ADAPTIVE VIDEO DELIVERY USING SEMANTICS

THÈSE N^{O} 3236 (2005)

PRÉSENTÉE À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut de traitement des signaux

SECTION DE GÉNIE ÉLECTRIQUE ET ÉLECTRONIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Olivier STEIGER

ingénieur électricien diplômé EPF
de nationalité suisse et originaire d'Altstätten (SG)

acceptée sur proposition du jury:

Prof. T. Ebrahimi, directeur de thèse
Dr A. Cavallaro, rapporteur
Prof. T. Chen, rapporteur
Prof. S. Süsstrunk, rapporteur

Lausanne, EPFL
2005

Remerciements

Durant mon séjour de quatre ans à l'Institut de Traitement des Signaux (ITS) de l'EPFL, j'ai eu l'occasion de rencontrer de nombreuses personnes qui, par leur amitié et par leur soutien, ont contribué à la réussite de mon travail. En premier lieu, j'aimerais chaleureusement remercier le Professeur Touradj Ebrahimi de m'avoir accueilli au sein de son groupe de recherche. Sa confiance inconditionnelle, ses conseils avisés et la générosité avec laquelle il m'a confié de nombreuses et variées responsabilités sans jamais douter de ma capacité à les assumer ont enrichi mon parcours professionnel de manière inestimable. De même, je remercie le Professeur Murat Kunt de m'avoir permis de rejoindre l'ITS. Je me souviendrai avec affection des discussions informelles que nous avons régulièrement menées.

Des remerciements tout particuliers vont à mon ami et collègue Andrea Cavallaro. Malgré la lourde charge de travail que lui impose son poste de professeur à l'Université de Londres, Andrea a su trouver le temps nécessaire pour poursuivre la recherche que nous avions débutée lors de mon travail de diplôme à l'EPFL. Notre collaboration a depuis été jalonnée d'innombrables discussions passionnantes, de multiples publications, et d'un séjour à Londres dont je garde le meilleur souvenir. Sans le soutien et l'amitié d'Andrea, ma vie à l'institut et mon travail de doctorat n'auraient pas eu la même qualité.

Un grand merci va également aux autres membres de mon jury de thèse, le Professeur Tsuhan Chen et Madame la Professeure Sabine Süsstrunk. Leurs remarques intéressantes et constructives ainsi que leurs encouragements m'ont permis d'affiner mon travail de manière appréciable.

Enfin, mes étudiants David, Eric, Guillaume, Cédric et Mathieu, ainsi que les secrétaires et informaticiens de l'institut, Marianne, Fabienne et Gilles, m'ont été d'un très grand secours tout au long de mon travail. Merci à vous!

Toutefois, si mon activité professionnelle m'a rempli de satisfaction, ce sont les nombreuses et belles amitiés nouées et entretenues au cours de cette période qui en constituent la principale richesse. Parmi mes collègues de l'ITS, il en est deux que je voudrais tout particulièrement remercier. Nicolas Aspert d'abord, pour les nombreuses activités musicales, les innombrables discussions et verrées ainsi pour les conseils informatiques avisés qu'il a toujours accepté de partager avec moi. Mon collègue de bureau François Mendels et sa famille ensuite, dont la bonne humeur, l'excellente cuisine et l'amitié ont égayé bon nombre de mes journées et soirées. Je voudrais aussi remercier Yannick, Ulrich, Olivier, Diego, Patricia, Xavier, Patric, Zen, David, Oscar, Rosa, Raphaël, Fred et toutes les autres personnes de l'institut pour les très bons moments passés en leur compagnie.

La pratique de la musique m'a également permis de rencontrer des amis chers et passionnants. Parmi ceux-ci, je voudrais tout particulièrement remercier Manuel Oswald, de l'Orchestre Symphonique Suisse des Jeunes (SJSO); je ne compte plus les bons moments passés en sa compagnie! Merci aussi à Lea, Georges, Piotr, Judith, Valentine, Dominik et Dominic du SJSO, à Nathalie, Benjamin, Claudia, Johannes, Claire, Isaline, Camille et Christophe de l'Orchestre Universitaire de

Lausanne (OSUL), et à tous les autres musiciens et amis que j'ai eu le plaisir de rencontrer.

Enfin, ma plus grande gratitude va à ma famille, en particulier à mes parents et à ma soeur Dominique, ainsi qu'à mon amie Gwendolyn et à mon ancienne amie Catherine. Votre amour est celui qui donne son vrai sens à ma vie.

Contents

Abstract

The diffusion of network appliances such as cellular phones, personal digital assistants and handheld computers has created the need to personalize the way media content is delivered to the end user. Moreover, recent devices, such as digital radio receivers with graphics displays, and new applications, such as intelligent visual surveillance, require novel forms of video analysis for content adaptation and summarization. To cope with these challenges, we propose an automatic method for the extraction of semantics from video, and we present a framework that exploits these semantics in order to provide adaptive video delivery.

First, an algorithm that relies on motion information to extract multiple semantic video objects is proposed. The algorithm operates in two stages. In the first stage, a statistical change detector produces the segmentation of moving objects from the background. This process is robust with regard to camera noise and does not need manual tuning along a sequence or for different sequences. In the second stage, feedbacks between an object partition and a region partition are used to track individual objects along the frames. These interactions allow us to cope with multiple, deformable objects, occlusions, splitting, appearance and disappearance of objects, and complex motion.

Subsequently, semantics are used to prioritize visual data in order to improve the performance of adaptive video delivery. The idea behind this approach is to organize the content so that a particular network or device does not inhibit the main content message. Specifically, we propose two new video adaptation strategies. The first strategy combines semantic analysis with a traditional frame-based video encoder. Background simplifications resulting from this approach do not penalize overall quality at low bitrates. The second strategy uses metadata to efficiently encode the main content message. The metadata-based representation of object's shape and motion suffices to convey the meaning and action of a scene when the objects are familiar.

The impact of different video adaptation strategies is then quantified with subjective experiments. We ask a panel of human observers to rate the quality of adapted video sequences on a normalized scale. From these results, we further derive an objective quality metric, the semantic peak signal-to-noise ratio (SPSNR), that accounts for different image areas and for their relevance to the observer in order to reflect the focus of attention of the human visual system. At last, we determine the adaptation strategy that provides maximum value for the end user by maximizing the SPSNR for given client resources at the time of delivery.

By combining semantic video analysis and adaptive delivery, the solution presented in this dissertation permits the distribution of video in complex media environments and supports a large variety of content-based applications.

Version abrégée

L'utilisation croissante de terminaux tels qu'ordinateurs personnels, téléphones portables et assistants numériques personnels (PDA) a conduit à de nouveaux besoins en matière de distribution personnalisée de données multimédia. Par ailleurs, de récents appareils comme les récepteurs de radio numérique avec affichage d'informations visuelles, et de nouvelles applications telles que la vidéosurveillance, nécessitent des outils d'analyse vidéo avancés afin de permettre l'adaptation et la récapitulation automatique du contenu. Afin de répondre à ces besoins, nous proposons dans cette thèse une méthode automatique d'extraction de données sémantiques, ainsi qu'une structure exploitant ces données afin de permettre la distribution adaptative de contenu vidéo.

D'abord, un algorithme basé sur de l'information de mouvement afin d'extraire de multiples objets sémantiques est proposé. Cet algorithme fonctionne en deux étapes. Dans une première étape, un détecteur de mouvement statistique identifie les régions correspondant à des objets animés. Cette approche est insensible au bruit de la caméra et ne requiert pas de réglages particuliers en cours de séquence ou pour différentes séquences. Dans une seconde étape, les interactions entre une représentation basée sur les objets et une autre représentation basée sur des régions homogènes sont exploitées afin de suivre le parcours des objets au cours du temps. Ces interactions nous permettent de faire face à des difficultés telles que la déformation et la séparation d'objets, les occlusions, l'apparition et la disparition d'objets, et les mouvements complexes.

Ensuite, ces données sémantiques sont utilisées pour prioriser l'information visuelle afin d'améliorer la distribution adaptative de la vidéo. L'idée sous-jacente à cette approche est d'organiser l'information de telle manière à ce qu'un réseau ou un terminal particuliers n'entravent pas le message prioritaire. Spécifiquement, nous proposons deux nouvelles stratégies d'adaptation. La première stratégie combine l'analyse sémantique à un encodeur vidéo traditionnel. A bas débits, les simplifications des parties d'images non prioritaires résultant de cette approche ne pénalisent pas la qualité globale de l'image. La seconde stratégie emploie des méta-données afin d'encoder le message prioritaire. La représentation ainsi obtenue suffit à communiquer la signification d'une scène lorsque les objets décrits nous sont familiers.

L'impact de différentes stratégies d'adaptation est ensuite quantifié à l'aide d'expériences subjectives. Vingt observateurs humains ont évalué la qualité résultant de l'adaptation sur une échelle normalisée. De ces résultats, nous dérivons une mesure de qualité objective, le SPSNR, qui tient compte de l'importance perceptuelle des différentes régions de l'images. Enfin, nous sélectionnons la stratégie qui offre le plus de valeur à l'utilisateur au moment de la distribution en maximisant le SPSNR pour un ensemble de ressources données.

Le regroupement proposé d'un mécanisme d'analyse sémantique et d'une structure de distribution adaptative soutient la distribution efficace de données au sein d'environnements multimédia complexes. De plus, notre solution permet une grande variété d'applications basées sur le contenu sémantique.

Abbreviations and Symbols

Abbreviations

ACR	Absolute Category Rating
CBR	Constant bitrate
CIE	Commission Internationale de l'Eclairage
CPDT	Cascaded pixel-domain transcoder
DCT	Discrete Cosine Transform
D	Descriptor
DS	Description Scheme
EOB	End of block
FCM	Fuzzy C-means
FGS	Fine Granularity Scalability
HVS	Human visual system
IDCT	Inverse Discrete Cosine Transform
LAN	Local area network
MB	Macroblock
MC	Motion compensation
MPEG	Moving Picture Experts Group
MSE	Mean square error
MV	Motion vector
MVR	Motion vector refinement
NN	Nearest neighbor
PSNR	Peak signal-to-noise ratio
QP	Quantization parameter
pdf	Probability density function
ROI	Region of interest
SNR	Signal-to-noise ratio
SPSNR	Semantic PSNR
SVO	Semantic video object
UMA	Universal Multimedia Access
VBR	Variable bitrate
VF	Value function
VLC	Variable length coding
VO	Video object
VoD	Video on Demand

Symbols related to the semantic video analysis framework

c	Number of classes for clustering
χ_2^q	Chi-squared distribution with q degrees of freedom
$\mathcal{D}(\cdot)$	Mahalanobis distance
$\mathbf{D}$	Distance matrix
F	Number of feature categories for data association
$\mathbf{f}_k$	Feature vector corresponding to the kth pixel, $\mathbf{f}_k = [f_{k1}, \ldots, f_{kF}]$
$\Gamma(\cdot)$	Gamma function
H_0	Null hypothesis for change detection
$K_i(n)$	Number of features for region i in frame n
m	Fuzziness exponent
$O_i(n)$	Object i in frame n
$\widetilde{O}_i(n)$	Prediction of object i in frame n
$\boldsymbol{\Phi}_i(n)$	Region descriptor for region i in frame n
$\widetilde{\boldsymbol{\Phi}}_i(n)$	Predicted region descriptor for region i in frame n
Π_o^n	Object partition in frame n
Π_r^n	Region partition in frame n
$\widetilde{\Pi}_r^n$	Predicted region partition in frame n
$R_i(n)$	Region i in frame n
$\widetilde{R}_i(n)$	Prediction of region i in frame n
$S_i(n)$	Set of connected pixels i in frame n
σ^2	Variance
$\tau(i,j)$	Locally adaptive change detection threshold at position (i,j)
U	Fuzzy partition
V	Feature space of data set containing N elements, $V = v_1, v_2, \ldots, v_N$
$\mathbf{v}$	Centroids vector, $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_c]$
$W_{(i,j)}$	Observation window centered on position (i,j)

Symbols related to the adaptive video delivery framework

$d(\cdot)$	Euclidean norm
$\|e\|$	Linear norm of residuals
M_{ijk}	Adapted content version, generated by transcoding the original item A_i using the adaptation operator O_j at resources k
r_o	Outlier ratio
r_p	Pearson linear correlation coefficient
r_s	Spearman rank-order correlation coefficient
$\overline{u}$	Mean score
$V(\cdot)$	Content value

Introduction 1

1.1 Motivations

The previous decade has seen a considerable growth of the amount of electronic information stored and delivered throughout the world. Along with that growth, the means of information transport and access have been widely diversified. The diffusion of network appliances such as cellular phones, personal digital assistants and hand-held computers creates new challenges for content delivery: *how to adapt the media transmission to various device capabilities, network characteristics and user preferences.* Indeed, each device is characterized by certain display capabilities and processing power. Moreover, such appliances are connected through different kinds of networks with diverse bandwidths. Finally, users with different preferences access the same multimedia content. Therefore there exists a need to personalize the way media content is delivered to the end user.

In addition to the above, recent devices, such as digital radio receivers, and new applications, such as intelligent visual surveillance, require novel forms of video analysis for content adaptation and summarization. Digital radios allow for the display of additional information alongside the traditional audio stream to enrich the audio content. For instance, digital audio broadcasting (DAB) allocates 128 Kb/s to streaming audio, whereas 8Kb/s can be used to send additional information, such as visual data [63]. Moreover, the growth of video surveillance systems poses challenging problems for the automatic analysis, interpretation and indexing of video data as well as for selective content filtering for privacy preservation. Finally, the instantaneous indexing of video content is also a desirable feature for sports broadcasting [181].

To cope with the above challenges, video content needs to be automatically analyzed and adapted to the needs of the specific application, to the capabilities of the connected terminal and network, and to the preferences of the user. However, automatic video analysis is a difficult task. Specifically, machines scan visual scenes in raster fashion, thereby analyzing measurable features such as the intensity and wavelength of incoming light. These features are not comprehensible for human beings and need to be translated into meaningful concepts like *objects* and *situations*. The automatic extraction of meaningful information from unstructured video data is referred to as *semantic video analysis*. Semantics-based representations of video content provide the user with flexibility in content-based access and manipulation. Moreover, this allows to achieve improved compression efficiency with object-based coding methods such as MPEG–4.

Also, in order to adapt video content to the capabilities of the connected terminal and network, *adaptive video delivery* is needed. The vast majority of present adaptation methods act on individual coding parameters such as video resolution and bandwidth, and they do not account for semantics. Such content-blind methods are generally suboptimal with respect to human perception. To make up for this drawback, recent content-based adaptation techniques look to exploit semantics in order to further improve performance and to provide novel functionalities such as transmoding and video enhancement. The work presented in this dissertation comes within that framework.

1.2 Investigated approach

This dissertation deals with the problem of exploiting semantics in order to provide adaptive video delivery. In this work, *adaptive video delivery* refers to the distribution of content that matches individual appliance and network resources while providing maximum value for the end user. *Semantics* represent a meaningful entity in the input data. In the digital video domain, this is called a *semantic video object*.

Our solution operates in two stages. In the first stage, meaningful information is extracted by means of semantic analysis. In particular, motion information is used to segment and track semantic vide objects. In the second stage, the semantics are used to prioritize visual data in order to improve the performance of adaptive video delivery. The idea behind this approach is to organize the content so that a particular network or device does not inhibit the main content message.

Semantic video object extraction is organized as follows. First, objects are segmented from the background using segmentation. In our implementation, a statistical change detection process that is robust with regard to camera noise and that does not need manual tuning is used to this end. Then, temporal tracking follows individual objects along the frames. The tracking mechanism is based on feedbacks between a region partition and an object partition. One the one hand, the region partition defines homogeneous groups of pixels corresponding to perceptually uniform regions. These regions are produced by a multi-feature clustering algorithm that operates on both spatial and temporal features. On the other hand, the object partition defines semantic video objects. These correspond to meaningful entities in the real world and they do not usually have invariant physical properties. With respect to the alternative approach of operating at the object level only, the interactions between regions and objects enable us to cope with deformable objects, complex motion and occlusions. The output of semantic video object extraction is then a set of video objects that are coherently labeled over time.

Adaptive video delivery is improved by employing semantics to prioritize visual data. Relevant areas are first extracted from video by means of semantic analysis. These areas are then encoded at a higher quality level or summarized in a textual form. Specifically, we propose two new video adaptation strategies. The first strategy combines semantic analysis with a traditional frame-based encoder. The areas not included in the region of interest are lowered in importance by using background simplification. Using a simplified background aims at taking advantage of the task-oriented behavior of the Human Visual System (HVS) for improving compression ratios. The second strategy uses metadata to efficiently encode the main content message. The use of metadata enables us to make the content more searchable and to improve visualization in video-based applications.

The impact of different video adaptation strategies is further quantified with subjective experiments. We ask a panel of human observers to rate the quality of adapted videos on a normalized scale. Statistical analysis is then used to derive quantitative observations from the results. From

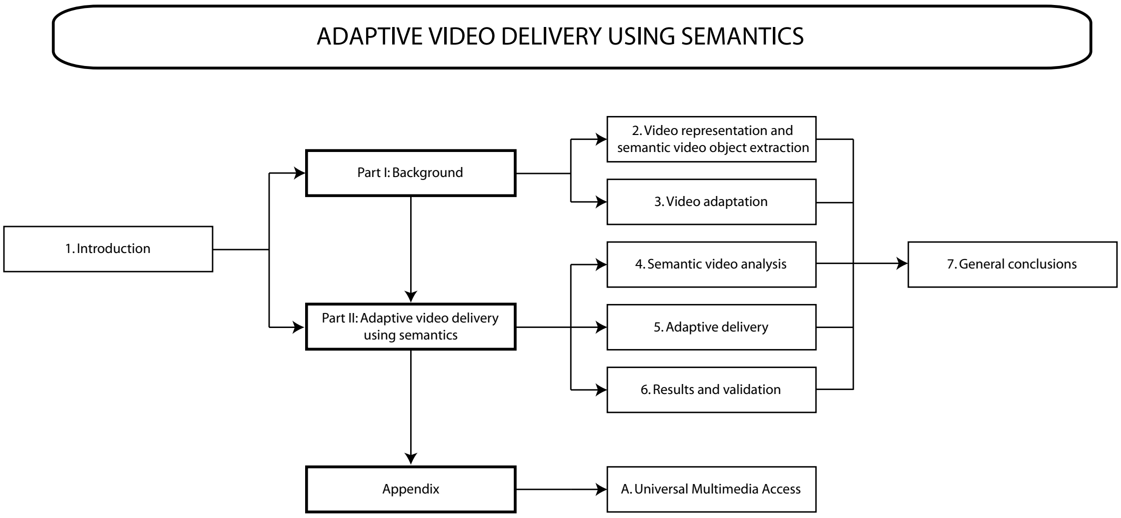

ADAPTIVE VIDEO DELIVERY USING SEMANTICS
1. Introduction
Part I: Background
Part II: Adaptive video delivery using semantics
Appendix
2. Video representation and semantic video object extraction
3. Video adaptation
4. Semantic video analysis
5. Adaptive delivery
6. Results and validation
A. Universal Multimedia Access
7. General conclusions

Figure 1.1: A taxonomy of the content of the dissertation.

these observations, we derive an objective quality metric, the semantic peak signal-to-noise ratio (SPSNR), that accounts for different image areas and for their relevance to the observer in order to reflect the focus of attention of the HVS. The objective metric is needed to overcome the limitations of subjective evaluation experiments that are expensive, time consuming and cannot be used to assess video quality in real time. At last, we are able to determine the strategy that matches individual appliance and network resources while providing maximum value for the end user. This is achieved by measuring the objective quality of different adaptation strategies for given resources at the time of delivery.

1.3 Organization of the dissertation

A taxonomy of the content of this dissertation is depicted in Figure 1.1. Background knowledge related to our work is reviewed in Part I. *Chapter 2* addresses the representation of video and reviews state of the art techniques for the extraction of semantic video objects. First, digital video and its properties are discussed. Then, a framework for the semantic modeling of video is presented. Finally, state of the art techniques for object segmentation and tracking are categorized according to their operations. *Chapter 3* reviews state of the art strategies for video adaptation, such as simulcast, scalable coding and transcoding/transmoding.

In Part II, our framework for adaptive video delivery using semantics is introduced. *Chapter 4* discusses possible semantics for the extraction of meaningful objects and, in particular, it describes the use of motion as semantics. An algorithm that relies on motion information and on temporal tracking to extract semantic video objects in cluttered environment is proposed. *Chapter 5* presents a framework that relies on semantics to provide adaptive video delivery. First, a number of complementary adaptation strategies are discussed. Then, the impact of different adaptation strategies is quantified with subjective and objective experiments. Finally, a mechanism for the selection of the optimal strategy is proposed. *Chapter 6* discusses experimental results obtained with standard test sequences and proposes a validation of our work in real applications. The behavior of the proposed semantic video object extraction algorithm in the presence of various difficulties is analyzed. Semantic video objects and their associated description are further used to provide intelligent visual surveillance. Then, the impact of different adaptation strategies is quantified. At last, the adaptive video delivery framework is tested with real sequences for different client resource profiles.

In order to set our work in the more general context of multimedia delivery, *Appendix A* discusses the use of adaptive video delivery to provide Universal Multimedia Access (UMA).

1.4 Main contributions

The significant contributions of the work presented in this dissertation are summarized below.

- ▷ *Definition of a general tracking strategy for multiple, simultaneous objects.* The strategy is based on feedbacks between an object partition and a region partition. These interactions allow us to cope with deformable objects, motion of non-rigid objects, partial and total occlusions, splitting, and appearance and disappearance of objects.
- ▷ *Definition of two new video adaptation strategies.* The first strategy combines semantic analysis with a traditional frame-based video encoder. The second strategy uses metadata to efficiently encode the main content message.
- ▷ *Quantification of the impact of different adaptation strategies with subjective experiments.* We show that background alterations resulting from semantic prefiltering do not impair overall quality at low bitrates. We also demonstrate that the metadata-based representation of object's shape and motion suffices to convey the meaning and action of a scene when the objects are familiar.
- ▷ *Definition of an objective quality metric, the semantic peak signal-to-noise ratio (SPSNR),* that accounts for different image areas and for their relevance to the observer in order to reflect the focus of attention of the HVS. The prediction performance of the objective metric is quantified with respect to the subjective experiments.
- ▷ *Definition of a strategy selection mechanism for adaptive video delivery.* The method determines the video adaptation strategy that provides maximum value for the end user under given appliance and network constraints. This is achieved by maximization of the objective video quality.

Part I

Background

Video representation and semantic video object extraction 2

2.1 Introduction

By nature, humans and machines “see” in very different manners. Machines scan visual scenes in raster fashion, thereby analyzing measurable features such as the intensity and wavelength of incoming light. Biological vision on the other hand is foveated, highly goal oriented and task dependent. The human camera-eye maneuvers for position and localizes the regions of interest thanks to a sophisticated control [256]. This observation is now seriously being taken into account by the computer vision community in order to represent raw visual data in a more structured form that reflects the *semantics* (i.e., the meaning) of the scene. This is the aim of *semantic modeling*, where the visual space is partitioned into meaningful entities and described by *descriptors* that summarize characteristic features of the partition elements. These structured representations of multimedia content provide the user with flexibility in content-based access and manipulation.

One of the most challenging problems in semantic modeling is the localization of regions of interest within the image. For video, this is referred to as *semantic video object extraction.* A *semantic video object* is defined as a collection of image pixels that corresponds to the projection of a real object in successive image planes or frames of a video sequence [29]. To accurately define the basic nature of semantic objects, the specifics of the vision task must be taken into account. For example, in a highway surveillance application, semantic video objects are vehicles, whereas in airport monitoring, semantic video objects might be people’s faces.

In this chapter, we address the problem of how to represent video in a structured form that reflects content semantics, and we review different techniques for the extraction of semantic video objects. Video representation is discussed in Section 2.2. First, we review digital video and its properties. Then, we present the semantic modeling of video. In Section 2.3, state of the art techniques for semantic video object segmentation and tracking are categorized according to their operations and reviewed.

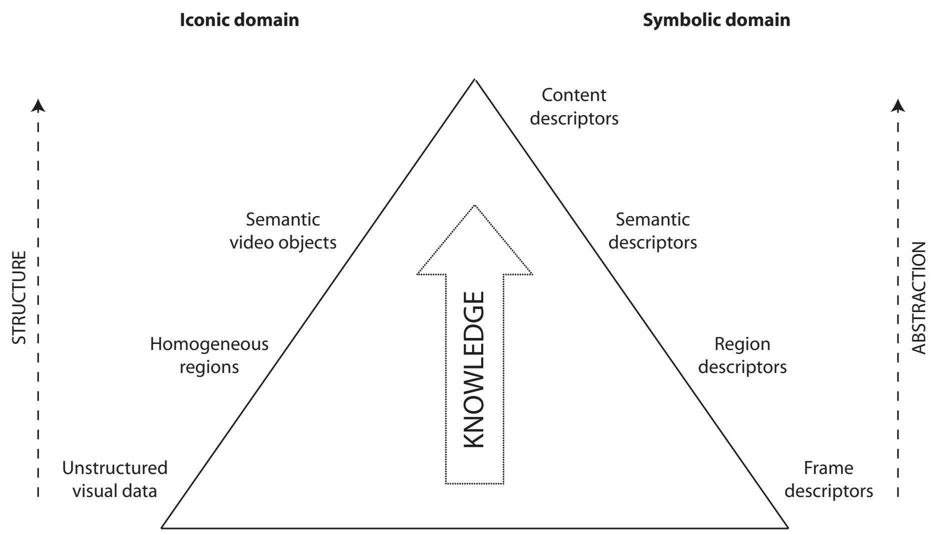


Figure 2.1: The knowledge pyramid. Video representation can take place at different levels, both in the *iconic* (pictorial) and in the *symbolic* (descriptive) domains. Increasing knowledge is provided by structuring visual in the iconic domain, and by using high-level descriptors in the symbolic domain. (Adapted from [29])

2.2 Video representation

Raw data delivered by a digital video camera does not by itself provide any knowledge about the meaning of the captured scene*. Such semantic information must first be extracted either by a human observer or by computer vision, and should then be represented by an appropriate notation and kept for further processing. Video representation can take place at different levels, both in the *iconic* (pictorial) and in the *symbolic* (descriptive) domains. This is depicted by the "knowledge pyramid" in Figure 2.1. The pyramid illustrates how increasing knowledge is provided by structuring video in the iconic domain, and by using high-level descriptors in the symbolic domain.

At the lowest level of organization in the iconic domain, we find the unstructured visual data delivered by the camera. The corresponding *frame descriptors* might represent the brightness or color histograms of the entire frame, for instance. Video might furthermore be structured into homogeneous regions and semantic video objects. *Region descriptors* summarize the values of features that characterize the regions, like region position, motion, shape, color, etc. Similarly, semantic video objects and object groups are described by *semantic descriptors*. The symbolic domain allows for one additional abstraction level. *Content descriptors* represent high-level features that are not actually visible in the frame, but that reflect additional knowledge, such as actor's names, geographical location of the scene, etc.

Next, we review the different video representation levels found in the knowledge pyramid. First, digital video and its fundamental properties are reviewed. Then, semantic objects and boundaries are discussed. Finally, semantic video modeling is presented.

*Note that, although mostly video will be discussed throughout this chapter, many of the addressed concepts apply to still images as well.

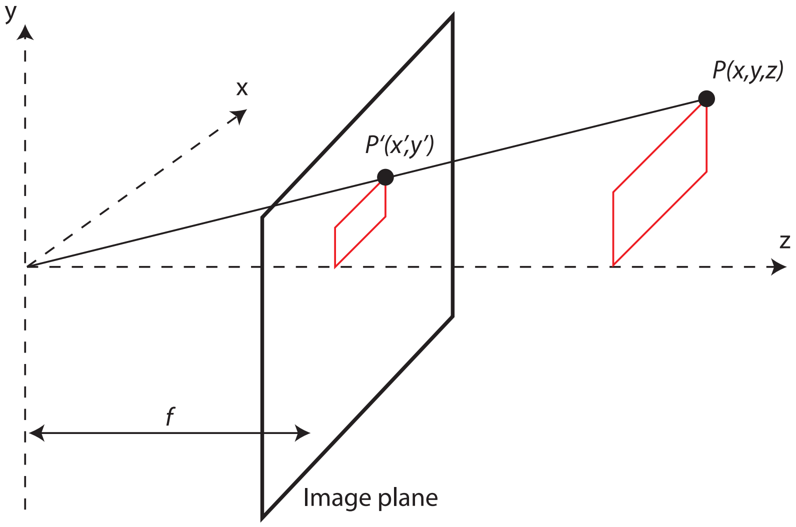


Figure 2.2: Perspective projection geometry. Point P with world coordinates (x, y, z) is projected to point P' with coordinates $(x' = xf/z, y' = yf/z)$ in the 2-D image plane. f is the focal length of the camera lens.

2.2.1 Digital video and its properties

The real world that surrounds us is intrinsically three-dimensional (3-D). The *image* formed on the human eye retina or captured by a TV camera is the result of a perspective projection [26] of the 3-D scene (Figure 2.2). A still image can be modeled by a continuous *image function* of two variables. A scalar function might be sufficient to describe monochromatic images, whereas vector functions are used in image processing to represent multi-spectral (color) images. Function arguments are the spatial coordinates (x, y). Similarly, video is modeled by a three-dimensional *video function*. Function arguments are the spatial coordinates (x, y), together with a time variable t. Function values correspond to the brightness or color at image points.

To process video by computer, it must be represented using an appropriate discrete data structure, like a matrix. Thus, the intrinsically continuous video signal must be converted to a digital signal by *digitization*. The digitization process consists of two steps: *sampling* and *quantization*. Sampling means that the video function $f(x, y, t)$ is sampled into a 3-D matrix with M rows, N columns and L layers. M corresponds to the *vertical frame resolution*, N is the *horizontal frame resolution*, and L is the total number of video frames. Quantization assigns an integer value to each continuous sample: the continuous range of the video function $f(x, y, t)$ is split into K intervals. For an in-depth discussion of the video sampling and quantization process, we refer the reader to the vast literature that is available about these topics [59, 84, 169, 211, 247].

A *digital image* consists of picture elements with finite size. These *pixels* carry information about the brightness/color of a particular location in the image. Usually (and we assume this hereafter), pixels are arranged into a rectangular sampling grid. Such an image is represented by a two-dimensional matrix whose elements are integer numbers or vectors corresponding to the quantization levels in the brightness or color scale. Similarly, digital video is represented by a three-dimensional matrix. Hereafter, we review some fundamental properties of digital video [211, 222]. Knowledge about these properties will be used throughout our work to extract and model the semantics of video signals.

Metric properties

Metric properties have a clear mathematical formulation that requires no other knowledge than the video itself. These properties can be easily extracted and processed by machines. However, their interpretation by the human viewer is awkward, as they do not reflect the way we look at images [174].

- **Distance.** The distance between two points with coordinates (i, j) and (h, k) may be defined in several different ways [19, 195]. The *Euclidean distance* D_E is defined by

  $$D_E\big[(i,j),(h,k)\big] = \sqrt{(i-h)^2 + (j-k)^2}. \qquad (2.1)$$

  The distance between two points can also be expressed as the minimum number of elementary steps in the digital grid which are needed to move from the first point to the second point. If only horizontal and vertical moves are allowed, the *city block distance* D_4 is obtained. If diagonal moves are allowed as well, we obtain the distance D_8, often called *chessboard distance*:

  $$D_4\big[(i,j),(h,k)\big] = |i-h| + |j-k| \qquad (2.2)$$

  $$D_8\big[(i,j),(h,k)\big] = \max\big\{|i-h|, |j-k|\big\}. \qquad (2.3)$$

- **Adjacency.** Pixel adjacency or *connectivity* describes the neighborhoods of a pixel [192, 195] (Figure 2.3). Any two pixels are called *4-neighbors*, or 4-connected, if they have distance $D_4 = 1$ from each other. Similarly, *8-neighbors*, or 8-connected pixels, are two pixels with $D_8 = 1$.

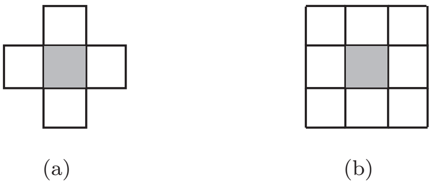


Figure 2.3: Pixel adjacency describes the neighborhoods of a pixel. (a) 4-neighbors. (b) 8-neighbors of the central pixel.

- **Regions and borders.** Let us define a *path* from pixel P to pixel Q as a sequence of points $A_1, A_2, \ldots, A_n$, where $A_1 = P$, $A_n = Q$, and A_{i+1} is a neighbor of A_i, $i = 1, \ldots, n-1$. Then a *region* is a contiguous set, that is, a set of pixels in which there is a path between any pair of its pixels, all of whose pixels also belong to the set [211].

  Assume that R_i are disjoint regions in the frame. Let region R be the union of all regions R_i. We can then define a set R^C which is the set complement of region R with respect to the frame. The subset of R^C which is contiguous with the frame limits is called *background*; the rest of R^C is called *holes*. A region without holes is called a *simply contiguous* region. A region with holes is called *multiply contiguous*.

  The *inner border* of a region R is the set of pixels within the region that have one or more neighbors outside R [6, 193]. This definition corresponds to an intuitive understanding of the border as a set of points at the limits of the region. The *outer border* is the complement of the inner border, that is, the border of the background. Regions and borders are illustrated in Figure 2.4.

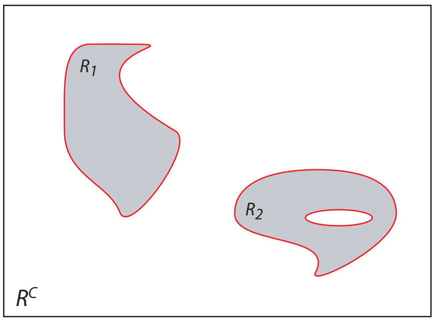


Figure 2.4: Regions and borders. The figure shows two grey regions on white background R^C; the region R_1 is simply contiguous, the region R_2 is multiply contiguous (i.e., region with hole). Region borders are highlighted.

- **Edges.** Edges are pixels where the intensity function (brightness) changes abruptly [53]. An edge vector is given by a magnitude and direction; the edge direction is perpendicular to the gradient direction which points in the direction of image function growth. Unlike border, which is a global concept related to a region, edge is a local property of a pixel and of its immediate neighborhood.
- **Histograms.** Histograms provide global information about video [211]. The brightness histogram $h_f(z)$ of a frame provides the frequency of the brightness value z in the frame. For color video with N spectral channels, one might compute a separate histogram for each channel, or a single N-dimensional histogram.

Perceptual properties

If video is to be analyzed by a human viewer, information should be expressed using variables which are easy to perceive. These are psycho-physical parameters such as color, texture, contrast, borders, etc. Therefore, principles of human perception should be taken into account by digital image processing systems.

- **Contrast.** Contrast is defined as the ratio between average brightness of an object and the background brightness [222]. The human eye is logarithmically sensitive to brightness. Therefore, for the same perception, higher brightness requires higher contrast. It is worthwhile to notice that apparent brightness depends very much on the brightness of the local background. This is illustrated in Figure 2.5. Two squares of the same brightness are superimposed on a dark and light background. Humans perceive the brightness of the squares as different.

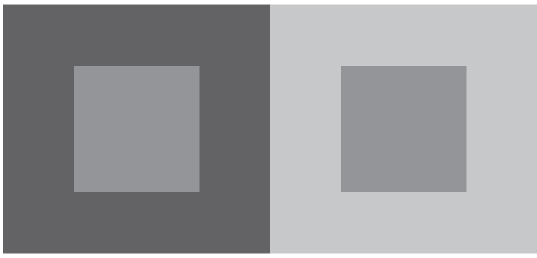


Figure 2.5: Conditional contrast. Two squares of the same brightness are superimposed on a dark and light background. Humans perceive the brightness of the squares as different.

- **Acuity.** *Spatial acuity* is the ability to detect details in a frame. This is defined as the reciprocal of the angular distance which must separate two contours in order that they may be recognized as discrete [87, 190]. The human eye is less sensitive to slow and fast changes in brightness (low or high spatial image frequency) than to intermediate changes. Moreover, acuity decreases with increasing distance from the optical axis and depends on ambient lightning. Similarly, *temporal acuity* refers to the visual sensitivity to a temporally varying pattern at different frequencies. It has also been found to be dependant on viewing distance, display brightness and ambient lightning, and exhibits bandpass properties [190, 247].

  Image resolution is tightly bounded with visual acuity; there is no sense in representing images with higher spatial or temporal resolution than that of the viewer (however, such high resolutions might quite be useful for image processing and computer vision).

- **Color.** Two attributes describe the color sensation of a human being: *luminance* and *chrominance* [68, 247]. *Luminance* refers to the perceived brightness of light, which is proportional to the total energy in the visible band. *Chrominance* describes the perceived color tone of light, which depends on the light wavelength. Chrominance is in turn characterized by two attributes: *hue* and *saturation*. *Hue* specifies the color, which depends on the peak wavelength of the light, whereas *saturation* describes how pure the color is, which depends on the spread or bandwidth of the light spectrum.

  The vast majority of color reproductions do not attempt to reconstruct the spectral composition of the original colors, but only to elicit the same or similar responses in the retina's three types of cones [100]. In television and computer monitors, these responses are produced by causing individually modulated beams of red, green and blue light to excite the cones (CIE RGB primary color system). On print media, color perception is produced by the combination of the base colors cyan, magenta yellow and black (CMYK). For video transmissions, luminance/chrominance coordinate systems derived from CIE XYZ primaries are utilized (YUV for PAL video signals, YIQ for NTSC, YDbDr for SECAM). Unlike the above, *perceptually uniform* color spaces try to mimic the logarithmic response of the eye, i.e., numerical distance in the space is proportional to perceived color difference. Popular perceptually uniform color spaces include CIE 1976 *Lab* and *Luv*. Major color systems, as well as coordinate conversion among these systems, are discussed in [247].

Video quality and noise

Video might be degraded during capture, transmission or processing, and quality measures can be used to assess the degree of degradation. Video quality assessment methods are divided into two categories: *subjective* and *objective* methods. *Subjective* quality assessment methods aid in the compilation and statistical analysis of sample ratings generated by humans. Typically, a selected group of viewers appraise video according to a list of criteria and give appropriate marks. The International Telecommunication Union ITU has issued a set of recommendations which standardize the assessment of subjective video quality [115, 116]. *Objective* methods aim to mathematically estimate the introduced impairment. The quality of the video under test $f(x, y, t)$ is usually estimated by comparison with a known reference video $g(x, y, t)$. Early assessment methods use simple measures such as the *mean square error* (MSE) $\sum\sum(g - f)^2$, the *mean absolute error* $\sum\sum|g - f|$, or the *maximal absolute error* $\max(|g - f|)$ [211]. Such methods fail to measure specific degradations due for instance to signal compression, and they do not take into account the perceptual characteristics of the human visual system (HVS). To meet these needs, perceptual video quality metrics predict image quality using models of the HVS [89, 262].

Noise is a particular class of image degradations due to random errors. Noise may be dependent on, or independent of, image content. *White noise* has a constant power spectrum $S(f) = c$, meaning that its intensity does not decrease with increasing frequency. A better approximation to noise that occurs in many practical cases is obtained by *white Gaussian noise*. Its probability density function is given by the Gaussian (normal) distribution with mean μ and standard deviation σ. *Quantization noise* appears when insufficient quantization levels are used in image compression. In this case, false contours appear. *Impulsive noise* is due to individual noisy pixels whose brightness differs significantly from that of the neighborhood. Saturated impulsive noise is normally called *salt-and-pepper* noise. Image quality degradations due to noise are generally assessed by the *signal-to-noise ratio* SNR:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{F}{E}\right), \tag{2.4}$$

where $F = \sum\sum f^2$ is the total square value of the observed signal $f(x, y)$, and $E = \sum\sum \nu^2$ is the total square value of the noise contribution $\nu(x, y)$.

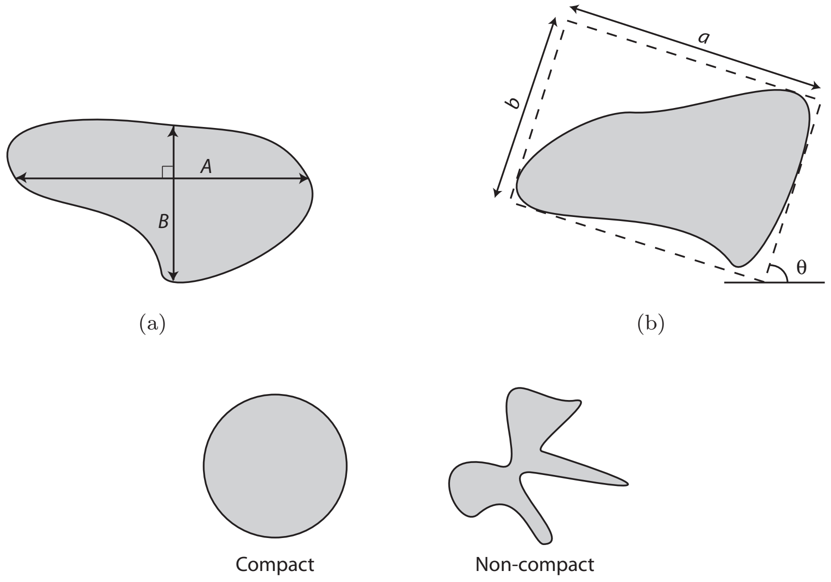
2.2.2 Objects, boundaries and their properties

While the concept of regions only uses the property ‘to be contiguous’, secondary properties can be attached to regions which originate in image perception. Regions that have a strong correlation with objects of the real world are commonly called (*semantic*) *objects*. A *semantic video object* is defined as a collection of image pixels that corresponds to the projection of a real object in successive frames of a video sequence [29]. The semantics of objects may change according to the vision task. For instance, a highway monitoring sequence might be looked at in many different ways. For traffic incident detection, semantic objects are vehicles and people; for accident prevention, semantic objects might be leafs, rain drops or snow flakes; for driver identification, semantic objects are people’s face or license plate numbers, and so on... Thus, the definition of the nature of semantic objects is a complex and sometimes delicate task.

Objects are characterized by a certain number of geometric properties* [211]. Because of the discrete character of digital video, these properties are sensitive to spatial resolution. The simplest and most natural property of an object is its *area*, given by the number of pixels of which the object consists. Other properties are used as well. The *eccentricity* is the ratio of major and minor axes of an object. *Elongatedness* can be evaluated as a ratio between the region area and the square of its maximum thickness. This is often approximated by the ratio of the length and the width of the object bounding rectangle. If the object is elongated, *direction* is the direction of the longer side of a minimum bounding rectangle. Region *compactness* is given by the ratio of squared object boundary length and object area. Some of the above geometric object properties are illustrated in Figure 2.6. In addition, semantic objects are characterized by the value of object pixels, by their distribution (e.g., texture), and by the variation of brightness patterns over time (*optical flow*).

Boundaries bear vital clues about the nature of an object. As a consequence, the human eye is more attracted by object boundaries than by borders of regions that are not bound to objects [222]. Boundaries exhibit a number of important properties such as *perimeter*, *curvature* and *bending energy* [211]. *Perimeter* measures the length of the object boundary. Vertical and horizontal steps have unit length, and the length of diagonal steps in 8-connectivity is $\sqrt{2}$. In the continuous case, *curvature* is defined as the rate of change of slope. In the discrete space, this definition must be slightly adapted to account for difficulties resulting from violation of curve smoothness. One scalar curvature descriptor finds the ration between the perimeter, and the number of boundary pixels where the boundary direction changes significantly. Another approach to calculating curvature from

*These properties apply to homogeneous regions that are not bound to objects as well.



(a)

(b)

(c)

Figure 2.6: Some geometric object properties. (a) Eccentricity is the ratio of the maximum chord A to the maximum chord B which is perpendicular to A. (b) Elongatedness is often approximated by the ratio between the length a and the width b of the object bounding rectangle. The object has direction Θ. (c) Compactness is given by the ratio of squared object boundary length and object area.

digital curves is based on convolution with a truncated Gaussian kernel [141]. The *bending energy* of boundary may be understood as the energy necessary to bend a rod to the boundary shape. It can be computed as a sum of squares of border curvature $c(k)$ over border length L.

2.2.3 Semantic modeling of video

The aim of semantic modeling is to represent raw video data in a more structured form that reflects the meaning of the scene. To emulate the goal oriented and task dependent properties of biological vision, the visual space is partitioned into homogeneous regions, semantic video objects, and object groups, as in Figure 2.7. Corresponding descriptors summarize characteristic features of partition elements. An additional description level represents content features that are not visible in the video. The below notations are based on those introduced by Cavallaro in [29].

Region-based representation

The lowest level of organization that video can take is its subdivision into non-overlapping, homogeneous regions. Regions might be represented in a *region partition* Π_r. The region partition Π_r^n for frame n consists of non-overlapping elements $R_i(n)$ satisfying the following criteria:

$$\begin{cases} I = \bigcup_{i=1}^{N_R^n} R_i(n), \\ R_i(n) \bigcap R_j(n) = \emptyset \qquad \text{if } i \neq j, \end{cases} \tag{2.5}$$

where I is the entire frame, and N_R^n is the number of regions in frame n.

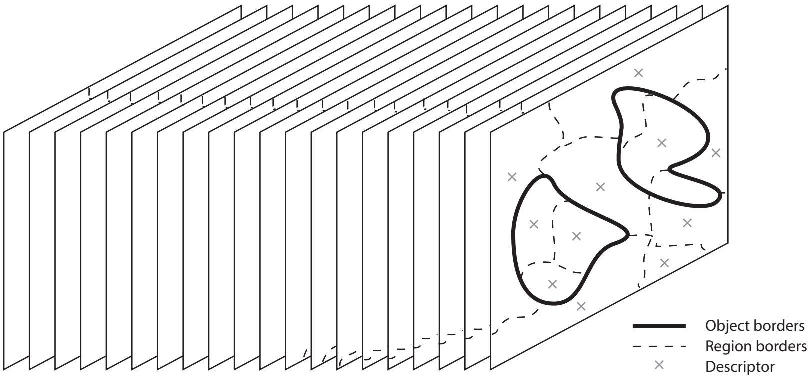


Figure 2.7: Semantic modeling of video. The image space is partitioned into homogeneous regions and semantic video objects. Corresponding descriptors summarize characteristic features of partition elements.

Each region might furthermore be described by a *region descriptor* $\mathbf{\Phi}_i(n)$. The region descriptor is a vector that summarizes the values of features that characterize the region:

$$\mathbf{\Phi}_i(n) = \left(\phi_i^1(n), \phi_i^2(n), \ldots, \phi_i^{K_i^n}(n)\right)^T. \tag{2.6}$$

K_i^n is the number of features used to describe region $R_i(n)$. The number and the kind of features may be different for each region. Typical features include position, motion vectors, color, etc.

Object-based representation

Semantic video objects are represented in an *object partition* Π_o. Unlike regions, the union of all objects is not the entire frame I, but a subset $F(n) \subseteq I$ called *foreground*; the *background* is the set complement $B(n) = I \setminus F(n)$. The object partition Π_o^n for frame n consists of non-overlapping elements O_i satisfying the following criteria:

$$\begin{cases} F(n) = \bigcup_{i=1}^{N_O^n} O_i(n), \\ O_i(n) \bigcap O_j(n) = \emptyset \qquad \text{if } i \neq j, \end{cases} \tag{2.7}$$

where N_O^n is the number of semantic video objects in frame n.

Sometimes, it might be useful to group objects that share common properties such as behavior, interactions or events. The *group partition* Π_g is computed from the object partition Π_o by merging elements. The *foreground partition* $F(n)$ is obtained by merging all objects of Π_o^n in a single element.

Semantic video objects and object groups are described by *semantic descriptors*. The semantic descriptor $\mathbf{\Psi}_j(n)$ can be expressed as a matrix of region descriptors:

$$\mathbf{\Psi}_j(n) = \Big(\mathbf{\Phi}_1(n), \mathbf{\Phi}_2(n), \ldots, \mathbf{\Phi}_{P_j^n}(n)\Big). \tag{2.8}$$

P_j^n is the number of region descriptors comprised in the j^{th} semantic descriptor in frame n. Thus, semantic video objects and groups are described by the descriptors of the regions they enclose.

Content-based representation

Regions and semantic video objects have representations both in the iconic and in the symbolic domains. On the other hand, some high-level features such as actor's names, geographical location and time & date of the filming, might not be actually visible in the video. Therefore, these features have no representation in the iconic domain. Since such high-level content features nevertheless provide useful knowledge about the captured scene, they should be expressed by a *content descriptor*. The content descriptor $\Omega(n)$ is a set whose elements are content features $\omega_k(n)$:

$$\Omega(n) = \bigcup_{k=1}^{Q^n} \omega_k(n), \tag{2.9}$$

where Q^n is the number of features comprised in the content descriptor for frame n. Content features can be of any type, including numerical values and text. As features may evolve over time (e.g., movie filmed at multiple locations), we use the frame-wise notation in Equation (2.9).

