Detectie en representatie van bewegende objecten voor videobewaking

Detection and Representation of Moving Objects for Video Surveillance

Chris Poppe

Promotor: prof. dr. ir. R. Van de Walle
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. J. Van Campenhout
Faculteit Ingenieurswetenschappen
Academiejaar 2008 - 2009

UNIVERSITEIT
GENT

ISBN 978-90-8578-285-8
NUR 980, 983
Wettelijk depot: D/2009/10.500/43

Simplicity is the ultimate sophistication.

(Leonardo da Vinci (1452–1519))

Dankwoord

Met het neerleggen van dit proefschrift besluit ik een periode van meer dan vier jaar. Gedurende die tijd heb ik de kans gekregen om deel uit te maken van een bijzonder boeiende wereld, die van het wetenschappelijk onderzoek. Ongetwijfeld zullen de gebeurtenissen tijdens deze periode een grote invloed hebben op de rest van mijn leven. Bovendien kreeg ik de kans om mijn werk over heel de wereld te gaan presenteren en demonstreren. Verschillende plaatsen waar ik geweest ben en de mensen die ik daar heb ontmoet zullen me altijd bijblijven. Een aantal mensen verdienen mijn bijzondere dank voor de totstandkoming van dit proefschrift.

In de eerste plaats wil ik mijn promoter, prof. dr. ir. Rik Van de Walle, bedanken. Zijn inzet en passie heeft Multimedia Lab doen uitgroeien tot een (inter)nationaal gerespecteerde onderzoeksgroep. Ik wil hem bedanken voor de steun die ik van hem ontvangen heb in mijn onderzoeksbeslissingen en voor de vele kansen die me aangeboden werden.

Daarnaast wil ik ook de huidige en vroegere collega's van Multimedia Lab bedanken. Reeds toen ik mijn eerste stappen als thesisstudent in Multimedia Lab deed, viel me de vriendschappelijke omgang en aangename werksfeer op. Ik heb de kans gehad om met verschillende mensen heel nauw samen te werken. De kritische opmerkingen en suggesties die ik van hen gekregen heb, hebben de kwaliteit van mijn werk absoluut verhoogd. Verder wil ik ook Rita Breems en Ellen Lammens bedanken voor de administratieve ondersteuning die zeer welkom was.

Graag had ik een aantal collega's specifiek bedankt. Peter Lambert en Erik Mannens, voor het vertrouwen dat ik van hen kreeg in het opvolgen van projecten. Gaëtan Martens, Sarah De Bruyne en Tom Paridaens voor de nauwe samenwerking. Zij zijn verantwoordelijk voor de tot standkoming van grote delen van mijn werk.

Ik wens ook mijn familie te bedanken voor het voortdurende vertrouwen die ik van hen kreeg en de steun die ze me gegeven hebben. Mijn vrienden verdienen ook een woord van dank voor de ontelbare leuke momenten die we samen hebben mogen beleven.

Om af te sluiten wil ik ook mijn vrouw, Katrien, bedanken voor de schitterende jaren die we reeds samen beleefd hebben. Zij heeft mij steeds gesteund in al wat ik deed en vooral in mijn doctoraatsonderzoek. Onvoorspelbaar maar mooi, zo voorzie ik de toekomst die we samen nog hebben, een toekomst waar ik elke dag naar uitkijk.

Chris Poppe
12 februari 2009

Summary

The last decade, we have witnessed an explosion of video surveillance cameras and systems. Such a system monitors the video feeds of several cameras and is responsible for management, visualization and storage of the recorded data. With the number of video cameras within one system increasing, it became clear that human operators are no longer capable of monitoring the amount of captured data. Hence, the need for intelligent video analytics arose and together with the increase of cameras came the raise of efforts in the domain of automatic video analysis. This new branch within the computer vision research field has soon become a large research topic, steered by industrial and governmental funds.

Different applications have been deployed like crowd control, detection of intruders, theft, left luggage detection, etc. New use cases were introduced, which benefit of the computer vision algorithms e.g., in traffic control (incident detection, speeding, intersection management, congestion monitoring and so on) or retail (people counting, activity patterns).

Much research has been done on all aspects of an intelligent video surveillance system. Algorithms were developed to detect and segment moving objects like people and cars. Tracking algorithms find the temporal dependencies between objects in different frames or camera views. Trajectories and body shape are used to determine and predict the behaviour of detected objects. Face and licence plate recognition are examples of more advanced techniques working on the segmented images.

When considering such video surveillance applications, two main restrictions are made. **Speed** and **accuracy** are of uttermost importance. Almost all of the aforementioned use cases require a fast and accurate initial segmentation of moving objects. Although this is the first step, it is also one of the most difficult ones due to it's dependence on the statistics of the captured video data. The major problems in accurately detecting and segmenting moving objects

are noise, shadows, lighting changes, background movement and so on. Additionally, occlusion, differences in size, speed, and shape of the moving objects, stopped objects or moved background objects make it difficult to create a reliable moving object detection system.

When object detection is incorporated in large-scale video surveillance systems, means are necessary to define how the objects, or other events of interest, can be signalled, communicated and stored. The usage of standardized (XML-based) metadata has been proposed to create interoperability within one system. However, when trying to combine different surveillance systems, or modules of different vendors, the usage of different XML-based standards introduces again interoperability issues. Different standards generally use different constructs to represent the same concepts.

Our research is situated within the context of an intelligent surveillance system, where we focus on moving object detection and the usage of different XML-based metadata standards. Firstly, we present two novel background subtraction techniques to detect moving objects when using a static camera. The first technique works on uncompressed video data, while the second analyses video sequences compressed according to a recent video standard (H.264/AVC). In the latter, we use features available in the H.264/AVC-compressed video stream to detect the moving objects. Lastly, we investigate the problems that occur when trying to describe these detected objects by using standardized XML-based metadata formats. We solve these issues in the context of (personal) content management systems, since these can be seen as a generalization of video surveillance systems.

First, we give an in-depth explanation about moving object detection in the pixel domain. In this case, the input of the algorithms are uncompressed image sequences and the purpose is to find those pixels in the images that correspond to moving objects. Finding moving objects in the pixel domain is a well-studied problem and numerous algorithms have been proposed. One of the most popular group of methods are background subtraction techniques. During the surveillance of a scene, a background model is created and dynamically updated. For each new image, deviations of the pixel values from the background model are detected and used to classify the observations as belonging to background or foreground. We elaborate on the related work in this field, and enumerate a number of problems that current background subtraction systems suffer from.

We present a robust moving object detection technique. The proposed system is a multi-modal spatio-temporal background subtraction technique that finds

those pixels corresponding to moving objects.

The temporal background subtraction system uses a weighted mixture of models and is combined with a fast spatial image segmentation technique. The models consist of an average, an upper and lower threshold, a maximum difference with the last background value, and an illumination allowance based on photon noise statistics of the image. The problem of gradual illumination changes is well-known and a number of methods exist in the literature to deal with it, but at the cost of high additional complexity. By incorporating the difference with the last background value, we presented a very fast and efficient technique to deal with the problem of gradual illumination changes. New pixel values are compared with the models of the mixture to find a match. Consequently, a decision (classification as foreground or background) is made based on the weights of the models. The parameters of the models are continuously updated according to the newly captured images to follow the dynamics of the surveilled environment. Finally, a shadow removal scheme has been included to improve the detection results.

The processing speed of video analytics algorithms is very important, especially in the context of a video surveillance system. We present an intelligent analysis mask, which allows to reduce the number of pixels that need to be analyzed for each frame. Lastly, a fast image segmentation is introduced, which applies edge detection to find segments in the images. Photon noise statistics are incorporated to remove noisy edges and holes in the edges are filled. This segmentation is combined with the background subtraction.

To evaluate the proposed system, we compare the results with those of related works in the field. Different sequences are analyzed and the results (in number of correctly classified pixels) are presented. Visual examples are included to give the reader a better idea of the benefits of our system. Lastly, execution times are compared to show the gains in speed that our method achieves.

Most video surveillance systems incorporate an aspect of digital video and rely on advanced video coding formats for the efficient representation, transmission, and/or storage of digital video content. Hence a decoding step is needed, before algorithms for video analytics can be applied. Many research efforts have been done to apply those algorithms directly on the compressed video sequences. As such, not relying on the pixel information, but on the features created during the encoding process. In this dissertation, we focused on the H.264/AVC specification for digital video coding, which has become the de facto benchmark in the domain of digital video coding with respect to compression.

H.264/AVC is a rather new video standard, but the first cameras supporting this video format have already hit the surveillance market. A number of approaches exist to detect moving objects in the H.264/AVC compressed domain. One common aspect of these systems is the usage of motion vector information available in the compressed bitstreams. However, motion vectors are created from a coding point of view and, hence, do not necessarily represent the actual motion in a video sequence. As a result, these algorithms spend a large amount of processing to remove noisy motion vectors.

We present a novel moving object detection method, working in the H.264/AVC compressed domain, that disregards the motion vector information entirely. The proposed method relies on the degree of compression that an encoder can achieve for different parts of the frames. We make the assumption that an encoder can compress parts of the background well, while it has more difficulties to compress parts of moving objects. To show that our assumption holds, we present an analysis of the bit usage for different macroblocks in the compressed bitstream. These observations are consequently used to create a compressed domain technique that finds macroblocks assumed to correspond with moving objects. During execution, a background model is created based on the sizes (in bits) of the macroblocks. For new images, the sizes of the macroblocks are compared with this model to detect unusual large sizes. The corresponding macroblocks are assumed to correspond to moving objects and are spatially and temporally filtered. However, the usage of macroblocks can be too coarse, since these consist of 16×16 pixels. Consequently, we refined the algorithm up to blocks of 4×4 pixels by analysing sub-macroblocks of the detected macroblocks that lie on edges of moving objects.

As with our algorithm in the pixel domain, an extensive comparison is made with related work in this field. We show that our system outperforms other moving object detection techniques that rely on the motion vector information. Different challenging surveillance sequences are used for the comparison. Moreover, since the features we use, being the amount of compression for blocks in a frame, can be extracted from a compressed video very fast, we largely outperform the related work in terms of processing time. As such we can detect moving objects up to 20 times faster than the related work in this domain.

Additionally, we evaluate the influence of the encoder configuration on the detection performance of the algorithm. This is a study that is neglected by most of the related work in this domain. We show that different configurations, hence resulting in different compressed video streams, have an influence on the performance of the algorithm, but that we are less dependent on it compared

to the approaches based on motion vectors.

A last aspect that we cover in this dissertation is how to deal with interoperability issues and the lack to describe semantic knowledge when using different XML-based metadata standards. We briefly discuss how this problem arises in the context of video surveillance systems. Next, we elaborate on how the same problems occur in the management of (personal) content. As such, our research is focused on how to solve those issues when applied to personal content management systems.

We present a novel layered architecture that uses Semantic Web technologies to combine different metadata standards in the lower layer, while the higher layer consists of specific ontologies representing metadata that is relevant for personal content management systems. The metadata standards and relations between them are represented as OWL ontologies and we have created the first formal representation (using OWL) of the DIG35 image metadata standard. We show how it is incorporated in the framework and how it can be combined with other metadata standards, like MPEG-7. For evaluation purposes a general use case scenario is presented, consisting of the upload of a resource, importing of the existing metadata fields, adding an annotation according to a different metadata format, and using high-level metadata to retrieve the content. We show how our proposed system is more suited to solve this use case than related work in the field.

To conclude, in this dissertation different aspects of an intelligent video surveillance system were considered. The emphasis lays on the detection of moving objects, both in pixel and compressed domain. Finally, the usage of XML-based metadata has been studied and we presented methods to enhance the interoperability. We hope to have convinced the reader that our work is relevant for current and future multimedia systems and can be applied in a wide range of applications, being video surveillance or content management. The application of H.264/AVC in the video surveillance market is only just beginning to bloom. Also, the usage of metadata to describe events will become even more important with the increase of cameras and systems in the future. Lastly, tool support for the Semantic Web Technologies is improving and these techniques are becoming more popular. As such, it is our belief that the results presented in this dissertation will only increase in value in the coming years.

Samenvatting

De laatste jaren is het gebruik van bewakingscamera's of videobewakingssystemen exponentieel toegenomen. Een dergelijk systeem beheert de data-input van verscheidene camera's en is verantwoordelijk voor het beheer, de visualisatie en de opslag van de opgenomen gegevens. Aangezien het aanal camera's binnen één systeem drastisch steeg, werd het duidelijk dat de menselijke operatoren niet meer in staat zijn om deze hoeveelheid informatie te verwerken. Vandaar ontstond de behoefte aan intelligente videobewakingssystemen en samen met het aantal camera's stegen ook de inspanningen op het gebied van automatische videoanalyse. Deze nieuwe tak binnen het onderzoekgebied van de computervisie is spoedig een groot onderzoekonderwerp geworden, dat door industriële en regeringsfondsen wordt gestuurd.

Tegenwoordig kent automatische video-analyse verschillende toepassingen zoals de detectie van indringers, het voorkomen van diefstal, controle van menigtes, detectie van ongelukken, opsporing van achtergelaten bagage, detectie van rondhangende personen, enz. Bovendien worden computervisie-algoritmes ook toegepast in andere domeinen zoals in het verkeer (snelheidscontrole, detectie van ongevallen, beheren van kruispunten, filecontrole, herkenning van nummerplaten, enz.) of in de kleinhandel (tellen van klanten, of registreren van activiteitspatronen).

Er werd reeds veel onderzoek verricht omtrent alle aspecten van intelligente videobewakingssystemen. Algoritmes werden ontwikkeld om bewegende objecten zoals mensen en voertuigen te ontdekken en te segmenteren. Opvolgalgoritmes volgen deze objecten over verschillende beelden of tussen verschillende camera's. De afgelegde trajecten en de lichaamsvorm worden gebruikt om het gedrag van de gedetecteerde objecten te bepalen en te voorspellen. Meer geavanceerde technieken zijn bijvoorbeeld nummerplaat- en gezichtsherkenning.

Videobewakingstoepassingen worden gekenmerkt door twee belangrijke aspecten; de **snelheid** en de **nauwkeurigheid** van detectie zijn van uiterst belang. Bijna alle voornoemde toepassingen vereisen een snelle en nauwkeurige detectie en segmentatie van de bewegende objecten. Hoewel deze detectie de eerste stap is, is het ook één van de moeilijkste, omwille van de afhankelijkheid van de opgenomen videosequentie. De belangrijkste problemen zijn ruis, schaduwen, lichtveranderingen en bewegingen in de omgeving. Bovendien maken occlusies, verschillen in grootte, snelheid en vorm van de bewegende objecten, objecten die halt houden of delen uit de omgeving die veranderen het moeilijk om een betrouwbaar detectiesysteem te maken.

Als objectdetectie-algoritmes worden toegepast in grote videobewakingsssystemen is er nood aan een manier om deze gedetecteerde objecten of gebeurtenissen te signaleren, mede te delen en op te slaan. Het gebruik van gestandaardiseerde (op XML-gebaseerde) metadatastandaarden werd voorgesteld om tot interoperabiliteit te komen binnen één systeem. Nochtans, wanneer verschillende bewakingssystemen, of modules van verschillende origine binnen één systeem, gecombineerd worden, introduceert het gebruik van verschillende XML-gebaseerde standaarden opnieuw interoperabiliteitsproblemen. De verschillende standaarden gebruiken namelijk verschillende constructies om dezelfde concepten te vertegenwoordigen.

Ons onderzoek is gesitueerd binnen de context van een intelligent videobewakingssysteem, waar wij ons enerzijds richten op het detecteren van bewegende objecten en anderzijds op het gebruik van verschillende XML-gebaseerde standaarden om deze objecten te beschrijven. Ten eerste, stellen we twee nieuwe achtergrondsubtractietechnieken voor om bewegende objecten te detecteren bij gebruik van een statische camera. De eerste techniek werkt op basis van ongecomprimeerde videosequenties en analyseert de pixels van elk beeld om bewegende objecten te detecteren. De tweede techniek analyseert sequenties die volgens een recente videostandaard (H.264/AVC) werden gecodeerd. Hierbij worden eigenschappen van de H.264/AVC-gecodeerde videostroom gebruikt om de bewegende objecten te detecteren. Ten slotte onderzoeken we de problemen die voorkomen wanneer de gedetecteerde objecten beschreven worden met gestandaardiseerde XML-gebaseerde metadataformaten. We lossen deze problemen echter op in de context van contentmanagementsystemen, aangezien deze als generalisatie van videobewakingssystemen kunnen worden gezien.

In eerste instantie wordt de detectie van bewegende objecten in het pixeldomein besproken. In dit geval bestaat de input van de algoritmes uit ongecomprimeerde videosequenties en het doel is die pixels in de beelden te vin-

den die overeenstemmen met bewegende objecten. De detectie van bewegende objecten in het pixeldomein is een uitgebreid bestudeerd probleem en talrijke methodes werden voorgesteld. Eén van de populairste groep methodes zijn achtergrondsubtractietechnieken. Tijdens het analyseren van de videodata wordt een achtergrondmodel gecreëerd en dynamisch bijgewerkt. Voor elk nieuw beeld worden de afwijkingen van de pixelwaarden ten opzichte van het achtergrondmodel bepaald en gebruikt om de observaties te classificeren als achtergrond (deel van de omgeving) of voorgrond (een bewegend object). We gaan uitvoerig in op het verwante werk in dit domein en sommen een aantal problemen op waaraan de huidige achtergrondsubtractiesystemen lijden.

In deze verhandeling stellen we een robuust detectie-algoritme voor bewegende objecten voor. Het voorgestelde systeem is een multimodale spatio-temporele achtergrondsubtractietechniek dat pixels vindt die overeenkomen met bewegende objecten. Er wordt gebruik gemaakt van een gewogen combinatie van achtergrondmodellen, gecombineerd met een snelle beeldsegmentatie. De modellen bestaan uit een gemiddelde waarde, een hogere en lagere drempelwaarde, een maximumverschil met de laatst waargenomen achtergrondwaarde, en een luminantiemaat die op de statistieken van de fotonruis in het beeld gebaseerd is. Het probleem van geleidelijke lichtveranderingen is bekend in de literatuur en een aantal oplossingen werden reeds voorgesteld, maar deze werken ten koste van hoge complexiteit. Door het verschil met de laatst waargenomen achtergrondwaarde te bekijken, wordt hier een zeer snelle en efficiënte techniek voorgesteld. Nieuwe pixelwaarden worden vergeleken met deze modellen om een overeenkomst te vinden. Nadien wordt een besluit (classificatie als voorgrond of achtergrond) genomen, gebaseerd op de gewichten van de modellen. Vervolgens worden de parameters van de modellen bijgewerkt aan de hand van de nieuwe pixelwaarden om de dynamiek van de omgeving op te volgen. Tot slot worden schaduwen behandeld om de detectieresultaten te verbeteren. De verwerkingssnelheid van de algoritmes is zeer belangrijk, vooral in de context van een videobewakingssysteem. Daarom stellen we een intelligent analysemasker voor, dat toestaat om het aantal pixels te verminderen die voor elk beeld geanalyseerd moeten worden. Ten slotte wordt een snelle beeldsegmentatie gentroduceerd die randopsporing toepast om het beeld op te delen in segmenten. De ruisstatistieken van de video worden geanalyseerd om foute randen te verwijderen en gaten op te vullen. Deze segmentatie wordt uiteindelijk gecombineerd met de achtergrondsubtractie.

Om het voorgestelde systeem te evalueren vergelijken we het met gerelateerd werk in dit domein. Verschillende videobewakingssequenties worden geanaly-

seerd en de resultaten (uitgedrukt in aantal van correct geclassificeerde pixels) worden voorgesteld. Om de lezer een beter idee te geven van de voordelen van ons systeem, worden een aantal visuele voorbeelden gegeven. Ten slotte worden de uitvoeringstijden vergeleken om de snelheidswinsten van onze methode aan te tonen.

Tegenwoordig omvatten de meeste videobewakingssystemen videocompressie voor de efficinte voorstelling, transmissie, en/of opslag van de opgenomen videosequenties. Indien de video gecodeerd werd, is een decodeerstap nodig, alvorens de video kan geanalyseerd worden met algoritmes werkend in het pixeldomein. Daarom worden reeds onderzoeksinspanningen geleverd naar algoritmes die rechtstreeks op de gecodeerde videosequenties kunnen toegepast worden. Deze baseren zich dus niet op de pixelwaardes, maar op de informatie die in de videosequentie overblijft nadat deze gecodeerd werd. In deze verhandeling concentreren wij ons op de H.264/AVC-specificatie voor digitale videocodering. Een specificatie die momenteel toonaangevend is qua videocompressie.

H.264/AVC is een eerder nieuwe videostandaard, maar de eerste camera's die dit videoformaat ondersteunen hebben reeds de videobewakingsmarkt gevonden. In de literatuur werden reeds een aantal methodes voorgesteld die bewegende objecten trachten te detecteren, rechtstreeks in het H.264/AVC-gecomprimeerd domein. Een gemeenschappelijk aspect van deze systemen is het gebruik van bewegingsvectoren die aanwezig zijn in de gecomprimeerde videosequenties. Nochtans worden deze bewegingsvectoren gecreerd vanuit het standpunt van videocodering, dus met het doel de compressie te verhogen. Vandaar vertegenwoordigen deze niet noodzakelijk de echte beweging aanwezig in de videosequentie. Dientengevolge besteden deze algoritmes een grote hoeveelheid werk aan het filteren van de bewegingsvectoren om relevante informatie over te houden voor de detectie van bewegende objecten.

Wij stellen hier een nieuw algoritme voor dat bewegende objecten detecteert, rechtstreeks in het H.264/AVC-gecomprimeerd domein, waarbij de bewegingsvectoren genegeerd worden. De voorgestelde methode baseert zich op de graad van compressie die een videocodec bereikt voor verschillende blokken van een beeld. Hierbij gaan we uit van de veronderstelling dat delen van het beeld die overeenkomen met achtergrond beter kunnen gecomprimeerd worden dan delen die overeenkomen met bewegende objecten. Om aan te tonen dat onze veronderstelling houdt, maken we een analyse van het bitgebruik van de verschillende macroblokken (blokken bestaande uit 16×16 pixels) in een gecodeerde videostroom. Deze observaties werden gebruikt om tot een detectie-algoritme te komen dat macroblokken vindt, waarvan verondersteld

wordt dat ze overeenkomen met bewegende objecten. Het algoritme creert een achtergrondmodel gebaseerd op de grootte (in bits) van de macroblokken. Voor elk nieuw beeld wordt de grootte van macroblokken vergeleken met dit model om ongebruikelijke grote blokken te ontdekken. Deze macroblokken komen vermoedelijk overeen met bewegende objecten en worden dan spatieel en temporeel gefilterd. Het gebruik van macroblokken kan te ruw zijn aangezien deze uit 16×16 pixels bestaan. Derhalve verfijnen wij het algoritme tot blokken van 4×4 pixels door de submacroblokken te analyseren.

Zoals met het algoritme in het pixeldomein wordt een uitgebreide vergelijking gemaakt met het verwante werk uit de literatuur. We tonen aan dat ons systeem betere resultaten geeft dan verwante technieken die gebaseerd zijn op bewegingsvectoren. Verschillende videobewakingssequenties worden gebruikt voor deze vergelijking.

De nodige informatie over de mate van compressie voor de blokken van een beeld kan heel vlug uit de gecodeerde video gehaald worden. Vandaar overtreft ons voorstel de verwante technieken in termen van verwerkingssnelheid. Onze techniek detecteert bewegende objecten tot 20 keer sneller dan het verwante werk in dit domein. Bovendien wordt er in deze verhandeling dieper ingegaan op de invloed die de configuratie van de gebruikte videocodec heeft op de prestaties van de algoritmes. Dit is een studie die in de literatuur ontbreekt bij het merendeel van gerelateerde technieken. Er wordt aangetoond dat de verschillende configuraties, resulterend in verschillende gecomprimeerde videostromen, een invloed hebben op de prestaties van de detectiealgoritmes. Maar vergeleken met de technieken gebaseerd op bewegingsvectoren is ons systeem minder afhankelijk van deze configuratieverschillen.

Een laatste aspect dat in deze verhandeling behandeld wordt, is hoe om te gaan met de interoperabiliteitsproblemen die gentroduceerd worden door het gebruik van verschillende XML-gebaseerde metadatastandaarden. We bespreken kort hoe dit probleem zich in de context van videobewakingssystemen voordoet. Daarna weiden we uit over hoe dezelfde problemen in het beheer van (persoonlijke) multimediale data voorkomen. Aangezien een videobewakingssysteem kan aanzien worden als een specifiek geval van contentmanagementsystemen, gaan we de problemen aanpakken in de bredere context van contentmanagement. Er wordt een nieuwe gelaagde architectuur voorgesteld die Semantische Webtechnologien gebruikt om verschillende metadataformaten in de lagere laag te combineren, terwijl de hogere laag uit specifieke ontologien bestaat die metadata vertegenwoordigen die relevant zijn voor persoonlijke contentmanagementsystemen. De metadataformaten en de relaties tussen hen worden voorgesteld door ontologien in OWL en we

hebben de eerste formele representatie, gebruikmakende van OWL, gemaakt van de DIG35-metadatastandaard. Deze ontologie wordt opgenomen in de architectuur en gecombineerd met andere metadatastandaarden zoals MPEG-7. Voor de evaluatie van de architectuur wordt een scenario opgesteld bestaande uit het importeren van multimediale data en metadata in het contentmanagementsysteem, het toevoegen van een annotatie volgens een gestandaardiseerd metadataformaat en het gebruiken van metadata om de multimediale data terug op te vragen. Vervolgens wordt er aangetoond hoe het voorgesteld systeem geschikter is om dit scenario te doorlopen dan het verwante werk in dit domein.

In deze verhandeling werden verschillende aspecten van een intelligent videobewakingssysteem behandeld. Een grote nadruk lag op de detectie van bewegende objecten, zowel in het pixel- als gecomprimeerde domein. Tot slot werd het gebruik van XML-gebaseerde metadata bestudeerd en hebben we methodes voorgesteld om de interoperabiliteit te verbeteren. We hopen dan ook de lezer overtuigd te hebben dat dit onderzoek een originele bijdrage heeft geleverd tot de ontwikkeling van een intelligent videobewakingssysteem.

Het gebruik van H.264/AVC-codering in de videobewakingsmarkt bevindt zich nog in een eerste fase. Bovendien zal het gebruik van metadata om gedetecteerde objecten of gebeurtenissen te beschrijven steeds belangrijker worden met de verhoging van het aantal camera's en videobewakingssystemen. Ten slotte stijgt de populariteit en het aantal ondersteunende softwarepakketten van de Semantische Webtechnologieën voortdurend. Als dusdanig is het ons geloof dat de resultaten die in deze verhandeling worden voorgesteld in de komende jaren nog in waarde zullen toenemen.

List of Abbreviations

AVC	Advanced Video Coding
AVSS	Advanced Video and Signal based Surveillance
BG	Background
BRICKS	Building Resources for Integrated Cultural Knowledge Services
CABAC	Context-based Adaptive Binary Arithmetic Coding
CANDELA	Content Analysis, Network DELivery, Architecture and supporting technologies
CAVIAR	Context Aware Vision using Image-based Active Recognition
CAVLC	Context-Adaptive Variable Length Coding
CCD	Charge Coupled Device
CIF	Common Intermediate Format
CMOS	Complementary Metal Oxide Semiconductor
CMS	Content Management System
COMM	Core Ontology for Multimedia
CVML	Computer Vision Markup Language
DANAE	Dynamic and distributed Adaptation of scalable multimedia coNtent in a context-Aware Environment
DCT	Discrete Cosine Transform
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DVD	Digital Versatile Disc
EPZS	Enhanced Predictive Zonal Search
eSMM	extended SMM
EXIF	Exchangeable Image File Format
FG	Foreground
FOAF	Friend of a Friend
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FP6	The Sixth Framework Programme
FRExt	Fidelity Range Extensions
HSV	Hue Saturation Value
I3A	not-for-profit International Imaging Industry Association
IPR	Intellectual Property Rights

IST	Information Society Technologies
JM	Joint Model reference software
JPEG	Joint Picture Experts Group
MB	Macroblock
MGM	Mixture of Gaussian Models
MMSem	W3C Multimedia Semantics Incubator Group
MPEG	Moving Picture Experts Group
MSO	Multimedia Structure Ontology
MV	Motion Vector
N3	Notation 3
OWL	Web Ontology Language
PCMS	Personal CMS
PECMAN	Personal Content Management Platform
PETS	Performance Evaluation of Tracking and Surveillance
PTZ	Pan-Tilt-Zoom
QCIF	Quarter CIF
QP	Quantization Parameter
RDD	Rights Data Dictionary
RDF	Resource Description Framework
RDFS	RDF Schema
RDQL	RDF Data Query Language
REL	MPEG-21 Rights Expression Language
ROC	Receiver Operator Characteristic
SHex	Simplified Hexagon Search
SMM	Simple Mixture of Models
SPARQL	SPARQL Protocol And RDF Query Language
SWRL	Semantic Web Rule Language
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UAV	Unmanned Aerial Vehicle
UMHex	Uneven Multi-Hexagon Search
VCEG	Video Coding Experts Group
VDO	Visual Descriptor Ontology
VEML	Video Event Markup Language
VERL	Video Event Representation Language
VS7	Visual Surveillance XML schema
XML	Extensible Markup Language
XPATH	XML Path Language
XSLT	eXtensible Stylesheet Language Transformations
W3C	World Wide Web Consortium

Contents

Chapter 1

Introduction

Big brother is watching you.
– George Orwell (1903 - 1950) – ”1984”.

In this dissertation, we present our work on the detection and representation of moving objects in video surveillance sequences. Two main research directions were taken, firstly, we focus on feature extraction, being the detection and segmentation of moving objects. This feature extraction step is handled in 2 domains, resulting in moving object detection techniques working in the pixel and in the compressed domain. The second direction is research on the representation of these extracted features by using standardized metadata. Here we extend our domain from video surveillance to content management systems and investigate how different metadata standards can be used together. In the next section, we describe the general context and explain some important concepts for the rest of the dissertation.

1.1 Context

The use of video cameras to surveille a certain environment has a long-lasting tradition. The first press reports, indicating the usage of video surveillance in public areas, date from 1965. Initially, the video surveillance systems were mostly used as a deterrent to prevent crime, however the systems were quickly adopted to monitor a wide range of actions (intrusion, loitering, left baggage, crowding, slip or fall detection, etc.). Moreover, visual surveillance has been successfully applied in different domains (insurance industry, traffic control, etc.).

With the growth of the video surveillance market came an increase in research efforts in the domain of automated video analysis, which became a large part of the computer vision research branch. The goal is to aid and improve the efficiency of human operators in the surveillance of large scenes by applying computer vision algorithms on the captured video data. These algorithms detect, track, and identify objects, recognize faces, detect abnormal behaviour, predict movement of objects, etc.

Finally, the internet made video surveillance systems work on a much larger scale than before, introducing distributed, multi-camera video surveillance systems. Recent efforts continue, the new trends are mobile video surveillance (Apple announced its interest in using the iPhone to deliver quality mobile video surveillance) and megapixel surveillance cameras [1].

Figure 1.1 shows the architecture of an intelligent video surveillance system.

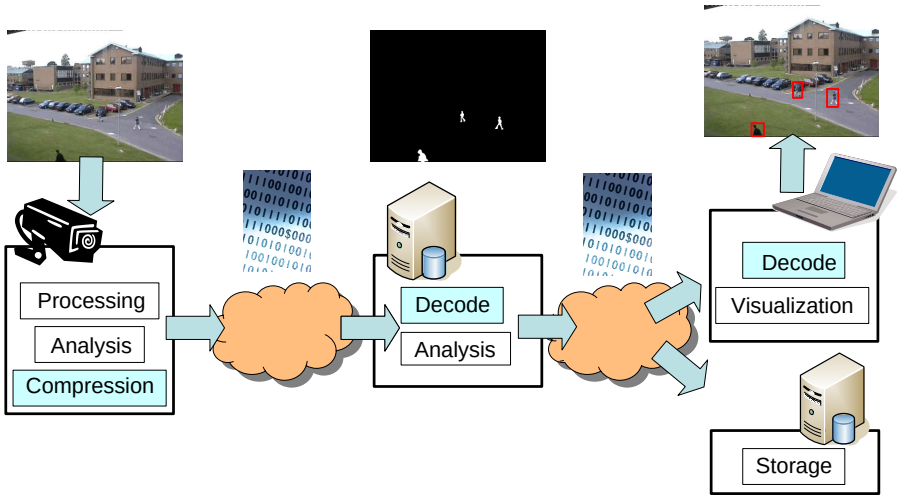


Figure 1.1: General Video Surveillance System.

