that would still allow the weak rule to carry out a merging. The merging process stops when t_{wr} is lower than the predefined lowest threshold t_{lwr}. Remember that the weak rule is built so as to also carry out a non-dynamic merging procedure. As described in Sec. 5.4.4, this type of merging occurs within each iteration of the dynamic graph update strategy.

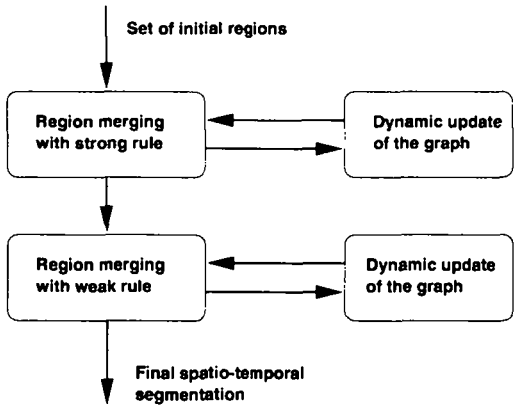

Figure 5.7: The region merging strategy. The set of initial regions is assumed to be provided. First, the strong rule serves to carry out the region merging, which is, in turn, used to update the graph. This procedure is iteratively carried out until no further merging occurs. At this stage, the weak rule is applied. The same strategy as for the strong rule is used.

The proposed graph clustering strategy takes into account the specificities of the problem at hand. It is able to efficiently exploit the information represented in the graph, while being robust to erroneous motions as well as to oddly defined regions. Furthermore, the graph is refined iteratively as the clustering proceeds. The proposed graph clustering strategy permits to robustly define the objects forming the scene in an unsupervised way.

5.5 Simulation results

5.5.1 Introduction

The proposed technique for spatio-temporal segmentation is independent of the way in which the set of initial regions is generated. In order to illustrate this, the proposed technique for spatio-temporal segmentation has been tested with two different ways of generating the set of initial regions. In one experiment, the set of initial regions is obtained through a quadtree based segmentation. The regions are temporally homogeneous, but their blocky shape does not take into account the spatial characteristics of the scene. In the other case, the set of initial regions is obtained using the method presented in Chapter 4. By construction, the regions are arbitrarily shaped. Not only are these regions temporally homogeneous, but they also reflect the spatial traits of the scene.

Results are presented for four video sequences: "Akiyo", "Table Tennis", "Foreman" and "Bream". The details of these video sequences are given in Appendix A. In all the simulations, the noise existing in the residual distributions is removed by Gaussian low-pass filtering. The filter width is heuristically set to 0.4σ where σ is the standard deviation of the error.

5.5.2 Results using quadtree based initial regions

The quadtree based generation of the set of initial regions uniquely relies on the temporal information. Initially the current frame is divided into square blocks of equal size, and their motion parameters are estimated. Blocks that show a DFD energy higher than a preset threshold are split into four equal sized

blocks. This procedure is iterated in order to get a recursive decomposition of the frame in form of a quadtree. This process is repeated until a lower limit on the size of blocks is reached.

The motion model is chosen to be the affine model similar to that of Sec. 4.4.2, while the motion estimator is the MAD. Let us however stress that, here, no global motion estimation and compensation is performed. Consequently, the local motion estimation encompasses the movement due to the camera motion. The estimation of the parameters of the affine motion model is performed through a matching technique. The computational complexity is reduced by building a Gaussian pyramid of the input images. This allows a non-exhaustive search while avoiding local minima. The final motion parameters at one level of the pyramid propagate as initial estimates onto the next level. Finally, a deterministic relaxation scheme is applied during the estimation procedure to avoid local minima. The sets of motion parameters obtained for neighboring regions are compared, the one providing the lowest prediction error being selected.

In order to perform the spatio-temporal segmentation, the weighting function $w(x)$ must be determined (see Sec. 5.3.5). Based on the weight function of the MAD motion estimator, we have

$$w(x) = \left|\frac{1}{x}\right| . \tag{5.26}$$

However, such a weighting function can not be used due to the discontinuity it presents at $x = 0$. We avoid this problem by defining $w(x)$ as follows

$$w(x) = \frac{1}{1+|x|} . \tag{5.27}$$

Figure 5.8 shows the simulation results obtained on a typical frame of the video sequence "Akiyo". When the current frame is compared to the previous frame, one may notice a slight movement of the head as the woman is rotating it upwards. This is accompanied by an upward motion of the whole body. The set of initial regions contains 267 regions. After applying the strong rule, the number of regions is reduced to 12. Among these regions, one may clearly distinguish between two types of regions. The first type of regions are those which are the premises of the objects composing the scene. For instance, a region which covers most of the woman is noticeable. The second type of regions are those which have no real semantic significance as the small region found on the cheek of the woman. When applying the weak rule, most of the regions of the second type are merged into those of the first type. Finally, the scene is decomposed into 4 objects: the background, the body with the face, the hair and a part of the right shoulder. The fact that the hair is considered a distinct object may be explained by the characteristics of the affine motion model. As explained in Sec. 2.3.3, the affine model describes the motion of a planar surface under orthographic projection. The face and the body of the woman are thus modeled through a single planar surface, while the curvature of the top the head results in an other planar surface. This leads to the creation of the object "hair". Finally, the object corresponding to a part of the right shoulder is an artifact arising from the set of initial regions. Among these initial regions, some are temporally inhomogeneous as they cover different objects of the scene. This results in having regions which encompass multiple motions and, which are left alone by the proposed spatio-temporal region merging.

Figure 5.9 shows the results of the proposed method for a typical frame of the video sequence "Table Tennis". Compared to the previous frame, the ball in the current frame has moved upwards, while the arm with the bat is slightly displaced downwards. The set of initial regions contains 246 regions. After applying the strong rule, the number of regions is reduced to 25. Again, two types of regions appear. The first type correspond to regions which are the premises of the objects of the scene. An example is the large region covering the arm or the region covering the ball. The second type are regions that have little or no semantic meaning. These latter regions are removed when applying the weak rule. The final set of objects contains three elements: the background, the ball, and the arm with the bat. The second hand is lost at the level of the weak rule. Due to its badly defined motion, it is put into the background.

Figure 5.10 shows the simulation results for a typical frame of the video sequence "Foreman". Compared to the previous frame, the current frame shows that the man has turned his head to his left, while the camera capturing the scene performs a zoom in. The set of initial regions contains 240 regions. Their number is reduced to 29 after applying the strong rule. At this stage, one clearly recognizes the region which is the premise of the background. The other regions mostly cover the face and the helmet of the man. Among these regions, a distinction may be made between those which show natural significance and those which do not. The latter regions disappear after applying the weak rule, resulting in the creation of 6 objects. The objects are the background, the left part of the face with part of the helmet, the lower right part of the face, the upper right part of the face, the hair and the back of the helmet. The reason for the creation of these different objects in the man's face and helmet are linked to the affine motion model. As already explained when commenting the simulation results for the "Akiyo" sequence, the affine model describes the motion of a planar surface under orthographic projection. Confronted to the man's head, the affine model decomposes it into a set of planar surfaces. Indeed, the resulting objects represent the different planar surfaces with which one could approximate the face and the helmet.

Despite the poor quality of the set of initial regions, the proposed technique for spatio-temporal segmentation yields meaningful results. Of course, the objects appear blocky, as the region merging algorithm starts with an initial set of blocky regions. Nevertheless, the initial set of regions may allow a precise definition of the objects composing the scene. For instance, the reader may refer to the results on the sequence "Table Tennis" (see Fig. 5.9). However, it may also happen that some initial regions are temporally and spatially not homogeneous. In that case, the proposed technique for region merging recognizes them. These regions are let alone and are not merged with other regions. Such a case occurs in the simulation results on the sequence "Akiyo" (see Fig. 5.8).

The poor quality of the set of initial regions limits the extent to which the scene may be analyzed. This limitation is especially visible at the level of the object spatial definition. However, the resulting spatio-temporal segmentations are very well suited to distinguish between background and foreground objects. In turn, this distinction enables a robust estimation of the camera motion by restricting the estimation to the background area. This application has been discussed in Sec. 4.4.2.

5.5.3 Results using spatio-temporally homogeneous initial regions

In this section, the initial regions are generated through the top-down technique presented in Chapter 4. By construction, these regions are spatio-temporally homogeneous. They thus define an ideal initial set from which to build the ensemble of the objects constituting the scene. The proposed technique for spatio-temporal segmentation is compared with the technique described by Dufaux *et al.* [44]. Their technique merges the regions into objects through a clustering in the motion parameter space, using the k-medoid clustering algorithm [84]. Notice that this technique is supervised as it requires as input the number of objects present in the scene.

The motion estimation is similar to the one described in Sec. 5.5.2. However, here, the camera motion is compensated for (see Sec. 4.4.2). Only the local motions are used as temporal information when performing the spatio-temporal segmentation. By removing the contribution due to the camera motion, the temporal characteristics of the different objects are indeed enhanced. This results in a more robust spatio-temporal segmentation.

Figure 5.11 shows the simulation results obtained for a typical frame of the video sequence "Akiyo". In order to facilitate comparison, the same frame as the one used in Fig. 5.8 is used. The set of initial regions contains 14 regions. When applying the strong rule, 4 regions are left. The weak rule further merges them,

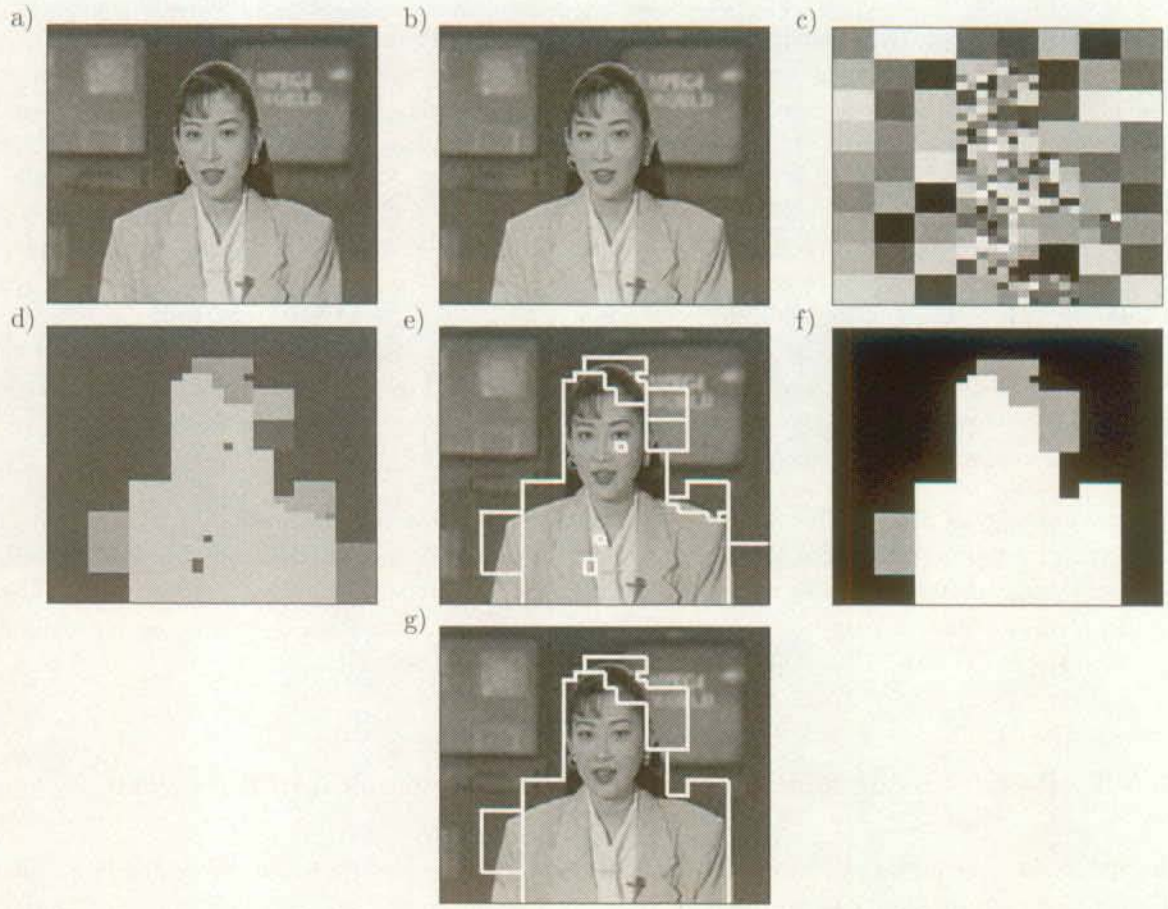

Figure 5.8: "Akiyo": Spatio-temporal segmentation using quadtree based initial regions. **(a)** Previous frame, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame.

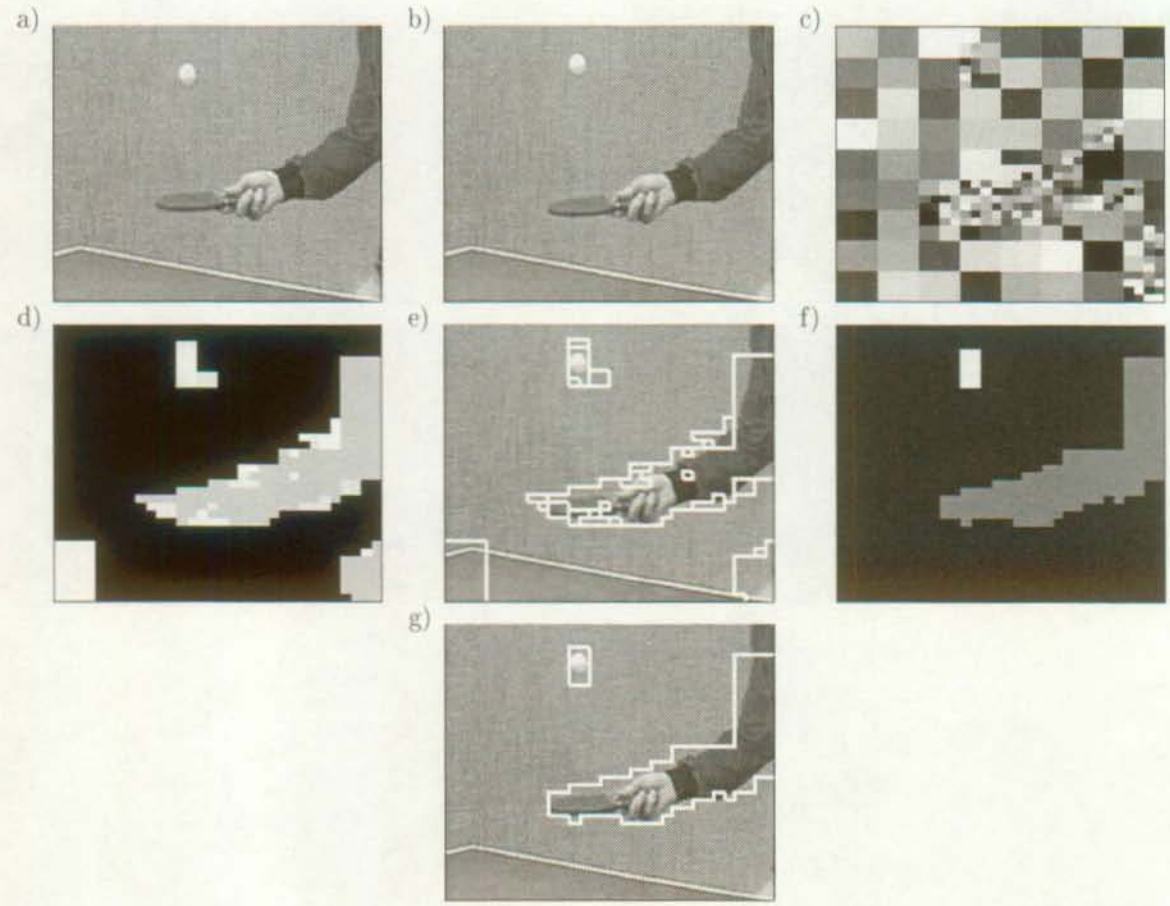

Figure 5.9: "Table Tennis": Spatio-temporal segmentation using quadtree based initial regions. **(a)** Previous frame, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame.

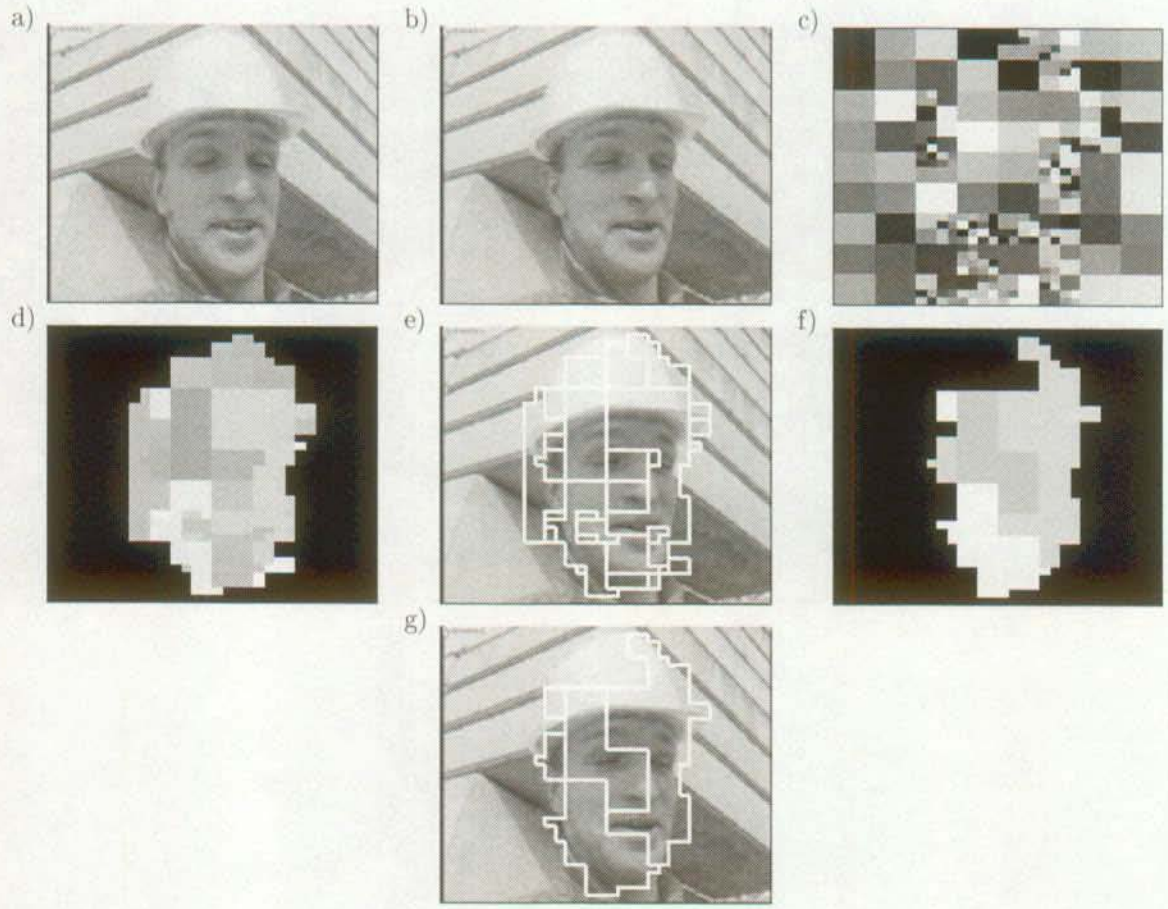

Figure 5.10: "Foreman": Spatio-temporal segmentation using quadtree based initial regions. **(a)** Previous frame, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame.

creating two final objects. These are the background and the woman. Notice the precise spatial definition of their boundaries. Furthermore, the hair is detected as being part of the woman despite the luminance and the motion of this region being very similar to the one of the background. The spatio-temporal segmentation based on the method proposed by Dufaux *et al.* is obtained by requiring two objects. Even with this additional information, the resulting spatio-temporal segmentation is significantly worse that the one derived by the proposed method. This experiment also demonstrates the importance of the initial regions. The spatio-temporally homogeneous initial regions clearly outperform the quadtree based initial regions (see Fig. 5.11 with Fig. 5.8).

In Fig. 5.12, the different stages of the spatio-temporal segmentation for a typical frame of the video sequence "Table Tennis" are shown. The current frame is the same as the one used in Fig. 5.9. The set of initial regions contains 31 regions. These are reduced to 11 when the strong rule is applied. By applying the weak rule, the final spatio-temporal segmentation is obtained. The scene is decomposed into 5 objects: the background, the arm, the ball, the hand with the bat and the second hand. The spatio-temporal segmentation based on the method proposed by Dufaux *et al.* is determined by requiring five objects. However, this additional information does not permit to derive a satisfactory spatio-temporal segmentation. Some parts of the arm are merged with the table, while other parts are merged with the background. Again, the performance of the spatio-temporally homogeneous initial regions compares favorably with the quadtree based initial regions (see Fig. 5.12 and Fig. 5.9).

The spatio-temporal segmentation results for a typical frame of the video sequence "Foreman" are given in Fig. 5.13. The current frame is the same as the one used in Fig. 5.10. The set of initial regions contains 105 regions. After applying the strong rule, the number of remaining regions is 21. Among them, the premises of the background, the face and the helmet are clearly noticeable. By applying the weak rule, the scene is determined to be formed of 5 objects. These are the background, the right part of the face and the neck, the left part of the face, the back of the helmet and, the rest of the helmet. The existence of these different objects modeling the face and the helmet has already been explained when commenting on Fig. 5.10. As the affine model represents the motion of a planar surface under orthographic projection, the spatio-temporal segmentation creates objects which are coherent with this definition. In our case, the face and the helmet are decomposed into objects, each of them representing a region which is roughly planar. The spatio-temporal segmentation based on the method proposed by Dufaux *et al.* is obtained by requiring two objects. Even with this additional information, the resulting spatio-temporal segmentation does not properly isolate the head of the man. A hole is indeed visible in the chin. Moreover, the man and his helmet are unconnected. Similar to the "Akiyo" and the "Table Tennis" video sequences, the spatio-temporal segmentation obtained with the spatio-temporally homogeneous initial regions compares favorably with one relying on the quadtree based initial regions (see Fig. 5.13 and Fig. 5.10).

Figure 5.14 gives a final illustration of the spatio-temporal segmentation obtained with spatio-temporally homogeneous initial regions. The set of initial regions contains 40 regions. These are reduced to 9 after the strong rule, while 2 objects are determined after the weak rule is applied. The objects are the background and the fish. Again, the performance of the proposed algorithm algorithm compares favorably with the performance of the algorithm proposed by Dufaux *et al.* (two objects are required).

In contrast to the initial regions based on a quadtree segmentation, the spatio-temporally homogeneous initial regions permit a very precise determination of the objects. This precision is reflected both in terms of spatial and temporal characteristics. The resulting decomposition of the scene enables to carry out a thorough analysis of the scene content. Moreover, the proposed algorithm has been shown to outperform the algorithm used as benchmark.

Figure 5.11: "Akiyo": Spatio-temporal segmentation using arbitrarily shaped initial regions. **(a)** Previous frame after global motion compensation, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame, **(h)** spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.*, **(i)** boundaries of the spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.* superimposed onto the current frame.

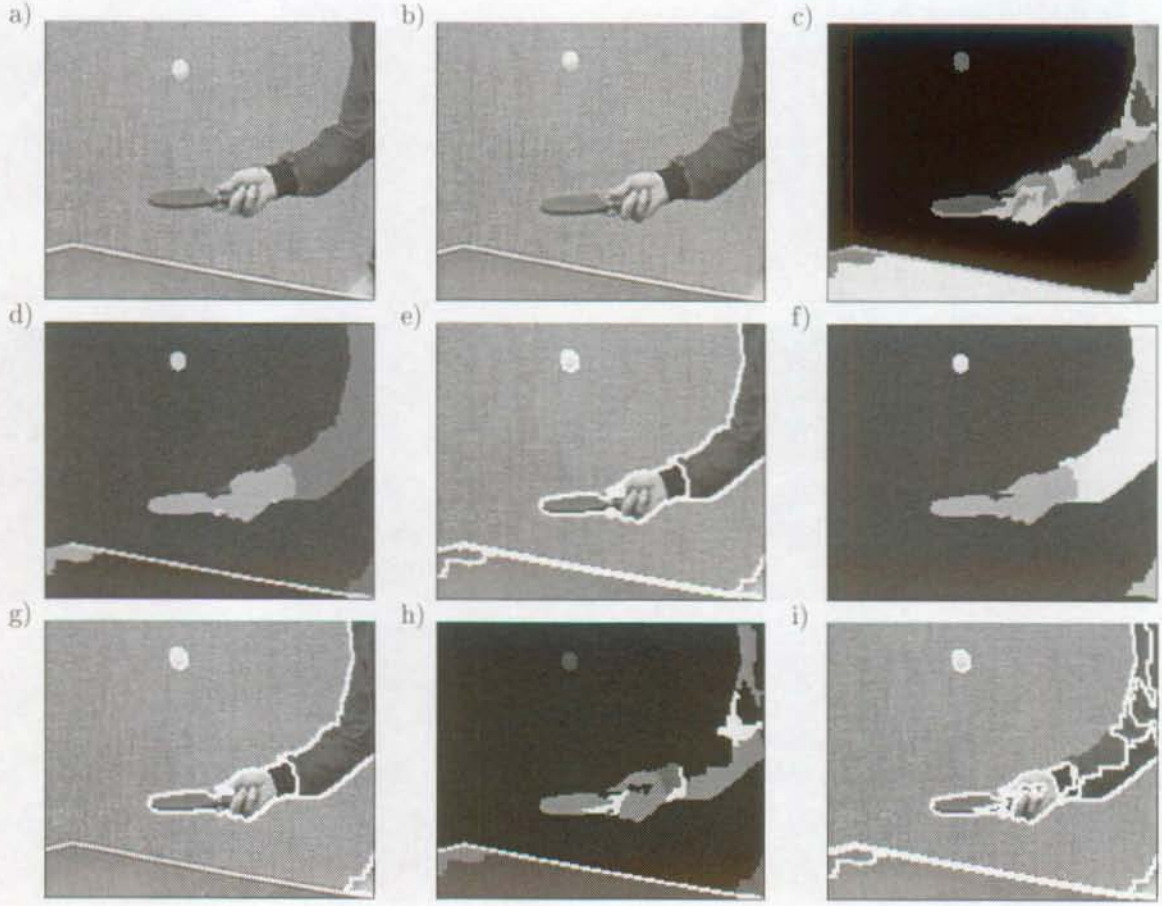

Figure 5.12: "Table Tennis": Spatio-temporal segmentation using arbitrarily shaped initial regions. **(a)** Previous frame after global motion compensation, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame, **(h)** spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.*, **(i)** boundaries of the spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.* superimposed onto the current frame.

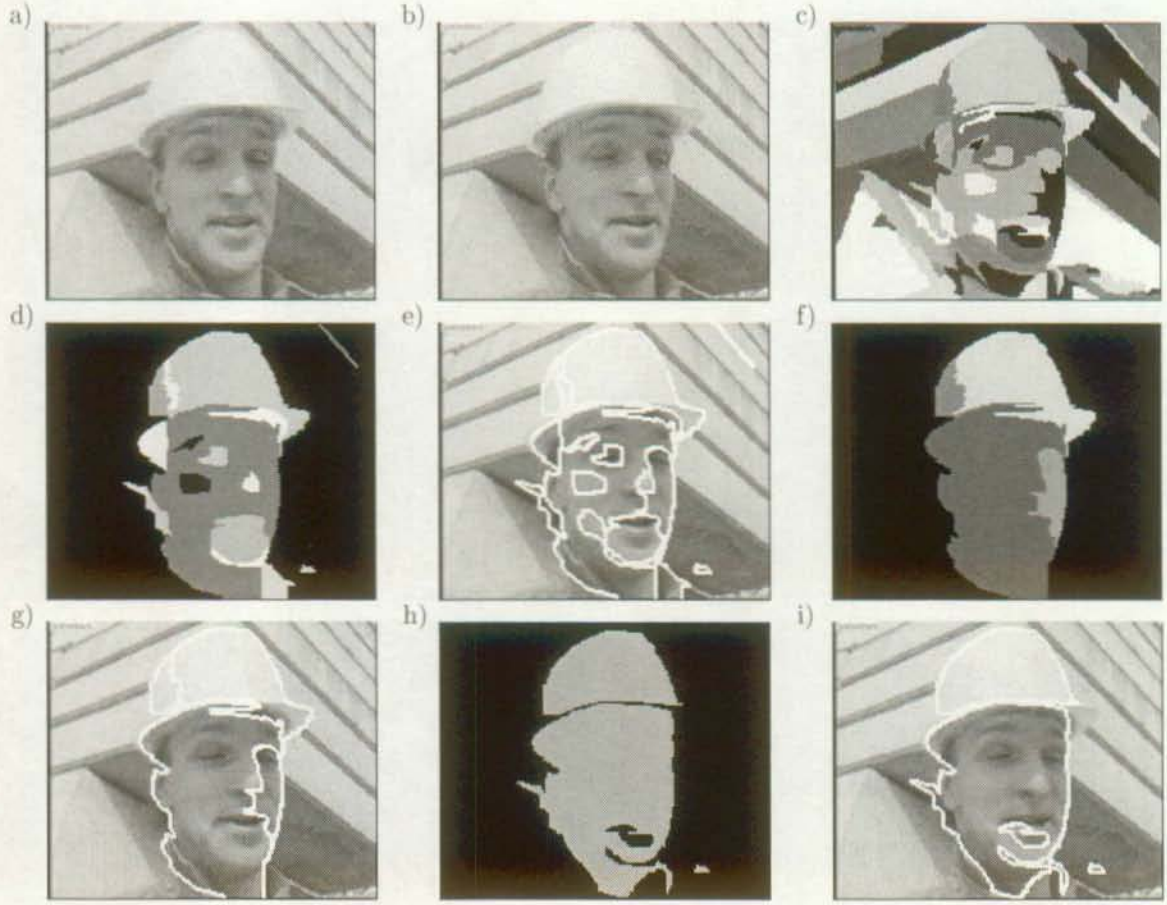

Figure 5.13: "Foreman": Spatio-temporal segmentation using arbitrarily shaped initial regions. **(a)** Previous frame after global motion compensation, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame, **(h)** spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.*, **(i)** boundaries of the spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.* superimposed onto the current frame.