Interactions between representation levels

Region-based, object-based and content-based representations are tightly related. The joint representation of region, object and content features allows advanced video access and manipulation for Universal Multimedia Access (Appendix A). Therefore, interaction mechanisms between different representation levels must be provided. In the *combined region-object representation*, a semantic video object, $O_i(n)$, is divided into $N_{R_i}^n$ spatiotemporal regions, such that

$$\begin{cases} O_i(n) = \bigcup_{j=1}^{N_{R_i}^n} R_{i,j}(n), \\ R_{i,j}(n) \bigcap R_{l,m}(n) = \emptyset \qquad \text{if } (i,j) \neq (l,m). \end{cases} \tag{2.10}$$

The number of regions $N_{R_i}^n$ for semantic object i is allowed to vary from frame to frame to account for possible object deformations, occlusion and rotation. Using Equation (2.7) and Equation (2.10), the foreground partition $F(n)$ can the be rewritten as

$$F(n) = \bigcup_{i=1}^{N_O^n} \bigcup_{j=1}^{N_{R_i}^n} R_{i,j}(n). \tag{2.11}$$

Similar to the foreground, the background may also be divided into homogeneous regions. The *background region partition* $B(n)$ then satisfies

$$\begin{cases} B(n) = \bigcup_{i=1}^{N_B^n} B_i(n), \\ B_i(n) \bigcap B_j(n) = \emptyset \qquad \text{if } i \neq j, \end{cases} \tag{2.12}$$

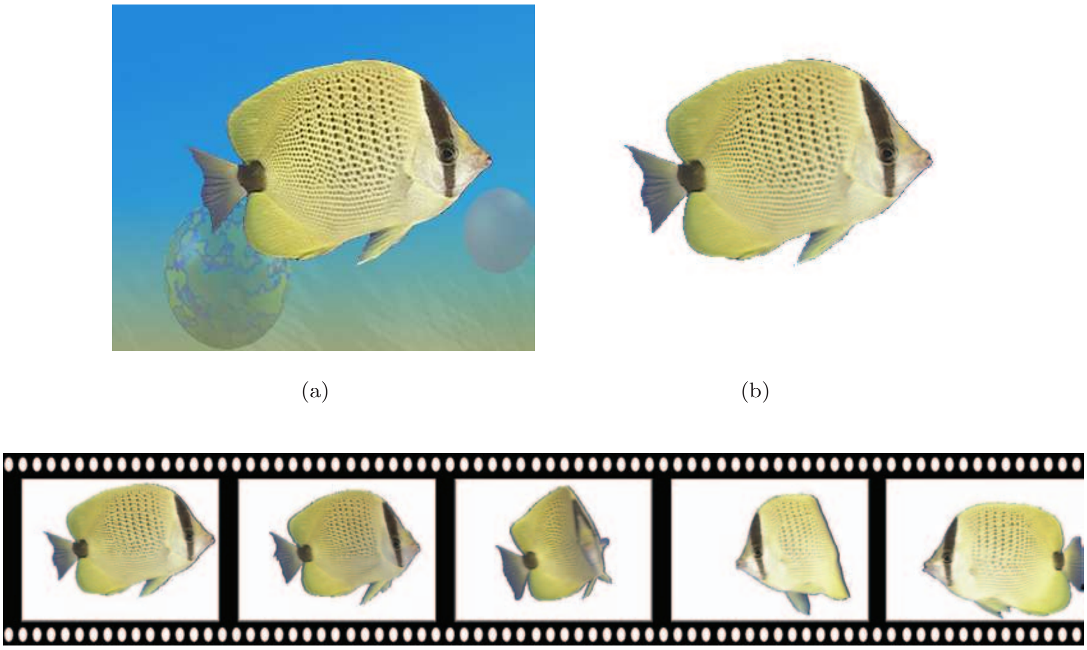
with N_B^n the number of background regions in frame n.

Another important interaction is that between iconic and symbolic representations. The *object-metadata descriptor* $\Psi_j^m(n)$ is used to attach a content descriptor, $\Omega(n)$, to a semantic descriptor, $\mathbf{\Psi}_j(n)$:

$$\Psi_j^m(n) = \Big\{ \mathbf{\Psi}_j(n), \Omega(n) \Big\}. \tag{2.13}$$

This enables us to attach metadata to individual objects.

The above representations require the visual space to be partitioned into meaningful entities. For video, this is achieved by means of semantic video object extraction. Thus, we next review state of the art video object segmentation and tracking techniques.



(a)

(b)

(c)

Figure 2.8: Semantic video object extraction. (a) Sample frame from the original video sequence. (b) Video object segmentation. (c) Video object tracking.

2.3 Semantic video object extraction

Semantic video object extraction refers to the process of *segmenting* and *tracking* semantic video objects (Figure 2.8). The goal of object segmentation is to divide the frame into parts that have a strong correlation with objects or areas of the real world. Semantic video object tracking is the problem of estimating and updating the configuration of an object over time. Semantic video object extraction is a complex task, as the meaning may change according to the application. Moreover, semantic video objects cannot generally be characterized by a simple homogeneity criterion (e.g., uniform color, uniform motion). In the following discussion, we constrain ourselves to the extraction of semantic video objects in the 2-D space using a single, monocular camera. Extensions to the 3-D space and multiple cameras are given by [15, 41, 52, 70].

2.3.1 Segmentation

Semantic video object segmentation, or *complete segmentation*, is a particular case of image segmentation. Image segmentation is the partition of an image into a set of non overlapping homogeneous regions whose union is the entire image or, equivalently, into the edges or boundaries of such regions [27, 142, 207, 211]. These regions correspond to perceptually uniform areas. The goal of semantic video object segmentation is to divide each frame into parts that have a strong correlation with objects or areas of the real world that are visible in the video.

Common approaches to semantic video object segmentation include *manual segmentation*, *matching*, *thresholding*, *edge-based segmentation*, *region-based segmentation*, and *motion-based segmentation*. These are reviewed next.

Manual segmentation

In the case of manual segmentation, semantic video objects are marked directly by the user. This procedure allows a perfect definition of object boundaries. However, it is extremely time consuming. Manual segmentation is thus the preferred approach for specific applications such as high quality film production [187]. It is also used to create a reference segmentation, for instance in order to assess the quality of automatic or semi-automatic video object extraction techniques [44, 67, 145].

Matching

Matching is used to locate known objects in a frame. The goal might be to find objects that are similar to a given template, or to locate occurrences of a selected object in successive video frames (tracking). Match-based algorithms localize all positions at which close copies of the searched template are located. Matching criteria need thus to be established to measure the similarity between the template and parts of the frame. The most basic matching criterion is the *exact match*, where an exact copy of the template is found. However, objects in natural video are usually corrupted by noise, geometric distortion, occlusion, etc. Therefore, a search for locations of *maximum match* is more appropriate. Popular distance metrics include the Euclidean distance as well as variants of the Hausdorff distance [60, 206].

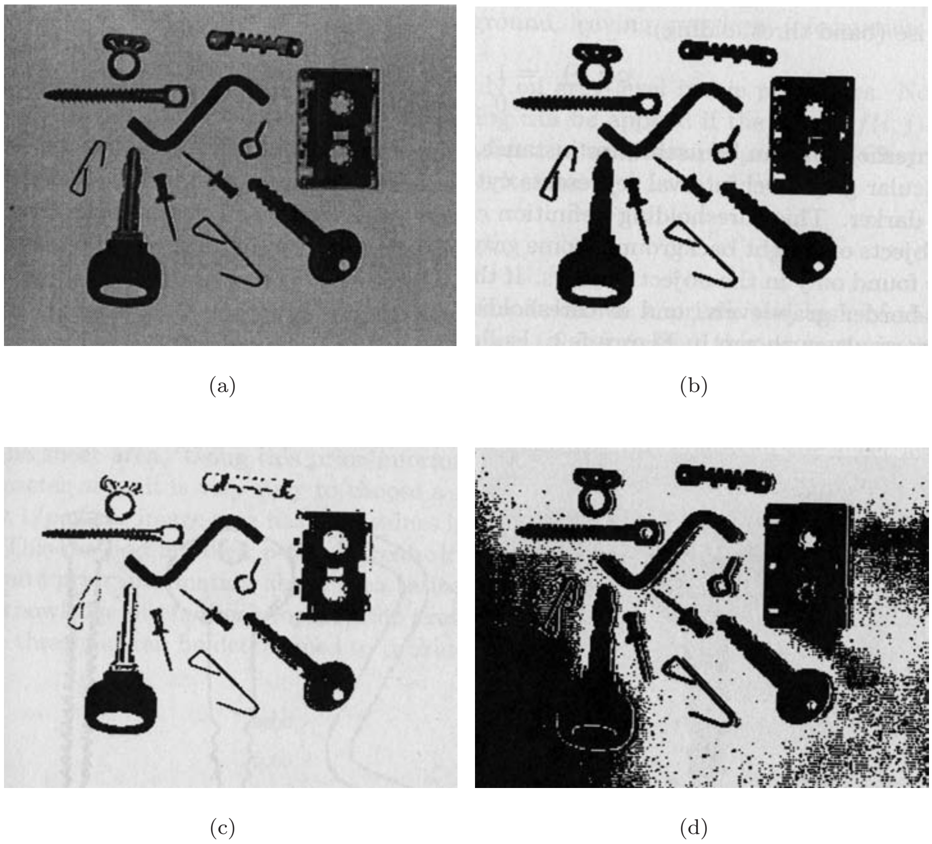
Basic match-based algorithms do not allow for any geometric transformation. This means that the searched object must match the template in size and orientation. A simple but particularly time consuming solution is to scale and rotate the template to all possible sizes and orientations. A better strategy replaces the rigid template by parts connected by rubber links. The goal is the search for good partial matches of template parts in locations that cause minimum force in rubber link connections between these parts [211]. This concept has been used by Umeki and Mizutani [226] to match local features using *Dynamic Link Matching* (DLM). DLM is inherently invariant against distortion and various geometric transformations like shift, scaling and rotation.

Matching is the most effective approach to segment and track occurrences of a specific, known object. However, the need for accurate object templates considerably restricts the application range of matching techniques. Also, this is time consuming even in the simplest case with no geometric transformations, but the process can be made faster if a good operation sequence is found [211].

Thresholding

Gray-level thresholding is the transformation of an input signal f to a binary output signal g, where a pixel $g(i, j, t)$ is classified into foreground if $f(i, j, t) \geqslant T$, and into background otherwise (or vice versa). T is the threshold. Gray-level thresholding is a suitable object segmentation method if objects do not touch each other, and if their gray-levels are clearly distinct from background gray-levels. When more information than is contained in one spectral band is required, thresholding can be extended to multi-spectral signals (e.g., color video). Sonka et al. [211] determine the thresholds independently in each spectral band and combines them into a single segmented video. Better results can be achieved by analyzing multi-dimensional histograms [85].

Basic thresholding as defined above has many variations. *Band thresholding* for instance classifies pixels with values from a set D, $f(i, j) \in D$, into objects, and into background otherwise (or vice versa). This approach is often used to segment objects photographed in front of a uniformly colored background. In the TV and movie jargon, multi-spectral band thresholding is referred to as *chroma-keying* or *blue screening*, and it is commonly used to immerse actors into virtual environments [187].

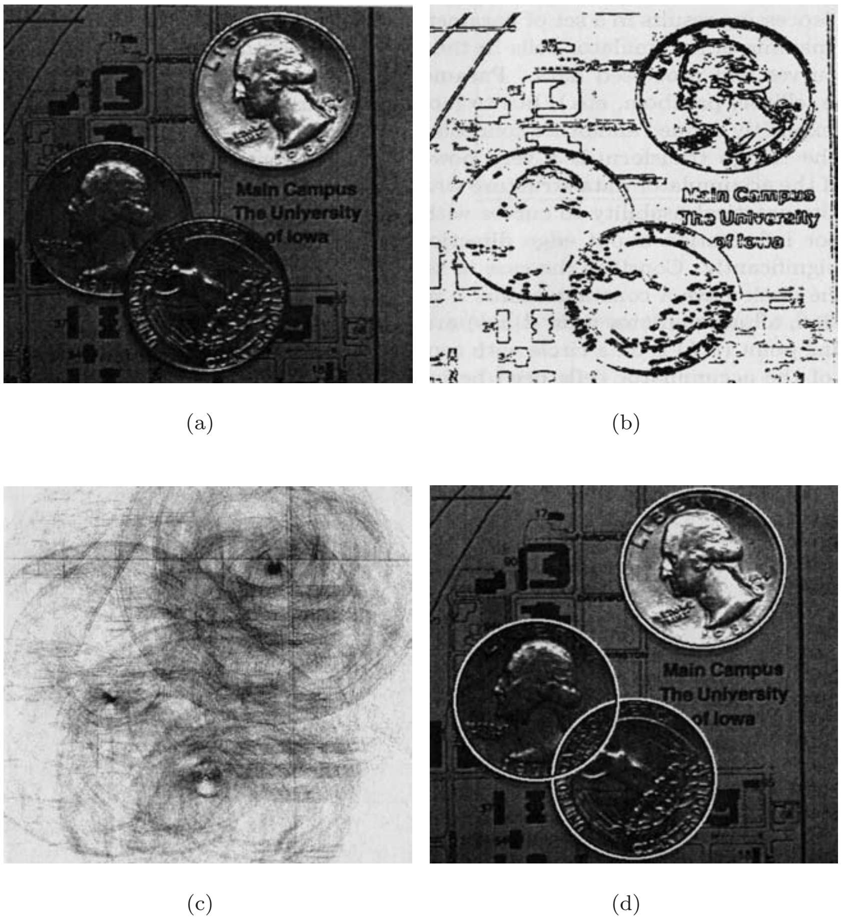


(a) (b) (c) (d)

Figure 2.9: Thresholding for semantic video object segmentation. (a) Original frame. (b) Threshold segmentation. (c) Threshold too low. (d) Threshold too high. (Courtesy of [211])

Correct threshold selection is crucial for successful object segmentation (Figure 2.9). If some property after segmentation is known *a priori*, the task of threshold selection is simplified. *p-tile threshold selection* exploits prior information about the ratio between the frame area occupied by objects and the background area. Based on the histogram, the method selects a threshold T such that $1/p$ of the frame area has values less than T, and the rest has gray values larger than T. However, such *a priori* knowledge is often not available. More complex threshold detection methods are based on histogram shape analysis. If a frame consists of objects of approximately the same gray-levels that differ from the gray-level of the background, the resulting histogram will be bi-modal. Based on this observation, the *mode threshold selection* method finds the highest local maxima in a bi-modal histogram and then detects the threshold as a minimum between them. The accuracy of the method can be improved by taking gray-level occurrences inside a local neighborhood into consideration when constructing a gray-level histogram. For instance, one might weight histogram contributions to suppress the influence of object borders, thus improving the peak-to-valley ratio of the histogram. *Optimal thresholding* methods approximate the histogram using a weighted sum of two or more probability densities with normal distribution [188]. The threshold is set at the gray-level corresponding to the minimum probability between the maxima of the normal distributions.

Thresholding is computationally inexpensive and fast, and can therefore easily be applied in real time. However, this cannot be used reliably in cluttered environments [196].



(a) (b)

(c) (d)

Figure 2.10: Hough transform – circle detection. (a) Original image. (b) Edge image. (c) Parameter space. (d) Detected circles. (Courtesy of [211])

Edge-based segmentation

Edge-based object segmentation relies on edges found by edge detectors. Edges are combined into edge chains that correspond better with borders in the frame. The final aim is to group local edges into an image where only edge chains with a correspondence to existing objects are present [65]. In very simple situations, *edge image thresholding* can produce object borders without any prior information. Simple thresholding of an edge image is applied to remove small edge values resulting from quantization noise, small lightning irregularities, etc. With most real-world videos, this approach is affected by severe over- or under-segmentation. Considering edge properties in the context of their mutual neighbors can improve the result. Rosenfeld et al. [194] for instance notice that a weak edge positioned between two strong edges is highly probably part of an object border. On the other hand, an edge positioned by itself is probably not part of any border. *Edge relaxation* techniques thus evaluate the confidence of each edge regarding its local context [85].

Whenever some prior knowledge is available for boundary detection, general problem-solving methods can be used [163]. Some authors have reformulated object border detection as a *graph searching* problem [147]. A graph is a general structure consisting of a set of nodes, and of oriented and numerically weighted arcs between the nodes. The border detection problem is transformed into a search for the optimal path in the weighted graph, the aim being to find the best path that connects the given starting and ending points. This is achieved by cost minimization, where the cost is generally given by the sum of all arc weights. *Dynamic programming* provides yet another formulation of the border detection problem [79]. The main idea is similar to graph searching, namely to find an optimal path from some starting point to some ending point. Unlike heuristic search however, dynamic programming can search optimal paths simultaneously from multiple starting and ending points. Thus, dynamic programming is the better approach when the location of these points is not exactly known. *Hough transforms* provide an effective solution to object segmentation which can even be used to segment partially occluded objects. The Hough transform is a tool that allows recognition of global patterns (shapes) in an image space by recognition of local patterns (ideally a point) in a transformed parameter space (Figure 2.10). Whereas the original Hough transform [91] was designed to detected straight lines and curves using analytic equations of object borderlines, generalized Hough transforms can find objects even when an analytic expression of the border is not known [101].

Edge-based object segmentation methods provide pixel-accurate object contours when the necessary *a priori* information is available. However, most of these methods are very sensitive to noise [211]. Due to the need for prior information, edge-based methods are generally semi-automatic; that is, some form of user intervention is needed to initialize the algorithm (manual definition of approximate object contour, shape sketch, . . .).

Region-based segmentation

Region-based methods construct object regions directly. The basic idea is to divide a frame into zones of maximum homogeneity, and then to pick out the zones that correspond to semantic objects. The criteria for homogeneity can be based on gray-level, color, texture, shape, semantic information, etc. In general, region-based methods can be divided in two groups: *region growing*, and *split-and-merge*. In region growing algorithms, a number of basic uniform regions (*seeds*) are given, and different strategies are applied to join surrounding neighborhoods. Split-and-merge algorithms start from nonuniform regions, subdivide them until uniform ones are obtained, and then apply some merging heuristics to fit them to maximal possible area.

The basic mechanism underlying all region growing algorithms is to start from some seeds and to grow them until they represent the entire frame. Therefore, a growth mechanism and a criterion checking the homogeneity of the regions after each growth step need to be defined. The simplest homogeneity criterion uses an average gray-level of the region, its color properties, or simple texture properties. Several more advanced homogeneity criteria operating in RGB coordinates are suggested by [223]. A basic growing rule simply considers all adjacent pixels using either 4-connectivity or 8-connectivity [219]. Instead, some approaches consider small adjacent blocks or use search windows to account for features such as texture or edges during homogeneity check [72, 104]. *Morphological operators* and *watersheds* have been used for region growing as well [152, 238].

Split-and-merge algorithms recursively split the frame into smaller and smaller regions until all individual regions are coherent, and then recursively merge these to produce larger coherent regions. The *quadtree* data structure is the most popular data structure in split-and-merge algorithms because of its simplicity and computational efficiency. The quadtree is a tree in which each node has exactly four descendants. The root of the tree corresponds to the entire frame, and each leaf node represents a homogeneous region. Splitting and merging correspond to building or removing parts of the quadtree. Merging of adjacent regions is allowed if they satisfy a homogeneity criterion. The conventional split-and-merge algorithm [90] is lacking in the adaptability to semantics because of its stiff quadtree-based structure. To directly reflect semantics to the segmentation results, Yang and

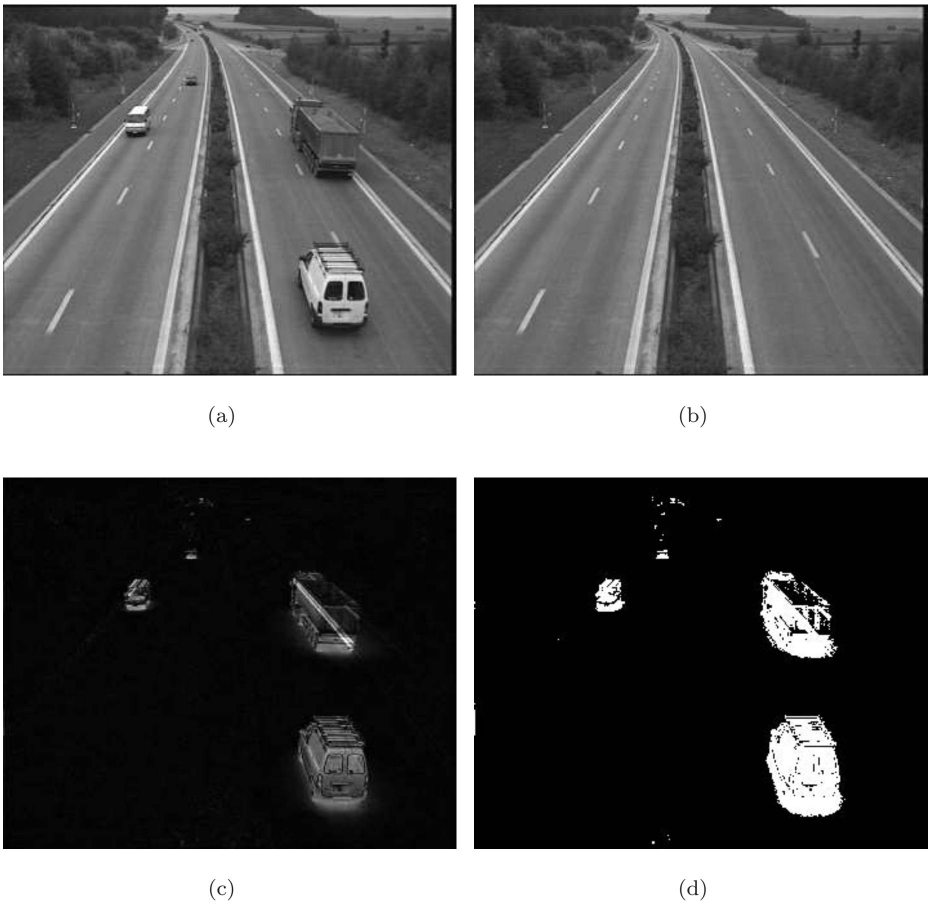


Figure 2.11: Motion-based segmentation through simple differencing. (a) Current frame. (b) Reference frame. (c) Difference image. (d) Binary change mask.

Lee [255] employ a thresholding technique in the splitting phase of the split-and-merge segmentation scheme. Numerous other variations in the split-and-merge strategy are reviewed in [142]. They rely on as diverse approaches as clustering, morphological operators, fuzzy expert systems, etc.

Region-based object segmentation techniques are generally better in noisy video, where borders would be difficult to detect. However, they are often affected by either over-segmentation or under-segmentation as a result of non-optimal parameter selection [211]. Thus, region post-processing is sometimes applied to improve classification results [1, 137].

Motion-based segmentation

Motion-based segmentation aims at detecting regions corresponding to moving objects, such as humans and vehicles. A significant approach for motion-based segmentation is *change detection* [184]. The core problem of change detection is to identify the set of pixels that are significantly "different" between a pair of images of the same scene, taken at two different times. Uninteresting forms of change, such as those caused by sensor noise or illumination variation, are thereby to be rejected. Change detection usually involves three distinct steps: *pre-processing*, application of a *decision rule*, and *post-processing*. Some change detection methods integrate these steps together, while others may not perform pre-processing or post-processing.

A necessary pre-processing step is *image registration*, the alignment of image pairs in the same coordinate frame. This is of particular importance when the camera is moving. Camera motion estimation [37, 61, 172, 266] is usually followed by either camera control [25, 165], or global motion compensation [117, 149]. Sometimes, the two steps of global motion estimation/compensation and change detection are integrated together [51, 166]. Some of the earliest attempts at illumination-invariant change detection made also use of *intensity normalization* [136]. The pixel intensity values in the second image are normalized to have the same mean and variance as those in the first image. *Homomorphic filtering* has been used to separate the illumination and reflectance components of the intensity signal [221]; the reflectance component is provided as input to the decision rule step of the change detection process.

The decision rule of change detection algorithms is the process that decides whether a pixel has changed or not. It can be formulated independently at each pixel $\mathbf{x}$, or it might be based on a small block of pixels in the neighborhood of $\mathbf{x}$. The latter is usually more robust to noise. A widespread decision rule is *simple differencing*, where the difference image $D(\mathbf{x}) = I_2(\mathbf{x}) - I_1(\mathbf{x})$ is thresholded to generate the binary change mask (Figure 2.11). For color video, the pixel-wise Euclidean distance is used instead [73]). The threshold is either chosen empirically, or computed for a desired false alarm rate [210]. The decision rule might also be cast as a *statistical hypothesis test*. The decision as to whether a change has occurred at a given pixel $\mathbf{x}$ or not corresponds to choosing one of two competing hypothesis: the *null hypothesis* $\mathcal{H}_0$, or the *alternate hypothesis* $\mathcal{H}_1$, corresponding to *no-change* and *change* decisions, respectively. The image pair $(I_1(\mathbf{x}), I_2(\mathbf{x}))$ is viewed as a random vector. Knowledge of the conditional joint probability functions $p(I_1(\mathbf{x}), I_2(\mathbf{x})|\mathcal{H}_0)$ and $p(I_1(\mathbf{x}), I_2(\mathbf{x})|\mathcal{H}_1)$ then allows to choose the hypothesis that best describes the intensity change at $\mathbf{x}$ using the framework of hypothesis testing (e.g., significance tests, likelihood ratio test [2, 3], probabilistic mixture models [16]). To exploit the close relationship between nearby pixels both in space and time, sophisticated change detection algorithms replace the actual image pair by *predictive models*. Spatial predictive models fit the intensity value of each image block to a polynomial function of the pixel coordinates $\mathbf{x}$ [93, 208]. Temporal prediction models pixel intensities over time as an autoregressive (AR) process [66]. The change mask is then generally obtained by statistical hypothesis tests. The need for post-processing arises when the results of change detection are noisy or inadequately smooth. This is typically resolved by morphological operators, contour relaxation, and by imposing a minimum region size [211].

Of course, there are other approaches for motion-based segmentation. *Optical-flow based segmentation* uses characteristics of flow vectors [9] over time to detect moving objects. Meyer et al. [151] for instance compute the displacement vector field to initialize a contour-based tracking algorithm for the extraction of articulated objects. *Probabilistic approaches* have been used as well. Friedman and Russell [75] implement a mixed Gaussian classification model for each pixel. This model classifies the pixel values into three predetermined distributions corresponding to background, foreground and shadow. According to the likelihood of membership, the model also updates the mixed components automatically for each class.

Motion-based segmentation is commonly used to automatically detect moving objects in video sequences. Change detection approaches are generally fast and can thus be used for real-time applications. However, they often show poor performance with significant illumination changes and noise. Robust methods based on local models tend to be slower [184]. Most optical-flow based methods on the other hand are computationally complex and very sensitive to noise, but they can be used to detect independently moving objects even in the presence of camera motion [96].

2.3.2 Tracking

Semantic video object tracking is the problem of estimating and updating the configuration of an object over time [88]. In this sense, tracking refers to a process that *follows* an object as it moves around. Real-world scenes normally involve multiple interacting and deforming objects. Therefore, tracking algorithms must be able to effectively deal with appearing and disappearing objects, temporal variations of the 2-D shape of semantic video objects due to perspective and deformable objects, occlusions and other interactions, and splitting of one object. The goal is to establish a stable track for each object.

Tracking methods can be divided into four major categories [96]: *feature-based tracking*, *active contour-based tracking*, *region-based tracking*, and *model-based tracking.* Algorithms from different categories might also be integrated together to form *hybrid tracking methods.* Next, state of the art tracking methods from each category are reviewed.

Feature-based tracking

Feature-based tracking algorithms provide estimates of the state of visual features, or *targets*, in successive frames. The target state is a vector that summarizes the past history of the target sufficiently in order to predict its future. It typically consists of both kinematic components (position, velocity, etc.) and feature components (e.g., spectral characteristics). Depending on the task at hand, global features, local features, dependance graphs or a combination of the above can be used. Global features are centroids, perimeters, areas, colors, etc. Local features include line segments, corner vertices, edges, and regions of high visual contrast. The features used in dependence graph-based algorithms include a variety of distances and geometric relations between features [69].

Kalman filters are being widely used for feature tracking [119, 211]. The goal of Kalman filtering is to combine the measurement taken from the target with the information provided by a motion model in order to obtain an optimal estimation of the target state. A major drawback of the Kalman filter in realistic tracking scenarios is the assumption of a single motion model. Targets undergoing occasional maneuvers cannot be tracked reliably. An important state estimator for such scenarios is the *Interacting Multiple Model filter* (IMM) [18, 148]. IMM filters use several possible motion models and a probabilistic switching between these models. The target's motion is described by one of the models during each sampling period. A two-model IMM filter for instance uses one model with a small maneuver level to represent motion when the target is not maneuvering, and another model with a larger maneuver level to describe the maneuvering phase.

Whenever multiple targets are to be tracked, an additional data association stage must establish which measurement, if any, is to be used in a state estimator. A number of data association algorithms are reported in the literature. The *Nearest Neighbor filter* (NN) selects the closest validated measurement to the predicted measurement to update the track [8, 134]. The *Joint Probabilistic Data Association filter* (JPDA) uses a weighted average of all validated measurements to update the track state [5, 8, 185]. The *Multiple Hypothesis Tracker* (MHT) [45, 186] forms a number of plausible ways (hypothesis) to partition the measurements into tracks and false alarms. Then, the probability of each hypothesis is evaluated. While NN filters works well with widely spaced targets, JPDA and MHT algorithms lead to more accurate results in cluttered target environments. Late research has also considered the use of *assignment algorithms* [182]. The combination of IMM state estimation and assignment algorithm-based data association has been shown to perform particularly well on multitarget feature tracking in complex scenes [4].

An adaptation of the above methods to object tracking is presented by Beymer et al. [12]. Here, the parts to be tracked are the corners of the objects. Tracking parts of objects results in stable tracks for the features under analysis even in case of partial occlusion of the object. However, the problem of grouping the features to determine which of them belong to the same object is a major drawback of these approaches. A solution is provided by Rosenberg and Wermann [191]. For each target, they measure the displacement probability, that is, the confidence that the target moves differently than the background. The boundary of a moving object maximizes the displacement probability of the targets it encompasses. This reflects the assumption that a moving object contains targets with a motion different than the background. The measurements are further used to decide whether a cluster belongs to a single moving object or to multiple objects.

As they operate on 2-D image planes, feature-based tracking algorithms are well fitted for real-time processing and multiple object tracking. They can also handle partial occlusion by using information on object motion, local features and dependency graphs. However, the stability of dealing effectively with occlusion, overlapping and interference of unrelated structures is generally poor [96].

Contour-based tracking

Contour-based methods track objects by representing their outlines as bounding contours and by updating these contours dynamically in successive frames. In the simple algorithm proposed by Gu and Lee [82], a human operator first indicates the rough boundary of a semantic video object in the initial frame. The initial boundary is then refined using morphological segmentation. Semantic video objects in the remaining frames are obtained using perspective motion compensation of the previous video object plus morphological boundary refinement. Due to the rigid body motion compensation followed by adjustments, large non-rigid movements cannot be handled by this method.

One improvement of the previous method is to use a deformable object motion model, such as *active contour models* [121, 220] or *2-D meshes*, instead of rigid motion. Paragios and Deriche [171] address the detection and the tracking problem in a common framework that employs a geodesic active contour objective function and a level set formulation scheme. To overcome the problem of partial occlusion, Peterfreund [178] has introduced a Kalman filter and optical flow measurements in the active contour model. During the update step of the snake state, spurious measurements which are not consistent with previous estimation of motion are rejected. Isard and Blake [103] describe complex motion models using stochastic differential equations and combine this approach with deformable templates. They achieve robust tracking of agile motion in clutter in near real time.

2-D meshes have been used to track video objects as well. Günsel et al. [83] represent video objects by 2-D triangular meshes, where all motion- and shape-related features of an object can be computed as a function of only spatial node configurations and their motion trajectories. Mesh propagation and refinement is then employed to compute the motion trajectories of all mesh node points. Zhao et al. [267] introduce the concept of occlusion mesh models to track multiple interacting and deforming objects. Essentially, they predict whether mesh nodes are going to be occluded by using nodes motion estimation. Such nodes are considered to be unreliable and deleted.

In general, contour-based tracking methods lead to an accurate video object shape definition. However, the tracking reliability is limited at the contour level [211]. Moreover, contour-based algorithms are highly sensitive to tracking initialization, making it difficult to start tracking without user intervention.

Region-based tracking

Region-based methods track objects according to variations of regions that roughly correspond to the 2-D shapes of video objects. In contrast to contour-based algorithms, the tracking strategy relies on information provided by the entire region. Examples of such information are motion, color and texture. Favalli et al. [71] exploit motion information contained in MPEG–2 bitstreams to roughly track video objects marked by the user. Their procedure is very sensitive to motion estimation errors and cannot operate at units smaller than one macroblock. To reduce the influence of motion estimation errors on tracking accuracy, Lee et al. [127] assign a confidence measure to each motion vector. MVs with low confidence are ignored by the tracking process. Sanchez and Dibos [198] perform motion estimation at the pixel level by employing optical flow. In addition, they recover the most likely trajectory of an occluded object from optical flow data at the border of the occlusion.

Color and texture have also been used for object localization and tracking. Comaniciu et al. [42] propose a nonrigid object tracking method based on color histogram matching. Histograms regularized by spatial masking with an isotropic kernel are used to model both the target object and candidates. For target localization, a similarity metric derived from the Bhattacharyya coefficient is minimized using the mean shift procedure. The algorithm successfully copes with camera motion, partial occlusions, clutter, and target scale variations. To account for possible non-rigidity of objects, Liu and Chen [138] treat the color distribution as a weighted color histogram instead of relying on kernel properties. They also include edge density as another tracking feature. Finally, they replace the Bhattacharyya similarity metric by a weighted combination of the Kullback-Leibler distance for color distributions, and a sigmoid function for edge density.

Although they work well in scenes containing only a few objects, region-based tracking algorithms cannot reliably handle occlusion between objects [96].

Model-based tracking

Model-based tracking algorithms match projected object models, produced with prior knowledge, to video data. Typically, these models reflect known geometric, kinematic, dynamic and/or appearance features of the object. Model-based tracking methods can be divided in two distinct categories: *rigid object tracking*, and *non-rigid object tracking*. One of the most common application of rigid object tracking is model-based vehicle tracking. Early work by Gardner and Lawton [78] uses a wire-frame surface model of the vehicle that is manually instantiated by the user. The model indicates the initial position, size and orientation of the vehicle. A translation-based Kalman filter is used to track local features. During feature extraction, the density of features at each frame position is determined by the model. The model also allows the tracker to intelligently partition the features. This simple system reliably tracks a single vehicle shot with an uncalibrated hand-held camera under favorable lightning conditions. Yet it is not able to handle complex vehicle movements due to the translational motion model of the tracker. To address this problem, Koller et al. [126] define a more complex object model that takes into account the geometry, but also the motion of the vehicle. An iterative approach is used to find the best correspondence between 3-D model edge segments and 2-D image segments in each frame. The inclusion of an illumination model allows to take shadows of the vehicle into account during the matching process. Motion estimation is then performed by a recursive estimator based on the vehicle motion model. The later kind of approaches has been extended to provide multi-vehicle tracking [97]. Regions of interest that contain moving vehicles are detected, and each is handled independently. So multi-vehicle tracking can be decomposed into several single-vehicle tracking tasks, provided that there is only light occlusion.

Human body tracking is a very common application of non-rigid object tracking. To track a walking person, Huang and Huang [99] introduce a 2-D articulated cardboard body model, where each body part is defined by an isosceles trapezoidal. They set up a mixture motion model for body movements and then solve body motion parameters in a statistical framework using the Expectation Maximization (EM) algorithm. Instead of an articulated cardboard model, Ning et al. [164] use truncated cones to model arms, legs, the torso and the neck, and a sphere for the head. They also define the posture of a walker by a 12-dimensional vector. The vector represents the global body position as well as joint angles. Both boundary and region information are then combined to achieve precise and robust pose estimation. Model-based tracking of individual body parts has been proposed as well. Nickels and Hutchinson [162] combine a Kalman feature tracker and object models to track complex articulated objects such as (robot-)arms. Five models are used: a general nonlinear system model, an object geometric model, an object appearance model, an imaging model and a dynamic model. The system model instantiates functions of the other models. The object geometric model describes the size and shape of each link of the articulated object and how the links move with respect to one another. The appearance model describes the color, texture and materials used on each link. The imaging model describes the camera used to film the scene. The dynamic model describes the assumptions about motion through the state space. These models serve to generate state estimate of the object for each frame, which is updated by the feature measurements obtained by the feature tracker. Pighin et al. [180] automatically recover the face position and facial expression given video footage of a person's face. They use a continuous optimization technique to fit a 3-D face model to each frame. The face model is based on a set of 3-D texture-mapped models, each corresponding to a particular basic facial expression, that are linearly combined using morphing.

By making use of prior knowledge of the 3-D contours or surfaces of objects, model-based tracking algorithms are intrinsically robust. They make it possible to solve the problem of tracking partially occluded objects. The algorithms also naturally acquire the 3-D pose of objects after setting up the geometrical correspondence between 2-D image coordinates and 3-D world coordinates by camera calibration. However, model-based tracking is computationally expensive and presents two major drawbacks. One is the need for object models with detailed geometry for all objects that could be found in the scene, the other is the lack of generality. This last drawback prevents the system from detecting objects that are not in the database.

Hybrid methods

The last group of tracking approaches is designed as a hybrid between some of the above techniques. These approaches exploit the respective advantages of the different techniques that are combined together. Rigoll et al. [189] design their solution to person tracking as a combination of model-based and feature-based tracking. The shape model of a person's body is automatically learned and acquired by a Pseudo-2D Hidden Markov Model (P2DHMM). The measurement vector generated by the P2DHMM is then used by a Kalman filter to track the person by estimation of a bounding box trajectory indicating the person's location within the entire video sequence. Zhou et al. [269] combine contour-based and region-based tracking to track a single deforming object accurately. They first extract a thin subregion that covers the contour of a video object. This subregion is then tracked using any region-based algorithm.

Some researchers use a combination of region-based and feature-based techniques. Marqués and Llach [146] and Tsaig and Averbuch [224] exploit the advantages of the two first by considering the object as an entity, and then by tracking its parts. These algorithms exploit a video representation as partition hierarchy and track video objects based on interactions between different levels of the hierarchy. The hierarchy is composed of an object level and a region level. The object level defines the topology of the video objects. The region level defines the topology of homogeneous areas constituting the objects. This characteristic allows the tracking system to deal with the deformation of objects. This flexibility is obtained at the cost of a higher computational complexity. Such complexity is due to the use of complex motion models to project and adapt the regions from one frame to another.

2.4 Summary

In this chapter, we have addressed the problem of how to represent video in a structured form that reflects content semantics. Structured video representations reflect the way human observers analyze visual scenes, and they provide the user with flexibility in content-based access and manipulation. In Chapter 5, semantics are employed to prioritize visual data in order to improve adaptive video delivery. In Chapter 6, such information enables video enhancement in order to put important objects in a conspicuous position for the monitoring personnel. Moreover, object descriptors are used for automated event detection. In Appendix A, the joint representation of region, object and content features enables us to provide Universal Multimedia Access.

In addition to the above, we have also reviewed state of the art techniques for the extraction of semantic video objects. In Chapter 4, the advantages of region-based and feature-based techniques are exploited to segment and track multiple moving objects in cluttered background. Our method is able to cope with object deformation and with various track management issues including the appearance and disappearance of objects, object splitting, and occlusions.

Video adaptation 3

3.1 Introduction

Today's digital world is populated by a steadily increasing amount of multimedia content. Such content exists in various modalities and formats, and is accessed over diverse networks and terminals by different users. To cope with individual user preferences, and with different networks and appliances, content adaptation is necessary. Content adaptation refers to the preparation of content that matches the resources of the connected terminal/network, as well as user preferences, in an optimal way. Typical adaptation parameters include the bitrate of the delivered stream, frame resolution and frame rate, audio bandwidth, the coding format, etc. Although the principles underlying content adaptation are similar for all modalities, the remainder of this chapter will focus on the adaptation of video. Still images, audio and text adaptation are notably discussed in [157].

The simplest way to provide video adaptation is to encode several versions for all possible resource profiles. The most adequate version is then selected among the coded ones according to the characteristics of the connected appliance and network. This approach is referred to as *simulcast*. However, the diversity of networks and terminals in a realistic delivery environment usually makes it unattainable to generate a distinct content version for each profile of resources. *Scalable coding* eliminates the need for multiple coexisting content versions by obtaining lower qualities, spatial resolutions and/or temporal resolutions by truncating certain layers or bits from the coded stream.

With simulcast and scalable coding, potential resource profiles need to be known prior to coding in order to select adequate encoding parameters. With transcoding on the other hand, the input bitstream is converted according to the needs of the connected appliance *on the fly*. *Video transcoding* is the process of converting an existing compressed video signal into another compressed video signal with different properties. *Transmoding* is a variation of transcoding, where the modality of the content is changed (e.g., video-to-text). Simulcast, scalable coding and transcoding/transmoding sometimes coexist in order to provide an optimal tradeoff between transcoding overhead and storage requirements [143].

In this chapter, we review different video adaptation strategies. Video adaptation is essential to provide adaptive delivery, where content needs to be prepared so as to matches individual appliance

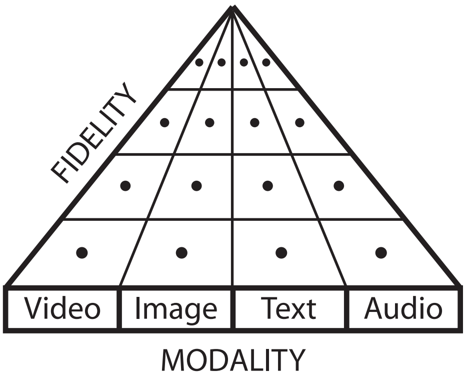


Figure 3.1: The InfoPyramid provides a multimodal, multiresolution representation for content items and for their transcoded versions. (Adapted from [209])

and network resources while providing maximum value for the end user. Simulcast is discussed in Section 3.2. Scalable coding methods are surveyed in Section 3.3. Transcoding and transmoding at last are reviewed in Section 3.4.

3.2 Simulcast

To handle varying channel environments and terminal resources by simulcast, the same video is simply coded several times, each with a different quality and/or resolution setting. The content version or *variation* that best matches a set of given terminal and network characteristics is later selected among these "pre-adapted" versions. This approach is very fast since no transformation of the video is required at the time of delivery. The primary shortcoming however is that massive storage capacity is needed to hold all content variations. Moreover, possible resource profiles must be known *a priori* in order for the corresponding variations to be generated.

To handle content variations, Li et al. [132, 209] propose a progressive data representation scheme called *InfoPyramid* (Figure 3.1). The InfoPyramid manages several variations of media objects with different *modalities* (e.g., video, still image, text and audio) and *fidelities* (summarized, compressed, scaled, etc.). Thus, it provides a multimodal, multiresolution representation for the content items and for their transcoded versions. Mohan et al. [153, 154, 209] as well as Kim et al. [123] further provide a mechanism for content selection. Their mechanism selects the best content variations from the InfoPyramid in order to meet the client resources while delivering the most "value". The solution builds on Shannon's Rate-Distortion (R-D) theory [205]. R-D theory is generalized to a *value-resource* framework by considering different variations of a content item as analogous to different compressions, and different client resources as analogous to the bitrate.

Simulcast is very inefficient, as higher-quality or resolution bitstreams essentially repeat the information that is already contained in the lower-quality or resolution stream, plus some additional information. Scalable video coding eliminates such redundancy by coding multiple fidelity levels into a single stream. The objective is to obtain lower qualities, spatial resolutions and/or temporal resolutions by simply truncating certain layers or bits from the scalable stream.

3.3 Scalable video coding

Basic modes of scalability include *SNR scalability*, *frequency scalability*, *spatial scalability*, and *temporal scalability*. *SNR scalability* provides gradual quality approximations of the original sequence. *Frequency scalability* includes different frequency components in each layer, with the base layer containing low-frequency components and the other layers containing increasingly high-frequency components. *Spatial scalability* is the representation of the same video in varying spatial resolutions. *Temporal scalability* is the representation in varying temporal resolutions or frame rates. Scalable coders can furthermore have *coarse granularity* (two or three layers – such coders are also called *layered coders*), or *fine granularity*. In the extreme case of fine granularity (*embedded coding*), the bitstream can be truncated at any point. *Content-blind scalability* methods perform the same operation over the entire video frame. When scalable content can be accessed at the object level, this is referred to as *object-based scalability*. Next, content-blind and object-based scalability methods are reviewed. Scalable video coding has also been extensively discussed by Wang et al. [247].

3.3.1 Content-blind scalability

Content-blind scalability methods do not take into account the semantic content of video; they perform the same operation over the entire frame. Many methods achieve scalability by first coding a coarse video representation into a *base layer*; the difference between the original frame and the reconstructed base layer is then coded into one or more *enhancement layers*. Alternatively, the temporal frame structure of predictive-coded video or intrinsic properties of waveform-based coding can be taken advantage of to reach scalability.

SNR scalability

SNR or *quality* scalability allows for the decoding of appropriate subsets of a single bitstream to generate gradual quality approximations of the original sequence. Typically, this is accomplished by coarsely quantizing the color values (in the original or in a transformed domain) in the *base layer*. The difference between the base layer and the original frame is then coded into one ore more *enhancement layers*. The block diagram of a simple two-level, SNR-scalable codec is shown in Figure 3.2. The raw video frame (or the motion compensated error frame) is DCT-transformed and quantized at the base level Q_1. The base-level DCT coefficients are then reconstructed by inverse quantization and subtracted from the original DCT coefficients. The residual is quantized by a quantization parameter Q_2 that is smaller than that of the base level. Quantized bits are encoded by Variable Length Coding (VLC). This is the mode of operation that has been chosen by H.263 and MPEG–2 to reach SNR scalability [107, 114].

The Fine Granularity Scalability (FGS) mode of MPEG–4 extends the above method to provide fine granularity [133, 183]. As before, a base-layer stream is produced using a relatively large quantization parameter. Then, for every coded frame, the differences (*refinement coefficients*) between the original DCT coefficients and the quantized coefficients in the base layer are coded into a fine-granularity stream. This is achieved by quantizing the refinement coefficients using a very small quantization parameter, and then by representing the quantized indices through *successive bit plane coding* [106]. Specifically, the absolute values of quantized refinement coefficients in each block are specified in binary representation. Starting from the highest bit plane that contains nonzero bits, each bit plane is successively coded using run-length coding, from the most significant bit to the least significant bit. Adaptive Motion-Compensation FGS (AMC-FGS) and similar techniques further enhance the coding efficiency of MPEG–4 FGS by exploiting temporal redundancies within the video stream [231, 253].

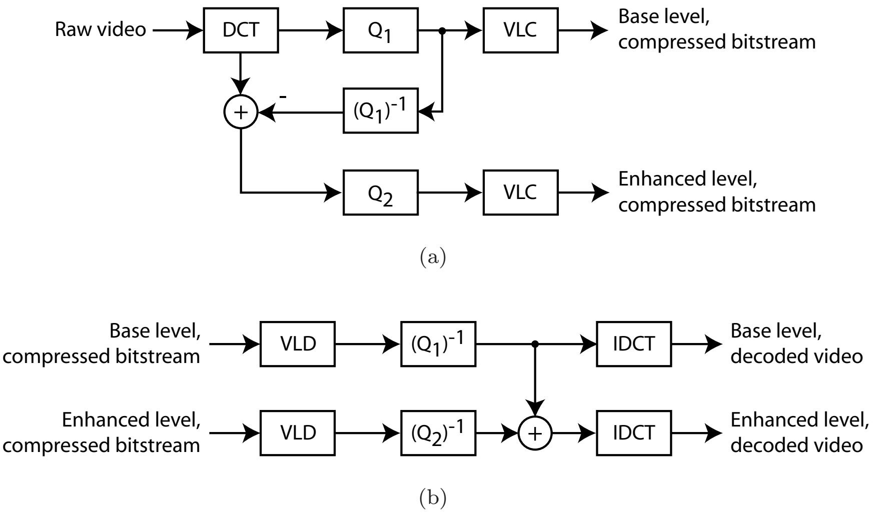

Raw video
DCT
Q_1
VLC
Base level, compressed bitstream
-
$(Q_1)^{-1}$
+
Q_2
VLC
Enhanced level, compressed bitstream
(a)
Base level, compressed bitstream
VLD
$(Q_1)^{-1}$
IDCT
Base level, decoded video
Enhanced level, compressed bitstream
VLD
$(Q_2)^{-1}$
+
IDCT
Enhanced level, decoded video
(b)

Figure 3.2: A two-level, SNR-scalable codec. The base layer is coarsely quantized, and the difference between the base layer and the original frame is coded into an enhancement layer. (a) Encoder. (b) Decoder.

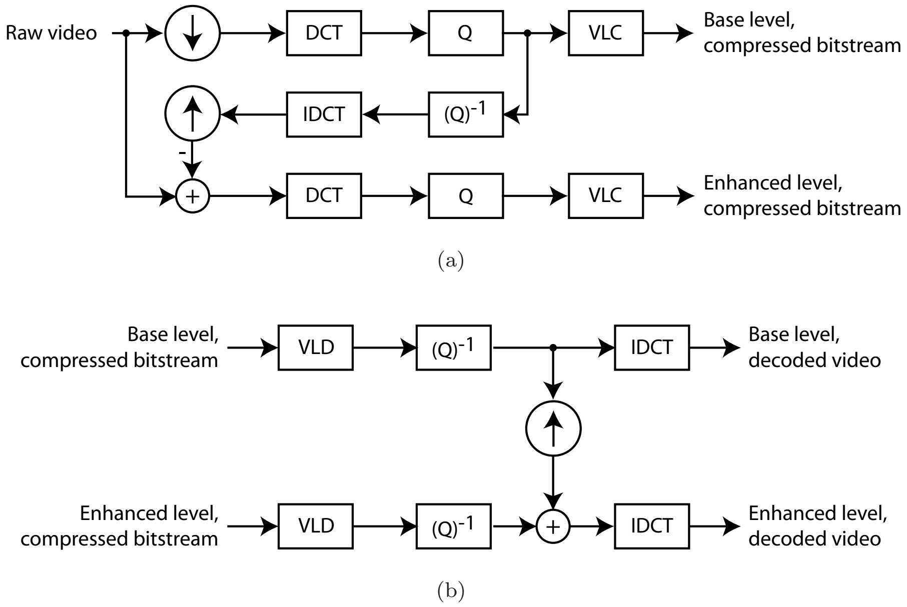
Frequency scalability

Frequency scalability is achieved by including different frequency components in each layer, with the base layer containing low-frequency components, and other layers containing increasingly higher-frequency components. This way, the base layer will provide a blurred version of the video, and the addition of enhancement layers will yield increasingly sharper versions. In the MPEG–2 coding standard, this is known as *data partitioning* [107]. Mode information, motion information, and first few DCT coefficients of each macroblock are included in the base layer. The remaining DCT coefficients are included in the enhancement layer.

Spatial scalability

Spatial scalability permits the transmission of video at different levels of spatial resolution in a single bitstream. First, a multiresolution decomposition of the original video is obtained. The lowest-resolution version is coded directly into the base layer. To produce the second layer, the decoded video from the base layer is interpolated to the second-lowest resolution, and the difference between the original and the interpolated video at that resolution is encoded. The bitstream for each of the following layers is produced in the same way: first, the video is interpolated to that resolution; then, the difference between the estimated and the original video at that resolution is encoded. A two-level spatially scalable codec implementing the above algorithm is shown in Figure 3.3. This method has been adopted to provide spatial scalability in MPEG–2 [107]. Dugad and Ahuja [62] obtain similar results by using two non-scalable video codecs (e.g., MPEG–2 main profile), along with a downsampler and an upsampler. Their motivation is to achieve the functionality of spatial scalability with standard equipment. Wang et al. [245] obtain fine granularity in the enhancement layer by utilizing bit plane techniques in a way that is similar to that used by MPEG–4 FGS to achieve quality scalability.

In contrast to spatial domain techniques, DCT domain techniques remove the unnecessary decompression and recompression procedures, and thus have the advantages of reduced computational



(a)

(b)

Figure 3.3: A two-level spatially/temporally scalable codec. The video at a reduced resolution is coded into the base layer. The difference between the original video and the interpolated base layer is coded into the enhancement layer. (a) Encoder. (b) Decoder.

complexity and storage requirements. In addition, significant processing speedup may be gained because of the lower data rate in the compressed domain. To implement spatial scalability in the DCT domain, DCT coefficients of lower-resolution frames are generated directly from the DCT coefficients of the original frames. This is typically achieved through linear operations on DCT blocks [94, 95].