A camera is used to record video sequences of a certain scene. In most cases a first form of processing is applied here, including contrast enhancement, noise reduction, etc. In case of a smart camera the processing also includes video analytics like motion detection. This processing can be used to reduce the amount of data that is sent from the camera. Such cameras only produce video streams if a certain amount of motion is detected in the scene (although this is called motion detection, the source of the change can also encompass noise or changing lighting conditions).

Since the beginnings of digital video surveillance, compression is used to reduce the bandwidth and storage costs. An encoder is used to remove the redundancy in and between the video frames and a compressed video stream is created. Finally, a decoder is needed to reconstruct the original video stream when visualization is needed.

An analysis module decodes the video stream and performs the actual analysis (dependent on the use case of the system). The results of the analysis are manifested in some kind of format. This could be a mark-up on the original video, a representation of the detected events by means of a metadata standard, a binary signal on an alarm channel, etc. A visualization module combines these outputs, decodes the video, and makes a comprehensive overview for the end-user or operator. Lastly, the video feeds and possible metadata are stored by a storage module.

In this dissertation we focus on the analysis by creating moving object detection algorithms in the pixel and compressed domain. We make an abstraction of where the actual analysis happens and focus on the algorithms that perform it. Additionally, we have investigated the use of different metadata standards for the representation of the analysis results.

1.2 Moving Object Detection

Video surveillance is proliferating worldwide, and although the efforts done to create smart autonomic video surveillance systems are increasing, providing **fast** and **accurate** solutions remains difficult [2].

Typical surveillance systems start with the detection and extraction of moving objects in image sequences. Generally, the objects that one wants to detect in an image are called foreground objects. In this dissertation we focus on moving objects, as such, moving vehicles need to be detected, but once a vehicle stops for a long period (e.g., a parked car) it should not be regarded as foreground anymore. The part of the image that does not correspond with those foreground objects is called background. Note that the background itself can be highly dynamic and change over time. Next, these foreground objects are tracked over time and are classified in known categories. Subsequently, intelligent decisions about the behaviour of these objects can be taken and alerts are issued if necessary. Since automation is the goal, it is desirable to achieve very high sensitivity with the lowest possible false alarm rates. Figure 1.2 shows a schema listing different components used within an analysis module [3].

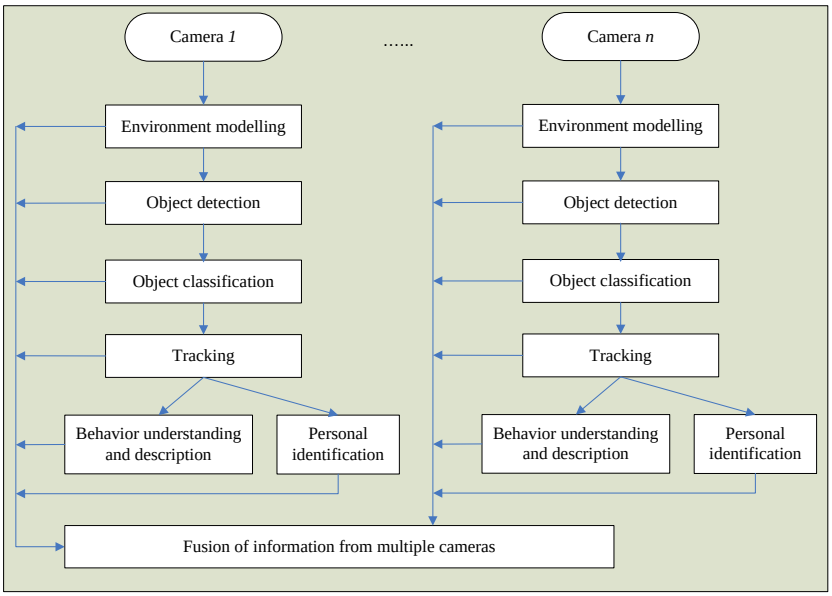


Figure 1.2: Structure of a general analysis module in video surveillance systems.

An analysis module consists of different stages covering low-level, intermediate-level, and high-level vision. A visual surveillance application may not contain all the stages in the framework and other stages may be added depending on the application. The stages of the framework are described in the following.

Environment modelling The process of detecting motion or segmenting moving objects involves environment modelling, motion segmentation and object classification. To be able to separate moving objects from the scene, a model for the environment or background is needed. The model makes it possible to determine, if a pixel or region in the frame corresponds to a background or a foreground object.

Object detection This stage detects and segments pixels or regions corresponding to foreground objects using the environment or background model. These pixels or regions are the focus for the later stages in the framework. Most segmentation methods make use of either spatial or temporal information in the video sequence. Environment modelling and object detection are regarded as low-level vision algorithms.

Object classification The detected regions from the motion segmentation stage may correspond to different targets in the scene. In traffic surveillance applications these targets may be humans and vehicles. Hence, it is essential to classify the detected regions. This is often in the literature considered as a standard pattern recognition task.

Tracking Having segmented the foreground objects, visual surveillance systems track these objects from frame to frame. Tracking foreground objects over time typically involves matching objects in consecutive frames using descriptions of e.g., points, lines, or blobs. The main problem related to the tracking stage is how to handle occlusion. By tracking foreground objects the system is able to supply information on the history of the objects in the video sequence.

Behaviour understanding and description Following tracking is the problem of understanding object behaviour. In this stage, high-level descriptions of actions and interactions are formed. Shapes of the segmented objects are analysed to interprete the actual behaviour of the objects. The object history from the tracking stage can be utilized in the interpretation of the object motion patterns. Understanding of behaviour can simply be seen as classification of time varying data. Typically, the behaviour is classified as normal or abnormal.

Personal identification This stage is at the same level as behaviour understanding and description. In this stage, a known identity is assigned to the tracked object. In case of surveillance of humans, biometric descriptions of the human face or gait can be used. A different approach is needed when assigning identities to vehicles. Given that the vehicles have a license plate and it is possible to extract an image of the license plates, it is rather straightforward to assign an identity to vehicles.

Fusion of information from multiple cameras Segmenting foreground objects, object classification, tracking, behaviour understanding, and personal identification can be accomplished using a single camera. However, using multiple cameras can overcome problems regarding occlusion and depth estimation of the objects, and is called high-level vision. In this stage, the information from the previous stages is collected. Additionally, fusion of low-level information may result in higher reliability of the overall system.

Within this dissertation we focus on video surveillance systems using only one camera. The use of multiple cameras or sensors increases the possibilities (and complexity) of a video surveillance system. For example, problems of occlusions can more easily be dealt with when using the feeds of different cameras. Nevertheless, even in multi-camera systems, processing is needed that focuses on streams of a single surveillance camera. Moreover, we make the assumption of a static camera. This excludes moving cameras, like those mounted on cars or unmanned aerial vehicles (UAVs), but also discards pan-tilt-zoom (PTZ) cameras. Much work has been done to compensate for camera-motion (disregarding the source of motion), so even when using a static camera these algorithms can be applied to simulate a static camera [4, 5].

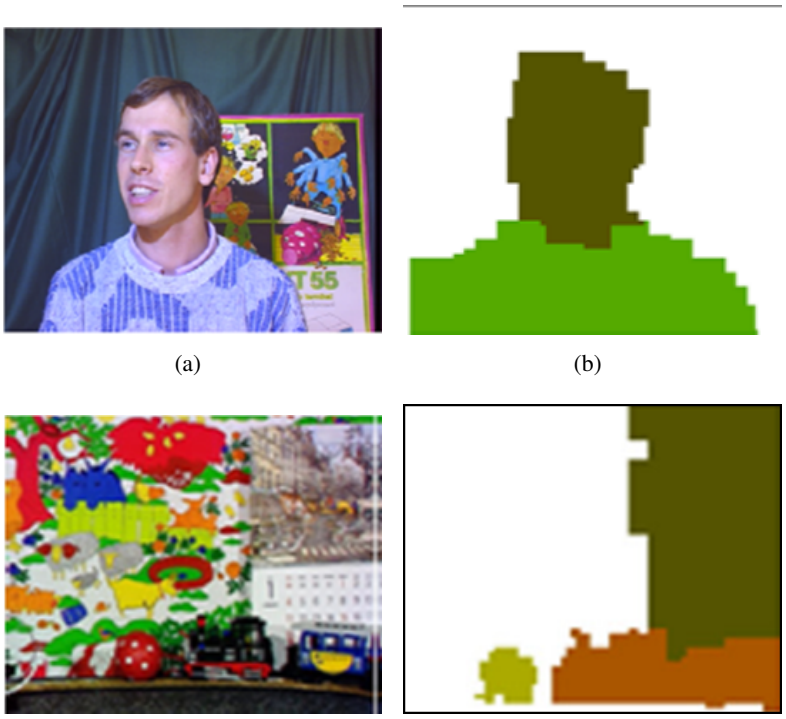
The recent rapid increase in the amount of surveillance cameras has led to a strong demand for automatic methods for processing their outputs. Nowadays, researchers are focusing on activity and behaviour analysis, to be able to make automated intelligent decisions. Detection of loitering, luggage abandoned by the owner, trespassing, and theft of items are typical examples of high level actions, which the computer vision community wants to tackle. All these actions require an initial detection of moving objects before any further analysis can be done. So the detection and segmentation of objects of interest in image sequences is the first processing step in visual surveillance applications.

Detecting and segmenting moving objects are closely related. If one detects a moving object, it already has been segmented to some point, likewise, if an object is segmented it is likely to be detected. It is important to clearly state the difference between object detection and object segmentation techniques.

Object segmentation by itself is not so much on the detection of the object but on providing a detailed segment representing the object. Fig. 1.3 shows two typical examples of outputs of a segmentation approach. Such approaches are typically used for object-based video coding, object recognition, or content based image retrieval.

Within the domain of video surveillance the detection of moving objects is about creating a foreground-background segmentation. The manner in which these objects are detected and segmented can differ. The centroid, bounding box, outer edge, block-based segmentation and pixel-wise segmentation are all examples of outputs of detection algorithms. It should be clear that some outputs put less restrictions on the algorithms than others.

In this work, when talking about object detection, the goal is to get a pixel-wise segmentation of the detected objects. Hence, an optimal system gives only those pixels as output that correspond with foreground objects.



(a) (b)

(c) (d)

Figure 1.3: Examples of object segmentation where regions with similar movement characteristics are segmented together.

This is an important aspect when considering the manner of evaluation of algorithms that detect moving objects. For the evaluation, a ground truth annotation is used. For the creation of such a ground truth, all the pixels of an image are manually labelled as belonging to the foreground or background. To reduce the amount of work that comes with this, for each video sequence, only a representative set of images is annotated. The output of the algorithms is then pixel-wisely compared with the ground truth. Next, different statistical performance measures can be used to evaluate how accurate the resulting detection resembles the ground truth annotation. In this dissertation, when discussing the algorithms that we have created, we will elaborate more on the ground truth and performance measures that are used for evaluation purposes.

Note that our goal is to create a binary classifier; a pixel is either part of some moving object, or part of the background. This has two main implications:

- If multiple moving objects are present in the scene, we can make no distinction between them.
- No probabilities are assigned to the detection.

The actual distinction between the detected objects is the responsibility of further modules in the analysis chain (as shown in Fig.1.2). An optimal object detection can only benefit a successful classification.

The use of a probabilistic output (denoting which areas of the detection are more probable to correspond to the moving object than others) could increase the usability of the object detection system. Such an output could allow for better tracking or identification. However, the binary detection is an approach that is common in related work in this field. Finally, the typical evaluation means count the numbers of pixels that are correctly or wrongly classified, so they rely on a binary output.

A final note that needs to be made, is the definition of a moving object. Conceptually, moving objects in the scene can be people, cars, bicycles, animals, etc. It is the responsibility of higher level modules to make a classification of the detected object. Events that should not be detected are fixed objects that have motion (e.g., waving trees and escalators). The detection of moving objects in dynamic scenes has been the subject of research for several years and different approaches exist [6].

Frame differencing One of the simplest forms of object detection where consecutive frames are subtracted from each other to find pixels that represent foreground objects. This technique typically yields only parts of the foreground objects.

Optical flow This technique uses the characteristics of motion vectors of moving objects to detect foreground objects. This flow is detected by back-tracking features of the current frame to the previous frames, as such, creating motion vectors. The size and direction of these vectors are consequently used to detect moving objects. However, most optical flow methods are considered to be computationally expensive and sensitive to noise [7].

Template matching This group of techniques relies on templates of certain predefined objects (e.g., cardboard and stick models) [8]. The images are then searched for good matches with these templates to facilitate object detection. However, the entire image has to be searched to find a match

with a template and the many possible poses, scales, and viewpoints make this method very complex.

Background subtraction This is one of the most used techniques for detecting moving objects. During the surveillance of a scene, a background model is created and dynamically updated. For each new image, deviations of the pixel values from the background model are detected and used to classify the observations as belonging to background or foreground. Note that frame differencing can be seen as a simple form of background subtraction where the background model consists of the previous frame.

Many different models have been proposed for background subtraction and the need to cope with multi-modal background environments has soon been established. This multi-modality manifests itself in regions where a single pixel value can follow multiple distributions according to the given situation of the environment (e.g., a moving tree). Hence, the literature contains background subtraction systems that use multiple models for each pixel.

Newly captured images need to be analysed entirely. This means that every new pixel has to be checked for consistency with the specific background model. Afterwards, a decision is taken for each pixel whether it represents background or foreground. Finally, adaptive background subtraction systems adjust the parameters of their models to better match the existing environment. Simple background subtraction algorithms can be very fast, even when analysing every pixel of an image. However, when using adaptive systems with multiple background models, the processing of each single pixel becomes hazardous. This is indeed a problem, since, initially, background subtraction systems gained popularity because they were much faster than other techniques like optical flow and template matching. Additionally, there are a number of important problems when using background subtraction algorithms (quick illumination changes, initialization with moving objects, ghosts, and shadows, etc.), as was reported in [9].

In this work we present a multi-modal spatio-temporal background subtraction technique, containing solutions to a number of problems that current background subtraction schemes suffer from. To overcome the complexity of the traditional methods we present a simpler mixture of models technique (SMM). Simple models are used and the system is tailored to achieve high speed and better accuracy, dealing with quick illumination changes, noise, and shadows. SMM is a temporal technique, meaning to make a decision about a specific pixel it uses information on the values of that pixel in the past. Spatial infor-

mation is used to extend SMM. We present a fast edge-based image segmentation that is combined with SMM to improve the detection results. This results in a robust moving object detection technique called extended SMM (eSMM). An extensive evaluation is performed to show the applicability in indoor and outdoor environments and results are presented on different challenging sequences.

Note that the presented system works in the pixel domain and is only applicable if we have the actual pixel values for each captured image. A practical video surveillance scenario includes video compression to reduce the used bandwidth and storage. Consequently, if one wants to analyse the captured images to find moving objects, a decoding step is needed before the algorithms described above can be executed. In this context, we ignore the fact that some processing could be done on the camera itself, so before the video is encoded.

To avoid the decoding step and to reuse the work done during the encoding, the literature holds several efforts that detect moving objects directly upon the compressed video stream. In this case, the goal is, like for the algorithms above, to detect moving objects fast and accurately. For this purpose, the compressed video stream is analysed and the specific coding constructs that are available in the stream are used to detect moving objects. Since the compressed video is a more compact representation of the original video stream, analytics working in the compressed domain can be faster than the pixel-domain approaches. Moreover, it is not necessary to fully decode the video stream before the analysis can be done, resulting in additional gains in time.

In this dissertation, in addition to our moving object detection technique that works in the pixel domain, we present a moving object detection technique in the compressed domain. We use H.264/AVC for the compression of the video sequences, which is the latest video coding standard of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [10]. This new video coding standard is likely to overtake the video surveillance market, allowing to further reduce bandwidth and storage costs compared to existing approaches.

When trying to detect moving objects by solely looking at an H.264/AVC compressed video stream, several features can be used. Related work in this domain mostly relies on the motion vectors created during compression. However, we will show that the usage of motion vectors introduces issues, which need additional complexity to be solved.

In this dissertation we present a novel alternative approach that works on a higher level. Based on our observations, we discovered the specific compres-

sion ratio of H.264/AVC compressed bitstreams to be more useful for object detection. The assumption is that the video encoder can compress parts of the background better than parts of an image containing foreground objects. This assumption forms the base of a moving object detection technique working in the H.264/AVC compressed domain.

1.3 Metadata

The output of our object detection systems (in pixel and compressed domain) are pixel-wise segmentations of the image in foreground and background regions. These could form the input for high-level analysis modules to make intelligent decisions on objects, classes, trajectories, and behaviours. When using a distributed video surveillance system, an interchange format is needed to describe and share such detection results. Such information, used to describe data, is generally called metadata and it has applications in a broad range of domains within computer science. In the context of video surveillance, different metadata standards have been proposed using the Extensible Markup Language (XML) as underlying language. XML allows to structure data according to an XML schema. The latter defines terms and constructs to represent the metadata and states the structure of the metadata. However, the usage of XML to describe detected objects or events can infer some problems.

In this dissertation, we will show that a number of different approaches exist in formatting the metadata associated with a video surveillance system. However, there is not one general metadata standard that is generally accepted, and most likely such a standard will not be introduced in the near future. Consequently, combining different metadata schemes with each other seems to be the only solution to create interoperability between different modules and systems. However, the usage of different XML-based metadata standards for the same purpose can introduce interoperability issues. The standards generally use different XML constructs to denote the same concept. As such, it can be hard to find similarities between annotations using these different standards.

Moreover, XML does not allow to explicitly define the semantics of the concepts that are described. Traditional metadata standards present an XML schema to define the structure and fields that can be used, and supply a textual description of the meaning of the different concepts. As such, the metadata is machine-readable but the semantics of the metadata fields is not.

These are general problems when using different XML-based metadata standards. In this dissertation, we have tackled this issue in the context of an-

notation of personal pictures. Nowadays, anybody can produce multimedia content and the internet allows to massively share personal content with others. Metadata is used to annotate these images for retrieval and discovery. Different XML-based metadata standards exist exactly for this purpose, but as mentioned above, these can introduce interoperability issues and lack the capabilities to include semantic meaning. We introduce the usage of Semantic Web technologies as a way to integrate these XML-based metadata standards. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [11]. For this purpose, different technologies are being developped and within this dissertation we deploy the Web Ontology Language (OWL) [12]. This language permits the definition of custom defined concepts and relationships on which can be reasoned. Finally, we present a layered architecture for semantic personal content management systems that uses OWL on all layers.

1.4 Outline

This dissertation is organized as follows. In the next chapter, we present our work done in the pixel domain to detect moving objects. We present a robust multi-modal spatio-temporal background subtraction system. First, we elaborate on general problems that object detection techniques are faced with. Second, related work within this domain is presented. Next, we give an in-depth overview of our system and analyse it by including a comparison with related work on speed and accuracy of detection. The latter is shown both on objective (by evaluating the correctness of the detection for each pixel of an image) and subjective (visual examples) results.

In Chapter 3, we present a novel high-speed moving object detection technique that works in the H.264/AVC compressed domain. Again, we elaborate on related work, presenting systems that detect moving objects in MPEG compressed domain, with a strong focus on H.264/AVC specific techniques. The H.264/AVC specification is briefly touched and important concepts for this dissertation are explained. Subsequently, we present our observations of the link between compression ratio and foreground objects that triggered our research in this domain. Next, the system, based on those observations, is outlined. As in Chapter 2, we compare the proposed algorithm with related work and discuss the performance (in terms of speed, objective and subjective results). Lastly, we study the influence of different encoding configurations on our and related work.

Chapter 4 focuses on the interoperability problems of XML-based metadata standards and we show how this is applicable to video surveillance systems. Our work to solve these problems was done in the domain of personal content management systems. So, after the initial elaboration on these problems in the context of video surveillance, it is shown how the same problems arise in other domains, like the management of personal (annotated) content. We present related work in this domain and create an architecture for semantic personal content management systems. An extensive elaboration is given on how Semantic Web technologies can be used to deal with the given problems and we present the benefits of our system by means of a general use case scenario. The system is compared with the related work, giving a detailed analysis of where our system outperforms the related work in solving this use case.

Finally, the conclusions and future work of this dissertation are presented in Chapter 5.

1.5 Overview Publications

The research activities that have lead to this dissertation resulted in a number of publications listed in the Science Citation Index: one paper is published in SPIE's *Optical Engineering*, two other papers are accepted for publication in Elsevier's *Journal of Visual Communication & Image Representation*. Our work also contributed to a paper appearing in *Multimedia Systems*. Next to this, the work described in this dissertation contributed to 11 papers as first author and 6 as co-author, which were presented at international conferences. Lastly, several contributions were submitted to the MPEG community.

1.5.1 SCI-listed Publications

1. Chris Poppe, Sarah De Bruyne, Tom Paridaens, Peter Lambert, and Rik Van de Walle. Moving Object Detection in the H.264/AVC Compressed Domain for Video Surveillance Applications. *Visual Communication and Image Representation*

2. Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. Personal Content Management System a Semantic Approach. *Visual Communication and Image Representation*, 47:131 – 144, February 2009

3. Chris Poppe, Gaëtan Martens, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Robust spatio-temporal multimodal background subtraction for video surveillance. *Optical Engineering*, 47:110101, October 2008

4. Davy De Schrijver, Chris Poppe, Sam Lerouge, Wesley De Neve, and Rik Van de Walle. MPEG-21 Bitstream Syntax Descriptions for Scalable Video Codecs. *Multimedia Systems*, 11:403–421, June 2006

1.5.2 Other Publications

1. Sarah De Bruyne, Chris Poppe, Steven Verstockt, Peter Lambert, and Rik Van de Walle. Estimating Motion Reliability to Improve Moving Object Detection in the H.264/AVC Domain. In *To be published in Proceedings of The International Conference on Multimedia and Expo (ICME) 2009*, June 2009

2. Chris Poppe, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Effect of H.264/AVC Compression on Object Detection for Video Surveillance. In *Proceedings of The International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2009*, May 2009

3. Davy Van Deursen, Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. XML to RDF Conversion: a Generic Approach. In *Proceedings of 4th International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS) 2008*, pages 138 – 144, 2008

4. Gaëtan Martens, Chris Poppe, Peter Lambert, and Rik Van de Walle. Unsupervised Texture Segmentation and Labeling using Biologically Inspired Features. In *Proceedings of MMSP2008*, pages 159–164, 2008

5. Chris Poppe, Gaëtan Martens, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Dealing with Gradual Lighting Changes in Video Surveillance for Indoor Environments. In *Proceedings of 15th World Congress on Intelligent Transport Systems*, November 2008

6. Chris Poppe, Sarah De Bruyne, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Intelligent Preprocessing for Fast Moving Object Detection. In *Proceedings of SPIE Security and Defense*, volume 6978, March 2008

7. Chris Poppe, Frederik De Keukelaere, Saar De Zutter, Sarah De Bruyne, Wesley De Neve, and Rik Van de Walle. Predictable Processing of Multimedia Content, Using MPEG-21 Digital Item Processing. In *Proceedings of the 8th Pacific Rim Conference on Multimedia*, volume 4810, pages 549 – 558, December 2007

8. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Improved Background Mixture Models for Video Surveillance Applications. *Lecture Notes in Computer Science, 8th Asian Conference on Computer Vision(ACCV 2007)*, 4843:251–260, 2007

9. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Dealing with Quick Illumination Changes when using Background Mixture Models. In *Proceedings 8th Asian Conference on Computer Vision*, volume 4843, Tokyo, November 2007

10. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Mixture Models Based Background Subtraction for Video Surveillance Applications. In *Lecture Notes in Computer Science: Proceedings 12th International Conference on Computer Analysis of Images and Patterns*, volume 4673, pages 28–35, August 2007

11. Chris Poppe, Saar De Zutter, Wesley De Neve, and Rik Van de Walle. Reconfigurable Multimedia: Putting the User in the Middle. In *Proceedings of COST298 conference: the Good, the Bad an the Unexpected*, pages 15–30, May 2007

12. Gaëtan Martens, Chris Poppe, and Rik Van de Walle. Enhanced Grating Cell Features for Unsupervised Texture Segmentation. In *Proceedings of Performance Evaluation for Computer Vision: 31ste AAPR/OAGM Workshop*, pages 9–16, Wien, May 2007

13. M. Ransburg, H. Hellwagner, R. Cazoulat, B. Pellan, C. Concolato, S. De Zutter, C. Poppe, R. Van de Walle, and A Hutter. Dynamic and Distributed Adaptation of Scalable Multimedia Content in a Context-Aware Environment. In *Proceedings of the WiCon'06 conference*, 2006

14. Chris Poppe, Frederik De Keukelaere, Saar De Zutter, and Rik Van de Walle. Advanced Multimedia Systems using MPEG-21 Digital Item Processing. In *Proceedings of the Eight IEEE International Symposium on Multimedia*, pages 785–786, San Diego, December 2006

15. Chris Poppe, Ingo Wolf, Sven Wischnowsky, Saar De Zutter, and Rik Van de Walle. Licensing System in an online MPEG-21 Environment. In

Proceedings of the IADIS international conference WWW/internet 2006, volume II, pages 315–319, Murcia, October 2006

16. Chris Poppe, Michael Ransburg, Saar De Zutter, and Rik Van de Walle. Interoperable Affective Context Collection using MPEG-21. In *Proceedings of the International Conference on Wireless Mobile and Multimedia Networks Proceedings*, volume II, pages 1603–1606, China, November 2006

17. Saar De Zutter, Frederik De Keukelaere, Chris Poppe, and Rik Van de Walle. Performance Analysis of MPEG-21 Technologies on Mobile Devices. In *Proceedings of SPIE-IST Electronic Imaging, Science and Technology*, volume 6074, page 12, San José, January 2006

1.5.3 MPEG Contributions

1. Chris Poppe, Saar De Zutter, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13965, Contribution to Utility Software for ISO/IEC 21000-10 DIP/AMD 1, MPEG-document, October 2006

2. Saar De Zutter, Chris Poppe, Davy De Schrijver, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13979, Update to Reference Software for Conformance to ISO/IEC 21000-10 DIP/AMD 1, MPEG-document, October 2006

3. Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13593, Contribution to Conformance Reference Software for ISO/IEC 21000-10 DIP AMD/1, MPEG-document, July 2006

4. Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13592, Contribution to Conformance for ISO/IEC 21000-10 DIP AMD/1, MPEG-document, July 2006

5. Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m12218, Contribution to WD of Amendment on C++ Binding, MPEG-document, July 2005

Chapter 2

Pixel Domain Object Detection

It's for your protection.
– Beverly Griffin (2002).

2.1 Introduction

Within the computer vision community several different approaches have been presented for the detection of moving objects in dynamic scenes [6]. One of the common techniques is background subtraction. During the surveillance of an environment, a background model is created and dynamically updated. Moving objects of interest, called foreground objects, are composed of those pixels that differ significantly from this background model. These background subtraction techniques quickly evolved from single static models to multi-modal, dynamically updated models.

First reports on background subtraction techniques used a static background model, created during a training phase. However, the environment that is monitored can be highly dynamic, caused by changing weather conditions, camera noise, changes in the background (like closed or opened doors, parked cars) and so on. Hence, creating a static background model for long-term use is not an option. Consequently, different ways were proposed to create a dynamic background model. Examples of this are the use of a running average or median for each pixel over the last n frames, as proposed in [39] and [40], respectively. Other techniques model each pixel with a Kalman filter [41] or use

a Gaussian distribution per pixel to represent the background [42]. When applying these techniques in a practical scenario, another issue arises. Motion in the background (such as waving trees) introduces a multi-modal behaviour of the actual background, so a background model consisting of multiple models is required for the same pixel position.

The Mixture of Gaussian Models (MGM) is one of the most popular (multi-modal) background subtraction techniques [43]. By using a mixture of Gaussian models and a dynamic update scheme, it can handle highly complex, multi-modal scenes with difficult situations like moving trees and bushes, clutter, noise, and permanent changes of the background. This technique will be discussed in-depth in Section 2.3.

Although it gives good results, the use of the Gaussian models and the update scheme are complex, which increases the processing times. Moreover, a number of problems remain hard to deal with. Toyama et al. discussed in detail several known problems when using background subtraction algorithms [44]. This list was also adopted by Javed et al. who selected a number of important problems that have not been addressed by most background subtraction algorithms [9]. We group them in the following items:

Illumination changes Gradual illumination changes, e.g., caused by moving clouds, can alter the appearance of the environment, resulting in corrupted background models. Additionally, sudden illumination changes, like a light that is switched on or off, have a large impact on most background subtraction systems. To decide whether a large change of pixel value is caused by such an event, global image processing is needed. Moreover, these illumination changes can typically occur in different parts of the environment, which even further complicates the detection of them.

Shadows Although related to the first item, this is a particular hard problem since shadows cast by moving objects are moving themselves. Advanced processing is generally needed to make the difference between moving objects and the shadows that they cast. Shadows cast by parts of the background can also cause problems since they are directly dependent on the light sources in the environment and can change with them.

Dynamic environments Typical examples of these are waving trees, flags, or other events, that should not be regarded as a moving object, but introduce changes in the background.

Ghosts This is the general name for objects that are part of the background and have moved to another location. In this case the part of the image where the object originally was located is seen for the first time and consequently regarded as foreground. This creates a ghost in the detection results since no real object is present at the detected spot. A typical example is a parked car that leaves.

Moved or stopped object This occurs when a moving object stops and becomes motionless for a while. The assumption is that a background subtraction system should be able to adapt itself to avoid continuous detections. This also applies to parts of the background that have changed (e.g., a closed door or window). Note that when a background object is relocated, we face both the problem of ghosts and moved objects.

Initialization with moving objects The problem occurs when it is hard to find images that have no or only few moving objects. This troubles the initialization of the background models and causes the performance of the algorithm to be low in the beginning. This problem can be seen as a mixture of the above problems and generally the same solutions apply.

To make the traditional MGM more robust to some of these known issues and to overcome the complexity, we present a simple mixture of models technique (SMM). Our models consist of an average, an upper and lower threshold, a maximum difference with the last background value, and an illumination allowance based on photon noise. Background subtraction techniques as discussed here are temporal techniques. They analyse the changes of pixel values over time to make a decision about background and foreground. However, in many cases only performing temporal background subtraction is insufficient, so spatial information is used to extend SMM. We present a fast edge-based image segmentation that is combined with SMM to improve the detection results, resulting in a technique called extended SMM (eSMM).

In this chapter, we first present related work that uses background subtraction to detect moving objects. Next, we elaborate on the mixture of Gaussian models technique and discuss the simpler models we propose. Subsequently, the spatial image segmentation is presented, which is combined with the temporal background subtraction. In Section 2.6 experimental results are shown and we end with the conclusions and an elaboration on our original contributions.

2.2 Related Work

Many efforts have been made in the field of object detection using background subtraction.

Haritaoglu et al. presented the W4 system in which they used a minimum value, maximum value, and maximum difference with the previous pixel to evaluate new pixel values [45]. However, these parameters are defined during a training period and are only periodically updated. Moreover, they do not use different models to cope with the multi-modal behaviour of complex environments.

Wu et al. gave a concise overview of background subtraction algorithms and compare MGM with their proposed technique [46]. They use a global Gaussian mixture model, built upon a difference image between the current image and an estimated background. Although their system is better for localization and contour preserving, it is more sensitive to complex environmental movements (such as waving trees).