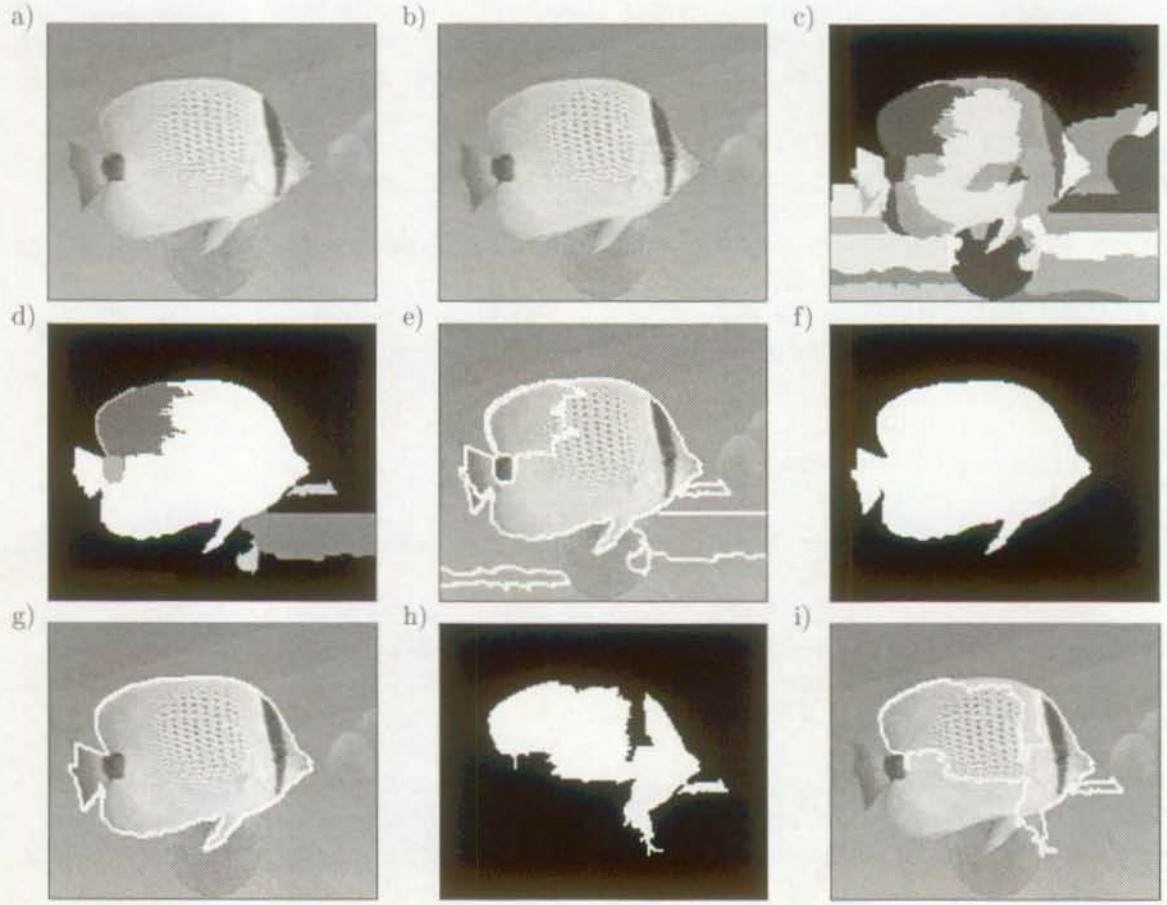

Figure 5.14: "Bream": Spatio-temporal segmentation using arbitrarily shaped initial regions. **(a)** Previous frame after global motion compensation, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation after having applied the strong rule, **(e)** boundaries of the spatio-temporal segmentation after having applied the strong rule superimposed onto the current frame, **(f)** spatio-temporal segmentation after having applied both rules, **(g)** boundaries of the spatio-temporal segmentation after having applied both rules superimposed onto the current frame, **(h)** spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.*, **(i)** boundaries of the spatio-temporal segmentation obtained with the technique proposed by Dufaux *et al.* superimposed onto the current frame.

5.5.4 Impact of the luminance information

As described in Sec. 5.3.7, the proposed spatio-temporal similarity integrates into a single measure both temporal and spatial information. Doubts may be raised about the necessity of the latter information. Indeed, it may be advocated that an object is primarily defined through its temporal coherence.

Figure 5.15 represents the different stages of the spatio-temporal segmentation for a frame of the video sequence "Table Tennis". The current frame corresponds to the time where the ball roughly reaches the top of its trajectory. Consequently, the motion of the ball is very small. Figure 5.15 reports two simulations of spatio-temporal segmentation with luminance factors f_L set to 1.0 and 0.0, respectively. The former case equates to a maximum use of the spatial information, while the latter case means that no spatial information is utilized. When setting f_L to 1.0, the spatio-temporal segmentation is able to create the object representing the ball. This is not possible when no spatial information is used (i.e. $f_L = 0.0$). The motion of the ball is indeed too small to discriminate it from the background.

We see that a robust spatio-temporal segmentation requires that both temporal and spatial information are used. Although an object is primarily defined through its temporal coherence, the spatial information may supply important contributions. This is especially the case when the temporal characteristics of the object are not discriminatory enough.

5.5.5 Impact of the pondering factor λ

In Sec. 5.3.4, it is advocated that the temporal similarity should rely on the temporal information existing in both the parametric representation and the residual distribution, the emphasis being on the latter. Based on these observations, the Modified Kolmogorov-Smirnov (MKS) test statistic, Q_{MKS}, has been proposed. However, one may wonder about the necessity of combining the two types of temporal information. In particular, the need for the parametric information, embodied by the pondering factor λ, may be questioned.

The importance of the pondering factor is demonstrated by modifying slightly the conditions of the simulation depicted in Fig. 5.11. More precisely, suppose we do not use the pondering factor λ. The results of the simulation are presented in Fig. 5.16. The spatio-temporal segmentation is formed of 3 objects: the background, the head with the right part of the body and, the left part of the body. Although only the temporal information available in the residual distribution is used, the resulting decomposition of the scene is quite accurate. However, the additional contribution of the parametric information would have allowed a better definition of the objects. This is illustrated by comparing Fig. 5.11(f) with Fig. 5.16(c).

5.6 Conclusions

In this chapter, a technique for unsupervised spatio-temporal segmentation has been presented. Starting from a set of initial regions, these are iteratively merged in order to determine the objects forming the scene. The regions are merged on the basis of their mutual spatio-temporal similarities. Built as a combination of temporal and spatial information, the spatio-temporal similarity is defined in the statistical framework of hypothesis testing. To that end, a novel test statistic for the temporal information is presented. Referred to as the Modified Kolmogorov-Smirnov test, it permits the simultaneous use of the temporal information existing in the residual distribution and in the motion parametric representation. The actual merging of the regions is carried out by using a weighted, directed graph. Two graph clustering

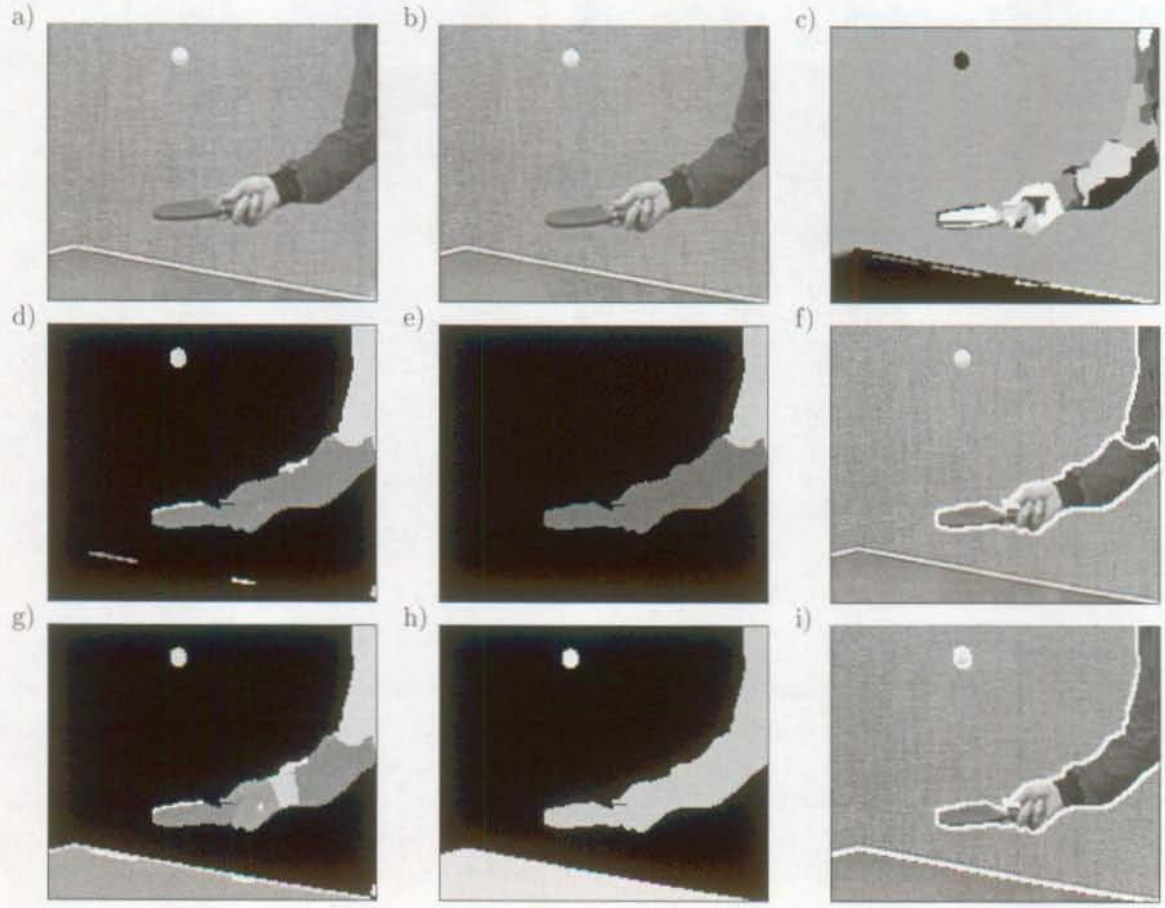

Figure 5.15: "Table Tennis": Impact of the luminance information on the spatio-temporal segmentation. **(a)** Previous frame after global motion compensation, **(b)** current frame, **(c)** set of initial regions, **(d)** spatio-temporal segmentation with a luminance factor $f_L = 0.0$ after having applied the strong rule, **(e)** spatio-temporal segmentation with a luminance factor $f_L = 0.0$ after having applied both rules, **(f)** boundaries of the spatio-temporal segmentation with a luminance factor $f_L = 0.0$ after having applied both rules superimposed onto the current frame, **(g)** spatio-temporal segmentation with a luminance factor $f_L = 1.0$ after having applied the strong rule, **(h)** spatio-temporal segmentation with a luminance factor $f_L = 1.0$ after having applied both rules, **(i)** boundaries of the spatio-temporal segmentation with a luminance factor $f_L = 1.0$ after having applied both rules superimposed onto the current frame.

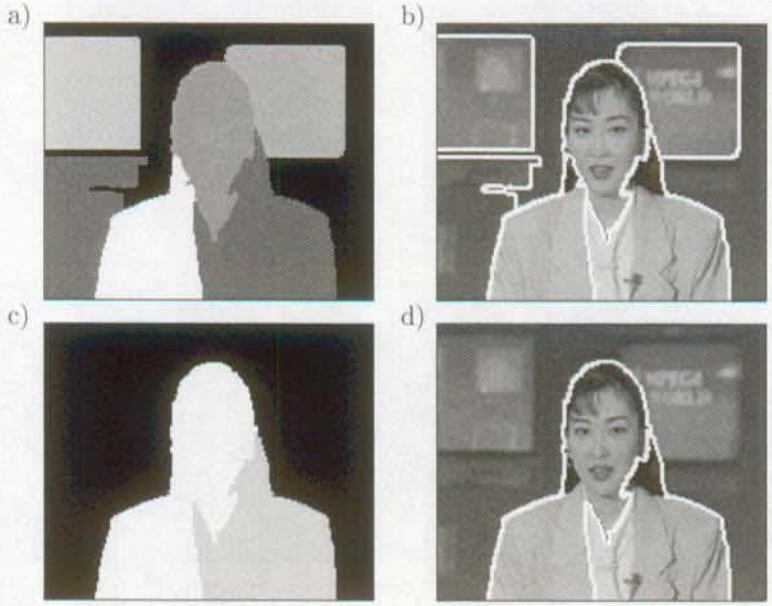

Figure 5.16: "Akiyo": Impact of not using the pondering factor λ for the spatio-temporal segmentation. **(a)** Spatio-temporal segmentation after applying only the strong rule, **(b)** the boundaries of the regions in **(a)**, superimposed onto the current frame, **(c)** final spatio-temporal segmentation, **(d)** the boundaries of the regions in **(c)**, superimposed on the current frame.

rules are proposed. They are denoted the strong rule and the weak rule. They are applied successively, each of them being embedded into a dynamic update strategy of the graph.

The proposed technique has been evaluated on different types of video sequences. Furthermore, different types of initial regions have been used. In all cases, the proposed technique is able to define the objects composing the scene. However, the spatial and temporal precision of the objects depends on the quality of the initial regions. The simulations have also shown the importance of using spatial information. Such information may contribute significantly when temporal information is not discriminatory enough. Finally, the necessity of combining the temporal information existing in both the residual distribution and the parametric representation has been demonstrated.

Chapter 6

Recursive spatio-temporal segmentation and object tracking

6.1 Introduction

The spatio-temporal segmentation is usually based on two consecutive frames and, as such, suffers from several drawbacks. In particular, estimation inaccuracies, noise, as well as the lack of decisive information may alter the interpretation of the scene. The resulting spatio-temporal segmentation may, therefore, not be representative of the scene content. Furthermore, the coherence of the spatio-temporal segmentation is not guaranteed through time. Assuming that no scene change occurs, the successive images of the video sequence are indeed formed by a similar set of objects. Consequently, the corresponding spatio-temporal segmentations are bound to have strong similarities with one another. By ensuring such stability, a dynamic understanding of the sequence as a whole is obtained. This is very well suited for content-based video coding and to the new functionalities which are being promoted in the framework of the MPEG-4 activities [76].

In order to obtain a robust spatio-temporal segmentation and to stabilize it through time, the information available in multiple frames should be used. One solution is to work off-line in a batch manner. The whole sequence is first captured. The current segmentation benefits from information coming from previous frames as well as from future frames. However, the batch approach introduces a time delay which may be unacceptable in such applications as video coding. The alternative to the batch approach is the recursive approach. The current segmentation benefits only from past information and hence can be performed on-line. In the literature, two classes of recursive spatio-temporal segmentation techniques have been proposed. The first class uses past information as an initialization for the current segmentation [126, 63, 95, 148, 26]. The objects are assumed to have a steady motion which permits the projection of the previous partition onto the current frame. The interaction between past and current information is limited to the projection step. The methods belonging to the second class use the past information to a fuller extent [112, 22, 29]. Not only does the past segmentation provide an initialization, but it is also used to constrain the current segmentation process. Typically, the current segmentation procedure may be penalized whenever the current segmentation deviates from the previous one.

In order to further stabilize the segmentation procedure, the dynamic image analysis denoted as object tracking may be carried out. The tracking procedure aims at solving the correspondence problem [7]. More precisely, it aims at identifying and pursuing the objects forming the scene throughout the video sequence. Based on that knowledge, one can define the spatio-temporal trajectories. Formally, the tracking boils down to assigning to an object the same label regardless of the frame it is in. Although tracking is intimately related to the spatio-temporal segmentation process, the former distinguishes itself by being a scene understanding technique (see Sec. 3.6.3). It indeed involves an identification of the different objects through time. Conversely, the spatio-temporal segmentation is an image processing technique which may be seen as an instantaneous measurement of the scene dynamics [100]. In the literature, different object tracking algorithms have been proposed [29, 112, 26, 22]. However, they do not track objects in the literal sense. These techniques indeed simply propose recursive spatio-temporal segmentation procedures, the tracking being limited to the projection of the previous labels onto the current frame. In case an object disappears and then appears again, different labels are used although it is the same object which is detected again. The lack of a proper characterization of the object prevents any identification procedure to be performed. In contrast to these techniques, Meyer and Bouthemy propose an object tracking technique that explicitly tracks the objects [100]. After having detected the objects, the object tracking is addressed through two interacting dynamic systems. These model the temporal evolution of the geometry and the motion of the objects in the scene. Although the tracking paradigm is properly addressed, this object tracking algorithm fails to interact with the spatio-temporal segmentation.

In this chapter, an unsupervised algorithm is presented that robustly segments a video sequence in terms

of multiple moving objects and tracks them through time. An overview of the proposed algorithm is outlined in Sec. 6.2. The algorithm is composed of two parts. They are respectively a recursive spatio-temporal segmentation method and an object tracking method. These two parts interact and mutually influence each other. Typically, the spatio-temporal segmentation impacts on the tracking information, while the latter is accredited to modify the spatio-temporal segmentation if necessary. The recursive spatio-temporal segmentation technique is presented in Sec. 6.3. Its underlying idea is to find the best trade-off between past and current information. This allows to stabilize the current segmentation on the basis of past information, while allowing for changes such as the appearance of a new object. The tracking algorithm is presented in Sec. 6.4. Starting from the current spatio-temporal segmentation, it tries to identify which objects are present. Each object is characterized through a set of distinctive features. Based on these, the tracking algorithm is carried out within the statistical framework of multiple hypotheses testing. A hierarchy of correspondence hypotheses is proposed which permits to robustly tackle the correspondence problem. Finally, Sec. 6.5 presents experimental results, while Sec. 6.6 draws the conclusions.

6.2 Overview of the proposed technique

The algorithm for recursive spatio-temporal segmentation and object tracking expects as input the tracked segmentation of the previous frame, i.e. the frame at time $t-1$. The segmentation is formed by the set, $\mathcal{O}(t-1)$, of $N_o(t-1)$ objects. Although this may be provided by the user, the spatio-temporal segmentation technique described in Chapter 5 may supply this segmentation automatically. The output of the proposed algorithm is the tracked segmentation of the current frame, i.e. at time t. This is given under the form of the ensemble $\mathcal{O}(t)$, which contains $N_o(t)$ objects. If no scene changes occurs, the ensembles $\mathcal{O}(t)$ and $\mathcal{O}(t-1)$ should share common elements, i.e. $\mathcal{O}(t) \bigcap \mathcal{O}(t-1) \neq \emptyset$. These common elements are the objects which are present in both frames. This remark may be extended to other previous frames. In particular, it may happen that an object disappear for a certain lapse of time and then appears again in the current frame. In that case the object should be present in the set of objects forming the current frame and the set of objects forming the frame captured just before it disappeared. It must be underlined that the numbering of the objects in the ensembles $\mathcal{O}(j)$, $j \in [t, t-1, t-2, ...]$, is usually not consecutive. This arises because labels of objects which disappear over time are not assigned to any subsequently newly detected object. Figure 6.1 gives an overview of the proposed algorithm.

The recursive spatio-temporal segmentation and object tracking consists of four successive steps. These steps are schematically: partition projection, region validation, constrained spatio-temporal segmentation and object tracking. The first three steps form the *recursive spatio-temporal segmentation* technique. They aim at combining past and current information in order to robustly derive the current spatio-temporal segmentation. Fundamentally distinct from the first three steps, the *object tracking* step uses the information about past objects to determine whether they are present in the current frame. This identification relies on the segmentation obtained by the recursive spatio-temporal segmentation technique. However, the object tracking algorithm is empowered to modify it if necessary. The proposed algorithm is in sharp contrast with the techniques which segment and track the objects in parallel [29, 112, 26, 22]. By mixing the two issues, these techniques do not exploit efficiently all of the available information. In particular, the tracking issue is not properly addressed. This loss of information renders the techniques less robust and less responsive to changes in the sequence, and may even alter the interpretation of the scene.

The recursive spatio-temporal segmentation aims at robustly partitioning the successive frames of the

sequence. To that end, both past and current spatio-temporal information is used. The recursive segmentation balances, on one hand, the requirement of being coherent with the previous segmentations and, on the other hand, the need to adapt to local phenomena in the current frame. This trade-off is achieved in three successive steps. In the first stage, a partition projection is performed. The tracked segmentation $\mathcal{O}(t-1)$ of the previous frame is projected onto the current frame. In order to have a precise spatial resolution, this partition projection is performed onto an oversegmentation of the current frame. Next, the regions resulting from the projection undergo a validation procedure. This decides whether the regions are likely to be well-defined objects. Two tests are carried out. One test checks their temporal homogeneity while the other test verifies their possible correspondence with previously detected objects. The regions that do not pass both of these tests are split. The last stage of the recursive spatio-temporal segmentation is the constrained spatio-temporal segmentation procedure. Starting from the set of regions resulting from the two first stages, it aims at merging them according to current spatio-temporal information, while being constrained through the information arising from the segmentation of the previous frame. The region merging is performed according to the technique presented in Chapter 5. The spatio-temporal similarity is however modified in order to integrate both past and current information. This region similarity is referred to as the *recursive spatio-temporal similarity*. Through these three successive stages, the recursive spatio-temporal segmentation is able to mix past and current information in order to robustly define the current spatio-temporal segmentation.

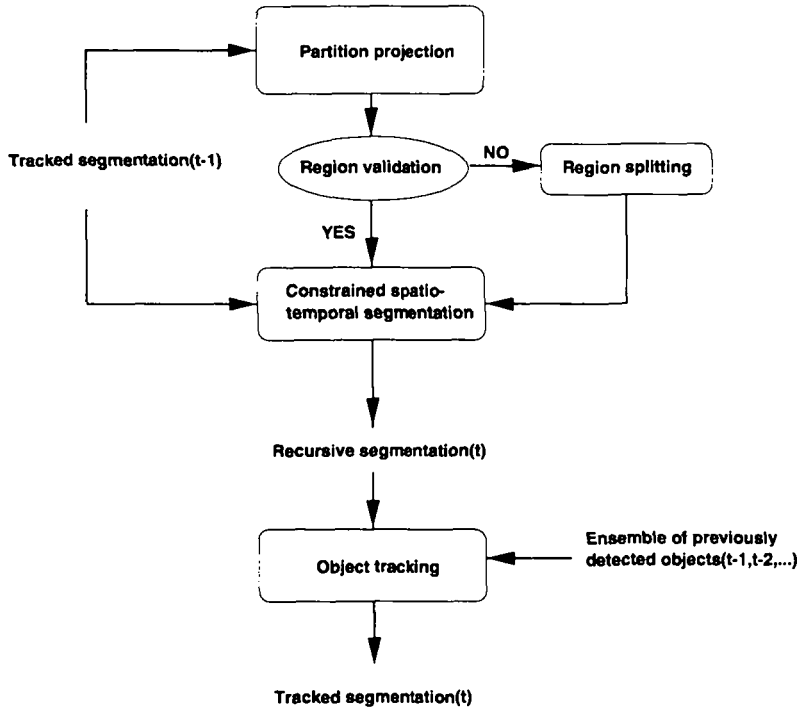

Figure 6.1: Overview of the proposed algorithm for recursive spatio-temporal segmentation and object tracking. The segmentation of the previous frame is assumed known. The recursive segmentation comprises three stages which are the partition projection, the region validation and the constrained spatio-temporal segmentation. The resulting recursive segmentation of the current frame is used by the object tracking algorithm.

Starting from the segmentation given by the recursive spatio-temporal segmentation procedure, the object tracking algorithm is applied. By considering all the objects previously detected, $\mathcal{O}(j)$, $j \in [t-1, t-2, ...]$,

it tackles the correspondence problem and identifies which objects are present in the current frame. The final output is the tracked segmentation $\mathcal{O}(t)$ of the current frame. The tracking algorithm permits to follow the objects through the video sequence, resulting in a high level understanding of the dynamic scene. In order to carry out its task, the tracking algorithm must be able to specifically identify each object. This is achieved by characterizing each object through a set of representative features. These features may be split into three categories: the temporal, the spatial, and the spatio-temporal features. With regard to the temporal features, they are naturally taken to be the parameters of the motion model. The spatial features have been chosen to be affine invariant parameters based on moments [88, 138]. This choice is motivated by the fact that the spatial features of the object must be invariant although the object is moving. Finally, the spatio-temporal features of the object are related to its trajectory. They simply assess the likely position of the object in the current frame under the assumption that the object motion changes smoothly over time. This ensemble of features permits to precisely characterize each object and to identify it in the statistical framework of multiple hypotheses testing. Based on the set of features, a hierarchy of correspondence hypotheses are tested successively. If necessary, the object tracking algorithm may modify the segmentation obtained by the recursive spatio-temporal segmentation algorithm. Furthermore, the tracking algorithm allows for objects to appear and disappear. Finally, it permits to build an object memory which resolves for complete occlusion/disocclusion phenomena [100].

6.3 Recursive spatio-temporal segmentation

6.3.1 Introduction

Generally speaking, the successive frames of a video sequence are very similar. In particular, they are roughly composed of the same set of objects. This implies that the successive spatio-temporal segmentations should show a high level of coherence with one another. This requirement for a stable segmentation should however accommodate for peculiar changes taking place in a given frame. For instance, objects may appear, disappear as well as suddenly change their motion or shape. The challenge lies in disentangling the possible contradictions between past and current information. The spatio-temporal segmentation procedure must find the right compromise between the requirement of stability and the necessity of allowing for changes.

6.3.2 Partition projection and region validation

The likelihood that an object found in the previous frame appears in the current one depends on the object definition. If the object is spatio-temporally homogeneous, it is very likely to appear intact in the current frame also. Conversely, if the object is intrinsically not homogeneous, it is likely to vanish as a single entity. This observation is the basis for the proposed partition projection and region validation algorithms.

The concept of partition projection has been widely used in the literature [148, 26]. It is based on two assumptions. The first is that the segmentation of the previous frame constitutes a good approximation for the current one. The second assumption is that the hypothesis of motion temporal continuity holds for each object (see Sec. 2.2.2). In practice, the partition projection is carried out by projecting the previous partition $\mathcal{O}(t-1)$ onto the current frame. The projection is performed according to the motions of the objects. Nevertheless, both the assumption of strong similarities between the previous and current segmentations as well as the assumption of the motion temporal continuity may be put into

fault. Moreover, the motion information used for the projection may also be inaccurate or erroneous.

The aim of the proposed partition projection algorithm is twofold. On one hand, it must be able to detect the possible breakdowns of the hypotheses of segmentation similarities and of temporal continuity of the motion. This also includes the case of erroneous temporal information. Also, it aims at preventing the shape of the object from being spoilt by the projection procedure. The partition projection is carried out as follows. The partition of the previous frame, $\mathcal{O}(t-1)$, is projected onto an oversegmentation of the current frame. The oversegmentation is derived according to the method proposed in Chapter 4. For each region of the oversegmentation, the partition projection procedure decides whether there is any suitable object in $\mathcal{O}(t-1)$ to which the region can be assigned. To that end, the two ensembles $\mathcal{E}_c$ and $\mathcal{E}_l$ are defined. The ensemble $\mathcal{E}_c$ is the set of objects in $\mathcal{O}(t-1)$ which, after projection, cover the region by more than a predefined threshold. The ensemble $\mathcal{E}_l$ is defined as the set of objects in $\mathcal{O}(t-1)$, whose luminance properties are in accordance with the luminance of the region of interest. For each object, this is checked by comparing the luminance median of the region with the luminance median computed only on the object pixels which are projected onto the region. The object is put into the ensemble $\mathcal{E}_l$ if the relative difference between the two medians is not larger than a predefined threshold. Finally, the region under consideration is assigned to the object O_K defined as

$$O_K = \min_{O_j \in \mathcal{E}_c \bigcap \mathcal{E}_l,\ MSE(M_{O_j}) \leq T_{MSE}} \left(MSE\left(M_{O_j}\right) \right) , \tag{6.1}$$

where M_{O_j} stands for the motion vector of the object O_j and $MSE\left(M_{O_j}\right)$ is the Mean Square Error obtained by motion compensating the region of interest with the parameters M_{O_j}. The quantity T_{MSE} is a predefined threshold.

According to Eq. (6.1), the region is assigned to the object O_K which has very specific properties. It belongs to both ensembles $\mathcal{E}_c$ and $\mathcal{E}_l$. This ensures that it possesses the right covering and luminance properties. Furthermore, the object O_K has the motion that best suits the region under consideration, while keeping the corresponding MSE lower that the threshold T_{MSE}. It must be underlined that it may happen that no object O_K is found. This occurs when either the intersection between $\mathcal{E}_c$ and $\mathcal{E}_l$ is empty or when the condition imposed by the threshold T_{MSE} is not satisfied. In that case, the region is not assigned to any object and is let alone. After having decided for each region of the oversegmentation to which object it is assigned, the regions assigned to the same object are merged. Regions that are too small are removed on the basis of the motion information. The removal is performed according to temporal information and uses the region removal technique presented in Sec. 4.5.

The proposed partition projection is illustrated in Fig. 6.2(a) to Fig. 6.2(e). In the selected scene, a man is playing table tennis. Only the arm and the hand of the man are visible. The previously tracked segmentation contains four objects, $\mathcal{O}(t-1) = \{O_1, O_2, O_4, O_7\}$. Note that the objects do not have a continuous numbering due to objects which have disappeared. The objects forming the scene are respectively the background, the arm, the ball and the hand with the bat. The objects are projected onto an oversegmentation of the current frame. In the resulting set of regions, one clearly notices the regions that have not been assigned to any object.