Temporal scalability

Temporal scalability is defined as the representation of the same video in varying temporal resolutions. The block diagram of a temporally scalable codec is the same as that of a spatially scalable codec (Figure 3.3). The only difference in the temporally scalable codec is the use of temporal downsampling and upsampling instead of spatial downsampling and upsampling. This procedure however can be notably simplified by taking advantage of the temporal frame structure of predictive-coded video (e.g., MPEG–x, H.26x). Temporal scalability from predictively coded video is provided by strategic placement of reference frames, and selective decoding of frames. In [43, 216], the first frame in a group of frames (GoF) is coded independently as a still image. The remaining frames are predicted from the first frame either directly or recursively. Once the reference frame is decoded, any of the other frames within the GoF can be immediately decoded. The result is a temporal subsampling of the original sequence, where the decoded frames are exactly equal to those that would appear in the full frame-rate decoded sequence. Similarly, non-reference frames are dropped to achieve lower temporal resolution with the H.264/AVC video coding standard [249].

Hybrid scalability

Hybrid scalability uses combinations of the basic schemes presented above to reach finer granularity. Spatiotemporal scalable video coding is realized by combining layered coding as shown in Figure 3.3, and selective frame decoding [58]. To support both SNR and temporal scalability through a single enhancement layer, van der Schaar and Radha [229, 230] multiplex the MPEG–4 FGS residual signal (for SNR scalability) and the motion-compensation residual signal obtained by fine-granular temporal scalability together. As compared to multilayer coding, where the SNR and the temporal scalability enhancement layers are separate, this single-layer structure eases tradeoffs between both scalability bandwidth overheads at transmission time.

Wavelet transform-based coding

The discrete wavelet transform provides a multiresolution/multifrequency expression of a signal with localization in both time and frequency [144]. Thus, subband/wavelet transformations have the benefit of naturally providing scalability. Wavelet-based scalable coding techniques for video can be classified into three categories [247]: (1) spatial-domain motion compensation, followed by 2-D wavelet transform; (2) wavelet transform followed by frequency-domain motion compensation; (3) 3-D wavelet transforms, with or without motion estimation. Techniques in the first category remove temporal redundancies by using a motion-compensated temporal wavelet transform. The transform follows the trajectory indicated by motion vectors obtained via spatial domain motion estimation. Thereafter, the frames resulting from temporal filtering are spatially-decomposed using a 2-D wavelet transform in order to reduce spatial redundancies. So-called *in-band predictive schemes* use a classical hybrid video codec architecture, with that exception that motion estimation and compensation take place after the spatial wavelet transform and not before. Thus, the multiresolution nature of the wavelet domain representation can be fully exploited. The above compression technologies have been investigated in the framework of MPEG–4 Scalable Video Coding (SVC) experiments [199]. Karlsson and Vetterli [120] first advocated the use of a separable 3-D discrete wavelet transform (3D-DWT) for video compression. Recent extensions of their solution have lead to video compression that is highly scalable all in the spatial, in the temporal and in the quality domains [200, 201].

3.3.2 Object-based scalability

Instead of performing the same operation over the entire frame, object-based scalability selectively enhances a particular region or object (Section 2.2.2). In object-based temporal scalability (OTS), the frame rate of a selected object is enhanced such that it has smoother motion than the remaining area. The MPEG–4 implementation of OTS, which is notably based on earlier work by Katata et al. [122], provides two types of enhancement structures [108, 252]. *Full temporal enhancement* is used when the texture information for the background object is fully available, whereas *partial temporal enhancement* is used when this is not the case. An example of partial temporal enhancement is shown in Figure 3.4(a). VideoObjectLayer 0 (VOL0) is the sequence of an entire frame with both an object and a background, whereas VOL1 represents a particular object in VOL0. VOL0 is coded at a low frame rate, and VOL1 is coded to achieve a higher frame rate. Figure 3.4(b) shows an example of full temporal enhancement, in which VideoObject 0 (VO0) is the sequence of an entire frame that only contains a background and has no scalability layer. VO1 is the sequence of a particular object. VO1 has two scalability layers, VOL0 and VOL1. Since VOL1 is coded to achieve a higher frame rate than VOL0, VOL0 is regarded as the base layer and VOL1 as an enhancement layer. Note that VO0 is not required to have the same frame rate as other VOs.

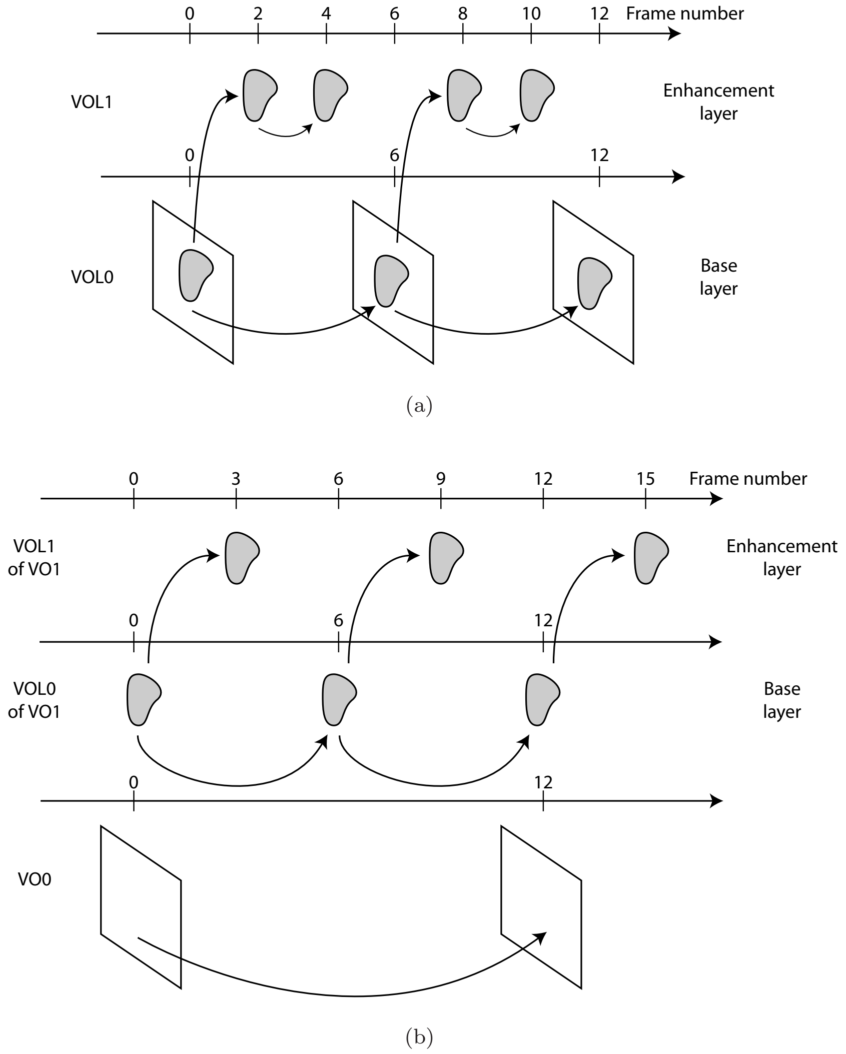


Figure 3.4: Object-based temporal scalability in MPEG–4. (a) Partial temporal enhancement of a video object using P-frames. (b) Full temporal enhancement of VO1. (Adapted from [252])

To reach object-based SNR scalability, van der Schaar and Lin [228] have modified the MPEG–4 FGS coding scheme to perform quality enhancement of selected regions. Specifically, Adaptive Quantization (AQ) is used to control the quantization factor on a macroblock basis. A lower quantization level is used for object macroblocks compared to the background. AQ is achieved through *bit plane shifting* of selected macroblocks within an FGS enhancement layer frame. Bit plane shifting is equivalent to a power-of-two multiplication of macroblock coefficients [56].

With simulcast and scalable coding, potential resource profiles need to be known prior to coding in order to select adequate encoding parameters. With transcoding on the other hand, the input bitstream is converted according to the needs of the connected appliance *on the fly*. Content-blind and content-based transcoding methods are reviewed next.

3.4 Transcoding

Video transcoding is a process for converting an existing compressed video signal into another compressed video signal with different properties. A basic transmission system including a transcoder is illustrated in Figure 3.5. The coded input stream is transformed by the transcoder so as to match the end decoder. Note that front encoding usually happens offline, whereas transcoding and end-decoding are performed in real time at the time of delivery.

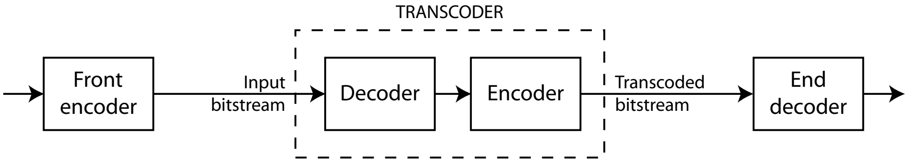


Figure 3.5: Basic transmission system including a transcoder.

The video transcoding literature has been mainly focused on three functionalities: *bitrate reduction*, *spatial resolution reduction*, and *temporal resolution reduction. Reducing the bitrate* is used to meet an available channel or storage capacity. *Spatial resolution reduction* and *temporal resolution reduction* permit content distribution to devices with various display capabilities and processing power. With the recent introduction of packet video services over mobile access networks, *error-resilience video transcoding* has gained a significant amount of attention as well. The aim is to increase the resilience of the original bitstream to transmission errors. Within each one of these functionalities, a further distinction is made between *homogeneous* and *heterogeneous* transcoding. With homogeneous transcoding, source and destination video are in the same compression format, such as MPEG–2 to MPEG–2 transcoding. Heterogeneous transcoding on the other hand transforms video from one compression format to another, such as MPEG–2 to MPEG–4 transcoding. Figure 3.6 illustrates some common transcoding operations. In this example, the original MPEG–2 video is transcoded to (a) a reduced bitrate; (b) a reduced spatial resolution and (c) a different compression format.

The remainder of this section provides a survey of video transcoding techniques. Content-blind transcoding is reviewed in Section 3.4.1. The output format is determined based on network and appliance constraints, independently of the characteristics of the content itself. Recent transcoding techniques consider content features to minimize the degradation of important image regions. These content-based transcoding techniques are reviewed in Section 3.4.2.

3.4.1 Content-blind transcoding

Content-blind solutions to video transcoding determine the output format based on network and appliance constraints, independently of the semantics of the content. Common methods for bitrate reduction, spatial resolution reduction, temporal resolution reduction and error-resilience coding are discussed next. For an in-depth review of content-blind video transcoding, the reader may also refer to the tutorial article on video transcoding architectures and techniques by Vetro et al. [233].

Bitrate reduction

The objective of bitrate reduction is to convert a compressed bitstream into lower rates without modifying its original structure. Ideally, the quality of the transcoded stream should have the quality of a bitstream directly generated with the reduced rate. The most straightforward way to achieve

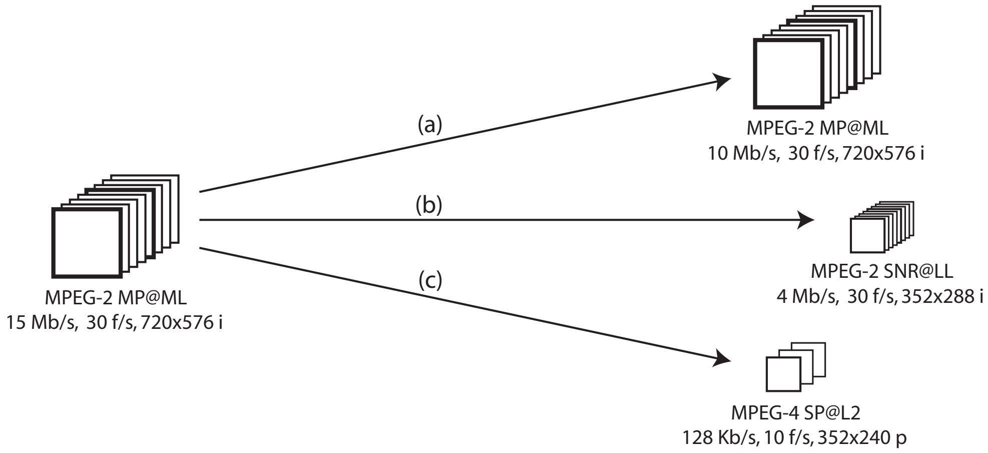

(a)
(b)
(c)
MPEG-2 MP@ML
15 Mb/s, 30 f/s, 720x576 i
MPEG-2 MP@ML
10 Mb/s, 30 f/s, 720x576 i
MPEG-2 SNR@LL
4 Mb/s, 30 f/s, 352x288 i
MPEG-4 SP@L2
128 Kb/s, 10 f/s, 352x240 p

Figure 3.6: Illustration of some common video transcoding operations. The original video is encoded in an MPEG–2 format (Main Profile at Main Level = MP@ML) at 15 Mb/s. The temporal rate is 30 frames/s, and the input resolution is 720x576 i (interlaced). (a) The original video is transcoded to a reduced bitrate of 10 Mb/s. (b) The original video is transcoded to the MPEG–2 SNR Profile at Low Level, at 4 Mb/s. The temporal rate is 30 frames/s, and the output resolution is 352x288 i. (c) The original video is transcoded to the MPEG–4 Simple Profile at Level 2, at 128 Kb/s. The temporal rate is 10 frames/s, and the output resolution is 352x240 p (progressive).

bitrate reduction is to decode the video bitstream and fully re-encode the reconstructed signal at the new rate. This is illustrated in Figure 3.7. Considering that the decoder decodes the incoming bitstream down to the pixel domain, and that the encoder re-encodes the video to a different rate, this structure is called *cascaded pixel-domain transcoder* (CPDT). The main advantage of the CPDT is its flexibility. Since transcoding is carried out in the pixel domain, decoded video characteristics, such as spatial size or color, can be modified, and extra information, such as digital watermarks, can be inserted into the video stream. Moreover, the CPDT can achieve drift-free transcoding*. However, this solution has high computational and memory demands.

By reusing information from the incoming bitstream, decoder and encoder contained in a transcoder can be significantly simplified to reduce the corresponding complexity, while still maintaining acceptable quality [130, 233]. Two examples of simplified architecture are shown in Figure 3.8. In the open-loop system shown in Figure 3.8(a), the input video bitstream is first partially decoded to the DCT coefficient level. The bitrate is then scaled down by re-quantizing all coefficients with a larger quantization step-size Q_2 [160]. This open-loop transcoder has very low complexity, as it does not require entire decoding and encoding loops, nor frame store memories for reconstructed pictures. However, open-loop architectures are subject to drift caused by re-quantization error. The closed-loop system in Figure 3.8(b) aims to eliminate the mismatch between predictive and residual components by using drift compensation. The re-quantization error is stored in a frame store and is fed back to the re-quantizer Q_2 to correct the re-quantization error introduced in the previous frame. The main difference between the CPDT and simplified closed-loop architectures is that reconstruction in the CPDT is requiring two reconstruction loops with one DCT and two IDCT, whereas one single reconstruction loop with one IDCT is used in the closed-loop system. Decoding

*Drift can be explained as the blurring or smoothing of successively predicted frames. It is caused by the loss of high frequency information, which creates a mismatch between the actual reference frame used for prediction in the encoder and the degraded reference frame used for prediction in the transcoder and decoder. As time goes on, this error propagates, resulting in severely degraded reconstructed frames [233, 260, 261].

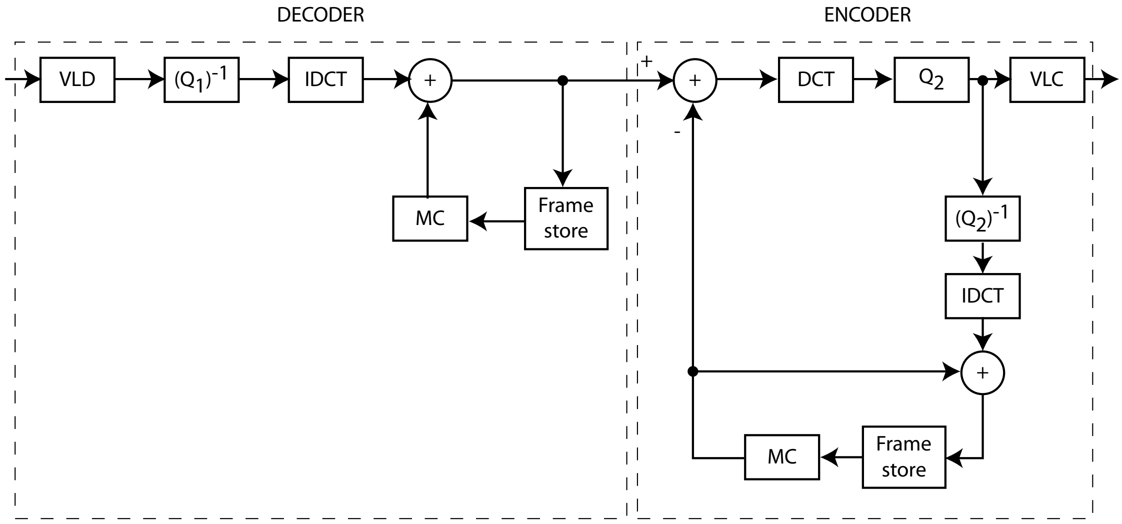


Figure 3.7: Cascaded pixel-domain transcoding architecture for bitrate reduction.

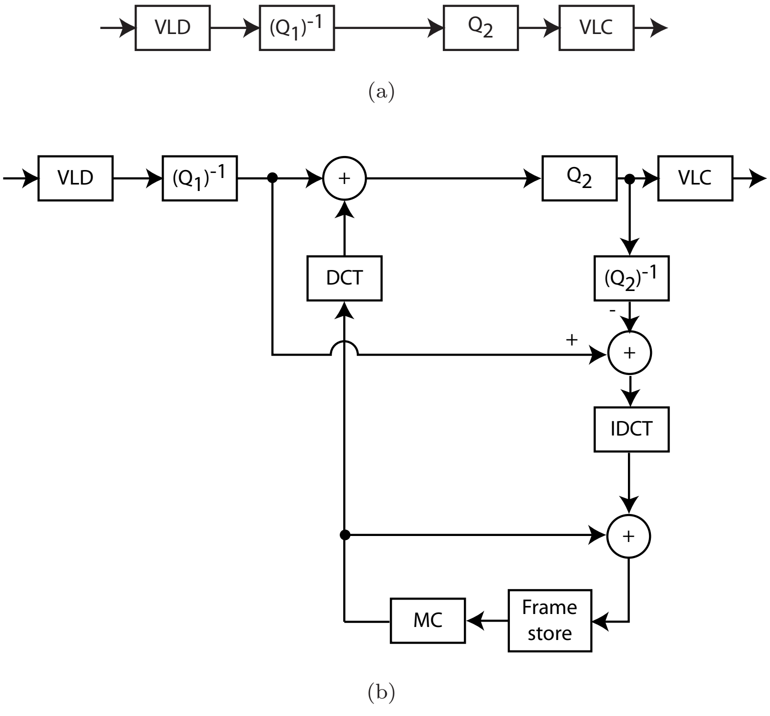


Figure 3.8: Simplified architectures for bitrate reduction. (a) Open-loop, partial decoding to DCT coefficients, then re-quantization. (b) Closed-loop, drift compensation for requantized data. (Adapted from [233]).

to the pixel-domain can be avoided altogether by performing motion compensation in the frequency domain as well. This is equivalent to removing the DCT and the IDCT from Figure 3.8(b). By performing bitrate reduction entirely in the frequency domain, Assunção and Ghanbari [7] have achieved computational complexity savings of up to 81% as compared to the CPDT.

Spatial resolution reduction

The emergence of mobile multimedia devices with limited display capabilities and the increasing demand for rich content consumption on such devices have created a strong need for efficient ways to reduce the spatial resolution of video. As for bitrate reduction, the cascaded pixel-domain transcoder fully decodes the input stream, and re-encode the reconstructed video at lower spatial resolution. Again, significant complexity savings with minimal loss of quality can be achieved by reusing information from the incoming bit-stream. For simplicity, the following discussion is limited to 2:1 spatial resolution reduction, e.g., CIF to QCIF. Rational downsizing video transcoding is discussed in [38, 218].

Motion vector (MV) reestimation is avoided by mapping incoming MVs to the lower spatial resolution [14, 202]. Usually, one new MV is calculated from an input set of four MVs, corresponding to four 16x16 macroblocks (MB). This is referred to as 4:1 mapping. Commonly, 4:1 mapping is achieved by applying a weighted average or median filters to the input vectors. The output vector's amplitude is scaled by two in order to account for the lower spatial resolution. Certain compression standards, such as MPEG–4 Visual [108] and H.263 [114], support advanced prediction modes that allow one MV per 8x8 luminance block. In that case, each input vector is mapped to one output vector with appropriate scaling by two. This is referred to as 1:1 mapping. Both motion vector mapping strategies are illustrated in Figure 3.9. 1:1 mapping provides a more accurate representation of the motion, but also uses more bits in the transcoded stream, since four MVs must be encoded per MB. Thus, an optimal strategy would adaptively select the best mapping based on a rate-distortion criterion.

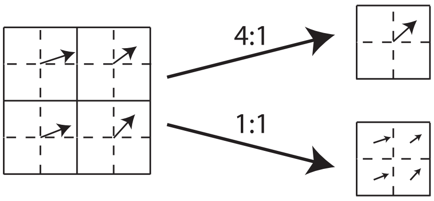


Figure 3.9: Motion vector mapping for spatial resolution reduction. One new motion vector (4:1 mapping) or four new motion vectors (1:1 mapping) are calculated from an input set of four motion vectors.

Simply reusing motion vectors computed at the original bitrate in the reduced-rate bitstream leads to non-optimal results due to the mismatch between prediction and residual components [258]. This loss of quality can be overcome without full-scale motion estimation by using motion vector refinement (MVR). The best-matching MV is searched in a small window around the point indicated by the base motion vector obtained from motion vector mapping. Typically, a search range of ±2 pixels instead of ±15 pixels or larger is used. This keeps the added complexity low while eliminating significant amount of noise in the transcoded signal. Complexity can be kept even lower by performing MVR entirely in the frequency domain [55].

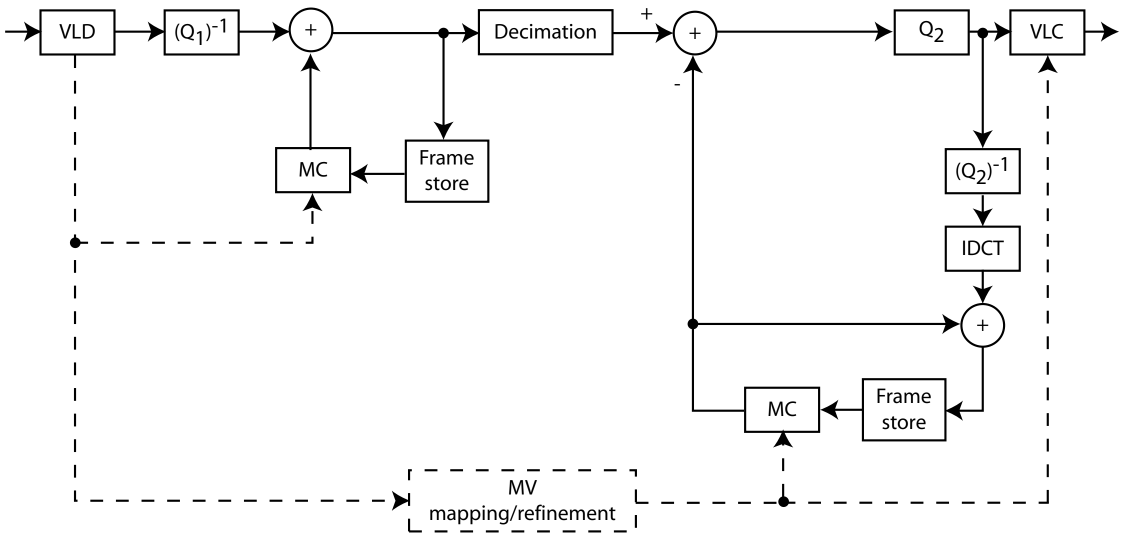


Figure 3.10: Simplified architecture for spatial resolution reduction in the frequency domain (adapted from [203, 204]).

In addition to the mapping of MB-level motion information to the lower resolution, the spatial dimension of the frame must be reduced. The easiest way to achieve this in the pixel domain is by representing every 2x2 pixels by a single pixel of their averaged value. Better results are obtained by DCT decimation [202]. The decimation is realized by retaining the 4x4 low-frequency coefficients of each DCT-transformed 8x8 image block. These coefficients are then inverse-transformed to reconstruct 4x4 pixels. Hence, four blocks become a new 8x8 pixel block. The attractive feature of this method is that most energy of the original frame is conserved as it is concentrated at low frequencies. Both pixel averaging and DCT decimation have been extended to the frequency domain [150, 204]. An example of architecture to perform spatial resolution reduction entirely in the frequency domain is shown in Figure 3.10. This architecture has very low complexity.

Temporal resolution reduction

Temporal resolution reduction modifies the frame rate in order to allow the distribution of content to devices with limited processing power, or so as to maintain higher quality of coded frames. Similar to motion vector mapping for spatial resolution reduction, full-scale motion vector re-estimation is avoided by motion vector composition [258]. As illustrated in Figure 3.11, a new motion vector is estimated to predict the current frame from the latest non-dropped frame. One possible way to generate such a motion vector is to use the vector sum of all intermediate motion vectors. In practice however, the best-matching macroblocks are not located on a MB-boundary, thus intermediate MVs might not be available. A way to estimate such intermediate vectors is to use the bilinear interpolation of motion vectors in the previous dropped frame, where the weighting of each vector is proportional to the amount of overlap with the predicted block. More accurate results can be obtained by a dominant vector selection scheme, where the motion vector associated with the largest overlapping region is chosen [39, 259]. Motion vector refinement is usually applied to the so-composed motion vector to improve the quality of the transcoded signal.

In addition to new motion vectors, new residuals must be estimated for the lower temporal resolution sequence. With pixel-domain architectures, the residual between the current frame and the latest non-dropped frame can be easily computed given the new motion vector estimates. Fung

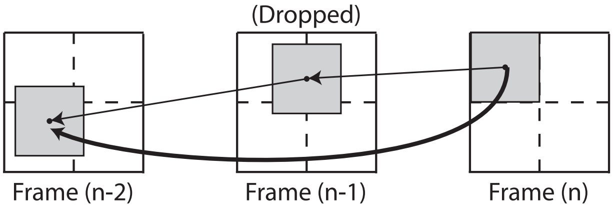


Figure 3.11: Motion vector composition. Since frame (n-1) is dropped, a new motion vector to predict frame (n) from frame (n-2) is estimated.

et al. [76] compute the new residual directly in the frequency domain. They achieve this by using a direct addition of the DCT coefficients for macroblocks without motion compensation, and a feedback loop for error compensation within motion-compensated MBs.

Error-resilience transcoding

The steadily increasing demand for multimedia content delivery over error-prone wireless transmission channels has motivated research on a new category of error-resilience video transcoders. The operation of the transcoder is shown in Figure 3.12. The incoming bitstream is decoded to the degree required to add resilience. Temporal and/or spatial resilience is then added by the transcoder. As resilience is improved at the cost of an increase in the overall bitrate, rate reduction techniques are used to recover the original rate. Finally, the resilient bitstream is requantized and variable-length re-encoded.

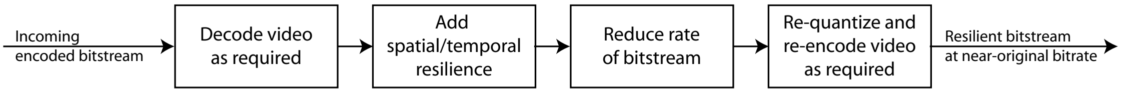


Figure 3.12: Error-resilience transcoding operation.

De los Reyes et al. and Dogan et al. [54, 57] consider two basic techniques of resilience: spatial localization and temporal localization. Spatial localization prevents errors caused by a bit error to propagate within a video frame. One approach to combat this effect is to shorten the length of a bitstream slice by adding additional synchronization headers. Another possible method is to limit spatial error propagation by limiting the reliance of motion compensation. If predictions are only made within the spatial extent of the current slice, errors will not propagate outside that slice. Temporal localization prevents the propagation of errors to subsequent frames. A common technique is to add intra-frames or intra-blocks to the original stream. These frames or blocks are used as references for subsequent temporal prediction, and thus reduce the duration of error propagation.

3.4.2 Content-based transcoding

Recent content-based transcoding techniques make use of content characteristics in order to minimize the degradation of important regions or to change the media nature of the input signal. *Intramedia* transcoding reduces the bitrate, spatial resolution or temporal resolution of the video. Unlike content-blind transcoding however, a different set of transcoding parameters is used for each class of relevance. For instance, foreground objects might be encoded with higher quality than the background. *Intermedia transcoding*, or *transmoding*, transforms the video to another modality, such

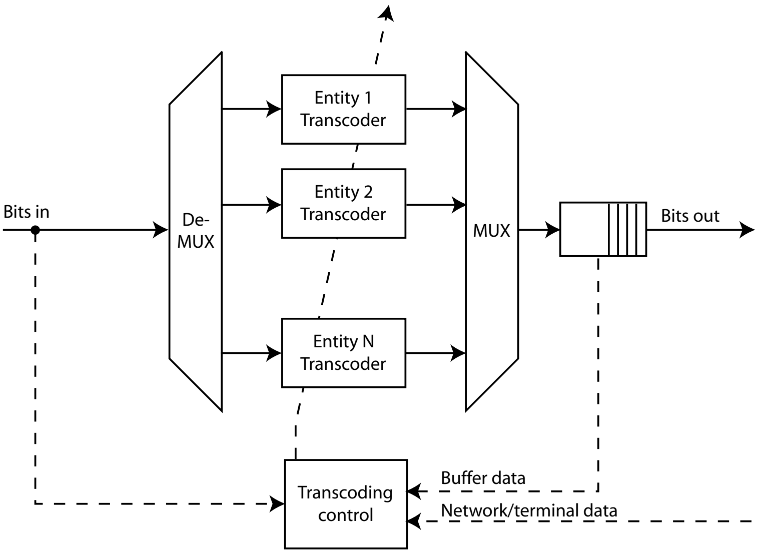


Figure 3.13: General configuration of a content-based intramedia transcoder. The input stream is split into a set of non-overlapping entities, such as video objects or shots. Each entity is transcoded using a different set of parameters. The individual transcoded bitstreams are recombined and sent through the network after buffering. (Adapted from [237])

as text or audio. This involves an additional analysis operation [10] in order to translate low-level information, such as shape, color and motion, into a high-level description of the video.

Intramedia transcoding

Similar to content-blind transcoding, content-based intramedia transcoding deals with the reduction of bitrate, spatial resolution, and temporal resolution. However, the transcoding process is controlled by content characteristics. The general configuration of a content-based intramedia transcoder is depicted in Figure 3.13. The input stream is split into a set of non-overlapping entities, such as video objects or shots. Each entity is transcoded using a different set of parameters. The individual transcoded bitstreams are recombined and sent through the network after buffering. Transcoding parameters are controlled by content features, buffer data, network conditions and terminal resources.

One class of content-based methods relies on global frame characteristics to control the transcoding process. Liang and Tan [135] establish a set of rules to select the transcoding operation to be applied to MPEG video frames (i.e., requantization, temporal resolution reduction, or spatial resolution reduction) based on motion activity and spatial activity. Motion activity is inferred by the average magnitude of the motion vectors of all the intercoded macroblocks in each frame. Spatial activity reflects the number of spatial details and is measured by the mean quantization scale of each frame. A wider set of frame-based characteristics of MPEG video is used by Huang et al. [98]. The proposed characteristics are region perceptibility, spatial complexity and temporal similarity. Region perceptibility represents the visual importance of each region within a picture and can be measured by the quantization scale of each macroblock. Spatial complexity indicates which region is complex (and thus needs more bits to be encoded) and is given by the percentage of zero-quantized DCT coefficients. Temporal similarity describes the temporal relationships of a video sequence, as given by coding types of each MB and motion vector information. Lei and Georganas [131] use scene change location information to control frame skipping and allocate the frame bit budget in MPEG–x and H.26x transcoding. They detect scene changes by considering the percentage of intracoded macroblocks in each frame. Whenever that percentage exceeds 40%, a scene change is detected. The algorithm then uses intracoding for all frames that follow a scene change.

Other methods consider the usage of semantic video objects as transcoding entities. This approach is called *object-based transcoding.* Vetro et al. [234, 236, 237] determine optimal quantization parameters and frame skip for each video object individually. The bitrate budget for each object is allocated by a difficulty hint, a weight indicating the relative encoding complexity of each object. Frame skip is controlled by a shape hint, which measures the difference between two consecutive shapes to determine whether an object can be temporally downsampled without visible composition problems. Key objects are selected based on motion activity and on bit complexity. A similar approach is followed by Cucchiara et al. [11, 46, 47, 48]. They subdivide the frame in a number of *classes of relevance.* For instance, one class of relevance might contain all foreground objects, whereas the background is allocated to a second class of relevance. Each class of relevance is furthermore given a *relevance measure* (weight) by the user. The transcoding system can then be programmed differently for each class of relevance. To measure transcoding performance, the authors introduce the Weighted PSNR metric (WPSNR), which is a PSNR measure accounting for the different classes of relevance present in the scene. The goal is to achieve higher quality on more relevant classes. Kim and Choi [124] also achieve different transcoding quality for visually important and unimportant regions. They reduce bandwidth requirements for non-important DCT blocks by removing high-frequency components through lowpass-filtering. To determine block importance directly in the DCT domain, the authors use their *discontinuity height* measure [125], which gives the contrast of dominant discontinuities within the block. Highly contrasted discontinuities are considered to be visually important.

Transmoding

The aim of transmoding is to change the modality of input content, while preserving core information (*main content message*). Examples of transmoding include speech-to-text (speech recognition) [156, 179] and text-to-speech (speech synthesis) [24, 77] translation. As for video, the most common transformations are video-to-text, video-to-still images, and audio-to-video.

Video-to-text transmoding is used for automatic annotation, indexing, and structuring of video. This is achieved through content extraction, followed by textual content representation. Content extraction is the identification of semantic entities, typically shots and objects, within the video. Content representation refers to the textual retranscription of such semantics. Harit et al. [86] use change detection and EigenTracking [17] to segment and track moving video objects. The objects are then grouped together in a semantically meaningful manner to form an appearance hierarchy. The final video representation consists of the appearance spaces of objects, along with their projection coefficients and affine location parameters in each frame in which they exist. This representation is stored in eXtensible Markup Language XML [241]. In addition to the above, Nagao et al. [158] use automatic scene detection to describe video clips using the Synchronized Multimedia Integration Language SMIL [240]. Another possible way to transform video into text is *text information extraction.* This involves detection, localization, tracking, extraction, enhancement and recognition of the text from a given video [118]. Such text data often provides valuable indications about video contents and structure. For instance, character extraction of license plates permits automatic vehicle identification for traffic surveillance [50].

Video summarization involves the transformation of input video into a series of representative still images. This is achieved by temporal segmentation or key frame extraction, followed by summary representation. Temporal segmentation separates the video into shots based on color features [265], edges [264], or feature point tracking [21]. Key frames are obtained either by simply selecting the first frame in each shot [159], or by detecting camera stops using motion information [251], for instance. Summary representations include sequential, hierarchical, pictorial and mosaic-based summaries. The sequential summary is simply a concatenation of the key frames, which are shown sequentially in time. In a hierarchical summary, key frames are grouped and organized so as to obtain a coarse-to-fine hierarchy of summaries (e.g., tree) [268]. Pictorial summaries show all key frames in a singe image. The frames are resized so as to reflect their importance in the video [257]. In a mosaic-based summary finally, multiple shot frames are aligned and integrated to construct a mosaic of the scene [102].

Another possible transmoding operation is the mapping from speech to talking faces. This can be used to improve the understanding of spoken text, and to animate virtual characters. Nakamura [161] estimates facial parameters from audio input using a HMM-based method. These parameters control deformations of a face model.

3.5 Summary

In this chapter, simulcast, scalable coding and transcoding/transmoding methods for video adaptation have been reviewed. Object-based methods that account for different image areas have been presented along with traditional frame-based methods. In Chapter 5 and Appendix A, content-based transcoding is used to provide adaptive video delivery. Relevant areas are extracted by means of semantic video analysis and encoded at a higher level of quality than the background. Transcoding is particularly useful for adaptive delivery, since no *a priori* knowledge about the resource profiles of the connected network and terminal needs to be available.

Part II

Adaptive video delivery using semantics

Semantic video analysis 4

4.1 Introduction

In order to extract *semantics* from unstructured data, semantic analysis is used. Semantics represent a meaningful entity in the input data. In the digital video domain, this is called a *semantic video object* SVO (Section 2.2.2). Object-based representations of multimedia content provide the user with flexibility in content-based access and manipulation. However, it is difficult to extract an SVO, because: 1) a unique definition of an SVO does not exist. Anything that represents a meaningful entity in the real world – for instance a ball, an aircraft, a building or a human body – could be classified as an SVO; 2) SVO extraction is basically a segmentation process, which is still considered one of the most difficult problems by the image analysis community; 3) traditional low-level visual homogeneity criteria (e.g., intensity, color, texture) do not lead to regions that immediately correspond to meaningful objects in the real world. Therefore, more sophisticated homogeneity criteria must be employed.

In the following, we discuss possible homogeneity criteria for the extraction of semantics and, in particular, we describe the use of motion. In Section 4.2, different semantics and the corresponding homogeneity criteria are reviewed in the general context. Motion information is then used to extract moving objects from video in Section 4.3. A statistical change detection process produces the segmentation of moving objects from the background. The selected approach operates in cluttered environment and is robust with regard to camera noise. A multilevel, temporal tracking strategy further enables us to distinguish different video objects when they have similar motion or in the presence of mutual occlusion. The tracking mechanism is based on feedbacks between an object partition and a region partition. These interactions allow us to cope with multiple simultaneous objects, motion of non-rigid objects, occlusions, and appearance and disappearance of objects. The reliable extraction of moving objects is an important aspect of several applications, such as sport broadcasting and visual surveillance. In Chapter 5, this will be used to prioritize visual data in order to improve adaptive video delivery.

4.2 Semantics

Although humans can identify meaningful information effortlessly, the automatic extraction of semantics still remains one of the fundamental research problems in the signal processing community [82, 139]. Semantic visual information extraction is difficult for various reasons: the definition of "semantics" is vague and task-dependent, limited mechanisms are available for extraction (Section 2.3), and there is a problem with background noise sensitivity.

Basically, semantic visual information extraction is a segmentation process. The goal is to separate a meaningful entity from the remaining parts in the visual domain. In any segmentation algorithm, the definition of homogeneity is a critical factor, and different definitions of homogeneity could lead to totally different segmentation results for the same input data. Also, a "good" homogeneity criterion always depends on the task at hand.

Geometry, color and motion rank among the most commonly employed homogeneity criteria. Geometry is used when the shape of the objects to be segmented is known *a priori*. In this case, which notably includes the detection of captions and text, template matching methods search for specific object features in terms of geometry [81]. For segmenting faces of people, regions-based segmentation methods using a homogeneous color criterion can be employed. The face detection task consists in finding the pixels whose spectral characteristics lie in a specific region in the chromaticity diagram [92]. To extract moving objects, motion information can be used. Several applications, such as sport broadcasting and visual surveillance, deal with segmenting moving objects. A typical tool used to tackle this problem is change detection (Section 2.3.1).

In general, a semantic video object may contain multiple shapes, colors and motions. Therefore, a single homogeneity criterion can only deal with a limited set of scenarios. This problem can be overcome by combining multiple criteria. For instance, Mech and Wollborn [149] improve the results of change detection by imposing spatial segmentation on each frame. Ueda and Mase [225] use an energy formulation to detect shapes via user-selected points (*supervised segmentation*). Wang [243] performs spatial segmentation for the initial frame, and temporal tracking for the successive frames. The extraction method we propose next relies on a combination of change detection and temporal tracking to extract moving objects. Temporal tracking allows us to distinguish multiple objects even when they have similar motion or in the presence of mutual occlusions.

4.3 Motion-based semantic video object extraction

Motion is an important cue to produce semantic video objects, since an SVO often has different motion features from the background. In this section, we propose an algorithm that relies on motion information and on temporal tracking to extract objects from video in cluttered environment. The algorithm is based on work that has been previously published by the author et al. in [31, 34]. First, a change detection process produces the segmentation of moving objects from the background. Then, temporal tracking is used to follow individual objects along the frames. The tracking mechanism is based on feedbacks between an object partition and a region partition. These interactions allow us to cope with multiple simultaneous objects, motion of non-rigid objects, occlusions, and appearance and disappearance of objects. The output is a set of video objects that are coherently labeled over time.

The block diagram of the proposed algorithm is depicted in Figure 4.1. The *object segmentation* module receives the video input and produces the object partition that identifies moving objects. In our implementation, change detection is used to this end. In the *region segmentation* step, each object is further decomposed into a set of non-overlapping, homogeneous regions. These are detected

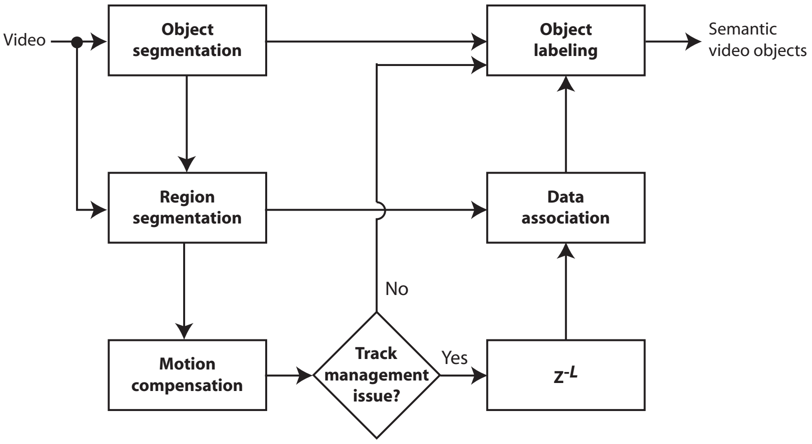


Figure 4.1: Block diagram of the proposed semantic video object extraction algorithm.

using a multi-feature clustering approach, and every region is represented by a region descriptor. The descriptor summarizes the value of the features in the corresponding region. Next, the tracking mechanism operates at the region level. The future position of regions as well as the value of the corresponding region descriptors are predicted by *motion compensation*. This step defines a tentative correspondence between the object partition in the current frame, n, and the object partition in the new frame, $n+1$. The correspondence helps to anticipate track management issues, and it provides an effective initialization for the clustering procedure of each object in the new frame. Whenever a track management issue is detected, *data association* is employed to validate the track of each region descriptor. At last, the tracks of objects are updated as a consequence of region tracking in the *object labeling* stage.

4.3.1 Object segmentation

Moving objects are segmented from the background by change detection. Different change detection techniques can be employed for moving camera and for static camera conditions (Section 2.3.1). If the camera moves, change detection aims at recognizing coherent and incoherent moving areas. The former correspond to background areas, the latter to video objects. If the camera is static, the goal of change detection is to segment moving objects (foreground) from the static background.

The video object segmentation we use [29] addresses the static camera problem and is applicable in the case of a moving camera after global motion compensation [37, 61, 172, 266]. The change detector decides whether in each pixel position, the foreground signal corresponding to an object is present. This decision is taken by thresholding the frame difference between the current frame and a frame representing the background (*background model*). The background model is dynamically generated based on temporal information [30]. The thresholding aims at discarding the effect of camera noise after frame differencing. A locally adaptive threshold, $\tau(i,j)$, is used that models the noise statistics and applies a significance test. To this end, we want to determine the probability that the frame difference at a given position (i,j) is due to noise, and not to other causes. Let us suppose that there is no moving object in the frame difference. We refer to this hypothesis as the *null hypothesis*, H_0. Let $g(i,j)$ be the sum of the absolute values of the frame difference in an observation window $W_{(i,j)}$ of q pixels around (i,j). Moreover, let us assume that the camera noise is additive and follows a Gaussian distribution with variance σ. Given H_0, the conditional probability density function (pdf) of the frame difference follows a χ^2_q distribution with q degrees of freedom

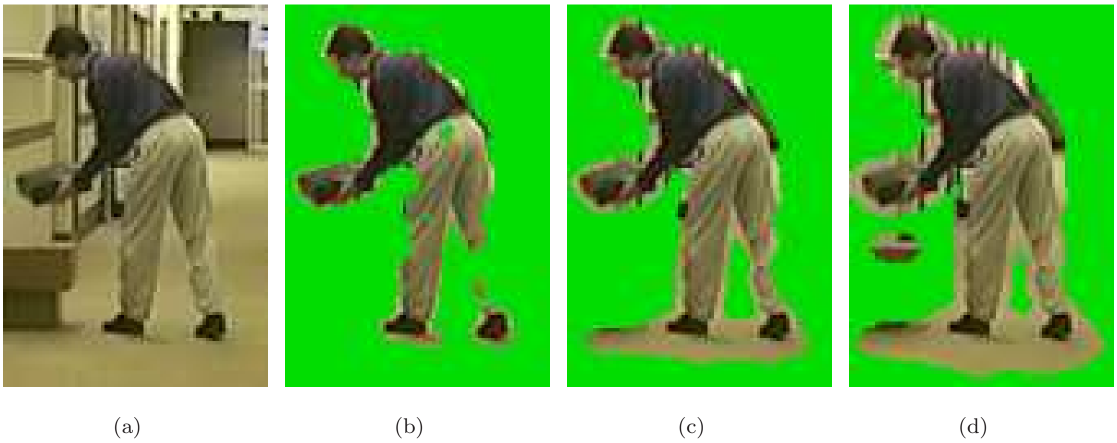


Figure 4.2: Change detection masks for different observation window sizes, q. (a) Details from the sequence *Hall monitor.* (b) $q = 9$. (c) $q = 25$. (d) $q = 49$.

defined by

$$f\big(g(i,j)|H_0\big) = \frac{1}{2^{q/2}\sigma^q\Gamma(q/2)} g(i,j)^{(q-2)/2} e^{-g(i,j)^2/2\sigma^2}, \tag{4.1}$$

where $\Gamma(\cdot)$ is the Gamma function, that can be evaluated as $\Gamma(x+1) = x\Gamma(x)$, and $\Gamma(x/2) = \sqrt{\pi}$. It is now possible to derive the significance test as

$$P\big\{g(i,j) \geqslant \tau(i,j)|H_0\big\} = \frac{\Gamma\big(q/2, g(i,j)^2/2\sigma^2\big)}{\Gamma(q/2)}. \tag{4.2}$$

When this probability is smaller than a certain significance level, α, we consider that H_0 is not satisfied at the pixel position (i,j). Therefore we label that pixel as belonging to a moving object. Otherwise, we label the pixel as belonging to the background. The significance level α is a stable parameter that does not need manual tuning along a sequence or for different sequences. Experimental results indicate that valid values fall in the range from 10^{-2} to 10^{-6}.

The variable q in Equation (4.2) represents the number of pixels in the observation window $W_{(i,j)}$, and thus the number of locations on which the statistics are computed. The effects of different values of q on change detection are illustrated in Figure 4.2. Increasing q makes the statistics more reliable, as it reduces the sensitivity to noise. However, the probability that the hypothesis H_0 remains valid for all the pixels in $W_{(i,j)}$ decreases as well. This leads to a wrong labeling along the edges of moving objects and to the corresponding *halo* effect. To obtain a good tradeoff between robustness to noise and accuracy in the detection, we choose $q = 25$ (5×5 window centered in (i,j)).

Video object segmentation produces the *object partition* Π_o^n at frame n. The object partition identifies the objects from the background and provides a mask defining the areas of the image containing the moving objects. Since the result of change detection is the classification of the pixels into two classes, namely foreground and background, no information is provided about different objects in the scene. For this reason, further processing is required to track the video objects (Figure 4.3).

4.3.2 Region segmentation

Each object in Π_o^n is processed separately and is decomposed into a set of non-overlapping regions to produce the *region partition* Π_r^n. Homogeneous regions are detected using a multi-feature clustering

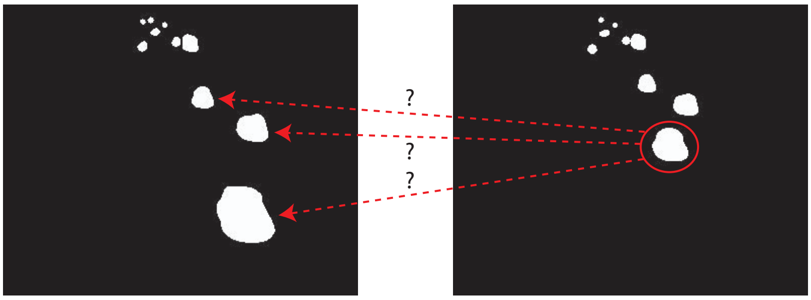


Figure 4.3: Example of object partition in two successive frames. The tracking algorithm is responsible for solving the correspondence problem between two temporal instances of the same object.

algorithm. The selected clustering method [27, 28] is based on spatially unconstrained fuzzy C-means (FCM) [13], which can be considered as a fuzzy generalization of the hard C-means algorithm. The feature space is composed of both spatial and temporal features. Spatial features are absolute position values x and y, color components from the perceptually uniform color space CIE *Lab* [247], and a measure of local texturedness based on variance [36]. The temporal features are the displacement vectors from the optical flow, computed via block matching.