Lee et al. improved MGM by introducing means to initialize the background models when moving objects are present in the environment [47]. They presented an online expectation maximization learning algorithm for training adaptive Gaussian mixtures. Their system allows to initialize the mixture models much faster than the original approach. Related to this topic, Zhang et al. presented an online background reconstruction method to cope with the initialization problem [48]. Additionally, they presented a change history map to control the foreground mergence time and made it independent of the learning rate. As such, they deal with the initialization problem and the problem of ghosts (background objects that are moved) but cannot deal with quick illumination changes.

Zivkovic et al. improved MGM by automatically selecting the number of components per pixel [49]. In some sequences this might be favourable, however in dynamic and complex environments the performance diminishes to that of the original technique.

Shan et al. presented an improved algorithm for motion detection based on MGM [50]. They use the HSV color space for shadow removal and use color histograms to deal with global illumination changes. Yang et al. adopted MGM and deal with fast changes by separating the update speed of the weights and the Gaussian parameters [51]. However, their technique is restrictedly suitable for slowly moving objects.

In [9], Javed et al. presented a number of important problems when using background subtraction algorithms, as discussed in the previous section. Accordingly, they proposed a system using pixel, region, and frame level processing, but their technique is based on a complex gradients-based algorithm. Unfortunately, the paper does not provide any information about the additional processing times needed for this technique.

Tian et al. [52] used a similar approach as the one used in [9] to deal with illumination changes. They presented a texture similarity measure based on gradient vectors, obtained by the Sobel operator. A fixed window is used for the retrieval of the gradient vectors, which largely determines the performance of their system (both in processing time and accuracy) .

Numerous techniques have been proposed to deal with shadows. An interesting overview on the detection of moving shadows is given in [53]. Prati et al. divide the shadow detection techniques in four classes, of which the deterministic non-model based approach shows the best results for the entire evaluation set used in the overview. Since the two critical requirements of a video surveillance system are accuracy and speed, not every shadow removal technique is appropriate. Multi-modal background subtraction techniques give good detection results, but the maintenance of several models for each pixel is computational expensive. Therefore, every additional processing task should be minimal. Furthermore, MGM was created to cope with highly dynamic environments, with the only assumption being the static camera. According to these constraints and following the results presented by Prati et al., we have chosen the technique described by Cucchiara et al. for a comparison with our system [54]. Results hereof are presented in Section 2.6.

As the related work suggests, using pixel-wise background subtraction is not enough in many cases, so spatial information is used to improve the results. Yokoyama et al. presented an object detection and tracking system based on edge detection [55]. They segment the image based on the Canny edge detection and remove edges with little motion (compared to the previous frame). Subsequently, they apply a line restoration step based on connected component analysis to recover edges that are wrongly deleted. Again, they use the background edges found in the previous frame to improve the results. Although they succeed in detecting moving objects this way, detection in multi-modal environments or in scenes with slowly moving objects is hard.

As can be learned from the related work, the use of multiple models is essential to deal with multi-modal environments. Moreover, difficult situations, like shadows and illumination changes, are hard to tackle without high additional

complexity. Finally, the related work shows that including spatial information can improve the detection. However, the discussed techniques either rely on complex methods to deal with these problems, or they do not present a general robust system. Based on these observations, we present a robust spatio-temporal multi-modal background subtraction system. In the next section we elaborate on MGM since it forms the base of our system.

2.3 Background Subtraction using MGM

When using background subtraction, a background model is created, which resembles the observed environment as closely as possible. A dynamic and complex environment typically introduces different background values for the same pixel (e.g., movement of branches), so using a single model to represent the background is insufficient. Therefore, MGM introduces a mixture of models and it was first proposed by Stauffer and Grimson in [43]. MGM is a time-adaptive per pixel subtraction technique in which every pixel in frame t is represented by a vector, called X_t, consisting of three color components (red, green, and blue). For every pixel a mixture of multivariate normal distributions, which are the actual models, is maintained and each of these models is assigned a weight. The Gaussian distribution G of the ith model in the mixture is given by:

$$G\left(X_t, \mu_{i,t}, \Sigma_{i,t}\right) = \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_{i,t}\right|}}\, e^{-\frac{1}{2}(X_t-\mu_{i,t})^T \Sigma_t^{-1} (X_t-\mu_{i,t})}. \tag{2.1}$$

The parameters are $\mu_{i,t}$ and $\Sigma_{i,t}$, which are the mean and covariance matrix of the distribution, respectively. For computational simplicity, the covariance matrix is assumed to be diagonal.

For every new pixel, a matching, an update, and a decision step are executed. In *the matching step*, the new pixel value is compared with the models of the mixture. A pixel is matched if its value occurs inside a confidence interval within 2.5 standard deviations from the mean of the model. In that case, the parameters of the corresponding distribution are updated according to (2.2), (2.3), and (2.4):

$$\mu_{i,t} = (1-\rho)\,\mu_{i,t-1} + \rho\,(X_t)\,, \tag{2.2}$$

$$\Sigma_{i,t} = (1-\rho)\,\Sigma_{i,t-1} + \rho\,(X_t - \mu_{i,t})\,(X_t - \mu_{i,t})^T\,, \tag{2.3}$$

$$\rho = \alpha G\left(X_t, \mu_{i,t-1}, \Sigma_{i,t-1}\right). \tag{2.4}$$

The learning rate, α, is a global parameter, and introduces a trade-off between fast adaptation and detection of slowly moving objects. Each model i has a weight, $w_{i,t}$, which is updated for every new image according to (2.5):

$$w_{i,t} = (1 - \alpha)\, w_{i,t-1} + \alpha M_{i,t} \; . \tag{2.5}$$

If the corresponding model introduced a match, $M_{i,t}$ is 1 , otherwise it is 0. Equations (2.2) to (2.5) represent *the update step*.

Finally, in *the decision step*, the models are sorted according to their weights. MGM assumes that background pixels occur more frequently than actual foreground pixels. For that reason, a threshold T based on these weights, is used to decide which models of the mixture depict background or foreground. Indeed, if a pixel value occurs recurrently, the weight of the corresponding model increases and, eventually, it is assumed to be background. In the rest of this chapter we will use the term *background models* to denote those models in the mixture that depict background.

If the current pixel value cannot be matched with any of the models, the model with the lowest weight is discarded and replaced by a normal distribution with a small weight, a mean equal to the current pixel value, and a large covariance. In this case the pixel is assumed to be a foreground pixel.

The next section shows the changes we propose to MGM, which result in a technique called Simple Mixture of Models (SMM).

2.4 Background Subtraction using SMM

The mixture of weighted Gaussian models used in MGM could be regarded as a collection of feature vectors, describing certain states of a pixel. In this sense, the feature vector consists of a weight, mean and variance (in case of gray values) or a vector of means and variances (in case of more-dimensional color spaces).

These feature vectors are then used within the matching, update, and decision steps. Our conviction is that the true strength of MGM does not lie in the use of the Gaussian models, but in the use of the mixture of weighted models within these three steps, allowing to deal with dynamic multi-modal environments.

Hence, we propose to use novel feature vectors, while inheriting and modifying the overall structure of MGM. The first feature is an intensity allowance based on statistics of photon noise. Secondly, we incorporate in the feature vector a maximum difference with previously recorded background values to cope with

gradual illumination changes. And finally, we include a mean and a dynamic upper and lower threshold (called $\phi_{up,i,t}$ and $\phi_{down,i,t}$, respectively), which replace the Gaussian parameters.

These new feature vectors form the new models used within SMM and accordingly the matching, update and decision steps are adjusted. To further improve the results, within the decision step, a shadow removal step is incorporated. Finally, to reduce the processing time, we introduce an analysis mask that decides which pixels in a frame will be processed according to the above techniques. The detection results of these pixels are subsequently interpolated to form a decision for the remaining pixels.

The entire process of SMM is visualized in Figure 2.1. First we use an analysis mask to decide which pixels of an image will be processed according to the matching, update and decision step. The matching step and the new feature vectors are explained in the next section. Subsequently, Section 2.4.2 discusses the new update step. Finally, Section 2.4.3 discusses the decision step, including the shadow removal step and usage of the analysis mask.

2.4.1 Matching step

The matching step, shown in Figure 2.1, is a modified version of the matching step of MGM.

The input of this step are those pixels that are selected by the analysis mask. For the other pixels in the image a decision will be made by interpolating the output of the analysed pixels.

Firstly, the new pixel value, that is being analysed, is evaluated according to an intensity allowance based on Skellam parameters to deal with the effect of photon noise (explained in the next section). If the difference of the new value and the mean of one of the models is smaller than the intensity allowance at that pixel position, a match is made.

Secondly, if no match was made, the pixel value is analysed to see if a gradual change in illumination has happened. Again, if a gradual change is detected, a match is made with the corresponding model.

Finally, if none of the above have raised a match, the new pixel value is compared with the means of the models using the dynamic upper and lower thresholds.

This last step is similar to the matching step of MGM, which uses the means and standard deviations of the Gaussian distribution. However, some papers

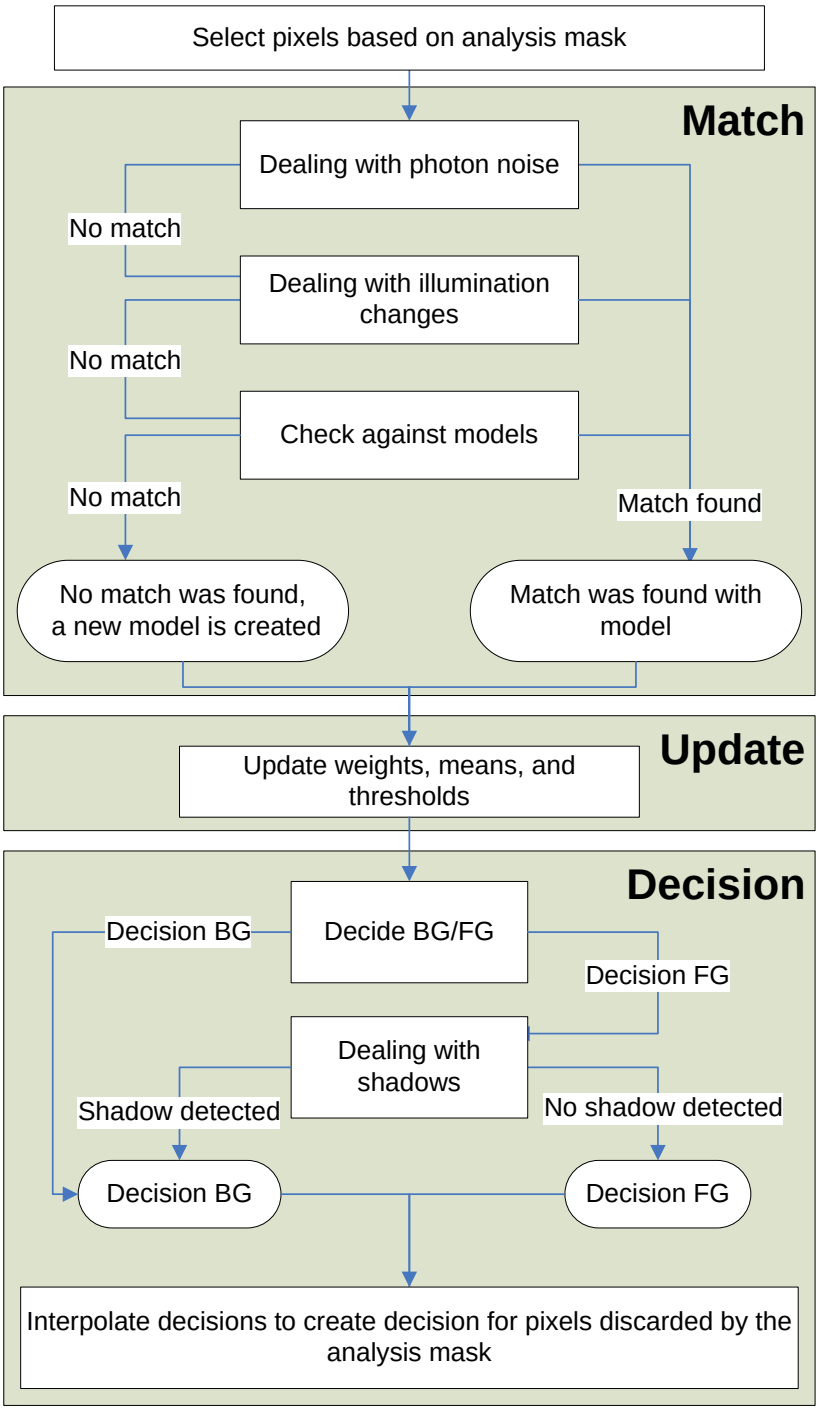


Figure 2.1: Flow chart representing the SMM algorithm.

have already shown that the assumption that data, gathered by surveillance cameras, follows a Gaussian distribution, does not always hold [56]. We believe that the true strength of MGM does not lie in the use of these Gaussian distributions, but in the use of the mixture of weighted models, which can deal with multi-modal environments. Additionally, in Section 2.4.2, we show how the calculation of the Gaussian probability density function, in the update step, can be avoided. Therefore, the conceptual use of the Gaussian distributions is discarded and we use a mean, and a dynamic upper and lower threshold.

Dealing with photon noise

Hwang et al. already claimed that photon noise is the most dominant noise component in CCD or CMOS cameras [56]. They made a prediction of the photon noise and applied this to create a more precise edge detection. Indeed, their results show that the Canny edge detection can be outperformed by taking the photon noise into account. In the next paragraphs we elaborate on their work and the noise model they use.

For each pixel position, they determine a Skellam distribution that models the differences in recorded pixel values over time. Subsequently, an acceptance region is created for this distribution. Finally, the maximum intensity difference that falls in this acceptance region is called the intensity allowance ΔX_A and is used to determine whether a change in pixel value is due to photon noise or to something else.

Their method is based on images of a static scene, so we extend it to multi-modal environments to make it usable in actual video surveillance systems. When dealing with multi-modal background environments, a single pixel value can follow multiple modes according to the given situation of the environment (e.g., a moving tree). Accordingly, for each mode m that we see during the gathering of the data, a Skellam distribution is determined. As such, we find for each pixel a number of intensity allowances $\Delta X_{A,m}$ according to the different modes. Next, we first discuss how Hwang et al. find ΔX_A, secondly we show how we extend their method to find $\Delta X_{A,m}$.

When using CCD or CMOS cameras the intensity is determined by the amount of photons that reach the camera. The number of photons is governed by the laws of quantum physics. Due to the statiscal nature of these photons the amount of photons that reach the camera will vary, resulting in photon noise. Such noise is usually modelled by a Poisson distribution [57], and the difference between two Poisson random variables is defined as a Skellam distribu-

tion [58].

The parameters of a Skellam distribution can be estimated by using the statistics of the Skellam distributions. The cumulative distribution function of a Skellam distribution is given by:

$$F\left(\Delta X;\mu_1,\mu_2\right)=\Sigma_{k=-\infty}^{\Delta X}e^{-(\mu_1+\mu_2)}\left(\frac{\mu_1}{\mu_2}\right)^{\frac{k}{2}}I_k\left(2\sqrt{\mu_1\mu_2}\right). \tag{2.6}$$

μ_1 and μ_2 are the means of the two Poisson distributions, and $I_k\left(z\right)$ is the modified Bessel function of the first kind [59]. The cumulative distribution function of a Skellam distribution is only defined at integer values and here ΔX represents the difference between two intensity values. If we can find the Skellam distribution for a certain pixel position, an allowance for intensity variation due to sensor noise can be determined. The acceptance region for an intensity difference ΔX can be defined as:

$$A\left(\Delta X;\mu_1,\mu_2\right)=F\left(\Delta X;\mu_1,\mu_2\right)-F\left(-\Delta X-1;\mu_1,\mu_2\right). \tag{2.7}$$

Equation (2.7) can be rewritten as:

$$A\left(\Delta X;\mu_1,\mu_2\right)=\Sigma_{k=-\Delta X}^{\Delta X}e^{-(\mu_1+\mu_2)}\left(\frac{\mu_1}{\mu_2}\right)^{\frac{k}{2}}I_k\left(2\sqrt{\mu_1\mu_2}\right). \tag{2.8}$$

The intensity allowance ΔX_A is determined by:

$$\begin{aligned}\Delta X_A \quad &= \quad \underset{\Delta X}{\arg\max}\; A\left(\Delta X;\mu_1,\mu_2\right)\\ &\text{s.t.} \quad A\left(\Delta X;\mu_1,\mu_2\right)\leq 1-\alpha_S.\end{aligned} \tag{2.9}$$

α_S determines the numbers of false positives [60]. The parameter α_S can be interpreted as follows: a high value means that the acceptance region and the intensity allowance is very small, so only very small differences between consecutive pixels will be regarded as created by photon noise. As such, if the photon noise causes larger differences, these will fall outside the acceptance region and the system will not regard them as being caused by photon noise, so more false positives will occur. Accordingly, a low α_S will increase the pixel difference that is assumed to be caused by photon noise. So, here the acceptance region is used to find the highest intensity difference that still matches the restrictions imposed by α_S. To benefit a practical implementation hereof, we give a recursive definition of equation (2.7), based on equation (2.8):

$$A\left(0;\mu_1,\mu_2\right) = e^{-(\mu_1+\mu_2)} I_0\left(2\sqrt{\mu_1\mu_2}\right)$$
$$A\left(\Delta X;\mu_1,\mu_2\right) = A\left(\Delta X-1;\mu_1,\mu_2\right) +$$
$$e^{-(\mu_1+\mu_2)}\left(\left(\frac{\mu_1}{\mu_2}\right)^{\frac{\Delta X}{2}} + \left(\frac{\mu_1}{\mu_2}\right)^{-\frac{\Delta X}{2}}\right) I_{\Delta X}\left(2\sqrt{\mu_1\mu_2}\right) \tag{2.10}$$

The parameters of the Skellam distribution (μ_1 and μ_2) can be found by estimating the mean μ_S and the variance σ_S^2 of the Skellam distribution [58]. The parameters and the mean μ_S and the variance σ_S^2 are related as follows:

$$\mu_S = \mu_1 - \mu_2, \tag{2.11}$$
$$\sigma_S^2 = \mu_1 + \mu_2. \tag{2.12}$$

Hwang et al. discussed three methods to obtain μ_S and σ_S^2. The first one calculates the average intensity differences between the same pixel positions in consecutive frames. A second approach is to search for a homogenously colored patch in an image and to calculate the average intensity difference within this patch. The third approach shows how they automatically select a homogeneous patch to calculate the parameters of the Skellam distributions.

At this point, for each pixel of a frame ΔX_A can be found. Next we discuss how the method of Hwang et al. can be extended to find $\Delta X_{A,m}$ for different modes.

Since we face large environments surveilled by a camera, the amount of large homogeneously colored patches will be small, due to textures and details in the environment. Therefore we adopt the first method of Hwang et al. to make an estimation of the Skellam parameters. In practical scenarios, when dealing with multi-modal background environments, it can be hard to get consecutive images representing the same background values. This multi-modality manifests itself in regions where a single pixel value can follow multiple modes according to the given situation of the environment (e.g., a moving tree). Different colors correspond to different photon noise distributions so, accordingly, for each mode m that we see during the gathering of the data, the parameters of the according Skellam distribution ($\mu_{S,m}$ and $\sigma_{S,m}^2$) need to be calculated.

To detect that a pixel value belongs to a new mode, we use variables Y_m that correspond to the first pixel value that is seen from a certain mode. Here, m stands for the current mode; initially it is set to 0. We assume that, if a pixel value X_t differs more than 30 from this variable, it is caused by the appearance of a new mode. In that case we create an additional variable Y_{m+1}, equal to

the current pixel value, to represent this new mode. As such, all the data can be assigned to the found modes. For each of the modes detected this way, a $\mu_{S,m}$ and $\sigma^2_{S,m}$ is created and maintained:

$$\mu_{S,m} = \frac{\sum_t (X_t - X_{t+1})}{n}, \tag{2.13}$$

$$\sigma^2_{S,m} = \frac{\sum_t (\mu_{S,m} - (X_t - X_{t+1}))^2}{n}, \tag{2.14}$$

$$\text{s.t. } |X_t - Y_m| \leq 30.$$

The value of 30 to determine whether a pixel value belongs to a different mode is based on work of Zang et al [61]. They have done an in-depth analysis of the optimal parameter settings for MGM (in which each mode is represented as a Gaussian distribution). They showed that when creating a new Gaussian model in the mixture, an initial standard deviation equal to 12 gives good results. This means that the matching criteria of 2.5 standard deviations results in a pixel difference of 30. Consequently, we use this value to decide if a new pixel value corresponds to a new mode.

As such, if a pixel is part of a dynamic background, each mode will result in a different Skellam distribution and corresponding intensity allowance $\Delta X_{A,m}$.

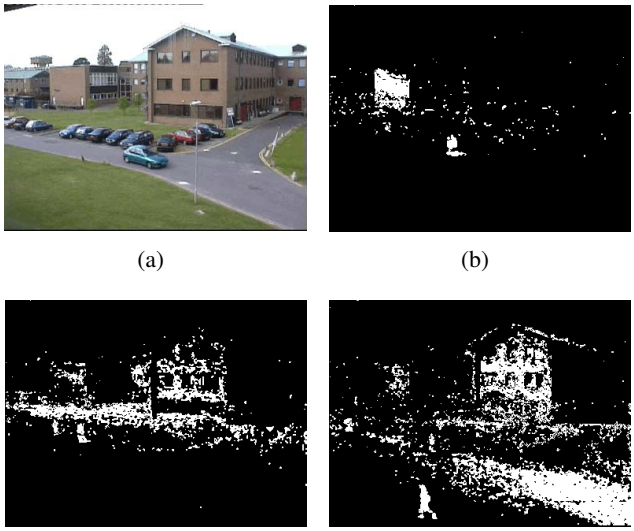
The first step of the matching step is to use these intensity allowances to find a match with one of the models in the mixture. So for each model i in the mixture we need to find the appropriate intensity allowance by comparing the mean of the model $\mu_{i,t}$ with the recorded variables Y_m for that pixel position. Subsequently, the corresponding $\Delta X_{A,m}$ is used to check if the new pixel value can be matched with that model. If the difference between the new pixel value and the mean of the model is smaller than $\Delta X_{A,m}$, a match is made. As a result, if the value of a background pixel has changed due to photon noise it will still be matched by the according model. Using the intensity allowances, which are calculated based on the statistics of the photon noise in the image, allows to deal with different levels of noise in the video sequences. If no match was found with the models, the matching step continues with checking whether an illumination change has occurred.

Dealing with illumination changes

One of the major problems one faces while creating an automated video surveillance system, is changing lighting conditions [6], [62]. Sudden gradual changes in illumination (e.g., due to a cloud gradually changing the lighting conditions of the environment) are hazardous for traditional pixel based background subtraction systems. The changes can manifest themselves over a long period (e.g., day turning into night) or over a very short period (e.g., cloud blocking the sun). MGM is an adaptive system and the learning rate is typically small (α is usually less than 0.01), so it is able to deal with the slow changes of light. These slow changes will result in a gradual change of the model parameters such that the new background pixel values will always be matched by the updated background models. However, dealing with fast gradual changes is difficult. If the learning rate is low, the models will not be adapted fast enough to incorporate these fast gradual lighting changes. Consequently, a pixel that is subject to fast gradual lighting changes will be considered to be foreground. MGM relies on a learning rate for both the update of the models parameters and the weights. Increasing the learning rate is not an option since parts of slow moving objects will be considered as part of the background. The models will also be highly sensitive for small noise and adapt their parameters too fast. As such, fast changes in lighting affect MGM drastically, as shown in Figure 2.2. The figure shows the results of applying MGM to the PetsD2TeC2 sequence (with a resolution of 384x288) provided by IBM Research [63] at several time points. Black pixels depict the background, white pixels are assumed to be foreground. The images depict a fragment of the monitored environment being subject of changing illumination circumstances in a relatively short time period, causing a repetitive increase of certain pixel values. As can be seen, the falsely detected regions can range from very small regions of misclassified pixels, to regions encompassing half of the image.

Figure 2.3 shows the influence of the lighting changes on the number of false detections (called false positives) for every 50th frame, when applying MGM. As can be seen, the lighting change, occurring at frame 2100 to 2850, increases the number of false positives drastically.

The shown sequence contains a good example of the problem we face. The change in illumination here is very fast (30 seconds) and gradual (a human observer has a hard job noticing it), but the effect is large. The reason why MGM fails is the following: in the matching step, MGM only takes the difference of a new pixel value with the means of the models into account and does not regard the actual previous value for that pixel. Although the means are updated



(a) (b)

(c) (d)

Figure 2.2: Impact of illumination changes on different time instances on the PetsD2TeC2 sequence: a typical input image (a), the output of MGM, with $\alpha = 0.01$, for frames 2115 (b), 2230 (c), and 2400 (d).

with every new pixel value, the adaptation is not fast enough to encompass the lighting change. As such, if values increase consistently, but with small intermediate steps, the pixel values will eventually fall out of the range of the models.

To solve this problem, we make the following assumption: if the value of a pixel was considered to be background and there is only a small change, then the new pixel value is background too. Consequently, we assume that a small change is not due to the appearance of a foreground object. As such, we need to store the actual previous pixel value that was regarded as background. For every background model in our mixture the last pixel value that matched the model is stored. When analysing a new pixel value, we look for the smallest difference with the stored background pixel values. If this difference is small, a match with the according background model is immediately made. Consequently, the changes proposed here will only be visible if a new pixel value has a small difference with one of the previous background values.

To monitor the difference, two independent thresholds (named the gradual thresholds $\phi_{grad,up,i,t}$ and $\phi_{grad,down,i,t}$) are used to cope with illumination

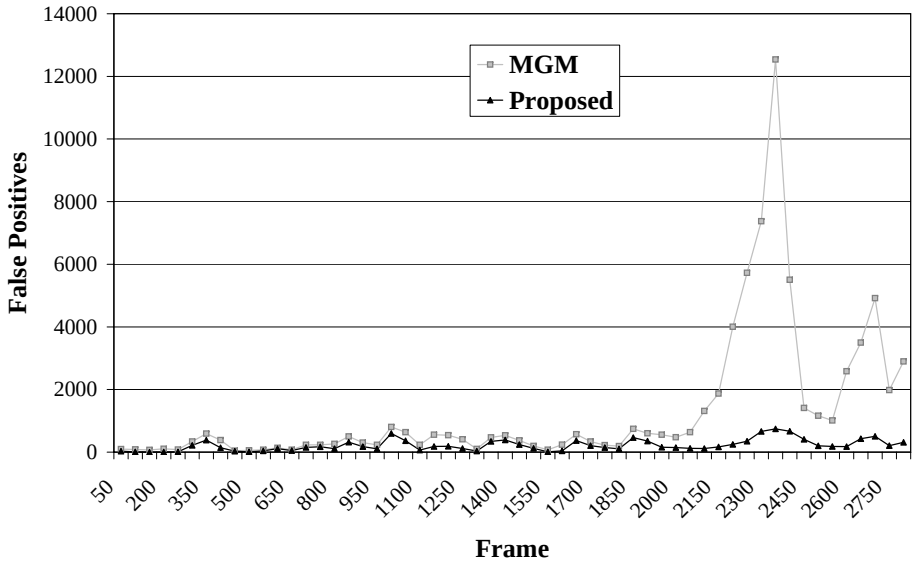
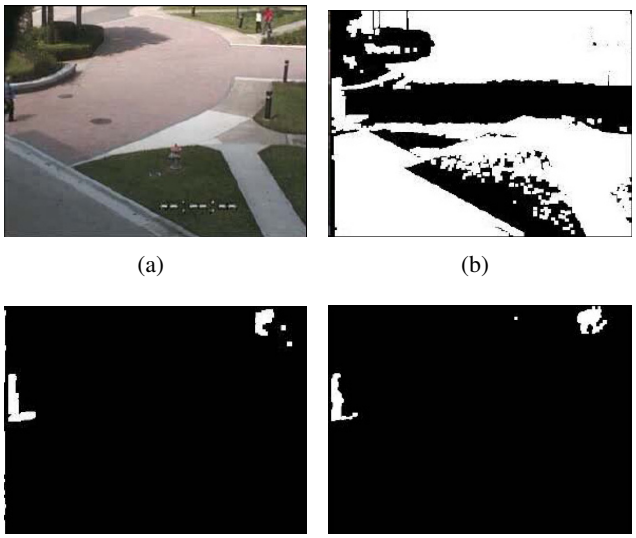


Figure 2.3: Impact of illumination changes on the number of false positives for the PetsD2TeC2 sequence.

changes that cause the scene to become lighter and darker. Since we already have a normal upper and lower threshold $\phi_{up,i,t}$ and $\phi_{down,i,t}$, the gradual thresholds are made dependent on these normal thresholds. In our experiments, the gradual thresholds are set to be 70% of the according normal thresholds. This value was experimentally determined on sequences with gradual illumination changes on which MGM fails. Higher values ($\geq$ 90%) can result in detection failures, since pixels corresponding with moving objects might be concidered to belong to a illumination change. If low values are used, not all illumination changes can be dealt with.

In Section 2.2 related work was discussed, including techniques that give a solution to the problem of gradual illumination changes. These techniques are based on pixel, region, and frame level processing, using a gradients-based algorithm, whereas our proposed solution to the illumination changes is solely pixel-based. Figure 2.4, shows the visual results of MGM, the work by Javed et al. [9], and the use of the gradual thresholds, for a scene that suffers from an illumination change. The figure shows that our pixel-based approach achieves similar results as more complex techniques in coping with illumination changes.



(a) (b)

(c) (d)

Figure 2.4: The influence of quick illumination changes in an outdoor surveilled environment: the current frame (a), output from MGM (b), output from [9] (c), and output by using the gradual thresholds of SMM (d).

We want to stress that the gradual changes in lighting described here are not only present in outdoor surveillance scenarios. When analysing the PETS 2007 datasets and the AVSS 2007 datasets, we discovered that the same problems arise in typical indoor scenarios. The influence of a gradual lighting change in an indoor environment on MGM can be seen in Figure 2.5. It shows input images of an airport and a subway, and the detection results of MGM and the usage of gradual thresholds. The origin of a gradual change of pixel values can be a change in illumination, soft shadows cast by objects or background, reflections, etc. Whatever the origin is, when fast gradual changes occur, it is most likely not due to the appearance of foreground objects. Consequently, using our proposed change will result in less falsely detected pixels, while keeping the same amount of true detections.

Note that within our proposed algorithm, when an illumination change occurs, initially it might be regarded as photon noise. However, when the illumination change continues, it will exceed the intensity allowance $\Delta X_{A,m}$ and will no longer be regarded as photon noise. From that point on the check against the gradual thresholds is triggered, so we can detect the illumination change.

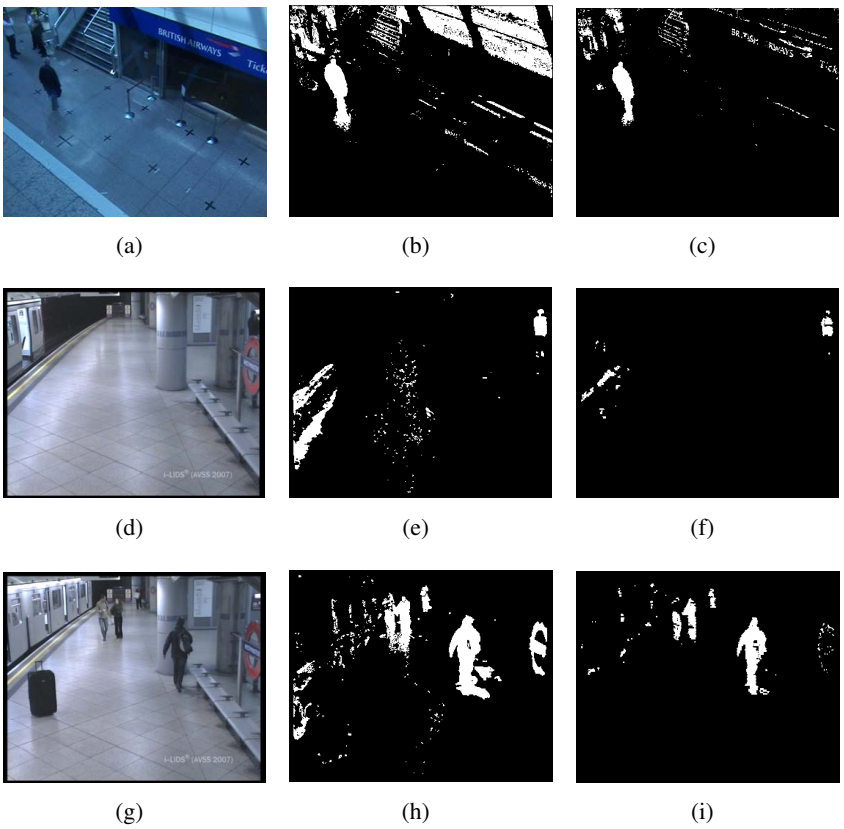


Figure 2.5: Left: current frame, centre: output by MGM, right: output by proposed system. First row shows the "s00-thirdview" sequence of the PETS2007 dataset, image 170. Second row shows the "AVSS AB easy" sequence of the AVSS2007 dataset, image 1290; image 3790.