The segmentation obtained by the partition projection procedure provides the initial approximation of the current spatio-temporal segmentation. However, the validity of each region has to be verified. A region is said to be *valid* if it satisfies the following two criteria. First, it must be temporally homogeneous. Second, it must directly relate to an object present in the previous frame. The regions not judged valid are split into smaller regions.

In order to be considered valid, a region must show an intrinsic semantic meaning. In other words, the region should naturally correspond to an object that was previously detected (i.e. an object belonging to

$\bigcup_{j=0}^{j=t-1} \mathcal{O}(j)$). If satisfied, this property accredits the hypothesis that the region is likely to be an object. The correspondence is verified on the basis of the tests referred as "**situation 1**" and "**situation 2**" in Sec. 6.4.6. The second criterion requires that the region is temporally homogeneous. This property is tested through the MSE obtained by motion compensating the region. In order to represent a reliable measure of the region temporal homogeneity, the MSE must avoid taking into account errors due to disocclusion. Therefore, the disocclusion effects are corrected using the approach presented in Sec. 4.4.3. Through backward motion compensation of the set of regions, the pixels of the previous frame that belong to disoccluded areas are detected. These are then removed from the computation of the MSE. The region is said temporally homogeneous if the corresponding MSE is lower that a preset threshold. In case the region does not satisfy both of the above criteria, it is split through an oversegmentation procedure. A typical example of the region validation procedure is presented from Fig. 6.2(e) to Fig. 6.2(l). Starting from the set of regions obtained by partition projection, Fig. 6.2(e), the regions that correspond to previously detected objects are determined. In Fig. 6.2(f), the regions that do not satisfy this correspondence criterion are shown in white. Note that the region defining the ball is not found to satisfy this criterion. This is due to the brisk acceleration that the ball undergoes between the two selected frames. Then, the criterion on temporal homogeneity is checked. In order to compute it, the effects due to disocclusion phenomena are corrected for. In Fig. 6.2(g) and Fig. 6.2(h), the DFD is shown respectively before and after their removal. The regions that are not considered as temporally homogeneous are depicted in white in Fig. 6.2(i). Note, in this set, the presence of the region corresponding to the hand with the bat. By combining the two criteria, the ensemble of regions that are not valid is derived. These are shown in white in Fig. 6.2(j). The final set of regions is given in Fig. 6.2(k). The valid regions are kept untouched, while the other regions are split through an oversegmentation procedure.

6.3.3 Constrained spatio-temporal segmentation

Through the partition projection and the region validation procedures, the current frame is segmented into a set of regions $\mathcal{R}(t)$. By construction, the set $\mathcal{R}(t)$ constitutes an oversegmented approximation to the current spatio-temporal segmentation. Indeed, it is built through a conservative selection of the secure pieces of information. Doubtful regions are split through an oversegmentation. This cautious approach renders the set of regions $\mathcal{R}(t)$ an optimal starting point upon which to construct the current spatio-temporal segmentation $\mathcal{O}(t)$.

Starting from the set $\mathcal{R}(t)$, the current spatio-temporal segmentation is performed through an algorithm identical to the region merging algorithm presented in Chapter 5, except for a different definition of the spatio-temporal similarity. It is modified in order to use past spatio-temporal information which arises from the spatio-temporal segmentation, $\mathcal{O}(t-1)$, of the previous frame. The new spatio-temporal similarity is referred to as the *recursive spatio-temporal similarity.*

6.3.4 The past spatio-temporal similarity

Given the regions A and B, it should be assessed whether they belong to the same object. In that respect, the spatio-temporal segmentation, $\mathcal{O}(t-1)$, of the previous frame contains useful pieces of information. This information is referred to as the *past spatio-temporal information*, as it depends on both the spatial and the temporal characteristics of the objects in $\mathcal{O}(t-1)$. Based on this information, the *past spatio-temporal similarity* between the regions A and B may be defined. Denoted P_{AB}, it is derived within the framework of hypothesis testing. The null hypothesis, H_0, states that the regions A and B belong to the same object as far as the past spatio-temporal information is concerned. The alternative hypothesis, H_1,

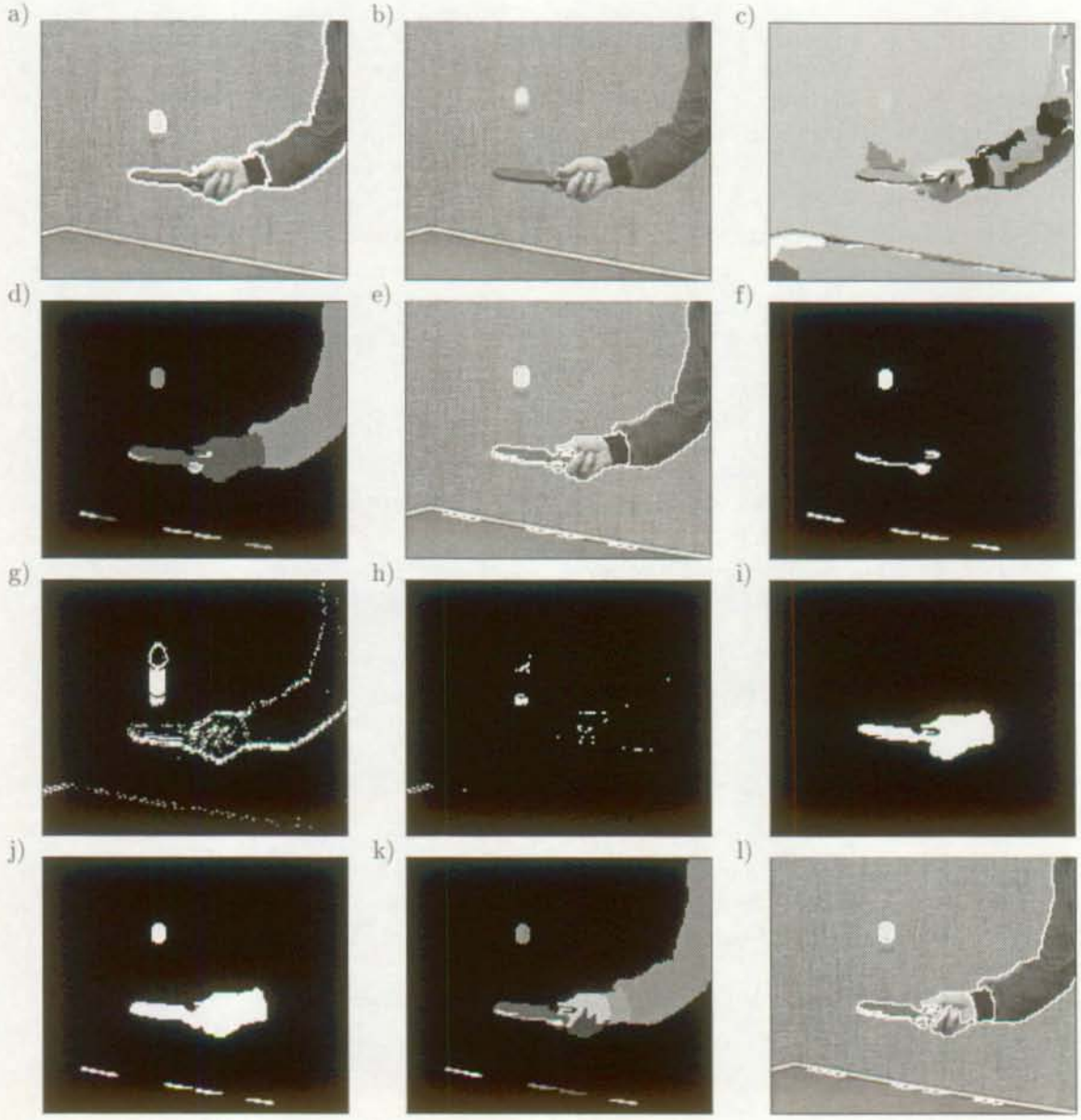

Figure 6.2: "Table Tennis": Example of partition projection and object validation procedures. **(a)** Previous frame superimposed with its four constituent objects, **(b)** current frame, **(c)** oversegmentation of the current frame, **(d)** ensemble of regions resulting from the partition projection, **(e)** boundaries of the regions resulting from the partition projection superimposed onto the current frame, **(f)** the regions (in white) which do not correspond to objects of the previous frame, **(g)** DFD before correcting for the disocclusion effects, **(h)** DFD after correcting for the disocclusion effects, **(i)** ensemble of regions (in white) which are not temporally homogeneous, **(j)** ensemble of regions (in white) which are not valid, **(k)**, final set of regions, **(l)** final set of regions superimposed onto the current frame.

simply negates the null hypothesis. The value of P_{AB} is naturally defined as being the significance level of the null hypothesis. This entails that the value of P_{AB} ranges from 100% to 0%. Unlike to the spatial and temporal similarities presented in respectively Sec. 5.3.3 and Sec. 5.3.4, the regions A and B do not need to be adjacent in order to have $P_{AB} \neq 0\%$.

In order to determine the value of P_{AB}, a relationship between the objects in $\mathcal{O}(t-1)$ and the regions A and B must be established. This is achieved by projecting the ensemble of objects onto the current frame. It is then checked whether the regions A and B show the same behavior when confronted to this projection. To that end, the different covering ratios C_{AK} and C_{BK}, $\forall K \in \mathcal{O}(t-1)$, are computed. The ratio C_{AK} is defined as the proportion of the region A which is covered by the projection of the object K. The other covering ratios may be derived by analogy. The idea is that if the two regions are mostly covered by the same object, they are likely to be part of the same current object. Conversely, strong differences in their respective covering ratios may hint that they should not be merged together. The hypothesis testing necessitates the definition of a test statistic Q_{st}. Starting from the covering ratios, the realization q_{st} of the test statistic Q_{st} is defined as

$$q_{st} = 1.0 - \max_{K \in \mathcal{O}(t-1)} (|C_{AK} - C_{BK}|) \ . \tag{6.2}$$

The proposed test statistic Q_{st} is representative of the difference of behavior existing between the regions A and B. When examining the projection of the different objects in $\mathcal{O}(t-1)$, the maximum discrepancy between the resulting covering ratios is taken as reference. This ensures that the proposed test is very conservative when assessing the past spatio-temporal information.

Assuming the test statistic Q_{st} to be uniformly distributed between 0 and 1, P_{AB} (i.e. the significance level) is given as a function of the realization q_{st}. Namely,

$$P_{AB} = P(Q_{st} \leq q_{st}) = \int_0^{q_{st}} dx \ . \tag{6.3}$$

Figure 6.3 depicts the relationship existing between q_{st} and P_{AB}. They are most obviously directly related to one another, P_{AB} reaching the maximum value of 100% when no differences exist in the covering ratios. Finally, it must be noted that the past spatio-temporal similarity is symmetric (i.e. $P_{AB} = P_{BA}$) and reflexive (i.e. $P_{AA} = 100\%$).

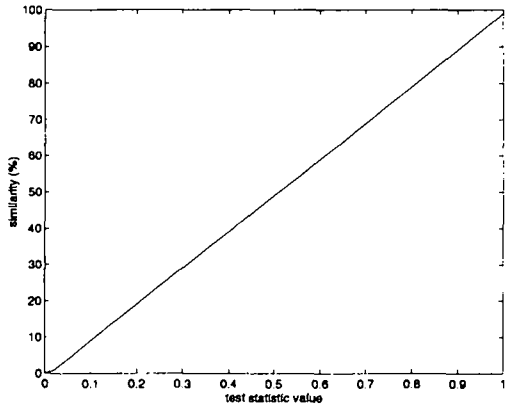

Figure 6.3: The past spatio-temporal similarity as a function of q_{st}. The two terms are directly proportional.

6.3.5 Defining the recursive spatio-temporal similarity

When evaluating the likelihood that the region A belongs to the same object as the region B, the measure of their similarity should be provided. Based on this information, the merging strategy presented in Sec. 5.4 builds the objects forming the scene. The proposed *recursive spatio-temporal similarity* includes two kinds of information. The first kind of information comes from the previous spatio-temporal segmentation. This ensures the stability of the successive spatio-temporal segmentations. The second kind of information arises from the current frame. If necessary, it must counteract the requirement of stability imposed by the first kind of information. The proposed recursive spatio-temporal similarity $recSim(AB)$ is written as follows

$$recSim(AB) \quad = \quad f_C \; T_{AB} \; P_{AB} + (1 - f_C) \; Sim(AB) \; , \tag{6.4}$$

where f_C, $f_C \in [0, 1]$, is the *constraint factor*. The term P_{AB} is the past spatio-temporal similarity and is given by Eq. (6.3). The term $Sim(AB)$ stands for the current spatio-temporal similarity and is defined by Eq. (5.19). The term T_{AB} represents the current temporal similarity. Its definition is similar to the one given in Eq. (5.13), the difference being that the regions A and B are no longer required to be adjacent. The temporal similarity T_{AB} is indeed also computed if the past spatio-temporal similarity P_{AB} is not zero.

Equation (6.4) shows that the recursive spatio-temporal similarity measure $recSim(AB)$ is composed of two terms. The first one characterizes the past spatio-temporal information. It indeed incorporates the term P_{AB} which represents the information arising from the objects detected in the previous frame. This information is however contingent to the current temporal similarity T_{AB}. In other words, the influence of the past spatio-temporal information is dictated by the current temporal similarity. This is in accordance with the fact that an object is primarily characterized by its temporal coherence. The second term stands for the current spatio-temporal information. Defined by Eq. (5.19), it is a combination of current temporal and spatial similarities. The respective importance of the two terms of Eq. (6.4) is regulated through the constraint factor f_C. It allows the user to define the trade-off between past and current information in the recursive spatio-temporal segmentation procedure. The larger the constraint factor f_C, the bigger the requirement for the current spatio-temporal segmentation to resemble the previous one. In case the constraint factor f_C is set to zero, the constrained spatio-temporal segmentation only uses current information and boils down to the technique presented in Chapter 5. In Fig. 6.4, an example of constrained spatio-temporal segmentation is presented. The initial set of regions is given in Fig. 6.4(a) and comprises 20 regions. It derives from the example of partition projection and region validation procedures presented in Fig. 6.2. Figures 6.2(l) and 6.4(a) are thus similar. After applying the strong rule, 6 regions are left. The weak rule permits to decrease them further to 4, the resulting spatio-temporal segmentation showing strong similarities with the one of the previous frame, i.e. Fig. 6.2(a). Indeed, the same set of objects seems to constitute both images. The determination of whether this is really the case necessitates the recourse to the object tracking algorithm.

6.4 Object tracking

6.4.1 Introduction

The recursive spatio-temporal segmentation technique described in Sec. 6.3 produces the spatio-temporal segmentation of the current frame. However, this segmentation stands on its own as no temporal linkage with segmentations of former frames is available yet. More precisely, the ensemble $\mathcal{O}(t)$ of the objects

Figure 6.4: "Table Tennis": Example of constrained spatio-temporal segmentation. **(a)** Set of regions obtained by partition projection and region validation, **(b)** partition after applying the strong rule and, **(c)** segmentation of the current frame after applying the weak rule.

forming the current scene have not yet been put into correspondence with previously detected objects. By solving the correspondence problem, the object tracking procedure is essential as it enables a pursuit of each detected object [100]. In doing so, a high level understanding of the scene dynamics is achieved. Furthermore, the tracking procedure may provide with valuable information which will improve the segmentation of the current frame.

In the following, L denotes an object present in the segmentation of the current frame, $L \in \mathcal{O}(t)$. K is an object which has been previously detected. The time at which it was last detected does not matter, i.e. $K \in \bigcup_{j=0}^{j=t-1} \mathcal{O}(j)$ where the initial time is 0.

6.4.2 Object characteristics and related hypothesis testings

In order to distinguish one object from an other, each object must be characterized by a set of features which allow to identify it uniquely. Three types of uncorrelated information are considered which permit to distinctively characterize each object. They are respectively the temporal, the spatial and the spatio-temporal information. The temporal information results from the object motion, while the spatial information arises from its shape and luminance properties. The spatio-temporal information relates to the object trajectory. Indeed, its spatial and temporal evolutions are correlated, meaning that the object is expected to appear at a given position at each successive time. Each type of information is described through representative parameters. The identification of an object relies on the parameter values and is carried out in the framework of hypothesis testing. To that end, a test statistic for each type of information is proposed.

6.4.3 The temporal identification

When considering an object, its related temporal information constitutes a distinctive feature to recognize it. Similar to Sec. 5.5, the affine motion model is chosen to represent the temporal information. For each object, a motion estimation procedure is thus carried out in order to determine the parameters of the affine motion model. Based on these, the null hypothesis, $H_0(temporal)$, that the object L of the current frame has the same motion as the object K from a previous frame is tested. The alternative hypothesis, $H_1(temporal)$, simply negates $H_0(temporal)$. In order to obtain the significance level of $H_0(temporal)$, the tracking algorithm relies on the Modified Kolmogorov-Smirnov test statistic, Q_{MKS}, presented in Sec. 5.3.4. This choice is motivated by the robustness of the test and its capacity to exploit all the

available motion information.

6.4.4 The spatial identification

Similar to the temporal information, the spatial information is important when identifying an object. This information is typically related to the shape and the texture of the object. In the framework of the tracking algorithm, the spatial information must be represented by features which are invariant under the transformation induced by motion. In our case, affine invariants are thus required. In [138], a moment-based approach to $2D$ and $3D$ object recognition is presented. In particular, affine moment invariants are derived which are very well suited to capture the spatial characteristics of an object undergoing an affine transformation [88]. Details on the derivation of the invariants are given in Appendix C. The tracking algorithm characterizes each object through two sets, $\vec{V}_{Lum}$ and $\vec{V}_{Sha}$, of five invariants, $\vec{V}_{Lum} = \{V_{Lum(1)}, V_{Lum(2)}, ..., V_{Lum(5)}\}$ and $\vec{V}_{Sha} = \{V_{Sha(1)}, V_{Sha(2)}, ..., V_{Sha(5)}\}$. The set $\vec{V}_{Lum}$ takes into account the luminance of the object while the set $\vec{V}_{Sha}$ only considers its shape. The motivation for selecting these five invariants is that they involve low order moments. This ensures the stability of the invariants even in case of small objects.

The sets $\vec{V}_{Lum}$ and $\vec{V}_{Sha}$ are used to test the null hypothesis, $H_0(spatial)$, of spatial correspondence. This hypothesis states that the object L of the current frame has the same spatial characteristics as the object K from a previous frame. Note that the hypothesis $H_0(spatial)$ is independently tested with respectively $\vec{V}_{Lum}$ and $\vec{V}_{Sha}$. It is accepted only in case both sets of data corroborate the hypothesis. The alternative hypothesis, $H_1(spatial)$, simply negates $H_0(spatial)$. In order to test the validity of $H_0(spatial)$, its significance level must be determined. To that end, a similar approach as the one described in Sec. 5.3.6 is adopted. Taking the case of $\vec{V}_{Lum}$, each of its components $V_{Lum(i)}$, $i \in [1, ..., 5]$, is modeled as a Gaussian random variable, while the corresponding test statistic is chosen to be the likelihood ratio test. In terms of $V_{Lum(i)}$, the significance level of $H_0(spatial)$ is given by Eq. (5.17), where $q(i)$ is defined as the difference of the $V_{Lum(i)}$ values between the two objects K and L. The standard deviation $\sigma(i)$ is estimated on the population composed by the parameters $V_{Lum(i)}$ of all the detected objects. The final significance according to the ensemble of invariants $\vec{V}_{Lum}$, is obtained through Eq. (5.18). A similar procedure is followed to obtain the significance level of $H_0(spatial)$ according to the set $\vec{V}_{Sha}$.

As previously said, the acceptance of the hypothesis $H_0(spatial)$ relies on the significance level derived from both the data in $\vec{V}_{Lum}$ and $\vec{V}_{Sha}$. This permits to ensure that the spatial correspondence testing between the objects K and L takes into account both luminance and shape information. However, a third check is necessary to evaluate the hypothesis $H_0(spatial)$. This check is performed on the surface disparity existing between the two objects. This additional condition is necessary due to the affine invariance of our object spatial characterization. Two objects may indeed very well have the same invariants although one is a scaled version of the other. In that case, the surface discrepancy data is useful to determine the validity of $H_0(spatial)$. To that end, the object K is projected onto the current frame and its resulting surface is measured. This is compared with the surface of the object L through the computation of the absolute value of the relative difference. In case the difference is above a predefined threshold, the hypothesis $H_0(spatial)$ is refused.

6.4.5 The spatio-temporal identification

When observing an object, its related temporal and spatial information are totally independent. There is indeed no relationship between the motion of an object and its spatial characteristics. However, the

object is expected to follow a certain route through time. In other terms, the object is expected to appear in given positions for given times (e.g. in successive frames). This is directly related to the hypothesis of motion temporal continuity described in Sec. 2.2.2.

The expectation of the object following a certain trajectory directly translates into a spatio-temporal hypothesis. The null hypothesis, $H_0(spatio-temporal)$, states that the object L in the current frame has the same trajectory of the object K from a previous frame. The alternative hypothesis, $H_1(spatio-temporal)$, simply negates $H_0(spatio-temporal)$. The null hypothesis may be further split into two hypotheses, $H_0(overseg)$ and $H_0(underseg)$. The hypothesis $H_0(overseg)$ defends the idea that the object L is simply a part of the object K. In other words, the current frame is oversegmented. Conversely, the hypothesis $H_0(underseg)$ says that the object K is a part of the object L. This means that the frame where the object K was last detected may have been oversegmented or that the current frame is undersegmented.

The significance levels of both the hypotheses $H_0(overseg)$ and $H_0(underseg)$ are obtained on the basis of the respective covering ratios of objects K and L. These ratios indeed allow to determine whether the two objects are at the same location in the current frame and, hence, whether their trajectories match. In the case of the hypothesis $H_0(overseg)$, the covering ratio C_{LK} is computed. It is defined as the proportion of the object L which is covered by the projection of the object K onto the current frame. In order to have a non zero value for C_{LK}, the object K must be the object whose projection mostly covers the object L. The significance level of the null hypothesis $H_0(overseg)$ is taken to be directly proportional to the value of the ratio C_{LK}. In case the latter reaches its maximum value of one, the null hypothesis $H_0(overseg)$ is given a significance level of 100%. Conversely the latter is 0% when the ratio C_{LK} is zero. A similar approach allows to derive the significance level of the null hypothesis $H_0(underseg)$. The only difference lies in the fact that the selected covering ratio is C_{KL}. It is defined as the proportion of the object K which, after projection onto the current frame, is covered by the object L. As for C_{LK}, the ratio C_{KL} is different from zero only if L is the object which mostly covers the projection of K.

6.4.6 Hierarchy of correspondence hypotheses

The task of the object tracking procedure is to identify the objects forming the current frame, i.e. the object in $\mathcal{O}(t)$. For each of these, the identification procedure consists of determining whether it corresponds to an object that was detected previously, i.e. an object in $\bigcup_{j=0}^{j=t-1} \mathcal{O}(j)$. In order to solve this correspondence problem, each object must be recognizable. To that end, the features presented in Sec. 6.4.2 are used. More precisely, the features and their related correspondence hypotheses are utilized. These are respectively the hypothesis of temporal correspondence, $H_0(temporal)$, the hypothesis of spatial correspondence, $H_0(spatial)$, the hypothesis of oversegmentation in the current frame, $H_0(overseg)$, and the hypothesis of undersegmentation in the current frame, $H_0(underseg)$.

The set of hypotheses $H_0(temporal)$, $H_0(spatial)$, $H_0(overseg)$, $H_0(underseg)$, have to be combined so as to encompass the whole range of possible correspondence situations. Furthermore, a hierarchy of correspondence situations must be established. Typically, the more constraining a correspondence situation is, the earlier it should be verified. The tracking algorithm must also be empowered to change the current segmentation. Based on the information it possesses, it may indeed judge necessary to create or destroy certain objects in $\mathcal{O}(t)$.

An overview of the proposed hierarchy of correspondence situations is given in Fig. 6.5. It is composed of three main stages. The first stage deals with the natural correspondences. It tackles the cases where no creation of new objects in the current spatio-temporal segmentation is required. In case the first stage

does not identify all the objects in the current segmentation, the second stage is applied. This checks whether the current segmentation is incomplete. For example, this may occur when an object detected in past frames stops moving and is assimilated into the background by the current segmentation. The second stage also aims at fixing the situations where objects are erroneously lost in the current segmentation. The third stage gathers the information from the first and second stages. Based on this, it determines whether new objects have appeared and whether objects have disappeared. The final correspondence information permits to obtain the tracked segmentation of the current frame. This is done by relabeling the current objects according to the labels of the objects they are put into correspondence with.

The first stage of the proposed correspondence hierarchy involves the objects in the current segmentation which have obvious relationships with previously detected objects. Four correspondence situations are checked sequentially in the order of decreasing requirements. In the tracking jargon, the object L in the current segmentation is called a *child*, while the object K which was previously detected is referred to as a *father*. If the child object L is identified and put into correspondence with the father object K, the former is said to *have found a father*, while the latter is said to have *found a child*. The four situations of natural correspondence are, by order of testing:

- **Situation 1**: Let us consider the father object K which has no child and the child object L which has no father. The hypotheses $H_0(temporal)$, $H_0(spatial)$, $H_0(underseg)$ and $H_0(overseg)$ are all accepted. The child object L is therefore assumed to perfectly correspond to the father object K.
- **Situation 2**: Let us consider the father object K with no child and the child object L with no father. While the hypothesis $H_0(temporal)$ is not accepted, the hypotheses $H_0(spatial)$, $H_0(underseg)$ and $H_0(overseg)$ are. This situation corresponds to the case where the child object L corresponds to the father object K, although there exists a significant change in the motion.
- **Situation 3**: Let us consider the father object K which may already have children and the child object L with no father. Say the hypotheses $H_0(spatial)$ and $H_0(underseg)$ are not accepted, but the hypotheses $H_0(temporal)$ and $H_0(overseg)$ are. This situation corresponds to an oversegmentation or an occlusion in the current frame. The child object L is a part of the father object K and is therefore put into correspondence with it.
- **Situation 4**: Again, we consider the father object K with no child and the child object L with no father. This time, say the hypotheses $H_0(temporal)$, $H_0(underseg)$ and $H_0(overseg)$ are not accepted, but the hypothesis $H_0(spatial)$ is. This case clearly corresponds to the object L having performed a maneuver. The abrupt change in its motion has fooled the hypotheses based on temporal and spatio-temporal information. However, the hypothesis testing on spatial information is able to recognize the object. The child object L is therefore put into correspondence with the father object K.

An example of these different situations of natural correspondence is given in Fig. 6.6. For the sake of simplicity, the potential father objects K are limited to those forming the previous frame. There are four potential father objects: the background, the arm, the ball and the hand with the bat. The recursive spatio-temporal segmentation of the current frame is composed of five objects. These are the background, the arm, the ball, while the hand is split into two parts, one of those being linked with the bat. After tracking, the segmentation of the current frame is composed of four objects, all of them having been put into correspondence with the respective natural fathers of the previous frame. More precisely, the background and the arm have been identified through the correspondence **situation 1**. Due to the motion acceleration, the ball is identified through the correspondence **situation 2**. Finally, the two parts of the hand have been merged. Indeed, the correspondence **situation 3** has detected that both

correspond to the object of the previous frame composed by the hand and the bat. In this illustration, the correspondence **situation 4** is not needed as all the objects are identified through the first three correspondence tests.

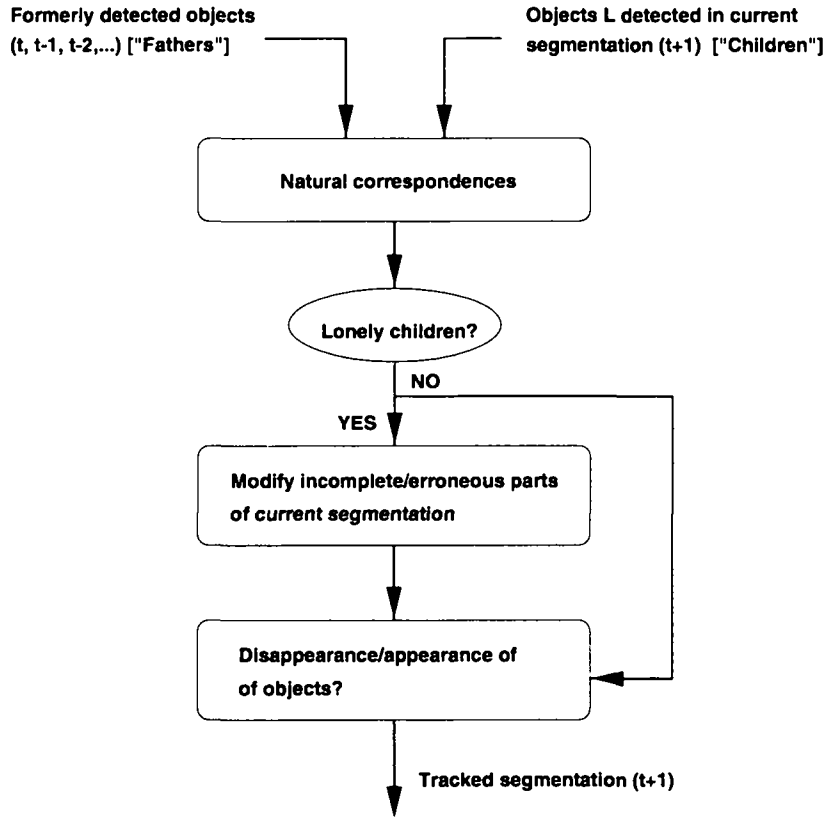

Figure 6.5: Hierarchy of correspondence hypotheses. In the first step, situations involving clear correspondences are tackled. These situations are referred to as natural correspondences. In the next step, the current segmentation may be altered by the tracking algorithm in order to recreate lost objects. The last step consists in checking whether new objects are present or objects have disappeared.