Let $\mathbf{f}_k = \left(f_{k1}, \ldots, f_{kF}\right)$ represent the feature vector corresponding to the k^{th} pixel, where f_{kj} is the value of the j^{th} feature at pixel k, and F is the number of features. Given the feature space $V = \left\{v_1, v_2, \ldots, v_N\right\}$, which represents our data set, and the desired number of classes c, $2 \leqslant c \leqslant N$, the fuzzy partition U of the data set containing N elements is defined by

$$U \mid \quad u_{ik} \in [0,1]\ \forall i,k; \qquad \sum_{i=1}^{c} u_{ik} = 1\ \forall k; \qquad 0 < \sum_{k=1}^{N} u_{ik} < N\ \forall i, \tag{4.3}$$

where u_{ik} represents the degree of belongingness of feature vector $\mathbf{f}_k$ to the class i. The clustering algorithm aims at evaluating the partition that minimizes the functional

$$J_{\text{FCM}}(U, \mathbf{v}) = \sum_{k=1}^{N} \sum_{i=1}^{c} u_{ik}^{m} \big(\mathcal{D}(\mathbf{v}_i, \mathbf{f}_k)\big)^2. \tag{4.4}$$

$\mathbf{v} = \left(\mathbf{v}_1, \ldots, \mathbf{v}_c\right)$ is the vector of the centroids corresponding to each of the classes, and $m \in [1, \infty)$ is a weighting exponent that controls the amount of fuzziness. The similarity $\mathcal{D}(\mathbf{v}_i, \mathbf{f}_k)$ between the ith centroid, $\mathbf{v}_i$, and the feature vector corresponding to the kth pixel, $\mathbf{f}_k$, is measured by the weighted Mahalanobis distance:

$$\mathcal{D}(\mathbf{v}_i, \mathbf{f}_k) = \sqrt{\sum_{j=0}^{F-1} w_{kj} \frac{(f_{kj} - v_{ij})^2}{\sigma_j^2}}. \tag{4.5}$$

To account for different ranges of variation, feature values are normalized with respect to their standard deviation over the entire image, σ_j. The importance of individual features in the clustering process is accounted for by weighting. w_{kj} represents the relative weight of the jth feature at pixel k. Castagno et al. [28] have obtained good results by giving the position information a constant weight of 10%, and the texture information a constant weight of 5%. The motion and the color information adaptively share the remaining 85% according to their reliability. Also, the fuzziness amount has been set to $m = 1.3$.

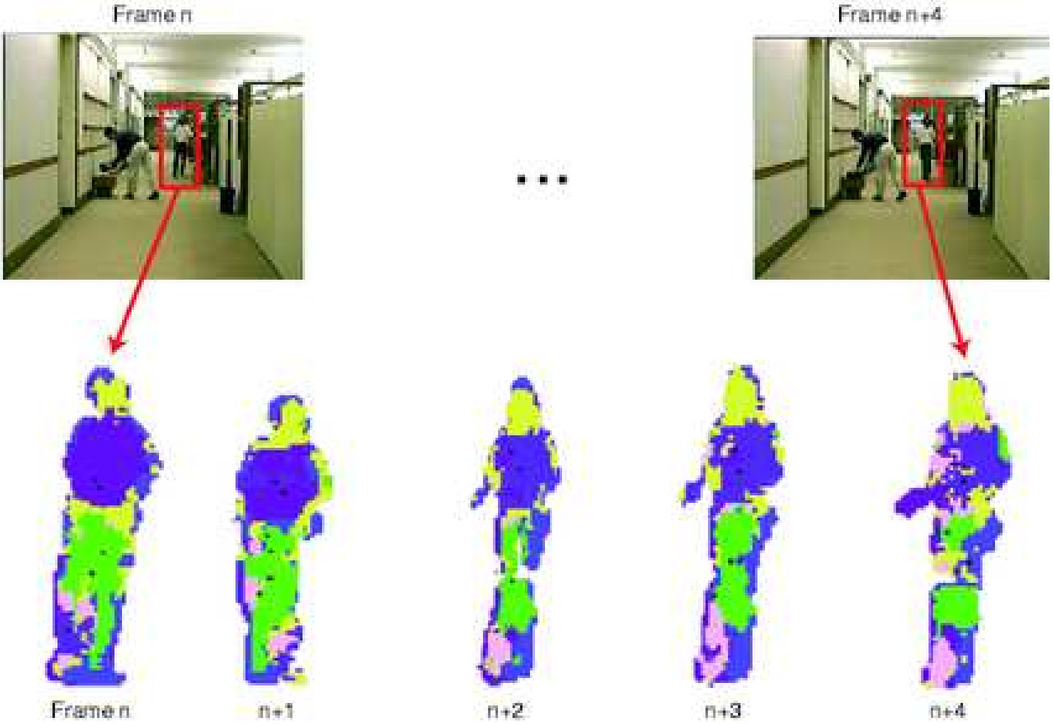


Figure 4.4: Example of region segmentation using multi-feature clustering. Homogeneous regions are computed in each object based on position, color, texture and motion information.

The fuzzy C-means algorithm iterates, evaluating at each step new centroids and a new fuzzy partition, until stability is reached. Once the iterations have stabilized, the fuzzy partition is hardened. Each pixel is assigned to the class for which it shows the highest degree of belongingness. Eventually, the motion-compensated segmentation results obtained for the current frame n will be used as initialization for the FCM procedure at the new frame $n + 1$. This allows us to start the initialization from a point which is likely to be close to a minimum. Thus, the segmentation results show good temporal coherence with the solution obtained from the previous frames, as illustrated in Figure 4.4.

4.3.3 Region tracking

Instead of tracking an entire object, our method relies on region tracking to achieve the correspondence of video objects in successive frames (Section 4.3.4). Tracking object's regions is an effective strategy, since it can cope with deformations, complex motion and occlusion. Region tracking operates in two steps. The first step projects the region descriptors from the current frame into the next frame, and implicitly provides a predicted region partition. The second step refines the region partition to account for the changes in the scene.

Region descriptor projection

The first step for tracking the region partition is the projection of the information at the current frame n into the next frame $n+1$. Each region, $R_i(n)$, is projected by applying motion compensation to its region descriptor, $\phi_i(n)$. This operation is referred to as region descriptor projection. Region descriptor projection updates the position values of a region descriptor by means of its estimated displacement. The region descriptor is defined as

$$\mathbf{\Phi}_i(n) = \Big(\phi_i^1(n), \phi_i^2(n), \phi_i^3(n), \phi_i^4(n), \ldots, \phi_i^{K_i(n)}(n)\Big)^T \tag{4.6}$$

where $K_i(n)$ is the number of features in frame n.

In our specific implementation, $K_i(n) = 8$. In particular, $\big(\phi_i^1(n), \phi_i^2(n)\big)$ represents the position of the region descriptor, and $\big(\phi_i^3(n), \phi_i^4(n)\big)$ its motion vector. The position and the motion vector of the region descriptor are given by the center of mass and by the mean displacement of the pixels belonging to the corresponding region, respectively. $\big(\phi_i^5(n), \phi_i^6(n), \phi_i^7(n)\big)$ represents the mean value of the three color components in the corresponding region, and $\phi_i^8(n)$ is the mean value of the texture feature [36]. The number and the type of features can change according to the application at hand.

The position predicted through motion compensation is given by

$$\begin{cases} \widetilde{\phi}_i^1(n+1) = \phi_i^1(n) + \phi_i^3(n) \\ \widetilde{\phi}_i^2(n+1) = \phi_i^2(n) + \phi_i^4(n) \end{cases} \tag{4.7}$$

The predicted region descriptor, $\widetilde{\mathbf{\Phi}}_i(n+1)$, retains the value of the other features unchanged from frame n to frame $n+1$, so that

$$\widetilde{\mathbf{\Phi}}_i(n+1) = \Big(\widetilde{\phi}_i^1(n+1), \widetilde{\phi}_i^2(n+1), \phi_i^3(n), \phi_i^4(n), \ldots, \phi_i^{K_i(n)}(n)\Big)^T \tag{4.8}$$

The result of region descriptor projection is a prediction of the region partition $\widetilde{\Pi}_r^{n+1}$ in the next frame.

Region partition refinement

The estimated feature values of the projected region descriptors should be refined to adapt the representation to the changes in the scene, to correct the inaccuracies of the projection, to account for region overlapping, and to compensate for changes in viewing conditions. In fact, besides the changes related to the dynamics of the scene, the visual attributes of region descriptors are modified over time due to noise from many sources. Examples of such sources are motion estimation errors, local illumination variations, and sensor noise.

The refinement of the predicted region partition takes place naturally through region segmentation. The projected region descriptors $\widetilde{\mathbf{\Phi}}_i(n+1)$ provide an effective initialization for the clustering process in the next frame. In addition, this initialization implicitly defines a correspondence between regions in frames n and $n+1$. The updated region partition, Π_r^{n+1}, is obtained through the clustering process described in Section 4.3.2. An updated region descriptor, $\mathbf{\Phi}_i(n+1)$, defined as

$$\mathbf{\Phi}_i(n+1) = \Big(\phi_i^1(n+1), \phi_i^2(n+1), \phi_i^3(n+1), \ldots, \phi_i^{K_i(n)}(n+1)\Big)^T, \tag{4.9}$$

is finally associated to each region.

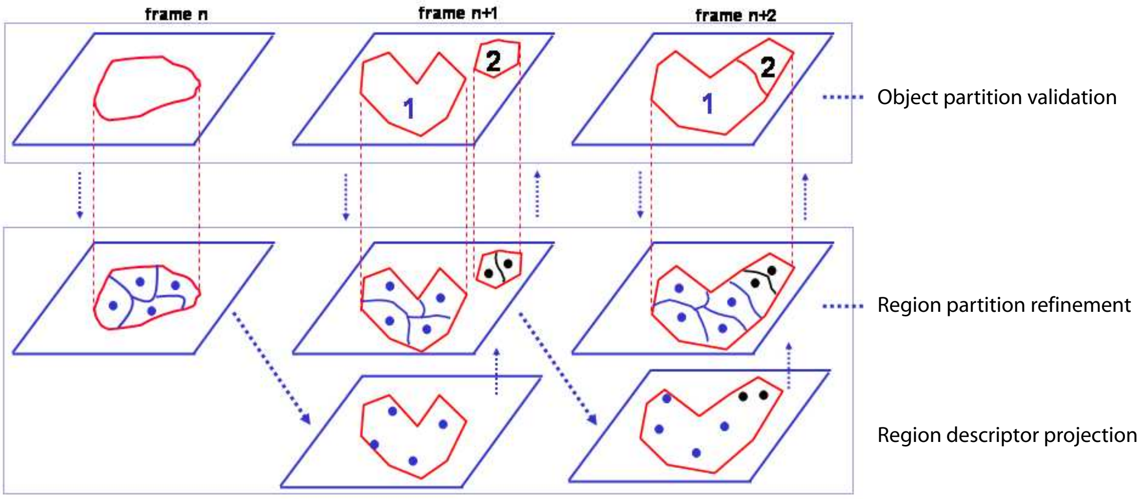

frame n
frame n+1
frame n+2
1
2
1
2
Object partition validation
Region partition refinement
Region descriptor projection

Figure 4.5: Multilevel region-object tracking. The temporal evolution of the object partition is computed through interactions with the region partition. These interactions exploit the tracking of the region partition (bottom) to associate the data from two successive object partitions (top).

4.3.4 Multilevel region-object tracking

The correspondence of video objects in successive frames is achieved through the correspondence of objects' regions. Given the object partition in the new frame and the region partition in the current frame, the proposed region-object tracking procedure performs two different tasks:

1. It defines a correspondence between the object partition in the current frame n and the object partition in the new frame $n + 1$;

2. It provides an effective initialization for the clustering procedure of each object in the new frame $n + 1$. This initialization implicitly defines a preliminary correspondence between the regions in frame n and the regions in frame $n + 1$.

The joint region-object tracking mechanism is organized in two major steps: object partition validation, and data association. The object partition validation step is a feedback from the region partition level to the object partition level, and results in a tentative correspondence. This helps to detect track management issues. The data association step operates at the region level, and validates the tracks through region descriptor correspondence. This second step generates the final correspondence.

Object partition validation

The object partition validation step initializes the tracking process and improves the accuracy of the object partition in case the physical objects in the scene are connected in the image plane. This is achieved through a top-down and a bottom-up interaction with the region partition (Figure 4.5). Before initializing the tracking procedure, each video object is decomposed into a set of non-overlapping regions by means of the multi-feature segmentation method in Section 4.3.2 (Figure

4.5, frame n). Each region $R_j(n)$ is characterized by its region descriptor $\mathbf{\Phi}_j(n)$. To initialize the tracking procedure, each region descriptor $\mathbf{\Phi}_j(n)$ is associated to the corresponding object, $O_i(n)$. After this association, the region descriptor is denoted with $\mathbf{\Phi}_{i,j}(n)$. This operation, referred to as *track initiation*, can be expressed as

$$\forall\ O_i(n) \quad i = 1, \ldots, N_O^n \quad \exists\ \mathbf{\Phi}_{i,j}(n) \quad j = 1, \ldots, N_{R_i}^n, \tag{4.10}$$

with N_O^n number of video objects in frame n, and $N_{R_i}^n$ number of regions for object i. This initialization takes place at the beginning of the tracking process and every time a new video object appears. In this context, a new video object is defined as a set of connected pixels in the object partition which is not associated to another tracked object.

After the initialization, the region descriptors are projected into the next frame by means of region tracking (Section 4.3.3). This operation implicitly corresponds to motion-compensating all the pixels in each region. Let $\mathbf{\Phi}_{i,j}(n)$ be the region descriptor for region $R_{i,j}(n)$. Region descriptor projection provides the predicted descriptor $\widetilde{\mathbf{\Phi}}_{i,j}(n+1)$, to which the predicted region $\widetilde{R}_{i,j}(n+1)$ implicitly corresponds. The predicted region is defined as

$$\widetilde{R}_{i,j}(n+1) = \Big\{(x', y', n+1) : (x, y, n) \in R_{i,j}(n), x' = x + \phi_{i,j}^3(n), y' = y + \phi_{i,j}^4(n)\Big\}, \tag{4.11}$$

where $\big(\phi_{i,j}^3(n), \phi_{i,j}^4(n)\big)$ is the motion vector of $\mathbf{\Phi}_{i,j}(n)$. After the projection, a bottom-up feedback from the region partition refines the topology of the object partition. This feedback generates a tentative correspondence by labeling the object partition Π_o^{n+1} according to the predicted region partition $\widetilde{\Pi}_r^{n+1}$. Once all the pixels in the next object partition are associated to the projected regions, we have a prediction as follows:

$$\widetilde{O}_i(n+1) = \Big\{(x + \phi_{i,j}^3(n), y + \phi_{i,j}^4(n), n+1) : \forall j \in O_i(n), (x, y, n) \in R_{i,j}(n)\Big\}. \tag{4.12}$$

This procedure is straightforward in case each set of connected pixels in Π_o^{n+1} receives projected region descriptors, and receives them from one object only. In such a case, the foregoing procedure suffices to guarantee the tracking. In reality, multiple simultaneous objects may occlude each other and therefore be included in the same set of connected pixels. In these cases, the tentative correspondence is verified through data association in order to define the final correspondence.

Detection of track management issues

Object partition validation permits to detect some of the track management issues, such as the appearance of new objects in the scene, partial and total occlusions, and splitting.

- A *new object* is detected when a connected set of pixels $S(n+1)$ in Π_o^{n+1} does not get any region descriptor from the projection mechanism. The detection of a new object triggers a track initiation (Equation (4.10)).

- An *occlusion* takes place when two or more objects interact, either by getting close one to each other, or by passing one in front of the other. A *partial occlusion* is detected when a connected set of pixels $S(n+1)$ in Π_o^{n+1} receives projected region descriptors from several objects. The object partition validation step separates the objects, that is, provides separate contours for each different object. This refinement is made possible by using the knowledge of the track at the region level, as shown in Figure 4.5 for frame $n+2$.

  In the event of a *total occlusion*, the occluded object disappears from the scene for a time interval corresponding to the duration of the occlusion, and then reappears. If the object does

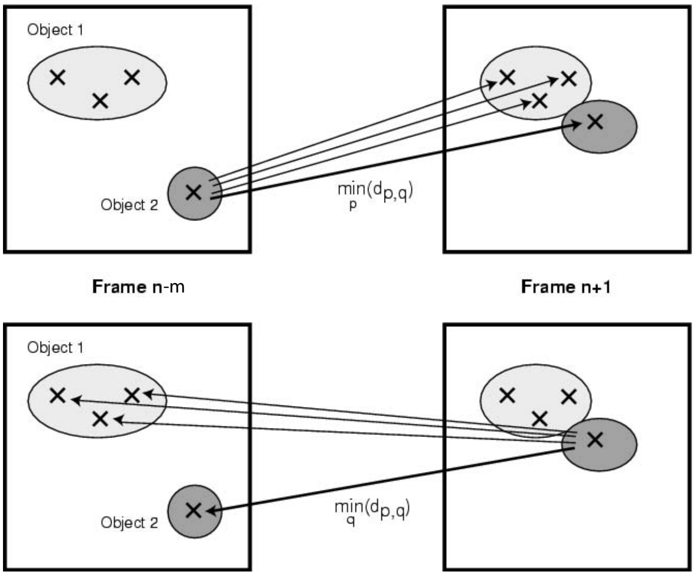


Figure 4.6: Data association. In the data association stage, the region descriptors are put in correspondence over time in order to validate their track.

not undergo massive deformation, maneuver or illumination changes during the occlusion, then the region descriptors of the object just *after* the occlusion will be similar to the motion-compensated region descriptors of the same object just *before* the occlusion. Thus, to relate objects reappearing after total occlusion to their original trajectory, the data association step should operate not only between subsequent frames, but also on a longer temporal window. The size, L, of the temporal window must be superior to the duration of the occlusion.

- A *splitting* corresponds to the separation of a connected set of pixels in the object partition into two or more subsets. This event is detected when two different disconnected sets of pixels $S_1(n+1)$ and $S_2(n+1)$ in Π_o^{n+1} get region descriptors projected from the same video object.

Data association

Data association validates the track of each region descriptor, and as a consequence updates the track of the object partition. This step is particularly important when faced with track management issues. In the data association stage, the region descriptors are put in correspondence over time. First, the predicted region partition $\widetilde{\Pi}_r^{n+1}$ is updated so as to obtain Π_r^{n+1} (Section 4.3.3). Then, the region descriptors corresponding to Π_r^{n+1} are compared with those of the L past frames Π_r^{n-m}, $m \in [0, L-1]$. The size of the temporal data association window, $L \in \mathbb{N}^*$, must be superior to the duration of total occlusions, but small enough to limit computational complexity. In our experiments, a good compromise between reliability and complexity has been obtained by giving the temporal window a size of $L = 50$ frames. With respect to the alternative approach of performing data association between subsequent frames only, a longer temporal window has the following advantages: 1) it enables us to handle total occlusions, as discussed above; 2) it favors tracking stability by reducing the influence of individual data association errors.

Specifically, we consider the proximity between region descriptors in Π_r^{n+1} and in Π_r^{n-m}. The proximity is computed by measuring the Mahalanobis distance in the feature space between the region descriptors in frame $n+1$ and those in past frames $n-m$ (Figure 4.6). To reduce the dimensionality of the problem, a gating process is introduced prior to the distance computation. The gating process allows us to preselect the candidates for data association by eliminating the couples of region descriptors that are highly unlikely to be temporally related. This preselection is based on an Euclidean distance criterion that considers the maximum allowable displacement of a region descriptor between two frames. This results in a lower complexity and favors stability. In our simulations, good results have been obtained by setting the maximum allowable displacement to an Euclidean distance of 50 pixels.

After the gating process, a pair-wise distance metric is applied to all the remaining region descriptors. The region descriptors include information from different sources that are encoded with varying number of features. For example, three features are used for color, and two for motion. We refer to such groups of similar features as *feature categories.* To avoid masking important information when computing the distance, we use separate distance measures, $\mathcal{D}_f(\cdot)$, for each feature category. Since the results of the separate proximity measures will be fused together, it is desirable that $\mathcal{D}_f(\cdot)$ returns a normalized result, especially in the case of poorly scaled or highly correlated features. For this reason we choose the Mahalanobis metric. To compute the proximity of two region descriptors, $\mathbf{\Phi}_{i,j}(n-m)$ and $\mathbf{\Phi}_{k,l}(n+1)$, the Mahalanobis distance can be expressed as

$$\mathcal{D}_f\big(\mathbf{\Phi}_{i,j}(n-m), \mathbf{\Phi}_{k,l}(n+1)\big) = \sqrt{\sum_{s=1}^{K} \frac{\big(\Phi_{i,j}(n-m)^s - \Phi_{k,l}(n+1)^s\big)^2}{\sigma_s^2}}, \tag{4.13}$$

where σ_s^2 is the variance of the s^{th} feature over the entire feature space and K is the number of features. The complete point-to-point similarity measure between $\mathbf{\Phi}_{i,j}(n-m)$ and $\mathbf{\Phi}_{k,l}(n+1)$ is obtained by fusing the distances computed within each category

$$\mathcal{D}\big(\mathbf{\Phi}_{i,j}(n-m), \mathbf{\Phi}_{k,l}(n+1)\big) = \frac{1}{F}\sum_{f=1}^{F} w_f \mathcal{D}_f\big(\mathbf{\Phi}_{i,j}(n-m)^s, \mathbf{\Phi}_{k,l}(n+1)^s\big), \tag{4.14}$$

where F is the number of feature categories and w_f is the weight which accounts for the reliability of each feature category. The value of F may change from frame to frame and from cluster to cluster. The value of the reliability is $w_f = 0$ for those features that have similar values in adjacent regions, and $w_f = 1$ otherwise. The use of the reliability parameter facilitates the data association process by eliminating undiscriminating features from the computation of the distance.

The result of the distance computation can be represented as a matrix $\mathbf{D} = \{d_{p,q}\}$, where each row, p, corresponds to a region descriptor in frame $n+1$, and each column, q, corresponds to a region descriptors in past frames $n-m$. We refer to this matrix as *distance matrix.* Each element of the distance matrix represents the distance between two region descriptors. The smallest element for each row and for each column identifies a possible correspondence between two region descriptors. This result is compared with that of the tentative correspondence to check if there is a conflict. A tentative correspondence between the $\bar{p}^{\text{th}}$ region descriptor in frame $n+1$ and the $\bar{q}^{\text{th}}$ region descriptor in past frames $n-m$ is confirmed if

$$d_{\bar{p},\bar{q}} = \min_q(d_{p,q}) = \min_p(d_{p,q}) \tag{4.15}$$

If the condition in Equation (4.15) is respected, the track is updated. Otherwise, the final correspondence between region descriptors that do not satisfy Equation (4.15) is obtained by means of an iterative process. During this process, the best point-to-point pairs are selected first. Then, the remaining ones are iteratively paired to obtain the final correspondence. This final correspondence is then exploited in the bottom-up feedback to update the object partition.

Object labeling

The predicted partition may not cover all the pixels of Π_o^{n+1}. For the object partition validation step to be complete, each pixel in Π_o^{n+1} has to be classified. If a connected component of Π_o^{n+1} receives region descriptors from one object only, all the unclassified pixels are assigned to that object. If a connected set of Π_o^{n+1} receives region descriptors from several objects, then the unclassified pixels are assigned to the closest projected region. The proximity is computed by measuring the Euclidean distance between the unclassified pixel and the center of mass of the projected regions. The output is a set of video objects that are coherently labeled over time.

4.4 Summary

In this chapter, we have reviewed semantics in the general context, and we have proposed a motion-based semantic video object extraction algorithm that operates in cluttered background. The segmentation of moving objects from the background is produced by a change detection process. The selected approach is robust with regard to camera noise and does not need manual tuning thanks to the use of a locally adaptive threshold. The tracking mechanism is based on feedbacks between an object partition and a region partition. Region descriptor projection produces a tentative correspondence between the object partition in the current frame and the object partition in the new frame. This correspondence is verified through data association in the event of track management issues.

With respect to the alternative approach of computing two separate region partitions in the current and next frames, and then pairing the region descriptors without considering any projection, the proposed approach has several advantages:

- it is computed with data that are already available;
- it is a simple operation;
- it provides an additional element to the final decision for the correspondence;
- it provides an educated initialization for the region partition algorithm in the next frame.

Our solution is capable of dealing with multiple simultaneous objects. Also, track management issues such as appearance and disappearance of objects, splitting and occlusions are resolved through interactions between regions and objects. In Section 6.2.2, the simultaneous tracking of multiple objects in real video will be employed to provide automated event detection and video enhancement for visual surveillance. In Chapter 5, semantic video analysis will be used to prioritize visual data in order to improve the performance of adaptive video delivery. The idea behind this approach is to organize the content so that a particular network or device does not inhibit the main content message.

Adaptive delivery 5

5.1 Introduction

In Chapter 4, semantic analysis has enabled us to extract meaningful information from video. In this chapter, such information is employed to prioritize visual data in order to improve adaptive video delivery. The idea behind this approach is to organize the content in such a way that a particular network or device does not inhibit the main content message. *Adaptive video delivery* consists in distributing content that matches individual appliance and network resources while providing maximum value for the end user. This is required whenever appliances and networks with different characteristics and end users with various preferences access the same content. In the following, a framework for adaptive video delivery is defined based on semantic video objects and on their associated metadata. The proposed framework extends work that has been published by the author et al. in [32, 33, 212, 213]. With reference to Figure 5.1, the input of the framework is a video sequence, and the output is an adapted content stream. The main components are efficient video adaptation methods or *strategies*, means of performance evaluation, and a strategy selection mechanism.

In Section 5.2, a number of complementary video adaptation strategies are discussed. In particular, two new strategies are proposed. The first strategy combines semantic analysis with a traditional frame-based video encoder. The idea behind this *semantic prefiltering* approach is to emulate the Human Visual System (HVS) to prioritize the visual data so as to improve the performance of frame-based coders. The second strategy uses metadata to efficiently encode the main content message. The use of metadata enables us not only to make the content more searchable, but also to improve visualization and to preserve privacy in video-based applications. In Section 5.3, the impact of different adaptation strategies is quantified with subjective experiments. We show that background alterations resulting from semantic prefiltering do not impair overall quality at low bitrates. We also demonstrate that the metadata-based representation of object's shape and motion suffices to convey the meaning and action of a scene when the objects are familiar. Moreover, we propose an objective quality metric that mimics the behavior of human observers. The metric overcomes the limitations of subjective evaluation experiments that are expensive, time consuming and cannot be used to assess video quality in real time. In Section 5.4 at last, we determine the strategy that provides most value for the end user by maximizing performance for a given set of appliance and network resources.

5.2 Video adaptation strategies

In this section, a number of complementary adaptation strategies that fit a wide variety of possible appliance resources, network capacities and user preferences are discussed. An overview of the proposed strategies is provided in Table 5.1. In order to improve the perceived content quality and to provide additional functionalities, such as privacy preservation and automatic video indexing, some of the strategies resort to video content analysis prior to encoding. Specifically, we use semantic analysis to extract relevant areas of a video. These areas are encoded at a higher level of quality or summarized in a textual form. The idea behind this approach is to organize the content so that a particular network or device does not inhibit the main content message. The main content message is dependent on the specific application. In particular, for applications such as visual surveillance and sport video, the main content message is defined based on motion information.

The flow diagram of the video adaptation strategies is depicted in the left part of Figure 5.1. The input video is split into foreground and background parts by means of semantic video analysis. After background simplification, both parts are re-composited together and coded through a frame-based encoder. This approach is referred to as *semantic frame-based encoding.* Alternatively, the frame-based encoder may be used to code the original video sequence, or a subsampled version of the sequence. With the *object-based encoding* method, foreground and background parts are coded separately through an object-based encoder. The background might possibly be simplified prior to coding. Furthermore, metadata are used to efficiently encode the main content message and to enhance relevant parts of a low-quality coded video. These approaches are referred to as *metadata-based encoding* and *metadata-enhanced encoding*, respectively. Some relevant video adaptation examples are illustrated in Figure 5.2.

5.2.1 Background simplification

Background simplification is applied prior to video coding and fulfills two distinct purposes: enhancement of relevant objects, and reduction of background information in order to achieve improved compression. The goal of objects enhancement is to lower the importance of the background so as to put foreground objects in a conspicuous position. This is particularly useful in cluttered environment. To achieve objects enhancement, superfluous visual details may be removed from the background by using a lowpass filter, as shown in Figure 5.3(b). Alternatively, the original background might be replaced by a sketch that provides only the necessary contextual information. In Figure 5.3(d), an edge image that indicates the position and direction of both highway lanes has been used to this end.

Improved compression is achieved by reducing the background information that needs to be coded. A possible solution is to suppress high-frequency components by using a lowpass filter, as depicted in Figure 5.3(b). Another way to take into account less relevant portions of an image before coding is to take advantage of the specifics of the coding algorithm. In the case of block-based coding, DCT coefficients corresponding to high frequencies can be heavily quantized, or set to zero. This is illustrated in Figure 5.3(c), where each background macroblock has been replaced by its DC value. The original background can also be replaced by a static background shot. This helps to eliminate inter-frame coding residues resulting from acquisition noise. At last, background areas might simply be set to a constant value, as shown in Figure 5.3(e).

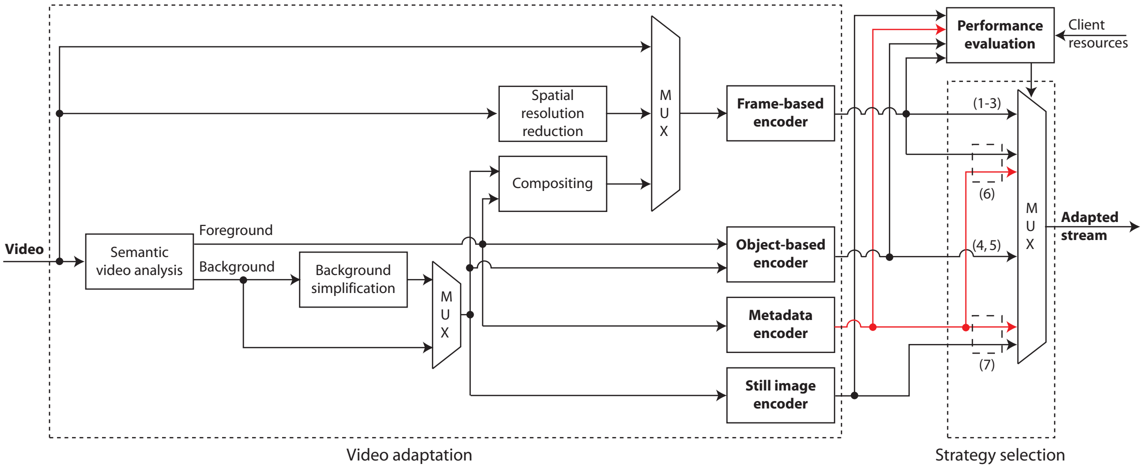
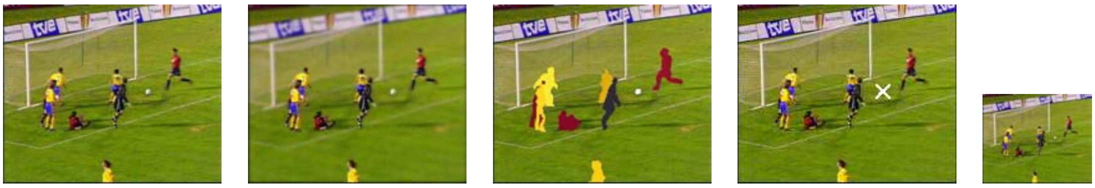


Figure 5.1: Flow diagram of the framework for adaptive video delivery.

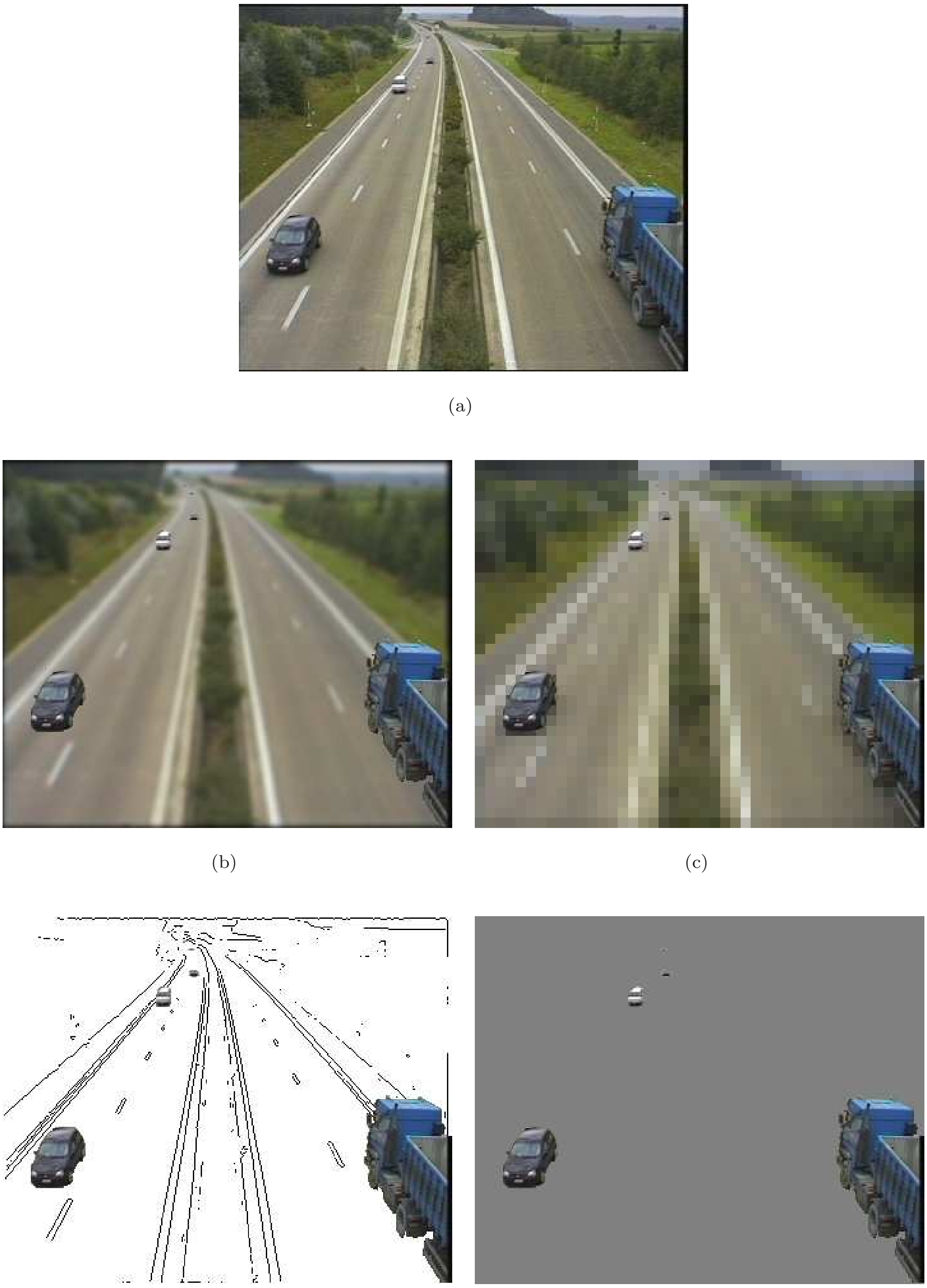
Encoding mode	Description	Use of semantics
frame-based	(1) coded original sequence (2) spatial resolution reduction (3) foreground composited with simplified background prior to encoding (*semantic prefiltering*)	No No Yes
object-based	(4) foreground and background encoded at full bitrate (5) foreground encoded at full bitrate, background simplified prior to encoding	Yes Yes
metadata-enhanced	(6) original video enhanced by object descriptors	Yes
metadata-based	(7) object descriptors superimposed on background	Yes

Table 5.1: Video adaptation strategies based on semantic video analysis and description.



(a) (b) (c) (d) (e)

Figure 5.2: Examples of video adaptation. (a) Sample frame from the sequence *Soccer*. (b) *Semantic frame-based encoding*: the background is selectively lowpass-filtered prior to encoding. (c) *Metadata-based encoding*: object shapes and color are superimposed on the background. (d) *Metadata-enhanced encoding*: metadata are used to enhance relevant portions of a video. (e) Spatial resolution reduction.



(a)

(b)

(c)

(d)

(e)

Figure 5.3: Background simplification for objects enhancement and compression improvement. (a) Sample frame from the *Highway* sequence. (b) The original background is lowpass-filtered. (c) Each background macroblock is replaced by its DC value. (d) An edge image is used instead of the original background. (e) Background areas are set to a constant value.

The compression improvements that have been achieved by means of different background simplification methods are compared in Table 5.2. Foreground and background parts of the *Highway* sequence have been encoded separately through the MoMuSys MPEG–4 VM reference software version 1.0 [109], using VM5+ global rate control. The overall bitrate has been set to 300 Kbit/s. The average bitrate required to code the background is reduced by 14 Kbit/s when a lowpass filter is used. It is more than halved when each background macroblock is replaced by its DC value. However, the human eye is not able to recover contextual information properly from such background. The highest compression savings are achieved when an edge image is used, but this does not provide any indications about the color or texture of the original background.

Background	**Original**	**Lowpass filtered**	**DC components**	**Edge image**
Bitrate (Kbit/s)	104	90	50	46

Table 5.2: Compression improvements achieved by different background simplification methods. The table gives the average bitrate required to encode the background of the *Highway* sequence using object-based MPEG–4. The overall bitrate has been set to 300 Kbit/s for all versions.

5.2.2 Semantic frame-based encoding

The semantic frame-based encoding mode exploits semantics in a traditional frame-based encoding framework (e.g., MPEG–1 [105]). The use of semantic video analysis followed by background simplification and compositing, referred here as *semantic prefiltering* (Figure 5.4), helps to support low bandwidth transmission. The areas belonging to the foreground class, or semantic objects, are used as region of interest. The areas not included in the region of interest are lowered in importance by using background simplification. Using a simplified background aims at taking advantage of the task-oriented behavior of the HVS for improving compression ratios. Recent work on foveation [113] demonstrated that using nonlinear integration of low-level visual cues mimicking the processing in primate occipital and posterior parietal cortex allows one to sensibly increase compression ratios. Moreover, the work reported in [22] demonstrated that an overall increase in image quality can be obtained when the increase in quality of the relevant areas of an image more than compensates for the decrease in quality of the image background. An example of this solution is reported in Figure 5.5(a). On the other hand, filtering the entire image inhibits the main content message (Figure 5.5(b)).

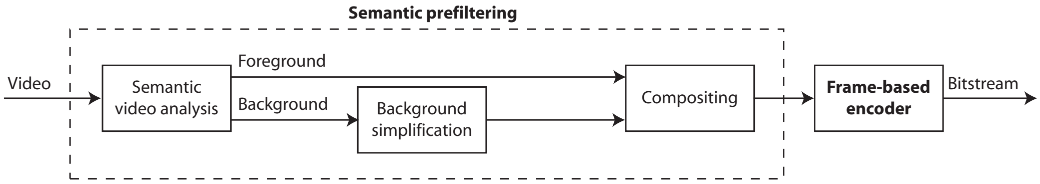


Figure 5.4: Semantic prefiltering is the process of semantic video analysis, followed by background simplification and compositing.

5.2.3 Object-based encoding

With object-based encoding, the encoder needs to support the coding of individual video objects (e.g., object-based MPEG–4 [108]). Each video object is assigned to a distinct object class, according

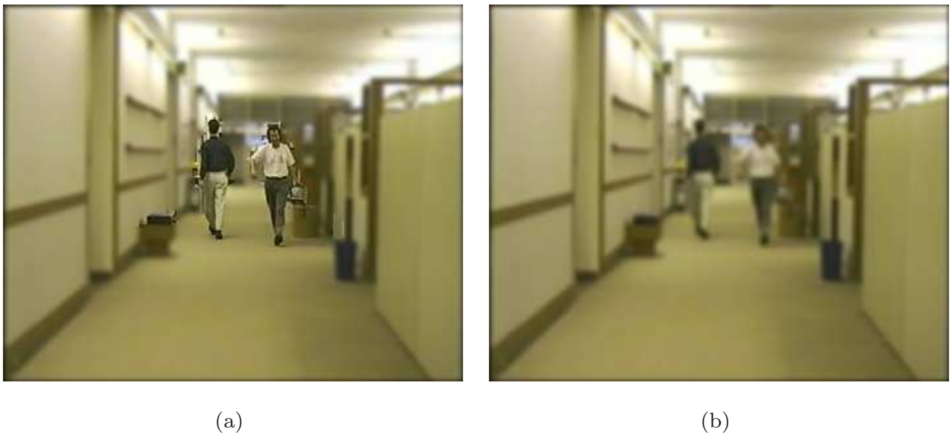


Figure 5.5: Selective lowpass-filtering simplifies the information in the background, while still retaining essential contextual information. (b) Filtering the entire image inhibits the main content message.

to its importance in the scene. The encoding quality can be set depending on the object class: the higher the relevance, the higher the encoding quality. One advantage of this approach is the possibility of controlling the sequencing of objects: video objects may be encoded with different degree of compression, thus allowing better granularity for the areas in the video that are of more interest to the viewer. Moreover, objects may be decoded in their order of priority, and the relevant content can be viewed without having to reconstruct the entire video (network limitations). Another advantage is the possibility of using a simplified background, so as to enhance the relevant objects (appliance limitations).

5.2.4 Metadata-based and metadata-enhanced encoding

A further processing of the video content is performed to cope with limited device or network capabilities as well as to automatically generate metadata (e.g., MPEG–7 [110, 111]). Such processing transforms the foreground objects extracted through semantic analysis into quantitative descriptors and permits video annotation. Video annotation is desirable for applications such as video surveillance, where terabytes of data are produced and need to be searched quickly. Moreover, the descriptors can be transmitted instead of the video content itself and superimposed by the terminal on a still background (*metadata-based encoding*). This approach is useful to preserve privacy in video surveillance applications as well as to reduce bandwidth requirements under critical network conditions. For example, an object identifier, a shape descriptor and a color descriptor are used in [212, 213]. The object identifier is a unique numerical identifier describing the spatial location of each object in the scene. The shape descriptor is used to represent the shape of an object, ranging from a bounding box to a polygonal representation with a different number of vertices. A progressive representation is used where the number of vertices corresponding to the best resolution is computed, and any number of vertices smaller that this maximum can be used according to the requirements of the application. The color descriptor defines up to eight dominant colors for each object. An example is shown in Figure 5.2(c), where the shape and color of soccer players have been superimposed on the background. Other features such as texture may be added in the description process. The choice of these additional features depends on the application at hand.

Conditions for evaluation	According to ITU-T Recommendation P.910 [115]
Assessment method	ACR (Absolute Category Rating)
Source of signals	Personal Computer
Monitor	18” computer monitor with digital interface
Viewing distance	6-8 H
Test sequences	*Akiyo*, *Hall monitor*, *Children*, *Coastguard* (Cif, 25 frames/s)
Presentation duration	8 sec. + max. 10 sec. for vote
Assessors and presentations	20 non-expert observers, 2 sessions per observer: 75 presentations for *frame-based* session, 12 presentations for *metadata-based* session

Table 5.3: Conditions for subjective evaluation experiments.

In addition to the above, the descriptors can be transmitted along with the video itself and used for rendering the video content. This solution, consisting in a mixture of video-based and text-based modalities, is here referred to as *metadata-enhanced encoding*. Using metadata-enhanced encoding, content descriptors help enhance parts of the video that are hidden or difficult to perceive due to heavy compression. In this case, the video itself is the background and the descriptors highlight relevant portions of the data. One example is shown in Figure 5.2(d), where the position of the ball in a soccer game has been highlighted for transmission to a PDA or a mobile phone.

5.3 Performance evaluation

Perceptual video quality assessment is a difficult task already when dealing with traditional coders [167]. When dealing with object-based coders and multiple modalities, the task becomes even more challenging. For this reason, we use a combination of subjective and objective evaluation techniques to quantify the impact of different video adaptation strategies. We show that background alterations resulting from semantic prefiltering do not impair overall quality at low bitrates. We also demonstrate that the metadata-based representation of object’s shape and motion suffices to convey the meaning and action of a scene when the objects are familiar. In addition to the above, we propose an objective quality metric, the semantic peak signal-to-noise ratio (SPSNR), that accounts for different image areas and for their relevance to the observer in order to reflect the focus of attention of the HVS. The metric overcomes the limitations of subjective evaluation experiments that are expensive, time consuming and cannot be used to assess video quality in real time.

5.3.1 Subjective evaluation

Experimental setup

Four test sequences from the MPEG–4 Video Content Set are used for subjective performance evaluation: *Akiyo*, *Hall monitor*, *Children* and *Coastguard* (Figure 5.6). The sequences include deforming and rigid objects of different size, complex as well as simple background, and different types of motion. For frame-based encoding, the TMPGEnc 2.521.58.169 MPEG–1 codec with constant bitrate (CBR) rate control is used. The coding structure is ‘IBBPBBPBBPBBPBBPBB’. Bitrates are chosen so as to range from the lowest bitrate supported by the codec, up to perceptually lossless coding.

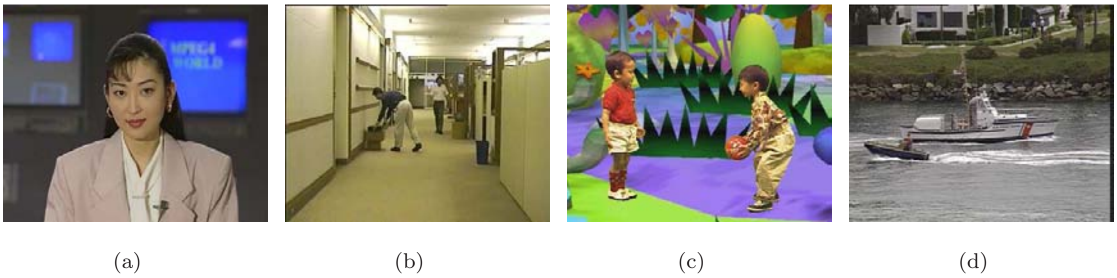


Figure 5.6: Sample frame from the test sequences for subjective performance evaluation. (a) *Akiyo*. (b) *Hall monitor*. (c) *Children*. (d) *Coastguard*.

Since we expect results to stabilize at high bitrates, tested rates are distributed exponentially: 200, 250, 300, 500 and 1000 Kbit/s for all sequences, plus 150 Kbit/s for *Akiyo* and *Hall monitor*, and 100 Kbit/s for *Akiyo*. Semantic prefiltering is either achieved by lowpass filtering, or by replacing the original background by a static background shot (*Hall monitor*). In the former case, the background is simplified using a Gaussian 9 × 9 lowpass filter with $\mu = 0$ and $\sigma = 2$. The foreground is hand-segmented in order to avoid bias due to segmentation errors.

For metadata-based encoding, the Expway 02/11/07 MPEG–7 *BiM* Payload coder is used. Here, coding bitrates depend on the complexity of the description. Video object's location, bounding box and approximate shape are summarized by MPEG–7 *Visual* Descriptors [110]. `Motion trajectory` gives the spatial location of objects' gravity center. `Region locator` defines the approximate shape of objects using a bounding box or a 30-sided polygon. The descriptors are organized so as to provide a layered description of the scene [213]. That is, location and shape can be defined in any desired combination and order.

The conditions for subjective evaluation experiments in Table 5.3 follow the *Absolute Category Rating* (ACR) method, according to ITU-T Recommendation P.910 [115]. ACR is well-suited for qualification tests (i.e., to compare the performance of different adaptation strategies), as the method does not use explicit references. Twenty non-expert observers of different ages and backgrounds are presented a series of video sequences in random order; the presentation order is modified for each observer. Each observer participates in two sessions: the *frame-based* session contains 75 presentations, and the *metadata-based* session contains 12 presentations. After each presentation, observers rate the quality of the sequence on a scale ranging from 0 (bad) to 100 (excellent). The presentation duration is 8 seconds, and a maximum of 10 seconds is allowed for voting. Before each session, the range of qualities is presented to the observers in a training phase.

Statistical analysis of subjective evaluation results

Subjective experiments produce distributions of integer values, each number corresponding to one vote. These distributions exhibit a number of variations due to the difference in judgement between observers, and to the effect of a variety of conditions associated with the experiments. Specifically, a *session* consists of a number of *presentations* L. A presentation is obtained by applying one of a number of *test conditions* J, to one of a number of *test sequences* K. Each combination of test sequence and test condition may be *repeated* a number of times R. The *mean score* for each presentation, $\overline{u}_{jkr}$, is thus given by

$$\overline{u}_{jkr} = \frac{1}{N}\sum_{i=1}^{N} u_{ijkr}, \tag{5.1}$$

```
1: Screening
for $l = 1$ to $L$ do
    Calculate mean score, $\overline{u}_{jkr}$
    Calculate standard deviation, $S_{jkr}$
    Calculate kurtosis coefficient, $\beta_{2jkr}$
end
for $j, k, r = 1, 1, 1$ to $J, K, R$ do
    if $2 \leqslant \beta_{2jkr} \leqslant 4$ then
        if $u_{ijkr} \geqslant \overline{u}_{jkr} + 2S_{jkr}$ then $P_i = P_i + 1$;
        if $u_{ijkr} \leqslant \overline{u}_{jkr} - 2S_{jkr}$ then $Q_i = Q_i + 1$;
    else
        if $u_{ijkr} \geqslant \overline{u}_{jkr} + \sqrt{20}S_{jkr}$ then $P_i = P_i + 1$;
        if $u_{ijkr} \leqslant \overline{u}_{jkr} - \sqrt{20}S_{jkr}$ then $Q_i = Q_i + 1$;
    end
end
if $\frac{P_i+Q_i}{J \cdot K \cdot R} > 0.05$ and $|\frac{P_i-Q_i}{P_i+Q_i}| < 0.3$ then reject observer $i$;
```

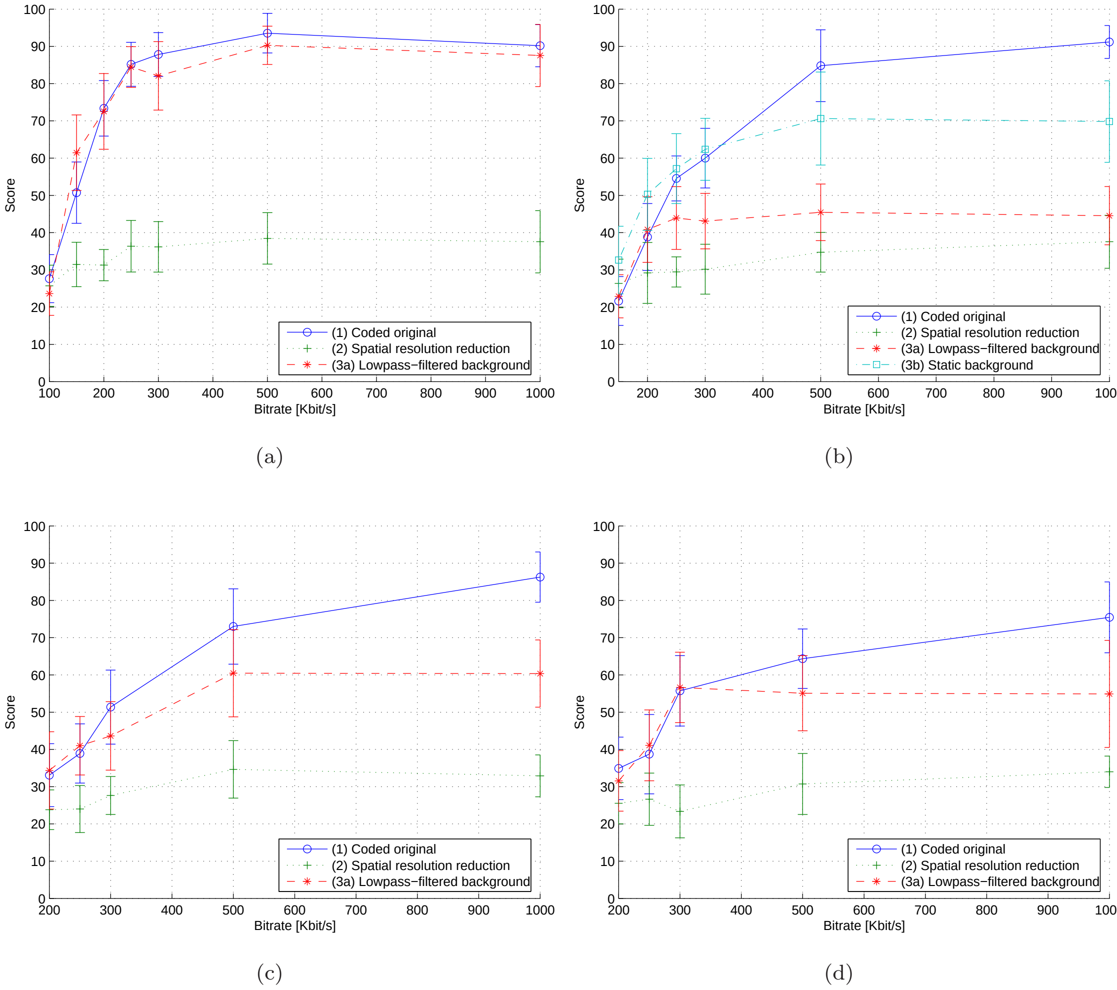
where u_{ijkr} is the score of observer i for test condition j, sequence k, and repetition r. N is the total number of observers. The associated *confidence interval* is derived from the standard deviation and size of each sample. It is proposed to use the 95% confidence interval, which is given by $[\overline{u}_{jkr} - \delta_{jkr}, \overline{u}_{jkr} + \delta_{jkr}]$, where $\delta_{jkr} = 1.96 \cdot (S_{jkr}/\sqrt{N})$. S_{jkr} is the standard deviation for each presentation.