Matching with Simple Models

Finally, if none of the above have raised a match, the models are evaluated using the upper and lower thresholds, $\phi_{up,i,t}$ and $\phi_{down,i,t}$, respectively. In case that the new pixel value is larger than the average, a match occurs if the difference is smaller than $\phi_{up,i,t}$. Likewise, for new pixel values that are smaller than the average, the difference is checked against $\phi_{down,i,t}$.

If no match has been found, the model with the lowest weight is discarded and a new one is created. The pixel value is then considered to match with this new model. After the matching, the same update steps as in MGM to adjust the mean, weights, and thresholds of the models are performed, with some modifications, discussed in the next section.

2.4.2 Update step

First, the update step of MGM demands heavy calculations, since for every matched model the probability distribution function is used to calculate the parameter ρ. Previous research has shown that this calculation can be skipped since this value usually is very small and does not change much [64]. We adopt this in our system and use a constant parameter ρ. This allows us to adjust the learning rate α and the update speed ρ separately. The learning rate influences the weights and consequently the speed at which foreground objects are learned into the background. The update speed represents how fast the individual models adapt their parameters to new matched pixel values.

Initially, we set the learning rate high to deal with moving objects that are present in the scene when the algorithm is started. Since the models are initialized based on the first pixel values they get, these foreground objects are wrongly considered to be part of the actual background. When these objects move, a ghost will appear, since the actual background pixels that are revealed don't match the misinterpreted background models. A high learning rate allows to learn these pixels into the background fast and gets rid of these initialization ghosts very quickly. This is specifically interesting when monitoring crowded scenes with fast moving objects (such as highways). Figure 2.12(f) shows how MGM suffers from these initialization ghosts. The system wrongly detects cars that were present in the first frame. The figure shows that our proposed system can deal with these since the initial high learning rate allows to quickly model the actual background. Note that this does not present a solution to deal with ghosts occurring during the surveillance of the scene (e.g., when a static background object like a parked vehicle starts to move). We refer to

other techniques to deal with this problem [65].

We reuse equations (2.2) and (2.5) to update the means and weights of our models. In SMM, we do not use the standard deviations to find a match but have an independent upper and lower matching threshold. The separation of these thresholds allows to update them individually according to following equations:

$$\phi_{up,i,t} = (1 - \rho/\,2)\phi_{up,i,t-1} + \rho\,(X_t - \mu_{i,t})\,, \tag{2.15}$$

$$\phi_{down,i,t} = (1 - \rho/\,2)\phi_{down,i,t-1} + \rho\,(\mu_{i,t} - X_t)\,. \tag{2.16}$$

If the difference with the new pixel value is smaller than half of the threshold, this will result in a decrease of the threshold. Otherwise the threshold will be increased and the amount of decrease and increase depends on ρ.

2.4.3 Decision step

In this step, a decision is made about the pixel being processed. This decision is divided in three steps and explained in the next sections. Firstly, the mixture of models is analysed based on the weights to decide which models depict foreground or background. That way a decision is found for the current pixel based on the matched model. Next, if the decision was foreground, we apply shadow analysis to correct the decision if necessary. Lastly, a decision is made for those pixels that were not analysed due to the analysis mask.

Decide between Background or Foreground

We alter the sorting of models as it is done in MGM, by introducing two preliminary steps. In most of the cases there is only one model that corresponds to the background. As such, the first step should be to check whether the weight of the matched model is larger than the predefined threshold. If not, the next step is to check if the weight of the matched model is smaller than all other values. In this case the model surely depicts foreground. Since these two cases occur frequently, a costly sorting step can be avoided. If neither of the initial conditions is fulfilled, the sorting as introduced in MGM can be applied and the threshold T is used to decide which models of the mixture depict background or foreground.

Dealing with shadows

MGM is not capable of dealing with shadows. Several techniques have been proposed to deal with this problem [53,54]. Since we are dealing with a multi-modal environment, we present a shadow removal scheme that follows the dynamics of the actual scene. When the decision for a certain pixel is foreground and all color components are darker than the corresponding background values, color components are normalized according to the following equations (we discard the time notation t here and X_r, X_g, and X_b represent the R, G and B color components respectively):

$$X_R = \frac{X_r}{X_r + X_g + X_b}, \tag{2.17}$$

$$X_G = \frac{X_g}{X_r + X_g + X_b}, \tag{2.18}$$

$$X_B = \frac{X_b}{X_r + X_g + X_b}. \tag{2.19}$$

These values are then compared with the normalized means of each background model. A shadow parameter (δ) is introduced for this comparison, so for each background model j present in the mixture we apply equation (2.20):

$$D = 0 \text{ if } (|X_R - \mu_{j,R}| < \delta) \wedge (|X_G - \mu_{j,G}| < \delta) \wedge (|X_B - \mu_{j,B}| < \delta)\,. \tag{2.20}$$

D stands for the decision and if equal to 0 a shadow has been detected. Consequently, the decision result of the temporal background subtraction will be converted to background.

Prati et al. divide shadow detection systems in four categories [53]. They first make a separation between deterministic and statistical approaches. The former approaches use an on/off decision about the pixel origin, the latter use probabilistic functions to describe the class membership. The deterministic approaches are further divided in systems that use model-based knowledge and systems that do not. The statistical approaches are divided in parametrical and non-parametrical methods. Prati et al. chose for each category a representative technique and made an extensive comparison. They concluded that, for general purposes with minimal assumptions, the deterministic non-model-based approach gives the best results. For the deterministic non-model-based approach they choose the Sakbot system [54]. This technique uses the HSV color

space and relies on 4 independent parameters to detect shadows. Our shadow detection system, which also falls in the deterministic non-model-based approach, only uses one parameter δ and avoids a costly conversion to the HSV color space. Shan et al. present a shadow removal scheme in HSV color space that achieves even better results than the Sakbot system [50]. Therefore, we give an objective and subjective comparison of our system with their work in Section 2.6. In this evaluation δ is set to 0.01 (a value determined by analysing a training sequence) and this value is consequently used for all sequences.

Analysis Mask

A common solution to increase the processing speed of pixel domain object detection techniques, is to decrease the frame rate at which the environment is monitored. Only regarding half or one fourth increases the processing speed linearly. However, temporal consistency is very important to successfully track moving objects, so working with a reduced frame rate makes this difficult. During tracking, detected objects are matched with objects detected in previous frames to find the object trajectories. This matching also allows to identify the objects in newly captured frames.

A second approach is to reduce the image resolution. Evidently, when downscaling the images we get a linear increase in speed of the surveillance systems. In most cases the aspect ratio of the images is kept the same. Thus, downscaling the image results typically in an image that is one fourth of the original. If one changes the aspect ratio (e.g., by only analysing the odd rows in an image) the objects become deformed, which can be bad for classification. Nowadays, captured surveillance images tend to be small. Large-scale surveillance systems work with multiple feeds, captured by a distributed camera system, and covering large sites. Therefore, to reduce bandwidth usage and processing times, low-cost cameras capture images in small resolutions (CIF to QCIF). Additionally, in the case of a surveillance camera covering wide ranges of an environment, specific objects of interest tend to be very small and only represent a couple of pixels of the entire frame. Therefore, simply downscaling the images will result in additional detection failures.

Finally, a popular method to increase the speed, is to apply course motion detection first before analysing the entire frame [66]. In this case, simple frame differencing is used. When nothing happens in the scene, the frames are skipped; vice versa, if a large amount of motion is detected, the frame is analysed. Although this works well in static controlled environments (e.g., indoor video surveillance), it would be bad in more complex environments (like

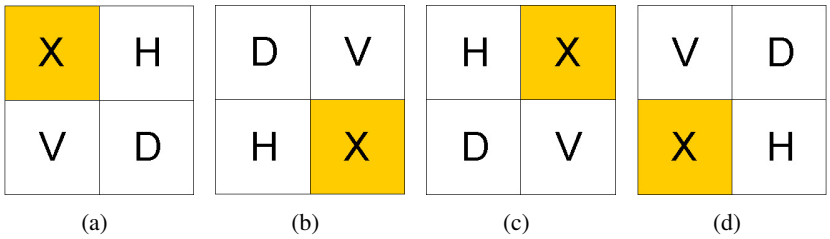


Figure 2.6: 2x2 patches that constitute the analysis masks for four consecutive frames. The patch for the first (a), second (b), third (c) and fourth (d) frame are shown.

outdoor scenes with movement of trees or crowded scenes) since movement will be detected everywhere. As such, the gain in speed is very dependent on the actual monitored environment and the activity within.

To increase the processing speed, we have chosen to use a checkerboard pattern for the evaluation of the frames. We use a mask, equal to the image size to decide which pixels to evaluate when processing a new image. The mask is constructed by 2×2 patches, which are repeated over the entire image. We use four different masks to make sure that every pixel in an image is evaluated at least once in four consecutive images. So for every new image the mask is switched. Figure 2.6 shows the patches that constitute the different masks for four consecutive frames. The positions marked with an X are the pixels for which the normal background subtraction is executed. The results are then used to make a decision about the surrounding pixels. Positions denoted with H will apply horizontal interpolation. This means that for the pixel at that position, the results of the pixels to the left and right are evaluated. If both pixels were considered foreground, then the decision for the current pixel is foreground, otherwise it is background. For the positions in the patch denoted by V we will take the upper and lower pixels into account. Finally, for the positions denoted by D , the upper left, upper right, lower left and lower right pixels are used. If three of them are considered foreground, the current pixel is foreground.

We want to achieve high sensitivity regarding the moving objects. Therefore, the interpolation of pixels might be insufficient. Typically, edges of detected objects can be falsely interpolated as background. Therefore, we add another layer of intelligence to the analysis mask. If the decision of the background subtraction for a certain pixel was foreground, then we update the masks such

that the surrounding pixels will be evaluated in the next frames. For each of these surrounding pixels we use a decreasing counter to decide whether it is analysed or not. Every time a foreground pixel is detected, the counter is reset to its original value. If a background pixel is detected, we decrease the counter until it reaches zero. In that case, the initial mask values are used again. Consequently, if an object enters the monitored environment and some of the objects' pixels are correctly detected, the immediate surroundings will be evaluated in the next frames, resulting in a more accurate detection.

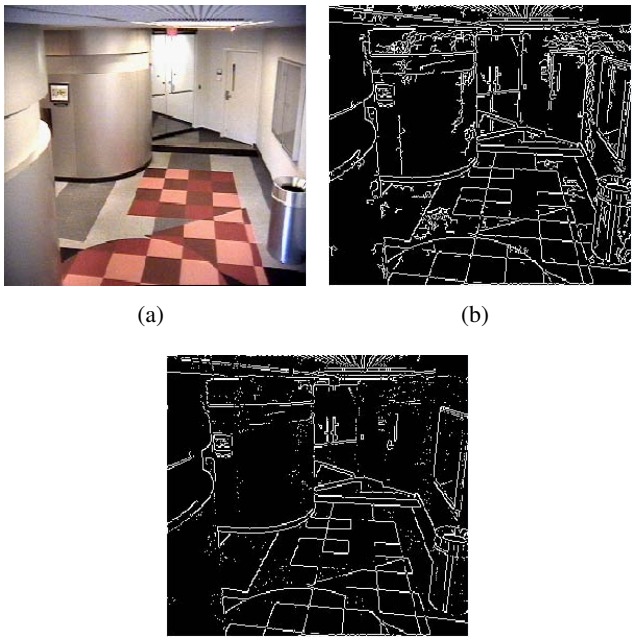
The latter step increases the detection accuracy but has as a disadvantage that it introduces a trade-off between detection and speed. Since the mask is now made dependent on the detection results, environments with many moving objects will result in the analysis of large parts of the frame. On the contrary, if no moving objects are present only one fourth of the pixels in a frame are analysed, resulting in faster execution times. If the speed needs to be kept constant, a maximum could be put on the number of additional pixels that are analysed and on the size of the counters used.

Moreover, the mask acts as a morphologic filter since very small blobs of pixels are deleted in advance. Objects that are large enough (larger than the minimal distance between the analysed pixels in the mask) are still detected due to interpolation. This is an advantage compared to plain frame rate reduction, since temporal consistency is very important for tracking of objects. Normal spatial reduction (down scaling of images) results in certain positions of the frame that are never evaluated. Especially when the surveillance system is monitoring a wide outdoor environment, some moving objects might constitute only a couple of pixels in the image. With our masks every pixel position in the surveilled content will be analysed once during four consecutive frames.

The disadvantage of the analysis mask is that edges of moving objects might be less accurately detected. However, using the spatial segmentation, as explained in the next section, reduces this effect.

2.5 Extended SMM (eSMM)

SMM takes for each pixel only the past pixel values into account, so only temporal information is used. In this section, we will include spatial information (such as neighbouring pixel values) to improve the temporal detection of SMM. For this approach, in parallel with SMM, the image is segmented into different regions according to edge information. For this purpose we first introduce our edge detection algorithm in Section 2.5.1. These edges are then



(a) (b)

(c)

Figure 2.7: Results of edge detection on Indoor sequence showing: the current frame (a), output from Canny edge detection (b), output from Skellam edge detection (c).

used to make an entire segmentation of the image, which is discussed in Section 2.5.2. Finally, this segmentation is combined with the output of SMM in Section 2.5.3 to improve the results of the temporal object detection.

2.5.1 Edge detection

We use the edge detection technique proposed by Hwang et al. in [56] to find edges in the image. Since this technique incorporates the photon noise, it achieves better results than the well-known Canny edge detection technique [67]. Hwang et al. worked on static images, or sequences of static images, resulting in an intensity allowance for every pixel position of the frame. However, we have the intensity allowances for different modes of the background as discussed in Section 2.4.1 so the edge detection can be further improved. Figure 2.7 shows the typical differences between the Canny edge detector and our edge detector based on the Skellam distributions. We have used 100 frames

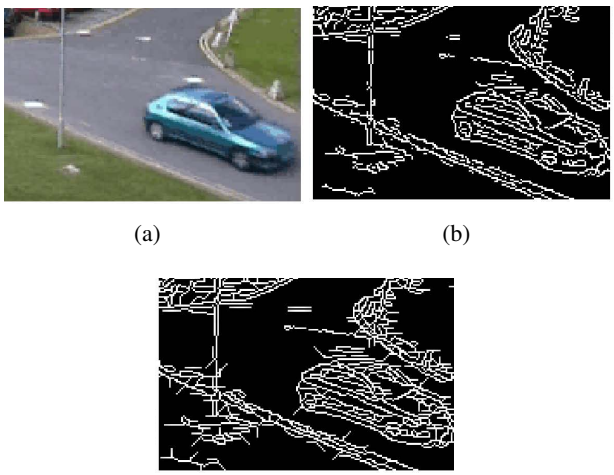


Figure 2.8: Results of Skellam edge detection: the current frame (a), output from Skellam edge detection (b), output from extended Skellam edge detection (c).

to determine the Skellam parameters and to create the intensity allowance data. Our Skellam-based edge detection succeeds in eliminating several edges that are due to noise. For every new image this edge detection is done, since in a dynamic environment edges tend to be highly dynamic themselves. Although the edge detection gives good results, it is not optimal and edges can be discontinuous. To segment the image based on these edges, it is important to have as many closed regions as possible. Therefore, an extension is built on the output of the edge detector. Loose ends are detected with a 3x3 search window and the ends are extended in the direction of the edge, we use an extension of 5 pixels at most. If more pixels are used, false segmentations will occur more frequently due to noise. Considering that surveillance cameras typically capture an entire environment, it is assumed that many small regions will be found. Therefore 5 pixels should be sufficient to close possible wholes in the edges. This parameter is dependent on the resolution of the images and the complexity of the monitored environment. Figure 2.8 shows the result of applying the canny edge detector and the extended version on an image. As can be seen, the extended version is able to close some regions that were left open.

2.5.2 Segmentation

After the extension step, the image is divided into several closed regions and we use a flood fill operation to give each segment a different value. The pixels representing the edge positions get the value of the smallest neighbouring region. This way, border pixels of small objects are assigned to the region of the object itself. As such, every pixel in the image can be assigned to one and only one region. Fast segmentation is an important factor since the results will be used in the further processing of our surveillance system. Although image segmentation techniques exist that obtain better results, these are more complex and less suited for processing surveillance video data in real-time [68].

2.5.3 Combining SMM with Spatial Image Segmentation

The goal is to improve the results of SMM by using the edge-based segmentation. To accomplish this, we use a two-pass matching step, as shown in following pseudo-code and explained in the next paragraph:

```
for each pixel(x,y) in frame t
  if SMM(x,y) == FG
    nrPix(segment(x,y))++
for each pixel(x,y) in frame t
  if SMM(x,y) == BG
    if nrPix(segment(x,y)) > 0.8*size(segment(x,y))
      SMM(x,y) = FG
    else
      if nrPix(segment(x,y)) > 0.6*size(segment(x,y))
          if smallestDiff(x,y) > theta * threshold(x,y)
            SMM(x,y) = FG
```

First, we store for each segment in the segmented image the number of corresponding foreground pixels resulting from SMM. In a second step, we go over all background pixels resulting from SMM and see which segments correspond with their location. If this segment contains many foreground pixels, it is assumed that the entire segment should represent foreground. The decision hereof happens according to a hysteresis threshold. When the segment has more than 80% foreground pixels, we denote each pixel of the segment as foreground. If there are less than 60% foreground pixels nothing happens. Finally, if there are more than 60% and less than 80% foreground pixels, an extra control step is applied. In that case the pixels of the segment are only

regarded as foreground if they differ enough from the background means. The smallest difference of the current pixel with the means of the background models is searched. If this difference lies within a smaller threshold than the one related to the background model the decision remains background. In the other case the pixel is denoted as foreground. This extra control step prevents that small, wrongly segmented, regions will be entirely regarded as foreground.

As such the detection result of SMM is extended to comply with the segmentation result. This way, foreground pixels that were not detected by only using the temporal information in SMM, can still be found. The segmentation only corrects pixels denoted as background. The number of detected foreground pixels will therefore only increase. We call the system extended SMM (eSMM).

Note that within eSMM the numbers of pixels that are falsely regarded as foreground, can increase. If a segment already contains many pixels that are wrongly considered by SMM to be foreground, caused by shadows, severe noise, etc., the entire segment might be regarded as foreground, increasing the false detections even more. This effect is discussed in the next section and it is shown that the effect is minor since SMM puts a lot of effort to minimize the number of false alarms.

2.6 Experimental Results

In this section we evaluate our proposed algorithm by comparing it to related work. First, we make an objective comparison to evaluate the accuracy of the proposed algorithm. Next, visual examples are given to show the added value of our system. Finally, a rough comparison of processing speed is given. Note that other comparisons (e.g., memory usage) could be usefull, but we restrict ourselves to the evaluation means that are typically used in the related work.

2.6.1 Objective Comparison

To evaluate our system, we present Receiver Operator Characteristic (ROC) graphs. If a real background pixel is misclassified as foreground, it is called a false positive (FP). If a foreground pixel is not detected, it is called a false negative (FN). Correctly classified foreground or background pixels are called true positives (TP) or negatives (TN), respectively. To find the false positives and negatives we have made a manual ground truth annotation for each sequence for every 50th frame. This ground truth was consequently compared

with the output of each algorithm to obtain the number of false positives and negatives for those frames. Finally, these values are summed for the entire sequence to create the ROC graphs. The x-axis shows the False Positive Rate (FPR), which defines how many incorrect positive results occur among all negative samples available during the test. In this case, it represents the number of pixels that were incorrectly considered as foreground, among all the real background pixels, called real negatives:

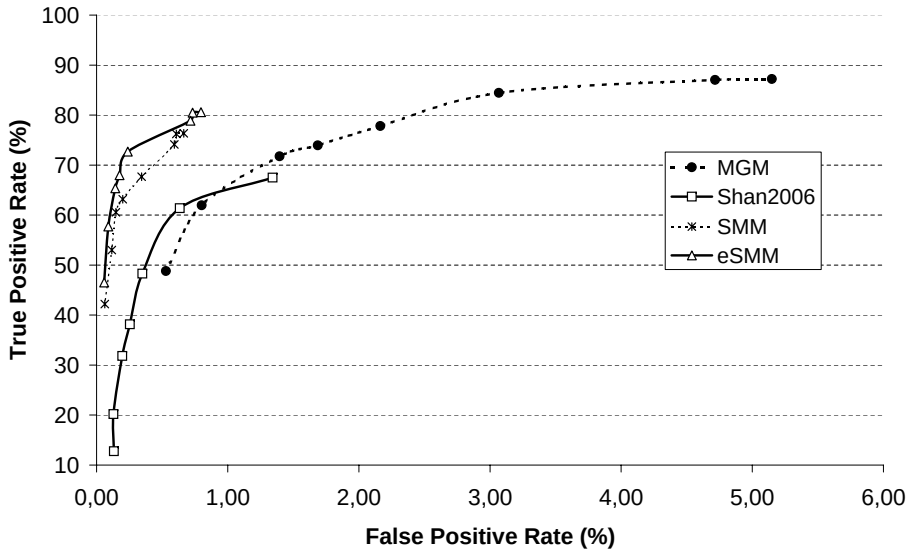
$$FPR = \frac{FP}{RealNegatives}. \tag{2.21}$$

The True Positive Rate (TPR), shown on the y-axis, is the sensitivity and denotes the percentage of the real foreground pixels, called real positives, that were correctly classified:

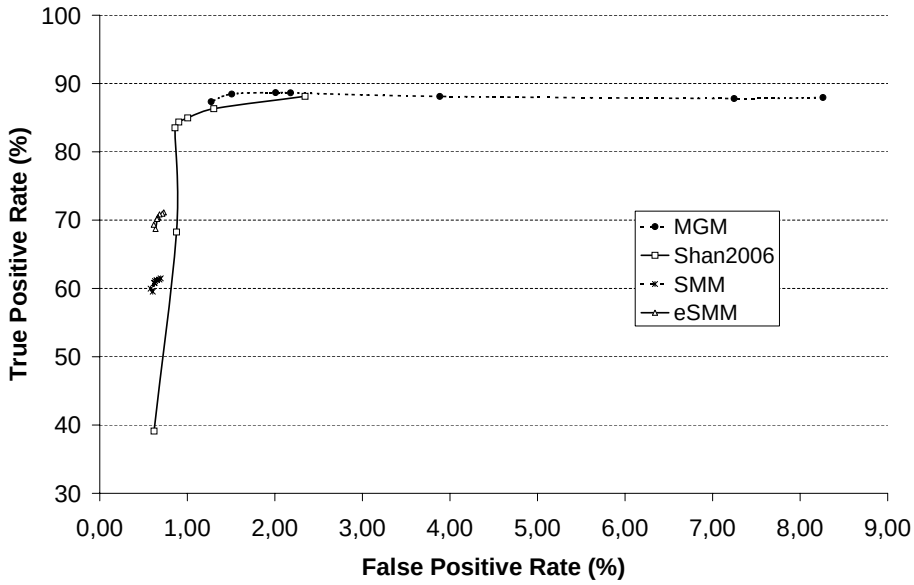
$$TPR = \frac{TP}{RealPositives}. \tag{2.22}$$

Good systems obtain a high TPR and low FPR. To get different values for the ROC curves, the learning rate α takes values from 0.0001 (very slow learning, the models are kept almost static) to 0.05 (very fast learning, new pixel values have much influence on the models). For each value of α, the TPR and FPR is calculated and plotted. Values recorded with a high learning rate will typically be situated to the lower left corner of the graph. Indeed, if a high learning rate is used, the parameters and weights of the models are adapted more drastically so they follow the dynamics of the background, resulting in less false positives. Additionally, slow moving objects will be learned into the background faster, resulting in more false negatives. When using a slow learning rate, objects will not be learned quickly into the background, moreover, the models are not capable of adapting to small variations of the background. This results in less false negatives and more false positives. In many surveillance scenarios, the amount of pixels that represents background is very large. Especially in outdoor environments, the moving objects will only occupy a small percentage of the scene. Therefore, the FPR needs to be very small, since for an image of CIF resolution (352 by 288 pixels) a FPR of 1% means we have about 1000 pixels misclassified as foreground. If the size of the objects in the images is small, the FPR needs to be very small too. Accordingly, having a TPR of 60% means that if there is an object encompassing 100 pixels, we detect 60 pixels of it.

Figure 2.9, Figure 2.10, and Figure 2.11 show a quantitative comparison between MGM, the algorithm of [50] (called Shan2006), SMM, and eSMM for

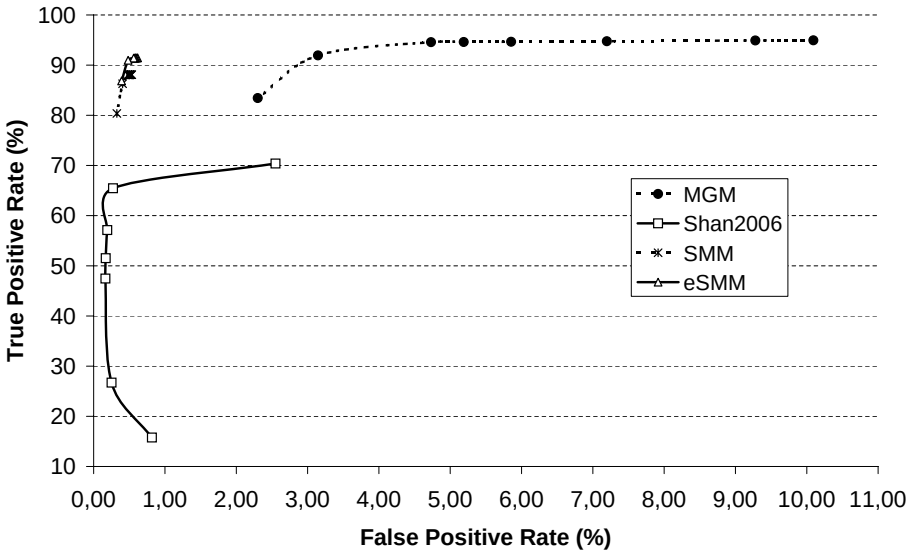


(a)

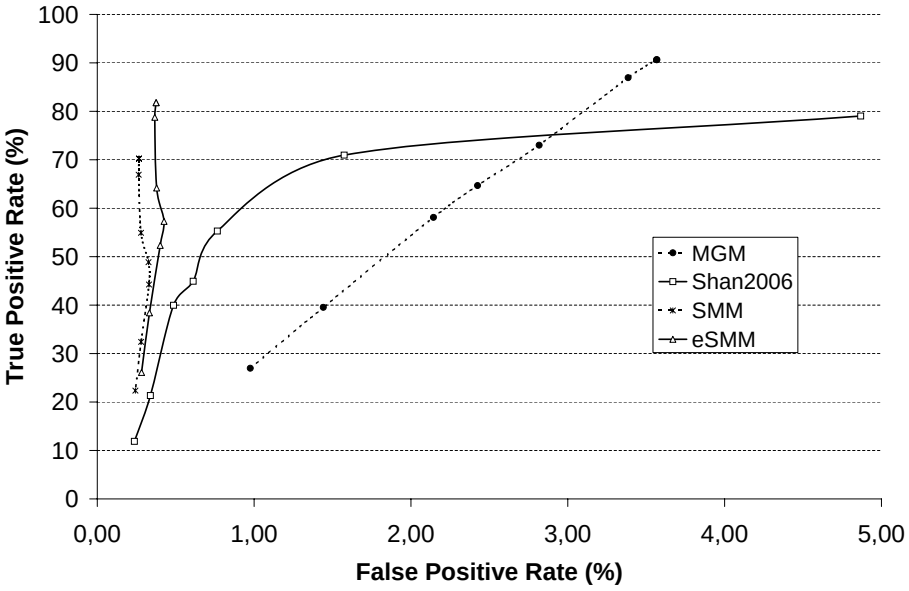


(b)

Figure 2.9: ROC graph for MGM, Shan2006, SMM, and eSMM on outdoor sequences: the PetsD2TeC2 sequence (a) and the Highway sequence (b).



(a)



(b)

Figure 2.10: ROC graph for MGM, Shan2006, SMM, and eSMM on indoor sequences: the Indoor sequence (a) and the Ismail sequence (b).

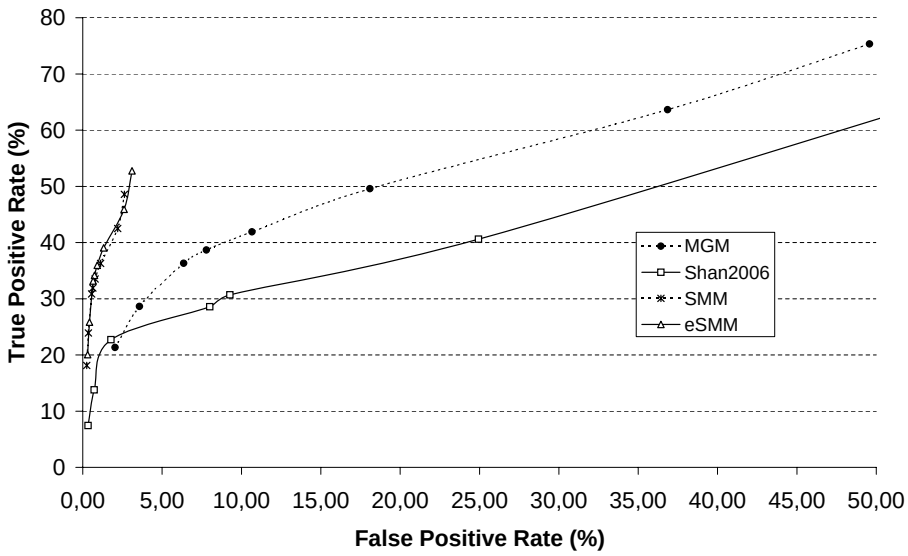


Figure 2.11: ROC graph for MGM, Shan2006, SMM, and eSMM on the Thirdview sequence.

challenging outdoor and indoor sequences. As can be seen in the figures, SMM has less erroneously interpreted background pixels than MGM and Shan2006. This is because we take the photon noise, gradual lighting changes and shadows into account. However, in most cases SMM introduces more false negatives. This is mostly due to parts of real foreground objects that are regarded as shadows or that fall in the intensity allowance of the models. We see that the same occurs for Shan2006. However, on average SMM has less false negatives (a higher TPR) than Shan2006. Moreover, the use of the edge-based image segmentation in eSMM increases the TPR even further. The combination of the spatial and temporal information introduces again some false positives, but it performs still better than MGM and Shan2006. This increase in false positives is mostly due to background pixels, erroneously interpreted as foreground by SMM that are extended by the segmentation technique. Indeed, if many of these misdetections are situated in the same segment, the segment will be regarded as foreground, introducing false detections. Additionally, since the segmentation technique used in our system was built to perform in real-time, it is not optimal and segments can occur that do not correspond to true segments in the image. If such segments contain many true positives the entire segment is regarded as foreground, which results in additional false positives. Better

segmentation techniques can definitely improve the FP rate of eSMM.