It may happen that, after checking the situations of natural correspondence (i.e. the first stage of the correspondence hierarchy), one or more child objects L have not been put into correspondence with any fathers. In that case, the second stage of the correspondence hierarchy is applied. Its task is to verify whether the current segmentation is too rough, and if so, to refine it. To that end, two correspondence situations are sequentially checked. They are referred to as respectively the **incomplete segmentation** situation and **erroneous segmentation** situation. The former situation is first checked and, in case lonely child objects L still remain, the **erroneous segmentation** situation is checked. As both situations may entail a modification of the current segmentation, each of them is followed by the check of the natural correspondence situations, (i.e. **situation 1**, **situation 2**, **situation 3** and **situation 4**). This permits to address natural correspondence situations that may appear through the modifications brought to the current spatio-temporal segmentation. Both the **incomplete segmentation** and the **erroneous segmentation** situations are iteratively checked as long as they modify the current segmentation. An overview of the second stage of the correspondence hierarchy is presented in Fig. 6.7. The **incomplete segmentation** and **erroneous segmentation** situations are now explained in greater detail.

- **Incomplete segmentation situation**: Let us consider the father object K with no child and the child object L with no father. Assume that the hypotheses $H_0(spatial)$ and $H_0(overseg)$ are not

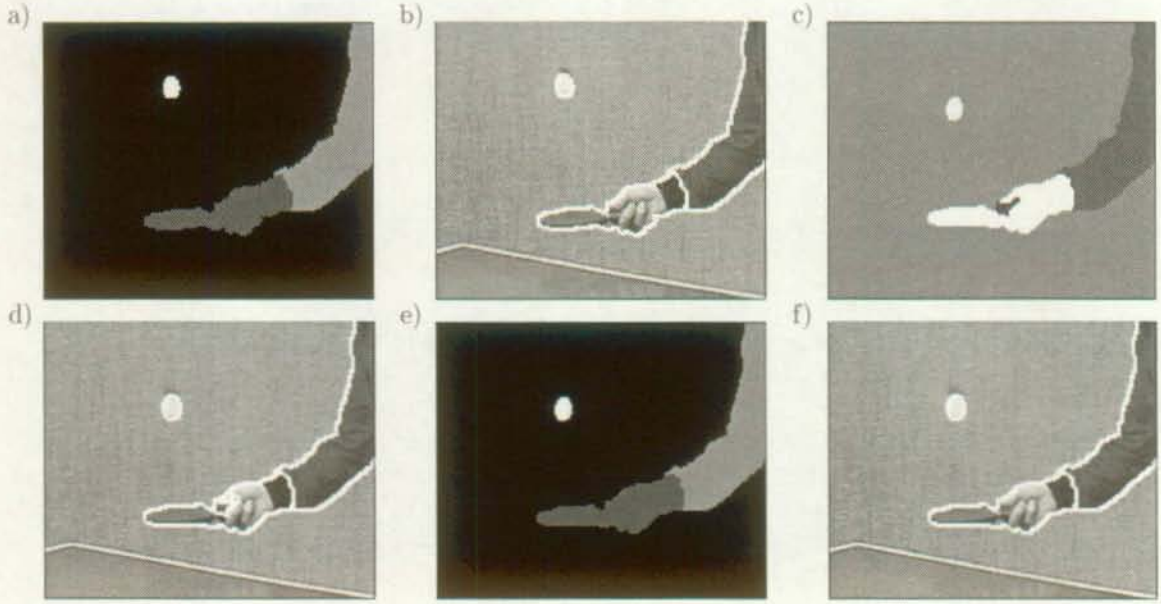

Figure 6.6: "Table Tennis": Example of the correspondence situations 1, 2, 3 and 4. **(a)** Set of objects $\mathcal{O}(t-1)$ composing the previous frame, **(b)** boundaries of the objects $\mathcal{O}(t-1)$ superimposed onto the previous frame, **(c)** partition of the current frame through recursive spatio-temporal segmentation, **(d)** boundaries of the partition obtained through recursive spatio-temporal segmentation superimposed onto the current frame, **(e)** final set of objects $\mathcal{O}(t)$ describing the current frame, **(f)** boundaries of the objects $\mathcal{O}(t)$ superimposed onto the current frame.

accepted, while the hypotheses $H_0(temporal)$ and $H_0(underseg)$ are. This situation may hint that the current segmentation is incomplete. More precisely, an object K may have stopped moving as an independent entity and was, rightfully, totally or partly merged with the child object L. However, there might be an interest in keeping the object which disappeared. In this work, this decision is based on the size ratio between the projection of the father object K and the child object L. If the ratio is smaller than a preset threshold, the hypothesis of oversegmentation in the previous frame is accepted. In case the ratio is bigger than the threshold, the hypothesis that the current segmentation is too rough is tested. To that end, a new object is tentatively created in the current segmentation. This is achieved by projecting the father K onto it, the projection procedure being similar to the one detailed in Sec. 6.3.2. At this point, the tentative new object and the father object K are submitted to the spatial correspondence hypothesis, $H_0(spatial)$. In case of acceptance, the newly created object is kept as a refinement of the current segmentation and is put into correspondence with the father object K.

- **Erroneous segmentation situation**: Let us consider the father object K with no child and the child object L which has no father. While the hypotheses $H_0(temporal)$, $H_0(spatial)$ and $H_0(overseg)$ are not accepted, the hypothesis $H_0(underseg)$ is. This situation may hint that the current segmentation is erroneous. In other words, the current segmentation may be too rough and the natural child of K may have been mistakenly merged with the child object L. In order to test this hypothesis, a new object is tentatively created in the current segmentation. This is achieved by projecting the father K onto the current frame through a procedure similar to the one detailed in Sec. 6.3.2. At this point, the newly defined object and the father object K are submitted to both the temporal and spatial correspondence hypotheses, $H_0(temporal)$ and $H_0(spatial)$. In case these hypotheses are accepted, the newly created object is considered a refinement of the current

segmentation and is put into correspondence with the father object K.

Figure 6.8 presents an example of the second stage of the correspondence hierarchy. For the sake of illustration simplicity, the potential father objects K are limited to those forming the previous frame. The potential father objects are the background and the woman. However, the recursive spatio-temporal segmentation procedure is unable to correctly segment the current frame. The frame is seen as being formed by a single object encompassing both the background and the woman. The object tracking algorithm detects the loss of the two objects. After verifying the correspondence hypotheses, it modifies the current segmentation in order to recreate both the woman and the background.

The third stage of the correspondence hierarchy collects the information derived in the first and second stages. Based on that, the third stage relabels the child objects L according to the label of their respective fathers K. It also detects whether any child object L has not found any father. This situation corresponds to the appearance of a new object, the latter receiving a new distinctive label number. Finally, any father object K which has not found any child is detected. The interpretation of this case may vary. If the father object K is present in the previous frame, we are confronted with the disappearance of an object. If the father object K is not present in the previous frame, the tracking algorithm simply states that the object K has still not reappeared. This last case occurs because the correspondence situations are tested between the objects in the current frame, i.e. the objects in $\mathcal{O}(t)$, and the ensemble of objects previously detected, i.e. the objects in $\bigcup_{j=0}^{j=t-1} \mathcal{O}(j)$. The latter ensemble contains objects which have been absent for many frames. However, they are kept in memory in case they reappear in the sequence. In case of disappearance, the object motion and its likely position are computed for each successive frame. This is achieved through a predictive Kalman filtering which predicts the trajectory of the object.

6.5 Simulation results

6.5.1 Introduction

In this section, the proposed algorithms for recursive spatio-temporal segmentation and object tracking are assessed. Their ability to detect and track multiple objects simultaneously is tested. The algorithms should also be able to modify the set of objects $\mathcal{O}(t)$ composing the scene. Depending on the situations, obsolete objects should be removed or new objects should be created. Finally, the efficiency of the memory that the object tracking algorithm possesses about each object should be demonstrated. In the following, the proposed algorithms are simply referred to as the object tracking algorithm.

Results are presented for three video sequences: "Akiyo", "Table Tennis" and "Foreman". These video sequences are described in details in Appendix A. In all the simulations, the first spatio-temporal segmentation is provided by the algorithm presented in Chapter 5. With respect to the motion estimation issue, the approach described in Sec. 5.5.3 is used. The motion is characterized through an affine model, while the estimator is the MAD. The recursive spatio-temporal segmentation uniquely relies on the temporal information arising from the local motions. Similarly, only the local motion information is used to characterize temporally each object. Indeed, the local motion represents the proper temporal feature of the object. This is not the case for the global motion. Arising from the motion of the camera, the global motion is indeed totally independent from the intrinsic motion of the different objects.

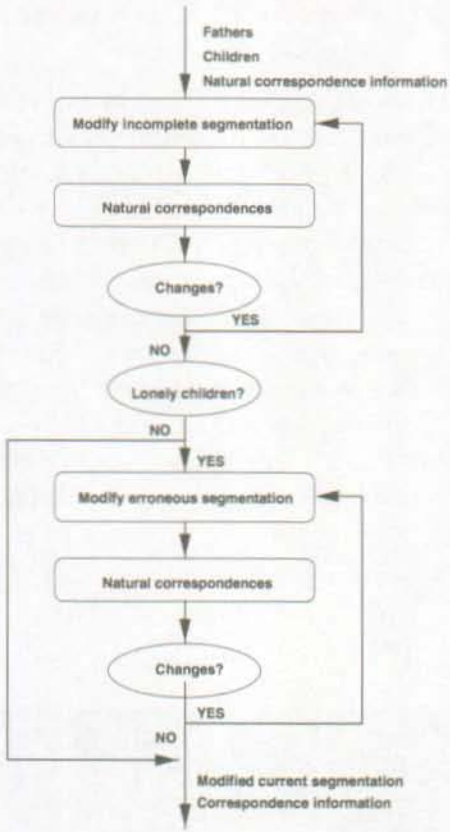

Figure 6.7: The second stage of the correspondence hierarchy. The current segmentation is modified if cases of incomplete or erroneous segmentation are detected. Each modification of the current segmentation is followed by a check of the natural correspondences (i.e. the first stage).

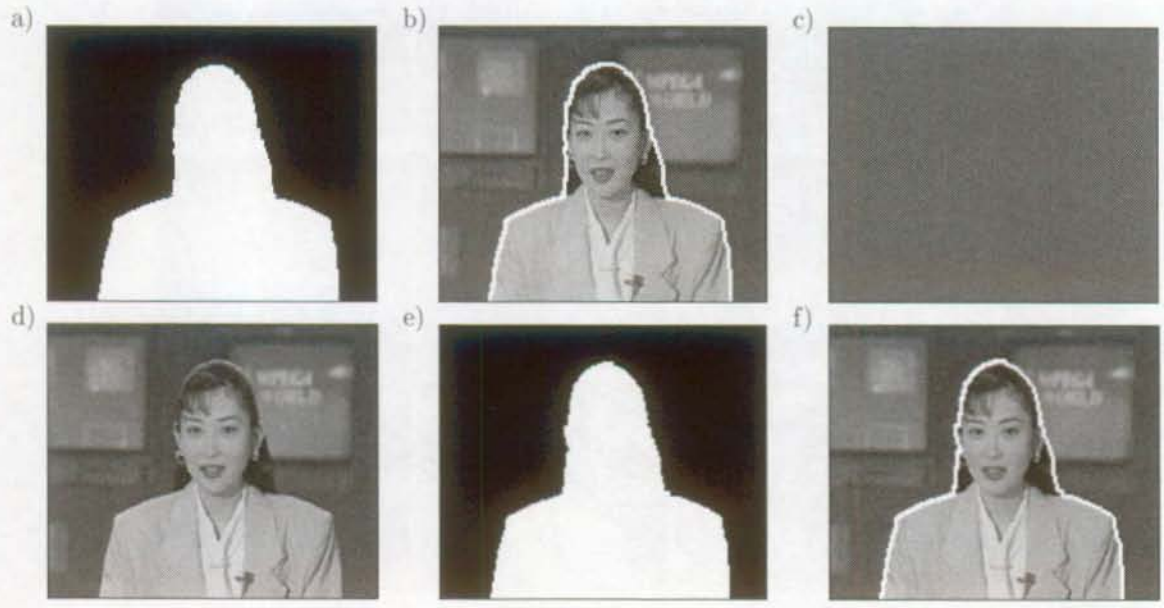

Figure 6.8: "Akiyo": Example of incomplete and erroneous segmentation situations. **(a)**, **(b)** Set of objects $\mathcal{O}(t-1)$ composing the previous frame, **(c)**, **(d)** partition of the current frame through recursive spatio-temporal segmentation and, **(e)**, **(f)** final set of objects $\mathcal{O}(t)$ describing the current frame.

6.5.2 Tracking the objects in different video sequences

When segmenting a sequence in terms of its objects, the partitions of the successive frames should show a high level of coherence. In particular, most of the objects should be found throughout the sequence. However, it may happen that the current frame does not hold enough information to properly segment the objects. For instance, an object may stop moving or have a very small motion. It is therefore at risk to be confoundedly put into the background. Figures 6.9 and 6.10 show the object tracking results for the video sequence "Akiyo" at time 1, 5, 9, and time 13, 16, 19 respectively. The initial spatio-temporal segmentation (i.e. time = 1) is provided by the algorithm presented in Chapter 5 (see Fig. 5.11(g)). The scene is decomposed into two objects: the woman and the background. This sequence is a typical example where a visually important object may be lost due to the lack of decisive information in certain frames. The body of the woman stays almost immobile during most of the sequence. Nevertheless, the object tracking algorithm clearly makes up for this lack of information. Both the woman and the background are robustly identified in all the frames. In terms of spatial resolution, the objects are sharply defined. In case of inaccuracies, the object tracking is able to correct for them in order to regain a proper shape of the object.

Figure 6.9: "Akiyo": Spatio-temporal segmentation and object tracking. **(a)** Frames at time 1, 5 and 9, **(b)** spatio-temporal segmentation at time 1, 5 and 9, **(c)** object "background" at time 1, 5 and 9, **(d)** object "woman" at time 1, 5 and 9.

Figure 6.10: "Akiyo": Spatio-temporal segmentation and object tracking. **(a)** Frames at time 13, 16 and 19, **(b)** spatio-temporal segmentation at time 13, 16 and 19, **(c)** object "background" at time 13, 16 and 19, **(d)** object "woman" at time 13, 16 and 19.

In order to fully demonstrate the capacity of the proposed object tracking algorithm, it must be applied to a scene with more than two objects. Indeed, the case where only two objects are present could be tackled through the concept of change detection. After having removed the global motion, the unchanged region would be identified as the background, while the changed region would be seen as the foreground object. Such an approach is however impossible to use in the presence of multiple moving objects in the foreground. Figures 6.11 and 6.12 show the object tracking results for the video sequence "Table Tennis" at time 1, 5, 9, and time 13, 16 and 19 respectively. The initial spatio-temporal segmentation (i.e. time = 1) is provided by the algorithm presented in Chapter 5 (see Fig. 5.12(g)). The scene is decomposed into five objects: the background, the arm, the ball, the right hand with the bat and, the left hand. This last object is not explicitly shown in the simulation results. The left hand indeed disappears as the sequence unravels. The object tracking algorithm allows for this disappearance and keeps on tracking the four remaining objects. These are perfectly identified throughout the sequence. In particular, the object tracking algorithm is capable to follow the objects despite of their maneuvers. For instance, the ball briskly changes its motion and position after hitting the bat. Although the temporal characteristics of the ball have suddenly changed, the tracking algorithm is able to identify it through its spatial features. The same remark holds for the arm and the hand with the bat. Finally, notice the perfect spatial definition of the objects. Again, the algorithm is able to correct for possible spatial inaccuracies in the object definition.

In order to implement the proposed object tracking algorithm, a number of hypotheses have been made. In particular, the objects are assumed to undergo a rigid body motion which may be characterized through an affine motion model. However, it may happen that this hypothesis constitutes a very coarse approximation. The object tracking algorithm should nevertheless be able to robustly segment and track the objects. Figures 6.13 and 6.14 show the object tracking results for the video sequence "Foreman" at time 1, 4, 8, and time 9, 16 and 19 respectively. The initial spatio-temporal segmentation (i.e. time = 1) is provided by the algorithm presented in Chapter 5 (see Fig. 5.13(g)). The scene is decomposed into 5 objects: the background, the right part of the face and the neck, the left part of the face, the back of the helmet and, the rest of the helmet. This last object is assigned the back of the helmet already at time 3. Consequently, the back of the helmet disappears as a single object. It is not represented in the simulation results. With regard to the two objects forming the face, they are robustly tracked up to time 8. This is achieved despite the non rigid motions which occur as the man speaks. However, the object tracking decides at time 9, that the face and the helmet do form a single object. To that end, it creates a new object "man". From this time on, the scene is decomposed into the object "background" and the object "man". This is achieved despite the maneuver that the man performs as he rotates his head. Furthermore, the object tracking is robustly performed even though the hypothesis of planar surfaces implied by the affine motion model is, in this situation, a very coarse approximation.

6.5.3 Improving the spatio-temporal segmentation

When segmenting successive frames recursively, a balance between past and current information has to be found. The necessity of having coherent successive segmentations has to accommodate for the changes occuring at the level of the current frame. These changes have to be recognized and the current spatio-temporal segmentation has to be modified consequently. Moreover, the integration of information on multiple frames should allow for an improvement of the spatio-temporal segmentation. Objects that are oddly defined should disappear and be replaced by better defined objects.

A simulation result for the video sequence "Akiyo" is given in Fig. 6.15. Starting from time 1, the next three spatio-temporal segmentations are shown. Initially, the scene is decomposed into three objects: the background, the right part of the body with head and, the left part of the body. These three objects

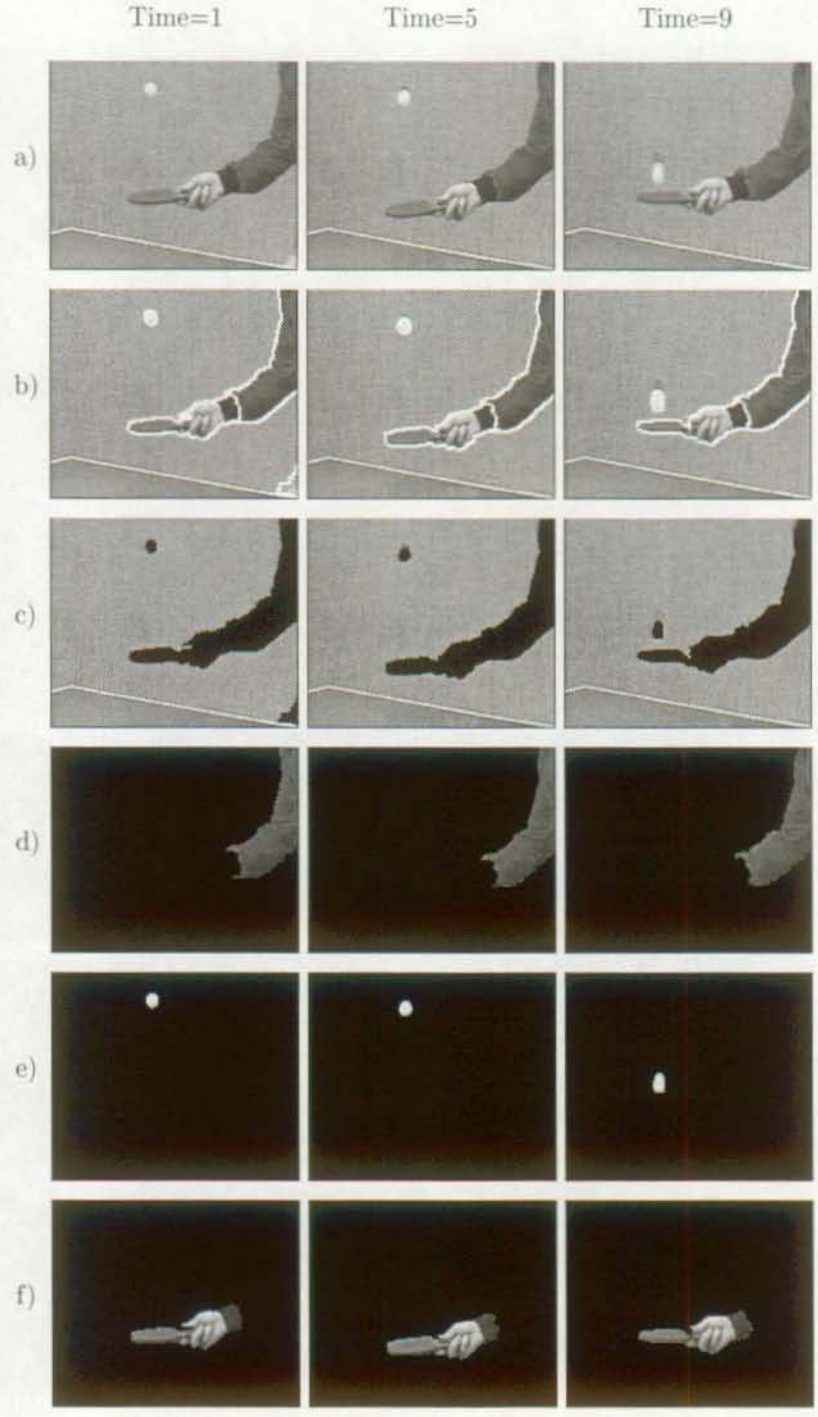

Figure 6.11: "Table Tennis": Spatio-temporal segmentation and object tracking. **(a)** Frames at time 1, 5 and 9, **(b)** spatio-temporal segmentation at time 1, 5 and 9, **(c)** object "background" at time 1, 5 and 9, **(d)** object "arm" at time 1, 5 and 9, **(e)** object "ball" at time 1, 5 and 9, **(f)** object "hand with bat" at time 1, 5 and 9.

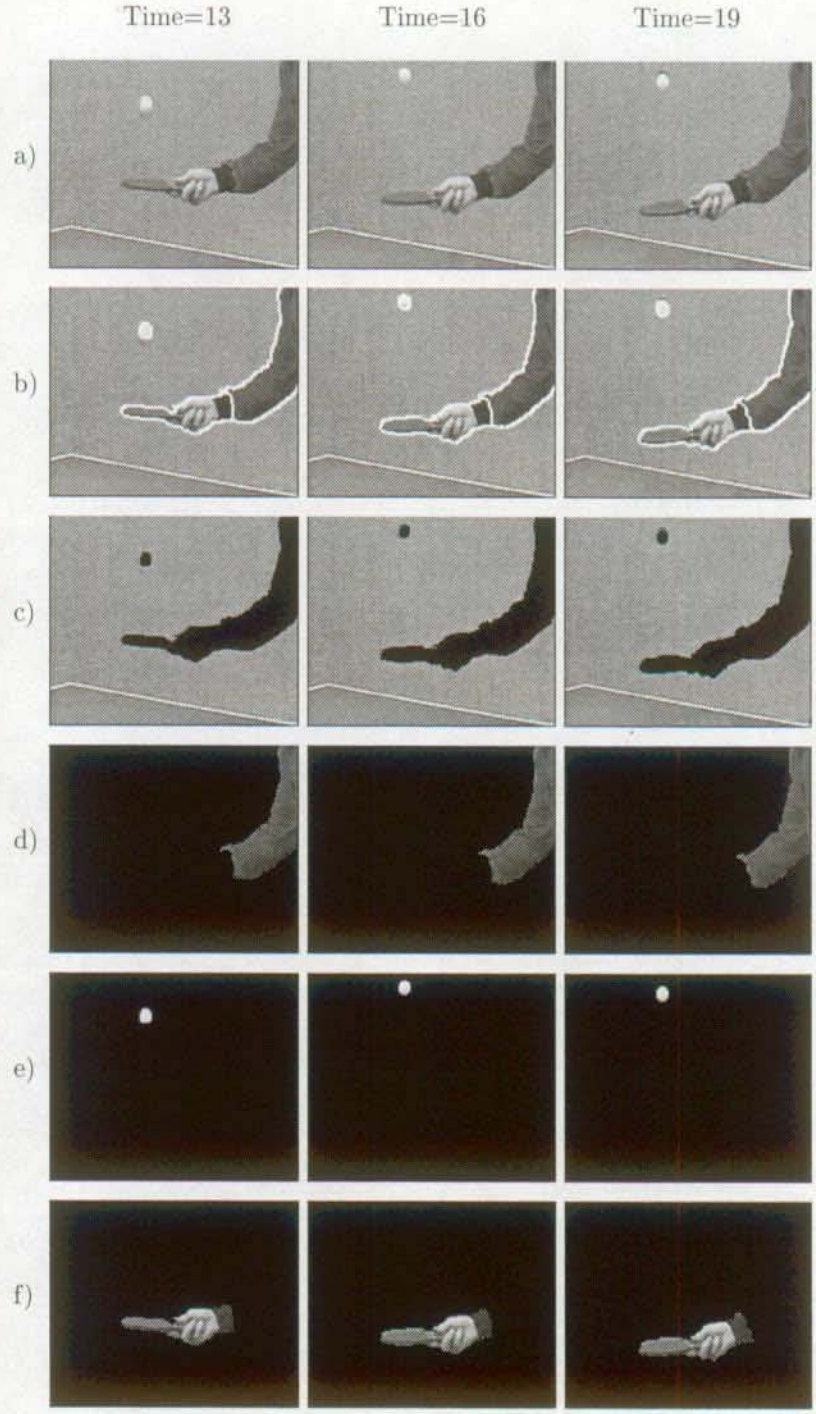

Figure 6.12: "Table Tennis": Spatio-temporal segmentation and object tracking. **(a)** Frames at time 13, 16 and 19, **(b)** spatio-temporal segmentation at time 13, 16 and 19, **(c)** object "background" at time 13, 16 and 19, **(d)** object "arm" at time 13, 16 and 19, **(e)** object "ball" at time 13, 16 and 19, **(f)** object "hand with bat" at time 13, 16 and 19.

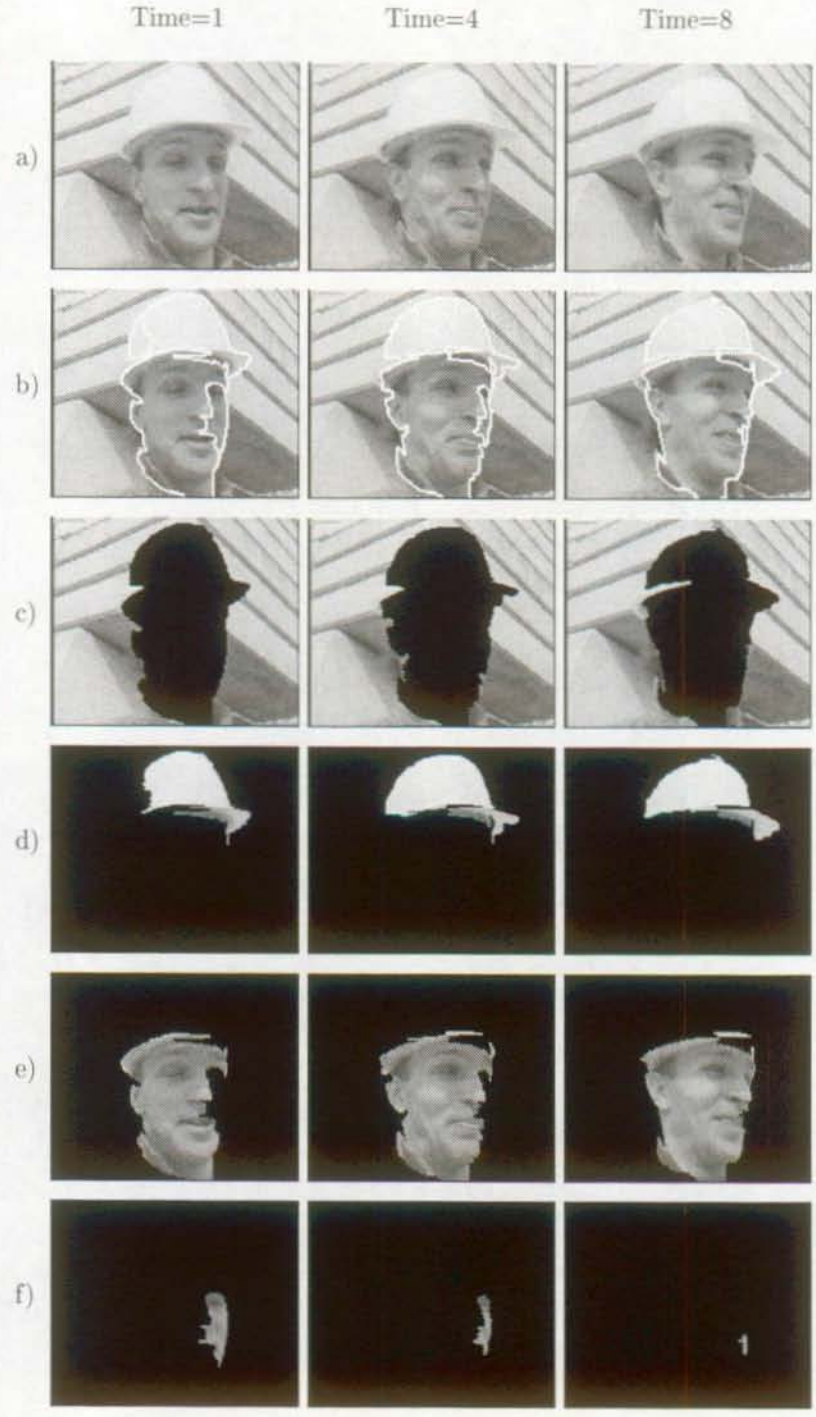

Figure 6.13: "Foreman": Spatio-temporal segmentation and object tracking. **(a)** Frames at time 1, 4 and 8, **(b)** spatio-temporal segmentation at time 1, 4 and 8, **(c)** object "background" at time 1, 4 and 8, **(d)** object "hat" at time 1, 4 and 8, **(e)** object "right part of the face" at time 1, 4 and 8, **(f)** object "left part of the face" at time 1, 4 and 8.