Votes from unreliable observers are discarded using a *screening procedure*, organized in two stages. The first stage ensures that responses were entered accurately and in accordance with the experimental instructions. In the second stage, the variability of the data is reduced using the two-step method described in Annex 2 of ITU-R Recommendation BT.500-11 [116]. This method operates as follows (Algorithm 1). First, an expected range of values is calculated for each presentation. The expected range depends on whether or not the subject distribution is normal. If the distribution is normal, i.e., kurtosis coefficient $\beta_{2jkr} \in [2, 4]$, then the expected range is $\overline{u}_{jkr} \pm 2S_{jkr}$. If the distribution is non-normal, the expected range is increased to $\overline{u}_{jkr} \pm \sqrt{20}S_{jkr}$. Then, the expected ranges are applied to the judgement of each observer. To be rejected, observer i has to record annoyance values outside the expected range for more than 5% of the presentations ($\frac{P_i+Q_i}{J \cdot K \cdot R} > 0.05$). In addition, the proportion of outlying observations on both sides of the range is required to be roughly equal: $|\frac{P_i-Q_i}{P_i+Q_i}| < 0.3$. Thus, a subject is rejected for being erratic on both sides of the range, but not for being always above or always below the expected range. The results of subjective quality evaluation experiments for the frame-based session and for the metadata-based session are summarized in Figure 5.7 and in Table 5.4, respectively. The mean quality and associated 95% confidence interval are given along with the coding bitrate.

Discussion: frame-based session

Following the classification of Table 5.1, the frame-based adaptation strategies under analysis are: (1) coded original sequence; (2) spatial resolution reduction; (3a) semantic prefiltering with lowpass-filtering; (3b) semantic prefiltering with static background (*Hall monitor*)[*]. From Figure 5.7, it is

[*] Semantic prefiltering with static background is only relevant for the sequence *Hall monitor*. Indeed, *Coastguard* has a changing background that cannot be replaced by a static version without significant loss of contextual information. The original backgrounds of *Akiyo* and *Children* are synthetic, and they are not corrupted by any noise. Thus,



(a)

(b)

(c)

(d)

Figure 5.7: Subjective evaluation results for frame-based session. The graphs show the mean quality and associated 95% confidence interval as a function of bitrate. (a) *Akiyo*. (b) *Hall monitor*. (c) *Children*. (d) *Coastguard*.

possible to notice that semantic prefiltering has a positive impact at low bitrates, in particular when the original background is replaced by a static frame or by a sprite representing the background (3b). At bitrates up to 300 Kbit/s, this increases the mean quality by up to 10 points as compared to the coded original (1). This is because inter-coded, static background blocks do not produce residue, so most of the available bitrate can be allocated to foreground objects.

Lowpass-filtering (3a) has a lesser impact. Viewers notice the improvement of foreground quality due to the additional bandwidth freed by the filter, but at the same time they are annoyed by the loss of background information. For *Akiyo*, the quality of lowpass-filtered and coded original versions is similar over the entire bitrate range. This is because the background of the original sequence is out of focus, and thus it has few high-frequency components. For *Hall monitor*, the mean quality of lowpass-filtering is slightly above that of the coded original (+1.5) at bitrates up to 200 Kbit/s. The same is true (+1.3) for *Children* at bitrates up to 250 Kbit/s. For *Coastguard*, lowpass-filtering has

replacing the original background by a static version would have no effect.

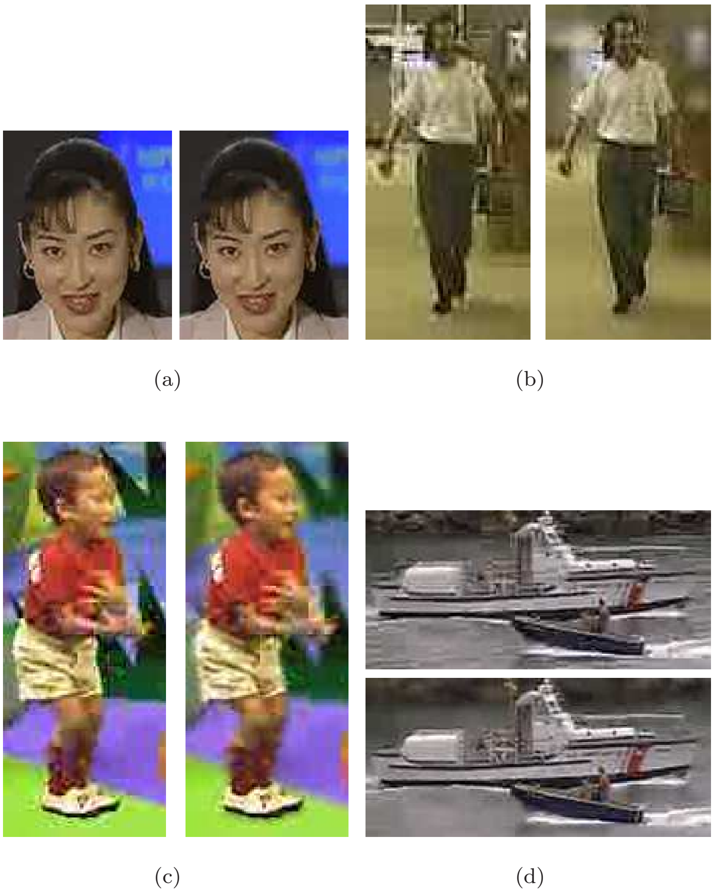


Figure 5.8: Frame details *with* and *without* semantic prefiltering. (Left/top) coded original. (Right/bottom) semantic prefiltering with lowpass filtering. (a) *Children*. (b) *Coastguard*. (c) *Hall monitor*. (d) *Akiyo*.

been rated above the coded original (+2.5) at bitrates of 250 and 300 Kbit/s, but below (-3.5) at the lowest bitrate of 200 Kbit/s. This is because at 200 Kbit/s, foreground objects are corrupted by heavy artifacts in both versions, whereas at 250 and 300 Kbit/s, lowpass-filtering notably reduces artifacts that are still visible in the coded original. The improvement of foreground quality can be verified in Figure 5.8. Semantic prefiltering notably enhances the face in *Children* and the boats in *Coastguard*. Note that the subjective mean scores in Figure 5.7 are sometimes decreasing even though they are expected to be increasing monotonically as a function of bitrate. This is due to the difficulty in assessing video sequences that have almost identical quality, such as *Akiyo* coded at 500 Kbit/s and at 1000 Kbit/s.

Background simplifications resulting from semantic prefiltering do not penalize overall quality at low bitrates (100-250 Kbit/s). In fact, image degradations are strong at such bitrates, and improvements on important image parts due to the additional bandwidth freed by background simplification are positively perceived. At high bitrates on the other hand, both foreground and background are coded at high quality. Thus, background alterations are easily noticed by observers and degrade the overall impression.

Description	(7a) Location	(7b) Bounding box	(7c) Polygon shape
Akiyo	$\overline{u} = 5$ (±6), r=4 Kb/s	$\overline{u} = 11$ (±8), r=11 Kb/s	$\overline{u} = 32$ (±14), r=20 Kb/s
Hall monitor	$\overline{u} = 5$ (±7), r=6 Kb/s	$\overline{u} = 14$ (±8), r=16 Kb/s	$\overline{u} = 61$ (±18), r=30 Kb/s
Children	$\overline{u} = 9$ (±6), r=12 Kb/s	$\overline{u} = 25$ (±9), r=32 Kb/s	$\overline{u} = 71$ (±13), r=58 Kb/s
Coastguard	$\overline{u} = 3$ (±4), r=8 Kb/s	$\overline{u} = 9$ (±7), r=21 Kb/s	$\overline{u} = 55$ (±8), r=39 Kb/s

Table 5.4: Subjective evaluation results for metadata-based session. The mean score $\overline{u}$ is given along with the 95% confidence interval (in brackets), and with the encoding cost r.

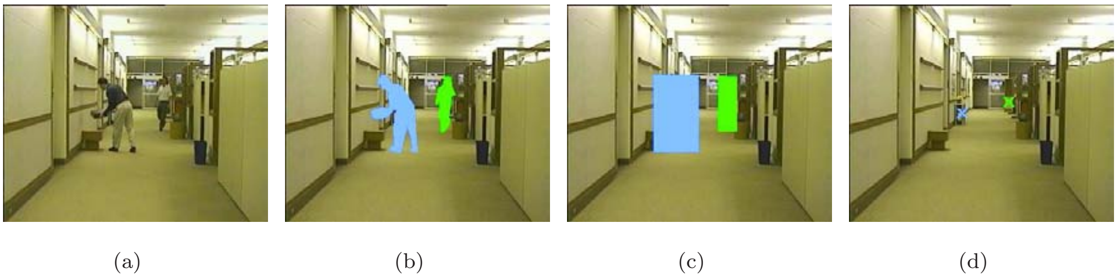


Figure 5.9: Metadata-based adaptation strategies. (a) Sample frame from the *Hall monitor* sequence. (b) Approximation of object's shape with 30-sided polygons. (c) Description of object's bounding box. (d) Description of object's location.

Discussion: metadata-based session

The metadata-based adaptation strategies under analysis are the following: (7a) description of objects' location; (7b) description of objects' bounding boxes; (7c) approximation of objects' shape with 30-sided polygons. A sample frame from each representation is given in Figure 5.9 for the sequence *Hall monitor*. Results in Table 5.4 demonstrate that the representation of objects' shape (7c) suffices to convey the meaning and action of a scene when the objects are familiar. The sequences *Hall monitor*, *Children* and *Coastguard* have all been given mean scores above 50. This is because the represented objects have familiar shape and behavior. However, missing texture and color information are penalizing for *Akiyo*, which has close-up shots and few action.

The description of objects' location (7a) has been rated below 10 for all test sequences. This is not surprising, as this description does not convey sufficient knowledge about the scene in the general case. In particular, the nature of the objects cannot be verified. However, location information might be valuable for automated event detection, as discussed in Chapter 6.2.2. The representation of bounding boxes (7b) has been rated low as well. Even though this representation does provide indications about the occupation of the image space by objects, it does not convey sufficient knowledge about the objects' nature. In *Children* however, the dynamics of the box that represents the bouncing ball enable the observer to partially understand the scene's action. This explains why the mean score is about 10 points higher for this sequence than for the remaining ones.

The bitrate required to code the objects' shape (7c) using MPEG–7 *BiM* is about one fifth of the bitrate for low-quality MPEG–1. However, subjective scores cannot be compared one-to-one between these two adaptation methods, as observers tend to rate frame-based strategies in terms of video quality, whereas they assess metadata-based strategies in terms of their ability to understand the meaning and action of the scene.

5.3.2 Objective evaluation

Quality metric

Subjective evaluation experiments are expensive, time consuming and cannot be used to assess video quality in real time. An objective evaluation metric would therefore be desirable. The widely used peak signal-to-noise ratio (PSNR) or its alternative, the mean squared error (MSE), cannot represent the exact perceptual quality because it disregards the viewing condition and the characteristics of human perception [80]. Perceptual quality metrics address this issue by taking into accounts certain aspects of the HVS, such as color space, contrast sensitivity, masking and detection [167, 263]. Recent metrics account for different image areas, or object classes, and for their relevance to the observer in order to reflect the focus of attention of the HVS [46, 47, 129, 140, 168, 177].

We take into account object classes through a distortion measure, the *semantic mean squared error*, SMSE, defined as:

$$\mathrm{SMSE} = \sum_{k=1}^{N} w_k \cdot \mathrm{MSE}_k, \tag{5.2}$$

where N is the number of object classes and w_k is the weight of class k. Class weights are chosen depending on the semantics, with $w_k \geqslant 0, \forall k = 1, \ldots, N$ and $\sum_{i=1}^{N} w_k = 1$. The mean squared error of each class, MSE_k, can be written as

$$\mathrm{MSE}_k = \frac{1}{|C_k|} \sum_{(i,j) \in C_k} d^2(i,j). \tag{5.3}$$

C_k is the set of pixels belonging to the object class k, and $|C_k|$ is its cardinality. The class membership of each pixel (i,j) is defined by semantic video analysis. The error $d(i,j)$ between the original image I_O and the distorted image I_D in Equation (5.3) is the pixel-wise color distance. The color distance is computed in the 1976 CIE *Lab* color space in order to consider perceptually uniform color distances with the Euclidean norm and is expressed as:

$$d(i,j) = \sqrt{\left(\Delta I^L(i,j)\right)^2 + \left(\Delta I^a(i,j)\right)^2 + \left(\Delta I^b(i,j)\right)^2}, \tag{5.4}$$

with $\Delta I^L(i,j) = I_O^L(i,j) - I_D^L(i,j)$, $\Delta I^a(i,j) = I_O^a(i,j) - I_D^a(i,j)$, and $\Delta I^b(i,j) = I_O^b(i,j) - I_D^b(i,j)$. The final quality evaluation metric, the *semantic peak signal-to-noise ratio*, SPSNR, is the following:

$$\mathrm{SPSNR} = 10 \log_{10} \left(\frac{V_{\max}^2}{\mathrm{SMSE}} \right), \tag{5.5}$$

where $V_{\max}$ is the maximum peak-to-peak value of the color range.

When the object classes are foreground and background, then $N = 2$ in Equation (5.2). If we furthermore denote with w_f the foreground weight, then SPSNR $\equiv$ PSNR when $w_f = 0.5$. The larger w_f, the more important the contribution of the foreground. When $w_f = 1$, then the foreground only is considered in the evaluation of the peak signal-to-noise ratio. An illustration of the impact of w_f in the distortion measure is given in Figure 5.10. The figure presents a comparison of the average SPSNR of the sequence *Hall monitor* for the different adaptation strategies described in Section 5.2 as a function of w_f. The value of w_f is computed as described in the following Section.

Foreground relevance

Subjective experiments quantify the amount of attention that we pay to the foreground and to the background. The foreground weight, w_f, is determined by maximizing the Pearson correlation

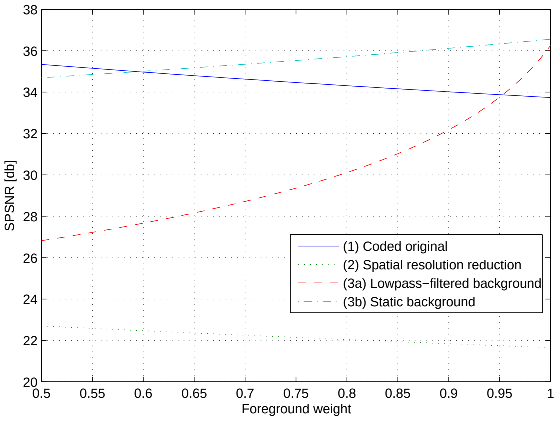


Figure 5.10: Illustration of the impact of w_f in the distortion measure: average SPSNR versus foreground weight for the *Hall monitor* sequence. Content-blind coding methods (1)-(2) decrease their performance when the foreground is given more importance. Methods based on semantic, (3a) and (3b), increase their performance when the foreground is given more importance.

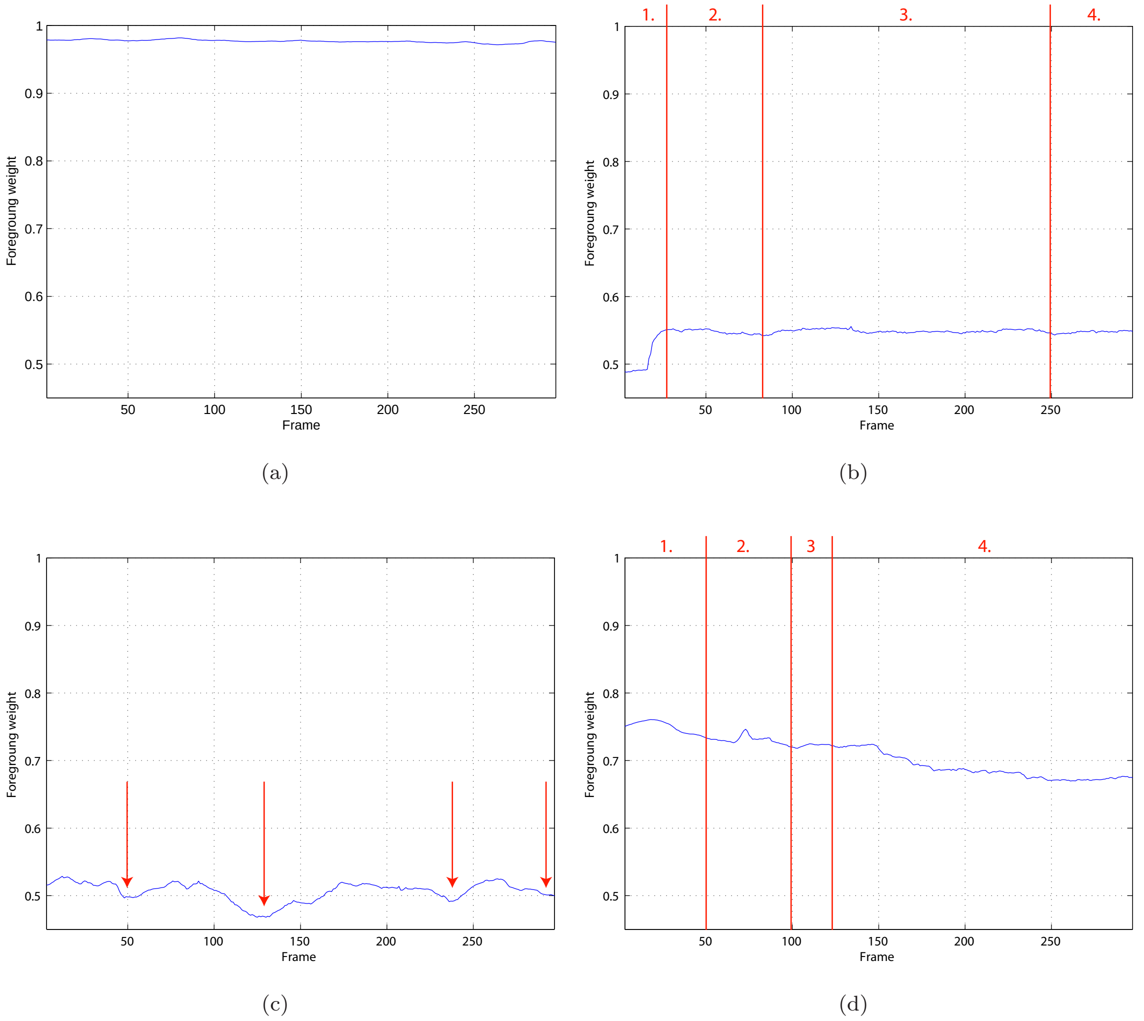
(Equation (5.7)) between SPSNR and subjective results. For the sequence *Akiyo*, where the foreground covers a large area and the background is simple, the observers focused mostly on foreground, thus leading to a value of $w_f = 0.97$. For *Hall monitor*, whose background is more complex and objects are smaller, the foreground attracted less attention ($w_f = 0.55$). The sequence *Children* has a very complex and colored background that attracted the observer's attention, thus resulting in foreground and background being equally weighted ($w_f = 0.5$). The sequence *Coastguard* contains camera motion. This prevented the observer from focusing on background steadily, even though it is quite complex. In this case, $w_f = 0.7$. In general, results confirm that large moving objects and complex background tend to attract user's attention.

Based on the data collected with subjective experiments, we predict the foreground weight using the following formula:

$$w_f = (\alpha - \beta \cdot \sigma_b) \cdot r + \gamma \cdot v + (\sigma_b + 1) \cdot \delta, \tag{5.6}$$

where r represents the portion of the image occupied by foreground pixels: $r = |C_f|/(|C_f| + |C_b|)$, with $|C_f|$ and $|C_b|$ representing the number of foreground and background pixels, respectively. The standard deviation of the luminance of background pixels, σ_b, is a simple measure of the texturedness of the background and represents the background complexity. The presence of camera motion is considered with v: $v = 1$ for moving camera, and $v = 0$ otherwise. α, β, γ and δ are constants whose values have been determined based on the results of the subjective experiments using least square optimization: $\alpha = 5.7$, $\beta = 0.108$, $\gamma = 0.2$ and $\delta = 0.01$.

Equation (5.6) has been used to compute the foreground weight, w_f, as a function of time. The corresponding graphs are shown in Figure 5.11, where important content segments are highlighted. As expected, w_f is almost constant for *Akiyo*, since the background complexity, σ_b, and the portion of the image occupied by foreground pixels, r, do not show any significant variations. This reflects the fact that in this sequence, there is no change in the filmed action. The action in *Hall monitor* is arranged in four segments. In the first segment (frames 1-24), a person enters the room from the left. In the second segment (frames 25-81), the person walks away from the camera, and a second person enters the room from the right. In the third segment (frames 82-249), the person on the right walks toward the camera, and the person on the left walks away from the camera and leaves the room. In the last segment (frames 250-300), the person on the right continues to walk toward the



(a)

(b)

(c)

(d)

Figure 5.11: Foreground weight, w_f, as a function of time. (a) *Akiyo*. (b) *Hall monitor*. (c) *Children*. (d) *Coastguard*. Important content segments are highlighted.

camera. The foreground weight, w_f, increases when the portion of the image occupied by foreground pixels, r, increases as well. For instance, the average foreground weight in the first segment, where the first person enters the room, is $w_f = 0.49$, whereas $w_f = 0.55$ in the third segment, where both people are visible. This reflects the fact that large moving objects tend to attract the attention. The foreground weight of the sequence *Children* goes through four local minima in the vicinity of frames 50, 130, 235 and 290. Each minimum corresponds to one of the children kneeling down to pick up the ball. As a consequence, the portion of the image occupied by foreground pixels, r, decreases, and the temporarily uncovered background tends to attract some additional attention. The action in *Coastguard* at last is organized in four segments. At the beginning of the sequence, one boat is visible in the scene. In the first segment (frames 1-50), a second boat enters the scene. In the second segment (frames 51-97), both boats are visible and the camera moves up. In the third segment (frames 98-120), the first boat leaves the scene. In the last segment (frames 121-300), only the second boat is visible. The effect of these segments is not clearly perceptible in Figure 5.11(d). The

	Akiyo	*Hall monitor*	*Children*	*Coastguard*
$\mathbf{w_f}$	0.97	0.55	0.50	0.7
$\mathbf{r_p(w_f)}$	0.95	0.90	0.95	0.92
$\mathbf{r_p(0.5)}$	0.87	0.89	0.95	0.90
$\mathbf{r_s(w_f)}$	0.90	0.84	0.95	0.93
$\mathbf{r_o(w_f)}$	0.10	0.11	0.07	0.07

Table 5.5: Characterization of the prediction performance of SPSNR. For each test sequence, the table shows the foreground weight w_{f}, along with the Pearson linear correlation coefficient r_p, Spearman rank-order correlation coefficient r_s, and outlier ratio r_o. The Pearson correlation of PSNR, $r_p(0.5)$, is given for reference.

reason is that in Equation (5.6), fluctuations of the background complexity, σ_b, affect the foreground weight to a larger extent than variations of the image portion occupied by foreground pixels, r. The fluctuations of σ_b result from background illumination changes due to the moving camera.

Results in Figure 5.11 reflect the fact that observer's attention tends to be attracted by large moving objects and complex background. However, the impact of background complexity fluctuations on the foreground weight is sometimes overrated (*Coastguard*). We would also like to point out that w_f is allowed to drop below 0.5, as in Figure 5.11(c). This accounts for the phenomenon that in certain occasions (e.g., complex background), observers might devote more attention to the background than to moving objects.

Discussion

The prediction performance of a visual quality metric with respect to subjective ratings is characterized by a number of attributes. These attributes are *accuracy*, *monotonicity* and *consistency* [239, 250]:

- *Accuracy* is the ability of a metric to predict subjective ratings with minimum average error. It can be determined by means of the Pearson linear correlation coefficient. For a set of N data pairs (x_i, y_i), the Pearson correlation, r_p, is defined as:

$$r_p = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}\sqrt{\sum(y_i - \overline{y})^2}}, \tag{5.7}$$

  where $\overline{x}$ and $\overline{y}$ are the means of the respective data sets.

- *Monotonicity* measures whether the increase/decrease in one variable are associated with increase/decrease in the other variable, independent of the magnitude of the increase/decrease. Ideally, differences of a metric's rating between two sequences should always have the same sign than the differences between the corresponding subjective ratings. The degree of monotonicity can be quantified by the Spearman rank-order correlation coefficient, r_s, which is defined as:

$$r_s = \frac{\sum(\chi_i - \overline{\chi})(\gamma_i - \overline{\gamma})}{\sqrt{\sum(\chi_i - \overline{\chi})^2}\sqrt{\sum(\gamma_i - \overline{\gamma})^2}}, \tag{5.8}$$

  where χ_i is the rank of x_i, and γ_i is the rank of y_i in the ordered data series; $\overline{\chi}$ and $\overline{\gamma}$ are the respective midranks.

- The *consistency* of a metric's prediction can be evaluated by measuring the number of outliers. An outlier is defined as a data point (x_i, y_i) for which the prediction error is larger than a

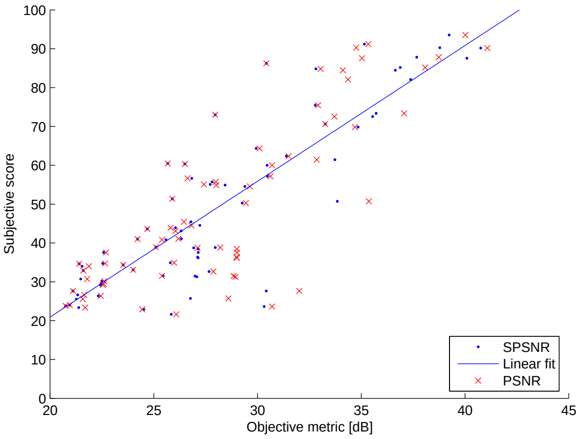


Figure 5.12: Scatter plot of subjective evaluation results versus model prediction for the complete data set. Blue dots indicate scores obtained using SPSNR, and red crosses indicate scores obtained using PSNR. The least-square linear fit of the data is shown in order to facilitate the visual estimation of prediction error.

certain threshold, such as twice the standard deviation σ_{y_i} of the subjective rating differences for this data point: $|x_i - y_i| > 2\sigma_{y_i}$. The outlier ratio, r_o, is then defined as the number of outliers determined in this fashion relative to the total number of data points N:

$$r_o = \frac{N_o}{N} \tag{5.9}$$

In the ideal match between a metric's output and subjective results, $r_p = 1$, $r_s = 1$ and $r_o = 0$.

Table 5.5 provides indicative information about the accuracy, monotonicity and consistency of the SPSNR metric. The accuracy of PSNR, $r_p(0.5)$, is given for reference. Pearson correlation, r_p, and Spearman correlation, r_s, are close to 1 for all sequences. Thus, accuracy and monotonicity of SPSNR are high. Outlier ratio, r_o, is in the vicinity of 10%, so the consistency of the metric is good as well. By comparing the Pearson correlation of SPSNR, $r_p(w_f)$, with the Pearson correlation of PSNR, $r_p(0.5)$, we further note that by taking into account semantics, accuracy is improved by up to 8% (*Akiyo*).

Improvements due to the use of semantics can further be verified in Figure 5.12. In the scatter plot, the SPSNR metric exhibits significantly less outliers than PSNR. I.e., scores predicted using SPSNR (blue dots) are generally closer to the ideal, linear prediction function (blue line) than scores predicted using PSNR (red crosses). This is established by computing the *linear norm of residuals*, $|e|$, for both SPSNR and PSNR. The norm of residuals is a measure of the goodness of fit, where a smaller value indicates a better fit than a larger value. Fit residuals are defined as the difference between the ordinate data point and the resulting fit for each abscissa data point. For SPSNR, we get $|e| = 91.6$, whereas for PSNR, we get $|e| = 105.8$. This indicates that the prediction performance of SPSNR is superior to that of PSNR.

5.4 Adaptation strategy selection

Strategy selection is needed to work out the adaptation strategy that provides most *value* for the end user, considering the individual resources of the connected appliance and network. Value [153] is a subjective measure of fidelity that has been introduced to overcome the difficulty of formulating a

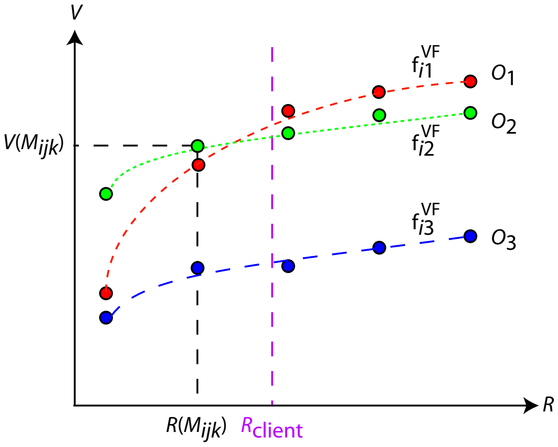


Figure 5.13: Adaptation strategy selection. For some content item A_i, value is plotted as a function of a single resource for three different adaptation operators O_{i1}, O_{i2} and O_{i3}. For each operator, a number of anchor nodes (colored circles) is defined. Also, a polynomial value function $\mathbf{f}_{ij}^{\mathrm{VF}}$ (dashed lines) is fit to the anchor nodes corresponding to each adaptation operator. The optimal adaptation strategy, O_1, is given by the value function that has maximum value at $R = R_{\mathrm{client}}$.

meaningful distortion measure when the adaptation is drastic (e.g., transmoding). For frame-based and object-based adaptation strategies, value can be measured using the objective video quality metric SPSNR that has been introduced in Section 5.3.2.

Specifically, let A_i be some original item, e.g., a video. The adapted version M_{ijk} is computed by transcoding A_i using the adaptation operator O_j at resources k. Each adaptation operator O_j implements an adaptation strategy in Table 5.1. The value of M_{ijk} resulting from the adaptation is denoted by $V(M_{ijk})$. Let us furthermore define the item resource vector for the item M_{ijk} as $\mathbf{R}(M_{ijk}) = \big(R(M_{ijk})^1, R(M_{ijk})^2, \ldots, R(M_{ijk})^r\big)^T$, where r is the number of different resources that have to be considered (e.g., bitrate, resolution, coding format, etc.). Similarly, the client resource vector is denoted by $\mathbf{R}_{\mathrm{client}} = \big(R^1_{\mathrm{client}}, R^2_{\mathrm{client}}, \ldots, R^r_{\mathrm{client}}\big)^T$. The selection of the optimal adaptation strategy can then be formalized by the following resource allocation problem [153]:

Problem 1 *For item A_i, find the adapted version M_{ijk} that has maximum value $V(M_{ijk})$ such that item resources $\mathbf{R}(M_{ijk})$ do not exceed client resources $\mathbf{R}_{\mathrm{client}}$:*

$$\max_{j,k} \Big\{ V(M_{ijk}) \Big\} \quad \textit{such that } R^n(M_{ijk}) \leqslant R^n_{\mathrm{client}} \quad \textit{for all } 1 \leqslant n \leqslant r \tag{5.10}$$

The simplest solution to Problem 1 is to define a number of *anchor nodes* $\big(O_j, \mathbf{R}(M_{ijk}), V(M_{ijk})\big)$. An anchor node expresses the value $V(M_{ijk})$ resulting from applying adaptation operator O_j at resources $\mathbf{R}(M_{ijk})$. The optimal adaptation strategy is then given by the anchor node that satisfies Equation (5.10). However, this solution might be suboptimal when the anchor nodes are unwisely defined. This is illustrated in Figure 5.13. The value (e.g., SPSNR) of some content item A_i is plotted as a function of a single resource (bitrate) for three different adaptation operators O_1, O_2 and O_3. Anchor nodes are represented by colored circles. In the example, the anchor node with $R(M_{ijk}) \leqslant R_{\mathrm{client}}$ that has maximum value corresponds to the operator O_2. However, the optimal adaptation strategy would be O_1, applied at $R(M_{ijk}) = R_{\mathrm{client}}$.

A more accurate solution is to fit a polynomial *value function* (VF) to the anchor nodes of each adaptation operator. The value function matrix for an item A_i is denoted as

$$\mathbf{F}_i^{\mathrm{VF}} = \left(\mathbf{f}_{i1}^{\mathrm{VF}}, \mathbf{f}_{i2}^{\mathrm{VF}}, \ldots, \mathbf{f}_{iJ}^{\mathrm{VF}}\right)^T = \begin{pmatrix} a_{i1,1} & a_{i1,2} & \cdots & a_{i1,p} \\ a_{i2,1} & a_{i2,2} & \cdots & a_{i2,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{iJ,1} & a_{iJ,2} & \cdots & a_{iJ,p} \end{pmatrix}, \tag{5.11}$$

where $a_{ij,k}$ are the coefficients of the order $p-1$ polynomial value function, $\mathbf{f}_{ij}^{\mathrm{VF}}$. J is the number of adaptation operators. The solution to Problem 1 is then given by the VF that has maximum value $\max_{j,\mathbf{R}}\left\{\mathbf{f}_{ij}^{VF}(\mathbf{R})\right\}$ such that $\mathbf{R} \leqslant \mathbf{R}_{\mathrm{client}}$. In the particular case where all VFs are monotonically increasing, the solution is located at $\mathbf{R} = \mathbf{R}_{\mathrm{client}}$. The latter is the case in Figure 5.13. A polynomial function is fit to the anchor nodes corresponding to each adaptation operator. The optimal adaptation strategy, O_1, is given by the value function that has maximum value at $R = R_{\mathrm{client}}$.

In Section 6.3.2, the above selection mechanism is applied on real video sequences to select among different adaptation strategies in realistic content delivery situations. We would like to point out that the discussed method requires value to be computed explicitly for each candidate strategy. Such calculations are time-consuming and need to be performed offline. A solution to this problem is value function prediction [246], where the value is estimated for each strategy based on content features instead of being actually measured.

5.5 Summary

We have defined a framework for adaptive video delivery based on semantic video objects and on their associated metadata. First, a number of complementary adaptation strategies that fit a wide variety of possible appliance resources, network capacities and user preferences have been discussed. In particular, two new adaptation strategies have been proposed. The first strategy combines semantic analysis with a traditional frame-based video encoder. The second strategy uses metadata to efficiently encode the main content message.

Then, the impact of different adaptation strategies has been quantified with subjective experiments. We have established that background alterations resulting from semantic prefiltering do not impair overall quality at low bitrates. We have also demonstrated that the metadata-based representation of object's shape and motion suffices to convey the meaning and action of a scene when the objects are familiar. The former is used to improve coding performance in bandwidth-critical applications such as wireless video delivery for mobile devices. The latter permits to preserve privacy in video surveillance applications as well as to reduce bandwidth requirements under critical network conditions. In addition to the above, we have proposed an objective quality metric, SPSNR, that mimics the behavior of human observers by taking into account semantics. This has enabled us to improve the prediction accuracy by up to 8% with respect to PSNR.

At last, we have proposed a strategy selection mechanism that provides optimal value for the end user by maximizing the adaptation performance under a given set of appliance and network constraints. In Chapter 6.2.2, metadata-based coding enables us to perform automated event detection and to achieve privacy preservation for visual surveillance. Moreover, metadata-enhanced coding is used to put relevant objects in a conspicuous situation for the monitoring personnel. In Appendix A, adaptive video delivery will be used to deliver information to all types of users under a wide variety of conditions in a transparent form.

Results and validation 6

6.1 Introduction

This chapter discusses experimental results obtained with standard test sequences and proposes a validation of our work in real applications. In Section 6.2, the motion-based video object extraction algorithm proposed in Chapter 4 is used to segment and track multiple objects in cluttered background. Based on sample frames and on trajectory graphs, we analyze the behavior of our solution in the presence of non-rigid objects, complex illumination conditions, and track management issues such as appearance and disappearance of objects, occlusions and splitting. We then discuss how semantic video objects and their associated description can be used to provide intelligent visual surveillance. The description of objects' features enables video enhancement in order to put important objects in a conspicuous position for the monitoring personnel. Moreover, descriptors are used for automated event detection.

In Section 6.3, the impact of different video adaptation strategies on the encoding performance of frame-based as well as object-based coders is assessed by means of rate-distortion analysis and by visual inspection. The additional cost of sending metadata for metadata-based and metadata-enhanced encoding is evaluated too. Then, the adaptive delivery framework presented in Chapter 5 is tested with real sequences for different client resource profiles. This experiment shows that our solution is capable of delivering content that matches individual appliance and network resources while providing maximum value for the end user.

6.2 Semantic video analysis

In this section, the motion-based semantic video object extraction algorithm presented in Section 4.3 is tested with real sequences. The videos expound various difficulties that bring the strengths and weaknesses of our solution to the fore. The algorithm is then used to provide intelligent visual surveillance, where semantic video objects and their associated description enable video enhancement and automatic event detection.

6.2.1 Motion-based semantic video object extraction in real sequences

Next, the motion-based semantic video object extraction algorithm in Section 4.3 is applied on real video sequences in order to segment and track multiple objects. With reference to Figure 4.1, the input of the algorithm is a video sequence, and the output is a set of video objects that are coherently labeled over time. The display of the results is organized as follows. First, the results of object segmentation are shown. Object segmentation defines the shape of the moving objects. The computed shape is represented as a mask (color coded in black) superimposed on the original background. Then, the results of object tracking are displayed. Each object is given a label by the tracking algorithm. Each label is coded with a different color for displaying purposes. Finally, the trajectories of the semantic video objects along the frames are reported. We would like to point out that the same set of parameters was used to generate all the results presented in this section.

Object segmentation and tracking results

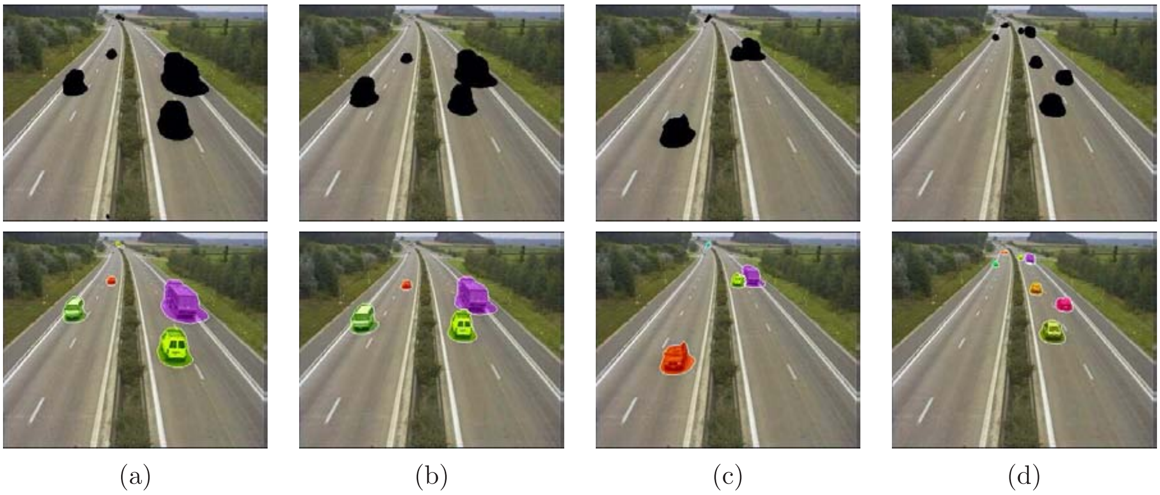


Figure 6.1: Extraction results for the sequence *Highway.* Top: object segmentation results. Bottom: video objects tracked over time.

Figure 6.1 shows sample frames from segmentation and tracking results of the test sequence *Highway* (330 frames, CIF, 25 Hz), from the MPEG–7 Video Content Set. This traffic surveillance sequence represents a highway with vehicles of different sizes driving on four lanes. Here, the goal of tracking is to manage multiple simultaneous objects, their merging, and their appearance and disappearance from the scene. Column (b) shows that the two objects on the right hand side are merged together in the object segmentation mask (top). The tracking algorithm is capable of separating the two objects (bottom) and of providing them with a coherent label over time. This is a consequence of region tracking (Section 4.3.3). Indeed, regions to keep their original label even when they are merged together in the object segmentation mask. Column (c) and column (d) show the status of the tracked vehicles after the van on the left hand side has left the scene. The disappearance of the object does not alter the extraction performance. In fact, each region in the object segmentation mask is processed independently and does not affect the extraction of other video objects. In the same way, all the other objects in the scene are separately tracked along the frames, as shown in Figure 6.8(a).

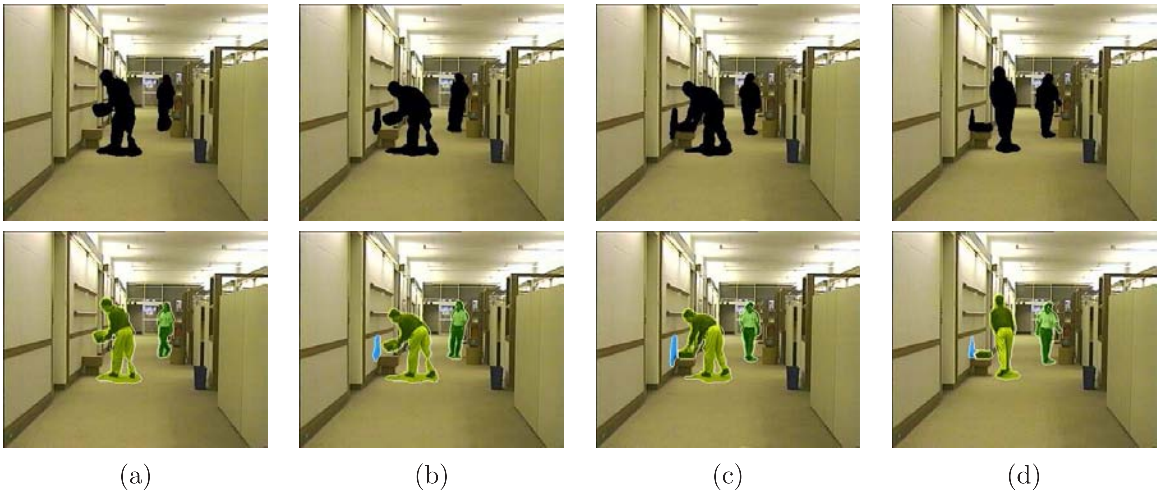


Figure 6.2: Extraction results for the sequence *Hall monitor*.

Results for the sequence *Hall monitor* (300 frames, CIF, 25 Hz), from the MPEG–4 Video Content Set, are shown in Figure 6.2. As opposed to the previous sequence, *Hall monitor* represents an indoor scene with deformable objects. The goal of tracking is to follow the two moving people separately. In this sequence, we want to highlight the behavior of the tracking algorithm in case of errors in object segmentation and in case of track management issues such as splitting. It is possible to notice in column (b) that the man is casting his shadow on the wall. Since no descriptors are projected into the region in the object segmentation mask that corresponds to the shadow, a new track is initiated. The shadow is therefore correctly identified as a new object by the tracking algorithm. The appearance of a new object, the shadow, does not alter the tracking of the man on the left hand side. When the shadow and the man merge in the object segmentation mask (column (c), top), the two objects are kept separated thanks to tracking (column (c), bottom). This allows to overcome the problem introduced by the object segmentation module which wrongly detected the shadow as an object. Another analysis module could be added to the system in order to identify the shadows [197]. Finally, column (d) shows the splitting of the man and of his suitcase on the left hand side. Despite the fact that the suitcase and the man are identified by two separate regions in the object segmentation mask, the suitcase is not interpreted as a new object, and thus correctly keeps the same label as the man.

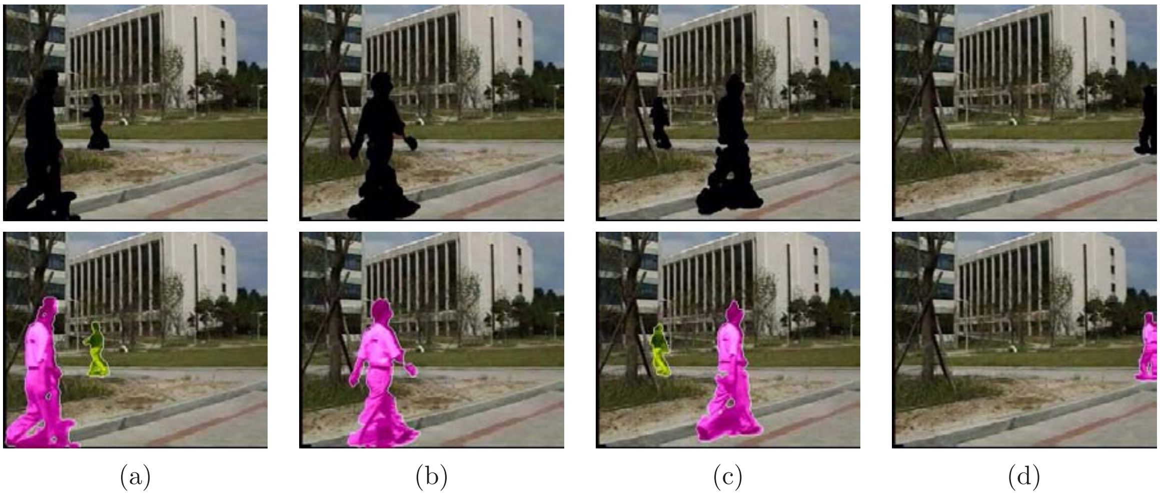


Figure 6.3: Extraction results for the sequence *Walkway*.

Figure 6.3 shows sample frames of the test sequence *Walkway* (290 frames, CIF, 25 Hz), from the MPEG–7 Video Content Set. The video represents two walking people in an outdoor setting. The difficulties of this sequence are the presence of simultaneous non-rigid objects, total occlusion, and the temporary disappearance of objects. In column (b), the smaller person is completely covered by the bigger person, and his trajectory is lost (total occlusion). To relate the reappearing person to his original trajectory (column (c)), the data association step operates not only between subsequent frames, but on a longer temporal window (i.e., 50 frames). Later, the bigger person is leaving the scene for about one second. As for total occlusion, the person's identity is recovered when he reappears in column (d).

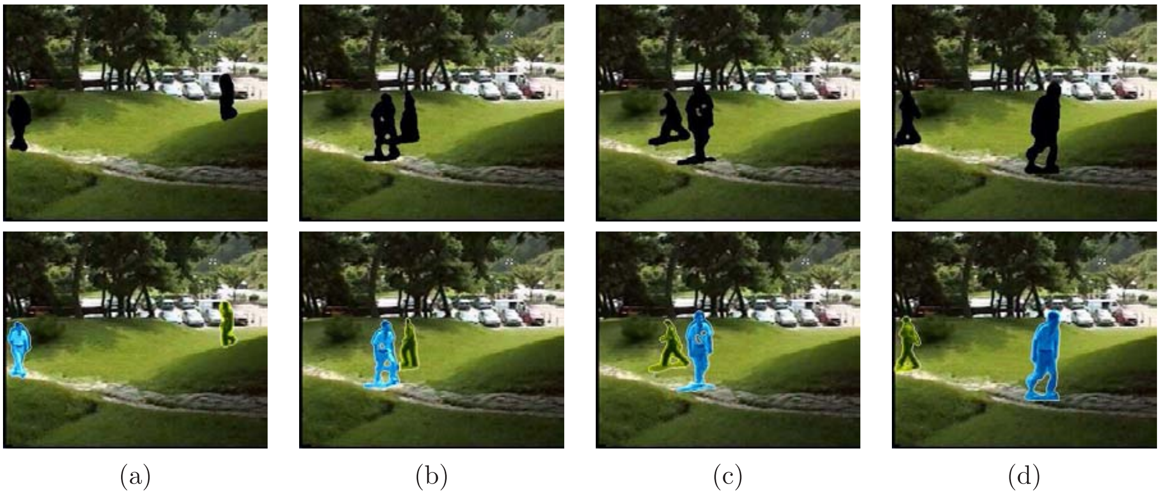


Figure 6.4: Extraction results for the sequence *Surveillance*.