Note that for the ROC graph in Figure 2.9(b) SMM and eSMM are each centred around the same values. The reason for this is that the Highway sequence shows a highway with many fast moving cars that pass by. The fast moving objects are never learned into the background, so changing the learning rate has not much influence on the detection results. The vehicles in the sequence have similar color distributions as the road they drive on. Moreover, the road reflects on the metal surface of the cars. MGM and Shan2006 are able to detect these parts but are very sensitive to noise in the image. This explains the higher number of false positives compared to SMM and eSMM. In the latter we notice that the segmentation is not optimal due to the resemblance of the objects to the background. As a result, eSMM is not able to close the holes in the cars, hence the different behaviour compared to MGM and Shan2006. Although eSMM has more false negatives in this sequence, we are still able to detect large parts of each vehicle. So, the bounding boxes of our detection and those of MGM and Shan2006 are very similar.

Figure 2.10(a) shows that, for high learning speeds, the number of false positives for Shan2006 increases again (visible in the bottom left corner of the graph). Within this technique the learning speed is divided by the weight of a model to obtain the update speed. As a result, a high learning speed, combined with low weights of the models, creates a very high update speed. This makes the models very sensitive to noise, which results in additional false positives.

2.6.2 Subjective Examples

A visual comparison on the analysis of indoor and outdoor sequences is given in Figure 2.12 and Figure 2.13, respectively. The figures show, from top to bottom, the current frame, the ground truth, the output of MGM, the output of Shan2006, and the output of eSMM on representative images of different sequences. Both indoor and outdoor environments are shown, which contain slow and fast moving objects of different sizes. As can be seen, eSMM deals better with several problems like lighting changes, shadows, reflections, and initial ghosts. This matches the objective results of the previous section in the sense that, when using eSMM, there are less false positives, while most of the objects are still detected.

The first column of Figure 2.12 shows a frame of the PetsD2TeC2 sequence, during a gradual illumination change. Shan2006 is less sensitive to this effect, however, it misses some of the moving objects entirely. Our proposed system

Table 2.1: Average execution times for MGM, SMM, segmentation and eSMM in milliseconds per frame for several sequences (resolutions are given for each sequence).

sequence	MGM		SMM		seg		eSMM	
	avg	stdv	avg	stdv	avg	stdv	avg	stdv
Pets (384x288)	120	18	34	6	55	2	100	11
Highway (320x240)	96	10	24	2	49	7	82	3
Indoor (340x240)	105	12	22	3	33	3	65	6
Ismail (320x240)	103	14	25	3	34	2	67	3
ThirdV (720x576)	922	64	277	26	129	11	427	31

can handle the illumination change and is still capable of detecting the moving objects. Looking at the results for the Highway sequence, it is noticed that both Shan2006 and eSMM can deal with ghosts. However, both algorithms are not capable of dealing with the strong shadows inherent to this sequence. Similarly, for indoor sequences, eSMM succeeds in reducing the FPs while providing a good segmentation of the moving objects.

2.6.3 Execution Times

Table 2.1 shows a comparison of the execution times for the different test sequences. For each sequence we recorded for each frame the time it takes to perform the object detection and consequently present the average values and standard deviations. For the implementation of SMM and eSMM we used the same code base as for MGM. This way we can make a fair comparison in execution times. The execution speeds of Shan2006 are not shown in the table since that system is also based on MGM, but introduces histogram calculations for each frame and applies color space conversions for each pixel, which makes it slower. All measurements were done on an Intel dual Core 2.13GHz processor with 2GB RAM. As shown, SMM is much faster than MGM. The gain in speed here is mostly due to the interpolation mask, the use of a constant ρ, which prevents the costly computation of the probability distribution function, and the smart sorting of the models. Although eSMM introduces improvements of the detection results, it also slows down the system. Nevertheless, eSMM is still faster than MGM. The processing time for segmenting the image according to the edges is also shown, since for each frame this step can be done entirely independent from SMM. Consequently, a parallel implementation would allow to reduce the processing time even further.

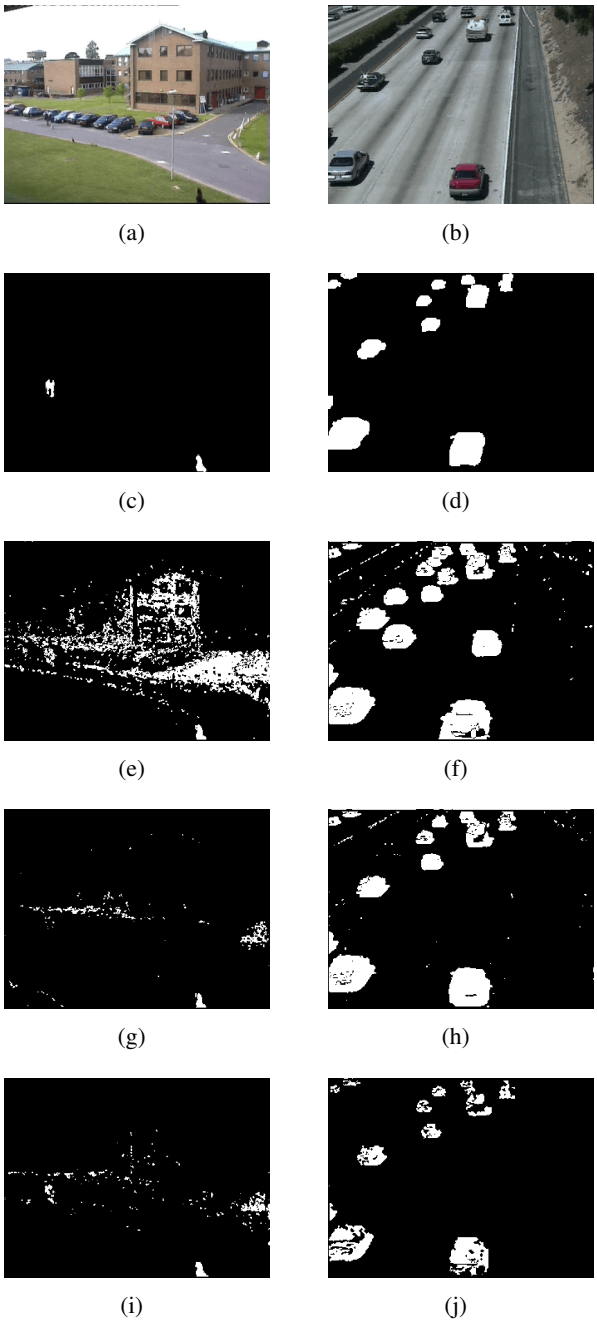


Figure 2.12: First row: current frame, second row: ground truth, third row: output by MGM, fourth row: output by Shan2006 [50], and fifth row: output by eSMM. The columns show the results for the PetsD2TeC2 sequence (frame 2300) and the Highway sequence (frame 50).

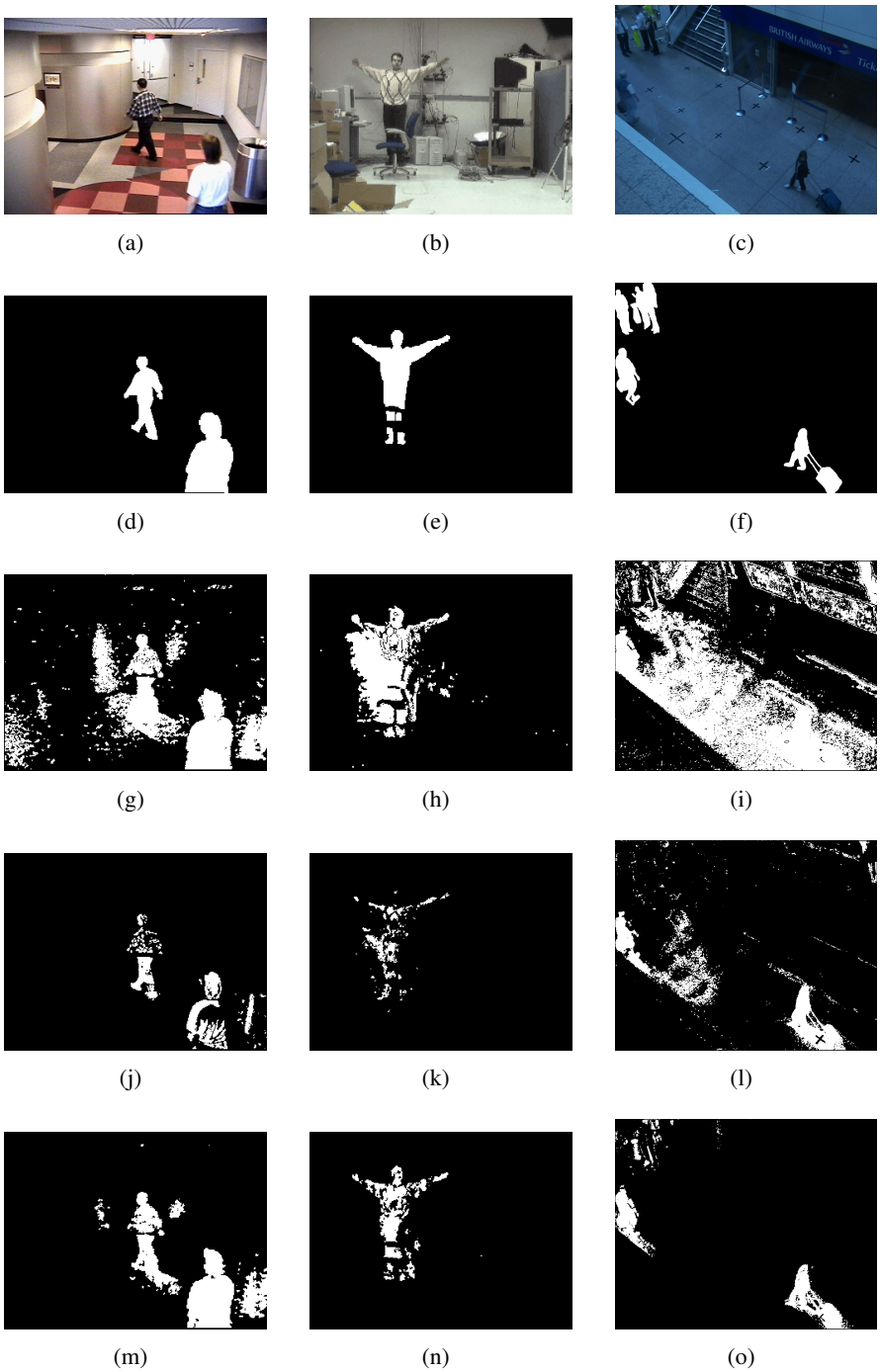


Figure 2.13: First row: current frame, second row: ground truth, third row: output by MGM, fourth row: output by Shan2006 [50], and fifth row: output by eSMM. The columns show the results for the Indoor sequence (frame 900), the Ismail sequence (frame 550), and the Thirdview sequence (frame 300).

2.7 Conclusions

In this chapter we presented a new robust background subtraction scheme based on a mixture of models that includes both temporal and spatial information. We proposed the use of simple models and altered matching, update, and decision steps. The matching step uses separate thresholds for lower and higher pixel values, instead of the Gaussian-based threshold in MGM. An intensity allowance is introduced that models photon noise by Skellam distributions. Additionally, a maximum difference with previous pixel values is used to successfully deal with gradual lighting changes. The decision step is extended with a simple and effective shadow removal scheme. The processing speed is raised by introducing a dynamic analysis mask that decides which pixels of an image are analysed. Subsequently, interpolation is used to make a decision about the pixels that were not analysed. We have shown that this mask has minor influence on the detection performance of the algorithm. Finally, to improve this temporal background subtraction technique even more, an image segmentation based on edge information is used. The segmentation is combined with the background subtraction to improve the detection results by reducing the number of false negatives.

Experimental results show that we get far less false positives than MGM, without an increase in misdetections. Additionally, a comparison is given with a more recent advanced background subtraction technique. It is shown that our system is more robust and decreases the number of false positives and negatives. Finally, a comparison of execution times shows that we can process frames up to 50% faster.

Our contributions regarding this algorithm to detect moving objects in the pixel domain can also be found in the following publications.

1. Chris Poppe, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Effect of H.264/AVC Compression on Object Detection for Video Surveillance. In *Proceedings of The International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2009*, May 2009

2. Chris Poppe, Gaëtan Martens, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Dealing with Gradual Lighting Changes in Video Surveillance for Indoor Environments. In *Proceedings of 15th World Congress on Intelligent Transport Systems*, November 2008

3. Chris Poppe, Gaëtan Martens, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Robust spatio-temporal multimodal background subtrac-

tion for video surveillance. *Optical Engineering*, 47:110101, October 2008

4. Chris Poppe, Sarah De Bruyne, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Intelligent Preprocessing for Fast Moving Object Detection. In *Proceedings of SPIE Security and Defense*, volume 6978, March 2008

5. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Mixture Models Based Background Subtraction for Video Surveillance Applications. In *Lecture Notes in Computer Science: Proceedings 12th International Conference on Computer Analysis of Images and Patterns*, volume 4673, pages 28–35, August 2007

6. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Dealing with Quick Illumination Changes when using Background Mixture Models. In *Proceedings 8th Asian Conference on Computer Vision*, volume 4843, Tokyo, November 2007

7. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Improved Background Mixture Models for Video Surveillance Applications. *Lecture Notes in Computer Science, 8th Asian Conference on Computer Vision(ACCV 2007)*, 4843:251–260, 2007

Chapter 3

Compressed Domain Object Detection

The eye is in the sky.
– Philip K. Dick (1957)

3.1 Introduction

Video surveillance systems will only rarely consist of a single camera. Such multi-camera surveillance systems come with a huge cost in bandwidth and storage. When capturing color images in RGB color space, at CIF resolution (352x288 pixels) and 25 frames per second, one needs a data throughput of 58 Mbps. In the context of video surveillance systems, the cameras need to run 24 hours a day for 7 days a week, as such creating a huge amount of data. Therefore, a practical video surveillance scenario includes video compression to reduce the used bandwidth and storage.

When looking at surveillance systems today, a common approach is to use cameras that compress the captured images with Motion JPEG. Motion JPEG uses intra-frame coding technology that is very similar in technology to the I-frame part of video coding standards such as MPEG-1 and MPEG-2. Using only intra-frame coding technology makes the degree of compression capability independent of the amount of motion in the scene, since temporal prediction is not being used. However, although the bitrate of Motion JPEG is substantially better than uncompressed video, in most cases it is substantially worse than that of video codecs that employ inter-frame motion compensation.

Consequently, many cameras provide the option to use more advanced video compression techniques. Nowadays, it is hard to find a surveillance camera that does not provide support for MPEG-based video codecs like MPEG-4. With H.264/AVC [10] a new video compression standard was introduced that outperforms MPEG-4 (and other codecs) [69]. Since this new codec outperforms other codecs in coding efficiency, it is no surprise that the first H.264/AVC cameras have already hit the surveillance market and it is assumed that more and more video surveillance data will be encoded in this new format [70]. We refer to Appendix A for an elaboration on concepts of H.264/AVC that are important in the context of this chapter.

If video compression is used and the analysis is not performed on the camera itself, a decoding step is needed before the captured images can be used as input for algorithms such as the one described in the previous chapter. To avoid this decoding step and to reuse the work done during the encoding, several efforts are done to detect moving objects directly upon the compressed video stream.

Several algorithms have been proposed to analyse video content in the MPEG compressed domain, which have good performance [71]. These algorithms typically rely on two types of features available in the compressed video: motion vectors (MVs) and transform coefficients. The MVs are treated as a sparse and noisy motion field. Transform coefficients are often used to construct DC images or are treated as a texture feature to measure similarity within blocks. However, H.264/AVC contains new features that make previous object detection techniques in MPEG compressed domain not directly reusable. The major differences with previous coding formats (like MPEG-2) are the variable block size motion compensation, which means a macroblock can now be partitioned into several smaller blocks, where each block has its own MVs. Furthermore, as spatial intra prediction is introduced, DC coefficients in intra-coded macroblocks no longer represent average energy, but only represent an energy difference.

When looking into object detection techniques that work in the H.264/AVC compressed domain, few approaches exist, mostly relying on the MV field. Due to its coding-oriented nature, the MV field is noisy, meaning the MV does not correspond to the true motion, and requires additional complexity to be processed. Hence, we present a novel alternative approach to detect moving objects that works on a higher level. When analysing an H.264/AVC compressed bitstream, the size (in bits) that a macroblock (MB) occupies, is recorded. Based on these sizes, a background model is created during a training period. New images are consequently compared with this model to yield

MBs that correspond to moving objects. Subsequently, these MBs are spatially and temporally filtered to remove noise. Finally, the sizes of the sixteen 4x4 blocks within a boundary MB are evaluated to make a more fine-grained segmentation.

The next section presents related work on moving object detection in the MPEG compressed domain, with a strong focus on techniques that work in the H.264/AVC compressed domain. Subsequently, we elaborate on the context of this research, showing some assumptions made about the H.264/AVC coding. Next, a number of observations are presented that were made during the analysis of H.264/AVC compressed sequences. Consequently, we propose a system, based on these observations, to find moving objects in the H.264/AVC compressed domain. Experimental results of our system are shown in Section 3.5 and concluding remarks are made in Section 3.6.

3.2 Related Work

Several techniques exist that deal with moving object detection in the MPEG-2 compressed domain. Zen et al. used the MV magnitudes to determine whether a block corresponds to a moving object [72]. According to the MV angle similarity, the blocks are spatially merged to reduce the effect of noisy MVs. Jamrozik and Hayes used a levelled watershed technique on a MV field that was accumulated over time [73]. Long et al. created a MV field by accumulation over time and used a median filter to clean the field [74]. Additionally, they created a feature vector consisting of the DCT coefficients to refine the MV field even further.

Most techniques that work in the MPEG-2 compressed domain are based on the MV field, hence it is no surprise that this approach is adopted in the literature by H.264/AVC compressed domain techniques. Thilak and Cruesere presented a system to track targets in H.264/AVC video [75]. They used the MV magnitudes to detect objects of interest. As a consequence, MVs that point in different directions are still considered as belonging to the same object. Moreover, their system relies on prior knowledge of the size of the target.

Zeng et al. classified MVs in edge, foreground, background and noise MVs, to create a moving object detection system in the H.264/AVC compressed domain [76]. In a post-processing step, the classified MV field is then submitted to Markovian labelling, to yield 4×4 blocks that correspond to moving objects. This labelling uses spatial and temporal information, and the block size to decide whether a MV corresponds to a moving object. Finally, backtrack-

ing is used to make a decision for the I pictures. They presented good results, however the system utilizes several parameters for the thresholds in the classification and the weights used in the labelling, and these have to be fine-tuned for different sequences.

This system has been reused by Yang et al. to perform moving object segmentation [77]. The labelling result is linked with an image consisting of DC coefficients that is obtained from partly decoded I pictures. Finally, an extra decoding step is performed on the regions around the edges of the found objects and edge information in the pixel domain is extracted for refinement. The performance (both in execution speed and segmentation accuracy), is very dependent on the employed thresholds and the size of the objects that are present in the scene.

Liu et al. created a normalized and median filtered MV field to perform moving object segmentation [78]. Subsequently, a complex binary partition tree filtering is used to segment the MV field. The complexity of the partition tree increases drastically with a noisy MV field.

As was the case in the MPEG-2 compressed domain analysis techniques, most techniques that work on the H.264/AVC compressed domain, are based on the MV field. However, as MVs are created from a coding point of view, they generally are created to optimally compress the video, not to optimally represent the real motion in the sequence. Consequently, MV fields can be very noisy and it is difficult to find real moving objects based solely on this field. Hence, we present an alternative technique that solely relies on the number of bits that a MB uses, in an H.264/AVC compressed bitstream, to perform moving object detection.

3.3 Context

The H.264/AVC video streams are generated using the Joint Model reference software (version JM 12.4), the encoder settings can be found in Appendix B. This work is restrained to the Baseline Profile of H.264/AVC, which is suitable for video surveillance applications thanks to the low coding complexity. As such, only I and P pictures can be used, each partition in a MB has at most one MV, and CAVLC entropy coding is used. This restriction is also made in related work and is most likely to be supported by surveillance cameras in the near future. In the rest of the chapter, we assume that a picture consists of only one slice, containing the entire picture. Hence we will use the terms I and P pictures. Additionally, we assume that the video sequences have an Intra

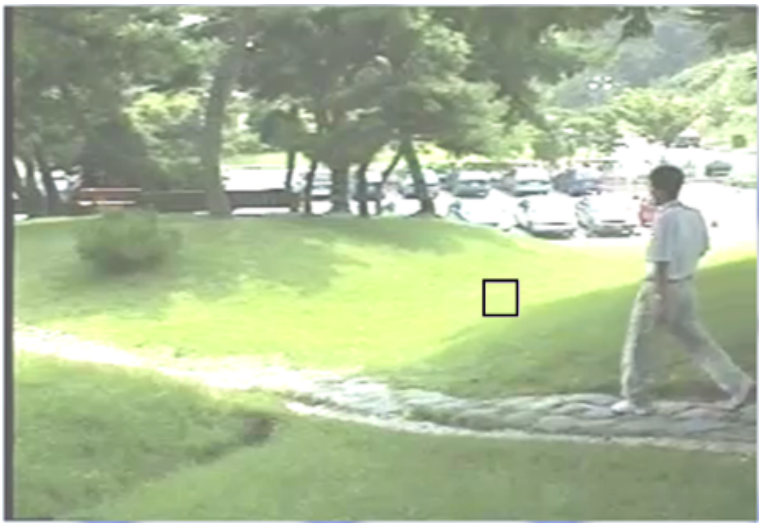
period of 16, meaning every 16th picture is an I picture, while the rest are P pictures.

The goal is to find moving objects, or foreground objects, with a fixed camera, so the actual background is assumed to be more or less static and visible over several different frames. Even when using a moving camera (e.g., pan-tilt-zoom camera), techniques exist to compensate for the movement and to create a stabilized image sequence [4, 5]. Note that the assumption of a static camera does not imply that the background is static. Clutter, moving bushes and trees, noise, etc. can make a highly dynamic background.

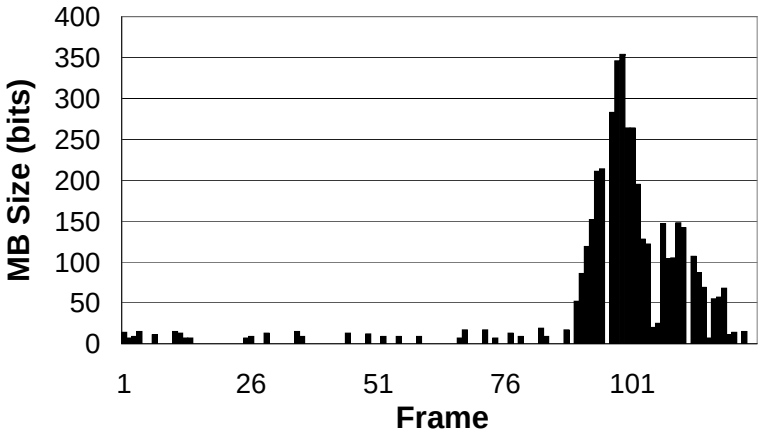
We observed that, during encoding, parts of the background can be predicted very accurately by intra or inter prediction. This results in high compression ratios for these parts, and consequently the encompassing MBs use a low amount of bits within the encoded bitstream, we will refer to the latter as *size of a MB*. This is visualized in Figure 3.1 and Figure 3.2, which show the data size (in bits) of two different MBs over several consecutive P pictures of the *Etri_od_A* sequence. Figure 3.1(a) shows a MB that contains homogenous background values. As such, this MB can achieve high compression. In contrast, Figure 3.2(a) shows a MB that contains a piece of the background with much detail so it is harder to compress and results in larger bit sizes. A second observation is that the blocks corresponding with moving objects are harder to predict. Figure 3.1(b) and 3.2(b) show the influence of a moving object upon the size of these MBs. A person walks through the scene during frame 92 to 120 for Figure 3.1(b) and frame 71 to 86 for Figure 3.2(b). When parts of the person occupy the analysed MBs, we see in both graphs a sudden raise in size. Typically, two peaks appear, corresponding to the edges of the moving object. The MBs that contain an edge of a moving object are more difficult to compress since it is hard to find a good match for this MB. In between the peaks, a lower MB size is seen, due to the internal region of the object that passes by the MB. This internal part can be better compressed since it does not change much over consecutive frames.

If parts of the background are dynamic (e.g., waving trees), they can still be predicted good if the motion is repetitive. If a foreground object appears, typically these will be predicted with less accuracy, resulting in larger amounts of bits. For high bitrates we also see that the MBs corresponding to edges of these moving objects will have small sub partitions.

A general conclusion is that MBs corresponding to (the edges of) moving objects will typically contain more bits in the bitstream than those representing background.



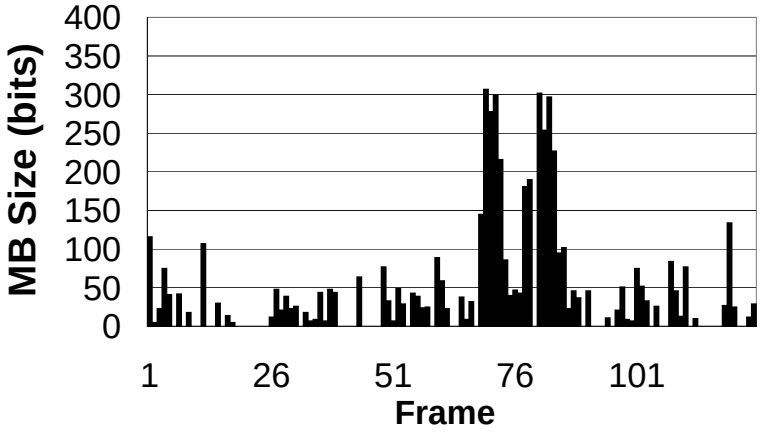
(a)



(b)

Figure 3.1: Influence of a passing object (frame 92 to 120) on the size of MB 190 in P pictures of the Etri_od_A sequence. The MB contains a homogenous part of the background.

(a)



(b)

Figure 3.2: Influence of a passing object (frame 71 to 86) on the size of MB 151 in P pictures of the Etri_od_A sequence. The MB contains a part of the background with much detail.

3.4 Proposed Algorithm

Our algorithm is based on the observations above and tries to detect the blocks that correspond to moving objects. P pictures are analysed using a two-step approach. In the first step, a background model is learned from the scene and subsequently used to find MBs that correspond with moving objects. This is explained in the next section. The second step consists of refining the found 16×16 MBs to the 4×4 subMB level by analysing the size of the transform coefficients within boundary MBs. This is discussed in Section 3.4.2. Finally, a detection for I pictures is generated based on the found objects within the surrounding P pictures, which will be discussed in Section 3.4.3. Figure 3.3 depicts the proposed algorithm on the MB level (for analysing P pictures) and will be referenced in the next sections.

3.4.1 MB Level

Our approach is a background subtraction technique, in which new images are compared to a background model to yield foreground regions.

From the observations of the previous section, it is clear that by using the size of a MB over consecutive frames, a distinction can be made between MBs representing foreground or background. Additionally, it is shown that different MBs have different behaviours, so a background model is used that consists of different values for each MB. During a training phase, in which no moving objects are present, a background model is constructed by recording, for each MB, the maximum size it reaches within the bitstream over several consecutive frames (denoted as $MB_{model,i}$ in Figure 3.3, with i the number of the MB). The number of frames that are used to create the background model forms a parameter of our system. Although it might be hard to obtain training images without moving objects, this is a common technique even in pixel domain object detection [79]. Furthermore, in environments that consistently show moving objects, the background model could be created by comparing the MB sizes within one frame to each other. The lowest ones are then assumed to represent background. Eventually, enough data can be accumulated over time to create a reliable background model that covers the entire image. However, this learning phase takes longer and is more error-prone.

After the initial training phase we compare, for each new frame, the size of the current MB (denoted as MB_i) with the corresponding value in the background model. If the difference between the size of the current MB and the value

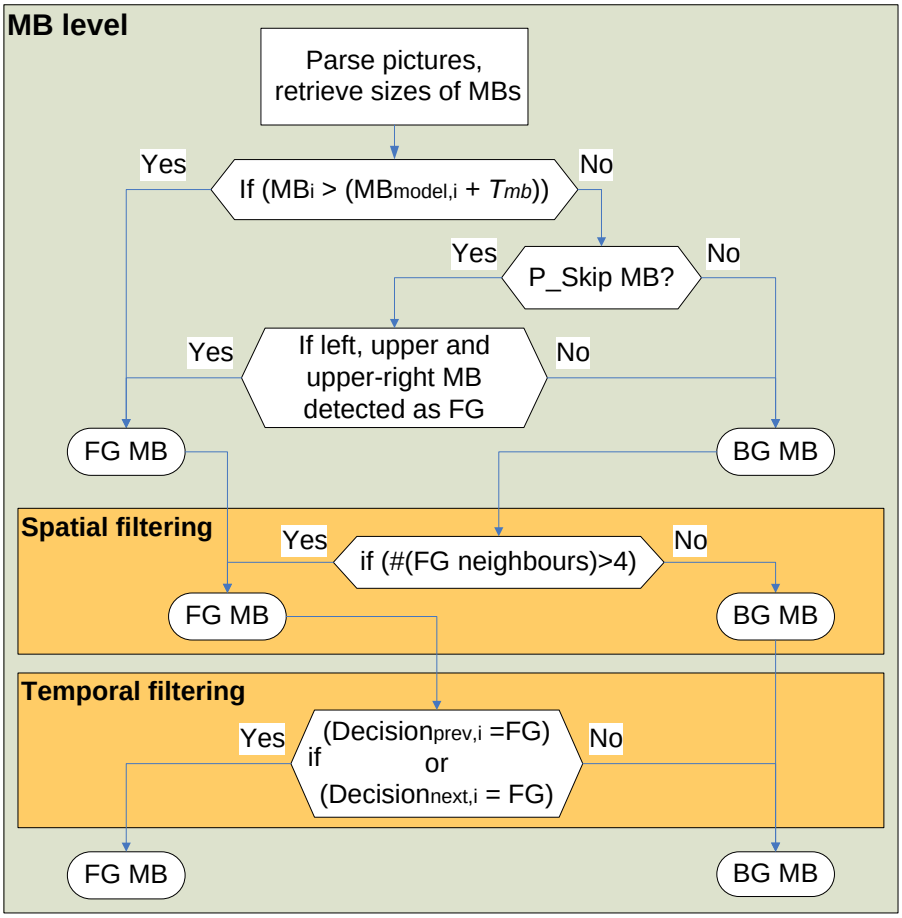


Figure 3.3: Flowchart of the proposed algorithm working on P pictures on the MB level.

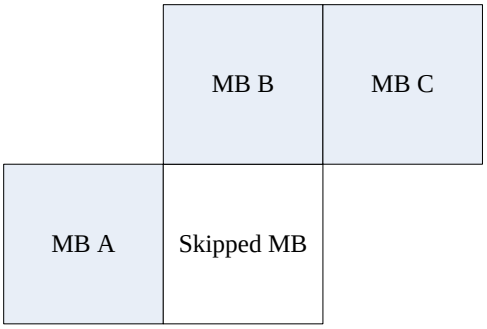


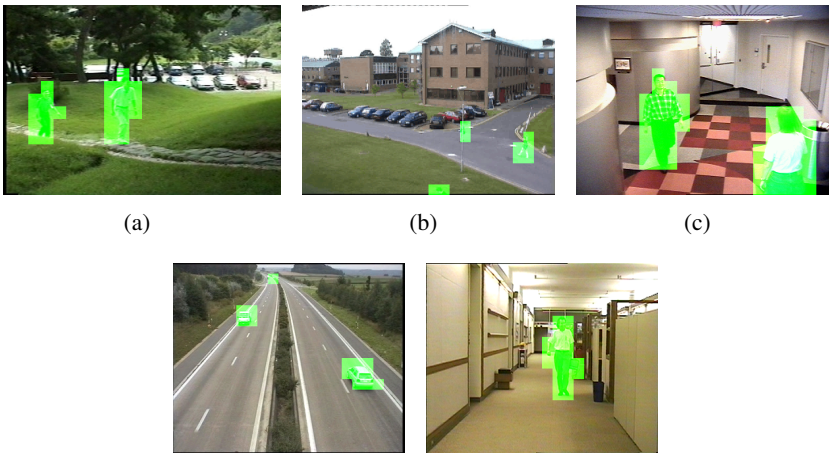
Figure 3.4: P_Skip MB and the surrounding MBs that are used to predict a MV.

of the corresponding MB in the model, is larger than a threshold T_{mb}, it is considered a foreground MB (FG MB). High values of T_{mb} result in detection misses, lower values result in many false positives. Hence, this threshold is dependent on the video sequence. The influence of this threshold is shown in the experimental results in Section 3.5.