Figure 6.14: "Foreman": Spatio-temporal segmentation and object tracking. **(a)** Frames at time 9, 16 and 19, **(b)** spatio-temporal segmentation at time 9, 16 and 19, **(c)** object "background" at time 9, 16 and 19, **(d)** object "man" at time 9, 16 and 19.

are perfectly tracked till the third frame. However, the fourth frame sees the creation of a new object which represents the woman as a whole. The algorithm has indeed recognized that the two objects that split the woman during the three first frame should be replaced by a single one. These objects are thus put into the object memory and substituted by the new object "woman". The subsequent frames are robustly split into the objects "background" and "woman". Notice that another clear example of the improvement of the segmentation through tracking is given in the previous section of simulation results. Indeed, the tracking allows to define the man of the sequence "Foreman" as a single object, although it is split in different objects in the initial segmentation.

Figure 6.15: "Akiyo": Improving the spatio-temporal segmentation through tracking. **(a)** Frames at time 1, 2, 3 and 4, **(b)** spatio-temporal segmentation at time 1, 2, 3 and 4, **(c)** object "right part of the body with head" at time 1, 2 and 3, **(d)** object "left part of the body" at time 1, 2 and 3, **(e)** object "woman" at time 4.

6.5.4 Using the object memory

When tracking an object, it may happen that it disappears for a while and then reappears in the scene. The disappearance may be due to several factors. One reason may be that the object is momentarily totally hidden by another object of the scene. The disappearance may also result from an incomplete spatio-temporal segmentation. In particular, the object may stop moving and is consequently confounded as a part of the background. Nevertheless, the object tracking algorithm should not lose the memory of the object. In this way, the algorithm is able to recognize the object when it reappears.

Figures 6.16 and 6.17 show simulation results for the video sequence "Table Tennis" at time 1, 2, 3, and time 4, 5, 6 respectively. Initially, the scene is decomposed into five objects: the background, the arm, the ball, the right hand with the bat and, the left hand. They reduce to three at the third frame as the objects representing the left hand and the ball disappear. These two disappearances show very different characteristics. The left hand is naturally no longer in the set of objects because it has gone out of the scene. In contrast, the ball is still visible. Its disappearance is due to the slow-down in its motion. At the level of the third frame, the ball is seen as being static and is confounded with the background. As the ball regains speed, it is again distinguished from the background. This happens at the level of the fifth frame. Despite the time vacancy, the object tracking algorithm is able to recognize the new object. The object memory identifies the new object as the ball which had disappeared and permits its further tracking.

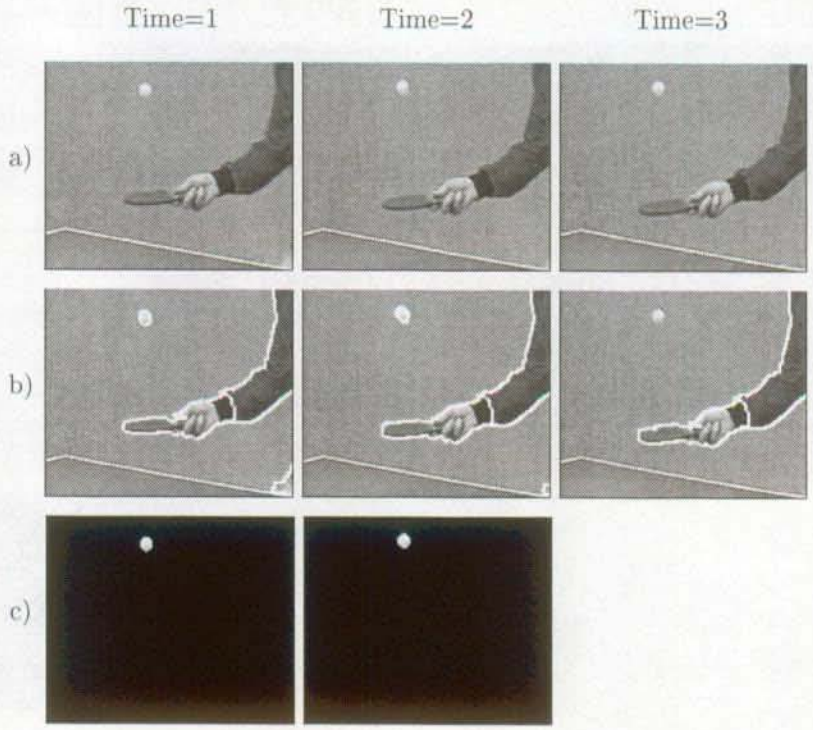

Figure 6.16: "Table Tennis": Using the object memory. **(a)** Frames at time 1, 2 and 3, **(b)** spatio-temporal segmentation at time 1, 2 and 3, **(c)** object "ball" at time 1, 2 and 3.

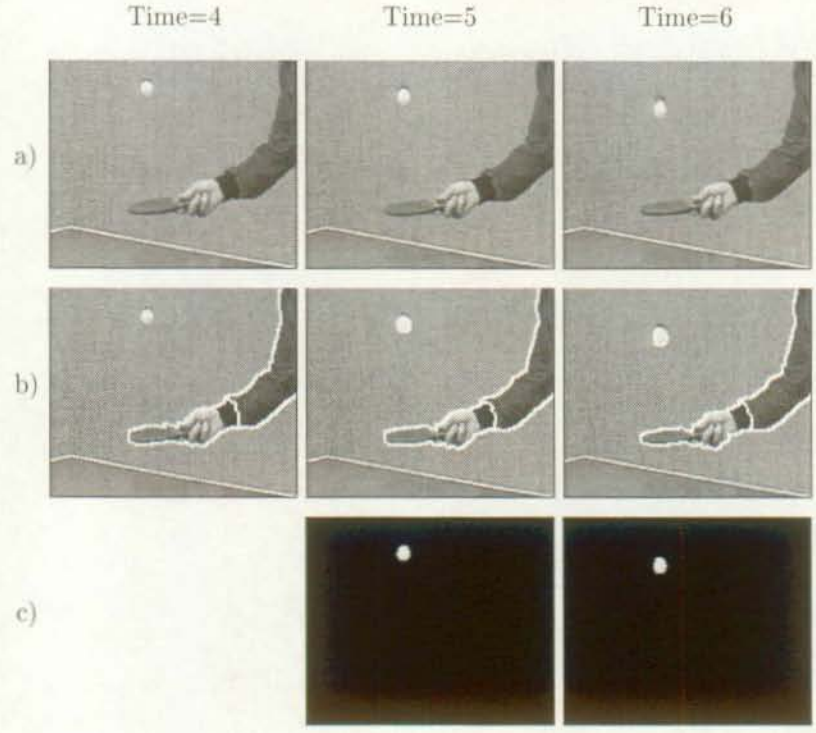

Figure 6.17: "Table Tennis": Using the object memory. (**a**) Frames at time 4, 5 and 6, (**b**) spatio-temporal segmentation at time 4, 5 and 6, (**c**) object "ball" at time 4, 5 and 6.

6.6 Conclusions

In this chapter, an unsupervised algorithm for recursive spatio-temporal segmentation and an unsupervised object tracking algorithm are presented. Not only is the spatio-temporal segmentation of the successive frames robustly derived, but also the objects forming the scene are tracked through time. Both algorithms deeply interact with one another. The recursive spatio-temporal segmentation procedure is formed of three steps: the partition projection, the region validation and the constrained spatio-temporal segmentation. The purpose is to combine efficiently past and current spatio-temporal information in order to robustly derive the partition of the current frame. This partition is used as input by the object tracking procedure. The objects are characterized through spatial, temporal and spatio-temporal features. These are used in the framework of a multiple hypotheses testing in order to solve the correspondence problem.

Experimental results on different video sequences demonstrate the efficiency of the proposed algorithms. The simulation results show that the spatio-temporal partitions of the successive frames are coherent with one another, while allowing for local changes to take place. Furthermore, the segmentation is improved by the object tracking algorithm. Also, the tracking algorithm is able to handle different types of objects simultaneously. It allows for objects to disappear as well as for new objects to appear. Finally, it is robust against momentary object disappearances and brisk maneuvers.

Chapter 7

Application to video sequence coding

7.1 Introduction

The importance of visual communication has increased tremendously in the last decades. The progress in micro-electronics and computer technology together with the creation of networks operating with various channel capacities is the basis of an infrastructure for a new era of telecommunications. New applications are preparing a revolution in everyday life of our modern society. Emerging applications such as video conferencing [125], cellular videophones [1] and multimedia [51, 36, 1] will have a great impact on today professional life, education and entertainment. However, the digital representation of the visual information in its canonic form leads to a huge amount of data. In order to meet the requirements of the new applications, powerful image sequence compression techniques are needed to drastically reduce the total bitrate. In the literature, two types of video coding schemes have been proposed. The techniques of the first type rely directly on a waveform representation of the video sequence and are commonly termed *first generation video coding* techniques. In contrast, an attempt to decompose the video sequence into visual primitives is pursued in the second category of video coding techniques. They are referred to as *second generation video coding* techniques.

First generation coding techniques have been mainly designed to optimize the compression performance. They are based on a waveform representation of the scene. The frames are decomposed into blocks of pixels which do not carry any semantic meaning. These are usually coded through predictive coding. More precisely, the temporal redundancy is reduced through motion compensation, while the spatial redundancy is diminished through transform coding. Belonging to the first class of techniques, standards have been defined. The standard MPEG-1 [74] operates at bitrates of about 1.5 Mbit/s and targets storage and transmission over communication channels as the integrated-services digital network (ISDN) or the local area network (LAN). The standard MPEG-2 [2] operates at bitrates around 10 Mbit/s and is designed for the compression of higher resolution video signals. The recommendation H.263 [77] was proposed by the International Telegraph and Telephone Consultative Committee (CCITT, now known as ITU-T). Based on this standard, video conferencing at bitrates less than 64 kbit/s has become feasible. The first generation coding techniques have proved their efficiency in terms of compression for a wide range of bitrates. However, they suffer from redhibitory drawbacks. Strong artifacts appear at low bitrates and no functionalities related to the semantic meaning of the scene are possible. Examples of such functionalities are object manipulation, object scalability and exact rate control for each object.

The second generation video coding techniques aim at overcoming the inherent limitations of the first generation video coding approach. To that end, the image is split into visual primitives such as contours or textures [87]. Expert groups have been created to pursue this objective. The major one is ISO/MPEG-4 [76]. In the literature, several attempts to define coding schemes based on this philosophy have been made [106, 128]. Two important examples of second generation video coding schemes have been proposed by Mussmann *et al.* and Salembier *et al.*, respectively.

Mussmann *et al.* [106, 59] have proposed an object based analysis-synthesis approach. It mainly addresses the coding of "head-shoulders" video sequences. In the first step, the scene is decomposed into arbitrarily shaped moving regions. This is achieved by detecting the areas in the image which show coherent motions. Each region is then described by three sets of parameters representative of respectively the shape, the motion and the texture information. The current frame is synthesized in two steps. First, the parameters describing the regions of the previous frame are used to synthesize an approximation of the current frame. In the second step, areas corresponding to an erroneous synthesis are detected. These areas are referred to as Motion Failure (MF) areas. The current frame synthesis is thus refined through the shape and color information of the MF areas. On the coding side, the information to be transmitted consists of two parts. On the one hand, the motion and shape parameters are transmitted for both the regions and the MF areas. On the other hand, texture information is only transmitted for the MF areas so as to reach

high coding efficiency.

Salembier *et al.* have proposed a second generation video coding scheme based on mathematical morphology [128]. The first frame of the video sequence is decomposed into regions showing spatial coherence. This is performed through the use of a modified *watershed* algorithm. After having coded the first frame and its segmentation, the segmentation of the successive frames relies on a predictive scheme. Consider the current frame at time t, the segmentation of the previous frame being known. The latter segmentation is projected onto the current frame, providing a first approximation of the current segmentation. This approximation serves as the starting point to derive the final partition. To that end, the modified *watershed* algorithm is used. Moreover, a temporal linkage between the partitions of successive frames is performed by forward motion projection. This aims at tracking the different regions through time. With respect to coding, the motion, the texture and the shape information of each region are transmitted. So as to decrease the amount of shape and texture information, a predictive coding scheme is applied.

However, the second generation video coding techniques described above do not rely on a semantic decomposition of the scene. In other words, the primitives do not correspond to spatio-temporally coherent objects. Therefore, no semantic correspondence between successive frames can be robustly established. This leads to a decreased compression performance and a major loss of functionalities. Also, the above described coding of the primitives does not allow for a scalable transmission together with an exact rate control.

7.2 The proposed video coding scheme

The proposed video coding scheme belongs to the class of the second generation video coding techniques. It is characterized by both its compression efficiency and its content-oriented functionalities. Furthermore, it is compatible with the current status of the MPEG-4 verification model [75]. The proposed video coding scheme relies on an image representation where the primitives are defined as being the objects forming the scene. The objects have their own conceptual meaning and correspond to perceptually meaningful objects. This semantic representation of the scene permits a complete interactivity with the visual information. This is in sharp contrast to classical region-based coding schemes where the regions have little semantic meaning since they are defined for compression purposes. On the coding side, the proposed coding scheme allows to address the compression of the sequence in terms of the objects forming the scene. For each object, the reduction of both temporal and spatial redundancies is thus performed independently from the other objects. In addition, the bitstream sent for each object allows for a progressive decoding of the texture information.

The main functionalities of the proposed coding scheme are the following:

- *Content-based object scalability:* Each object of the scene has its own internal representation and may be decoded independently.
- *Content-based coding scalability:* Each object is allocated an individual bitrate. The total bitrate is the sum of all the object bitrates.
- *Object-based spatial scalability:* Each object bitstream is completely embedded and can be progressively decoded.
- *Object memory:* Each object is tracked through time allowing for thorough understanding of the scene. In particular this permits the building of an object memory.

The proposed video coding scheme decomposes the video sequence into its constituting objects. This is performed through the techniques presented in Chapter 6, the first frame being partitioned through the technique described in Chapter 5. The first frame of the sequence is coded in intra mode while the following frames are coded in predictive mode (i.e. inter mode). The predictive coding of each object works as follows. Its texture and its shape are predicted by using temporal information. To that end, an object based motion estimation is performed between the current frame and the respective sprite of each object. The resulting motion information is used to obtain a prediction of each object in the current frame. The object based motion estimation is described in greater detail in Sec. 7.3. The prediction errors for both the shape and the texture are then computed for each object. These errors are then encoded independently from the ones of the other objects. The coding of the texture information is described in Sec. 7.4, while the predictive shape coding is detailed in Sec. 7.5. Note that the shape is coded in a lossless fashion. The texture and the shape of each object being coded independently, it is possible to allocate a variable bitrate depending on the desired object coding quality. In order to take into account the loss of information due to the coding process, the generation of the object sprites is based on the decoded objects. The sprites are generated online by dynamically integrating the information of each object at successive times. This is explained in Sec. 7.6. Finally, Sec. 7.7 presents experimental results, while Sec. 7.8 draws the conclusions. An overview of the proposed video coding scheme is given in Fig. 7.1.

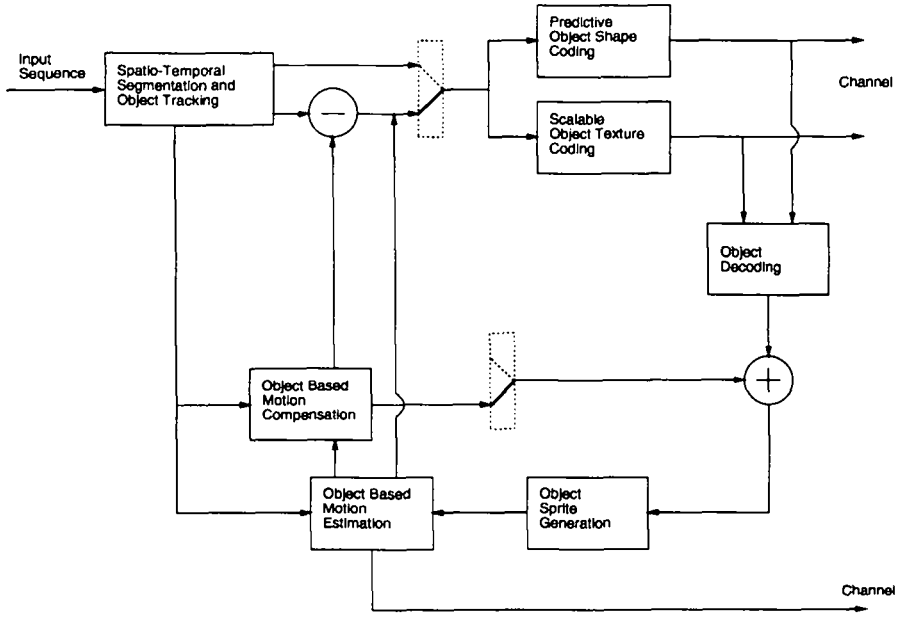

Figure 7.1: Proposed video coding scheme. The scene is first decomposed into objects. The shape, texture, motion and related DFD are coded for each object. This coding is performed independently from the other objects.

7.3 Object based motion estimation

The estimation of the motion of an object involves the choice of the motion model, the motion estimator and the estimation procedure. In order to be coherent with the procedure that defines the objects (see Chapter 6), the affine model is chosen to characterize the motion of the objects. With respect to the motion estimator, the MSE estimator is selected. The MSE is indeed directly related to the Peak Signal to Noise Ratio (PSNR) criterion through which the quality of the coded frames is assessed. Similar to Sec. 5.5.2, the motion estimation is carried out through a matching technique. The computational

complexity is reduced by embedding the estimation into a Gaussian pyramidal structure of the input frames.

The motion of each object is estimated on the basis of its respective sprite. In other terms, the motion estimation is carried out between the current frame and the sprite of the object. This permits to take advantage of the temporal integration that the sprite is performing. In particular, problems linked to disocclusion phenomena are tackled. Furthermore, the use of the object sprite as the support for the motion estimation has important implications in terms of functionality. Indeed, the use of the sprite permits to determine the motion of an object independently from the other objects in the scene. Consequently, the object may be decoded autonomously from the other objects.

By construction, the objects are mainly defined on the basis of their temporal coherence. However, it may happen that a single set of affine motion parameters is not sufficient in the framework of video coding. This situation arises from the use of the recursive spatio-temporal segmentation and object tracking algorithms (see Chapter 6). Indeed, as these algorithms strive for a coherent segmentation of the successive frames, objects in some frames may see their temporal homogeneity being affected. For these objects, the motion of the texture is consequently no longer satisfactorily represented by a single set of affine parameters. Practically, this situation is recognized by checking the energy (i.e. MSE) present in the DFD of the object. In case the energy is higher than a present threshold, the object is split through a quadtree segmentation. For each of the resulting blocks, a translational motion is then estimated. This splitting procedure allows for an optimization of the trade-off existing between motion information and DFD information when coding texture information. These motions and the corresponding quadtree are sent along the set of affine motion parameters. The latter set is still needed in order to carry out the predictive shape coding described in Sec. 7.5.

7.4 Scalable object texture coding

Given a region, the coding of its texture is always based on three main subsystems. The first subsystem performs a transformation of the input data. In most standards such as H.263 and MPEG-2, this is performed using a Discrete Cosine Transform (DCT) applied on N by N blocks. The second subsystem performs the quantization of the transformed coefficients, while the third subsystem codes the quantized coefficients.

In the literature [136, 49], the suboptimality of the DCT transformation block has been demonstrated leading to the definition of other approaches to texture coding. Among the latter, the transformation referred to as *subband decomposition* has received much attention. In the proposed video coding scheme, the texture coding is performed through the subband decomposition proposed by Egger [47]. This transform is used for both intra mode texture coding and inter mode (i.e. DFD) texture coding. The proposed texture coding has the following characteristics [48]:

- The wavelet transform is optimized for coding arbitrarily-shaped objects.
- The absence of information across scales is predicted. This is achieved using trees of zero-valued coefficients.
- A successive approximation quantization procedure of the wavelet coefficients is performed.
- The quantized symbols are coded using adaptive arithmetic coding.

The wavelet transform performs the decorrelation of the data into a multiresolution structure using a general subband transform. The transform is performed in a separable way as in conventional decomposition schemes. The operation goes as follows. As a first step, the lines are processed. For each line all pixels belonging to the region are written in a vector. This vector is then decomposed into N bands by means of a filter bank. A filter bank requires the input to have a size being a multiple of the number of samples. In order to overcome this problem, these extra pixels are not processed in the filter bank but, are put in the lowpass subband after being multiplied by the gain of the lowpass analysis filter. Note that, for $N = 2$, there is a maximum of one extra pixel per line. This procedure is illustrated in Fig. 7.2 for a two-band filter bank (i.e. $N = 2$). It is important to mention that linear phase filters are appropriate for this decomposition because they allow for a symmetric extension of each line. This reduces greatly the border effects which are introduced otherwise.

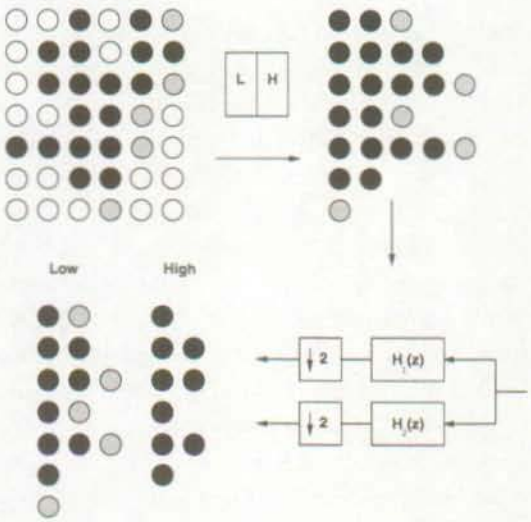

Figure 7.2: Decomposition of a region into two subbands. Black circles are pixels belonging to the region. Grey circles represent extra pixels which are not processed.

After the decomposition, the filtered samples have to be placed into the subbands such that each subband preserves the original shape of the region. This is important for the purpose of obtaining a maximum decorrelation through the filter bank. For every group of N pixels of the original region, the samples are evenly distributed into the N subbands. This is illustrated in Fig. 7.3. This procedure assures that the subbands are very near in shape to the original region scaled by a factor of N.

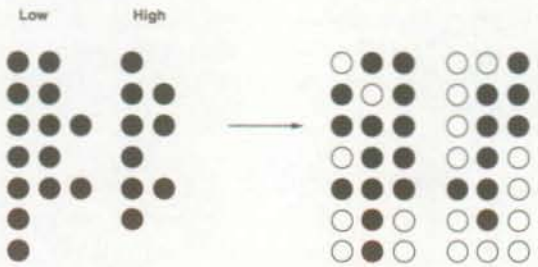

Figure 7.3: Placing of the decomposed samples in the two subbands.

After performing the subband transform, the zerotree prediction [136], in turn, allows for a further improvement of the energy compaction by taking into account the remaining intra-band correlation. Finally, a successive approximation quantization provides with an embedded bitstream and permits an exact rate control. The embedded bitstream together with the multiresolution structure allows the

additional feature of progressive transmission of each object.

7.5 Predictive object shape coding

In the literature, shape coding techniques have not received as much attention as texture coding techniques. Nevertheless a variety of techniques including quadtree, chain coding, and polygonal approximation techniques have been studied. However, none of these can easily be extended to efficiently code a sequence of binary masks representing the shape of each object of the scene. Therefore, a different approach to shape coding is taken here, and which is extended to exploit the temporal redundancy.

This approach is based on finite state machines and arithmetic coding [83]. The principle is quite simple: each binary mask is coded pixel by pixel in a scanline order. For each pixel, the state of the finite state machine is defined by the values of pixels within a template. This template typically includes pixels in the close vicinity of the pixel to be coded. With each state, a probability is associated which gives the likelihood that the current pixel belongs to the object. In turn, these probabilities are used to drive an arithmetic coder. In this dissertation, the approach proposed by Bossen and Ebrahimi [53] is adopted for intra mode coding (i.e. the first mask in the sequence). The template consists of ten pixels and is depicted in Fig. 7.4(a). For the encoding of the subsequent masks of the objects, the shape coding is performed in a predictive mode (i.e. inter mode). In doing so, the coding efficiency is improved by taking advantage of the temporal redundancy. Given a predicted object mask P, obtained by warping the previous mask with the motion parameters of the object, the current mask M is efficiently encoded using a template which includes pixels from both the prediction mask P and the current mask M. The template used for inter mode coding consists of five pixels from the prediction mask P and two pixels from the current mask M. It is depicted in Fig. 7.4(b). The proposed template allows to take advantage of the very strong correlation existing between the prediction mask P and the current mask M.

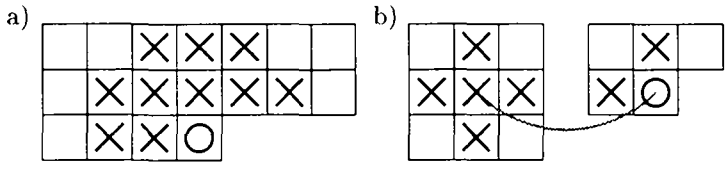

Figure 7.4: Templates used for coding the object shape information. The circle represents the pixel to be coded, and the crosses the pixels belonging to the templates. **(a)** Template used for intra mode coding, **(b)** template used for inter mode coding (the left side stands for the part of the template in the prediction mask P, while the right side represents the part of the template in the current mask M. The grey curve defines the alignment between the two parts of the template).

For a given object, the table of probabilities giving the likelihood that a pixel is in the object is used to drive an arithmetic coder. This table of probabilities is usually non constant as it adapts itself to the input data. However, there are several advantages in forcing the probabilities to be constant. First, less memory is required since less bookkeeping information needs to be stored. Then, the probabilities may be represented with fixed point numbers, which allow to use an arithmetic coder with no division operation, thus increasing computation speed. Representing all probabilities smaller than 0.5 with a power of two even yields a multiplication free and faster arithmetic coder.

For the intra mode coding, the constant probability table is derived from the analysis of several typical sequences. These probabilities are quantized and represented using 16-bit fixed point numbers. With

respect to the inter mode coding, the constant probability table arises directly from the object to be coded. Therefore, the table for each object needs to be coded as well. This is achieved by approximating each probability in the table by a power of two. More precisely, the approximation is carried out either on the probability itself or on its complement to one. The table can thus be efficiently encoded for transmission.

7.6 Online dynamic object sprite generation

Assuming that no scene change occurs, the successive frames of a video sequence are very similar in terms of their content. More precisely, the same set of objects may be found in the different frames. The main difference lies in the fact that these objects have change their relative positions. This is caused by the motion which animates each object. Also, the motion of the camera during the scene capture induces motion in the scene. An efficient integration of the motion information is a key component for video coding applications as well as for scene analysis purposes.

In video coding applications, the integration of the motion information has been proposed for the object representing the background of the scene. In doing so, the problem of coding the disoccluded background areas is overcome [43]. In the literature, two types of techniques have been described. The techniques of the first type build a background memory [65, 154]. These techniques identify still regions as background and store them in a long term memory. However, the model of a still background does no longer hold for more complex scenes which include camera motion (e.g. pan or zoom). The camera motion is taken into account by the techniques of the second type. These techniques build mosaic representations [71, 130] which are also referred to as salient still. Basically, these techniques estimate the camera motion through global motion estimation and align the frames in the sequence by canceling the contribution of camera motion. The mosaic is built by temporal integration of the aligned frames. However, the mosaic is constructed without distinction between background and foreground. Dufaux and Moscheni [43] propose to combine the ideas of background memory and mosaic representation. After defining the regions belonging to the background, a mosaic of only those regions is built dynamically. The mosaic content is dynamically updated by gradually integrating the information of the individual frames.

The idea of building a dynamic mosaic for the background may be extended to all the objects forming the scene. We refer to these mosaics as *sprites*. This subject has received much attention in the framework of MPEG-4 [75]. For each object, the generation of its corresponding sprite requires the identification of the object throughout the video sequence. In this way, only the information arising from the object is used to update the corresponding sprite. In the framework of the proposed video coding scheme, this identification procedure trivially arises from the scene representation that is used. Consider a frame where N_o objects O_j are present. For each object, its corresponding sprite is dynamically built by aligning its position in the different frames with respect to a coordinate system. We propose to take the position in the current frame as the time-varying reference. Let us define $\vec{M}_j$ as the set of parameters characterizing the motion of the object O_j between the current frame, at time t, and the previous frame, at time $t-1$. The sprite $S_j(\vec{r},t)$ of an object O_j is generated as follows [43]

$$S_j(\vec{r},t) = (1-\beta\delta(\vec{r}))\ S_j(T(\vec{r},\vec{M}_j),t-1) + \beta\delta(\vec{r})I(\vec{r},t)\ , \tag{7.1}$$

where β is a weighting factor which controls the update of the sprite, $\beta \in [0,1]$, $\delta(\vec{r}) = 1$ if $\vec{r}$ belongs to the object O_j and 0 otherwise. $I(\vec{r},t)$ is the pixel intensity value at position $\vec{r}$ and time t. Finally, $T(\vec{r},\vec{M}_j)$ is the position in $S_j(\vec{r},t-1)$ which corresponds, through motion compensation, to the position $\vec{r}$ in $S_j(\vec{r},t)$.

As expressed by Eq. (7.1), the proposed sprite generation permits a flexible integration of past and current information. The extent to which the previous sprite, $S_j(\vec{r}, t-1)$, influences the current one, $S_j(\vec{r}, t)$, depends on the factor β. In case β is unity, no previous information is used. Conversely, the case where β is zero signifies that no current information is used. If the current pixel $\vec{r}$ does not belong to the object of interest O_j in the current frame, its luminance value is computed only on the basis of the previous sprite.

7.7 Simulation results

In this section, we present simulation results for the proposed video coding scheme. Simulations are carried out on the video sequences "Table Tennis" and "Akiyo". First, the general behavior of the proposed video coding scheme is detailed for various bitrates. Then, the video coding standard H.263 [77] is taken as a benchmark to assess the compression efficiency of the proposed scheme. In addition, the functionalities offered by the proposed scheme to the end-user are illustrated.