Total occlusion and object deformations also take place in the sequence *Surveillance* (240 frames, CIF, 25 Hz), from the MPEG–7 Video Content Set. Corresponding results are shown in Figure 6.4. Both persons are tracked correctly along the frames, despite the total occlusion of the person

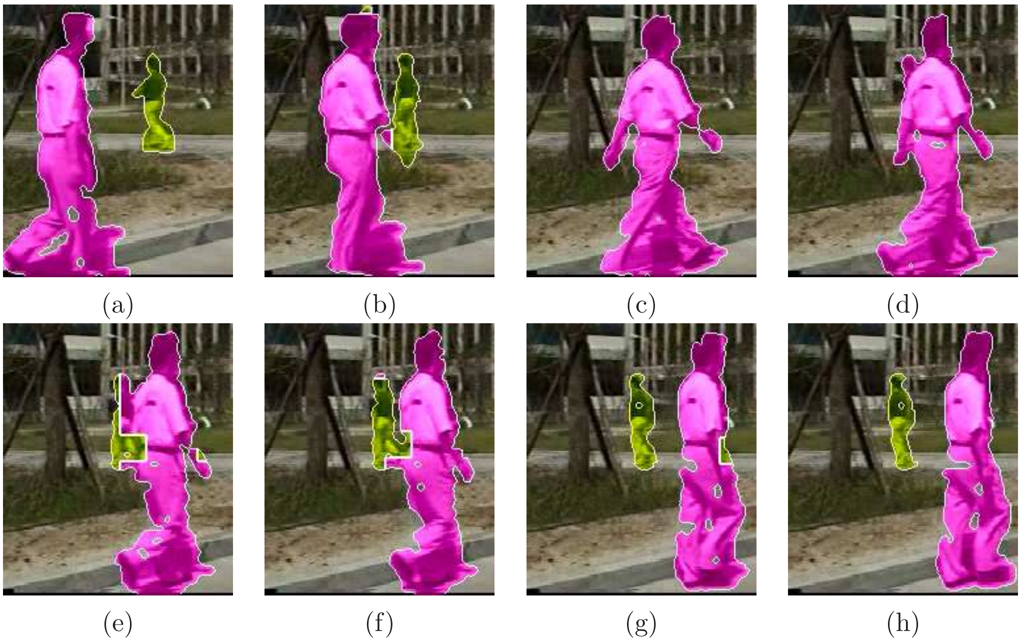


Figure 6.5: Total occlusion handling by the tracking algorithm for the sequence *Walkway*.

walking from the right to the left by the other person. It is important to notice that even when the segmentation mask does not separate the two persons (column (b), top), the tracking algorithm keeps their individual identities (column (b), bottom).

The behavior of the tracking algorithm in the presence of total occlusions is further analyzed in Figure 6.5. The figure shows a zoom from the tracking results of the sequence *Walkway*. In frame (a), each person belongs to a separate region in the object segmentation mask and has a distinct label. In frame (b), the two persons are merged together in the object segmentation mask, but the tracking algorithm is capable of separating both objects. In frame (c), the smaller person is totally occluded by the bigger one, and his trajectory is lost. When the head of the occluded person reappears in frame (d), his trajectory is not yet recovered. The trajectory is recovered in frames (e) and (f), where regions belonging to the occluded person are correctly identified and labeled. In frames (g) and (f) at last, each person belongs again to a separate region in the object segmentation mask and is labeled correctly. Note that in frame (g), one region of the bigger person is incorrectly labeled. This has a negative impact on the computation of the object's trajectory, as discussed later in the text.

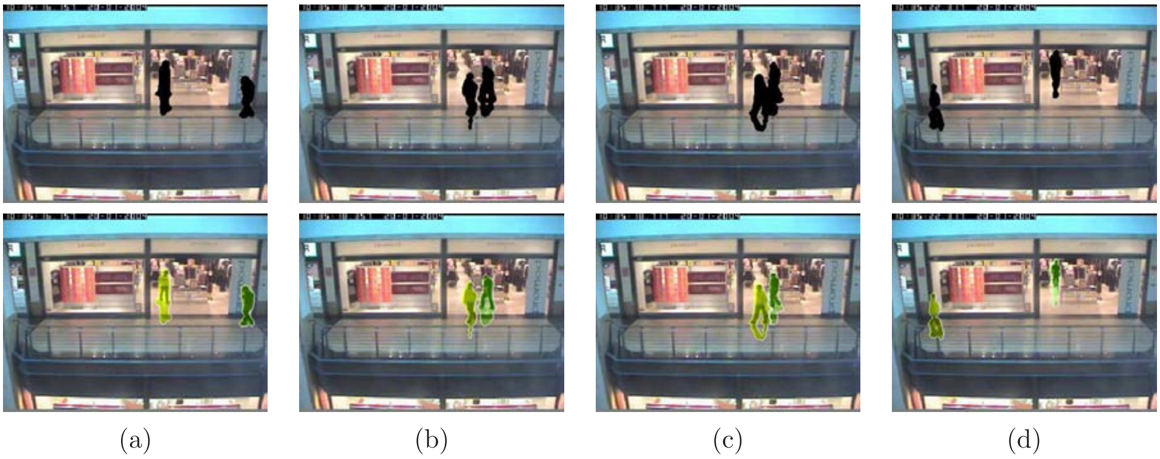


Figure 6.6: Extraction results for the sequence *Caviar*.

The behavior of video object extraction in the presence of complex illumination conditions is analyzed in Figure 6.6. The sample frames are from the test sequence 'EnterExitCrossingPaths1front' (180 frames, 384 × 288, 25Hz), from the EC-funded project *Caviar*. The video represents an indoor scene with several reflective surfaces. The change detector erroneously includes image parts corresponding to reflections into the object segmentation mask (top). Nevertheless, the tracking algorithm is capable of putting the reflections in correspondence with the reflected persons. This is the case even when both persons are merged in the same region of the object segmentation mask, as shown in column (d), bottom.

Finally, we further analyze the behavior of the proposed tracking algorithm in case of errors in the object segmentation results. Figure 6.7 shows a zoom from the sequence *Surveillance*. The segmentation mask (top) does not define the shape of the person correctly. In particular (columns

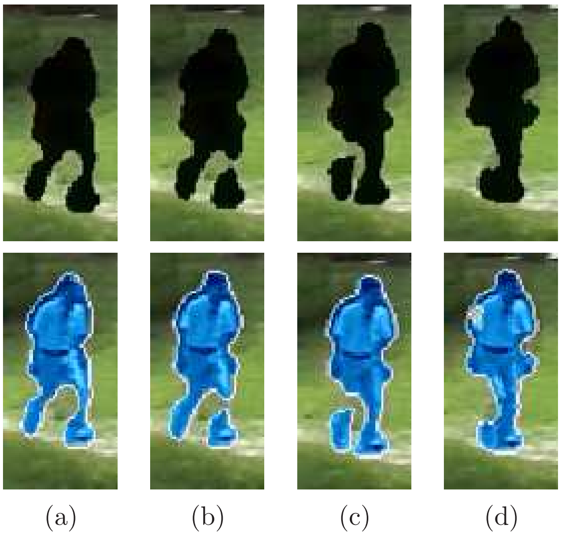


Figure 6.7: Example of robustness of the proposed tracking algorithm in case of errors in the object segmentation module.

(b) and (c)), a leg of the man is identified by a set of pixels which is not connected to the rest of the body. Instead of initiating a new track for the unconnected part, the projection of the regions allows one to keep the track of the full object, thus recovering the identity when the segmentation is correct (column (d)). The interactions between the region partition and the object partition help in overcoming this problem, and the objects are correctly tracked.

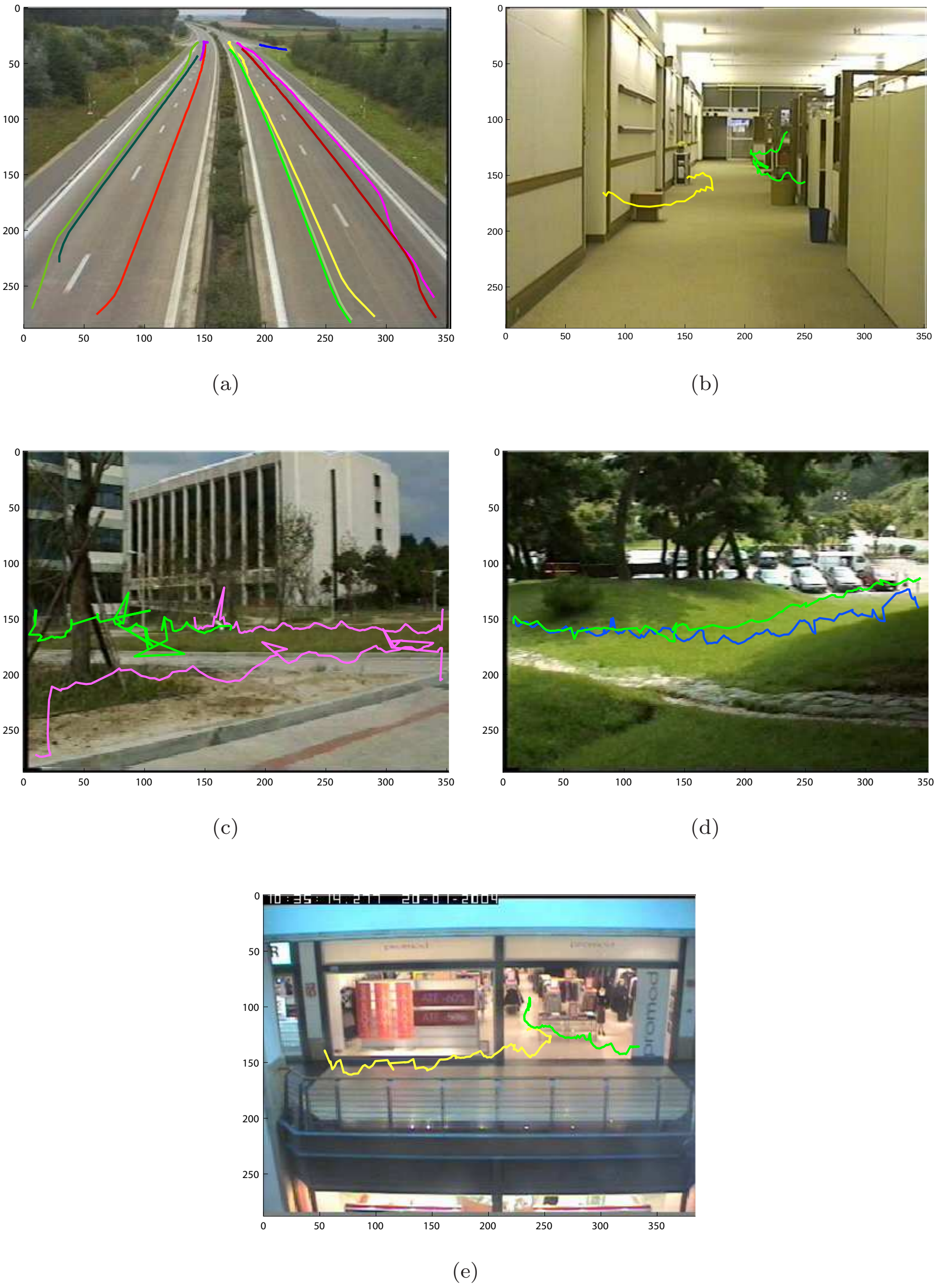
Object trajectories

The trajectories of the semantic video objects in the test sequences are plotted in Figure 6.8. The trajectories are given by the position of video objects' gravity centers along the frames. Results for the sequence *Highway* are displayed in Figure 6.8(a). The algorithm is capable of tracking objects separately even when they belong to the same region in the object segmentation mask (e.g., violet and green trajectories on the right lane). Observe that all trajectories on top of the frame are terminated before the cars reach the horizon. This is because video objects whose size is inferior to 20 pixels are suppressed in the post-processing stage, in order to limit the influence of noise on the tracking process. In Figure 6.8(b), the trajectories of both persons in *Hall monitor* are displayed. It is useful to point out that in order to obtain the trajectories of moving people, one might use the center of video objects' lower boundaries instead of their gravity centers. This might improve the readability of the trajectories, but it does also increase the impact of possible segmentation errors on the trajectories (e.g., erroneous inclusion of shadows). Results for *Walkway* are shown in Figure 6.8(c). The green trajectory is not interrupted when the corresponding person is totally occluded by the other person. However, the trajectory becomes more erratic during the occlusion. This is due to the erroneous inclusion of individual regions from one objects into the other object, as shown in Figure 6.5(g). As a consequence, the object's gravity center is displaced. At last, object trajectories from the sequences *Surveillance* and *Caviar* are displayed in Figure 6.8(d) and in Figure 6.8(e), respectively. These trajectories are not affected by occlusions or merging objects.

Next, we discuss how semantic video objects and their associated description can be used by a content understanding step that monitors the behavior of objects in the scene. In visual surveillance applications, this information helps the content understanding module in describing events in the scene and in generating alarms in the event of dangerous situations. Furthermore, the description of objects' features enables video enhancement in order to put important objects in a conspicuous position for the monitoring personnel.

6.2.2 Visual surveillance applications

The problem of remote visual surveillance of unattended environments has received growing attention in recent years. Nowadays, applications include monitoring of indoor and outdoor environments, quality control in industrial applications, and military applications. However, event monitoring by human operators is rather boring, tedious and error-prone. Thus, intelligent surveillance systems aim at employing video analysis to automatically select, enhance and interpret visual information [96]. Several approaches make use of artificial intelligence for incident detection [254], activity recognition [170], and personal identification [244]. These methods are usually limited to specific situations due to the use of machine learning. Selective enhancement is another interesting feature which highlights important image regions (e.g., moving objects) by using visual markers [23], or by selective coding (i.e., important regions are coded at a higher quality than the background [25]). However, semantic information extracted by video analysis is not available individually at the receiver's side. In order to make up for this drawback, recent approaches summarize semantics in a content description. This can be used alone or in conjunction with the coded video [49].



(a)

(b)

(c)

(d)

(e)

Figure 6.8: Trajectories of semantic video objects. The trajectories are given by the position of video objects' gravity centers along the frames. (a) *Highway*. (b) *Hall monitor*. (c) *Walkway*. (d) *Surveillance*. (e) *Caviar*.

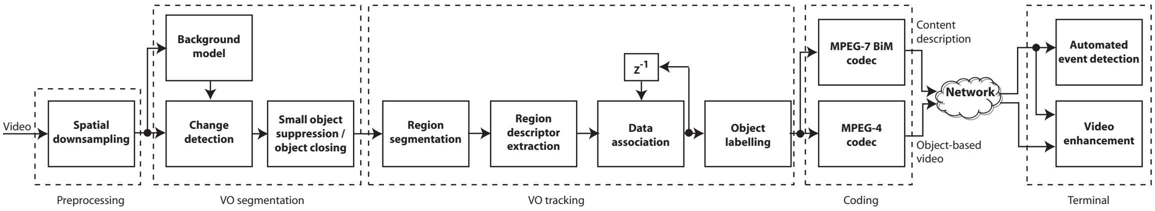


Figure 6.9: Block diagram of the system for real-time generation of annotated video.

In this section, we discuss how semantic video analysis can be used to generate annotated video in real-time [214]. In the block diagram in Figure 6.9, semantic video objects are first extracted by means of motion-based video object segmentation and tracking, and then coded with MPEG–4. Moreover, an MPEG–7 description of object features is generated. In surveillance, the description enables video enhancement in order to put important objects in a conspicuous position for the monitoring personnel. Moreover, descriptors are used for automated event detection. Also, the description can be stored in a database for further processing (video indexing). Our solution operates in cluttered environment and enables interoperability with third-party applications by making use of MPEG standards for video coding and description. While in similar work [49] content annotation summarizes *events* detected by means of artificial intelligence, the proposed MPEG–7 description captures high-level object features. Therefore, our solution is not bound to any particular setup.

Real-time video object extraction and description

In the motion-based semantic video object extraction algorithm presented in Section 4.3, a statistical change detection process produces the segmentation of moving objects from the background. The subsequent tracking mechanism relies on feedbacks between an object partition and a region partition to follow multiple, simultaneous objects along the frames. The region partition is generated by a clustering method based on spatially unconstrained fuzzy C-means (FCM). This solution produces regions that correspond to homogeneous areas of the objects. However, the computational complexities of the object segmentation process and of the region segmentation stage do not enable us to achieve real-time performance.

The effectiveness of visual surveillance systems however depends crucially on their short reaction time. Thus, real-time performance is essential. In order to achieve real-time video object extraction, the algorithm in Section 4.3 has been modified as follows:

- *Preprocessing.* High resolution input signals (e.g., from a digital video camera) are downsampled to CIF resolution in order to limit the complexity of the change detection process. The complexity is proportional to the input resolution, since change detection operates on each individual image pixel to produce the object partition.

- *VO segmentation.* Instead of the statistical change detector, an RGB change detector is used to segment moving objects from the background. The pixel-wise difference between each color channel of the input frame and of the background model is thresholded to produce the object partition Π_o^n at frame n. Π_o^n is further regularized by eliminating small connected sets of pixels, and by suppressing small holes using morphology. Depending of the scene's illumination and contrast properties, RGB change detection requires manual threshold adjustments along a sequence or for different sequences. Nevertheless, this simple method is capable of producing reliable results with indoor and outdoor surveillance video, as shown in Figure 6.11.

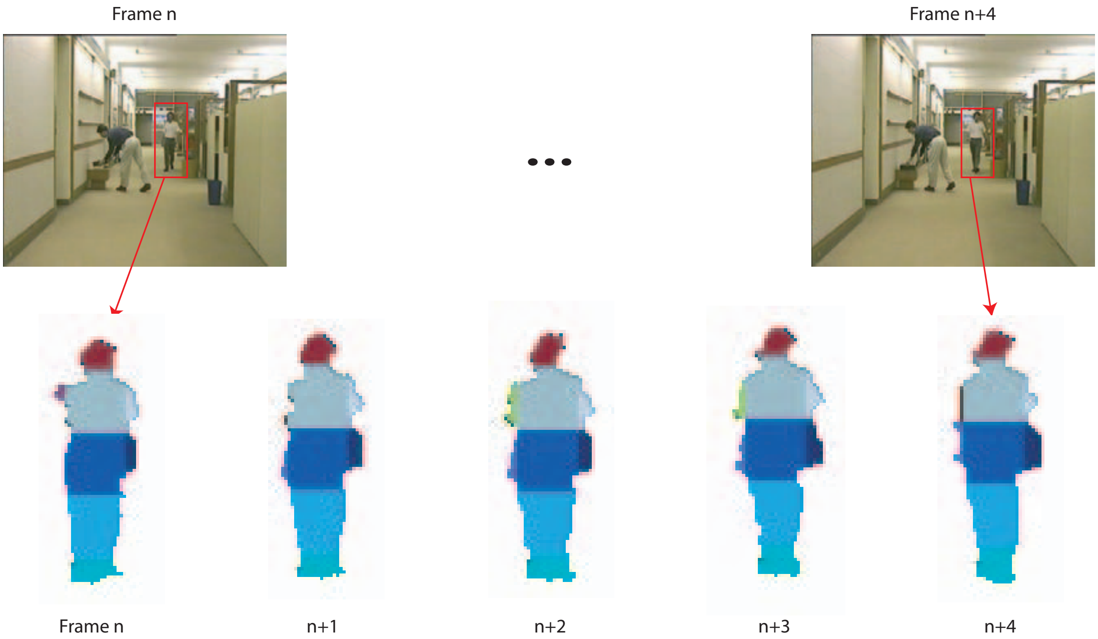


Figure 6.10: Example of region segmentation using a "chessboard" grid. An individual grid is used for each video object, which is centered on the object's gravity center. The size of individual squares has been set to 20 × 20 pixels.

- *Region segmentation.* In place of the fuzzy C-means clustering, a "chessboard" grid is used to define the object partition. An individual grid is used for each video object, which is centered on the object's gravity center. The size of the squares is selected by the user and should be small enough to permit multiple regions per object. In our experiments, a size of 20 × 20 pixels has been used. With that region segmentation method, the tracking performance is degraded in the presence of severe object deformations. However, inaccuracies resulting from meaningless regions are mostly compensated for by the data association process (Section 4.3.4). An example of region segmentation using a chessboard grid is shown in Figure 6.10. A similar example using fuzzy C-means clustering is given in Figure 4.4.

After video object extraction, the video object's location, shape and color are summarized by MPEG–7 *Visual* Descriptors [110]. In Table 6.1, `Motion trajectory` gives the spatial location of objects (e.g., gravity center). `Region locator` approximates the shape of objects by their bounding box. Accurate shape is given by `contour shape`. `Dominant color` at last defines salient object colors. The descriptors are organized so as to provide a layered description of the scene [212, 213]. That is, location, shape and color can be defined in any desired combination and order.

Feature	descriptor	Purpose
Location	`Motion trajectory`	Spatial location
Shape	`Region locator`	Bounding box
	`Contour shape`	Closed contour shape
Color	`Dominant color`	Salient color

Table 6.1: MPEG–7 Descriptors for object features.

Next, the performance of the system in Figure 6.9 is demonstrated using four surveillance videos: (1) *Surveillance*, from the MPEG–7 Video Content Set; (2) *Hall monitor*, from the MPEG–4 Video Content Set; (3) *PETS'2000*; (4) *PETS'2001*, from the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. In our experiments, the sequences were processed in real-time (25 frames/s) on a 2.8 GHz Pentium 4 PC. The MoMuSys MPEG-4 VM reference software version 1.0 video encoder is used. Binary MPEG–7 is generated by the Expway MPEG-7 *BiM* Payload encoder/decoder version 02/11/07.

Video enhancement

Video enhancement is illustrated in Figure 6.11. In the left part, moving objects are enhanced by their bounding box. Boxes are defined by MPEG–7 `Region locator` and rendered by the terminal. Video enhancement helps lowering fatigue of the monitoring personnel that is due to extended concentration, since relevant objects are put in a conspicuous position. By comparison with the common approach that consists in producing enhanced video at the encoder's side, enhancement by the receiver requires only low additional resources for transmission and provides additional flexibility. For instance, the receiver might switch amongst available features or enhance individual objects. In the right part of Figure 6.11, `contour shape` is rendered on a static background shot. With this representation, the identity of moving objects (e.g., people) remains hidden, thereby enabling privacy preservation. Also, the representation of object's shape suffices to convey the meaning and action of a scene when the objects are familiar (Section 5.3.1).

Description	**Location**	**Box**	**Shape**
Surveillance	19 Kbit/s	52 Kbit/s	87 Kbit/s
Hall monitor	16 Kbit/s	45 Kbit/s	98 Kbit/s
PETS'2000	18 Kbit/s	49 Kbit/s	85 Kbit/s

Table 6.2: Average bitrate required to transmit MPEG–7 *BiM* description at different levels of details.

The average bitrate for binary MPEG–7 description at different levels of details is shown in Table 6.2. As compared to medium-quality MPEG–4 video coding (500 Kbit/s), these figures are low. The cost for MPEG–7 encoding can further be reduced by updating description features only in frames where significant changes arise (e.g., large object deformation).

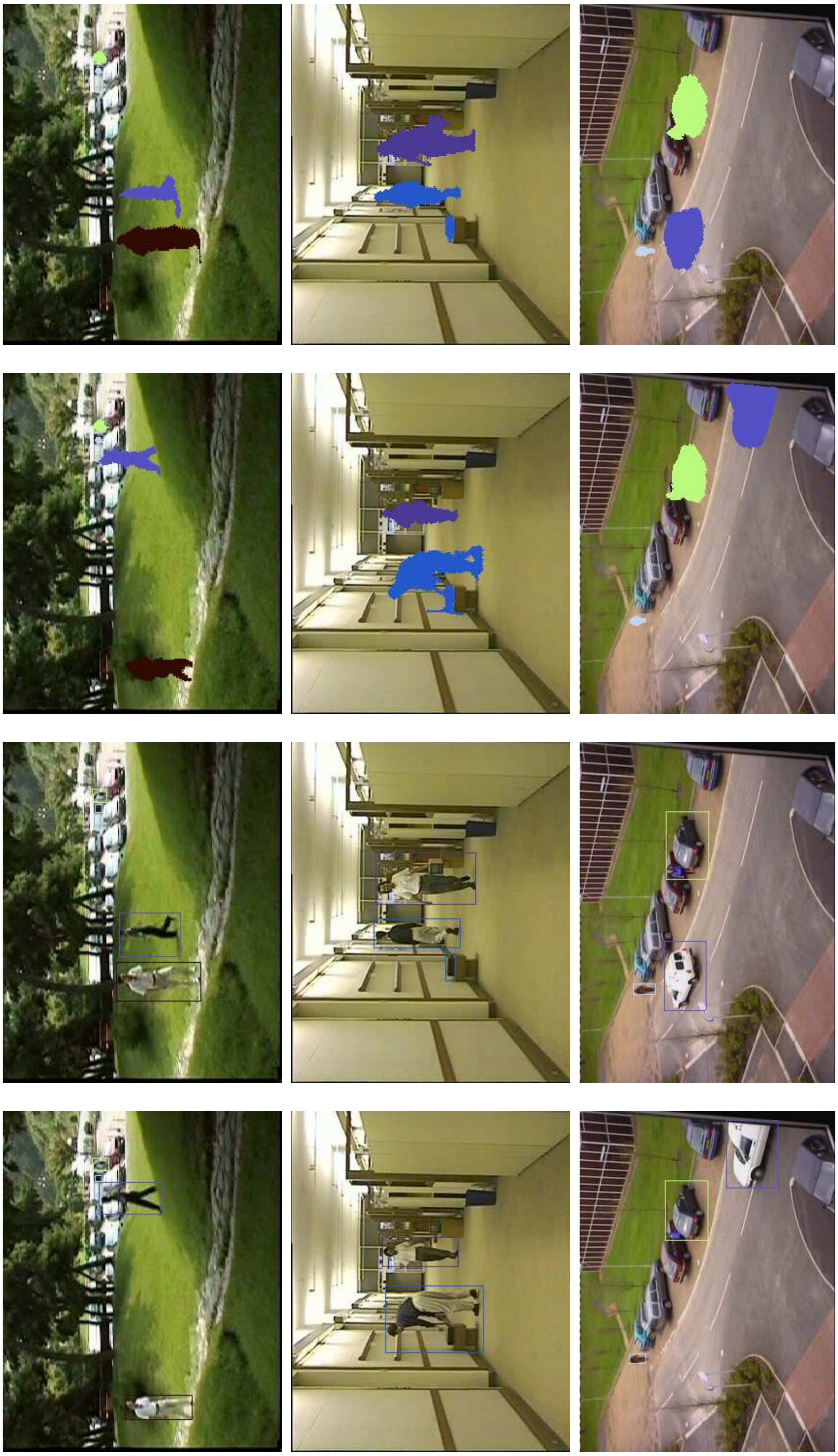


Figure 6.11: Enhancement of surveillance video by MPEG–7 descriptors. (Left) Moving objects are highlighted by their bounding box. (Right) The contour shape of objects is rendered on a static background. (a) *Surveillance*. (b) *Hall monitor*. (c) *PETS'2000*.

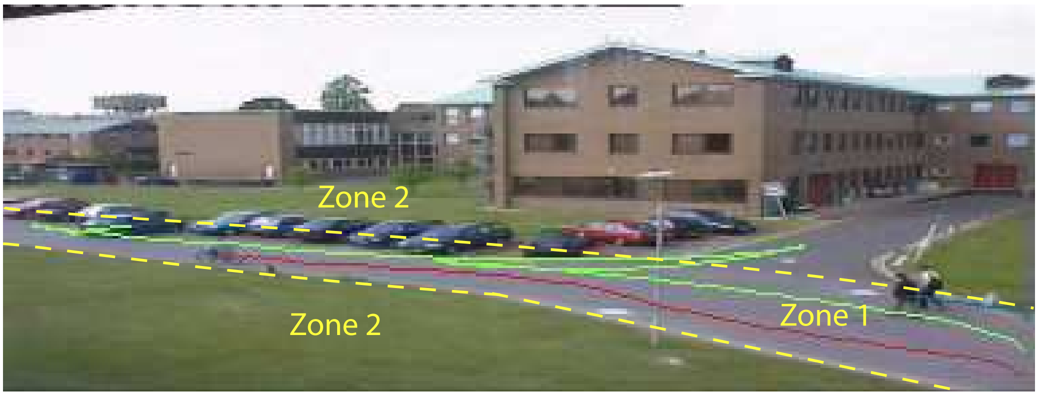


Figure 6.12: Automated event detection. The setup in *PETS'2001* has been subdivided in two distinct zones. Each time the trajectory of an object (solid lines) enters Zone 2 for more than one second, an intrusion alarm is generated.

Automated event detection

To perform automated event detection, the setup in *PETS'2001* has been divided into two distinct zones, as shown in Figure 6.12. Zone 1 corresponds to the authorized area, whereas Zone 2 is restricted. The goal is to automatically generate an intrusion alarm each time an object enters Zone 2. To arrive at this, the location of object's gravity center in successive frames is described by `Motion trajectory`. When an object enters Zone 2 for more than one second, an alarm is generated. To evaluate detection performance, we compare the number of automatic alarms to groundtruth. For the entire sequence (1780 frames; 10 moving objects), the system has generated three correct alarms, zero false alarms, and zero misses. This simple experiment illustrates how automated event detection is achieved by taking advantage of the physical scene description provided by MPEG–7.

By using a calibrated camera, these results could be complemented so as to provide the trajectories in the 3-D scene. Furthermore, the knowledge of the path of each object within a video sequence permits interactive applications such as video-based hyperlinks, video editing, enhancement of relevant objects, and object-based indexing.

Despite its simplicity, the presented visual surveillance application applies to various situations and can be extended in several ways. Additional functionality for privacy preservation is provided by scrambling image regions corresponding to moving objects. More object features and an object recognition step might be considered for automated event detection. At last, video enhancement can be used to highlight small objects, such as a football, in sports broadcasting. This *metadata-enhanced encoding* approach has already been discussed in the context of adaptive video delivery in Chapter 5. Results and validation experiments for adaptive video delivery are reported next.

6.3 Adaptive delivery

In this section, the impact of different video adaptation strategies on the encoding performance of frame-based as well as object-based coders is assessed by means of rate-distortion analysis and by visual inspection. The additional cost of sending metadata for metadata-based and metadata-enhanced encoding is evaluated too. Then, the adaptive delivery framework proposed in Chapter 5 is tested with real sequences for different client resource profiles. The results show that our solution is capable of delivering content that matches individual appliance and network resources while providing maximum value for the end user.

6.3.1 Evaluation of different adaptation strategies

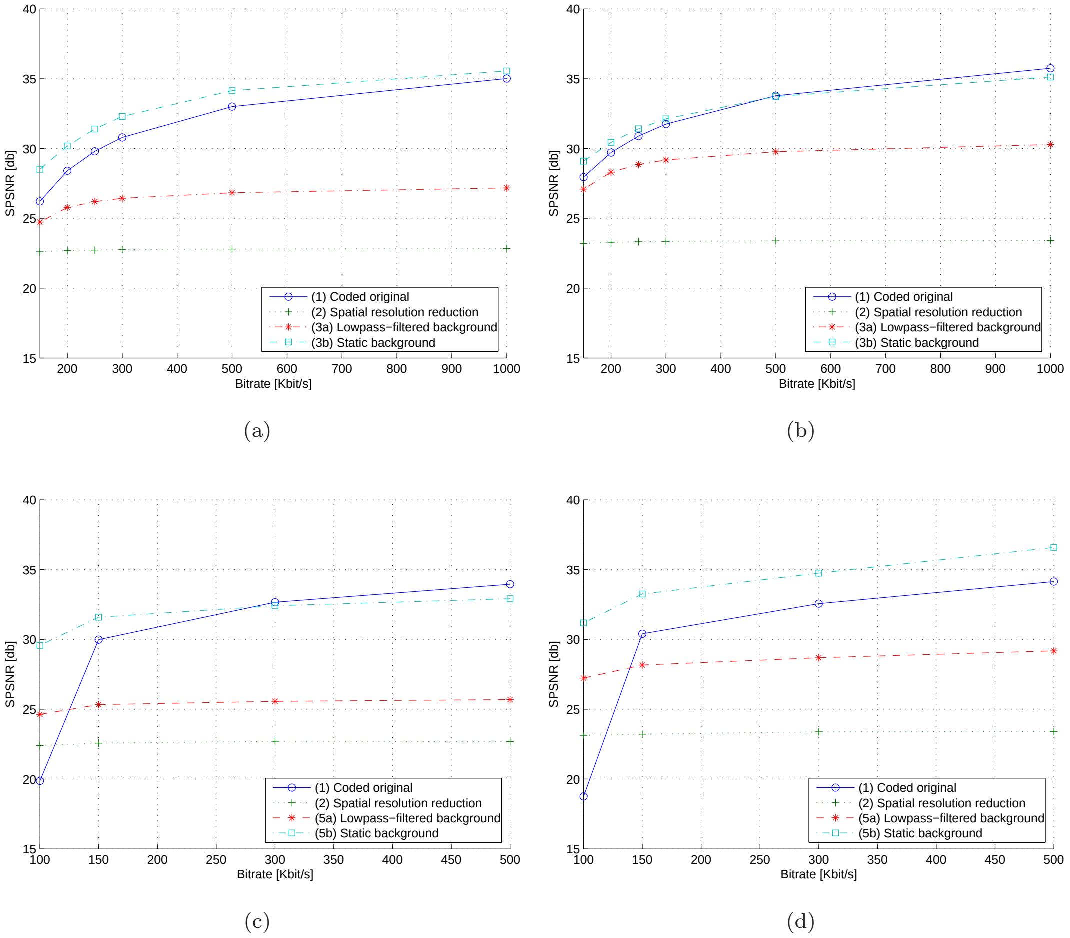
Next, the performances of the adaptation strategies presented in Section 5.2 are compared by means of objective quality evaluation using SPSNR (Section 5.3.2) and by visual inspection. The cost of sending metadata for metadata-based and metadata-enhanced encoding is evaluated as well. Sample results are shown from the MPEG–4 test sequence *Hall monitor* and from the MPEG–7 test sequence *Highway*. Both sequences are in CIF format at 25 Hz. With respect to Table 5.1, the adaptation strategies under analysis are: (1) coded original sequence; (2) spatial resolution reduction (from CIF to QCIF); (3a,5a) semantic prefiltering with lowpass-filtering; (3b,5b) semantic prefiltering with static background. In the former case, the background is simplified using a Gaussian 9×9 lowpass filter with $\mu = 0$ and $\sigma = 2$. Semantic video analysis is carried out using the motion-based video object extraction algorithm in Section 4.3.

The following coders have been used in the encoding process: (i) TMPGEnc 2.521.58.169 using constant bitrate (CBR) rate control for frame-based MPEG–1; (ii) MoMuSys MPEG–4 VM reference software version 1.0 using VM5+ global rate control for object-based MPEG–4; (iii) Expway MPEG–7 *BiM* Payload encoder/decoder version 02/11/07 for MPEG–7 metadata. Coding bitrates are chosen so as to range from the lowest bitrate supported by the codec, up to perceptually lossless coding. The value of the foreground weight used in the objective evaluation is computed for each frame using Equation (5.6). The average weights are $w_f = 0.55$ for *Hall monitor* ($r = 0.04$, $\sigma_b = 54$, $v = 0$), and $w_f = 0.53$ for *Highway* ($r = 0.07$, $\sigma_b = 48$, $v = 0$).

Rate-distortion analysis

Figure 6.13 shows the rate-distortion diagrams for the test sequences. The average SPSNR (Equation (5.5)) for four adaptation strategies is plotted against the encoding bitrate. Figure 6.13 (a) and (b) show the rate-distortion diagrams for MPEG–1 at bitrates between 150 Kbit/s and 1000 Kbit/s. At low bitrates (150-300 Kbit/s), semantic encoding with static background (3b) leads to a larger SPSNR than the content-blind methods (1-2). This is because inter-coded static background blocks do not produce residue and most of the available bitrate can be allocated to foreground objects. The same can be observed in the temporal analysis in Figure 6.14. At 150 Kbit/s, the objective quality of semantic encoding with static background (3b) is always larger than that of the coded original sequence (1). In Figure 6.13 (c) and (d), foreground and background parts are encoded in two separate streams using object-based MPEG–4 at bitrates between 100 Kbit/s and 500 Kbit/s. Here, semantic analysis is used by all four adaptation strategies. It is possible to notice that quality is improved at low bitrates by lowpass filtering the background (5a) or by using a still frame representing the background (5b).

These improvements can further be verified by visual inspection. Figure 6.15 shows a sample frame coded with MPEG–1 at 150 Kbit/s, with and without semantic prefiltering. Figure 6.16 shows magnified excepts of both test sequences coded with MPEG–1 at 150 Kbit/s. Figure 6.16 (top) shows the person that carries a monitor in *Hall monitor*. The amount of coding artifacts is notably reduced by semantic prefiltering ((c) and (d)). In particular, the person's mouth and the monitor are visible in (d), whereas they are corrupted by coding artifacts with the non-semantic strategies. Similar observations can be made for Figure 6.16 (bottom), which shows a blue truck entering the scene at the beginning of the *Highway* sequence. Coding artifacts are less disturbing on the object in (c) and (d) than in (a) and (b). Moreover, the front-left wheel of the truck is only visible with semantic prefiltering ((c) and (d)). Next, we evaluate the cost of sending metadata for metadata-based and metadata-enhanced encoding.



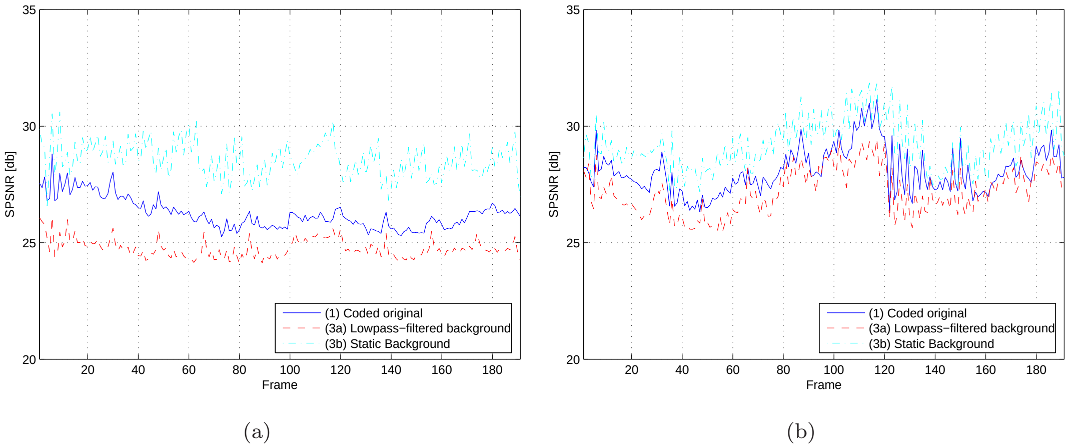
(a)

(b)

(c)

(d)

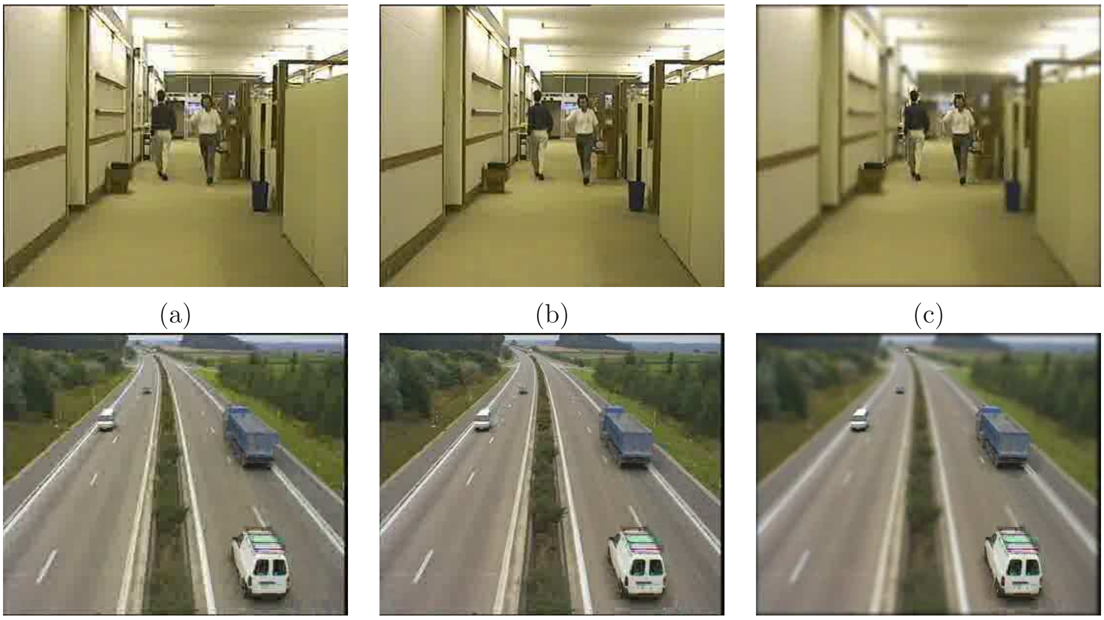
Figure 6.13: Rate-distortion diagrams. (a) *Hall monitor*, MPEG–1. (b) *Highway*, MPEG–1. (c) *Hall monitor*, MPEG–4 object-based. (d) *Highway*, MPEG–4 object-based.



(a)

(b)

Figure 6.14: SPSNR as a function time for test sequences coded with MPEG–1 at 150 Kbit/s, *with* and *without* semantic prefiltering. (a) *Hall monitor*. (b) *Highway*.



(a) (b) (c)

Figure 6.15: Frame 190 of *Hall monitor* (top) and frame 44 of *Highway* (bottom) coded with MPEG–1 at 150 Kbit/s using different adaptation strategies. (a) Coded original sequence. (b) Static background. (c) Lowpass-filtered background.

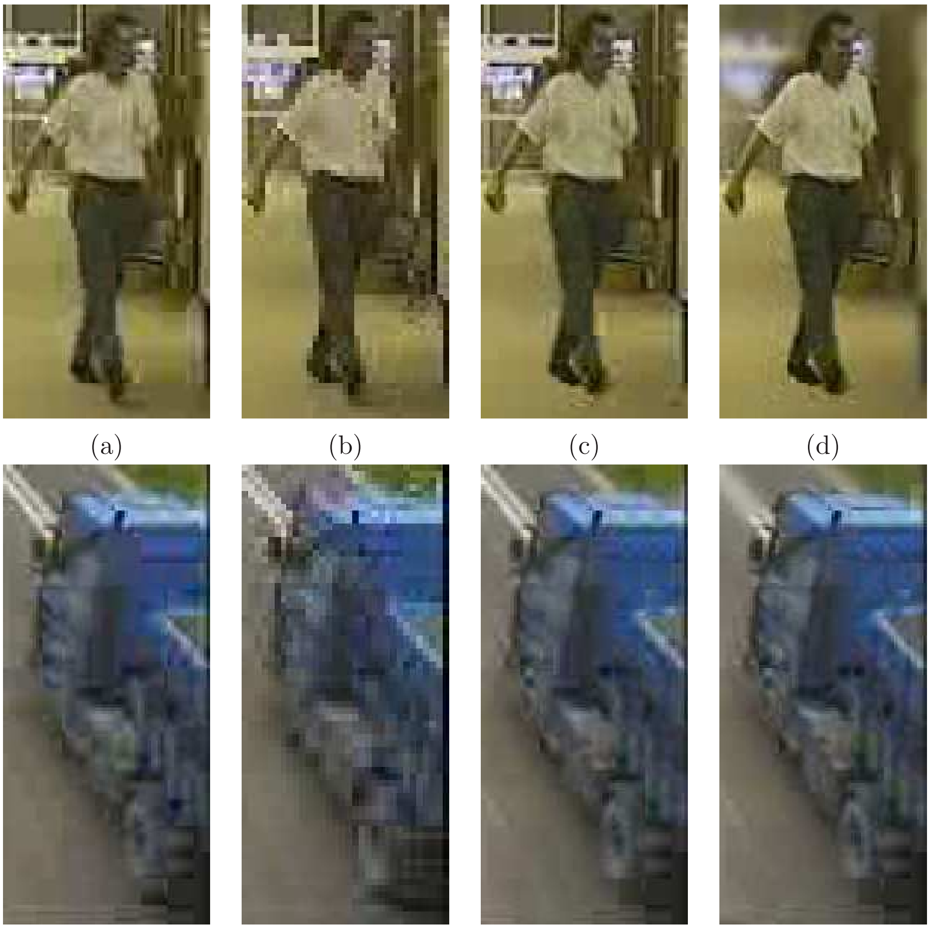
Cost of sending metadata

Table 6.3 shows the bitrate required by three types of description for *Hall monitor* and *Highway* using MPEG–7 binary format (*BiM*). MPEG–7 binary format is used for sending summary information to terminals with limited capabilities and to enhance heavily compressed videos. The descriptions are represented by the spatial locators of the foreground objects, their bounding boxes, and an approximation of their shape with 30-sided polygons, respectively. The metadata size increases with the description complexity and with the number of objects in the scene (*Hall monitor* versus *Highway*). However, the cost for metadata-enhanced encoding can be reduced by sending the description of critical objects only.

Description	**Location**	**Bounding box**	**Polygon shape**
Hall monitor	21 Kbit/s	59 Kbit/s	94 Kbit/s
Highway	26 Kbit/s	66 Kbit/s	101 Kbit/s

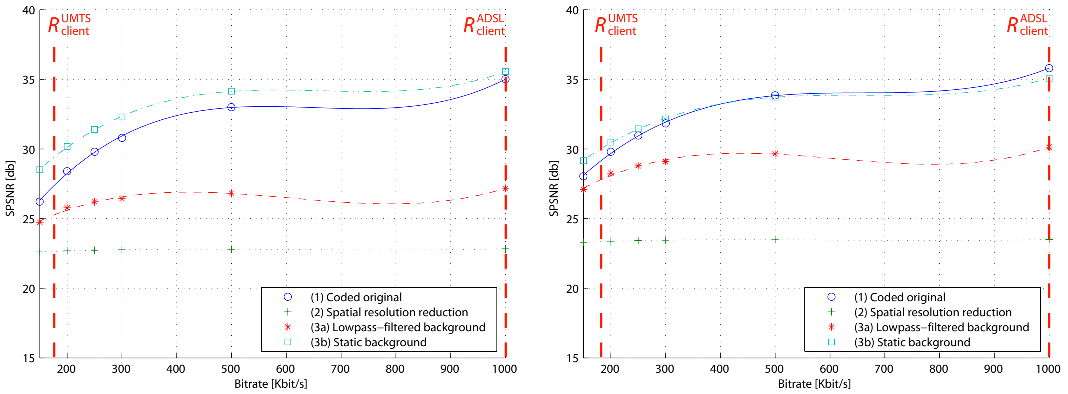
Table 6.3: Average bitrate of MPEG–7 *BiM* sequence description.

In the following, the adaptation strategies that have been evaluated in this section are used in order to provide adaptive video delivery.



(a) (b) (c) (d)

Figure 6.16: Details of frame 280 of *Hall monitor* (top) and frame 16 of *Highway* (bottom). The sequences are encoded with MPEG–1 at 150 Kbit/s using different adaptation strategies. (a) Coded original sequence. (b) Spatial resolution reduction. (c) Static background. (d) Lowpass-filtered background.



(a)

(b)

Figure 6.17: Rate-distortion diagrams for strategy selection. Anchor nodes are represented along with the corresponding cubic polynomial value functions. The client resources under analysis are highlighted using vertical lines. (a) *Hall monitor*. (b) *Highway*.

6.3.2 Adaptive delivery of real sequences

In this section, the adaptive delivery framework proposed in Chapter 5 is tested with real sequences for realistic client resource profiles. In particular, the mechanism discussed in Section 5.4 is applied to the selection of the adaptation strategy that provides most value for the end user. The results are verified by visual inspection and by objective quality evaluation using SPSNR.

In our experiments, the following frame-based adaptation strategies are compared: (1) coded original sequence; (2) spatial resolution reduction; (3a) semantic prefiltering with lowpass-filtering; (3b) semantic prefiltering with static background. A single resource, i.e. bitrate, is considered. In order to assess the performance of the selection mechanism both for low-quality and for high-quality video, the bitrate of the client is set to $R_{\text{client}}^{\text{UMTS}} = 176$ Kbit/s and to $R_{\text{client}}^{\text{ADSL}} = 1000$ Kbit/s. The former corresponds to the bandwidth supported by the UMTS multimedia protocol. The latter is sometimes used for video streaming over asymmetric digital subscriber lines (ADSL).

Definition of the resource allocation problem

To solve the resource allocation problem defined in Section 5.4, we first set a number of anchor nodes. These nodes are located at the following bitrates: 150, 200, 250, 300, 500 and 1000 Kbit/s. In the rate-distortion diagrams in Figure 6.17, each data point represents one anchor node.

Then, a polynomial value function is fit to the anchor nodes of each adaptation strategy. In our experiments, we have found cubic value functions to lead to optimal results. For *Hall monitor*, the value function matrix (Equation (5.11)) is given by

$$\mathbf{F}_{\text{Hall}}^{\text{VF}} = \begin{pmatrix} 17.74 & 7.30 \cdot 10^{-2} & -1.14 \cdot 10^{-4} & 5.86 \cdot 10^{-8} \\ 22.28 & 2.87 \cdot 10^{-3} & -5.03 \cdot 10^{-6} & 2.70 \cdot 10^{-9} \\ 21.02 & 3.38 \cdot 10^{-2} & -6.14 \cdot 10^{-5} & 3.37 \cdot 10^{-8} \\ 21.63 & 5.85 \cdot 10^{-2} & -8.95 \cdot 10^{-5} & 4.49 \cdot 10^{-8} \end{pmatrix}. \tag{6.1}$$

For *highway*, the value function matrix is

$$\mathbf{F}^{\mathrm{VF}}_{\mathrm{Highway}} = \begin{pmatrix} 121.25 & 3.78 \cdot 10^{-2} & -8.74 \cdot 10^{-5} & 4.41 \cdot 10^{-8} \\ 22.98 & 2.86 \cdot 10^{-3} & -5.10 \cdot 10^{-6} & 2.76 \cdot 10^{-9} \\ 22.83 & 3.84 \cdot 10^{-2} & -6.82 \cdot 10^{-5} & 3.71 \cdot 10^{-8} \\ 23.80 & 4.53 \cdot 10^{-2} & -6.81 \cdot 10^{-5} & 3.40 \cdot 10^{-8} \end{pmatrix}. \tag{6.2}$$

For *Hall monitor*, evaluating the value function at $R_{\mathrm{client}} \leqslant 176$ Kbit/s leads to the following maximal SPSNR: 27.4 dB for coded original (1); 22.6 dB for spatial resolution reduction (2); 25.3 dB for semantic prefiltering with lowpass-filtering (3a); 29.4 dB for semantic prefiltering with static background (3b). Thus, according to Equation (5.10), the adaptation strategy that provides most value for the end user is semantic prefiltering with static background (3b). At $R_{\mathrm{client}} \leqslant 1000$ Kbit/s, the maximal SPSNR is: 35 dB for the coded original; 22.8 dB for spatial resolution reduction; 27.2 dB for semantic prefiltering with lowpass-filtering; 35.6 dB for semantic prefiltering with static background. Thus, the selected adaptation strategy is semantic prefiltering with static background (3b) as well.

For *Highway*, the resource allocation problem is solved in a similar way. The adaptation strategies that provide most value for the end user are found to be semantic prefiltering with static background (3b) at 176 Kbit/s, and the coded original sequence (1) at 1000 Kbit/s. These results are next verified by visual inspection and by objective quality evaluation. The former is done by inspecting sample frames from sequences coded using different strategies. The latter is achieved by measuring SPSNR at 176 Kbit/s and at 1000 Kbit/s for each strategy.