The combination of the background model and threshold T_{mb} allows to detect those MBs with an unusually large size and thus have high probability to correspond with a moving object. However, as shown in Figure 3.2(b), the MBs corresponding with internal regions of a moving object can have lower sizes. To detect these, additional processing is needed.

If a moving object is large, P_Skip MBs tend to occur within the object, creating holes in the detection. In case of a P_Skip MB, the decoder calculates a MV based on the surrounding MBs (MB A, B and C in Figure 3.4). The lack of residual data of such MBs makes it impossible to compare them with the generated background model. However, during the decoding process a MV is inferred for these P_Skip MBs based on other MBs, so we can assume that the P_Skip MB has the same behaviour as the surrounding ones. Therefore, if all these MBs were denoted as FG MB in the previous step, the P_Skip MB is also regarded as a FG MB, otherwise the MB is classified as a background MB (BG MB).

The BG MBs form the input of a spatial filtering phase, as shown in Figure 3.3, to deal with other MBs (not P_Skip MBs) that lie within a moving object. The spatial filtering consists of a median filter that is applied on an 8-connected neighbourhood of MBs that are detected as background. So if more than 4 of the neighbouring MBs are foreground, the MB is also considered to be a



(a) (b) (c)

(d) (e)

Figure 3.5: Output of proposed for (a) Etri_od_A sequence (frame 680), (b) PetsD2TeC2 (frame 700), (c) Indoor (frame 1530), (d) Speedway2 (frame 1705), and (e) Hallmonitor (frame 255).

FG MB. This spatial filtering is iteratively repeated until no more changes occur. As such, we can deal with MBs that have low residual data because they lie within an object and generally can be predicted well during the encoding process.

Using the background model and the spatial filtering allows to detect the moving objects, but noise in the video that causes the size of a MB to temporally rise, will also be regarded as foreground. Consequently, temporal information is needed to deal with these situations. Figure 3.1(b) also shows the temporal consistency of an object that passes by, the size of the MB is increased over several consecutive frames. Since the MBs occupy regions of 16 by 16 pixels, an object that passes by a MB will mostly occupy that region over several consecutive frames, so the MB will consistently be detected as foreground. Based on this observation, a temporal filter is used to remove noisy MBs that only appear shortly. This filter is kept very simple for fast processing; if a FG MB is not detected as foreground in the previous or next frame it is rejected and treated as BG MB (in the figure the decision that was made for that MB in the previous and next frame is denoted as $D_{prev,i}$ and $D_{next,i}$, respectively). Note that this temporal filter introduces a delay of one frame.

The system presented above allows to detect moving objects up to the 16×16

MB level, each MB of a frame is classified as foreground or background. Figure 3.5 shows typical detection results of the system on different sequences containing a variety of objects. From these figures it is clear that the detected FG MBs have a high probability to be correct, but the 16×16 detection is rather coarse to accurately detect the arbitrarily shaped objects.

To elaborate on this, we present precision and recall values for the MB level approach on the PetsD2TeC2 and Indoor sequence in Figure 3.6(a) and Figure 3.6(b), respectively. These sequences are generally used test sequences, which contain slow and fast moving objects of small and large sizes [80]. For both sequences we compare the output of our proposed system against two different ground-truth annotations, resulting in two graphs. The first, GT_pix, represents the comparison with a pixel-based ground truth that was manually made for every 50th frame of the sequence. The second graph, denoted as GT_MB, uses a MB-based ground-truth annotation. This MB-based ground truth is automatically generated from the original pixel-based ground truth. In this case if one pixel of a MB is considered foreground in the pixel-based ground truth, all the 16×16 pixels of that MB are denoted as foreground in the MB-based ground truth.

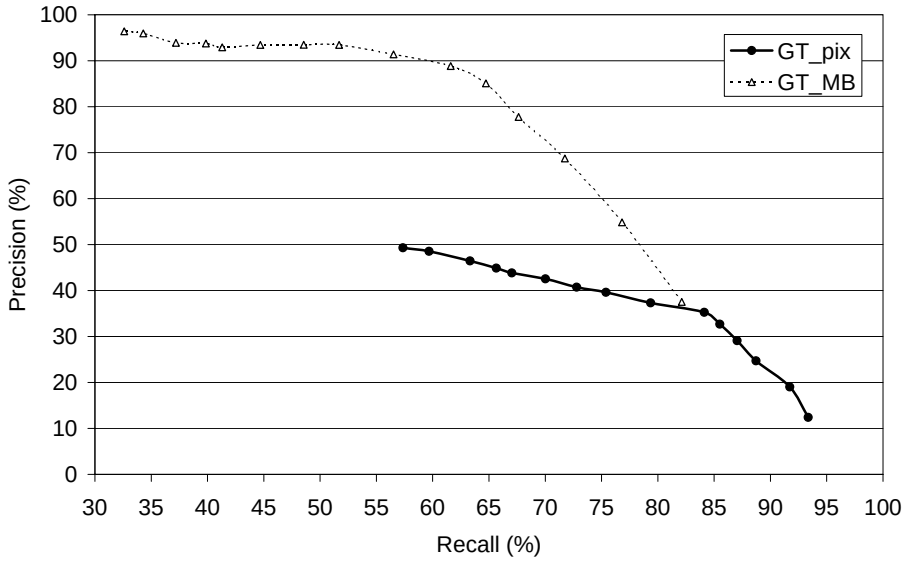
The actual graphs are constructed as follows. The same definitions as in the previous chapter hold; a false positive (FP) and false negative (FN) denote a misclassified background pixel and foreground pixel, respectively. Accordingly, a real foreground pixel or background pixel that is correctly classified is called a true positive or true negative, respectively. The X-axis shows the recall, which defines how many positive samples have been detected among all positive samples available during the test. In this case, it represents the ratio of pixels that were correctly considered as foreground, to all the real foreground pixels:

$$recall = TPR = \frac{TruePositives}{Realpositives}. \tag{3.1}$$

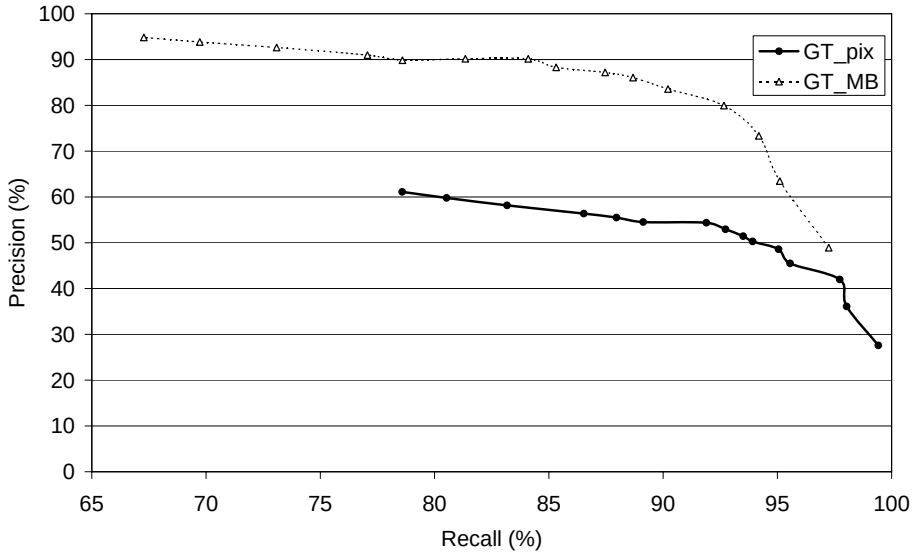
The precision, shown on the Y-axis, denotes the percentage of the pixels that are classified as foreground, which actually are real foreground pixels:

$$precision = \frac{TruePositives}{TruePositives + FalsePositives}. \tag{3.2}$$

The output of the algorithm is compared to the ground-truth annotations to calculate the precision and recall values. Subsequently, these values are summed for the entire sequence to create one point of a graph. Finally, an entire graph is constructed by varying the threshold T_{mb} (0 to 140), and plotting the average precision and recall value of the entire sequence for each threshold value.



(a)



(b)

Figure 3.6: Results of the proposed MB-level algorithm using a pixel-based and MB-based ground-truth for (a) the PetsD2TeC2 sequence and (b) the Indoor sequence.

Good systems obtain high precision and recall values. Low values for T_{mb} tend to result in higher recall and lower precision values, while higher values are situated more to the left of the graph. Indeed, if T_{mb} is set low, many MBs are detected as foreground, even if they only differ slightly from the BG model. Hence, noise creates many false positives, which decreases the precision.

The graphs show that, when using a pixel-based ground truth, the precision is rather low, meaning that many of the pixels that are detected as foreground by the system actually are background. However, the graphs also show that the precision of GT_MB is much higher than that of GT_pix, meaning that the MBs that are detected have high probability to be correct, but the coarse MB-based detection causes many pixels to be misdetected. Indeed, within the MBs that are located at the edges of the objects, there are many pixels that do not correspond to real foreground, which was also noticeable in the visual results shown in Figure 3.5. Therefore, an extension of the algorithm to the sub-macroblock (subMB) level for these boundary MBs promises to increase the detection performance. This subMB analysis is the topic of the next section.

3.4.2 SubMB Level

Figure 3.7 depicts the subMB level of the proposed algorithm. The subMB level takes the decisions of the MB level as input and tries to make a decision for the 4×4 blocks (denoted as $subMB_j$ in Figure 3.7) within the MBs. All the 4×4 blocks corresponding to the 16×16 BG MBs are regarded as background (denoted as BG subMB). Since the MBs that are detected as foreground on the MB level have high probability to be correct, we restrict ourselves to these FG MBs for further investigation. Moreover, as can be seen from Figure 3.5 the coarseness of the detection especially causes problems within the MBs situated at object boundaries. Therefore, the first step is to check if a FG MB is such a boundary MB and only use these MBs during the subMB level analysis. In this sense, a FG MB is considered to be a boundary MB if his left, right, upper, or lower neighbour MB was detected as background during the MB level analysis. For all other FG MBs, the decision of representing FG is applied to all the containing 4×4 blocks (resulting in FG subMBs).

Within a MB the transformation of the residual data occurs on 4×4 blocks [81]. Since this is the transformed residual data, the assumption is that 4×4 blocks with a high amount of bits are those that were the hardest to compress. Therefore, the proposed algorithm is extended with the following. For each boundary MB, again by simple parsing, the sizes in bits that these transform coefficients use within the bitstream are gathered. These sizes are compared with

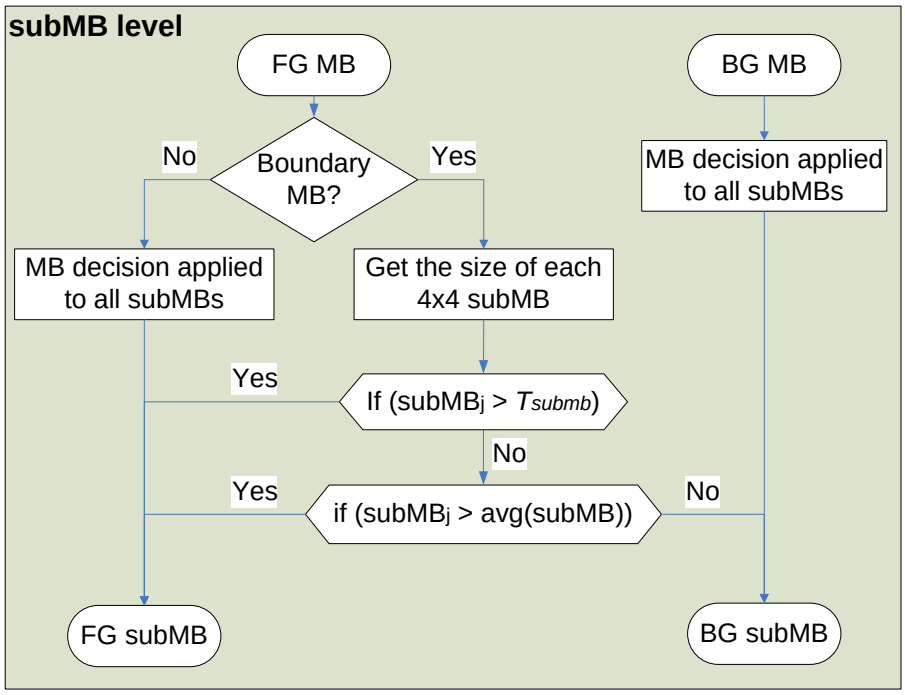


Figure 3.7: Flowchart of the proposed algorithm working on P pictures on the subMB level.

the average size of the 4×4 blocks within the current MB. The 4×4 blocks that have a lower amount of bits than the average are considered to be background, those with a higher amount are considered to be foreground.

However, this check can be too strict. If a MB is totally covered by a moving object, the proposed step will still consider the smallest 4×4 blocks within that MB as background, even though it can have a large size. Hence, a new threshold, T_{submb}, is introduced on subMB level. The resulting algorithm consists of the following (see Figure 3.7), if a 4×4 block is larger than this threshold it is immediately regarded as a FG subMB. This way we can prevent that 4×4 blocks of large sizes are falsely regarded as background. If the 4×4 block is smaller, it is compared with the average size of the 4×4 blocks in the current MB. Note that, according to our experiments, using the median instead of the average does not influence the results much.

If T_{submb} is set to 0, all the 4×4 blocks within the boundary MBs are considered as foreground. In that case, the algorithm detects objects up to the MB level as discussed above. When applying higher values for T_{submb}, more 4×4 blocks will be compared to the average, so a more fine-grained detection is possible. This threshold should be adjusted according to the noise (e.g., photon noise) in the sequence. Sequences with much noise require higher values for T_{submb}. In all experiments we set T_{submb} to 10, a value that was experimentally determined on test sequences.

3.4.3 I pictures

Until now, only P pictures were considered in the system. However, next to P pictures, an H.264/AVC bitstream also contains I pictures, for which all included MBs are intra predicted. For these MBs the temporal redundancy in the video can not be exploited. Therefore, sizes of MBs within an I picture are typically much larger than those within P pictures and these MBs cannot be evaluated in the same manner as the algorithm did for the P pictures. Hence, the incorporation of I pictures would result in a background model with high values for each MB, which makes it not usable in our approach. Consequently, we present an alternative solution to process the I pictures.

Typically, I pictures are sparse compared to the P pictures in a compressed video sequence. Therefore, to allow fast processing of an I picture, a simple interpolation of the detection results of the previous and following P pictures is used. The interpolation consists of a binary and-operation, so if a MB at a certain position in the picture was detected as foreground in the previous and

following picture, it is denoted as foreground in the current I picture.

Note that intra predicted MBs can occur in P pictures too. The difference is that in that case, the encoder chooses to use intra prediction for a MB if a gain in compression can be achieved. As such, these can still be evaluated as above. In contrast, in I pictures, the intra predicted MBs are the only type of MBs that can be used. As such, these MBs typically result in lower compression.

3.5 Experimental Results

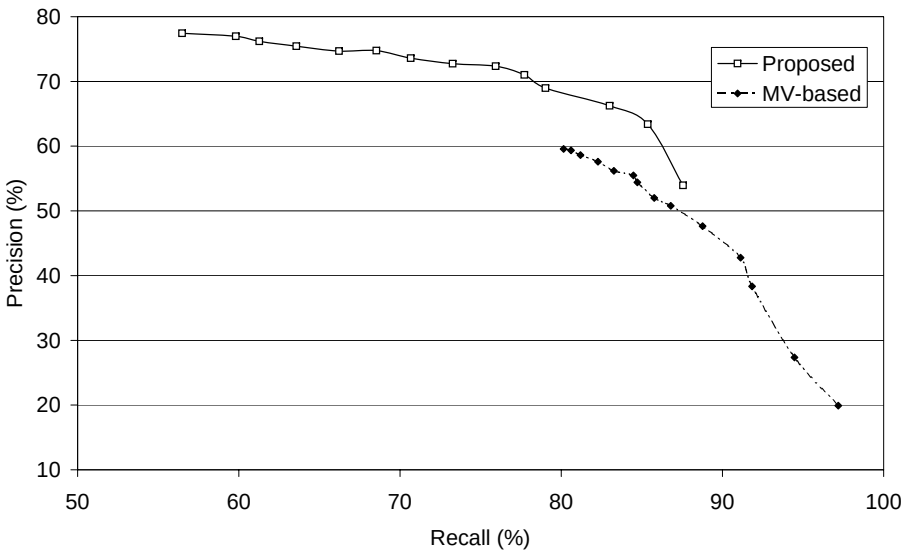
The algorithm explained above allows to detect moving objects up to the 4×4 level. In this section we present an exhaustive evaluation of the proposed system. Next, we compare the detection performance with the related work and give a speed and visual comparison. Finally, we show how different encoder configurations influence our algorithm. Although the syntax of an H.264/AVC bitstream is standardized, the configuration of the encoder can result in completely different bitstreams. As such, we believe this is an important point of evaluation, which is missing in most of the related work.

Within this section we have chosen to use precision-recall graphs over the ROC-graphs (used in the previous chapter). The use of precision and recall to evaluate the system is a common approach in the related work within the compressed domain analysis. Whereas ROC-graphs are more common in techniques that work in the pixel-domain. Appendix C discusses and compares these two evaluation methods to show that there is only minor difference. We believe that the publication of precision and recall values of our system will encourage other authors to compare their results with ours.

3.5.1 Objective Comparison

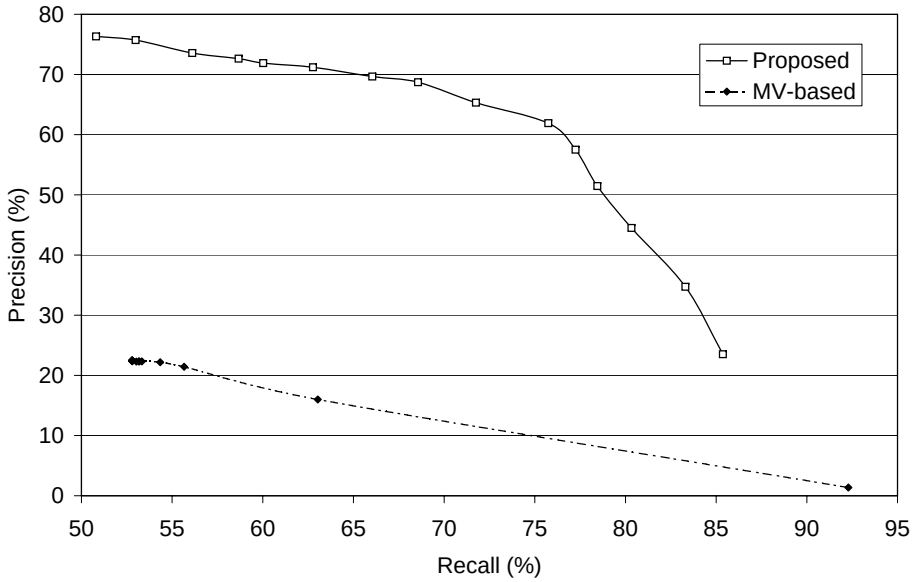
In this section we present a comparison of our proposed algorithm with the work of Zeng et al., a MV-based approach (presented in Section 3.2). Since they are relying on MVs they can also detect moving objects up to the 4×4 level. As such, to give a fair comparison, we will compare the output of both systems with a subMB-based ground-truth. In this sense, to create the ground-truth, a 4×4 block of pixels is regarded as foreground if one of the pixels is considered foreground in the pixel-based ground-truth. The comparison with such a subMB-based ground-truth is commonly used in the related work.

Figure 3.8 and Figure 3.9 show the Precision-Recall graphs for the related

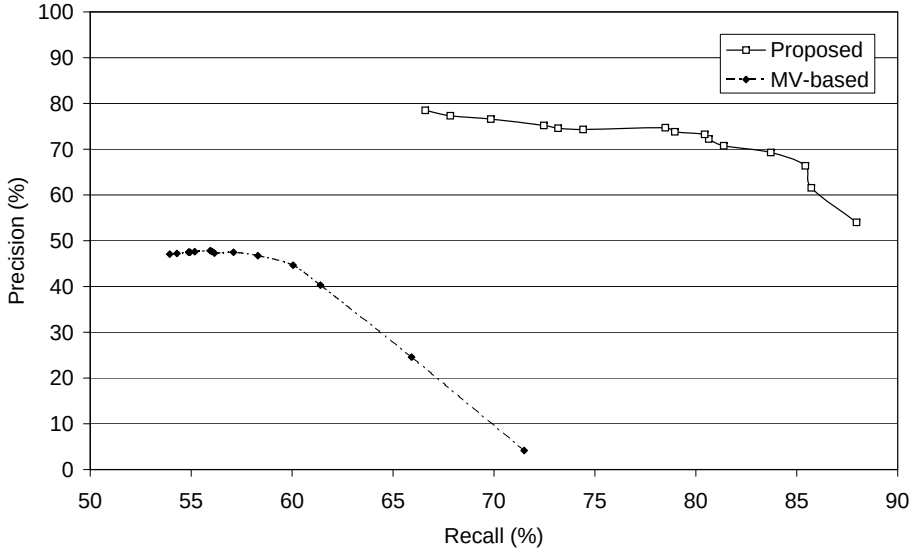


(a)

Figure 3.8: Precision-Recall graph of MV-based and proposed for the *Etri_od_A* sequence.



(a)



(b)

Figure 3.9: Precision-Recall graph of MV-based and proposed for (a) the PetsD2TeC2 sequence, and (b) the Indoor sequence.

Table 3.1: Parameter settings for the MV-based approach for different test sequences.

sequence	Th_b	Th_f	α	β	γ	η
Etri_od_A	1	3	4	5	7	6
PetsD2TeC2	1	2	3	3	5	6
Indoor	2	8	4	3	5	2

and proposed approach on different sequences. To create these graphs for the MV-based approach we vary one of the thresholds (Th_e, a threshold used to find edge blocks), while the others are set to the optimal values. These values are shown in Table 3.1 and for an in-depth explanation of the thresholds we refer to [76]. To find these values we have experimentally determined the optimal threshold settings for each sequence, since different sequences require a different configuration. Note that, for our proposed algorithm, T_{MB} is again varied (0 to 140) and T_{submb} is set to a fixed value of 10 for all sequences.

Figure 3.8 shows the detection results on the *Etri_od_A* sequence. As can be seen, our proposed system has higher precision values, where the related work has higher recall values. The high precision of our system is due to the two-step approach; the analysis on the MB level succeeds in removing noisy MBs, the found FG MBs are then refined in the subMB step. Note that Zeng et al. published slightly different precision and recall values for this sequence in their paper (they obtain an average precision of 71.3 % for a recall of 87.2%). However, to create those values they used a ground truth that was based on the block structure that was present within the specific encoded sequence. As such, different encoder configurations yield different versions of the ground truth. In contrast, our ground truth is fixed and created based on the pixel-based ground truth, so it can be used for different encoders and configurations. Moreover, Zeng et al. used only 105 frames of the *Etri_od_A* sequence, whereas we make an evaluation of the entire sequence. Lastly, no information has been given about the specific settings of the encoder or the parameters of their system. This explains the difference with their presented results.

Figure 3.9 holds two additional sequences, which are evaluated to see how both algorithms perform on more challenging examples of real video surveillance scenarios, which contain noise, shadows, reflections, and changing lighting conditions [80]. As the figure shows, the MV-based approach, which directly works on a 4×4 level for the entire image, suffers from the noisy MV field in the sequences. Since MVs are created from a coding perspective, the noise, shadows and lighting changes result in several MVs that do not correspond to

real moving objects, but that are wrongly classified by the MV-based approach. Using stricter parameter settings to filter out the noise, results in too many false negatives, since then small or slow moving objects are not detected.

On the other hand, our approach achieves higher precision and recall values. This is especially due to the low number of false positives in our system. The initial MB level analysis succeeds in filtering out much of the noise in the data, while the subMB level analysis refines the detection. As our system only uses the residual information of the bitstream, it is not affected by the MVs that are chosen by the encoder. Although noise and shadows do increase the size of the affected MBs, it is not enough to be regarded as a foreground, hence our system is more resistant to these situations.

3.5.2 Subjective Examples

Figure 3.10 shows typical results of the MV-based approach and the proposed system where pixels detected as foreground or background are colored white or black, respectively. Each column shows the outputs for one image of the according sequence to visually show the differences of the algorithms. As can be seen, the MV-based approach is more sensitive to noise, for each image several FPs are present. Our proposed approach on MB level succeeds in finding the MBs that correspond with moving objects, however the 16×16 size is coarse (especially when detecting people). The subMB level approach is able to refine the detected MBs. Note that, in some cases, the refinement leads to additional false negatives (as shown in Figure 3.10(o)), since some parts of the boundary MBs are wrongly considered as background.

3.5.3 Execution Times

Table 3.2 shows the execution performance of the algorithms for different sequences in frames per second. All measurements were done on an Intel Core 2 Duo 2.13GHz processor with 2GB RAM. These values include the parsing of the H.264/AVC compressed bitstream and the actual analysis to detect the moving objects. As shown, our system achieves very high execution speeds, both when working on the MB level and subMB level. In contrast, the related work, based on MVs, achieves processing speeds of 30 frames per second for CIF resolution [76, 78]. In the table we have added a version of the Indoor sequence, coded with a fixed QP of 8, to show the influence this has on the execution speed. We noticed that sequences encoded with a very small QP, tend to have more MVs of different sizes. As a result, the MV-based approach

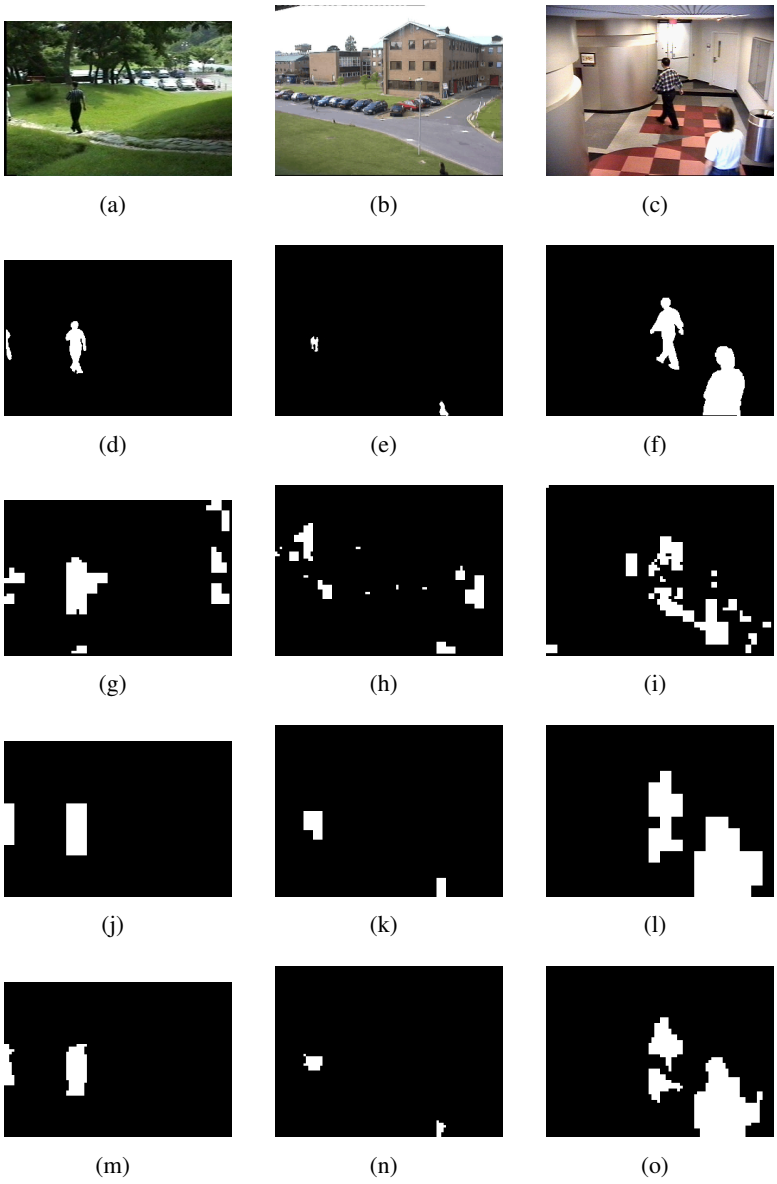


Figure 3.10: First row: current frame, second row: ground-truth, third row: output of MV-based approach, fourth row: output of proposed system on MB-level, fifth row: output of proposed approach on subMB-level. The columns show the results for the *Etri_od_A* sequence (frame 300), the PetsD2TeC2 sequence (frame 2300) and the Indoor sequence (frame 50).

needs to analyse more blocks, resulting in a raise of execution times. This behaviour is similar to the fact that many moving objects in a scene will slow down the system, which was reported in their paper. The table shows that our system is also affected by this different QP setting, however, the loss in speed is mostly due to the fact that the parsing of the bitstream takes more time. Indeed, at low QPs (or high bitrates) less Skipped MBs are used and the MBs are more partitioned, which makes the parsing slower. In the next section, we will analyse the influence of different QPs on the actual detection results of both approaches.

Table 3.2: Average execution speeds in frames per second.

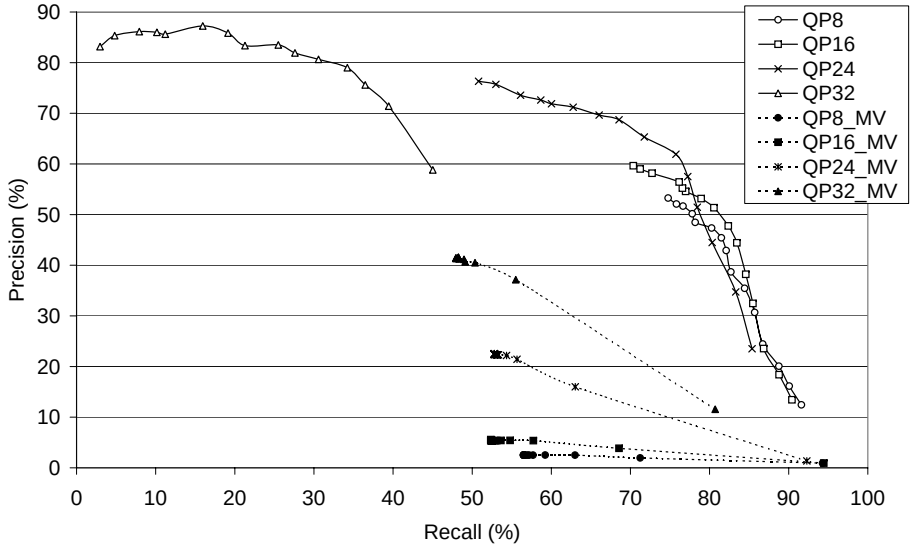
sequence	MV		proposedMB		proposedsubMB	
	avg	stdv	avg	stdv	avg	stdv
Etri_od_A (352x240)	28	0.4	662	5.8	613	6.2
PetsD2TeC2 (384x288)	22	0.4	448	7.7	403	7.2
Indoor (340x240)	31	0.6	751	11.1	648	9.8
Indoor QP8 (340x240)	6.5	0.4	425	9.5	389	8.1

3.5.4 Influence of encoder configuration

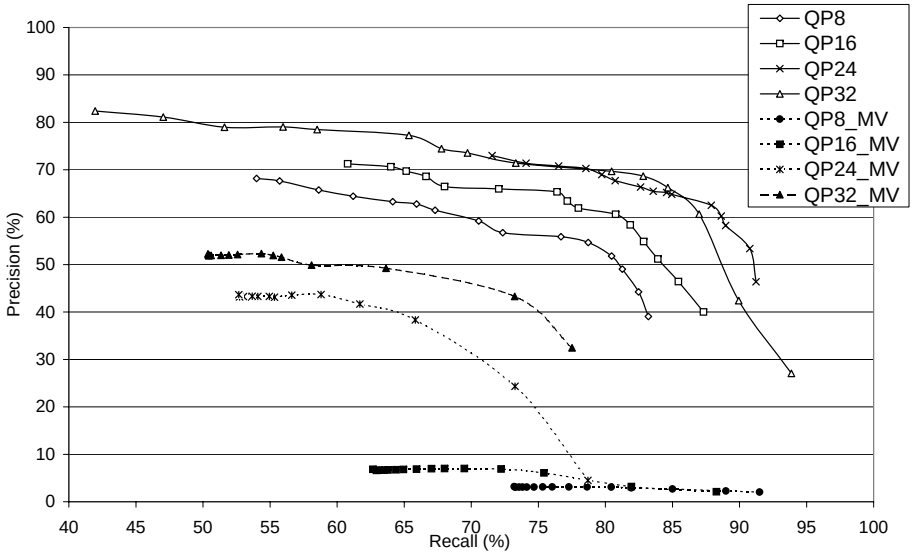
The configuration of the encoder has a large influence on the resulting bitstream. Different settings result in different decisions that are made during the encoding process. To make a detailed analysis of our system, we tested our proposed algorithm and the related work to see the influence of different QPs, bitrates, and used motion estimation methods.