A first assessment of the proposed video coding scheme is made using the video sequence "Table Tennis". The sequence involves multiple moving objects and is therefore representative of a natural scene. The scene is decomposed into five objects. These are the background, the arm, the right hand with the bat, the ball and the left hand. The video sequence "Table Tennis" has been compressed at respectively 64 kbits/s and 40 kbits/s. For both simulations, the evolutions of the respective global PSNR are given in Fig. 7.5, while examples of decoded frames are shown in Fig. 7.6. It can be seen that the objects are clearly defined and that no artifacts around the contours are visible. This is a direct consequence of the fact that the proposed video coding scheme relies on a decomposition of the scene into its constituting objects. Furthermore, the quality of each object has been chosen in order to reflect the importance of the object in the scene. In particular, the emphasis has been put on the foreground objects. These are the hand with the bat, the arm and the ball. Since the background is visually less important, it has been coded more coarsely. Due to this object-based bit allocation, the global PSNR is not representative of the visual quality of the decoded sequence. This is illustrated by comparing the global PSNR reported in Fig. 7.5 with the visual quality of the decoded frames (see Fig. 7.6).

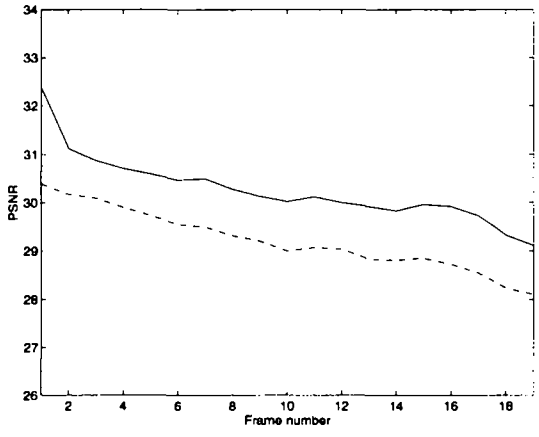

Figure 7.5: "Table Tennis": Evolution of the global PSNR at respectively 64 kbits/s (full line) and 40 kbits/s (dashed line).

In order to have a more meaningful assessment of the coding quality, the PSNR of each object is examined. Figure 7.7 compares the evolution of the PSNR for the object "background" and the object "hand with

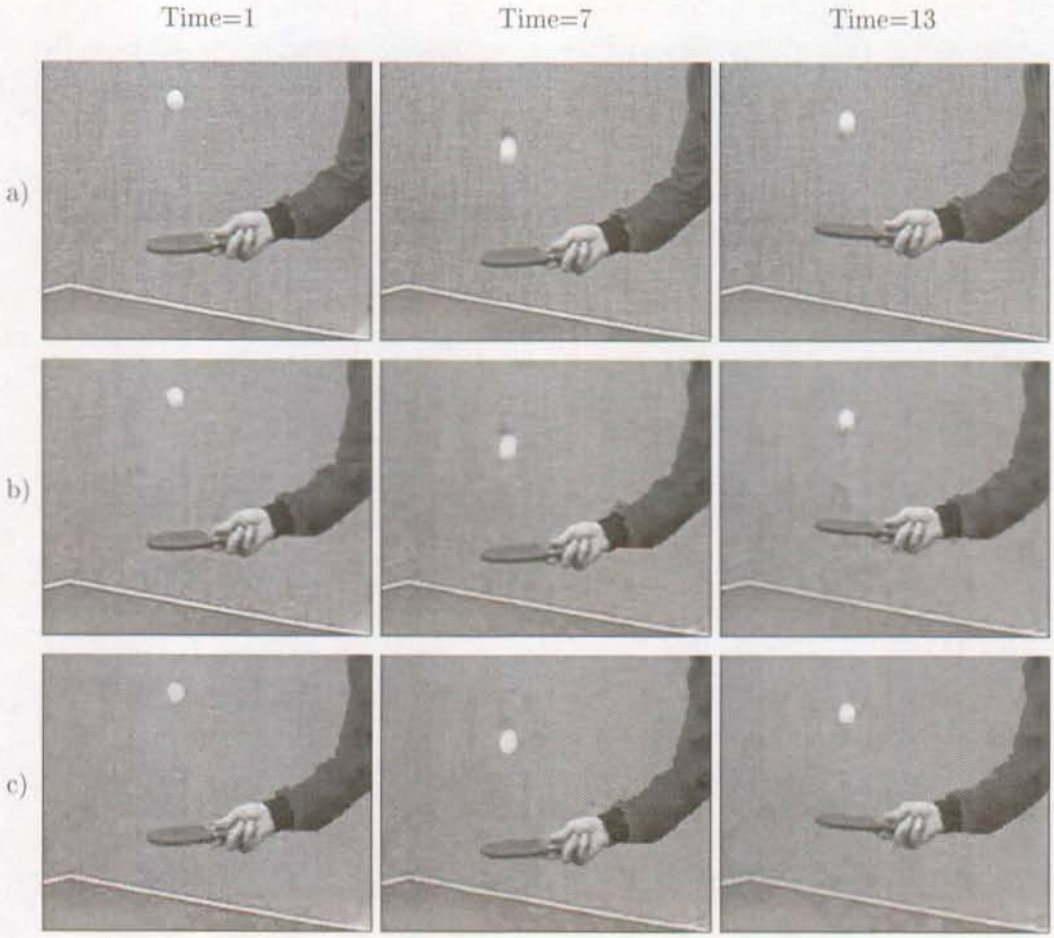

Figure 7.6: "Table Tennis": Coding results at different bitrates for the proposed video coding scheme. **(a)** Original frames at time 1, 7 and 13, **(b)** sequence coded at 64 kbits/s, **(c)** sequence coded at 40 kbits/s.

the bat" at a total bitrate of 64 kbits/s and 40 kbits/s, respectively. It can be clearly observed that the coding quality of the foreground object is much higher than the coding quality of the background, at a total bitrate of 64 kbits/s. This derives from the fact that the foreground objects are considered visually more important than the background, and therefore, receive a bigger share of the total bitrate for texture information. However, as the total bitrate decreases, the coding quality of the object "hand with the bat" decreases faster than the coding quality of the background. This can be noticed when comparing the PSNR of the hand and the background at a total bitrate of 40 kbits/s. Both objects have roughly the same PSNR, although the same share of the total bitrate for texture information has been allocated among the objects as for the simulation at 64 kbits/s. The latter phenomenon is explained as follows. The evolution of the background is predicted quite accurately through motion compensation and, therefore, little DFD has to be transmitted. In the case of the object "hand with bat", a larger DFD has to be sent. When the available bitrate to send the DFD information decreases, the quality of the object "hand with bat" is thus more strongly affected than the quality of the object "background". Furthermore, notice the big impact that the decreased bitrate has on the intra frame coding of the object "hand with bat".

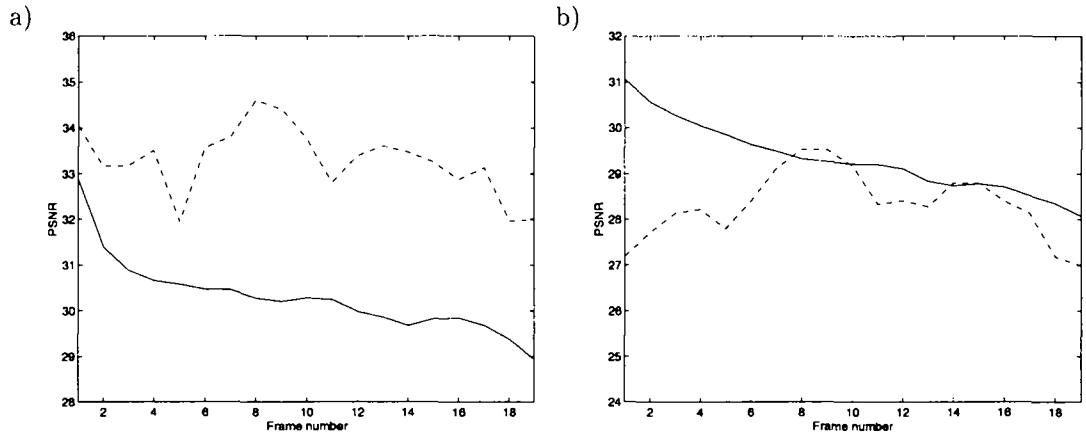

Figure 7.7: "Table Tennis": Evolution of the PSNR for respectively the object "background" (full line) and the object "hand with bat" (dashed line). **(a)** Sequence coded at 64 kbits/s and, **(b)** sequence coded at 40 kbits/s.

The object oriented functionalities of the proposed video coding scheme permit to focus the bit allocation on important objects. This strategy deeply differs from what is currently performed in standard video coding schemes. Figure 7.8 compares the performance of the proposed video coding scheme with the one of the video coding standard H.263, on the same sequence and at the same bitrates. In order to allow for an assessment of the visual quality, a set of frames coded with H.263 are displayed in Fig. 7.9. These frames correspond to the ones reported in Fig. 7.6. At a total bitrate of 64 kbits/s, the coding functionalities of the proposed video coding scheme clearly permit a higher coding quality of the foreground object "hand with bat" than the one obtained with H.263. The gain of coding quality on the foreground objects is however balanced by a deterioration of the background object. Nevertheless, the relative low quality of the background object does not spoil the decoded sequence. Indeed, the background object constitutes the object with the least visual significance. When comparing the visual quality of the decoded sequences, the degradations introduced by the proposed video coding scheme are less annoying than those introduced by H.263. In particular, no ringing effect is visible around the objects (e.g. the arm) and the contours are precisely defined (e.g. the ball). At a total bitrate of 40 kbits/s, the performance of the proposed video coding scheme, when coding the foreground object "hand with bat", are slightly better than those of H.263. Although it is still allocating most of the total bitrate to the foreground objects, the proposed

video coding scheme is no longer able to outperform H.263 in terms of PSNR. As explained when we will comment on Table 7.1, this results from the increased burden of coding the shape information when going to lower bitrates. This phenomenon is clearly visible when examining the coding results of the object "hand with bat" in intra mode. As far as the visual quality is concerned, the same conclusions as for the simulation at 64 kbits/s may be drawn. In particular, notice the sharpness of the object contours when using the proposed video coding scheme.

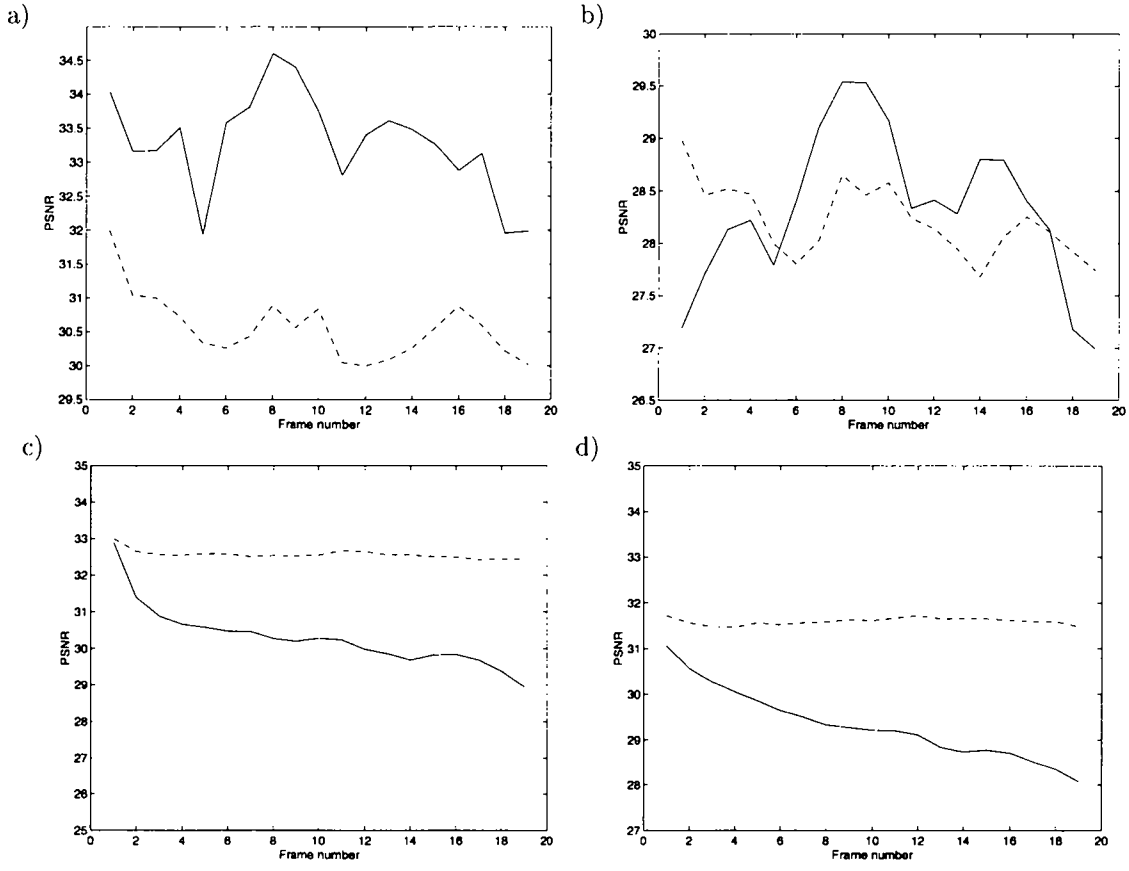

Figure 7.8: "Table Tennis": Evolution of the PSNR for respectively the proposed coding scheme (full line) and the standard H.263 (dashed line). **(a)** Sequence coded at 64 kbits/s, evolution of the PSNR on the object "hand with the bat", **(b)** sequence coded at 40 kbits/s, evolution of the PSNR on the object "hand with the bat", **(c)** sequence coded at 64 kbits/s, evolution of the PSNR on the object "background" and, **(d)** sequence coded at 40 kbits/s, evolution of the PSNR on the object "background".

In Table 7.1, the bit allocation of the proposed video coding scheme among texture information, shape information and motion information is reported. The simulation analyzed is the one on the sequence "Table Tennis" at 64 kbits/s. A distinction is made between intra mode and inter mode. As it could be expected, the intra mode represents a large share of the total bitrate, although it only serves to code the initial frame. For the intra mode, the shape information represents around 6% of the total frame cost. For the inter mode, 30% of the total cost is allocated to the shape information, 10% for the coding of the motion vectors, and 60% is attributed to the coding of the DFD. Note that the additional cost to obtain the object-oriented functionalities is roughly represented by the cost of the shape information and can be estimated at around 22% of the total cost. However, the shape being transmitted in a lossless fashion, the burden to obtain the object-oriented functionalities is bound to increase as the total bitrate decreases. For instance, in the simulation on the sequence "Table Tennis" at 40 kbits/s, the

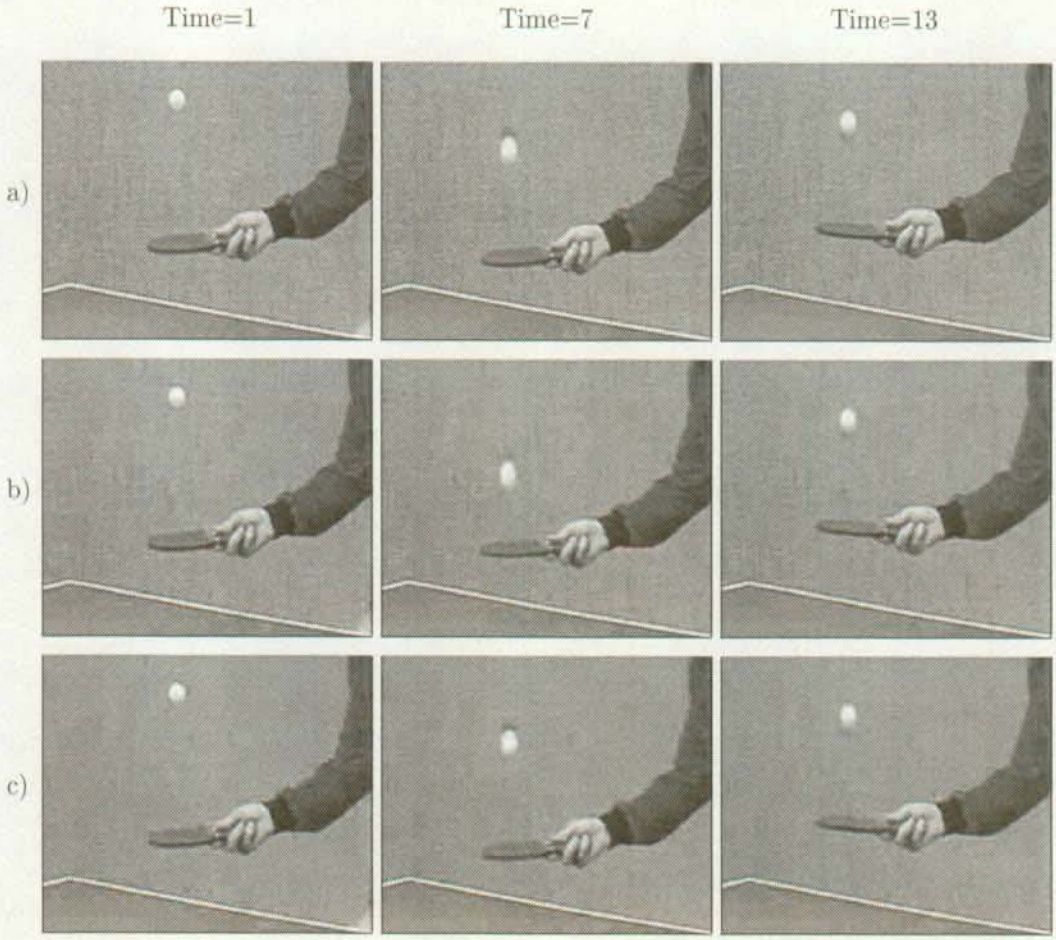

Figure 7.9: "Table Tennis": Coding results at different bitrates for the video coding standard H.263. **(a)** Original frames at time 1, 7 and 13, **(b)** sequence coded at 64 kbits/s, **(c)** sequence coded at 40 kbits/s.

cost of the shape information represents 36% of the total bitrate. At this stage, the transmission of the texture information is significantly altered due to the decreased share of the total bitrate it receives. This phenomenon is clearly explicited in Fig. 7.8. More precisely, the proposed video coding scheme outperforms, on foreground objects, the video coding standard H.263 at 64 kbits/s, while this is no longer so clear at 40 kbits/s. As it could have been expected, the functionalities of the proposed video coding scheme induce a burden which increases as the total bitrate becomes smaller.

	Intra mode	Inter mode	Total
Texture information	14700	22809	37509
Shape information	991	10219	11210
Motion information	0	3296	3296
Total bitrate	15691	36324	52015

Table 7.1: "Table Tennis": Bitrate repartition over the 19 first frames at 64 kbits/s. The frame rate is 25 Hz.

In a further experiment, the video sequence "Akiyo" is coded. The scene is formed of two objects being the background and the woman. The sequence has been compressed at respectively 25 kbits/s and 19 kbits/s. For both simulations, the evolution of their respective global PSNR is given in Fig. 7.10, while examples of decoded frames are shown in Fig. 7.11. It can be observed that the contours of the objects are clearly defined as for the simulation results obtained on the sequence "Table Tennis". This is achieved, although the bitrates are fairly low, thanks to the object-based representation of the scene. However, one can notice that the reconstructed object "woman" shows coding artifacts at the level of the face. These are due to the non-rigid motions occuring when the woman is speaking, and opening and closing her eyes. The non-rigid motions entail a poor performance of the texture prediction, while the DFD coding has not enough bitrate to compensate for the resulting errors. The latter phenomenon is illustrated in Fig. 7.12. Although the object "woman" receives the biggest share of the total bitrate for texture information, it is clearly visible that the coding quality of the woman is much lower than the one of the background. This discrepancy would have been even larger if the texture prediction had been enforced through a single set of affine motion parameters. Indeed, in the case of the object "woman", the splitting of the shape through a quadtree segmentation, which is explained in Sec. 7.3, arises frequently. In Fig. 7.13, an example is given where a single set of affine parameters is not sufficient to satisfactorily motion compensate the texture of the object "woman". Therefore, it is decomposed through the quadtree segmentation. This refinement of the segmentation allows for a significant improvement of the prediction. For instance, notice the striking amelioration when predicting the opening of the mouth.

In Table 7.2, the bit allocation of the proposed video coding scheme among the texture information, shape information and motion information is detailed. The simulation analyzed is the one on the sequence "Akiyo" at 25 kbits/s. For the intra mode, the shape information represents around 3% of the total frame cost. For the inter mode, 18% of the total cost is allocated to the shape information, 12% for the coding of the motion vectors and the quadtrees, and 70% is attributed to the coding of the DFD. Note that, the additional cost to obtain the object-oriented functionalities being roughly represented by the cost of the shape information, it can be computed for this simulation at around 11% of the total cost. Compared to the what is obtained for the sequence "Table Tennis" (see Table 7.1), the burden of transmitting the shape information appears much lower in the case of the sequence "Akiyo". This is due to the fact the latter sequence only contains two objects. It results in less shape information to be coded when compared to the sequence "Table Tennis" which is formed of five objects. Note that the share of the total bitrate required for the motion information is roughly the same for both sequences.

Typical examples of sprites are shown in Fig. 7.14. For each object, the initial sprite (i.e. at time 1)

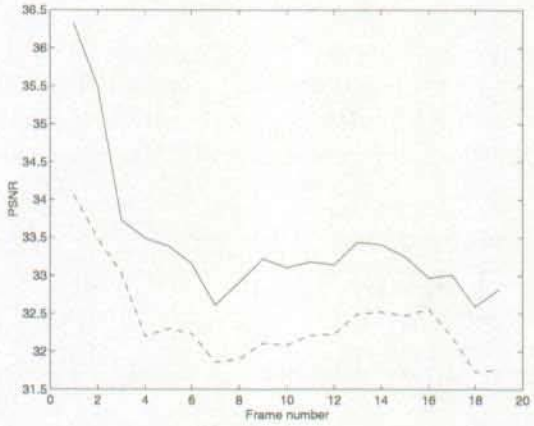

Figure 7.10: "Akiyo": Evolution of the global PSNR at respectively 25 kbits/s (full line) and 19 kbits/s (dashed line).

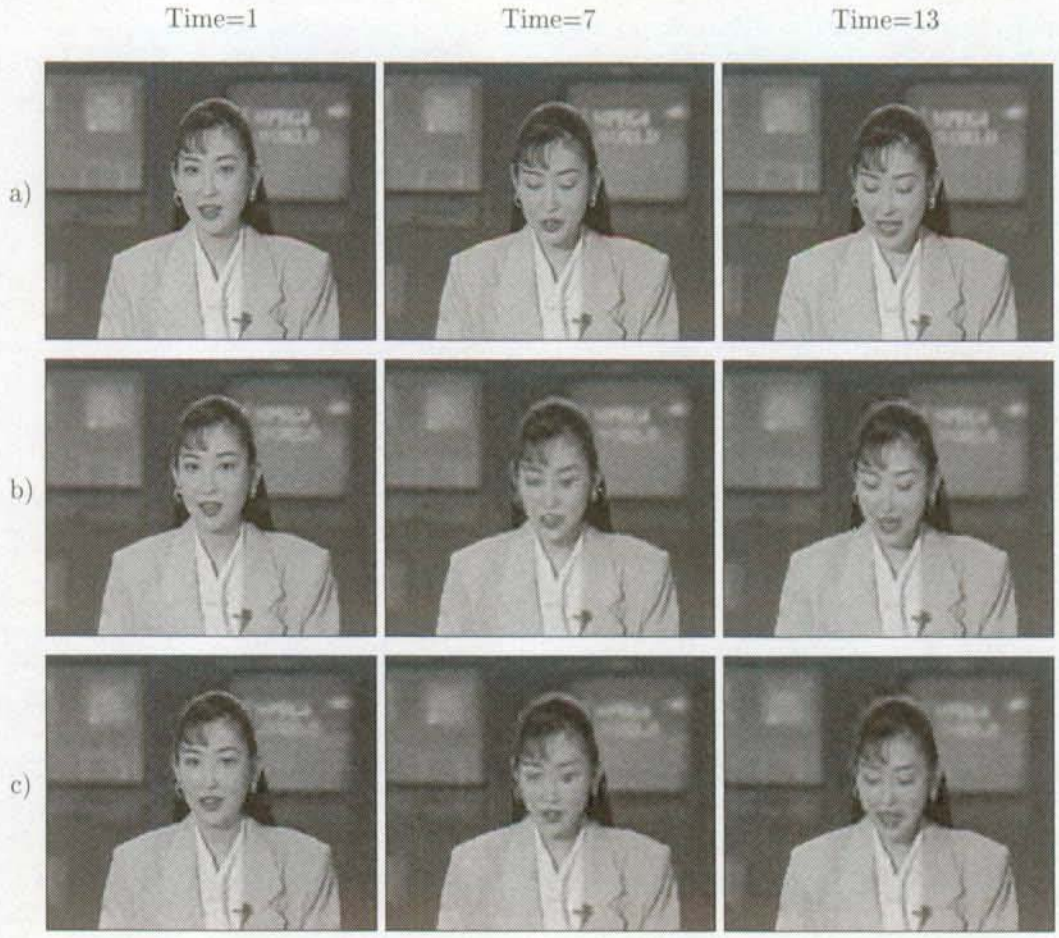

Figure 7.11: "Akiyo": Coding results at different bitrates for the proposed video coding scheme. **(a)** Original frames at time 1, 7 and 13, **(b)** sequence coded at 25 kbits/s, **(c)** sequence coded at 19 kbits/s.

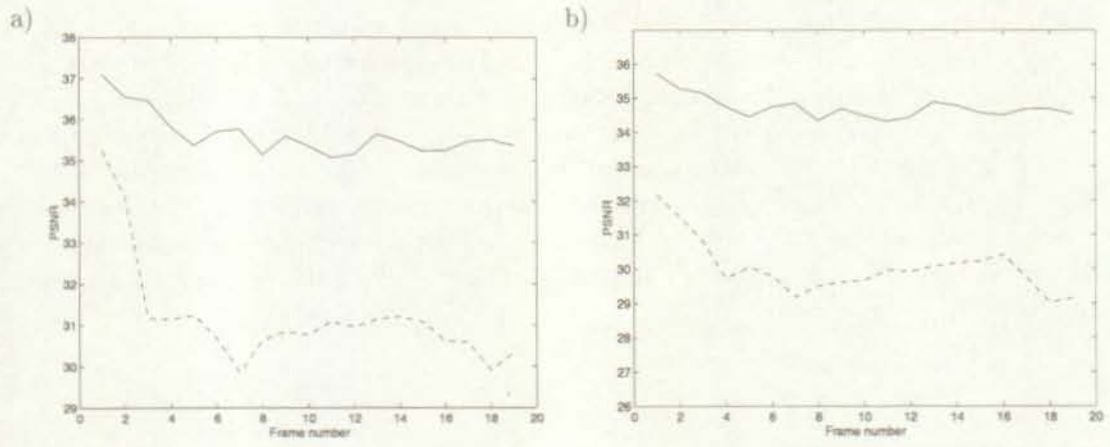

Figure 7.12: "Akiyo": Evolution of the PSNR for respectively the object "background" (full line) and the object "woman" (dashed line). **(a)** Sequence is coded at 25 kbits/s and, **(a)** sequence is coded at 19 kbits/s.

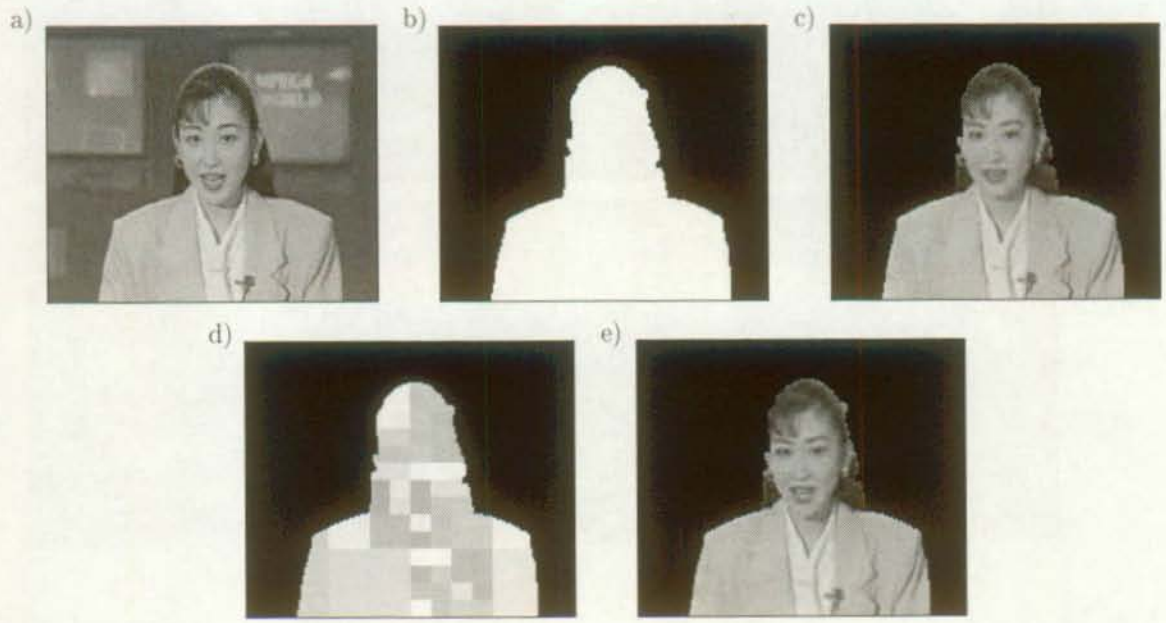

Figure 7.13: "Akiyo": Example of quadtree based segmentation of an object. **(a)** Frame to be coded, **(b)** segmentation of the object "woman", **(c)** temporal prediction of the current frame using a single affine parameter set for object "woman", **(d)** quadtree based splitting of the object "woman" and, **(e)** temporal prediction of the current frame using the quadtree approximation of the object "woman".