Results

In Figure 6.18, sample frames are shown for the sequence *Hall monitor*. At 176 Kbit/s (left column), the person's face and the monitor have slightly more details with the semantic strategies (c) and (d) than with the non-semantic strategies (a) and (b). Also, the background is severely corrupted by coding artifacts in the coded original (a). This is particularly visible on background edges. At 1000 Kbit/s (right column), spatial resolution reduction (b) and lowpass-filtered background (c) have substantially lower quality than the coded original (a) and static background (d). On the other hand, it is difficult to perceive differences between the coded original (a) and static background (d).

In Figure 6.19, sample frames are shown for the sequence *Highway*. At 176 Kbit/s, the background of semantic prefiltering with static background (d) has higher quality than the background of the coded original (a). In particular, the white painted lines on the road are sharper with static background (d). At 1000 Kbit/s however, the shadow cast by the truck stops in an unnatural way in static background (d). These artificial boundaries result from the object segmentation process used by the semantic prefiltering step. These boundaries are visually annoying and lead to a lower perceptual quality for static background (d) than for the coded original (a).

The SPSNR for the two test sequences coded at 176 Kbit/s and at 1000 Kbit/s using different adaptation strategies is given in Table 6.4. As expected, the highest objective quality for *Hall monitor* is achieved by using semantic prefiltering with static background at both 176 Kbit/s and 1000 Kbit/s. For *Highway*, the highest SPSNR obtained by using semantic prefiltering with static background at 176 Kbit/s, and by the coded original at 1000 Kbit/s.

Both visual inspection and objective quality evaluation results confirm that the adaptive delivery framework proposed in Chapter 5 is capable of determining the adaptation strategy that leads to the best perceptual video quality. In fact, the strategies that have been selected for delivery have also the highest SPSNR in all tested cases. Also, the corresponding quality improvements are visible in the sample frames for all cases but for *Hall monitor* at 1000 Kbit/s.

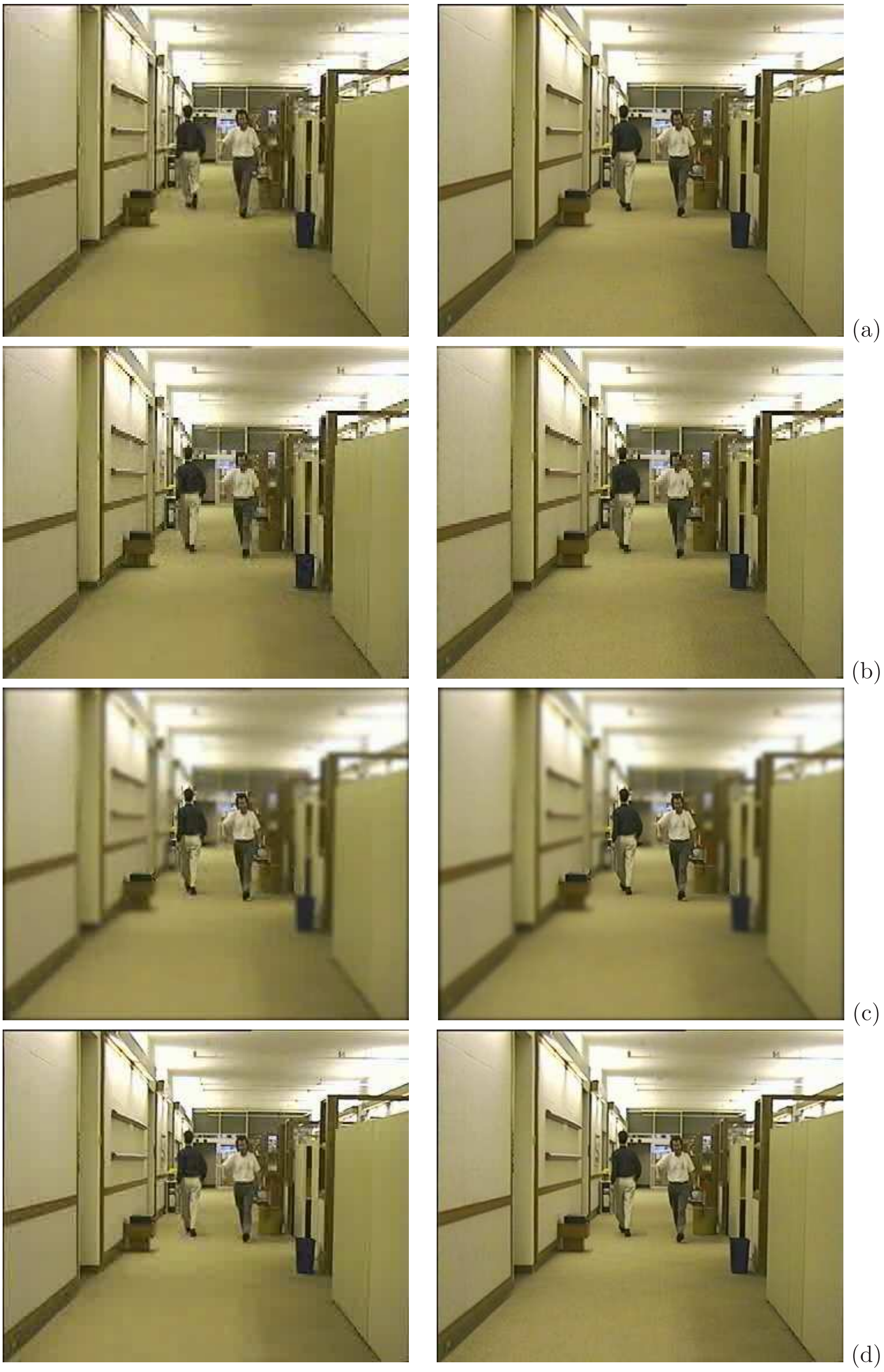


Figure 6.18: Frame 190 from *Hall monitor* for different adaptation strategies. The coding bitrates are: (left column) 176 Kbit/s; (right column) 1000 Kbit/s. The strategies under analysis are: (a) Coded original. (b) Spatial resolution reduction. (c) Lowpass-filtered background. (d) Static background.

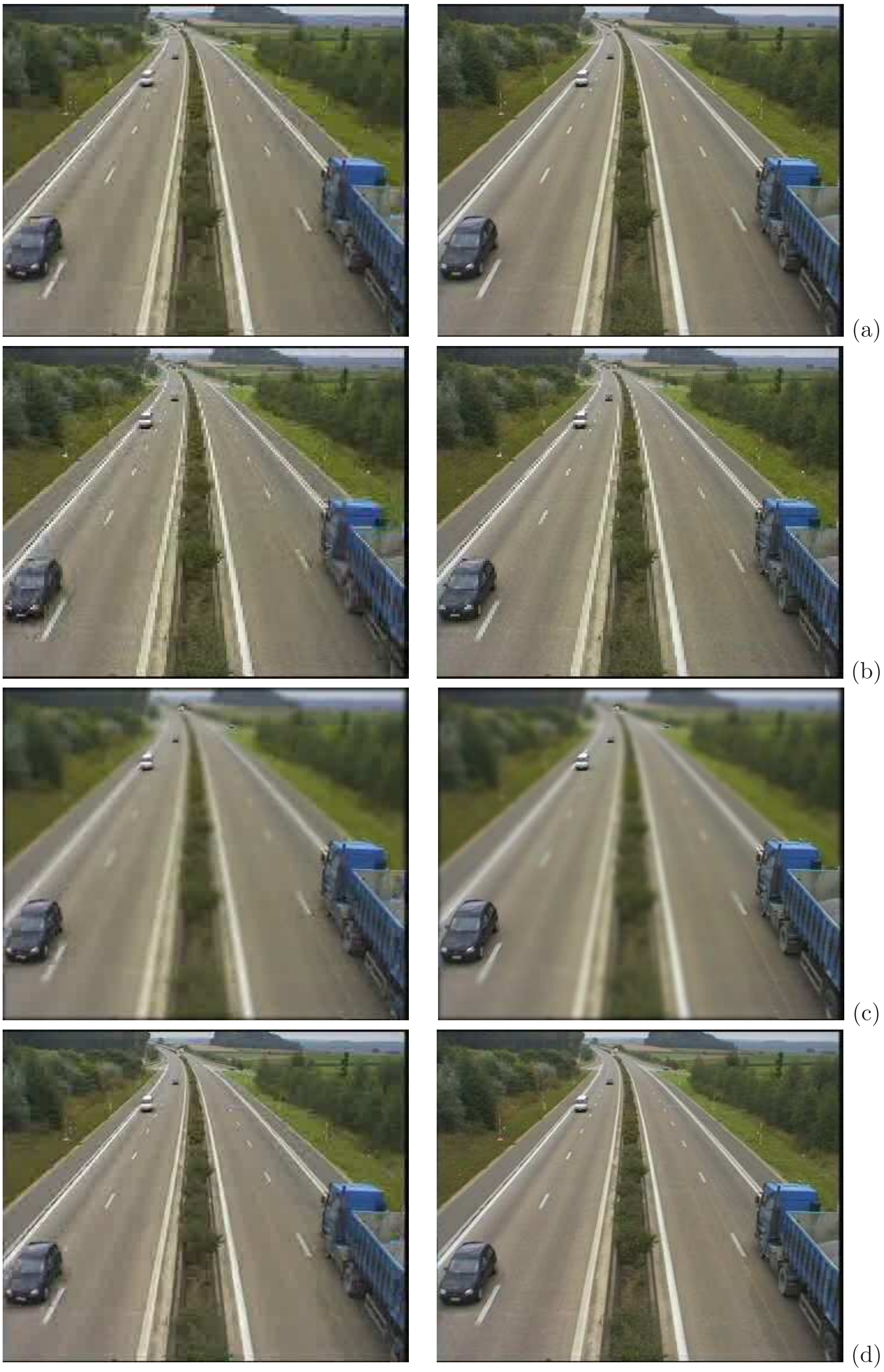


Figure 6.19: Frame 20 from *Highway* for different adaptation strategies. The coding bitrates are: (left column) 176 Kbit/s; (right column) 1000 Kbit/s. The strategies under analysis are: (a) Coded original. (b) Spatial resolution reduction. (c) Lowpass-filtered background. (d) Static background.

Bitrate	176 Kbit/s	1000 Kbit/s
Hall monitor		
Coded original	27.5 dB	35.0 dB
Spatial resolution reduction	22.7 dB	22.8 dB
Lowpass-filtered background	25.3 dB	27.2 dB
Static background	29.4 dB	35.6 dB
Highway		
Coded original	29.0 dB	35.8 dB
Spatial resolution reduction	23.4 dB	23.5 dB
Lowpass-filtered background	27.7 dB	30.1 dB
Static background	29.8 dB	35.1 dB

Table 6.4: SPSNR for the sequences *Hall monitor* and *Highway* coded at 176 Kbit/s and at 1000 Kbit/s using different adaptation strategies.

6.4 Summary

In this chapter, we have discussed experimental results obtained with standard test sequences and we have proposed a validation of our work in real applications.

In Section 6.2, the behavior of the motion-based semantic video object extraction algorithm presented in Chapter 4 has been analyzed. The algorithm produces reliable results in the presence of difficulties such as object *appearance* and *disappearance*, *deformable objects*, *merging*, *occlusions*, *splitting*, and *complex illumination conditions*. The *appearance* and the *disappearance* of objects do not to alter extraction performance. In fact, each region in the object segmentation mask is processed independently and does not affect the extraction of other video objects. *Deformable objects* such as people are coped with by tracking object's regions rather than the entire object. Thus, the loss of individual regions due to deformations does not corrupt the other regions' tracks. The tracking algorithm is further capable of separating multiple objects and of providing them with a coherent label over time when several objects are *merged* together in the object segmentation mask. This is a consequence of the region tracking process: regions keep their original label even when they are merged in a single object partition. The correspondence obtained through region tracking is further verified by data association between regions' descriptors in successive frames. To relate objects to their original trajectory after *total occlusions*, the data association step operates not only between subsequent frames, but on a longer temporal window. This enables us to deal with occlusions of objects by background elements, mutual occlusions, and temporal object disappearance. *Object splitting* is handled by allowing separate regions in the object partition to get their label from the same object. This mechanism further enables us to correct object segmentation errors: when an object is identified by two separate regions in the object segmentation mask, the projection of the region description allows us to keep the track of the full object instead of initiating a new track for the unconnected part. The latter is particularly useful in the presence of background noise. In the presence of *complex illumination conditions* at last (e.g., reflective surfaces), the tracking mechanism allows to put superfluous regions in the object segmentation mask in correspondence with the corresponding objects.

Semantic video objects and their associated trajectories have further been used by a content understanding step that monitors the behavior of objects in the scene to provide event detection for visual surveillance. A simple experiment has been performed, where authorized and restricted areas have been defined in an outdoor surveillance video. By comparing the position of video objects to the coordinates of the respective areas, intrusion alarms were generated correctly. The description of objects' features also enables video enhancement in order to put important objects in a conspicuous position. This helps lowering fatigue of the monitoring personnel that is due to extended concentration.

In Section 6.3, the performance of different video adaptation strategies proposed in Chapter 5 has been compared by means of objective quality evaluation and by visual inspection. At low bitrates, the video quality is increased by taking into account content semantics with both frame-based encoders (MPEG–1) and object-based encoders (MPEG–4). At such bitrates, the amount of coding artifacts affecting foreground objects is notably reduced by using semantic prefiltering. Therefore, semantic prefiltering is particularly useful to increase quality in bandwidth-critical applications such as mobile video. The additional cost of sending the description of object's shape and location for metadata-based and metadata-enhanced encoding has further been found to be below 100 Kbit/s. This cost is low as compared to medium-quality video coding with MPEG–1 or with MPEG–4.

Then, strategy selection has been used to provide adaptive delivery with real sequences for realistic client resource profiles. Visual inspection and objective quality evaluation using SPSNR confirm that the adaptive delivery framework is capable of determining the adaptation strategy that provides the most value for the user. In all tested cases, the strategy that has been selected for delivery by the mechanism leads also to the highest SPSNR. In most cases, the corresponding quality improvements are clearly visible in the adapted sequences. We would like to point out that with respect to the alternative approach of measuring value for each strategy at $R = R_{\text{client}}$ prior to delivery, the use of value functions permits to select the optimal strategy without explicit value computation for the present client resources. This helps to support resource fluctuations at the time of delivery.

In Appendix A, adaptive video delivery is discussed in the more general context of Universal Multimedia Access (UMA). In UMA, information is delivered to numerous users under a wide variety of conditions in a transparent form. With respect to adaptive delivery, UMA does not only handle the resources of access networks and terminals, but also the individual preferences of end users and the preparation of content for efficient search and browsing.

General conclusions 7

7.1 Summary of achievements

The increasing diversification of the means of information transport and access has created a need to personalize the way media content is delivered to the end user. Moreover, recent devices, such as digital radio receivers with graphics displays, and new applications, such as intelligent visual surveillance, require novel forms of video analysis for content adaptation and summarization. To cope with these challenges, we have proposed an automatic method for the extraction of semantics from video, together with a framework that exploits these semantics in order to provide adaptive video delivery.

After a short review of semantics in the general context, we have first introduced an algorithm that relies on motion information to extract multiple, simultaneous video objects in cluttered environment. Several applications, such as sport broadcasting and visual surveillance, exploit motion information to produce meaningful objects. The task of semantic video object extraction has been split into two stages, namely *object segmentation* and *object tracking*. The *object segmentation* stage deals with the segmentation of objects from the background. In our implementation, a statistical change detector decides whether in each pixel position, the foreground signal corresponding to an object is present. The selected approach is robust with respect to camera noise, and it does not need manual tuning along a sequence or for different sequences. The latter is important in outdoor surveillance applications, where slow illumination changes are common. This further enables our solution to operate in a wide variety of situations.

The *object tracking* stage follows individual objects along the frames. Temporal tracking allows to distinguish multiple objects even when they have similar motion or in the presence of mutual occlusions. This is essential in complex environment, where several objects interact together, and it further enables us to correct possible object segmentation errors. The tracking algorithm is based on feedbacks between a region partition and an object partition. The region partition defines homogeneous groups of pixels corresponding to perceptually uniform regions, whereas the object partition defines semantic video objects. The correspondence of semantic objects in successive frames is then achieved through the correspondence of individual regions comprising an object. With respect to the alternative approach of tracking semantic objects directly, region-object tracking provides several significant advantages. Deformable objects such as people are handled correctly, since the loss of individual regions due to self occlusions does not corrupt the tracks of the other regions. The method is also capable of separating multiple objects in the presence of mutual occlusions, since regions keep their original label when they are merged in a single area of the object segmentation mask. This further helps to correct object segmentation errors. Finally, data association between region descriptors in successive frames enables us to handle total occlusions and temporal object disappearances.

Subsequently, semantics have been used to improve adaptive video delivery. The presented framework comprises three connected parts: a number of complementary *video adaptation strategies*, means of *performance evaluation* for individual strategies, and a *strategy selection mechanism.* Some of the proposed *video adaptation strategies* use semantics in order to organize the content so that a particular network or device does not inhibit the main content message. Specifically, semantic prefiltering combines semantic video analysis with a traditional frame-based encoder. Relevant areas are first extracted from video by means of semantic analysis. The areas not included in the region of interest are then lowered in importance by using background simplification. Using a simplified background aims at taking advantage of the task-oriented behavior of the Human Visual System (HVS) for improving compression ratios. Metadata-based and metadata-enhanced encoding on the other hand uses textual description of video object features to efficiently encode the main content message. The use of metadata enables us to make the content more searchable and to improve visualization in video-based applications.

Performance evaluation has been used to evaluate the perceptual quality resulting from different adaptation strategies. A series of subjective experiments involving twenty human observers enabled us to quantify the impact of semantic prefiltering and of metadata-based encoding. Results show that background simplifications resulting from semantic prefiltering do not penalize overall quality at low bitrates, and that the metadata-based representation of object's shape suffices to convey the meaning and action of a scene when the objects are familiar. This is particularly important for bandwidth-critical applications such as mobile video. Based on these observations, we have derived an objective quality metric, the semantic peak signal-to-noise ratio (SPSNR), that accounts for different image areas and for their relevance to the observer in order to reflect the focus of attention of the HVS. The prediction accuracy of SPSNR is superior by up to 8% with respect to PSNR. Moreover, the metric allows to quantify the amount of attention that we pay to the foreground and to the background as a function of video content.

At last, a *strategy selection mechanism* that determines the optimal adaptation strategy with respect to the resources of the connected client has been proposed. This problem is formalized by a resource allocation problem, where content value is maximized under a given set of client resource constraints. In our implementation, the selection mechanism relies on SPSNR to measure the value of the delivered video.

To validate our approach, semantic video analysis and adaptive video delivery have been tested with real sequences and used to provide intelligent visual surveillance and adaptive delivery for realistic client resource profiles. Video object extraction is found to operate reliably in sequences including various difficulties. Video objects and their associated description further enable us to enhance surveillance video and to perform automated event detection. At last, the adaptive video delivery framework is capable of selecting the adaptation strategy that leads to optimal perceptual quality for low-quality as well as high-quality applications.

To conclude our summary, we would like to point out that each specific component of the system may be replaced by a more adequate one, depending on the particular requirements of the application at hand. For instance, motion-based semantic video object extraction could be replaced by a face detection process for crowds monitoring. This flexibility is a consequence of the modularity of the proposed system.

7.2 Perspectives

The work proposed in this dissertation can be improved and extended in several ways. Some directions for further work are proposed below.

- ▷ The semantic video object extraction algorithm discussed in Chapter 4 addresses the static camera problem. One natural extension is to deal with *moving camera* sequences by integrating global motion information.

- ▷ The semantic video analysis discussed in Chapter 4 has been laid out for a single camera. The method could be extended to multiple, overlapping cameras in order to recover the perspective of the scene (3D analysis). This might notably help to solve the problem of tracking occluded objects. By considering multiple non-overlapping cameras, the method could further be used for the surveillance of vast areas.

- ▷ Semantic video analysis operates at the region level and at the object level. In addition to these, the content level could be considered by adding an additional *object recognition* step to the system. This would allow the identification of the nature of visible objects, which could in turn be used in order to refine the description of the video and to improve the reliability of video object extraction by eliminating irrelevant objects.

- ▷ The video adaptation strategies proposed in Chapter 5 operate in the spatial domain. These strategies could be extended to the *temporal domain*, for instance by using individual frame rates for different image areas as a function of motion activity.

- ▷ In Chapter 5, the SPSNR video quality metric is proposed to assess the value of different adaptation strategies. The metric operates on video, but does not lead to relevant results for strategies such as metadata-based and metadata-enhanced encoding. An objective metric for *cross-modality performance evaluation* would be very desirable. Such a metric might for instance be created based on additional subjective experiments.

- ▷ The SPSNR video quality metric is in essence a PSNR, where image areas are weighted according to their relevance to the observer. In some cases, it might be desirable to replace the PSNR by more *advanced quality metrics*. Such extensions are notably addressed in [35].

- ▷ The adaptation strategy selection mechanism discussed in Chapter 5 operates on a finite set of adaptation operators. Future work might consider the selection of optimal *adaptation parameters* as well. This could for instance be used to control the variance of the lowpass filter used to perform background simplification.

- ▷ The adaptation strategy selection mechanism requires value to be computed explicitly for each candidate strategy. Such calculations are time-consuming and need to be performed offline. *Value function prediction* [246] helps to predict value based on content features without explicit computation.

Appendix

Universal Multimedia Access

A

A.1 Introduction

Adaptive video delivery is an important component of a wider framework where information is delivered to numerous users under a wide variety of conditions in a transparent form. This is referred to as Universal Multimedia Access (UMA) [175, 232]. A UMA environment enables us to handle the individual preferences of multiple end-users, the technical characteristics of various networks and appliances, and content items of different nature. Therefore, UMA has a wealth of potential applications in broadcasting, the World Wide Web and mobile telecommunications, where all types of users, content, appliances and networks coexist [173].

To target the specific challenges of UMA, several researchers have been relying on the MPEG family of standards. Van Beek et al. [227] use MPEG–7 Variation and Summary tools to manage alternative content versions. Fossbakk et al. [74] achieve similar results by using MPEG–21 Digital Item Declaration. Sun et al. [176, 217] rely on MPEG–21 usage environment description tools to provide resource adaptation in a streaming media environment. The use of MPEG–21 in order to handle adaptive content delivery has also been considered by Bormans et al. [20]. Another concern in the past has been the problem of generating and coding appropriate content items. Some of the basic technology required for media conversion to support mobile users is reviewed by Vetro and Sun [235]. Lee et al. [128] propose a scheme for generating transcoded video sequences that fit the respective display size of a variety of client devices. Their scheme uses a transcoding algorithm based on perceptual hints. MPEG–4 fine-granular scalability coding (FGS) has been adapted to the transmission of video over wireless and mobile networks by van der Schaar and Radha [231]. Wang et al. [242] combine MPEG–4 FGS and MPEG–21 into a unique system for real-time video streaming over heterogeneous networks with time-varying conditions. Finally, content adaptation to different user profiles for digest video delivery and for mobile multimedia services have been reported by Echigo et al. [64] and Chen et al. [40], respectively.

In this appendix, we present a complete UMA framework based on video [215]. Within that framework, the method in Part II of our dissertation has been employed to provide adaptive video delivery. The block diagram in Figure A.1 is subdivided into two connected parts, namely content

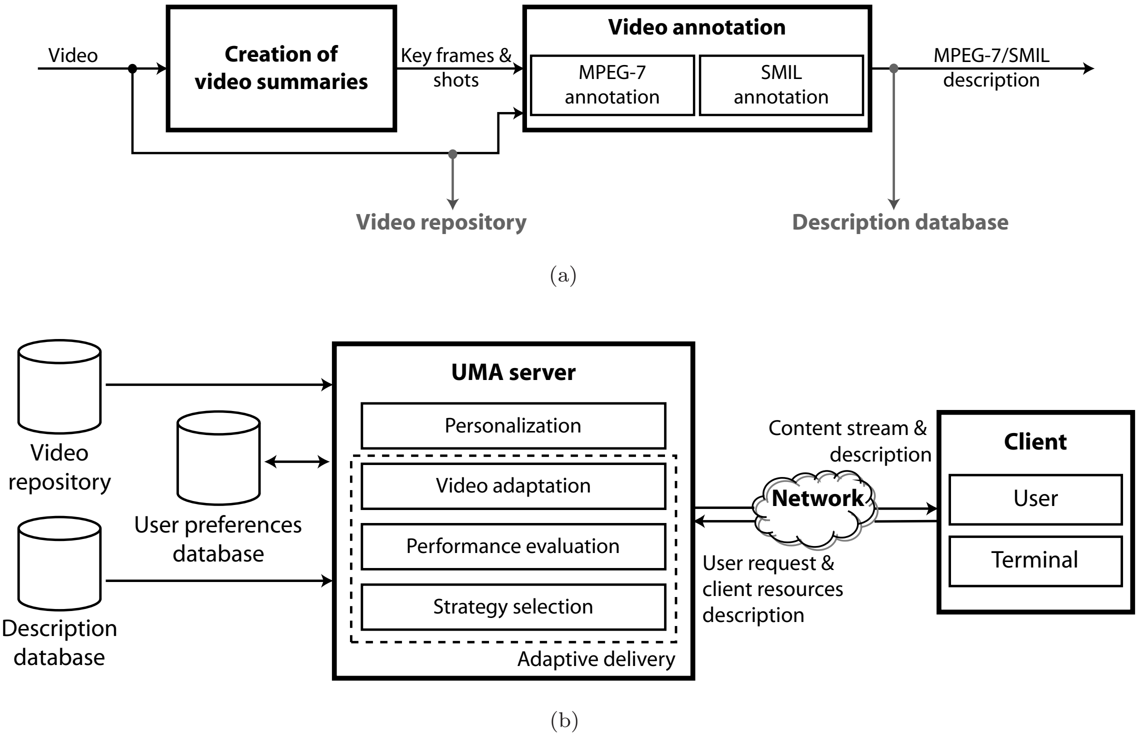


Figure A.1: The presented universal multimedia access framework is subdivided into two connected parts. (a) Content preparation. (b) Content delivery.

preparation and content delivery. The goal of content preparation is to create video summaries for fast database browsing and to generate content-based description or *metadata*. Content delivery deals with the adaptive streaming of video material to the connected client and with the personalization of the service according to individual users' preferences. The power and novelty of our solution ensue from the integration of automatic summarization, open standards-based annotation, personalization and adaptive delivery in a unified framework that is transparent to the user. The UMA framework has been developed, implemented and tested by the author et al. in the EC–funded R&D project PERSEO.

The remainder of this appendix is organized as follows. Content preparation, i.e. the creation of video summaries and the annotation of content using open standards, is addressed in Section A.2. Personalization and adaptive delivery are discussed in Section A.3. At last, we report the implementation of the proposed UMA framework by the PERSEO consortium in Section A.4.

A.2 Content preparation

Content preparation includes the creation of summaries for fast database browsing and the annotation of video using open standards. Video summaries are extracted by means of automatic algorithms. For content-based annotation, a set of MPEG–7 descriptors is provided along with SMIL descriptors for the authoring of interactive video.

A.2.1 Creation of video summaries

The creation of video summaries is of particular interest in a UMA environment, where large content collections are searched by devices with limited capabilities. Summary creation mainly consists in segmenting the video into elementary units and then extracting representative frames from these units. Figure A.2 shows the block diagram of a simple system for video summarization [4]. First, the original video is processed in order to extract low-level visual primitives such as color, motion and texture. Based on these primitives, the video is segmented into basic units called *shots* through temporal segmentation. A shot is defined as a sequence of frames captured by one camera in a single continuous action in time and space [155]. Once the shot boundaries are detected, the salient content of each shot is represented in terms of a small number of frames, called *key frames*. Temporal segmentation and key frame extraction make up the video parsing process.

Different summary representations are then created from the detected shots and from the extracted key frames. The main objective of these representations is to allow a content-based browsing of the video. In the hierarchical summary, key frames are grouped and organized in order to obtain a coarse-to-fine hierarchy of summaries, i.e., the content of video is represented at multiple levels of detail, from coarse summaries to detailed summaries [268]. The sequential summary is simply a concatenation of the key frames which can be shown sequentially in time, for example as an animated slide show. In the mosaic-based summary, each shot is decomposed into static and dynamic components [102]. The static appearance is represented by a mosaic, which is constructed by aligning and integrating frames. The dynamic behavior of the moving objects is represented by their trajectories and characteristic appearances. In the pictorial summary at last, key frames are resized into corresponding subimages according to the importance of the frames (*video posters*) [257].

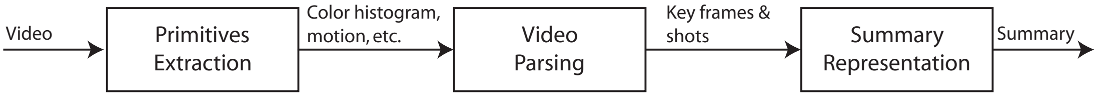


Figure A.2: Block diagram of a simple system for video summarization.

A.2.2 Video annotation

To provide effective access to large content collections, metadata is associated to the video. Interoperability with third-party applications is allowed by the use of open standards. The MPEG–7 Multimedia Content Description Interface defines a standardized set of descriptors for multimedia content features. The Synchronized Multimedia Integration Language SMIL permits simple authoring of interactive presentations.

MPEG–7 annotation

Table A.1 lists a set of MPEG–7 descriptors [111] for annotation that fit the specific needs of UMA. *Description metadata* captures the version, author and history of the description. This is mainly used to record the annotation history at the content provider's side. *Media information* identifies and describes media-specific information of the video. In particular, this defines a unique content ID and URL, as well as information about the file and coding formats. Later on, this media-specific information permits to determine whether the connected terminal has sufficient capabilities to playback the content item without prior adaptation. *Creation information* describes information about the creation, production and classification of multimedia content. The *semantic description* is used to describe real-life concepts or narratives which are depicted by or related to the content.

MPEG–7 Descriptor	Purpose	Extraction
Description metadata		
`DescriptionMetadata`	Description version, author and history	Manual
Media information		
`MediaIdentification D`	Unique content identifier	Automatic
`MediaFormat D`	Information about file and coding format	Automatic
`MediaInstance DS`	URL of content file	Automatic
Creation information		
`Creation DS`	Content title, abstract, creator and creation tool	Manual
`Classification DS`	Content form, genre, subject, purpose, language, ...	Manual
Usage description		
`Rights datatype`	Information about right holders and content usage	Manual
Semantic description		
`AgentObject DS`	*Who*: describe persons or organizations	Manual
`Event DS`	*What*: describe a semantic activity	Manual
`SemanticPlace DS`	*Where*: describe location	Manual
`SemanticTime DS`	*When*: describe time	Manual
Structure description		
`Segment DS`	Specify temporal segments in multimedia content	Automatic
Access tools		
`SummarySegmentGroup DS`	Define video summaries using key frames and shots	Automatic

Table A.1: MPEG–7 descriptors for video annotation.

These enable us to perform refined content search based on keywords. Additionally, this information is matched with user preferences to sort content in terms of relevance. *Usage description* holds information about right holders and content usage. This allows the system to restrict delivery to entitled user categories. *Structure description* helps to access specific content segments rapidly. At last, video summaries are defined by MPEG–7 *access tools* that capture the temporal location of key frames and shot boundaries. At the time of delivery, this information is used by the terminal to render the summary.

The extraction of several of the above features is performed automatically. Automatic feature extraction is usually more effective in terms of extraction time, but might sometimes lead to inaccurate results. These can be corrected by manual tuning. Key frames and shots are automatically extracted using the method in Section A.2.1. Media information is read from the content file's header. The remaining features are annotated by hand.

SMIL annotation

Interactive video contains visual elements that the user can interact with. An interactive element is defined by its shape and color, its starting time, its ending time, and its position or trajectory throughout the video. The manual definition of these parameters is a lengthy and tedious process. However, they can also be defined using the automatic extraction algorithm presented in Chapter 4, because interactive elements are normally related to semantic objects in the video. Interactive elements are further assigned actions to be performed when clicking on them. Possible actions include opening a web page, sending a message (e.g., email, SMS or MMS), and forwarding the

SMIL Descriptor	Purpose	Editing mode
Media Object Modules		
`video`	Define interactive video	Automatic
`img`	Define interactive image	Automatic
Layout Modules		
`layout`	Spatial layout of the links	Auto. or Manual
`region`	Spatial coordinates of a link	Auto. or Manual
`z-index`	Links layer hierarchy	Automatic
Linking Modules		
`a`	Object-dependent link information (URL, behavior, ...).	Manual
`area`	Region-dependent link information	Manual
Animation Modules		
`animate`	Temporal behavior of the links	Auto. or Manual
Timing and Synchronization Module		
`par`	Parallel playback	Automatic

Table A.2: SMIL 2.0 Modules and Elements for interactive video.

player to another multimedia element. Thus, interactive elements defines contextual links among the items in the video database and on the internet.

Interactive video is created as SMIL objects referenced from the content description. This means that no new video is created. Instead, the description is used by the terminal to provide interactivity on the fly. Since no new media are created, multiple interactive video versions can be stored at low additional cost. Table A.2 lists the SMIL 2.0 Modules and Elements [240] used to perform the editing of interactive video. The *media object module* describes the media objects used to embed dynamic links. The *layout module* is used for the positioning of the interactive areas on the visual rendering surface. The *linking module* defines the attributes and elements of navigational hyperlinks. The *Animation Module* contains elements and attributes for incorporating animations into a timeline. The *timing and synchronization Module* finally is used for synchronization between video and embedded links.

A.3 Content delivery

Content delivery involves two distinct parties: the UMA server and the client. The UMA server is in charge of personalization according to individual users' preferences, and it provides adaptive video delivery. This process is transparent for the user, leaving him off the tough task of dealing with hardware characteristics. The client, which comprises both the end user and the end user's access terminal, is characterized by a set of resources that are defined by MPEG–21 descriptors. The interaction between the server and the client follows four steps:

1. *Personalization.* All available content items are sorted and listed according to the personal preferences of the connected user. Preferred items are listed first; undesired items are sorted out.

2. *Generic search.* The user searches the sorted list based on keywords (i.e., title, author, location, action, etc.). For each item, a video summary is proposed.

3. *Content selection.* The user selects the item to be delivered.

4. *Adaptive delivery.* The selected item is adapted and streamed to the terminal. Interactive elements are rendered by the terminal.

The personalization mechanism is discussed next. User preferences are acquired from the user in a questionnaire and updated by the server according to the user's usage history. Then, the description of client resources using MPEG–21 is addressed. This is exploited by the server to provide adaptive delivery.

A.3.1 Personalization

Personalization is utilized by consumers for accessing multimedia content that fits their personal preferences. By using the user's preferences in conjunction with a history of the actions that he has carried out over a specific period of time, the server dynamically updates the user profile. In our framework, user preference and the usage history are described using MPEG–7 user interaction tools [111]. These define two types of preferences: *filtering and search preferences*, and *browsing preferences.* The former are used to describe filtering or searching preferences in terms of attributes related to the creation, the classification and the source of the content. The latter describe user preferences pertaining to navigation of and access to content. In particular, a user may express preferences of the type and content of video summaries. User preferences and the usage history are stored in the user preferences database maintained by the server. Initial user preferences are acquired from the user in a questionnaire.

User preferences handling is based on a continuous check of the user's interaction with the system. Depending on the content the user is accessing, the system dynamically updates the MPEG–7 user profile. Updates depend on the category the selected video item belongs to, and on past user's history accessing categorized video material. In order to take both these factors into account, the user profile is update according to a Q-Learning based mechanism [248], where states (categories in the user profile) belonging to branches with similar semantic meaning get rewarded; non selected media are punished (negatively rewarded). As a result of the continuous process of dynamically adjusting and updating the user profile, the system is able to select and to filter all the annotated content according to the profile. The selection of content from the video database according to the user profile is based on the following guidelines:

1. Content in the database is filtered based on the user's preferences. That is, content belonging to undesired categories is sorted out.

2. Content is sorted according to the usage history: the more relevant preferences (those that have been more actively accessed in the past) are presented before those that have been accessed less, or not at all.

As a result of the process, a list of classified, sorted and pruned video content regarding the user's preferences is generated.

A.3.2 Adaptive delivery and client resources description

The diversity of networks and terminals in a realistic UMA environment makes it unattainable to generate a distinct content version for each profile of capabilities. Thus, adaptive delivery is needed. As discussed in Chapter 5, the selection of the adaptation strategy that provides the most value for the end user requires some knowledge about the resources of the connected client and network.

MPEG–21 Descriptor	Purpose
Codec capabilities	
Decoding capabilities	
`TransportFormat`	Transport formats the terminal is capable of decoding
`VideoFormat`	Video formats the terminal is capable of decoding
`AudioFormat`	Audio formats the terminal is capable of decoding
Video parameters	
`BitRate`	Nominal bitrate in bit/s.
`Maximum`	Max. value for BitRate in case of variable bitrate
`Average`	Avg. value for BitRate in case of variable bitrate
Audio parameters	
`BitRate`	Nominal bitrate in bit/s.
`Maximum`	Max. value for BitRate in case of variable bitrate
`Average`	Avg. value for BitRate in case of variable bitrate
Input/output capabilities	
Display capabilities	
`Resolution`	Resolution in pixels
`BitsPerPixel`	Color depth in bits
`ColorCapable`	Describes whether display is color capable
`RefreshRate`	Refresh rate in Hz.
Audio output capabilities	
`LowFrequency`	Lower bound of audio frequency range in Hz.
`HighFrequency`	Higher bound of audio frequency range in Hz.
`NumChannels`	Number of supported audio channels

Table A.3: MPEG–21 descriptors for client resources.

In our context, the term *client* refers to both the user and the user's terminal. The user can be any person or agent, characterized by his own set of preferences. Possible access terminals range from high-performance appliances like digital TV sets and personal computers (PC), down to mobile devices such as personal digital assistants (PDA), mobile phones and wearable computers. Each client is characterized by a set of resources that describe the terminal's own technical features. Client resources are stored in the appliance and accessed by the content server at the time of connection. Alternatively, capabilities of devices with limited memory can be stored in the content server and retrieved according to the client's individual ID.

Client resources are described using the MPEG–21 DIA [112] usage environment descriptors in Table A.3. *Codec capabilities* specify the decoding capabilities of a connected terminal. These includes the transport, video and audio formats that can be decoded, as well as video and audio bitrate specifications. *Input and output capabilities* describe the I/O capabilities of the terminal. For video, resolution, color depth and refresh rate of the display are specified. Audio parameters include the output frequency range and the number of audio channels.

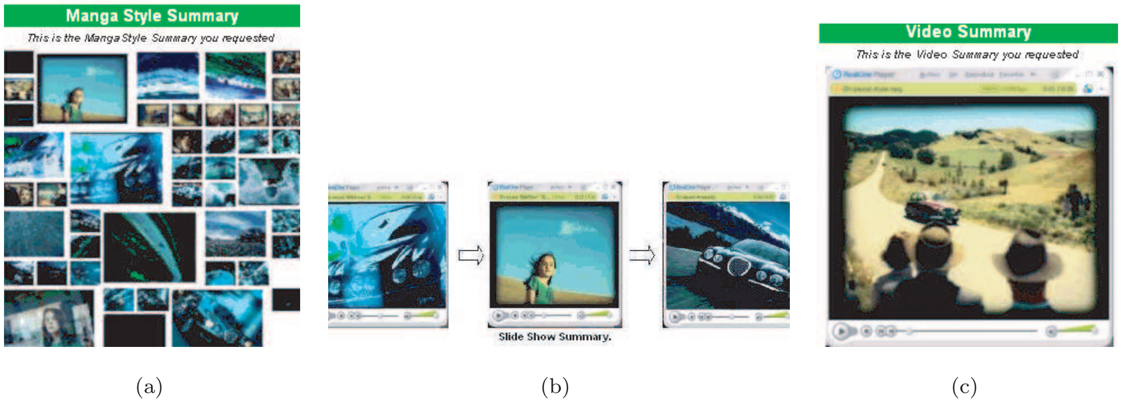

Manga Style Summary
This is the Manga Style Summary you requested
Slide Show Summary.
Video Summary
This is the Video Summary you requested
(a)
(b)
(c)

Figure A.3: Three video summary representations are supported by PERSEO. (a) Pictorial summary (*video poster*). (b) Sequential summary using key frames (*slide Show*). (c) Sequential summary using key sequences (*video Summary*).

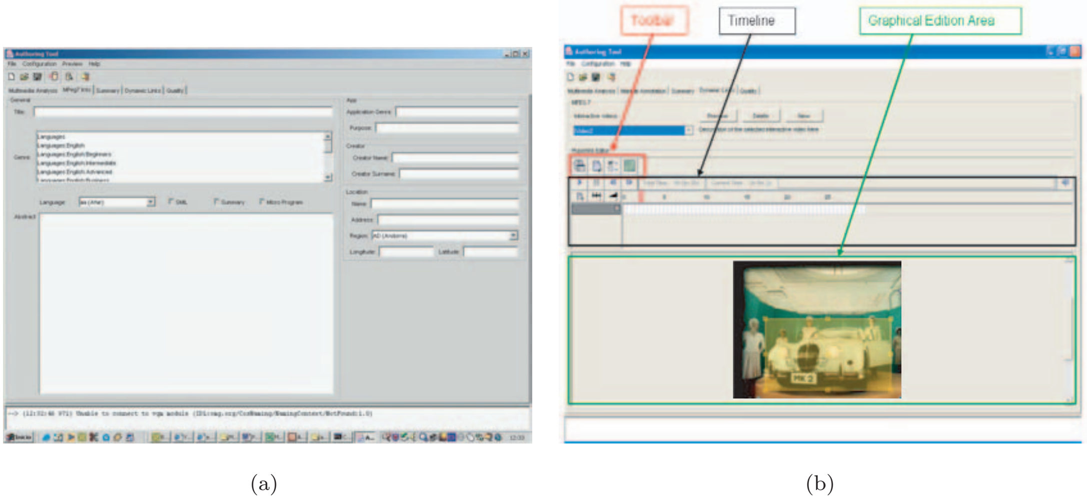
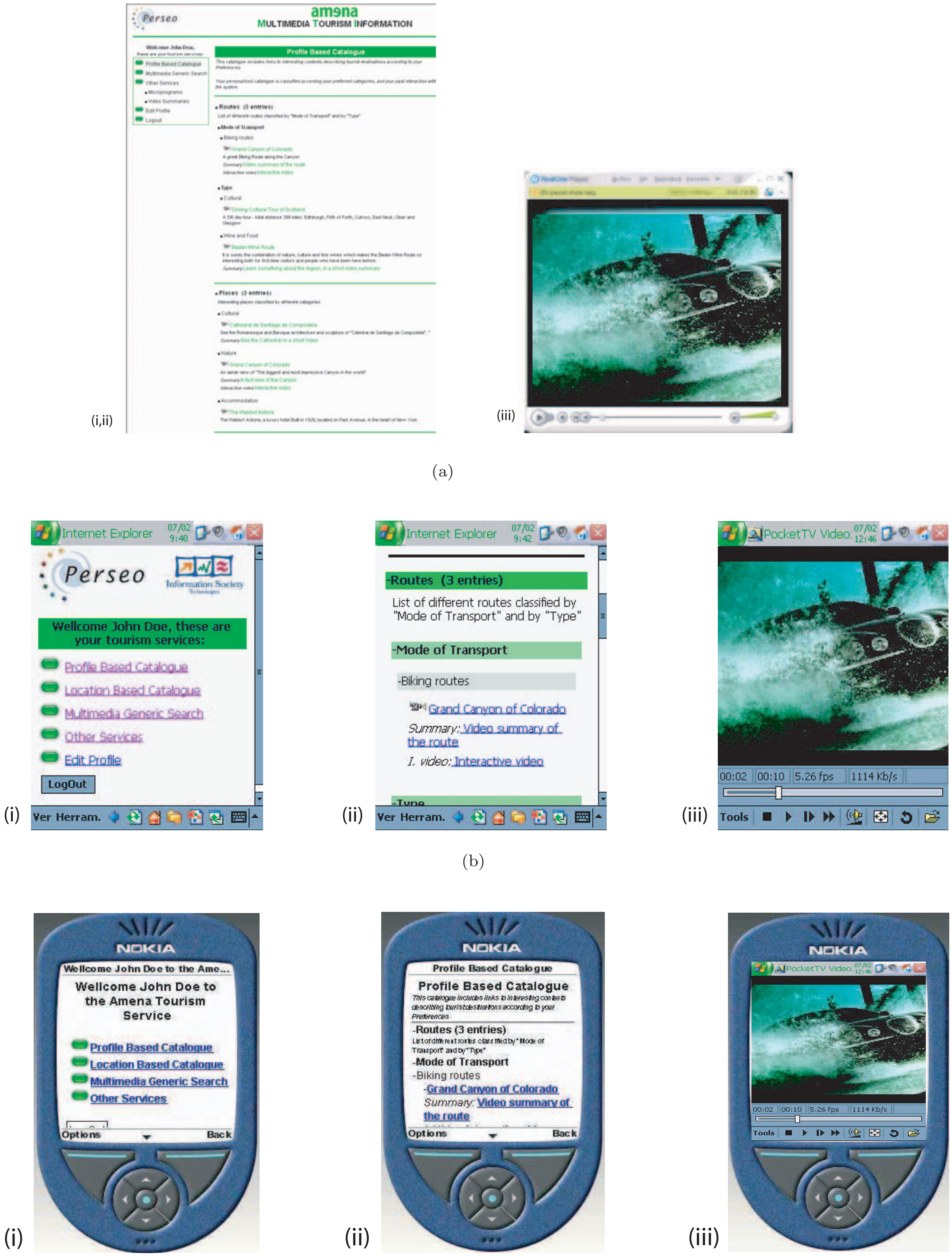


Figure A.4: Graphical user interfaces of PERSEO's content preparation software. (a) MPEG–7 content annotation. (b) Interactive video authoring.



Perseo
amena
MULTIMEDIA TOURISM INFORMATION

Profile Based Catalogue

Internet Explorer 07/02 9:40

Perseo Information Society Technologies

Wellcome John Doe, these are your tourism services:

- Profile Based Catalogue
- Location Based Catalogue
- Multimedia Generic Search
- Other Services
- Edit Profile

LogOut

Ver Herram.

Internet Explorer 07/02 9:42

-Routes (3 entries)

List of different routes classified by "Mode of Transport" and by "Type"

-Mode of Transport

-Biking routes

Grand Canyon of Colorado

Summary: Video summary of the route

I. video: Interactive video

-Type

PocketTV Video 07/02 12:46

00:02 | 00:10 | 5.26 fps | 1114 Kb/s

Tools

NOKIA

Wellcome John Doe to the Ame...

Wellcome John Doe to the Amena Tourism Service

- Profile Based Catalogue
- Location Based Catalogue
- Multimedia Generic Search
- Other Services

Options Back

Profile Based Catalogue

Profile Based Catalogue

This catalogue includes links to interesting contents describing touristic destinations according to your Preferences

-Routes (3 entries)

List of different routes classified by "Mode of Transport" and by "Type"

-Mode of Transport

-Biking routes

-Grand Canyon of Colorado

Summary: Video summary of the route

Options Back

Figure A.5: Interaction of the UMA server with different terminals. (a) PC. (b) PDA. (c) GPRS-capable mobile phone. Three successive operations are depicted: (i) Options menu. (ii) User profile-based content catalogue. (iii) Video delivery.

A.4 The PERSEO implementation

The presented content preparation and delivery framework has been implemented in the EC–funded R&D project PERSEO and tested by selected groups of end-users. PERSEO is a UMA system that covers all the aspects of multimedia production, post-production, annotation, database management and final cross-media and multi-device publication. The PERSEO implementation comprises two distinct modules: content preparation software, and the UMA server. The subdivision into two modules reflects the organization of the framework in Figure A.1. Content preparation software provides facilities for summary generation and for content annotation. Three summary representations are supported by the application: pictorial summaries, sequential summaries using key frames and sequential summaries using shots. Pictorial summaries show the most important frames laid out in a web page by chronological and importance criteria. Sequential summaries using key frames display the key frames of the video with a latency of a few seconds between them. The sequential summary using shots is a trailer of the video: a set of relevant shots in chronological order. These summary representations are depicted in Figure A.3. MPEG–7 content annotation is done by means of a graphical user interface (GUI). The customizable GUI in Figure A.4(a) is subdivided into input areas corresponding to the different annotation features required by the application at hand. User input is automatically translated to MPEG–7 by the software. Another GUI is provided for interactive video authoring in Figure A.4(b). Interactive elements are drawn onto the video frames, and their temporal extent is specified in the timeline.

The UMA server handles personalization and adaptive delivery. Initial preferences are acquired from the user by filling a questionnaire and updated dynamically according to a Q-Learning based mechanism. Content adaptation operations performed by the server include bitrate reduction, spatial resolution reduction and temporal resolution reduction. The interaction of the server with a PC, a color PDA and a GPRS-capable mobile phone is depicted in Figure A.5[*]. (i) After the login procedure, possible options are displayed in a menu. The user can access a content catalogue sorted according to his personal preferences or according to his current location (when available), perform a generic search, edit his own profile, or access other services. (ii) The user profile-based catalogue shows a sorted list of playable media items, together with available summaries. (iii) The selected content is displayed using the adequate frame rate, resolution and color depth.

A.5 Summary

The PERSEO implementation of the presented UMA framework has been utilized to provide two different applications to a selected group of end-users: e-learning and tourism information. The e-learning application provides video course material to professional and non-professional users, using static and mobile devices. In the tourism application, video material on various European cities is accessed by mobile devices. Usability validation has been achieved by focus group techniques using evaluation forms based on the quality attributes required for the developed services. Focus groups validation considered two main segments to analyze: (i) a residential segment aged 18 to 35. This segment was split in two focus groups, a first group of young people aged 18 to 25, and a second group aged 25 to 35. (ii) Professional users. Here, only one focus group has been formed.

The evaluation of the service has been positive in general terms in all the groups, and somewhat more critical in the segment of 25 to 35 years and in professional users. The service is considered to be attractive, and has been very useful in specific situations. The user validation phase leads to the following conclusions:

[*] For improved readability, simulations that accurately reflect the actual PERSEO client display are shown.

- **Personalization** is considered as a positive feature, due to the comfort, effectiveness and speed that the service gains. Due to the very nature of personalization, these advantages are not distinctly noticed in the everyday use of the service.

- **Video Summaries** are very well considered, as they help searching large content collections efficiently.

- **Generic search** on the content list is considered easy to use and positive. The result list should be limited in order to minimize rejection of the service. An extensive list of videos can generate deception and rejection of the service. The user's familiarity with internet search engines contributes to the acceptance of the idea.