The QP has a strong influence on the amount of compression that is achieved. The higher the QP, the more the data is compressed. When using a fixed QP, the visual quality of the video is more or less fixed, but the used bitrate for each frame can differ a lot. Figure 3.11 shows the precision and recall values when varying the QP for different sequences. Again, we vary the threshold T_{mb} for our algorithm and threshold Th_e for the MV-based approach. These algorithms are applied to the same sequence encoded at different QP values, resulting in different graphs ($QP8$ to $QP32$ for our algorithm, $QP8\_MV$ to $QP32\_MV$ for the MV-based approach).

High QPs result in high compression of the data, so the size of the resulting MBs decreases. As can be seen, the detection performance of the algorithm drops in that case. More specifically, the recall decreases since many MBs are wrongly considered to be background due to the low sizes. Vice versa, low



(a)



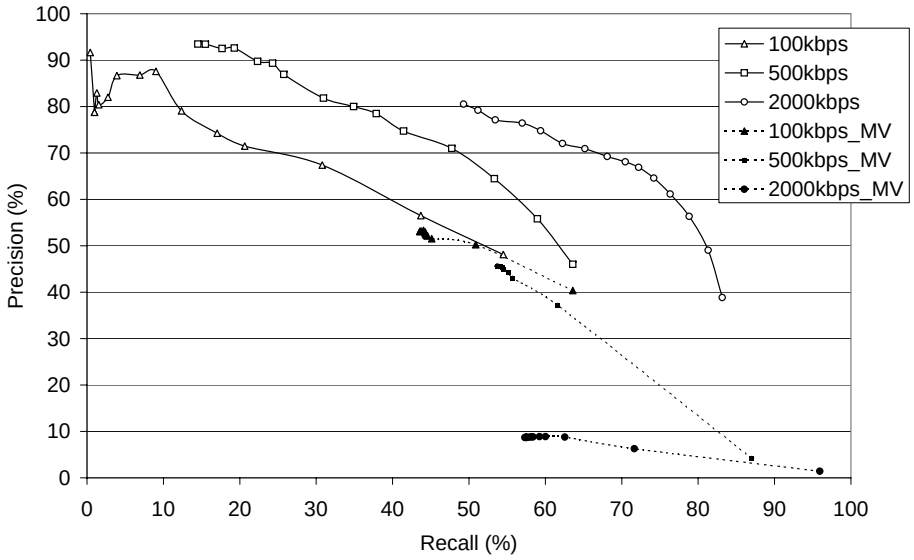
(b)

Figure 3.11: Precision-Recall graph for fixed QP versions of (a) the PetsD2TeC2 sequence and (b) the Indoor sequence.

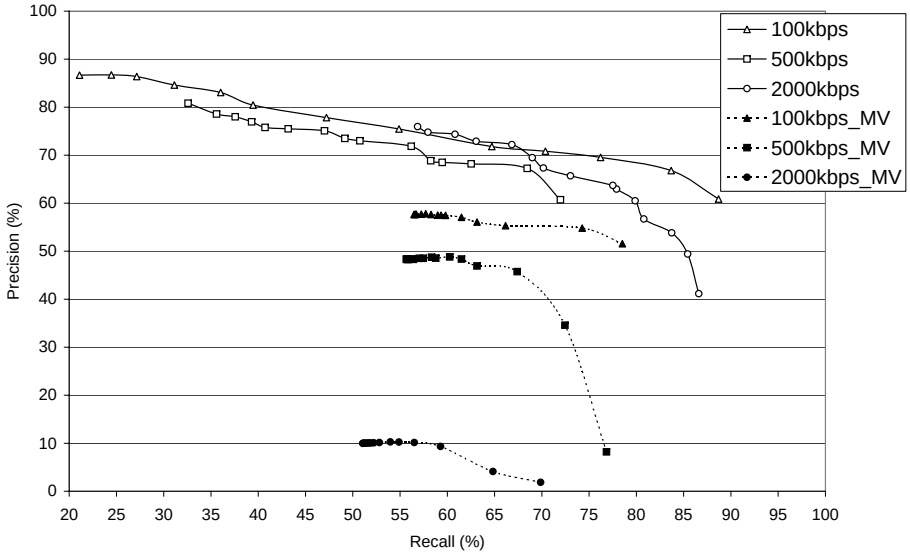
QPs lead to low compression, so MBs containing data that is harder to compress (like those corresponding with foreground objects) will have a larger size than MBs that are easy to compress (like those covering with background). The graphs show that for different QPs a different threshold yields to the best performance (high precision and high recall), hence, we could make the threshold dependent on the used QP and choose a high threshold for low QPs and vice versa.

In many cases an encoder needs to achieve a consistent fixed bit rate for transport purposes. A rate control mechanism within the encoder is responsible for adapting the QP for each MB to match a given bit rate. Figure 3.12 shows the graphs for different sequences when varying the bit rate ($100kbps$ to $2000kbps$ for our algorithm, $100kbps\_MV$ to $2000kbps\_MV$ for the MV-based approach). Note that different bit rates influence the performance of the system. Using a high fixed bit rate generally results in small QPs when encoding the MBs. Hence, the same global behaviour as in Figure 3.11 is visible.

An encoder is free to implement its own method for estimating the motion and the H.264/AVC reference software contains several methods for this. We created sequences using the Uneven Multi-Hexagon Search (UMHex), Simplified Hexagon Search (SHex), and Enhanced Predictive Zonal Search (EPZS). The last method uses the default pattern (Extended Diamond). The chosen methods apply different search patterns to find the best match of a specific MB, and as such each of these methods have an influence on the speed of encoding and the resulting MVs that are found. Figure 3.13 shows the graphs for the MV-based and the proposed approach, when using these different motion estimation methods. It can be seen that the MV-based approach is more dependent on the chosen motion estimation method than the proposed approach. A different motion estimation method can result in totally different MVs, resulting in a different behaviour of the MV-based object detection. In contrast, for high thresholds, our system is more or less independent of this parameter. Although, the different MVs result in different residual data, the system is still able to accurately detect the MBs that correspond to moving objects.

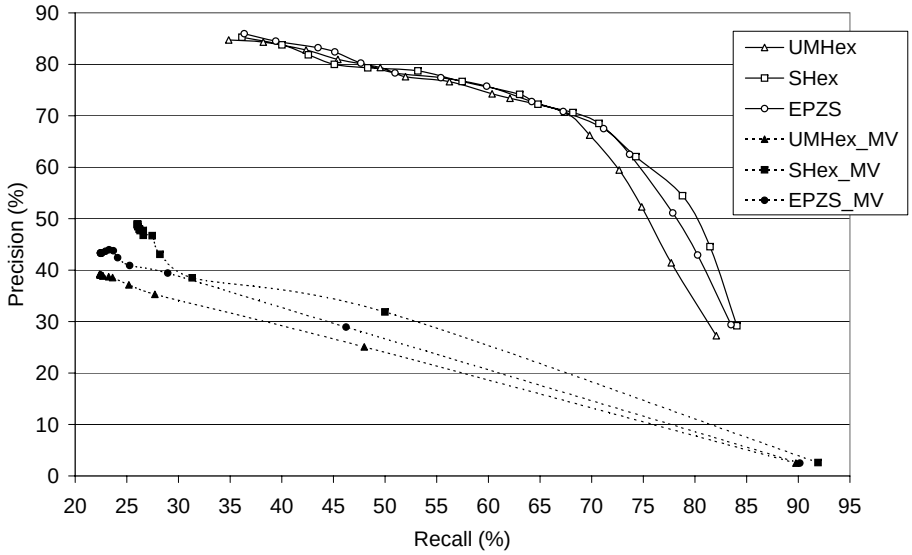


(a)

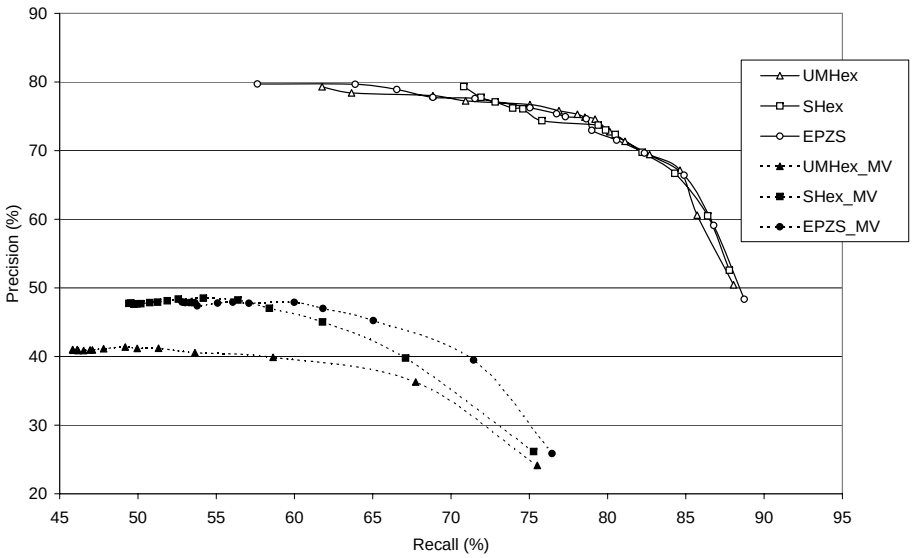


(b)

Figure 3.12: Precision-Recall graph for fixed bitrate versions of (a) the PetsD2TeC2 sequence and (b) the Indoor sequence.



(a)



(b)

Figure 3.13: Precision-Recall graph for different motion estimation methods for (a) the PetsD2TeC2 sequence and (b) the Indoor sequence.

3.6 Conclusions

Video compression is and will remain omni-present in video surveillance systems. Hence, the use of analysis systems that work on the compressed video directly offer several benefits such as the avoidance of full decoding before initial decisions can be taken and the re-use of features calculated during the encoding process. A general approach to detect moving objects in a compressed video stream is to rely on the MVs that were created during the encoding. However, these MVs are created from a coding perspective and do not necesarily represent the actual motion of the objects in a sequence.

Our main contribution in this research area is the introduction of an algorithm that entirely disregards the MV field. As such, we present the first moving object detection system in H.264/AVC compressed domain that relies on the data sizes of the MBs in the compressed video stream.

The proposed method works at high speeds and accurately detects moving objects in H.264/AVC compressed video surveillance sequences. In contrast to the related work, that presents MV-based approaches, our system purely relies on the structure of the compressed bitstream. During a training phase, the numbers of bits that MBs use within a frame are used to create an effective background model. Subsequently, the MB sizes of new images are compared to this model to yield possible foreground regions, which are then spatially and temporally filtered. Additionally, the sizes of the 4×4 transform coefficients within boundary MBs are incorporated to refine the detection results. A comparison on challenging sequences shows that our system achieves better precision and recall values than the related MV-based approaches. Additionally, since the algorithm is restricted to the syntax level, very high execution speeds are achieved (up to 20 times faster than the MV-based approaches).

The work that was presented in this chapter is currently accepted for publication in the Journal of Visual Communication and Image Representation:

1. Chris Poppe, Sarah De Bruyne, Tom Paridaens, Peter Lambert, and Rik Van de Walle. Moving Object Detection in the H.264/AVC Compressed Domain for Video Surveillance Applications. *Visual Communication and Image Representation*

Additionally, work in this domain can be found in following publication:

1. Sarah De Bruyne, Chris Poppe, Steven Verstockt, Peter Lambert, and Rik Van de Walle. Estimating Motion Reliability to Improve Moving

Object Detection in the H.264/AVC Domain. In *To be published in Proceedings of The International Conference on Multimedia and Expo (ICME) 2009*, June 2009

Furthermore, our work has been presented as a demo at the 10th European Conference on Computer Vision (ECCV). Finally, it lead to the best poster award at the 9th Firw Phd Symposium for following publication:

1. Chris Poppe and Rik Van de Walle. High-speed Moving Object Detection in H.264/AVC Compressed Domain. In *Proceedings 9th FirW Phd Symposium*, pages 204–205, December 2008

Chapter 4

Metadata: Representing Detected Objects

Quis cusodiet ipsos custodes? - Who shall guard the guardians?
– Decimus Iunius Iuvenalis - 'Satire VI' (c. 100)

4.1 Introduction

Until now, we focused on the detection of moving objects, where we presented approaches in two different domains, being the pixel and the compressed domain. The detection results, whether they consist of the foreground pixels or bounding boxes, are generally regarded as low-level features. To make intelligent decisions when an object is detected, more high-level information is required (e.g., location, timing information, shape or speed of the object).

During the creation of large distributed surveillance systems the need arises to share these extracted features (both low- and high-level) between different modules or even between different systems. The features can be regarded as annotations of the captured video sequences. This information that describes data is generally called metadata and it has applications in a broad range of domains within computer science.

During our participation in the research projects DANAE [1] and PECMAN[2], we came in touch with such metadata applications in a different domain than video

[1] http://danae.rd.francetelecom.com/index.php
[2] http://www.ibbt.be/en/project/pecman

surveillance, being (personal) content management. A video surveillance system could be seen as specific case of such a content management system, the main difference is that the metadata is generated by video analytics, whereas metadata used for personal content is generally user-created (e.g., tags and annotations of pictures). Next, we discuss how video surveillance systems use metadata and what the problems are of using XML-based metadata standards in this case. Additionally, we introduce the concept of personal content management and show how the same problems occur. In the rest of this chapter we present our solutions to these problems in the context of personal content management systems.

4.1.1 Metadata in Video Surveillance

Black et al. presented a framework for event detection and video content analysis within a multi-camera surveillance system [83]. They made a data model suited to describe images, objects (and their motion), and semantic aspects. This data model is represented in different layers and was modelled in a database. Metadata is generated based on the layers to combine all the information and to increase the efficiency of querying. The actual metadata format was not reported, but their future work suggested using a common metadata standard for cooperation with other surveillance systems.

As the previous example suggests, to make metadata practically usable for information exchange between two or more modules, a common machine-readable metadata format is needed. This format describes which metadata can be used to describe the information of interest and how the metadata is structured. When using a common metadata format, software tools for automated manipulation can be created. One popular format of metadata is XML (eXtensible Markup Language), which allows to structure the information so that it is machine-parseable. In this case, an XML schema (according to the XML Schema language [84]) can be used to describe the structure, and the actual metadata is represented by an XML document or instance that conforms to the schema.

This approach can already be found in existing video surveillance systems (e.g., the CANDELA project [85] that uses MPEG-7 [86] to describe the features). Different metadata formats have been suggested to describe video surveillance related metadata.

Zerzour et al. presented the VIGILANT system and created a semantic model for content and event based indexing of surveillance video [87]. It consists of a

data model described using constructs from the KL-ONE language (used to explicitly represent conceptual information as a structured inheritance network). However, this language is not widely used for describing video surveillance metadata and disturbs the integration with different systems.

The need for describing video analytics results with a common metadata format also arises when considering the evaluation (and comparison) of different video surveillance systems. Young and Ferryman presented a dataset of video sequences for evaluation of different algorithms and defined a common XML-based format to describe the detection results [88]. It contains information on objects and trajectories. The use of the common format allows to automatically and objectively analyse the performance of different algorithms.

List et al. presented an XML-based Computer Vision Markup Language (CVML) [89]. Additionally, they offered a free software library called CoreLibrary that assists people in handling the language. It has been used to describe hand-labelled ground truth datasets as part of the CAVIAR project[3].

Annesley et al. gave an interesting overview of the usage of MPEG-7 for video surveillance in general [90]. They presented examples on how MPEG-7 descriptors can be used. Since MPEG-7 is a large (and complex) metadata standard, they proposed a video surveillance specific profile to limit the amount of descriptors that need to be supported. Additionally, they created a Visual Surveillance XML schema (VS7) that uses some of the MPEG-7 descriptors and contains new types.

As can be seen, a number of different approaches exist in formatting the metadata associated with a video surveillance system. However, there is not one general metadata standard that is generally accepted, and most likely such a standard will not be introduced in the near future. Consequently, combining different metadata schemes with each other seems to be the only solution to create interoperability between different modules and systems.

However, combining different XML-based formats, created from different perspectives, is hard. For example, CVML and VS7 can be used to denote the same concepts, but the structure and terms used are very different. Mapping such XML fragments on each other is obviously a cumbersome task. The usage of eXtensible Stylesheet Language Transformations (XSLT) stylesheets [91], which were specifically created to transform XML instances, cannot always encompass the differences between different metadata standards. Additionally, when using XML to describe metadata, it is hard to describe semantic aspects. XML was mainly created to structure information and in many cases

[3] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

a metadata standard consists of an XML schema to denote the structure and the metadata fields that can be used, and a complementary textual description of the actual semantics of the metadata fields.

Recent efforts create languages that allow to explicitly define semantic information. An example of this within the context of video surveillance systems is the Video Event Representation Language (VERL), suggested by Nevatia et al. [92, 93]. It is used to describe events and relations in video sequences using an ontology. Additionally, a Video Event Markup Language (VEML) was created that allows to annotate instances of the events described in VERL. Initially, VEML was a proprietary language constructed in XML, but in the final version, the base format used is the Web Ontology Language (OWL) [12], designed by the W3C Web Ontology Working Group. However, Nevatia et al. reported problems for describing the entire VERL ontology with OWL, so not all constructs are available as OWL instances. Integration with the MPEG-7 standard was proposed as future work but no information was given on how this could be done.

The issues described here are general problems when trying to combine different metadata standards or formats. Within this dissertation, we tackled this interoperability problem in a different domain than that of video surveillance systems. We focused on metadata that is used for annotation of images within a personal content management system. We propose the use of Semantic Web technologies to deal with the problem of using diverse metadata standards and to create a semantic personal content management system. The next section gives a general introduction to the problem that one faces when creating such systems.

4.1.2 Personal Content Management Systems (PCMSs)

These days, the amount of multimedia content is increasing considerably. Several concurrent trends are making this happen. Current software made creating, sharing, and editing multimedia resources a common and easy task. Digital camera sales overtook film-based cameras in 2003 and the numbers keep on increasing [94]. Online communities that allow sharing of pictures or videos are blooming and users can easily and rapidly annotate their own or others content. This increasing amount and diversity of content, metadata, and users, makes it a hard task to manage, retrieve, and share the multimedia content. In this sense, Content Management Systems (CMS) have been used for several years. However, nowadays the user is becoming a producer of content, and there is a need to manage this personal content as well, hence the introduction

of a Personal Content Management System (PCMS). In this case, the user creates, annotates, organizes, and shares his personal content. In this context, the metadata have never had such an important impact on the capability of PCMSs to manage, retrieve, and describe content [95]. Nevertheless, creating a PCMS that can work with different sorts of metadata is a difficult task. Numerous metadata formats exist and generally their structure is defined in a metadata scheme (e.g., using XML Schema), while the actual meaning of the metadata fields is given in plain text [86, 96, 97]. This leads to several interoperability issues since there is no formal representation of the underlying semantics. For example, different MPEG-7 metadata constructs can be used to denote the same semantic concept [98]. Even in more recent standards, like MPEG-21, we can see problems by not formally defining the semantics, as we have shown in [99]. This interoperability problem was the main topic of the W3C Multimedia Semantics Incubator Group (MMSem) in which we have actively participated within the photo use case [100]. We adopt the vision of this group and use semantic web technologies to solve these interoperability issues within the context of a PCMS.

We introduce a semantic approach to build a PCMS. This work is part of the PeCMan (PErsonal Content MANagement) project, in which a PCMS is created that is completely metadata-driven. We propose a metadata model, representing system-, security-, and user-related metadata in the context of a PCMS. The model is used as an upper ontology (expressed by an OWL schema) that is linked to a set of lower-level metadata ontologies (e.g., formal representations of MPEG-7, Dublin Core, and DIG35). To show the extensibility of our system, we create an ontology conform to the DIG35 specification and include it within the proposed model.

The next section gives related work within the context of using semantic technologies to align different XML sources. In Section 4.3, we elaborate on the created PCMS metadata model. Subsequently, in Section 4.4 we discuss the interoperability issues that we faced when combining existing metadata schemes with our metadata model. Accordingly, a layered approach is presented that builds upon and combines formal representations of existing metadata schemes. In Section 4.4.3 we elaborate on the DIG35 ontology that we created and Section 4.4.4 shows the mappings between the different ontologies. Section 4.5 presents the metadata service that is built around our model and discusses the used technologies. A use case scenario is shown in Section 4.7 to illustrate our system and, finally, conclusions are drawn in Section 4.8.

4.2 Related Work

Semantic technology is a vivid area of research and a number of different approaches exist in creating semantic multimedia management systems. Two broad categories can be found in the related work, consisting of approaches that focus on integration of metadata standards and approaches that propose semantic multimedia systems.

The first category tries to integrate the various multimedia metadata formats by creating formal representations and/or frameworks to integrate them. The following paragraphs describe related work that represents this category.

Hunter et al. describe a semantic web architecture in which RDF schema [101] and XML schema are combined [102]. The RDF schema defines the domain-specific semantic knowledge by specifying type hierarchies and definitions, while the XML schema specifies recommended encodings of metadata elements by defining structures, occurrence constraints, and data types. A global RDF schema (called MetaNet) is used to merge domain-specific knowledge and it is combined with XSLT to accomplish semantic, structural, and syntactic mapping. MetaNet is a general thesaurus of common metadata terms that contains semantic relations (e.g., broader and narrower terms) and is used as a sort of super ontology. However, to reach its full potential, the super ontology should provide more and diverse semantic relations between the commonly used metadata terms. Consequently, using MetaNet allows for some semantic relations but it is not suited to act as a super ontology. The actual mapping between different ontologies happens through XSLT transformations. Since XSLT only relies on template matching, it can be used to transform one XML construct in another, but it is not suited to represent actual semantic relations between different concepts. Consequently, Hunter reports on the difficulties that arise due to lexical mismatching of metadata terms. Moreover, the stylesheets do not make the actual semantic relations publically available for other ontologies.

The structure of the ontologies in our system is mostly related to the work of Cruz et al. [103]. They introduce an ontology-based framework for XML semantic integration in which, for each XML source that they integrate, a local RDF ontology is created and merged in a global RDF-based ontology. During this mapping, a table is created that is used to translate queries over the RDF data of the global ontology to queries over the XML sources. The major difference with their approach is that we make explicit use of OWL to create and link the ontologies whereas they define a proprietary mapping table and introduce an extension to RDF. Moreover, they assume that every concept in the

local ontologies is mapped to a concept in the global ontology. In our case, we have defined the global ontology by analysing the specific needs of a PCMS. Consequently, not all the concepts present in every included metadata standard need to be mapped to our upper ontology.

Hunter and Little defined a framework enabling semantic indexing and retrieval of multimedia content based on the ABC ontology [104]. As in our approach, ontologies are manually created based on the XML schemas that are used. The ontologies are manually linked by using the ABC ontology as a core ontology [105]. This core ontology offers concepts and relations that allow to describe knowledge (events, actions, entities, etc.) at a high level. As such, domain-specific ontologies can use the ABC ontology to map specific concepts on each other, creating rich semantic representations across domains. However, the ABC ontology is not suitable to relate different multimedia metadata formats (which describe multimedia and not that much existing concepts or events) to each other. Indeed, only one multimedia metadata format (MPEG-7) is used in combination with two domain-specific ontologies, which are linked through the ABC ontology. XML schemas are linked to the ontologies through a proprietary system that extends the given schemas with specific attributes. Using XPath queries that process these schemes, XML instances are accordingly mapped upon instances of the corresponding ontologies. Rules are used to deduce additional relations between instances.

Vallet et al. used ontological knowledge to create a personalized content retrieval system [106]. They presented ways to learn and predict the interest of a user for a specific multimedia document to personalize the retrieval process. More specifically, the system adds a weight to concepts of the available domain ontologies and these are adjusted with every query a user does. However, the way that these concepts are related to the actual multimedia documents is similar to free-tagging systems. Each multimedia document is associated with a vector of weighted domain concepts. For example, if a picture is annotated with the concept $Paris$ it is not possible to decide whether the image depicts the city of Paris or if the picture is taken in Paris.

Arndt et al. created a well-founded multimedia ontology, called the Core Ontology for Multimedia (COMM) [98]. The ontology is based on the MPEG-7 standard and uses the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) as modelling basis [107]. In this sense they formally describe MPEG-7 annotations, however mapping other existing metadata formats onto this model is a cumbersome task.

The second category presents general solutions to make a semantic multimedia

management system. These include metadata representations and an architecture that allows to reason upon, adapt, and query this information.

Dönderler et al. presented an overview of the design and implementation of a video database management system [108]. The system allows to create semantic annotations of multimedia (video) by annotation or extraction. For this purpose a video-annotation tool and a feature extractor were proposed. The resulting annotations are subsequently used by a query processor to allow advanced querying. However, the semantic knowledge is hard-coded as a database, which makes it hard to change the system.

Shallauer et al. presented a description infrastructure for audiovisual media processing systems [109]. The system consists of an internal metadata model and access tools to use it. The model used to describe the audiovisual media is a self-defined profile of the MPEG-7 standard (called the Detailed Audiovisual Profile). By defining this profile they discarded the generality and complexity of the entire MPEG-7 standard. Similar work was done by Annesley et al, who defined a profile for the use of MPEG-7 within video surveillance systems, as discussed in Section 4.1. However, semantic constraints within the Detailed Audiovisual Profile are only described in textual form in the profile description. Therefore, these constraints need to be enforced (hard-coded) by the software tools used in the system.

Petridis et al. created a knowledge infrastructure and experimental platform for semantic annotation of multimedia content [110]. They use DOLCE as core ontology to link domain-specific ontologies (e.g., describing sport events like tennis) to multimedia ontologies. The latter consist of the Visual Descriptor Ontology (VDO) [111], an ontology based on MPEG-7's Visual Part [112] and a Multimedia Structure Ontology (MSO) using the MPEG-7 Multimedia Description Scheme [113]. However, their infrastructure is solely made to semi-automatically combine low-level visual features into semantic descriptions of the multimedia resource. Unfortunately, they do not consider other relevant metadata (e.g., creation location and human annotations) that is usually available and can be used to infer semantic information. Moreover, they do not allow other multimedia metadata schemes in their system.

Asirelli et al. presented an infrastructure for MultiMedia Metadata Management (4M) [114]. The infrastructure enables the collection, analysis, and integration of media for semantic annotation, search, and retrieval. It consists of an MPEG-7 feature extraction module, an XML database for storage of MPEG-7 features as XML files, an algorithm ontology, an MPEG-7 ontology, and an integration unit that merges the different modules together. As such, they suc-

ceed in reasoning about MPEG-7 features and linking different algorithms to extract MPEG-7 features. However, the restriction of using MPEG-7 XML files prevents the use of other existing metadata formats.

Garcia and Celma proposed a system architecture to achieve semantic multimedia metadata integration and retrieval by building upon an automatically generated MPEG-7 ontology [115]. They presented an automatic XML Schema to OWL mapping and apply it to the MPEG-7 standard to create an OWL Full MPEG-7 ontology. However, this mapping is fully automatic and can only extract semantics out of the structure of the XML schema. Hence, semantics not defined within the schema are not present in the resulting ontology. A system architecture to achieve semantic multimedia metadata integration and retrieval is proposed. This system uses the MPEG-7 ontology as upper ontology to which other ontologies are mapped. An RDF storage system is used for management and the data is queried using RDQL (RDF Data Query Language). However, by using the automatically generated MPEG-7 ontology, the semantic knowledge that can be represented is restricted. Additionally, MPEG-7 has gained only moderate popularity due to its complexity, and numerous metadata standards exist that work on the same conceptual level [116]. Therefore, building the upper ontology solely out of MPEG-7 concepts is too restrictive.

As the related work shows, no current architecture allows to incorporate and combine several different existing metadata formats for use within a PCMS. The possibility to include and work with different metadata formats is a prerequisite for the creation of a practically useful PCMS. Different metadata standards exist and instances of these metadata schemes are everywhere. Therefore, to work with these formats, we need to address the interoperability between the different standards and cannot rely on one metadata standard alone (like MPEG-7). Consequently, we present a semantic layered metadata model and describe a metadata service tailored for a PCMS.

4.3 PCMS Metadata Model

In this dissertation, when talking about a PCMS, we imagine a metadata-driven distributed management system where the multimedia content can be placed on several devices (e.g., desktop PC or cell phone), connected over different networks. The actual content is indexed in a separate module, called indexer, that makes intelligent decisions on the technical details of the content storage. As such, the storage capacities and network traffic can be optimized [117]. The metadata related to the content is managed by a metadata service. The indexer makes use of this metadata service to decide upon the location of the content. For this purpose, system-related metadata is created and linked to the content. As such, an end-user only adds metadata to content, while the system interprets this to move the content to its best location. Accordingly, a security service maintains policy rules, based on specific security-related metadata fields of the content. Lastly, to allow personalized management of the content and to enforce the community aspect of the content management system, user-centric metadata is used.

To structure the metadata within the PCMS, we present a metadata model, which is depicted in Figure 4.1. These metadata fields were defined according to a number of use cases inherent to a PCMS. These are:

- Registration of a resource.
- Retrieval of a resource.
- Annotation of the resource (to enhance the retrieval of content).
  - Manual annotation of the resource. This involves tagging or adding metadata according to a specific metadata scheme.
  - Automatic annotation of the resource. This includes feature extraction (e.g., face recognition tools) and importing existing metadata (e.g., reading EXIF header of a picture file).
- Metadata-driven security. This means the possibility to include access rules based on the metadata of the content.

Figure 4.1 shows the metadata classes that can be regarded as independent of the actual type of the multimedia document (e.g., image, video, or audio) and

are discussed in the next section. Metadata that is specific for a certain type is collected within subclasses of the $ManualAnnotation$ class and is the topic of Section 4.3.2.

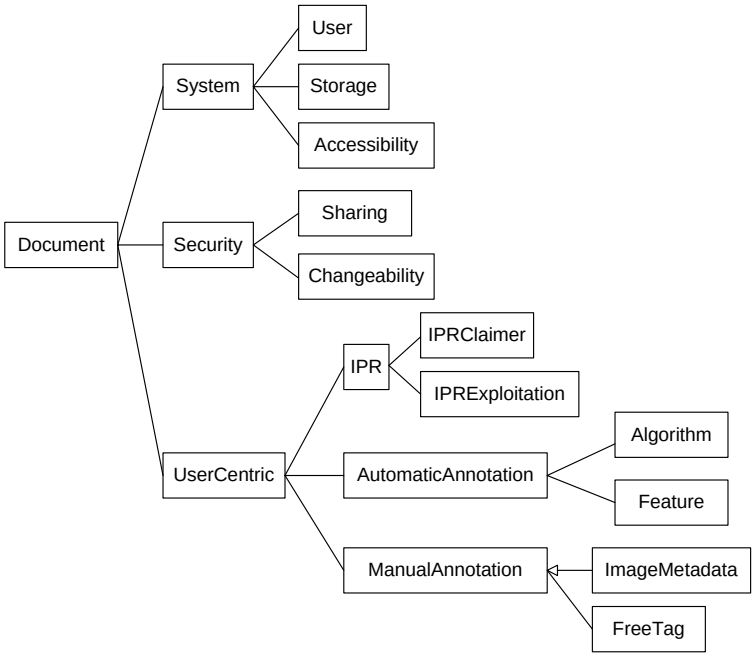


Figure 4.1: PCMS Class Hierarchy.