	Intra mode	Inter mode	Total
Texture information	20275	18248	38523
Shape information	494	4787	5281
Motion information	0	3278	3278
Total bitrate	20769	26313	47082

Table 7.2: "Akiyo": Bitrate repartition over the 19 first frames at 25 kbits/s. The frame rate is 10 Hz.

is generated using only the information of the particular object. In order to fill the areas left vacant through the extraction of the other objects, the texture of the object of interest is extended using a morphological padding algorithm. As time passes by, the sprite integrates the information about the object. For example, the initial background sprite only takes the information of the background object. The ball, the arm, the left hand and the right hand with the bat are replaced by an extension of the background. Through time, the sprite is updated by integrating the information about the background in the successive frames. In doing so, the sprite permits to overcome the problems arising from disocclusion phenomena. As far as the object motion is concerned, the sprite permits a motion estimation which is independent from the other objects of the scene. Thus, it allows the definition of an independent bitstream for each object.

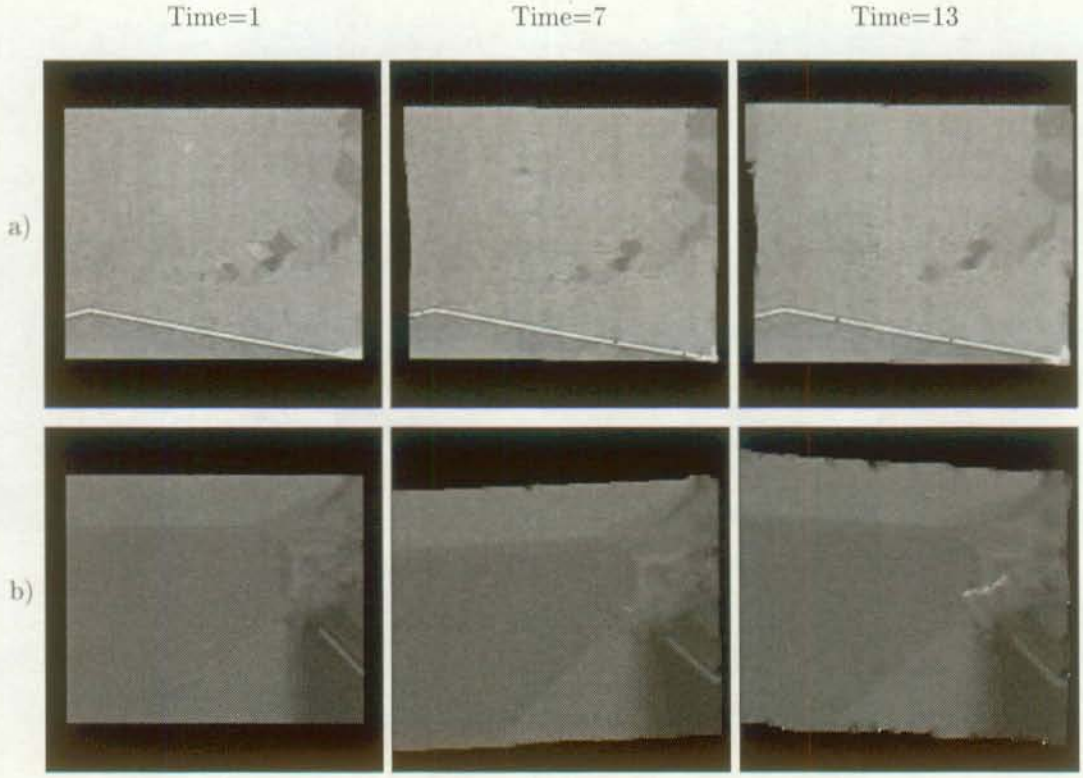

Figure 7.14: "Table Tennis": Examples of sprite generation. **(a)** Sprite of the object "background" at time 1, 7 and 13, **(b)** sprite of the object "arm" at time 1, 7 and 13.

The shape coding is performed using the method described in Section 7.5. As already explained, the shape information is sent in a lossless fashion. As it could have been expected, the intra mode coding is more expensive in terms of bitrate than the inter mode coding. In the latter and compared to the intra mode, the proposed predictive shape coding technique achieves a bitrate reduction of 30% and 50% for the video sequences "Table Tennis" and "Akiyo", respectively. This arises from the strong temporal redundancy existing between the successive masks of the objects. In Fig. 7.15, the shape of the object "arm" in the current frame and its prediction from the previous frame are portrayed. Using temporal information, the current mask can be predicted quite accurately from the previous one.

As far as the end-user is concerned, the functionalities of the proposed video coding scheme allow her/him to have an extended control over the decoded sequence. The control is both in terms of the scene content and coding quality of the different objects. Such interactions are illustrated in Fig. 7.16. The foreground

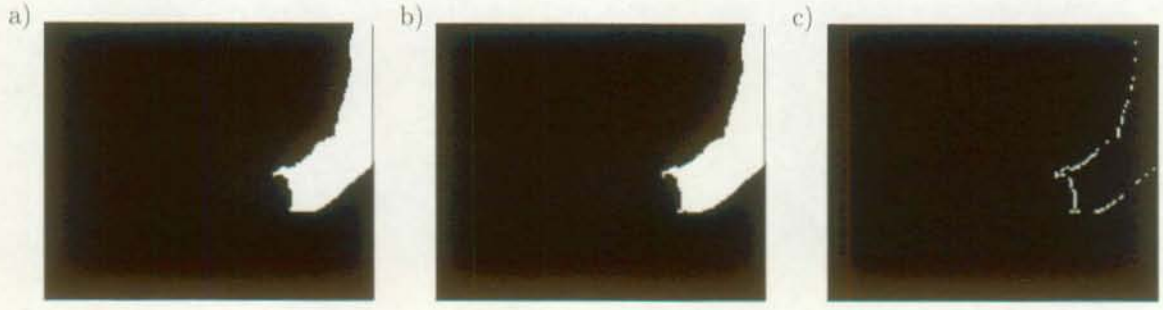

Figure 7.15: "Table Tennis": Example of predictive shape coding for the object arm. **(a)** Shape of the object "arm" in the previous frame, **(b)** shape of the object "arm" in the current frame, **(c)** the difference between current shape and motion predicted shape shows the quality of the prediction.

objects of the sequence "Table Tennis" are coded and sent to the end-user, while the object "background" is not transmitted. The bit allocation among the foreground object depends on the visual importance and coding difficulty of the different objects. For instance, a big share of the available bitrate for texture information is assigned to the object "hand with the bat", while the objects "ball" and "arm" are coded more coarsely. At the decoder side, the end-user reconstructs the video sequence on the basis of the information he receives about each object. The decoding of each object being independent, he may freely choose, for instance, to put a substitute background and change the texture of the object "arm". In Fig. 7.16, the latter texture is taken from the video sequence "Mobile & Calendar", while the background comes from the video sequence "Bream".

7.8 Conclusions

In this chapter a video coding scheme is proposed. Based on a decomposition of the scene into its objects, it allows for full interactivity with the visual information. In particular, each object of the scene is represented independently from the other objects. This allows a content-based object scalability at the decoder side. Furthermore, a different coding quality can be attribute to each object resulting in a content-based coding scalability. Also, the texture information for each object is encoded progressively allowing an object-based spatial scalability.

In the simulation results, the importance of allocating a different coding quality to each object is demonstrated. Indeed, by focusing on foreground objects, one can significantly improve the visual quality of the decoded video sequence. Furthermore, the proposed scheme does not introduce any visually annoying artifacts. However, it must be underlined that, as the total bitrate decreases, the inherent functionalities of the proposed video coding scheme represent a heavier burden. Finally, the ability for the end-user to interact with the content of the decoded sequence is clearly demonstrated.

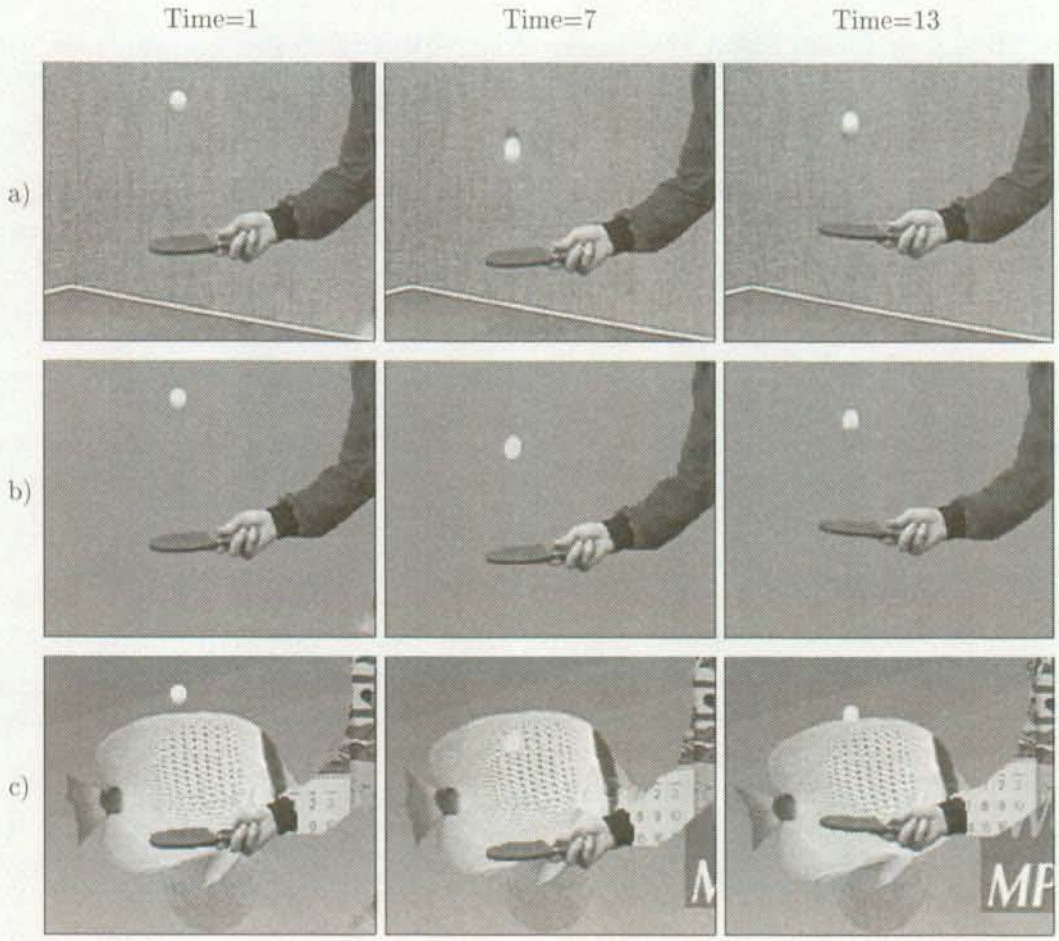

Figure 7.16: "Table Tennis": Example of content based object scalability. **(a)** Original frames at time 1, 7 and 13, **(b)** sequence coded at 64 kbits/s, the background is not transmitted, **(c)** decoded sequence where the end-user has interactively chosen a substitute background and a new texture for the arm.

Chapter 8

Conclusions

8.1 Summary of achievements

This chapter concludes the dissertation by summarizing the main developments and achievements of this work.

In this dissertation, an alternative to the waveform representation of the visual information has been proposed. The waveform representation directly derives from the scene capture process and is not representative of the semantic information existing in the scene. The proposed representation aims at being both semantically meaningful and malleable. Such characteristics are obtained by describing the scene through its constituting objects. We define the objects to be entities that are temporally and spatially coherent through time. The resulting representation allows for an extended interaction with the visual information.

The objects forming the scene are primarily characterized through their temporal characteristics. Chapter 2 presents the state of the art for estimating the motion information when the scene segmentation is given. The simultaneous definition of the motion and the segmentation is addressed in Chapter 3. The concepts of spatio-temporal segmentation and object tracking are described.

In Chapter 4, a top-down technique for defining spatio-temporally homogeneous regions is presented. The technique combines spatial, temporal and change information. The regions are gradually refined until they are spatially and temporally homogeneous. The effects of disocclusion phenomena and of the camera motion are also taken into account. Simulation results show the efficiency of the proposed method. The effectiveness of combining spatial, temporal and change information is clearly illustrated. The resulting regions define a precise partition of the frame into visually meaningful areas. Finally, the necessity of addressing the phenomenon of disocclusion is demonstrated.

Chapter 5 presents an unsupervised bottom-up technique for spatio-temporal segmentation. It assumes that a set of initial regions is available. These regions are iteratively merged in order to define the objects constituting the scene. The tendency of the regions to be merged together is assessed through their mutual spatio-temporal similarity. The similarity measure is defined in the framework of hypothesis testing. The measure of similarity incorporates temporal and spatial information into a single measure. In particular, the temporal information derives from the temporal information existing both in the motion parametric representation and the residual distributions. The MKS test statistic is proposed to assess the hypothesis of temporal similarity. The merging of the regions relies on a graph representation. Two mutually complementary graph clustering rules are presented: the strong rule and the weak rule. These rules are applied successively and permit a robust derivation of the objects in the scene. The whole merging procedure is embedded in a dynamic graph update strategy. In the simulations, two types of initial regions are used. The first one derives from a quadtree segmentation, while the second one is generated through the top-down technique proposed in Chapter 4. For both types of regions, the resulting spatio-temporal segmentations encompass the objects forming the scene. However, the definition of the objects depends on the type of initial regions. A very precise definition is obtained when using the initial regions that are generated by the technique proposed in Chapter 4. Also, the genuineness of the proposed spatio-temporal similarity and the efficiency of the MKS test statistic are clearly demonstrated.

In Chapter 6, an unsupervised algorithm for recursive spatio-temporal segmentation and an unsupervised algorithm for object tracking are presented. Although distinct, both algorithms deeply interact. Starting from the segmentation of the previous frame, the recursive spatio-temporal segmentation aims at robustly partitioning the current frame. To that end, past and current information is combined. This is achieved in three successive steps: partition projection, region validation and constrained spatio-temporal segmentation. The resulting segmentation of the current frame is coherent with the segmentation of the

previous frame. However, this requirement of stability accommodates for possible changes occuring in the current frame. The object tracking algorithm addresses the correspondence problem. It tries to identify the objects in the current segmentation and to match them with the objects detected in previous frames. To that end, each object is characterized through spatial, temporal and spatio-temporal features. The identification is achieved by using these features in the context of a hierarchy of correspondence hypotheses. Such features also permit the tracking algorithm to pursue objects that may be lost momentarily. Simulation results demonstrate the capacity of the proposed algorithms to segment and track simultaneously objects with very different temporal and spatial characteristics. The appearance of new objects and the disappearance of objects are robustly dealt with. Moreover, the recursive integration of the information over successive frames leads to an improved segmentation. Finally, the effectiveness of the object memory is shown.

An application of the representation of the visual information in terms of objects is given in Chapter 7. More precisely, a second generation video coding scheme is presented which permits a scalable coding both in terms of the scene content and the objects quality. To that end, each object forming the scene is encoded independently from the other objects. Predictive texture coding and predictive shape coding techniques are described which allow to exploit the temporal redundancy. Also, a technique for generating dynamically the sprite of each object is proposed. Simulation results demonstrate the capacity of the proposed video coding scheme of combining efficiency with functionalities. In particular, the proposed video coding scheme allows to focus the biggest share of the total bitrate on objects which are visually important. The experiments also demonstrate that the functionalities of the proposed scheme constitute a higher burden in term of efficiency as the total bitrate decreases. Finally, examples of possible interactions with the decoded sequence are given.

8.2 Possible extensions

The work presented in this dissertation may be extended in many directions. Some directions for further work are proposed below.

- The generation of spatio-temporally homogeneous regions should rely on the information present in multiple frames. A recursive procedure can be foreseen.
- The motion may be characterized through a more complex model than the affine motion model. In a further step, the hypothesis of rigid body motion should be removed.
- The spatio-temporal similarity may be extended in order to incorporate semantic information. To that end, a data-base of the objects commonly found in the nature may be built. The similarity measure would be updated whenever a region in the frame is likely to possess the characteristics of a part of an object present in the data-base.
- The region merging algorithm for spatio-temporal segmentation should be extended in order to allow for the splitting of troublesome regions. This may be achieved by combining the region merging algorithm with the top-down technique used to generate spatio-temporally homogeneous regions.
- The recursive spatio-temporal segmentation should take into account the spatial information arising from the objects found in the previous frame. More precisely, the recursive spatio-temporal similarity could be modified in order to incorporate this information. For instance, this may be achieved through the use of additive invariants as proposed by Eggers *et al.* [50].

- A Kalman filtering algorithm could be used to predict the temporal behavior of the objects. This information would be useful for the partition projection procedure and for the verification of the different correspondence hypotheses [156].
- The information arising from the object tracking procedure should be better exploited. For instance, the time interval during which each object has been tracked may give important insights about the semantic meaning of the object. Therefore, the suspicion of erroneous segmentation may be confirmed or discarded by analyzing the past behavior of the objects which are lost in the current frame.
- A lossy scheme for coding the shape information could be very helpful. Indeed, it would allow to decrease the cost of the shape information in case of very low bitrate coding.
- The efficiency of the proposed video coding scheme could be enhanced by more thorough rate control. All the existing control strategies could be readily implemented. In particular, a rate control based on the quality of the decoded objects could be foreseen.

Appendix A

Description of the video sequences

The video sequences used in this dissertation come from the MPEG-4 library of test sequences and are in QCIF format. The selected video sequences are commonly referred to as "Akiyo", "Table Tennis", "Foreman" and "Bream". No time subsampling is performed with the exception of the sequence "Akiyo". More precisely, the sequences 'Akiyo", "Table Tennis", "Foreman" and "Bream" have a frame rate of 10 Hz, 25 Hz, 25 Hz and 30 Hz, respectively.

Sequence "Akiyo"

In this sequence, a woman seats in front of a static background. Only the upper part of the woman is visible. Two typical frames of the sequence are presented in Fig. A.1. Visually, the scene may be decomposed into two objects: the woman and the background.

The difficulty of the scene lies in the relative small motion. In order to enhance the temporal changes, the original sequence is temporally subsampled. The resulting sequence is temporally sampled at 10 Hz. The motion remains however small. The distinction between the two objects still constitutes a challenging problem. Another difficulty of the scene lies in the hair of the woman. It is spatially almost indistinguishable from the background.

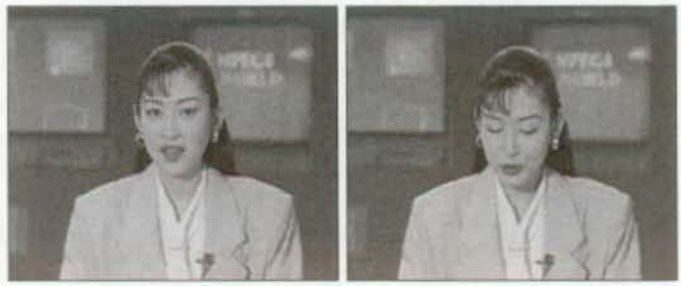

Figure A.1: Sequence "Akiyo": Two typical frames.

Sequence "Table Tennis"

In this sequence, a man is playing table tennis. Only the hands and one arm of the man are visible. Two typical frames of the sequence are presented in Fig. A.2. Visually, the scene may be decomposed into six main objects: the two hands, the arm, the table, the ball and the background.

The main difficulty of the scene lies in the relative high number of objects composing it. Furthermore, the characteristics of the different objects render the analysis of the scene challenging. The ball is small, has a fast motion and, performs brisk maneuver when touching the bat. The left hand is badly defined spatially and disappears from scene. Finally, the background is very noisy.

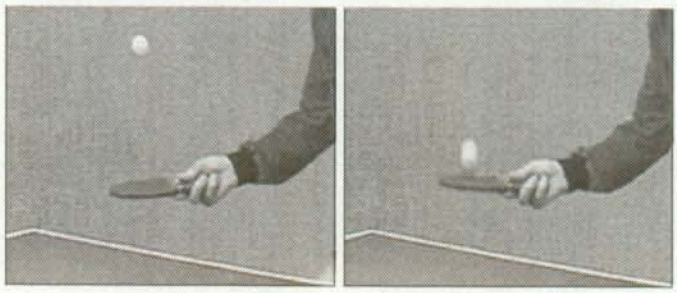

Figure A.2: Sequence "Table Tennis": Two typical frames.

Sequence "Foreman"

In this sequence, a man is speaking in front of the camera. Only the upper part of his body and his head are visible. Two typical frames of the sequence are presented in Fig. A.3. Visually, the scene may be decomposed into two main objects: the man's head with its helmet and the background.

The main difficulty of the scene lies in the closeness of the scene to the camera. Strong depth changes are present. Furthermore, the man wears an helmet whose luminance is almost uniform. Consequently, the related motion is badly defined. Problems may also occur due to the non-rigid motion taking place at the level of the man's mouth.

Figure A.3: Sequence "Foreman": Two typical frames.

Sequence "Bream"

In this sequence, a fish is swimming in front of an artificially generated background. At a certain point, a board progressively appears at the bottom right of the scene. Two typical frames of the sequence are presented in Fig. A.4. Visually, the scene may be decomposed into three main objects: the fish, the board and the background.

The difficulty of the scene is due to the background. It is quite noisy and encompasses regions which are transparent. The problem of transparency is also present in the board. This last object appears during the sequence and partially covers the fish. Furthermore, the fish is composed of regions which are spatially very dissimilar in luminance.

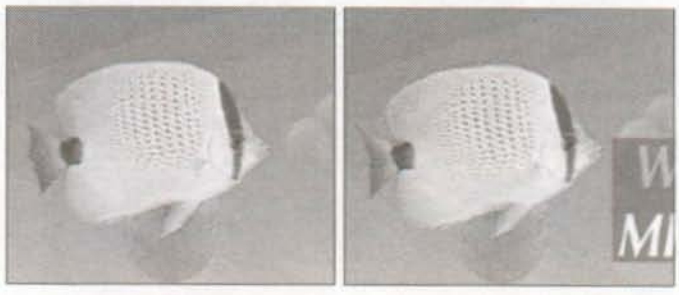

Figure A.4: Sequence "Bream": Two typical frames.

Appendix B

Planar surface under perspective projection

This appendix computes the coefficients $a_1, ..., a_8$ in Eq. (2.12) and Eq. (2.13), respectively.

For a planar surface under perspective projection, the instantaneous velocity in the image plane is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \\ a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \end{pmatrix} , \tag{B.1}$$

where

$$\begin{aligned} a_1 &= \omega_y + T_x k_z , && \text{(B.2)} \\ a_2 &= T_x k_x - T_z k_z , && \text{(B.3)} \\ a_3 &= -\omega_z + T_x k_y , && \text{(B.4)} \\ a_4 &= -\omega_x + T_y k_z , && \text{(B.5)} \\ a_5 &= \omega_z + T_y k_x , && \text{(B.6)} \\ a_6 &= T_y k_y - T_z k_z , && \text{(B.7)} \\ a_7 &= \omega_y - T_z k_x , && \text{(B.8)} \\ a_8 &= -\omega_x - T_z k_y . && \text{(B.9)} \end{aligned}$$

Similarly, for a planar surface under perspective projection, the displacement in the image plane is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{a_1 + a_2 x + a_3 y}{a_7 x + a_8 y + a_9} \\ \frac{a_4 + a_5 x + a_6 y}{a_7 x + a_8 y + a_9} \end{pmatrix} , \tag{B.10}$$

where

$$a_1 = \Omega_y + D_x k_z \ , \tag{B.11}$$

$$a_2 = 1 + D_x k_x \ , \tag{B.12}$$

$$a_3 = -\Omega_z + D_x k_y \ , \tag{B.13}$$

$$a_4 = -\Omega_x - D_y k_z \ , \tag{B.14}$$

$$a_5 = \Omega_z + D_y k_x \ , \tag{B.15}$$

$$a_6 = 1 + D_y k_y \ , \tag{B.16}$$

$$a_7 = -\Omega_y + D_z k_x \ , \tag{B.17}$$

$$a_8 = \Omega_x + D_z k_y \ , \tag{B.18}$$

$$a_9 = 1 + D_z k_z \ . \tag{B.19}$$

Appendix C

Affine moment invariants of a $2D$ object

This appendix describes affine moment invariants which permit the characterization of a $2D$ object which is undergoing an affine transformation. For more details, see the Taubin and Cooper [138]. The affine transformation is assumed to be nonsingular.

Let us consider the ensemble of q pixels belonging to a given object. This ensemble is denoted $\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_q\}$ where $\vec{x}_i$ refers to the pixel with the coordinates x_i and y_i, i.e. $\vec{x}_i = (x_j, y_i)^T$. The total mass $|\mu|$ of the object is defined as

$$|\mu| = \sum_{i=1}^{q} \mu(\vec{x}_i) \ . \tag{C.1}$$

where $\mu(\vec{x})$ is the density function. In the case where the luminance information is taken into account, $\mu(\vec{x})$ is simply the luminance at the pixel $\vec{x}$ if the latter belongs to the object and zero otherwise. When only the shape information is considered, $\mu(\vec{x})$ has a value of one if the pixel $\vec{x}$ belongs to the object and a zero value otherwise.

The total mass $|\mu|$ permits to compute the means $\overline{x}$ and $\overline{y}$, respectively

$$\overline{x} = \frac{1}{|\mu|} \sum_{i=1}^{q} x_i \mu(\vec{x}_i) \ , \tag{C.2}$$

$$\overline{y} = \frac{1}{|\mu|} \sum_{i=1}^{q} y_i \mu(\vec{x}_i) \ . \tag{C.3}$$

Based on the above definitions, the centered moment $M_{(k,l)}$, $k, l \in \mathbb{N}$, of degree $d = k + l$ is written as

$$M_{(k,l)} = \frac{1}{|\mu|} \sum_{i=1}^{q} (x_i - \overline{x})^k (y_i - \overline{y})^l \mu(\vec{x}_i) \ . \tag{C.4}$$

The construction of affine invariant quantities relies on matrices composed by centered moments. In particular, we are looking for matrices which are covariant on one side and contravariant on the other side of the affine transformation. Their eigenvalues are indeed absolute invariants of the affine transformation.

In this dissertation, the set of invariants $\vec{V}$ is composed of five such eigenvalues, $\vec{V} = \{V_1, V_2, ..., V_5\}$. They are the eigenvalues of the following matrices

$$\begin{array}{l} M'_{[1,2]}M'_{[2,1]}\ , \\ M'_{[1,2]}M'_{[2,2]}M'_{[2,1]}\ , \\ M'_{[0,2]}M'_{[2,2]}M'_{[2,0]}\ , \end{array} \tag{C.5}$$

with

$$M'_{[1,2]} = M'^{T}_{[2,1]} = \begin{pmatrix} \frac{1}{\sqrt{2}}M'_{(3,0)} & M'_{(2,1)} & \frac{1}{\sqrt{2}}M'_{(1,2)} \\ \frac{1}{\sqrt{2}}M'_{(2,1)} & M'_{(1,2)} & \frac{1}{\sqrt{2}}M'_{(0,3)} \end{pmatrix}\ , \tag{C.6}$$

$$M'_{[2,2]} = \begin{pmatrix} \frac{1}{2}M'_{(4,0)} & \frac{1}{\sqrt{2}}M'_{(3,1)} & \frac{1}{2}M'_{(2,2)} \\ \frac{1}{\sqrt{2}}M'_{(3,1)} & M'_{(2,2)} & \frac{1}{\sqrt{2}}M'_{(3,1)} \\ \frac{1}{2}M'_{(2,2)} & \frac{1}{\sqrt{2}}M'_{(1,3)} & \frac{1}{2}M'_{(0,4)} \end{pmatrix}\ , \tag{C.7}$$

and

$$M'_{[0,2]} = M'_{[2,0]} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}\ . \tag{C.8}$$

The notation $M'_{(k,l)}$ signifies that the centered moments are not computed with respect to the original coordinate system. They are specified in the canonical frame of reference defined by the transformation $\vec{x}' = L\,\vec{x}$, with L being the 2x2 lower triangular matrix corresponding to the Cholesky decomposition of $M_{[1,1]}$. That is

$$LL^T = M_{[1,1]}\ , \tag{C.9}$$

with (C.10)

$$M_{[1,1]} = \begin{pmatrix} M_{(2,0)} & M_{(1,1)} \\ M_{(1,1)} & M_{(0,2)} \end{pmatrix}\ . \tag{C.11}$$

The matrix L is expressed as

$$L = \begin{pmatrix} \sqrt{M_{(2,0)}} & 0 \\ \frac{M_{(1,1)}}{\sqrt{M_{(2,0)}}} & \sqrt{\left(M_{(0,2)} - \left(\frac{M_{(1,1)}}{\sqrt{M_{(2,0)}}}\right)^2\right)} \end{pmatrix}\ . \tag{C.12}$$

Bibliography

[1] "Special Issue on Visual Communications", *AT&T Technology*, Vol. 7, No. 3, Fall 1992.

[2] ISO/IEC DIS 13818-2, "Information Technology -- Generic Coding of Moving Pictures and Associated Audio Information -- Part 2: Video", Technical report, International Organization for Standardization, 1994.

[3] T. Aach, Kaup A., and R. Mester, "Statistical model-based change detection in moving video", *Signal Processing*, Vol. vol. 31, no. 2, pp. 165-180, March 1993.

[4] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects", *IEEE Trans. Pattern Anal. and Machine Intell.*, Vol. vol. PAMI-7, no. 4, pp. 384-401, July 1985.

[5] G. Adiv, "Inherent ambiguities in recovering 3d motion and structure from a noisy flow field", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-11, no. 5, pp. 477-489, May 1989.

[6] D. Adolph and R. Buschmann, "1.15 Mbit/s coding of video signals including global motion compensation", *Signal Processing: Image Communication*, Vol. vol. 3, nos. 2-3, pp. 259-274, June 1991.

[7] J.K. Aggarwal, L.S. Davis, and W.N. Martin, "Correspondence processes in dynamic scene analysis", *Proc. IEEE*, Vol. 69, No. 5, pp. 562–572, May 1981.

[8] J.K. Aggarwal and W.N. Martin, "Analyzing dynamic scenes containing multiple moving objects", in T.S. Huang, editor, *Image Sequence Analysis*, pp. 355–380. Springer-Verlag, New-York, 1981.