- The UMA service has been found **particularly useful in mobile applications**. This is mainly because of adaptive video delivery that enables the user to get optimal content quality on terminals with limited capabilities.

- The video must have strong and practical information content. **The service can only be as good as the proposed content.**

- Regarding the **usability** of the developed applications, all the groups considered it as an easy to use service. The basic nature of the groups made that they were used to manage mobile phones and PDAs. The users felt confident that, with a little training, they would learn to efficiently use all the features of the system in a short time period.

The present solution can be extended in several ways. The support of additional modalities such as text, speech, stereoscopic video and 3-D will allow the emergence of novel applications and the delivery to additional appliances such as digital radio receivers, text terminals and wearable computers with head-mounted displays. *Transmoding* will be a key technology to the seamless integration of multiple modalities. Also, additional personalization criteria such as geographical location and time might be taken into account in order to provide selective services.

Bibliography

[1] T. Aach, U. Franke, R. Mester (1989). Top-down image segmentation using object detection and contour relaxation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'89*, vol. 3, pp. 1703–1706, Glasgow, Scotland.

[2] T. Aach, A. Kaup (1995). Bayesian algorithms for adaptive change detection in image sequences using Markov random fields. *Signal Processing: Image Communication* **7**(2):147–160.

[3] T. Aach, A. Kaup, R. Mester (1993). Statistical model-based change detection in moving video. *Signal Processing* **31**(2):165–180.

[4] Y. Abdeljaoued (2001). *Feature Point Extraction and Tracking for Video Summarization and Manipulation.* Ph.D. thesis, No. 2513, Ecole Polytechnique Fédérale de Lausanne (EPFL).

[5] S. S. Ahmeda, M. Keche, I. Harrison, M. S. Woolfson (1997). Adaptive joint probabilistic data association algorithm for tracking multiple targets in cluttered environment. *IEE Proceedings on Radar, Sonar and Navigation* **144**(6):309–314.

[6] J. C. Alexander, A. I. Thaler (1971). The boundary count of digital pictures. *Journal of the ACM* **18**(1):105–112.

[7] P. A. A. Assunção, M. Ghanbari (1998). A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG–2 bit streams. *IEEE Transactions on Circuits and Systems for Video Technology* **8**(8):953–967.

[8] Y. Bar-Shalom, T. E. Fortmann (1988). *Tracking and Data Association.* Academic Press, San Diego, USA.

[9] J. L. Barron, D. J. Fleet, S. S. Beauchemin (1994). Performance of optical flow techniques. *International Journal of Computer Vision* **12**(1):43–77.

[10] M. Bennamoun, G. J. Mamic (2002). *Object Recognition: Fundamentals and Case Studies.* Springer, London, UK.

[11] M. Bertini, R. Cucchiara, A. D. Bimbo, A. Prati (2003). Object and event detection for semantic annotation and transcoding. In *Proceedings of IEEE International Conference on Multimedia and Expo, ICME'03*, vol. 2, pp. 421–424, Baltimore, USA.

[12] D. Beymer, P. McLauchlan, B. Coifman, J. Malik (1997). A real-time computer vision system for measuring traffic parameters. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'97*, pp. 495–501, San Juan, Puerto Rico.

[13] J. C. Bezdek (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum, New York, USA.

[14] N. Björk, C. Christopoulos (1998). Transcoder architectures for video coding. *IEEE Transactions on Consumer Electronics* **44**(1):88–98.

[15] J. Black, T. Ellis (2001). Multi camera image tracking. In *Proceedings of International Workshop on Performance Evaluation of Tracking and Surveillance, PETS'01*, Hawai, USA.

[16] M. J. Black, D. J. Fleet, Y. Yacoob (2000). Robustly estimating changes in image appearance. *Computer Vision and Image Understanding* **78**(1):8–31.

[17] M. J. Black, A. D. Jepson (1996). *EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation.* Tech. Rep. RBCV-TR-96-50, University of Toronto.

[18] H. A. P. Blom, Y. Bar-Shalom (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control* **33**(8):780–783.

[19] G. Borgefors (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing* **34**(3):344–371.

[20] J. Bormans, J. Gelissen, A. Perkis (2003). MPEG–21: The 21st century multimedia framework. *IEEE Signal Processing Magazine* **20**(2):53–62.

[21] P. Bouthemy, M. Gelgon, F. Ganansia (1999). A unified approach to shot change detection and camera motion characterization. *IEEE Transactions on Circuits and Systems for Video Technology* **9**(7):1030–1044.

[22] A. P. Bradley, F. W. M. Stentiford (2003). Visual attention for region of interest coding in JPEG 2000. *Journal of Visual Communication and Image Representation* **14**:232–250.

[23] M. Bramberger, J. Brunner, B. Rinner, H. Schwabach (2004). Real-time video analysis on an embedded smart camera for traffic surveillance. In *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 174–181.

[24] A. Breen (1992). Speech synthesis models: a review. *Electronics & Communication Engineering Journal* **4**(1):19–31.

[25] A. D. Bue, D. Comaniciu, V. Ramesh, C. Regazzoni (2002). Smart cameras with real-time video object generation. In *Proceedings of IEEE International Conference on Image Processing, ICIP'02*, vol. 3, pp. 429–432, Rochester, USA.

[26] I. Carlbom, J. Paciorek (1978). Planar geometric projections and viewing transformations. *ACM Computing Surveys* **10**(4):465–502.

[27] R. Castagno (1998). *Video Segmentation Based on Multiple Features for Interactive and Automatic Multimedia Applications.* Ph.D. thesis, No. 1894, Ecole Polytechnique Fédérale de Lausanne (EPFL).

[28] R. Castagno, T. Ebrahimi, M. Kunt (1998). Video segmentation based on multiple features for interactive multimedia applications. *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5):562–571.

[29] A. Cavallaro (2002). *From Visual Information to Knowledge: Semantic Video Object Segmentation, Tracking and Description.* Ph.D. thesis, No. 2515, Ecole Polytechnique Fédérale de Lausanne (EPFL).

[30] A. Cavallaro, T. Ebrahimi (2001). Video object extraction based on adaptive background and statistical change detection. In *Proceedings of SPIE Electronic Imaging – Visual Communications and Image Processing*, pp. 465–475, San Jose, USA.

[31] A. Cavallaro, O. Steiger, T. Ebrahimi (2002). Multiple video object tracking in complex scenes. In *Proceedings of Tenth ACM International Conference on Multimedia*, pp. 523–532, Juan Les Pins, France.

[32] A. Cavallaro, O. Steiger, T. Ebrahimi (2003). Semantic segmentation and description for video transcoding. In *Proceedings of IEEE International Conference on Multimedia and Expo, ICME'03*, vol. 3, pp. 597–600, Baltimore, USA.

[33] A. Cavallaro, O. Steiger, T. Ebrahimi (2004). Perceptual prefiltering for video coding. In *Proceedings of IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP'04*, Singapore.

[34] A. Cavallaro, O. Steiger, T. Ebrahimi (2005). Tracking video objects in cluttered background. *IEEE Transactions on Circuits and Systems for Video Technology* **15**(4).

[35] A. Cavallaro, S. Winkler (2004). Segmentation-driven perceptual quality metrics. In *Proceedings of IEEE International Conference on Image Processing, ICIP'04*, pp. 3543–3546, Singapore.

[36] A. Cavallaro, F. Ziliani, R. C. T. Ebrahimi (2000). Vehicle extraction based on focus of attention multi feature segmentation and tracking. In *Proceedings of European Signal Processing Conference, EUSIPCO'00*, pp. 2161–2164, Tampere, Finland.

[37] W. C. Chan, O. C. Au, M. F. Fu (2002). Improved global motion estimation using prediction and early termination. In *Proceedings of International Conference on Image Processing, ICIP'02*, vol. 2, pp. 285–288, Rochester, USA.

[38] L.-P. Chau, Y. Liang, Y.-P. Tan (2001). Motion vector re-estimation for fractional-scale video transcoding. In *Proceedings of IEEE International Conference on Information Technology: Coding and Computing, ITCC'01*, pp. 212–215, Las Vegas, USA.

[39] M.-J. Chen, M.-C. Chu, C.-W. Pan (2002). Efficient motion-estimation algorithm for reduced frame-rate video transcoder. *IEEE Transactions on Circuits and Systems for Video Technology* **12**(4):269–275.

[40] Y.-F. Chen et al. (2002). Personalized multimedia services using a mobile service platform. In *Proceedings of Conference on Wireless Communications and Networking, WCNC2002*, vol. 2, pp. 918–925.

[41] R. T. Collins, A. J. Lipton, H. Fujiyoshi, T. Kanade (2001). Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE* **89**(10):1456–1477.

[42] D. Comaniciu, V. Ramesh, P. Meer (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(5):564–577.

[43] G. J. Conklin, S. S. Hemami (1999). A comparison of temporal scalability techniques. *IEEE Transactions on Circuits and Systems for Video Technology* **9**(6):909–919.

[44] P. Correia, F. Pereira (2000). Objective evaluation of relative segmentation quality. In *Proceedings of IEEE International Conference on Image Processing, ICIP'00*, vol. 1, pp. 308–311, Vancouver, Canada.

[45] I. J. Cox, S. Hingorani (1996). An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(2):138–150.

[46] R. Cucchiara, C. Grana, A. Prati (2002). A framework for semantic video transcoding. In *Proceedings Atti del Workshop 'Percezione e Visione delle Macchine' (in conjunction with AIIA 2002)*, Siena, Italy.

[47] R. Cucchiara, C. Grana, A. Prati (2002). Semantic transcoding for live video server. In *Proceedings of Tenth ACM International Conference on Multimedia*, pp. 223–226, Juan Les Pins, France.

[48] R. Cucchiara, C. Grana, A. Prati (2003). Semantic video transcoding using classes of relevance. *International Journal of Image and Graphics* **3**(1):145–169.

[49] R. Cucchiara, C. Grana, A. Prati, R. Vezzani (2003). Computer vision techniques for PDA accessibility of in-house video surveillance. In *ACM SIGMM International Workshop on Video Surveillance*, pp. 87–97.

[50] Y. Cui, Q. Huang (1997). Character extraction of license plates from video. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'97*, pp. 502–507, San Juan, Puerto Rico.

[51] R. Dahyot, P. Charbonnier, F. Heitz (2001). Unsupervised statistical detection of changing objects in camera-in-motion video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'01*, vol. 1, pp. 638–641, Thessaloniki, Greece.

[52] J. P. David A. Forsyth (2002). *Computer Vision: A Modern Approach.* Prentice Hall, Upper Saddle River, USA.

[53] L. S. Davis (1975). A survey of edge detection techniques. *Computer Graphics and Image Processing* **4**(3):248–270.

[54] G. de los Reyes, A. R. Reibman, S.-F. Chang, J. C.-I. Chuang (2000). Error-resilient transcoding for video over wireless channels. *IEEE Journal on Selected Areas in Communications* **18**(6):1063–1074.

[55] K. deok Seo, J. kyoon Kim (2002). Motion vector refinement for video downsampling in the DCT domain. *IEEE Signal Processing Letters* **9**(11):356–359.

[56] M. V. der Schaar, Y. Chen, H. Radha (1999). *Adaptive Quantization Modes for Fine-Granular Scalability.* Tech. Rep. ISO/IEC M4938, ISO/IEC JTC 1/SC29/WG11.

[57] S. Dogan et al. (2002). Error-resilient video transcoding for robust internetwork communications using GPRS. *IEEE Transactions on Circuits and Systems for Video Technology* **12**(6):453–464.

[58] M. Domański, S. Maćkowiak, Ł. Błasak, A. Łuczak (2002). Efficient hybrid video coders with spatial and temporal scalability. In *Proceedings of IEEE International Conference on Multimedia and Expo, ICME'02*, vol. 1, pp. 133–136, Lausanne, Switzerland.

[59] E. Dubois (1995). The sampling and reconstruction of time-varying imagery with applications in video systems. In T. S. Rzeszewski (ed.), *Digital Video – Concepts and Applications Across Industries*, pp. 5–25, IEEE Press, London, UK.

[60] M.-P. Dubuisson, A. K. Jain (1994). A modified Hausdorff distance for object matching. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, ICPR'94*, vol. 1, pp. 566–568, Jerusalem, Israel.

[61] F. Dufaux, J. Konrad (2000). Efficient, robust, and fast global motion estimation for video coding. *IEEE Transactions on Image Processing* **9**(3):497–501.

[62] R. Dugad, N. Ahuja (2003). A scheme for spatial scalability using nonscalable encoders. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(10):993–999.

[63] I. P. Duncumb, P. F. Gadd, G. Wu, J. L. Alty (2004). *Visual Radio: Should We Paint Pictures with Words, or Pictures?* Tech. rep., Loughborough University.

[64] T. Echigo et al. (2001). Personalized delivery of digest video managed on MPEG–7. *Proceedings of International Conference on Information Technology: Coding and Computing* pp. 216–220.

[65] J. H. Elder, A. Krupnik, L. A. Johnston (2003). Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(6):661–674.

[66] A. S. Elfishavy, S. B. Kesler, A. S. Abutaleb (1991). Adaptive algorithms for change detection in image sequence. *Signal Processing* **23**(2):179–191.

[67] C. E. Erdem, B. Sankur (2000). Performance evaluation metrics for object-based video segmentation. In *Proceedings of X European Signal Processing Conference, EUSIPCO'00*, vol. 2, pp. 917–920, Tampere, Finland.

[68] M. D. Fairchild (1997). *Color Appearance Models.* Addison-Wesley, Reading, USA.

[69] T.-J. Fan, G. Medioni, R. Nevatia (1989). Recognizing 3-D objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**(1):1140–1156.

[70] O. Faugeras (1993). *Three-Dimensional Computer Vision (Artificial Intelligence).* MIT Press, New Jersey, USA.

[71] L. Favalli, A. Mecocci, F. Moschetti (2000). Object tracking for retrieval applications in MPEG–2. *IEEE Transactions on Circuits and Systems for Video Technology* **10**(3):427–432.

[72] M. N. Fesharaki, G. R. Hellenstrand (1993). *Real-Time Color Image Segmentation.* Tech. Rep. SCS&E 9316, University of New South Wales.

[73] R. B. Fisher (1999). *Change Detection in Color Images.* Tech. rep., Edinburgh University.

[74] E. Fossbakk, P. Manzanares, J. L. Yago, A. Perkis (2001). An MPEG–21 framework for streaming media. In *Proceedings of IEEE Workshop on Multimedia Signal Processing 2001*, pp. 147–152.

[75] N. Friedman, S. Russell (1997). Image segmentation in video sequences: a probabilistic approach, uai'97. In *Proceedings of 13th Conference on Uncertaintly in Artificial Intelligence*, pp. 1–3.

[76] K.-T. Fung, Y.-L. Chan, W.-C. Siu (2002). New architecture for dynamic frame-skipping transcoder. *IEEE Transactions on Image Processing* **11**(8):886–900.

[77] S. Furui, T. Kikuchi, Y. Shinnaka, C. Hori (2004). Speech-to-text and speech-to-speech summarization of spontaneous speech. *IEEE Transactions on Speech and Audio Processing* **12**(4):401–408.

[78] W. F. Gardner, D. T. Lawton (1996). Interactive model-based vehicle tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(11):1115–1121.

[79] D. Geiger, A. Gupta, L. A. Costa, J. Vlontzos (1995). Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(3):294–302.

[80] B. Girod (1993). What's wrong with mean-squared error. In A. B. Watson (ed.), *Digital Images and Human Vision*, pp. 207–220, MIT Press, Cambridge, USA.

[81] J. Gllavata, R. Ewerth, B. Freisleben (2003). A robust algorithm for text detection in images. In *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, ISPA'03*, vol. 2, pp. 611–616, Rome, Italy.

[82] C. Gu, M.-C. Lee (1998). Semiautomatic segmentation and tracking of semantic video objects. *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5):572–584.

[83] B. Günsel, A. M. Tekalp, P. J. L. van Beek (1998). Content-based access to video objects: Temporal segmentation, visual summarization, and feature extraction. *Signal Processing* **66**(2):261–280.

[84] F. Halsall (2001). *Multimedia Communications: Applications, Networks, Protocols and Standards*. Addison-Wesley, Reading, USA.

[85] A. R. Hanson, E. M. Riseman (1978). Segmentation of natural scenes. In A. Hanson, E. Riseman (eds.), *Computer Vision Systems*, pp. 129–164, Academic Press, San Diego, USA.

[86] G. Harit, S. Chaudhury, G. Garg, P. K. Sharma (2003). A framework for video representation and transcoding using appearance spaces. In *Proceedings of International Conference on Multimedia and Expo, ICME'03*, vol. 3, pp. 593–596, Baltimore, USA.

[87] S. Hecht (1928). The relation between visual acuity and illumination. *The Journal of General Physiology* **11**(3):255–281.

[88] Z. M. Hefed (1999). Object tracking. *IEEE Potentials* **18**(3):10–13.

[89] A. Hekstra et al. (2002). PVQM – a perceptual video quality measure. *Signal Processing: Image Communication* **17**(10):781–798.

[90] S. L. Horowitz, T. Pavlidis (1976). Picture segmentation by a tree traversal algorithm. *Journal of the ACM* **23**(2):368–388.

[91] P. V. C. Hough (1962). A method and means for recognizing complex patterns. US Patent 3,069,654.

[92] R. L. Hsu, M. Abdel-Mottaleb, A. Jain (2002). Face detection on color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5):696–706.

[93] Y. Z. Hsu, H.-H. Nagel, G. Reckers (1984). New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics, and Image Processing* **26**(1):73–106.

[94] Q. Hu, S. Panchanathan (1996). A comparative evaluation of spatial scalability techniques in the compressed domain. In *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering, CCECE'96*, vol. 1, pp. 474–477, Calgary, Canada.

[95] Q. Hu, S. Panchanathan (1998). Image/video spatial scalability in compressed domain. *IEEE Transactions on Industrial Electronics* **45**(1):23–31.

[96] W. Hu, T. Tan, L. Wang, S. Maybank (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics* **34**(3):334–352.

[97] W. Hu et al. (2004). Traffic accident prediction using 3-D model-based vehicle tracking. *IEEE Transactions on Vehicular Technology* **53**(3):677–694.

[98] K.-L. Huang et al. (2002). A frame-based MPEG characteristics extraction tool and its applications in video transcoding. *IEEE Transactions on Consumer Electronics* **48**(3):522–532.

[99] Y. Huang, T. S. Huang (2002). Model-based human body tracking. In *Proceedings of 16th International Conference on Pattern Recognition, ICPR'02*, vol. 1, pp. 552–555, Quebec, Canada.

[100] R. W. G. Hunt (1999). Why is black and white so important in colour? In L. W. MacDonald, M. R. Luo (eds.), *Colour Imaging*, pp. 3–15, Wiley, Chichester, UK.

[101] J. Illingworth, J. Kittler (1988). Survey of the Hough transform. *Computer Vision, Graphics, and Image Processing* **44**(1):87–116.

[102] M. Irani, P. Anandan (1998). Video indexing based on mosaic representations. *Proceedings of the IEEE* **86**(5):905–921.

[103] M. Isard, A. Blake (1996). Contour tracking by stochastic propagation of conditional density. In *Proceedings of European Conference on Computer Vision*, vol. 1, pp. 343–356, Cambridge, UK.

[104] I. A. Ismaili, D. F. Gillies (1994). Colour image segmentation using regression analysis in RGB space. *Machine Graphics & Vision* **3**(1/2):373–384.

[105] ISO/IEC (1993). *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video.* Tech. Rep. ISO/IEC 11172-2:1993, ISO/IEC JTC 1/SC29/WG11.

[106] ISO/IEC (1997). *Bitplane Coding of DCT Coefficients.* Tech. Rep. ISO/IEC MPEG97/M2691, ISO/IEC JTC 1/SC29/WG11.

[107] ISO/IEC (2000). *Information technology – Generic coding of moving pictures and associated audio information: Video.* Tech. Rep. ISO/IEC 13818-2:2000, ISO/IEC JTC 1/SC29/WG11.

[108] ISO/IEC (2001). *Information Technology – Coding of Audio-Visual Objects – Part 2: Visual, 2nd Ed.* Tech. Rep. ISO/IEC FDIS 14496-2, ISO/IEC JTC 1/SC29/WG11.

[109] ISO/IEC (2001). *Information Technology – Coding of Audio-Visual Objects – Part 5: Reference software.* Tech. Rep. ISO/IEC 14496-5:2001/FPDAM2, ISO/IEC JTC 1/SC29/WG11.

[110] ISO/IEC (2001). *Information Technology – Multimedia Content Description Interface – Part 3: Visual.* Tech. Rep. ISO/IEC FDIS 15938-3, ISO/IEC JTC 1/SC29/WG11.

[111] ISO/IEC (2001). *Information Technology – Multimedia Content Description Interface – Part 5: Multimedia Description Schemes.* Tech. Rep. ISO/IEC FDIS 15938-5, ISO/IEC JTC 1/SC29/WG11.

[112] ISO/IEC (2004). *Information Technology – Multimedia Framework (MPEG–21) – Part 7: Digital Item Adaptation.* Tech. Rep. ISO/IEC FDIS 21000-7, ISO/IEC JTC 1/SC29/WG11.

[113] L. Itti (2004). Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing* **13**(10):1304–1318.

[114] ITU (1998). *H.263: Video Coding for Low Bitrate Communications.* Tech. rep., ITU-T Recommendation.

[115] ITU (1999). *Subjective Video Quality Assessment Methods for Multimedia Applications.* Tech. Rep. P.910, ITU-T Recommandation.

[116] ITU (2002). *Methodology for the Subjective Assessment of the Quality of Television Pictures.* Tech. Rep. BT.500-11, ITU-R Recommandation.

[117] B. Jung, G. S. Sukhatme (2004). Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *Proceedings of International Conference on Intelligent Autonomous Systems*, pp. 980–987, Amsterdam, NL.

[118] K. Jung, K. I. Kim, A. K. Jain (2004). Text information extraction in images and video: a survey. *Pattern Recognition* **37**(5):977–997.

[119] R. E. Kalman (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering* **82**:35–45.

[120] G. Karlsson, M. Vetterli (1988). Three-dimensional subband coding of video. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP'88*, vol. 2, pp. 1100–1103, New-York, USA.

[121] M. Kass, A. Witkin, D. Terzopoulos (1987). Snakes: Active contour models. *International Journal of Computer Vision* **1**(4):321–331.

[122] H. Katata, N. Ito, H. Kusao (1997). Temporal-scalable coding based on image content. *IEEE Transactions on Circuits and Systems for Video Technology* **7**(1):52–59.

[123] J.-G. Kim, Y. Wang, S.-F. Chang (2003). Content-adaptive utility-based video adaptation. In *Proceedings of International Conference on Multimedia and Expo, ICME'03*, vol. 3, pp. 281–284, Baltimore, USA.

[124] T. Kim, J. S. Choi (2002). Content-based video transcoding in compressed domain. *Signal Processing: Image Communication* **17**:497–507.

[125] T. Kim, J. H. Han (2001). Model-based discontinuity evaluation in the DCT domain,. *Signal Processing* **81**(4):871–882.

[126] D. Koller, K. Danilidis, H. Nagel (1993). Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision* **10**(3):257–281.

[127] J.-S. Lee, K.-Y. Rhee, S.-D. Kim (2001). Moving target tracking algorithm based on the confidence measure of motion vectors. In *Proceedings of International Conference on Image Processing, ICIP'01*, vol. 1, pp. 369–372, Thessaloniki, Greece.

[128] K. Lee et al. (2001). Perception-based image transcoding for universal multimedia access. In *Proceedings of 2001 IEEE International Conference on Image Processing, ICIP'01*, vol. 2, pp. 475–478.

[129] S. Lee, M. S. Pattichis, A. C. Bovik (2002). Foveated video quality assessment. *IEEE Transactions on Multimedia* **4**(1):129–132.

[130] Z. Lei, N. D. Georganas (2002). Rate adaptation transcoding for precoded video streams. In *Proceedings of Tenth ACM International Conference on Multimedia*, pp. 127–136, Juan Les Pins, France.

[131] Z. Lei, N. D. Georganas (2004). Adaptive video transcoding and streaming over wireless channels. *The Journal of Systems and Software* **To appear**.

[132] C.-S. Li, R. Mohan, J. R. Smith (1998). Multimedia content description in the infopyramid. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, vol. 6, pp. 3789–3792, Seattle, USA.

[133] W. Li (2001). Overview of fine granularity scalability in MPEG–4 video standard. *IEEE Transactions on Circuits and Systems for Video Technology* **11**(3):301–317.

[134] X. R. Li, Y. Bar-Shalom (1996). Tracking in clutter with nearest neighbor filters: Analysis and performance. *IEEE Transactions on Aerospace and Electronic Systems* **32**(3):995–1010.

[135] Y. Liang, Y.-P. Tan (2001). A new content-based hybrid video transcoding method. In *Proceedings of IEEE International Conference on Image Processing, ICIP'01*, vol. 1, pp. 429–432, Thessaloniki, Greece.

[136] R. Lillestrand (1972). Techniques for change detection. *IEEE Transactions on Computers* **21**(7):654–659.

[137] Y. Liow, T. Pavlidis (1988). Enhancements of the split-and-merge algorithm for image segmentation. In *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1567–1572, Philadelphia, USA.

[138] T.-L. Liu, H.-T. Chen (2004). Real-time tracking using trust-region methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(3):397–402.

[139] F. Long, D. Feng, H. Peng, W.-C. Siu (2001). Extracting semantic video objects. *IEEE Magazine on Computer Graphics and Applications* **21**(1):48–55.

[140] R. D. A. Lotufo, W. D. F. D. Silva, A. X. Falcao, A. C. F. Pessoa (1988). Morphological image segmentation applied to video quality assessment. In *Proceedings of IEEE International Symposium on Computer Graphics, Image Processing, and Vision, SIBGRAPI'98*, pp. 468–475, Rio de Janeiro, Brazil.

[141] D. G. Lowe (1989). Organization of smooth image curves at multiple scales. *International Journal of Computer Vision* **1**:119–130.

[142] L. Lucchese, S. K. Mitra (2001). Color image segmentation: A state-of-the-art survey. *Proceedings of Indian National Science Academy (INSA-A)* **67**(2):207–221.

[143] W. Y. Lum, F. C. M. Lau (2002). On balancing between transcoding overhead and spatial consumption in content adaptation. In *Proceedings of 8th Annual International Conference on Mobile Computing and Networking*, pp. 239–250, Glasgow, Scotland.

[144] S. Mallat (1999). *A Wavelet Tour of Signal Processing.* Academic Press, San Diego, USA, 2$^{\text{nd}}$ edn.

[145] X. Marichal, P. Villegas (2000). Objective evaluation of segmentation masks in video sequences. In *Proceedings of X European Signal Processing Conference, EUSIPCO'00*, vol. 4, pp. 2193–2196, Atlanta, USA.

[146] F. Marqués, J. Llach (1998). Tracking of generic objects for video object generation. In *Proceedings of IEEE International Conference on Image Processing, ICIP'98*, vol. 3, pp. 628–632, Chicago, USA.

[147] A. Martelli (1972). Edge detection using heuristic search methods. *Computer Graphics and Image Processing* **1**:169–182.

[148] E. Mazor, A. Averbuch, Y. Bar-Shalom, J. Dayan (1998). Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems* **34**(1):103–123.

[149] R. Mech, M. Wollborn (1998). A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera. *Signal Processing* **66**:203–217.

[150] N. Merhav, V. Bhaskaran (1996). A transform domain approach to spatial domain image scaling. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'96*, vol. 4, pp. 2403–2406, Atlanta, USA.

[151] D. Meyer, J. Posl, H. Niemann (1998). Gait classification with HMMs for trajectories of body parts extracted by mixture densities. In *Proceedings of British Machine Vision Conference*, pp. 459–468, Southampton, UK.

[152] F. Meyer, S. Beucher (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation* **1**:21–46.

[153] R. Mohan, J. R. Smith, C.-S. Li (1999). Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia* **1**(1):104–114.

[154] R. Mohan, J. R. Smith, C.-S. Li (1999). Content adaptation framework: Bringing the internet to information appliances. In *Proceedings of IEEE Global Telecommunications Conference, Globecom'99*, vol. 4, pp. 2015–2021, Amsterdam, NL.

[155] J. Monaco (1981). *How to Read a Movie.* Oxford Press, Oxford, UK.

[156] N. Morgan, H. Bourlard (1995). Continuous speech recognition. *IEEE Signal Processing Magazine* **12**(3):24–42.

[157] K. Nagao (2003). *Digital Content Annotation and Transcoding.* Artech House Publishers, Norwood, USA.

[158] K. Nagao, Y. Shirai, K. Squire (2001). Semantic annotation and transcoding: Making Web content more accessible. *IEEE Multimedia* **8**(2):69–81.

[159] A. Nagasaka, Y. Tanaka (1992). Automatic video indexing and full-video search for object appearances. In *Proceedings of 2nd Working Conference on Visual Database Systems*, pp. 119–133.

[160] Y. Nakajima, H. Hori, T. Kanoh (1995). Rate conversion of MPEG coded video by re-quantization process. In *Proceedings of IEEE International Conference on Image Processing, ICIP'95*, vol. 3, pp. 408–411, Washington, USA.

[161] S. Nakamura (2000). HMM-based transmodal mapping from audio speech to talking faces. In *Proceedings of IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing X*, vol. 1, pp. 33–42.

[162] K. Nickels, S. Hutchinson (2001). Model-based tracking of complex articulated objects. *IEEE Transactions on Robotics and Automation* **17**(1):28–36.

[163] N. J. Nilsson (1982). *Principle of Artificial Intelligence.* Springer, London, UK.

[164] H. Ning, L. Wang, W. Hu, T. Tan (2002). Model-based tracking of human walking in monocular image sequences. In *Proceedings of IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, TENCON'02*, vol. 1, pp. 537–540, Beijing, China.

[165] F. Oberti, L. Marcenaro, C. S. Regazzoni (2002). Real-time change detection methods for video-surveillance systems with mobile camera. In *Proceedings of XI European Signal Processing Conference, EUSIPCO'02*, Toulouse, France.

[166] J. Oh, P. Sankuratri (2002). Automatic distinction of camera and object motions in video sequences. In *Proceedings of IEEE International Conference on Multimedia and Expo, ICME'02*, vol. 1, pp. 81–84, Lausanne, Switzerland.

[167] S. Olsson, M. Stroppiana, J. Baïna (1997). Objective methods for assessment of video quality: State of the art. *IEEE Transactions on Broadcasting* **43**(4):487–495.

[168] E. Ong et al. (2004). Visual distortion assessment with emphasis on spatially transitional regions. *IEEE Transactions on Circuits and Systems for Video Technology* **14**(4):559–566.

[169] A. V. Oppenheim, R. W. Schafer (1989). *Discrete-Time Signal Processing.* Prentice Hall Signal Processing Series. Prentice Hall, Upper Saddle River, USA.

[170] J. Owens, A. Hunter (2000). Application of the self-organizing map to trajectory classification. In *Proceedings of Third IEEE International Workshop on Visual Surveillance*, pp. 77–83.

[171] N. Paragios, R. Deriche (2000). Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(3):266–279.

[172] J.-I. Park, S. Inoue, Y. Iwadate (1998). Estimating camera parameters from motion vectors of digital video. In *Proceedings of IEEE Second Workshop on Multimedia Signal Processing*, pp. 105–110.

[173] F. Pereira, I. Burnett (2003). Universal multimedia experiences for tomorrow. *IEEE Signal Processing Magazine* **20**(2):63–73.

[174] D. N. Perkins (1983). Why the human perceiver is a bad machine. In J. Beck, B. Hope, A. Rosenfeld (eds.), *Human and Machine Vision*, vol. 8 of *Notes and Reports in Computer Science and Applied Mathematics*, pp. 341–364, Academic Press, San Diego, CA.

[175] A. Perkis et al. (2001). Universal multimedia access from wired and wireless systems. *Birkhauser Boston Transactions on Circuits, Systems and Signal Processing* **10**(3):387–402.

[176] A. Perkis et al. (2002). A streaming media engine using digital item adaptation. In *Proceedings of IEEE Workshop on Multimedia Signal Processing 2002*, pp. 73–76.

[177] A. C. F. Pessoa, A. X. F. ao, A. E. F. da Silva, R. M. Nishihara (1998). Video quality assessment using objective parameters based on image segmentation. In *Proceedings of SBT/IEEE International Telecommunications Symposium, ITS'98*, vol. 2, pp. 498–503, São Paulo, Brazil.

[178] N. Peterfreund (1999). Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(6):564–569.

[179] J. W. Picone (1993). Signal modeling techniques in speech recognition. *Proceedings of the IEEE* **81**(9):1215–1247.

[180] F. Pighin, R. Szeliski, D. H. Salesin (1999). Resynthesizing facial animation through 3d model-based tracking. In *Proceedings of Seventh IEEE International Conference on Computer Vision, ICCV'99*, pp. 143–150, Kerkyra, Greece.

[181] G. S. Pingali, A. Opalach, Y. D. Jean, I. B. Carlbom (2002). Instantly indexed multimedia databases of real world events. *IEEE Transactions on Multimedia* **4**(2):269–282.

[182] R. L. Popp, K. R. Pattipati, Y. Bar-Shalom (2001). m-best S-D assignment algorithm with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems* **37**(1):22–39.

[183] H. Radha, M. van der Schaar, Y. Chen (2001). The MPEG–4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Transactions on Multimedia* **3**(1):53–68.

[184] R. J. Radke, S. Andra, O. Al-Kofahi, B. Roysam (2005). Image change detection algorithms: A systematic survey. *To appear in IEEE Transactions on Image Processing* .

[185] C. Rasmussen, G. D. Hager (2001). Probabilistic data assiciation methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(6):560–576.

[186] D. B. Reid (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control* **24**(6):843–854.

[187] R. Rickitt (2000). *Special Effects: The History and Technique.* Watson-Guptill Publications, New York, USA.

[188] T. W. Ridler, S. Calvard (1978). Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man and Cybernetics* **8**(8):630–632.

[189] G. Rigoll, H. Breit, F. Wallhoff (2003). Robust tracking of persons in real-world scenarios using a statistical computer vision approach. *Image and Vision Computing* **22**(7):571–582.

[190] A. W. Rix, A. Bourret, M. P. Hollier (1999). Models of human perception. *BT Technology Journal* **17**(1):24–34.

[191] Y. Rosenberg, M. Wermann (1998). Real-time object tracking from a moving video camera: A software approach on a PC. In *Proceedings of Fourth IEEE Workshop on Applications of Computer Vision, WACV'98*, pp. 238–239, Princeton, USA.

[192] A. Rosenfeld (1970). Connectivity in digital pictures. *Journal of the ACM* **17**(1):146–160.

[193] A. Rosenfeld (1973). Arcs and curves in digital pictures. *Journal of the ACM* **20**(1):81–87.

[194] A. Rosenfeld, R. A. Hummel, S. Zucker (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics* **6**:420–433.

[195] A. Rosenfeld, J. L. Pfaltz (1966). Sequential operations in digital picture processing. *Journal of the ACM* **13**(4):471–494.

[196] P. K. Sahoo, S. Soltani, A. K. C. Wong, Y. C. Chen (1988). Survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing* **41**(2):233–260.

[197] E. Salvador (2004). *Shadow Segmentation and Tracking in Real-World Conditions.* Ph.D. thesis, No. 3076, Swiss Federal Institute of Technology (EPFL).

[198] O. Sanchez, F. Dibos (2004). Displacement following of hidden objects in a video sequence. *International Journal of Computer Vision* **57**(2):91–105.

[199] P. Schelkens et al. (2004). A comparative study of scalable video coding schemes utilizing wavelet technology. *Proceedings of SPIE* **5266**:147–156.

[200] A. Secker, D. Taubman (2001). Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting. In *Proceedings of IEEE International Conference on Image Processing, ICIP'02*, vol. 2, pp. 1029–1032, Rochester, USA.

[201] A. Secker, D. Taubman (2002). Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation. In *Proceedings of IEEE International Conference on Image Processing, ICIP'02*, vol. 3, pp. 749–752, Rochester, USA.

[202] T. Shanableh, M. Ghanbari (2000). Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats. *IEEE Transactions on Multimedia* **2**(2):101–110.

[203] T. Shanableh, M. Ghanbari (2001). Transcoding architectures for DCT-domain heterogeneous video transcoding. In *Proceedings of IEEE International Conference on Image Processing, ICIP'01*, vol. 1, pp. 433–436, Thessaloniki, Greece.

[204] T. Shanableh, M. Ghanbari (2003). Hybrid DCT/pixel domain architecture for heterogeneous video transcoding. *Signal Processing: Image Communication* **18**(8):601–620.

[205] C. E. Shannon (1948). A mathematical theory of communication. *Bell System Technical Journal* **27**(4):379–423, 623–656.

[206] D.-G. Sim, O.-K. Kwon, R.-H. Park (1999). Object matching algorithms using robust Hausdorff distance measures. *IEEE Transactions on Image Processing* **8**(3):425–429.

[207] W. Skarbek, A. Koschan (1994). *Colour Image Segmentation – A Survey.* Tech. Rep. 94-32, Technische Universität Berlin.

[208] K. Skifstad, R. Jain (1989). Illumination independent change detection for real world image sequences. *Computer Vision, Graphics, and Image Processing* **46**(3):387–399.

[209] J. R. Smith, R. Mohan, C.-S. Li (1999). Scalable multimedia delivery for pervasive computing. In *Proceedings of Seventh ACM International Conference on Multimedia*, pp. 131–140, Orlando, USA.

[210] P. Smits, A. Annoni (2000). Towards specification-driven change detection. *IEEE Transactions on Geoscience and Remote Sensing* **38**(3):1484–1488.

[211] M. Sonka, V. Hlavac, R. Boyle (1998). *Image Processing, Analysis and Machine Vision.* PWS Publishing, Pacific Growe, USA, 2nd edn.

[212] O. Steiger, A. Cavallaro, T. Ebrahimi (2002). MPEG–7 description of generic video objects for scene reconstruction. In *Proceedings of SPIE Conference on Visual Communications and Image Processing, VCIP'02*, vol. 4671, pp. 947–958, San Jose, USA.

[213] O. Steiger, A. Cavallaro, T. Ebrahimi (2004). *MPEG–7 Description for Scalable Video Reconstruction.* Tech. Rep. 09.04, Swiss Federal Institute of Technology (EPFL).

[214] O. Steiger, A. Cavallaro, T. Ebrahimi (2005). Real-time generation of annotated video for surveillance. In *Proceedings of IEE Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2005.* (To appear).

[215] O. Steiger, D. Marimón, T. Ebrahimi (2003). MPEG-based personalized content delivery. In *Proceedings of IEEE International Conference on Image Processing, ICIP'03*, vol. 3, pp. 45–48, Barcelona, Spain.

[216] H. Sun, W. Kwok (1995). MPEG video coding with temporal scalability. In *Proceedings of IEEE International Conference on Communications, ICC'95*, vol. 3, pp. 1742–1746, Seattle, USA.

[217] H. Sun, A. Vetro, K. Asai (2003). Resource adaptation based on MPEG–21 usage environment descriptions. In *Proceedings of 2003 IEEE International Symposium on Circuits and Systems, ISCAS'03*, vol. 2, pp. 536–539.

[218] Y.-P. Tan, Y. Liang, H. Sun (2003). On the methods and performances of rational downsizing video transcoding. *Signal Processing: Image Communication* **19**(1):47–65.

[219] R. I. Taylor, P. H. Lewis (1992). Colour image segmentation using boundary relaxation. In *Proceedings of 11th International Conference on Pattern Recognition, IAPR'92*, vol. 3, pp. 721–724, Den Haag, NL.

[220] D. Terzopoulos, R. Szeliski (1992). Tracking with Kalman snakes. In A. Blake, A. Yuille (eds.), *Active Vision*, pp. 3–20, MIT Press, Boston, USA.

[221] D. Toth, T. Aach, V. Metzler (2000). Illumination-invariant change detection. In *Proceedings of 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 170–177, Austin, USA.

[222] M. J. Tovée (1996). *An Introduction to the Visual System.* Cambridge University Press, Cambridge, UK.

[223] A. Tremeau, N. Borel (1997). A region growing and merging algorithm to color segmentation. *Pattern Recognition* **30**(7):1191–1204.

[224] Y. Tsaig, A. Averbuch (2002). Automatic segmentation of moving objects in video sequences: a region labeling approach. *IEEE Transactions on Circuits and Systems for Video Technology* **12**(7):597–612.

[225] N. Ueda, K. Mase (1992). Tracking moving contours using energy-minimizing elastic contour models. In *Proceedings of the Second European Conference on Computer Vision, ECCV'92*, vol. 588, pp. 453–457, Santa Margherita, Italy.

[226] H. Umeki, H. Mizutani (1996). Object recognition by dynamic link matching with multiple blob formation. In *Proceedings of International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, pp. 237–245, Venice, Italy.

[227] P. van Beek, J. R. Smith, T. Ebrahimi (2003). Metadata-driven multimedia access. *IEEE Signal Processing Magazine* **20**(2):40–52.

[228] M. van der Schaar, Y.-T. Lin (2001). Content-based selective enhancement for streaming video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'01*, vol. 2, pp. 977–980, Thessaloniki, Greece.

[229] M. van der Schaar, H. Radha (2000). A novel MPEG–4 based hybrid temporal-snr scalability for internet video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'00*, vol. 3, pp. 548–551, Vancouver, Canada.

[230] M. van der Schaar, H. Radha (2001). A hybrid temporal-snr fine-granular scalability for internet video. *IEEE Transactions on Circuits and Systems for Video Technology* **11**(3):318–331.

[231] M. van der Schaar, H. Radha (2002). Adaptive motion-compensation fine-granular-scalability (AMC-FGS) for wireless video. *IEEE Transactions on Circuits and Systems for Video Technology* **12**(6):360–371.

[232] A. Vetro, C. Christopoulos, T. Ebrahimi (2003). Universal multimedia access. *IEEE Signal Processing Magazine* **20**(2):16.

[233] A. Vetro, C. Christopoulos, H. Sun (2003). Video transcoding architectures and techniques: an overview. *IEEE Signal Processing Magazine* **20**(2):18–29.

[234] A. Vetro, H. Sun (2001). Encoding and transcoding multiple video objects with variable temporal resolution. In *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'01*, vol. 5, pp. 21–24, Sydney, Australia.

[235] A. Vetro, H. Sun (2001). Media conversion to support mobile users. In *Proceedings of Canadian Conference on Electrical and Computer Engineering 2001*, vol. 1, pp. 607–612.

[236] A. Vetro, H. Sun, Y. Wang (2000). Object-based transcoding for scalable quality of service. In *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'00*, vol. 4, pp. 17–20, Geneva, Switzerland.

[237] A. Vetro, H. Sun, Y. Wang (2001). Object-based transcoding for adaptable video content delivery. *IEEE Transactions on Circuits and Systems for Video Technology* **11**(3):387–401.

[238] L. Vincent, P. Soille (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(6):583–598.

[239] VQEG (2000). *Final Report from the Video Quality Experts Group on the Validation of Objective Models for Video Quality Assessment*. Tech. rep., Video Quality Experts Group.

[240] W3C (2001). *W3C Recommendation of the Synchronized Multimedia Integration Language (SMIL) 2.0*. Tech. Rep. W3C REC-smil20, W3C SYMM Working Group.

[241] W3C (2004). *Extensible Markup Language (XML) 1.0 (Third Edition)*. Tech. Rep. W3C Recommendation 04 February 2004, W3C Core Working Group.

[242] C.-N. Wang et al. (2003). FGS-based video streaming test-bed for MPEG–21 universal multimedia access with digital item adaptation. In *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems, ISCAS'03*, vol. 2, pp. II:364–367.

[243] D. Wang (1998). Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5):539–545.

[244] L. Wang, T. Tan, W. Hu, H. Ning (2003). Automatic gait recognition based on statistical shape analysis. *IEEE Transactions on Image Processing* **12**:1120–1131.

[245] Q. Wang et al. (2001). Fine-granularity spatially scalable video coding. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'01*, vol. 3, pp. 1801–1804, Salt Lake City, USA.

[246] Y. Wang, J.-G. Kim, S.-F. Chang (2003). Content-based utility function prediction for real-time MPEG–4 video transcoding. In *Proceedings of IEEE International Conference on Image Processing, ICIP'03*, vol. 1, pp. 189–192, Barcelona, Spain.

[247] Y. Wang, J. Ostermann, Y.-Q. Zhang (2001). *Video Processing and Communications*. Prentice Hall, Upper Saddle River, USA.

[248] J. C. H. Watkins, P. Dayan (1992). Q-learning. *Machine Learning* **8**:279–292.

[249] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra (2003). Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7):560–576.

[250] S. Winkler (2000). *Vision Models and Quality Metrics for Image Processing Applications*. Ph.D. thesis, No. 2313, Swiss Federal Institute of Technology (EPFL).

[251] W. Wolf (1996). Key frame selection by motion analysis. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP'96*, vol. 2, pp. 1228–1231, Atlanta, USA.

[252] M. Wollborn, I. Moccagatta, U. Benzler (2002). Natural video coding. In F. Pereira, T. Ebrahimi (eds.), *The MPEG–4 Book*, pp. 293–381, Prentica Hall, Upper Saddle River, USA.

[253] F. Wu, S. Li, Y. Q. Zhang (2000). DCT-prediction based progressive fine granularity scalability. In *Proceedings of IEEE International Conference on Image Processing, ICIP'00*, vol. 3, pp. 556–559, Vancouver, Canada.

[254] Y. Wu et al. (2003). Invariant feature extraction and biased statistical inference for video surveillance. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 284–289.

[255] H. S. Yang, S. U. Lee (1997). Split-and-merge segmentation employing thresholding technique. In *Proceedings of IEEE International Conference on Image Processing, ICIP'97*, vol. 1, pp. 239–242, Washington, USA.

[256] A. Yarbus (1967). *Eye Movements and Vision.* Plenum Press, New York, USA.

[257] M. Yeung, B. Yeo (1997). Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology* **7**(5):771–785.

[258] J. You, M.-T. Sun, C.-W. Lin (1999). Motion vector refinement for high-performance transcoding. *IEEE Transactions on Multimedia* **1**(1):30–39.

[259] J. Youn, M.-T. Sun (1999). A fast motion vector composition method for temporal transcoding. In *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'99*, vol. 4, pp. 243–247, Orlando, USA.

[260] J. Youn, M.-T. Sun (2000). Video transcoding with H.263 bit-streams. *Journal of Visual Communications and Image Representation* **11**(4):385–403.

[261] J. Youn, M.-T. Sun, J. Xing (1999). Video transcoder architectures for bit rate scaling of H.263 bit streams. In *Proceedings of Seventh ACM International Conference on Multimedia*, pp. 243–250, Orlando, USA.

[262] Z. Yu, H. Wu, S. Winkler, T. Chen (2002). Vision-model-based impairment metric to evaluate blocking artifacts in digital video. *Proceedings of the IEEE* **90**(1):154–169.

[263] Z. Yu, H. R. Wu (2000). Human visual system based objective digital video quality metrics. In *Proceedings of 5th IEEE International Conference on Signal Processing, WCCC-ICSP 2000*, vol. 2, pp. 1088–1095.

[264] R. Zabih, J. Miller, K. Mai (1993). Feature-based algorithms for detecting and classifying scene breaks. In *Proceedings of ACM Conference on Multimedia*, pp. 189–200, San Fransisco, USA.

[265] H. J. Zhang, J. Wu, D. Zhong, S. W. Smoliar (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition* **30**(4):643–658.

[266] T. Zhang, C. Tomasi (1999). Fast, robust, and consistent camera motion estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, ICPR'99*, vol. 1, pp. 164–170, Limerick, Ireland.

[267] J.-W. Zhao, P. Wang, C.-Q. Liu (2002). An object tracking algorithm based on occlusion mesh models. In *Proceedings of First International Conference on Machine Learning and Cybernetics*, pp. 288–292, Beijing, China.

[268] D. Zhong, H. J. Zhang, S. Chang (1996). Clustering methods for video browsing and annotation. In *Proceedings of SPIE International Symposium on Electrical Imaging: Storage and Retrieval for Image and Video Databases*, pp. 239–246, San Jose, USA.

[269] K. Zhou, Q. Dai, J. Wu, G. Er (2002). Fast tracking of semantic video object based on motion prediction and subregion extraction. In *Proceedings of International Conference on Image Processing, ICIP'02*, vol. 3, pp. 621–624, Rochester, USA.

Olivier STEIGER
Rue du Maupas 19a
1004 Lausanne
Mobile: ++41 79 772.91.85
E-mail: olivier.steiger@epfl.ch

Age: 28
Swiss
Single

EDUCATION

2001 - 2005 **PhD in image and video processing,** EPFL, Lausanne, Switzerland
Dissertation title: “Adaptive Video Delivery Using Semantics”

1996 - 2001 **Master in electrical & computer engineering,** EPFL, Lausanne, Switzerland
Dissertation title: “Smart Camera for MPEG-7”

1992 - 1996 **Maturity (science),** Collège Ste-Croix, Fribourg, Switzerland

EXPERIENCE

2001 – present **Research Assistant,** Swiss Federal Institute of Technology Lausanne (EPFL)
- R&D on video analysis, description and transcoding; author of 9 refereed publications
- Responsible for EPFL contribution to EC-funded project *PERSEO*
- Co-responsible for EPFL contribution to EC-funded network of excellence *VISNET*
- Co-advisor of 6 MSc students
- Teaching assistant, lecture on image and video processing
- Participation in MPEG standardization activities
- Website design and administration

Fall 2003 **Visiting scholar** at Queen Mary, University of London

2003 **Organization of a conference and website design**
International Conference on Visual Communications and Image Processing (VCIP)

2002 **Tutorial lecture,** Engineering School Fribourg, Switzerland
Title: “MPEG Standards and the MPEG-7 Camera”

1999 **Practical training,** Vibro-Meter SA, Fribourg, Switzerland
Design, implementation and testing of an FPGA for avionic devices

1998 – 1999 **Exchange year** at Carnegie Mellon University, Pittsburgh, USA
Honor: entry on the *Dean’s List of Carnegie Institute of Technology*

LANGUAGES

German: mother tongue
French: second language (bilingual)
English: excellent, 620 points at TOEFL

COMPUTER SKILLS

Programming: C, C++, Visual Basic, Java, LaTeX, VHDL, Verilog
Applications: Matlab, LabView, MS-Office

PERSONAL INTERESTS

Music: plays the violin since the age of 10 years in various orchestras and ensembles, including the *Swiss Youth Symphony Orchestra* (SJSO), *Orchestre Symphonique Universitaire de Lausanne* (OSUL), and *Association Universitaire de Musique de Chambre Lausanne* (AUMC)
Theatre: member of the *Théâtre de l’Arbanel*, Fribourg, from 1995 to 2003