[9] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images-a review", *Proc. IEEE*, Vol. vol. 76, no. 8, pp. 917-935, August 1988.

[10] M. Allmen and C.R. Dyer, "Computing spatiotemporal relations for dynamic perceptual organisation", *CVGIP: Image understanding*, Vol. vol. 58, no. 3, pp. 338-351, November 1993.

[11] P. Anandan, J.R. Bergen, K.J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation", in M.I. Sezan and R.L. Lagendijk, editors, *Motion Analysis and Image Sequence Processing*, pp. 1–22. Kluwer Academic Publishers, 1993.

[12] S. Ayer, *Sequential and Competitive Methods for Estimation of Multiple Motions*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1995.

[13] S. Ayer and H.S. Sawhney, "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding", *in Fifth International Conference on Computer Vision*, pp. 777–784, Cambridge, MA, June 1995.

[14] S. Ayer and P. Schroeter, "Hierarchical robust motion estimation for segmentation of moving objects", *in IEEE Workshop on Image and Multidimensional Signal Processing*, pp. 122–123, Cannes, France, September 1993.

[15] S. Ayer, P. Schroeter, and J. Bigün, "Tracking based on hierachical multiple motion estimation and robust regression", *in Time Varying Image Processing and Moving Objects Recognitions*, Florence, Italy, Juin 1993.

[16] S. Ayer, P. Schroeter, and J. Bigün, "Segmentation of moving objects by robust motion parameter estimation over multiple frames", *in ECCV'94*, Vol. 2, pp. 316–327, Stockholm, Sweden, May 1994.

[17] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, Inc., 1988.

[18] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of optical flow techniques", *Int. Journal of Computer Vision*, Vol. vol. 12, no. 1, pp. 43-77, 1994.

[19] J.R. Bergen, J.B. Burt, J. Hingorani, and S. Peleg, "A three-frame algorithm for estimating two-component image motion", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-14, pp. 886-896, September 1992.

[20] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg, "Computing two motions from three frames", *in ICCV'90*, pp. 27–32, 1990.

[21] M. Bertero, T.A. Poggio, and V. Torre, "Ill-posed problems in early vision", *Proc. IEEE*, Vol. vol. 76, pp. 869-887, 1988.

[22] M.J. Black, "Combining intensity and motion for incremental segmentation and tracking over long image sequences", in Sandini G., editor, *Second European Conference on Computer Vision*, pp. 485–493, Santa Margherita, Italy, May 1992.

[23] M.J. Black, "Recursive non-linear estimation of discontinous flow fields", *in ECCV'94*, Vol. 1, pp. 138–145, Stockholm, Sweden, May 1994.

[24] M.J. Black and P. Anandan, "A framework for the robust estimation of optical flow", *in ICCV'94*, pp. 231–236, Berlin, Germany, May 1993.

[25] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, 1987.

[26] P. Bouthemy and E. Francois, "Motion segmentation and qualitative dynamic scene analysis from an image sequence", *Int. Journal of Computer Vision*, Vol. vol. 10, no. 2, pp. 157-182, 1993.

[27] P. Bouthemy and P. Lalande, "Detection and tracking of moving objects based on a statistical regularization method in space and time", *in ECCV'90*, pp. 307–311, Antibes, France, April 1990.

[28] P. Bouthemy and J. Santillana Rivero, "A hierarchical likelihood approach for region segmentation according to motion-based criteria", *in ICCV'87*, pp. 463–467, London, UK, 1987.

[29] N. Brady and N. O'Connor, "Object detection and tracking using an em-based motion estimation and segmentation framework", *in Proc. ICIP'96*, Vol. I, pp. 925–928, Lausanne, Switzerland, September 1996.

[30] C.R. Brice and C.L. Fennema, "Scene analysis using regions", *Artificial Intelligence*, Vol. vol 1, pp. 205-226, 1970.

[31] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code", *IEEE Trans. Commun.*, Vol. vol. COM-31, no. 4, pp. 482-540, April 1983.

[32] P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczynski, W.A. Lee, A. Leung, J. Lubin, and H. Shvaytser, "Object tracking with a moving camera, an application of dynamic motion analysis", *in IEEE Proc. Workshop on Visual Motion*, pp. 2–12, Irvine,CA, March 1989.

[33] P.J. Burt, R. Hingorani, and R.J. Kolczynski, "Mechanisms for isolating component patterns in the sequential analysis of multiple motion", *in IEEE Workshop on Visual Motion*, pp. 187–193, Princeton, NJ, October 7-9 1991.

[34] E. Chalom and V.M. Jr. Bove, "Segmentation of frames in a video sequence using motion and other attributes", *in SPIE Proc. Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 230–241, 1995.

[35] E. Chalom and V.M. Jr. Bove, "Segmentation of an image sequence using multi-dimensional image attributes", *in Proc. ICIP'96*, Lausanne, Switzerland, September 1996.

[36] S.K. Chang, "Image Information Systems", *Proceedings of the IEEE*, Vol. 73, No. 4, pp. 754–764, April 1985.

[37] I.J. Cox and S.L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-18, no. 2, pp. 138-150, February 1996.

[38] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences", *Signal Processing: Image Communication*, Vol. vol. 3, no. 1, pp. 23-56, February 1991.

[39] J. Droulez, A. Grantyn, and P.-P. Vidal, "Les bases neuronales du contrôle oculomoteur", *in Le courrier du CNRS*, May 1992.

[40] B. Duc, P. Schroeter, and J. Bigün, "Spatio-temporal robust motion estimation and segmentation", *in 6th International Conference on Computer Analysis of Images and Patterns*, pp. 238–245, Prague, September 6-8 1995.

[41] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution", *Proc. IEEE*, Vol. vol. 83, no. 6, pp. 858-876, June 1995.

[42] F. Dufaux and F. Moscheni, "Segmentation-based motion estimation for second generation video coding techniques", in L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*, pp. 219-263. Kluwer Academic Publishers, 1995.

[43] F. Dufaux and F. Moscheni, "Background mosaicking for low bit rate video coding", *in Proc. ICIP'96*, Vol. I, pp. 673–676, Lausanne, Switzerland, September 1996.

[44] F. Dufaux, F. Moscheni, and A. Lippman, "Spatio-temporal segmentation based on motion and static segmentation", *in Proc. ICIP'95*, Vol. 1, pp. 306–309, Washington, DC, October 1995.

[45] T. Ebrahimi, "A new technique for motion field segmentation and coding for very low bitrate video coding applications", *in Proc. ICIP'94*, Vol. II, pp. 433–437, Austin, TX, November 1994.

[46] T. Ebrahimi, E. Reusens, and W. Li, "New trends in very low bitrate coding", *Proc. IEEE*, Vol. vol. 83, no. 6, pp. 877-891, June 1995.

[47] O. Egger, *Region Representation Using Nonlinear Techniques with Applications to Image and Video Coding*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1997.

[48] O. Egger, T. Ebrahimi, and M. Kunt, "Arbitrarily-Shaped Wavelet Packets for Zerotree Coding", *Proc. ICASSP'96*, Vol. IV, pp. 2335–2338, May 1996.

[49] O. Egger and W. Li., "Subband Coding of Images Using Asymmetrical Filter Banks", to be published in *IEEE Transactions on Image Processing*, April 1995.

[50] H. Eggers, F. Moscheni, and R. Castagno, "Robust object tracking based on spatial characterization of objects by additive invariants", *in Proc. ICASSP'97*, Munich, Germany. *To be published*, April 1997.

[51] M.C.J. Elton, "Visual Communication Systems: Trials and Experiences", *Proceedings of the IEEE*, Vol. 73, No. 4, pp. 700–705, April 1985.

[52] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences", *Computer Graphics and Image Processing*, Vol. vol. 43, pp. 150-177, August 1988.

[53] F.Bossen and T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation", *in ICASSP, to be published*, 1997.

[54] D.J. Fleet and A.D. Jepson, "Velocity extraction without form interpretation", *in Proc. 3rd Workshop on Computer Vision: Representation and Control*, pp. 179–185, October 1985.

[55] G.L. Foresti, P. Matteuci, C.S. Regazzoni, and S. Spaggiari, "A visual surveillance system for autonomous vehicle risk avoidance", *in Proc. EUSIPCO 94*, pp. 1369–1373, Edinburgh, U.K., September 1994.

[56] C.S. Fuh and P. Maragos, "Affine models for image matching and motion detection", *in Proc. ICASSP'91*, Vol. IV, pp. 2409–2412, Toronto, Canada, May 1991.

[57] N. Garcia, F. Jaurgarguizar, and J.I. Ronda, "Pixel-based video compression schemes", in L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*, pp. 31–78. Kluwer Academic Publishers, 1995.

[58] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images,", *IEEE Trans. Pattern Anal. and Machine Intell.*, Vol. vol. PAMI-6, no. 6, pp. 721-741, November 1984.

[59] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates", *IEEE Trans. Circuits and Systems for Video Technology*, Vol. vol.4, no. 3, pp.228-235, June 1994.

[60] B. Girod, "Motion compensation: visual aspects, accuracy and fundamental limits", in M.I. Sezan and R.L. Lagendijk, editors, *Motion Analysis and Image Sequence Processing*, pp. 125–152. Kluwer Academic Publishers, 1993.

[61] F. Glazer, "Multilevel relaxation in low-level computer vision", in A. Rosenfeld, editor, *Multiresolution Image Processing Analysis*, pp. 312–330. Springer-Verlag, 1984.

[62] C. Gu, *Multivalued morphology and segmentation-based coding*, PhD thesis, Swiss Federal Institute of Technology, Lausanne, 1996.

[63] C. Gu, T. Ebrahimi, and M. Kunt, "Morphological spatio-temporal segmentation for content-based video coding", *in VLBV'95*, Tokyo, Japan, November 1995.

[64] H. Gu, Y. Shirai, and M. Asada, "Mdl-based segmentation and motion modelling in a long image sequence of scene with multiple independently moving objects", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-18, no. 1, pp. 58-64, January 1996.

[65] D. Hepper, "Efficiency analysis and application of uncovered background prediction in a low bit rate image coder", *IEEE Trans. Commun.*, Vol. vol. COM-38, no. 9, pp. 1578-1584, September 1990.

[66] M. Hoetter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation", *Signal Processing*, Vol. vol. 15, no. 3, pp. 315-334, October 1988.

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

[68] B.K.P. Horn and B.G. Schunck, "Determining optical flow", *Artif. Intell.*, Vol. vol. 17, pp. 185-193, 1981.

[69] Y.Z. Hsu, H.H Nagel, and G. Reckers, "New likelihood test methods for change detection in image sequences", *Computer Vision, Graphics, and Image Processing*, Vol. vol. 26, pp. 73-106, 1984.

[70] T.S. Huang, editor, *Image Sequence Processing and Dynamic Scene Analysis*, Springer-Verlag, 1983.

[71] M. Irani, S. Hsu, and P. Anandan, "Mosaic-based video compression", *in SPIE Proc. Digital Video Compression: Algorithms and Technologies*, Vol. 2419, San Jose, CA, February 1995.

[72] M. Irani, B. Rousso, and S. Peleg, "Detecting and tracking multiple moving objects using temporal integration", in Sandini G., editor, *Second European Conference on Computer Vision*, pp. 282–287. Springer-Verlag, S.Margherita, Italy, 1992.

[73] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions", *Int. Journal of Computer Vision*, Vol. vol. 12, no. 1, pp. 5-16, 1994.

[74] ISO/IEC JTC1 CD 11172, "Information Technology - Coding of Moving pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s - Part 2: Coding of Moving Picture Information", Technical report, International Organization for Standardization, 1991.

[75] ISO/IEC JTC1/SC29/WG11, "MPEG-4 Video Verification Model Version 5.1", Technical report, International Standardisation Organization, December 1996.

[76] Motion Picture Expert Group ISO/IEC JTC1/SC29/WG11, "MPEG-4 proposal package description (ppd)", Technical report, International Standardisation Organization, July 1995.

[77] Draft ITU-T, "Recommendation H.263 - Video coding for narrow telecommunication channels at < 64kbit/s", Technical report, July 1995.

[78] B. Jähne, *Digital Image Processing*, Springer Verlag, 1993.

[79] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[80] A.K. Jain and C.D. Dubes, *Algorithms for clustering data*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[81] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun.*, Vol. vol. COM-29, no. 12, pp. 1799-1808, December 1981.

[82] A.D. Jepson and M.J. Black, "Mixture models for optical flow computation", *in IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, pp. 760-761, New-York, USA, June 1993.

[83] G.G. Langdon Jr. and J. Rissanen, "Compression of black-white images with arithmetic coding", *IEEE Transactions on Communications*, Vol. COM-29, No. 6, pp. 858-867, June 1981.

[84] L. Kaufman and P.J. Rousseeuw, *Finding groups in data*, John Wiley&Sons, Inc., New York, 1990.

[85] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding of video conferencing", *in Proc. Nat. Telecommun. Conf.*, pp. G5.3.1–G5.3.5, New Orleans, LA, December 1981.

[86] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high compression image coding", *IEEE Trans. Circuits and Syst.*, Vol. vol. CAS-34, no. 11, pp. 1306-1336, November 1987.

[87] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding techniques", *Proc. IEEE*, Vol. vol. 73, no. 4, pp. 549-575, April 1985.

[88] C.Y. Lee and D.B. Cooper, "Structure from motion: A region based approach using affine transformations and moment invariants", *in IEEE int. Conf. on Robotics and Automation*, Vol. 3, pp. 120-127, Atlanta,GA, 1993.

[89] D. LeGall, "MPEG: A video compression standard for multimedia", *Commun. of the ACM*, Vol. vol. 34, no. 4, pp. 47-58, April 1991.

[90] D. LeGall, "The MPEG video compression algorithm", *Signal Processing: Image Communications*, Vol. vol. 4, no. 2, pp. 129-140, April 1992.

[91] F. Leymarie and D. Levine, "Tracking deformable objects in the plane using an active contour model", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-15, no. 6, pp. 617-634, June 1993.

[92] S.P. Liou and R.C. Jain, "Qualitative motion analysis using a spatio-temporal approach", *in Proc. CVPR'91*, pp. 726–727, Lahaina, Maui, Hawai, 1991.

[93] A. Lippman, *Re-inventing television*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1995.

[94] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", *in Proceedings Image Understanding Workshop*, pp. 121–130, 1981.

[95] F. Marqués, M. Pardàs, and P. Salembier, "Coding-oriented segmentation of video sequences", in L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*, pp. 79–123. Kluwer Academic Publishers, 1995.

[96] D. Marr, *Vision*, W. H. Freeman and Company, New York, 1983.

[97] R. Mech and P. Gerken, "Automatic segmentation of moving objects", Technical report, ISO-IEC/JTC1/SC29/WG11 MPEG96/M0989, Tampere, Finland, July 1996.

[98] P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim, "Robust regression methods for computer vision: a review", *Int. Journal of Computer Vision*, Vol. vol. 6, no. 1, pp. 59-70, 1991.

[99] F. Meyer and P. Bouthemy, "Region-based tracking in an image sequence", in Sandini G., editor, *Second European Conference on Computer Vision*, pp. 476–484. Springer-Verlag, 1992.

[100] F. G. Meyer and P. Bouthemy, "Region-based tracking using affine motion models in long image sequences", *Computer Vision, Graphics, and Image Processing*, Vol. 60, No. 2, pp. 119–140, 1994.

[101] N.C. Mohanty, *Signal processing: signals, filtering, and detection*, Van Nostrand Reinhold Company, New York, 1987.

[102] C.R. Moloney and E. Dubois, "Estimation of motion fields from image sequences with illumination variation", *in Proc. ICASSP'91*, Vol. IV, pp. 2425–2428, Toronto, Canada, May 1991.

[103] F. Moscheni, F. Dufaux, and M. Kunt, "A new two-stage global/local motion estimation based on a background/foreground segmentation", *in Proc. ICASSP'95*, Vol. 4, pp. 2261–2264, Detroit, MI, May 1995.

[104] F. Moscheni, F. Dufaux, and M. Kunt, "Object tracking based on temporal and spatial information", *in Proc. ICASSP'96*, Vol. 4, pp. 1914–1917, Atlanta, GA, May 1996.

[105] D.W. Murray and B.F. Buxton, "Scene segmentation from visual motion using global optimization", *IEEE Trans. Pattern Anal. and Machine Intell.*, Vol. vol. PAMI-9, no. 2, pp. 220-228, March 1987.

[106] H.G. Musmann, M. Hoetter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images", *Signal Processing: Image Communication*, Vol. vol. 1, no. 2, pp. 117-138, October 1989.

[107] H.G. Musmann, P. Pirsch, and H.J. Grallert, "Advances in picture coding", *Proc. IEEE*, Vol. vol. 73, pp. 523-548, April 1985.

[108] H.-H. Nagel, "Constraints for the estimation of displacement vector fields from image sequences", *in Proc. Int. Joint Conf. on Artificial Intelligence*, pp. 945–951, Karlsruhe, Germany, August 1983.

[109] H.H. Nagel, "Displacement vectors dereived from second-order intensity varaitions in image sequences", *Computer Vision, Graphics, and Image Processing*, Vol. vol. 21, pp. 85-117, 1983.

[110] S. Negahdaripour and C.H. Yu, "A generalized brightness change model for computing optical flow", *in IEEE Proc. Int. Conf. on Computer Vision*, pp. 2–11, Berlin, Germany, May 1993.

[111] S.A Niyogi and E.H. Adelson, "Analyzing and recognizing walking figures in xyt", *in Proc. CVPR'94*, pp. 469–474, Seattle, WA, 1994.

[112] J.-M. Odobez, *Estimation, détection et segmentation du mouvement: une approche robuste et markovienne*, PhD thesis, Université de Rennes I, Rennes, France, 1994.

[113] J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models in complex image sequences", *in 7th European Conference on Signal Processing, EUSIPCO'94*, Edinburgh, Scotland, September 1994.

[114] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1991.

[115] M. Pardas and P. Salembier, "3D morphological segmentation and motion estimation for image sequences", *Signal Processing*, Vol. vol. 38, no. 2, pp. 31-43, September 1994.

[116] M. Pardas and P. Salembier, "Time-recursive segmentation of image sequences", in EURASIP, editor, *EUSIPCO 94, VII European Signal Processing Conference*, pp. 18–21, Edinburgh, U.K., September 1994.

[117] M. Pardas, P. Salembier, and B. Gonzalez, "Motion and region overlapping motion estimation for segmentation-based video coding", *in Proc. ICIP'94*, Vol. II, pp. 428–432, Austin, TX, November 1994.

[118] J.W. Park and S.U Lee, "Joint image segmentation and motion estimation for low bit rate video coding", *in Proc. ICIP'96*, Vol. II, pp. 501–504, Lausanne, Switzerland, September 1996.

[119] S. Peleg and H. Rom, "Motion based segmentation", *in IEEE Int. Conference on Pattern Recognition*, pp. 109–113, 1990.

[120] A. Pentland, B. Horowitz, and S. Sclaroff, "Non-rigid motion and structure from contour", *in IEEE Proc. Workshop on Visual Motion*, pp. 288–293, Princeton, NJ, October 1991.

[121] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical recipes in C*, Cambridge, University press, 1992.

[122] E.M. Riseman and M.A. Arbib, "Computational techniques in the visual segmentation of static scenes questionable predictions", *Computer Vision, Graphics and Image Processing*, Vol. vol. 6, pp. 221-276, 1977.

[123] J. Rissanen, "Modelling by shortest description length", *Automatica*, Vol. vol. 14, pp. 465-471, 1978.

[124] P.J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, John Wiley & Sons, Inc, New York, 1987.

[125] S. Sabri and B. Prasada, "Video Conferencing Systems", *Proceedings of the IEEE*, Vol. 73, No. 4, pp. 671-688, April 1985.

[126] P. Salembier, "Morphological multiscale segmentation for image coding", *Signal Processing*, Vol. vol. 38, No. 3, pp. 359-386, September 1994.

[127] P. Salembier and M. Pardàs, "Hierarchical morphological segmentation for image sequence coding", *IEEE Trans. Image Proces.*, Vol. vol. 3, no. 5, pp. 639-651, September 1994.

[128] P. Salembier, L. Torres, F. Meyer, and C. Gu, "Region-based video coding using mathematical morpholgy", *Proc. IEEE*, Vol. vol. 83, no. 6, pp. 843-857, June 1995.

[129] A. Savitzky and M.J.E. Golay, ", *Analytical Chemistry*, Vol. vol. 36, pp. 1627-1639, 1994.

[130] H.S. Sawhney, S. Ayer, and M. Gorkani, "Model-based 2d & 3d dominant motion estimation for mosaicing and video representation", *in ICCV'95*, pp. 583–590, Cambridge,MA, June 1995.

[131] R.J. Schalkoff and E.S. McVey, "A model and tracking algorithm for a class of video targets", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-4, no. 1, pp. 2-10, January 1982.

[132] P. Schroeter and S. Ayer, "Multi-frame based segmentation of moving objects by combining luminance and motion", *in Proc. EUSIPCO 94*, Edinburgh, U.K., September 1994.

[133] M. Schütz and T. Ebrahimi, "Matching error based criterion of region merging for joint motion estimation and segmentation techniques", *in Proc. ICIP'96*, Vol. II, pp. 509–512, Lausanne, Switzerland, September 1996.

[134] M.I. Sezan and R.L. Lagendijk, editors, *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, 1993.

[135] CCITT SG XV, "Recommendation H.261 -video codec for audiovisual services at p*64kbit/s", Technical Report COM XV-R37-E, August 1990.

[136] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3445–3462, December 1993.

[137] A. Singh, *Optical Flow Computation: a Unified Perspective*, IEEE Computer Society Press, 1991.

[138] G. Taubin and D.B. Cooper, "Object recognition based on moment (or algebraic) invariants", in J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pp. 375-397. MIT Press, 1992.

[139] W.B. Thompson, "Combining motion and contrast for segmentation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. vol. PAMI-2, no. 6, pp. 543-549, November 1980.

[140] W.B. Thompson and T.-G. Pong, "Detecting moving objects", *Int. Journal of Computer Vision*, Vol. vol. 4, pp. 39-57, 1990.

[141] C. Toklu, M. Tekalp, A.T Erdem, and M.I. Sezan, "2-d mesh-based tracking of deformabel objects with occlusion", *in Proc. ICIP'96*, Vol. I, pp. 933-936, Lausanne, Switzerland, September 1996.

[142] L. Torres and M. Kunt, *Video Coding: The Second Generation Approach*, Kluwer Academic Publishers, 1995.

[143] P. Treves and J. Konrad, "Motion estimation and compensation under varying illumination", *in Proc. ICIP'94*, Vol. I, pp. 373-377, Austin, TX, November 1994.

[144] Y.T. Tse and R.L. Baker, "Global zoom/pan estimation and compensation for video compression", *in Proc. ICASSP'91*, Vol. IV, pp. 2725-2728, Toronto, Canada, May 1991.

[145] G. Tziritas and C. Labit, *Motion analysis for image sequence coding*, Elsevier, 1994.

[146] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties", MIT, A.I. 917, December 1986.

[147] B.A. Wandell, *Foundations of Vision*, Sinauer Associates, Inc., 1995.

[148] J.Y.A. Wang and E.H. Adelson, "Representing moving images with layers", *IEEE Trans. Image Proces.*, Vol. vol. 3, no. 5, pp. 625-638, September 1994.

[149] J.Y.A. Wang and E.H. Adelson, "Spatio-temporal segmentation of video data", *in SPIE Proc. Image and Video Processing II*, Vol. 2182, San Jose, CA, February 1994.

[150] J. Wiklund and G.H. Grandlund, "Image sequence analysis for object tracking", *in Proc. 5th Scandinavian Conference on Image Analysis*, pp. 641-648, Stockholm, Sweden, June 1987.

[151] R. Wilson and M. Spann, *Image segmentation and uncertainty*, Research Studies Press Ltd., Letchmore,Hertfordshire,England, 1988.

[152] S.F. Wu and J. Kittler, "A differential method for simultaneous estimation of rotation, change of scale and translation", *Signal Processing: Image Communication*, Vol. vol. 2, no. 1, pp. 69-80, May 1990.

[153] S.F. Wu and J. Kittler, "A gradient-based method for general motion estimation nad segmentation", *Journal of Visual Communication and Image Representation*, Vol. vol. 4, no. 1, pp. 25-38, March 1993.

[154] X. Yuan, "Hierarchical uncovered background prediction in a low bit-rate video coder", *in Picture Coding Symposium '93*, p. 12.1, Lausanne, Switzerland, March 1993.

[155] H. Zheng and S.D. Blostein, "Motion-based object segmentation and estimation using the mdl principle", *To appear in: IEEE Trans. Image Processing*, September 1995.

[156] F. Ziliani, "Inseguimento di regioni in movimento in sequenze di immagini con predizione del campo di spostamento secondo kalman", Technical report, Brescia, Italy, October 1996.

Curriculum Vitae

Fabrice Moscheni was born the October 23, 1967, in Sainte-Croix, Switzerland. He received the M.S. in Physics from the Swiss Federal Institute of Technology at Lausanne (EPFL) in 1991. From 1991 and during one year, he worked at FUJITSU Laboratories Ltd., Atsugi, Japan. He then joined the Signal Processing Laboratory of EPFL as a research assistant and, later, as a Ph.D. student. His work involved the supervision of EPFL and ERASMUS students as well as taking part in the European projects dTTb and Vadis. During the summer 1994, he worked at Logitech, Fremont, USA, on the development of a new camera for video conferencing.

His main research interests are in video sequence processing and analysis, motion estimation, second generation coding and higher order statistics. He is a member of the Swiss Technical Society and IEEE.

Bibliography

[1] F. Moscheni, S. Bhattacharjee, and M. Kunt. A framework for spatio-temporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. Submitted paper.

[2] F. Moscheni, F. Ziliani, and M. Kunt. Recursive spatio-temporal segmentation and object tracking for video sequence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. Submitted paper.

[3] H. Eggers, F. Moscheni, and R. Castagno. Robust object tracking based on spatial characterization of objects by additive invariants. In *Proc. ICASSP'97*, Munich, Germany. *To be published*, April 1997.

[4] F. Dufaux and F. Moscheni. Background mosaicking for low bit rate video coding. In *Proc. ICIP'96*, volume I, pages 673–676, Lausanne, Switzerland, September 1996.

[5] F. Moscheni and F. Ziliani. Object tracking based on temporal and spatial information. Technical report, ISO-IEC/JTC1/SC29/WG11 MPEG96/M0962, Tampere, Finland, July 1996.

[6] F. Moscheni and F. Dufaux. Region merging based on robust statistical testing. In *SPIE Proc. Visual Communications and Image Processing '96*, volume 2727, pages 1118–1129, Orlando, Florida, March 1996.

[7] F. Moscheni, F. Dufaux, and M. Kunt. Object tracking based on temporal and spatial information. In *Proc. ICASSP'96*, volume 4, pages 1914–1917, Atlanta, GA, May 1996.

[8] F. Moscheni and S. Bhattacharjee. Robust region merging for spatio-temporal segmentation. In *Proc. ICIP'96*, volume I, pages 501–504, Lausanne, Switzerland, September 1996.

[9] F. Dufaux and F. Moscheni. Motion estimation techniques for digital TV: A review and a new contribution. *Proc. IEEE*, vol. 83, no. 6, pp. 858-876, June 1995.

[10] F. Dufaux, F. Moscheni, and A. Lippman. Spatio-temporal segmentation based on motion and static segmentation. In *Proc. ICIP'95*, volume 1, pages 306–309, Washington, DC, October 1995.

[11] F. Dufaux and F. Moscheni. Segmentation-based motion estimation for second generation video coding techniques. In L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*, pages 219–263. Kluwer Academic Publishers, 1995.

[12] I. Moccagatta, M. Schütz, F. Moscheni, and F. Dufaux. A VQ-based motion field refinement technique for image sequence coding. In *Proc. European Symposium on Advanced Networks and Services*, Amsterdam, The Netherlands, March 1995.

[13] F. Moscheni, F. Dufaux, and M. Kunt. A new two-stage global/local motion estimation based on a background/foreground segmentation. In *Proc. ICASSP'95*, volume 4, pages 2261–2264, Detroit, MI, May 1995.

[14] F. Moscheni and J.-M. Vesin. A genetic algorithm for motion estimation. In *Quinzième Colloque sur le Traitement du Signal et des Images, GRESTI'95*, volume 1, pages 825–828, Juan-Les-Pins, France, September 1995.

[15] F. Dufaux, I. Moccagatta, F. Moscheni, and H. Nicolas. Vector quantization based motion field segmentation under the entropy criterion. *Journal of Visual Communication and Image Representation*, vol. 5, no. 4, pp. 356-369, December 1994.

[16] F. Dufaux, F. Moscheni, and M. Schütz. Motion compensated wavelet transform coding. In *Picture Coding Symposium '94*, pages 13–16, Sacramento, CA, September 1994.

[17] I. Moccagatta, F. Moscheni, M. Schütz, and F. Dufaux. A motion field segmentation to improve moving edges reconstruction in video coding. In *Proc. ICIP'94*, volume III, pages 751–755, Austin, TX, November 1994.

[18] F. Moscheni, F. Dufaux, I. Moccagatta, and M. Schütz. A new motion field enhancement technique for video coding. In *Proc. EUSIPCO 94*, pages 692–696, Edinburgh, U.K., September 1994.

[19] H. Nicolas and F. Moscheni. Multi-level motion estimation for image sequence coding. In *Proc. EUSIPCO 94*, pages 688–691, Edinburgh, U.K., September 1994.

[20] F. Moscheni, F. Dufaux, and H. Nicolas. Entropy criterion for optimal bit allocation between motion and prediction error information. In *SPIE Proc. Visual Communications and Image Processing '93*, volume 2094, pages 235–242, Cambridge, MA, November 1993.

[21] H. Nicolas and F. Moscheni. Temporal redundancy reduction using a motion model hierarchy and tracking for image sequence coding. In *SPIE Proc. Visual Communications and Image Processing '93*, volume 2094, pages 1548–1557, Cambridge, MA, November 1993.