4.3.1 Content-independent metadata

A resource (e.g., a picture, video, audio file, or textual document) is called a $Document$ and forms the base of our metadata model. The metadata itself reflects the modular architecture of a PCMS and is split up into system-, security-, and user-related metadata fields.

System Metadata

The system metadata holds the owner, storage info, and accessibility rules. The owner is a registered user within the PCMS. The user has an ID, login name, and some additional settings. The storage info holds information about

the current location of the document. This location is influenced by the accessibility rules. If the accessibility is set to “online”, the document is placed on an accessible server. A value of “offline” signifies that the document is never accessible for other devices connected to the PCMS. The last option is to set the accessibility to “always available”, which means that the document is stored both online and on a detachable device.

Security Metadata

The security-related metadata describes the rights that govern the access of other users concerning a specific document. This access is twofold, it includes access to the multimedia resource and to the available metadata. We refer to sharing settings for the former and changeability settings for the latter (the usage of the term ’sharing’ is not common for expressing access to metatadata, hence the term ’changeability’ is used for this purpose). The sharing settings hold a share level, which states whether the document is shared with some users, or is private. If the document is shared, specific permission settings can be used that contain the rights of a user or user group upon the document (e.g., view or modify). If the document is not shared, the permissions are ignored. To manage the access to the metadata, the permissions of a user or user group concerning the metadata are stored within the changeability settings. Fragment 4.1 shows an XML serialization of the security-related metadata. This fragment shows that the actual content is private, but the metadata is visible. This allows to deal with copyright issues so that the content is inaccessible but the metadata can be browsed.

```
<Security>
  <SharingSettings>
    <ShareLevel>private</ShareLevel>
  </SharingSettings>
  <ChangeabilitySettings>
    <Permissions>tag</Permissions>
    <Permissions>see</Permissions>
    <User>urn:pecman:user001</User>
  </ChangeabilitySettings>
</Security>
```

Fragment 4.1: Example of PCMS metadata fragment expressing copyrighted content, serialized in XML.

User-Centric Metadata

The user-centric metadata contains metadata for describing IPR (Intellectual Property Rights), automatic, and manual annotations. We split IPR-related metadata on the security and user level. The security-related metadata as discussed above is typically only accessible through and within the PCMS. The subjects of the permissions are consequently the PCMS users. On the other hand, the IPR-related metadata on the user level is regarded as a global protection scheme. The owner of the document defines the exploitation rights of a certain IPR claimer. The latter is a person (which could be a PCMS user) or organization. The actual exploitation can hold metadata to identify a specific IPR mechanism (e.g., watermark or registration), metadata to impose restrictions upon the use of the document, and metadata to specify obligations resulting from the use of the document (e.g., a fee for watching a movie). This IPR mechanism is in line with the IPR systems described in MPEG-21 Rights Expression Language (REL) and Rights Data Dictionary (RDD) [118]. These define standardized language constructs that can be used to create a licensing system that allows a user to define the rights that other persons have upon his content [119].

Automatic annotation concerns the algorithmic extraction of relevant features from the multimedia resource. The need for this kind of metadata increases with the number of multimedia resources that are stored within the PCMS. If many documents need to be annotated, manual annotation is not an option. Accordingly, automatic annotations are used that describe the content as accurate as possible. Numerous approaches and algorithms exist that try to bridge the semantic gap [120]. A generic approach is used to allow a plethora of algorithms. In this approach, the algorithm itself is described on a high level (e.g., name and reference) and the result of the algorithm can be stored as features, consisting of a header and actual data.

The last category of user-centric metadata are the manual annotations. A manual annotation differs from an automatic annotation in the sense that the former is metadata generated by a human expert, whereas the latter is created by an algorithm that works on the actual multimedia resource.

One general form of manual annotations is free tagging of multimedia content. As can be seen from the numerous Web applications that allow tagging content (e.g., Flickr or Facebook), the popularity and, consequently, necessity of tagging is enormous. However, due to its nature, it is difficult to assign a semantic meaning to free tagging. Different users assign different keywords or tags to the same content. Moreover, several different tags can be used to identify the

same conceptual entity (e.g., house, home, building, or auntie's place). Finally, different spellings, languages, or dialects cause even more problems. To reduce the semantic gap, we introduce a $Context$ field that tries to represent, on a high level, the semantic meaning of the actual tag. This structuring of free tags was inspired by MPEG-7. The field can take the values "who", "what", "where", "when", or "misc" to refer to a person, a subject, a place, the time, or something else, respectively. The major advantage of tagging is the degree of freedom that the user has and the low effort to create a tag. Therefore, we restrict the context fields to those mentioned above.

In addition to the free tagging scenario, we can have other metadata that describe the multimedia resource. However, these are mostly dependent on the actual type of the resource. In this dissertation, we restrict our discussion of the PCMS metadata model to images (the class $ImageMetadata$ is a subclass of the $ManualAnnotation$ class), but the metadata structure of other document types (e.g., audio, video, and textual documents) is similar and some constructs can be reused.

4.3.2 Image-related Metadata

We have actively participated in the creation of an overview on the state-of-the-art of multimedia metadata formats within MMSem [121]. This overview was matched with the general requirements of the PeCMan project to yield the structure of the image metadata incorporated in our metadata model. In this sense common metadata concepts in different existing metadata formats were analysed to see whether they aid in solving the use cases. This image metadata is split up in basic, creation, and content metadata as shown in Figure 4.2.

The basic image parameters define general information about the content (e.g., file name, width and height, and the coding format).

The creational metadata encompasses both general and detailed creation information. The general creation information has fields for the creation time, the image creator, and the image source. This image source field can take the fixed values "digital camera" or "computer graphics" to denote that the image was taken by a digital camera or was created using software, respectively. With the detailed image creation information, one can specify which software or what camera was used in the creation process.

The third category of image metadata, namely the content descriptive metadata, describes the content of an image or a specific region of the image as described by the $Position$. By explicitly including this field we can make

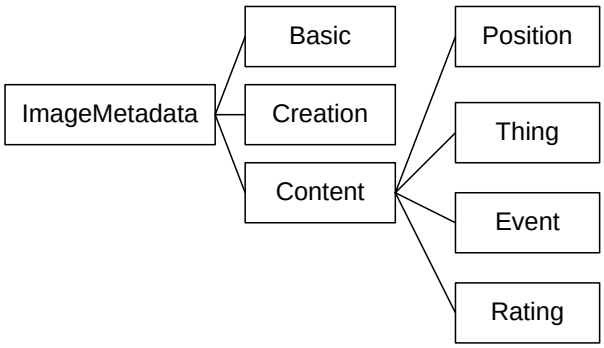


Figure 4.2: Model of image-related metadata.

an annotation about an actual region within an image. This region can be described as a comment ($Comment$), as a point ($Point$), a bounding box ($BoundingBox$), or as a region ($Region$). The latter is defined in terms Bézier curves, represented as $Splines$, as shown in Figure 4.3.

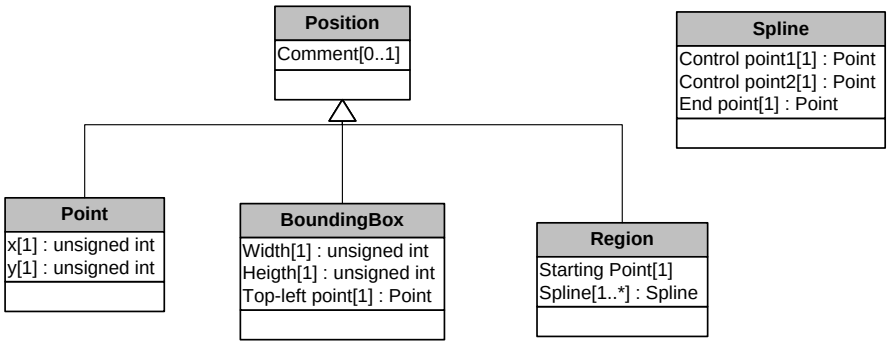


Figure 4.3: Class Diagram of the Position class.

The actual annotation that describes the defined region can be a textual comment, depicted item, depicted event, and rating. The $DepictedItem$ field describes a tangible $Thing$ (see Figure 4.4) that is depicted; this can be an object or living thing, but also a person ($Person$), a group of people ($PersonGroup$) or an organization ($Organization$).

An event is described by an $EventDescription$ and relates to participating

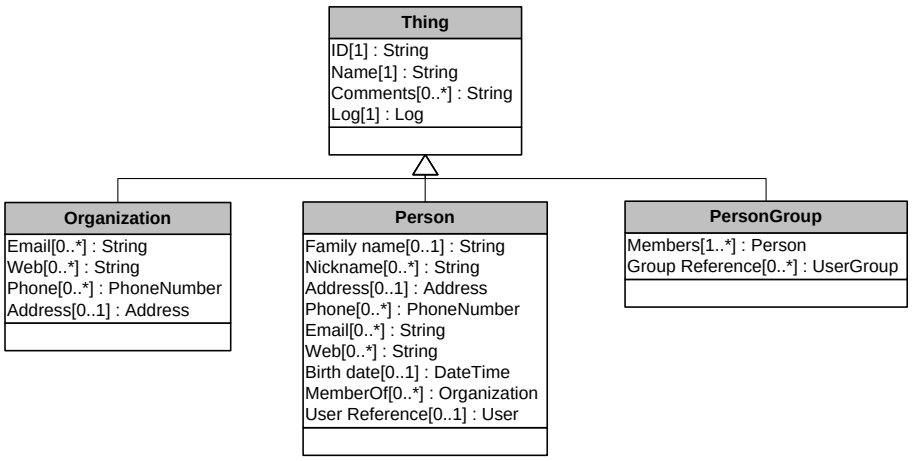


Figure 4.4: Class Diagram of the Thing class.

things, as shown in Figure 4.5. The field $Eventtype$ specifies the type of event (e.g., "soccer game" or "wedding party") and events can be related with each other through the $RelatedEvent$ field.

The rating is defined by a value or score that is given to the content, and a minimum and maximum value to denote the range of the score (e.g., the rating is 3 on a scale from 0 to 5). The possibility to rate each other's content is especially interesting in a community context where people can tag other users' content (e.g., Flickr).

As shown in this section, we have created a metadata model specifically tailored for personal content management systems. The model offers a general overview of the metadata fields of interest and has a large descriptive potential, due to the inclusion of automatic annotations, free tagging, and content descriptive metadata. In the next section, we elaborate on the practical implementation of this model and show how existing metadata standards are combined with the model to allow the description of the content in various formats.

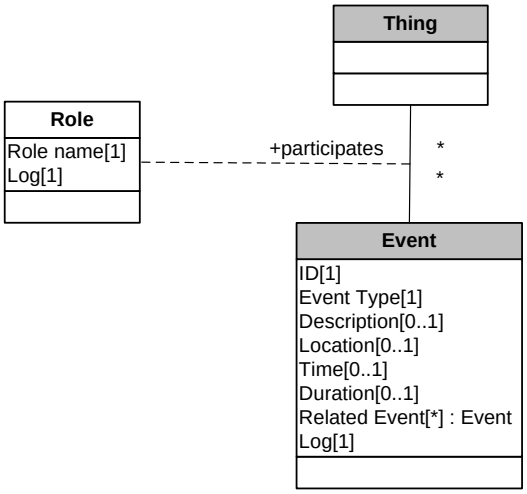


Figure 4.5: Class Diagram of the EventDescription class.

4.4 Semantic PCMS

4.4.1 Interoperability Issues

The metadata model, introduced in the previous section, is specifically tailored to be compact and easy to reuse. Therefore, it is not intended to replace existing multimedia metadata standards. Based on the metadata model, we created an XML schema that allows to specify the structure of the metadata used in our system. To allow the actual end-user to freely annotate content, other existing standardized metadata schemas need to be included. However, when trying to match the XML schemas of different standards we face interoperability problems. These problems were already signalled by the W3C Multimedia Semantics Incubator Group [100]. Although each of the standardized formats introduces interoperability amongst applications that use that standardized metadata scheme, issues occur when using different metadata schemes together. Fragment 4.2 and Fragment 4.3 show two annotations of the same picture but according to different metadata schemes. Both annotations describe a unique identifier, the person who created the picture and the software that was used, the location and time of the creation, and give an identifier used for IPR purposes.

Another example can be found in the context of video surveillance systems.

```
<METADATA>
  <BASIC_IMAGE_PARAM>
    <BASIC_IMAGE_INFO>
      <IMAGE_ID>
        <UID>098f2470-bae0-11cd-b579-08002b30bfeb </UID>
        <ID_TYPE>http://www.digitalimaging.org/dig35/UUID</
            ID_TYPE>
      </IMAGE_ID>
    </BASIC_IMAGE_INFO>
  </BASIC_IMAGE_PARAM>
  <IMAGE_CREATION>
    <IMAGE_CREATOR>
      <PERSON_NAME>
        <NAME_COMP TYPE="Given">Yoshiaki<NAME_COMP>
        <NAME_COMP TYPE="Family">Shibata<NAME_COMP>
      </PERSON_NAME>
    </IMAGE_CREATOR>
    <SOFTWARE_CREATION>
      <SOFWARE_INFO>
        <Model>Wizzo Extracto</MODEL>
        <VERSION>2</VERSION
      </SOFTWARE_INFO>
    </SOFTWARE_CREATION>
  </IMAGE_CREATION>
  <CONTENT_DESCRIPTION>
    <LOCATION>
      <ADDRESS>
        <ADDR_COMP TYPE="City">Tokyo</ADDR_COMP>
        <COUNTRY>jp</COUNTRY>
      </ADDRESS>
      <CAPTURE_TIME>
        <EXACT>2000-10-10T19:45:00+09:00<EXACT>
      </CAPTURE_TIME>
    </LOCATION>
  </CONTENT_DESCRIPTION>
  <IPR>
    <IPR_IDENTIFICATION>
      <IPR_IDENTIFIER><IPR_ID>RID#</IPR_ID></IPR_IDENTIFIER
          >
    </IPR_IDENTIFICATION>
  </IPR>
</METADATA>
```

Fragment 4.2: Example of metadata expressed in DIG35-format.

```
<Mpeg7>
  <DescriptionMetadata>
    <Confidence>1.0</Confidence>
    <Version>1.1</Version>
    <LastUpdate>2001-09-20T03:20:25+09:00</LastUpdate>
    <PublicIdentifier type="UUID">098f2470-bae0-11cd-b579
        -08002b30bfeb</PublicIdentifier>
    <PrivateIdentifier>completeDescriptionExample</
        PrivateIdentifier>
    <Creator>
      <Role href="creatorCS"><Name>Creator</Name></Role>
      <Agent xsi:type="PersonType">
        <Name>
          <GivenName>Yoshiaki</GivenName>
          <FamilyName>Shibata</FamilyName>
        </Name>
      </Agent>
    </Creator>
    <CreationLocation>
      <Region>jp</Region>
      <AdministrativeUnit>Tokyo</AdministrativeUnit>
    </CreationLocation>
    <CreationTime>2000-10-10T19:45:00+09:00</CreationTime>
    <Instrument>
      <Tool><Name>Wizzo Extracto ver. 2</Name></Tool>
      <Setting name="sensitivity" value="0.5"/>
    </Instrument>
    <Rights><RightsID>RID#</RightsID></Rights>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <!-- more elements here -->
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Fragment 4.3: Example of metadata expressed in MPEG-7 format.

```
<frame number="50">
   <objectlist>
      <object id="0">
         <orientation>148</orientation>
         <box xc="77" yc="73" w="21" h="16"/>
         <appearance>visible</appearance>
         <hypothesislist>
            <hypothesis id="1" prev="1.0" evaluation="1.0">
               <movement evaluation="1.0">
                   walking
               </movement>
               <role evaluation="1.0">walker</role>
               <context evaluation="1.0">walking</context>
               <situation evaluation="1.0">
                   moving
               </situation>
            </hypothesis>
         </hypothesislist>
      </object>
   </objectlist>
</frame>
```

Fragment 4.4: Example of CVML metadata fragment describing a moving person.

Fragment 4.4 shows a fragment that describes the event of a detected person using CVML constructs [89]. Similarly, Fragment 4.5 shows an XML description of a detected person using the Visual Surveillance XML schema (VS7) [90].

These examples illustrate the issues of interoperability created when using multiple metadata standards. The same concepts are described but in a totally different format. Note that even when using one single standard (e.g., MPEG-7) to describe a resource, issues in interoperability can exist due to a lack of precise semantics [122]. As these examples show, using XML Schema is not sufficient. Consequently we use Semantic Web technologies to deal with these issues by creating a semantic metadata model, discussed in the next section.

4.4.2 Semantic Metadata Model

Semantic web technologies allow to alleviate the interoperability issues within one metadata standard. For example, efforts have been undertaken to translate MPEG-7 into an ontology and through appropriate frameworks to enable its

```
<VS7:VS7 xmlns:VS7="xsdVS7" xmlns:mp7="
    urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org
    /2001/XMLSchema-instance" xsi:schemaLocation="xsdVS7 vs7
    .xsd">
  <DescriptionMetadata>
    <mp7:Comment>
      <mp7:FreeTextAnnotation>
        A Visual Surveillance Schema (VS7) document
      </mp7:FreeTextAnnotation>
    </mp7:Comment>
  </DescriptionMetadata>
  <Media id="A">
    <MediaInstance>
      <mp7:InstanceIdentifier>
        Camera1
      </mp7:InstanceIdentifier>
      <mp7:MediaLocator>
        <mp7:MediaUri>
          file:/K:/camera1.avi
        </mp7:MediaUri>
      </mp7:MediaLocator>
    </MediaInstance>
  </Media>
  <LLID id="LLID1">
    <TemporalMask>
      <mp7:SubInterval>
        <mp7:MediaRelIncrTimePoint mediaTimeUnit="
            PT1N25F" mediaTimeBase="../../../Media[0]">
          1058
        </mp7:MediaRelIncrTimePoint>
        <mp7:MediaIncrDuration>1</mp7:MediaIncrDuration>
      </mp7:SubInterval>
    </TemporalMask>
    <Mask>
      <BB mp7:dim="4">187 162 282 409</BB>
      <ScalableColor numOfCoeff="16"
          numOfBitplanesDiscarded="0">
        <mp7:Coeff>
          -202 59 27 42 5 11 19 14 6 13 11 22 6 11 16 7
        </mp7:Coeff>
      </ScalableColor>
    </Mask>
  </LLID>
</VS7>
```

Fragment 4.5: Example of VS7 metadata fragment describing a detected object.

integration with other ontologies, thus enhancing interoperability [98,102,115, 122].

To solve interoperability issues inherent to the use of several different metadata schemes, the ideal scenario would be to create a commonly accepted multimedia (metadata) ontology that encompasses all the concepts needed for annotation of multimedia resources. However, this is not feasible as can be seen by the many different existing metadata standards. Moreover, systems, companies, and communities usually want the freedom to employ very specific annotations for internal purposes while other annotations may be made public. Therefore, we chose to create a specific upper ontology, based on our metadata model, called the PeCMan ontology, using OWL. Relevant parts of this ontology are located in Appendix D.

Different multimedia metadata ontologies can be linked to this upper ontology. As such, a hierarchical system of two layers is created. The upper layer contains concepts on the system, security, and user level suited for content management systems. The lower layer exists of several multimedia ontologies, which can be used to describe multimedia resources in various application scenarios.

The number of existing multimedia metadata standards has shown that there is not one commonly accepted standard that provides everything. Different standards are used, whether they are small and simple (PhotoRDF [123]) or broad and complex (MPEG-7). Conceptually, the multimedia metadata formats are on the same level, i.e. they all describe content. Consequently, we regard each metadata format as equally important and will handle the ontologies representing them as such.

Figure 4.6 shows a layered metadata model consisting of the created PCMS model and the underlying metadata standards. Between the different ontologies so called mappings are needed, which consist of mapping ontologies and inference rules. These define the relations between classes, properties, and instances of involved ontologies and are discussed in Section 4.4.4.

The way we organize the different metadata schemes allows intelligent reasoning on different levels. The combination of the formal representation of different metadata standards in the lower layer allows broadening the search space when looking for content based on certain fields of a specific metadata scheme. The personal content management model that works as an upper ontology allows the application of the semantic knowledge to make intelligent decisions on system level (e.g., make content accessible for the depicted user, move an image to another storage device based on the size of the resource).

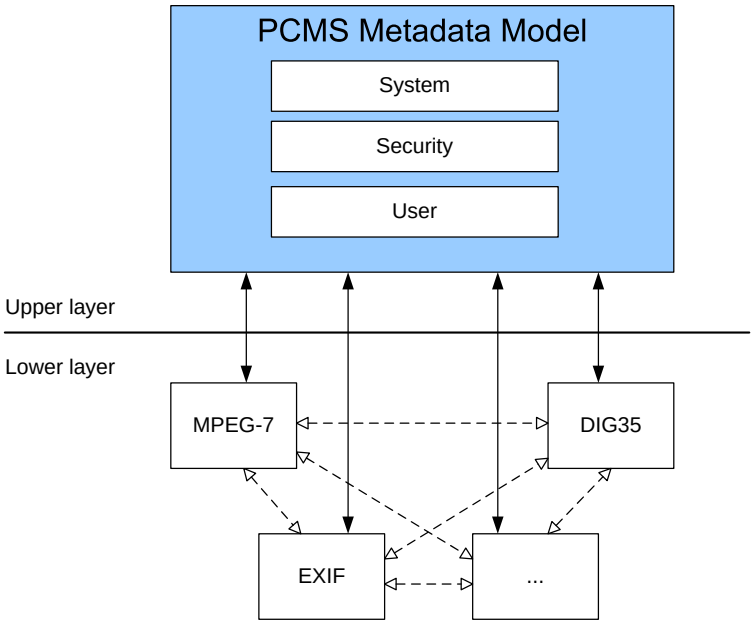


Figure 4.6: Layered metadata model; the dashed arrows denote mappings between ontologies in the lower layer, the full arrows denote cross-layer mappings.

4.4.3 Lower layer

The layered approach allows including new and existing ontologies. To show the extensibility of the system, we add a new metadata ontology based on the DIG35 metadata format. Since for this standard no formal representation exists today, we have built a DIG35 ontology.

DIG35 was developed by the DIG35 Initiative Group, which is part of I3A (not-for-profit International Imaging Industry Association), the largest imaging industry group worldwide. The DIG35 specification includes a standard set of metadata for digital images, which promotes interoperability and extensibility, as well as a uniform underlying construct to support interoperability of metadata between various digital imaging devices. The DIG35 Initiative Group chose for XML as the recommended reference implementation structure for their metadata scheme (see Fragment 4.2 for an example of DIG35 XML instance data). Although this allows to define the structure of the actual metadata, it cannot sufficiently describe the underlying semantics. When semantic reasoning is the goal, a formal representation of this metadata schema

is needed. Therefore, within the context of MMSem, we created the DIG35 ontology using OWL. [4]

For the development of the formal representation we have chosen not to use an automatic conversion like Garcia et al. [115], since, as was the case with the MPEG-7 standard, DIG35 defines the underlying semantics of the different metadata fields within a textual specification. Consequently, we manually created classes and properties based on the DIG35 specification, which allows us to define semantic relationships that are not derivable from the provided XML schema. The final ontology is an OWL DL ontology, consisting of approximately 150 classes.

According to the different parts that are covered by the metadata specification, we created different OWL schemas. These consist of image creation, basic image parameters, content description, image history, and IPR ontologies. Note that the image-related metadata in the PeCMan ontology is similar. Additionally, a number of fundamental ontologies were created, which actually are domain-independent and could be reused for different purposes. These ontologies represent fundamental concepts like addresses, persons, dates and times, events, locations, and other concepts that are also used in the DIG35 specification. These concepts can easily be matched with other existing ontologies (like FOAF or Dublin Core) through a separate mapping ontology.

4.4.4 Mapping and rules

A mapping ontology typically consists of basic OWL or RDFS constructs (e.g., owl:equivalentClass and rdfs:subPropertyOf) between concepts of different ontologies. Fragment 4.6 shows an example of a mapping between the DIG35 ontology and the PeCMan ontology. The figure shows how standard OWL constructs are used to, for example, map a DIG35 $Person$ class on the conceptually equivalent PeCMan $Person$ class (line 5).

Note that, for practical implementations, a mapping as presented above is not sufficient. Rules are needed to create advanced conditional relationships, for example to declare instance equivalence when certain properties match. Within a PCMS, one wants to offer content retrieval based on metadata. As such, the actual instances of this data are sometimes not known beforehand. Hence, we cannot define relations on them in the pre-determined mapping ontologies. However, by defining rules we can automatically link new instances to those

[4] The DIG35 ontology can be found at
http://multimedialab.elis.ugent.be/users/gmartens/ontologies/DIG35

```
<owl:ObjectProperty rdf:about="../DIG35/PersonDescription.
    owl#person">
  <rdfs:subPropertyOf rdf:resource="../Pecman/Content.owl#
      depictedItem"/>
</owl:ObjectProperty>

<owl:Class rdf:about="../DIG35/Person.owl#Person">
  <owl:equivalentClass rdf:resource="../Pecman/Person.owl#
      Person"/>
</owl:Class>

<owl:Class rdf:about="../DIG35/Content.owl#Content">
  <owl:equivalentClass rdf:resource="../Pecman/Content.owl#
      ImageContentDescription"/>
</owl:Class>
```

Fragment 4.6: Example of mapping using OWL constructs within the DIG35 to PeCMan mapping (the namespaces were abbreviated for layout purposes).

that are stored within the system. Fragment 4.7 shows such a rule (we adopt the informal notation declared in the SWRL (Semantic Web Rule Language) submission to give a human readable form of the rules [124]). The rule relates individuals of the class $Person$ within the DIG35 ontology to individuals of the class $Person$ within the PeCMan ontology. If instances of these classes have the same values for their corresponding name and family name properties, they are considered to be equal. This rule makes it possible to match an instance of a PeCMan $Person$ to an instance of a DIG35 $Person$ based on their properties.

```
dig35:Person(?x1) ∧ pecman:Person(?x2)
∧ dig35:givenName(?x1,?y1) ∧ pecman:name(?x2,?y2)
∧ dig35:familyName(?x1,?y3) ∧ pecman:familyName(?x2,?y4)
∧ (?y1=?y2) ∧ (?y3=?y4)

⇒ owl:sameAs(?x1,?x2)
```

Fragment 4.7: Rule for mapping PeCMan $Person$ instances to DIG35 $Person$ instances.

Furthermore, rules are needed to relate certain constructs in different ontologies. For example the rule shown in Fragment 4.8 states that if an in-

stance of the DIG35 $Content$ class *x1* is related to an instance *x3* through the $contentPersonDescription$ and $person$ object properties, it can be inferred that *x1* is the subject and *x3* is the object of the property $depictedItem$ from the PeCMan ontology (in this example the actual classes of instances *x2* and *x3*, being $PersonDescription$ and $Person$ respectively, can be omitted). As such we can find depicted persons within the PeCMan ontology, based on descriptions in the DIG35 ontology.

```
1  dig35:Content(?x1) ∧ dig35:person(?x2,?x3)
2                     ∧ dig35:contentPersonDescription(?x1,?x2)
3  ⇒ pecman:depictedItem(?x1,?x3)
```

Fragment 4.8: Rule to relate the $depictedItem$ property to a specific construct in DIG35.

In the context of video surveillance systems, one can follow the same steps for the creation of a semantic metadata model as discussed above. First, a general metadata model for video surveillance needs to be created. Subsequently, this can be included in a layered model in which the lower layer contains formal representations of existing metadata formats used in video surveillance (like CVML and VS7). Similarly, rules and mappings are needed as described above.

The next section discusses some practical aspects of the creation of a semantic PCMS. More specifically, we elaborate on how it is used within the PeCMan project.

4.5 Metadata Service

Within the PeCMan project, the metadata service needs to allow the upload, storage, management, and retrieval of metadata. The underlying framework for the metadata service is a distributed architecture based on Web services, from the BRICKS project, which is discussed in the next section [125]. This framework incorporates the metadata schema as we defined above and allows to reason on RDF triples based on the present ontologies.

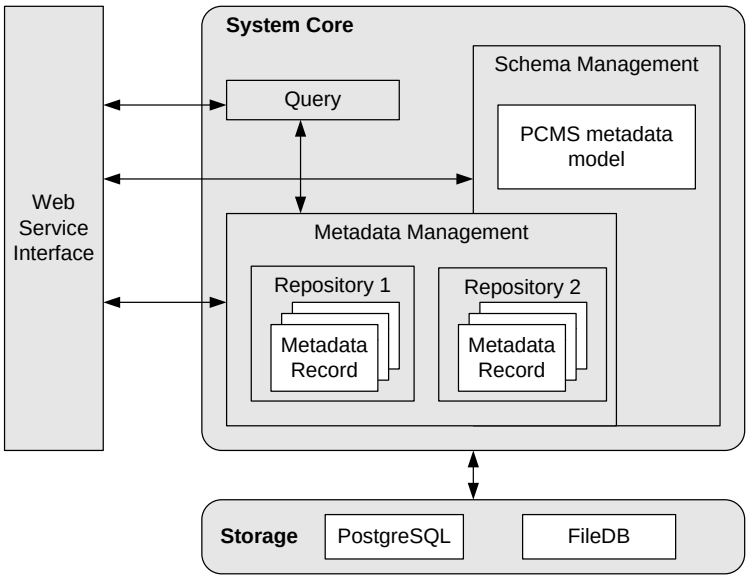


Figure 4.7: Metadata architecture of Bricks environment.

4.5.1 Metadata Management

BRICKS (Building Resources for Integrated Cultural Knowledge Services) was a European project, funded by the IST priority in FP6[5]. It aims to establish the organizational and technological foundations of a digital library, which refers to a networked system of services over globally available collections of digital multimedia documents, providing several layers of knowledge to a variety of users and access methods. The actual infrastructure consists of integrated, but independent, software units, which are deployed on the nodes of the architecture. These nodes, called $Bnodes$, can connect to each other through a peer-to-peer mechanism and use available resources for content and metadata management. Each node is characterized by a Web interface giving access to administration, cataloguing, consultation, annotation, and personalization of content. Our metadata service is implemented within such a $Bnode$. Figure 4.7 shows the internal structure of the metadata service. The $SchemaManagement$ module stores arbitrary metadata schemas, both standard and proprietary, which are described in OWL DL. These schemas describe the actual metadata, denoted as $MetadataRecords$. The query module

[5]http://cordis.europa.eu/ist/

uses Lucene[6] for simple text-search querying. Moreover, advanced ontology queries are expressed in SPARQL (SPARQL Protocol And RDF Query Language [126]) and executed by an underlying Jena[7] RDF query processor.

An external content manager is used to store and manage object identifiers that refer to the actual multimedia content. Within PeCMan, the management of this content is done through a global index that keeps track of the location of the resources. Consequently, within the metadata service, the metadata is linked to these object identifiers. The metadata service holds a Web service giving acces to setters and getters of specific metadata fields. Internally these are translated to the appropriate operations on the RDF triple store, according to the OWL schemas.

Within PeCMan, the metadata access happens through a Web service, invoking methods upon specific metadata fields. However, since many metadata exists in XML format, we need a way to obtain the actual RDF instances of the different ontologies, as such avoiding the usage of the methods for the insertion of the metadata. For this purpose, we introduce a generic XML to RDF converter, discussed in Section 4.6, which allows the automatic transformation of the XML metadata to RDF.

4.6 XML to RDF Conversion

Today, many metadata standards are expressed in XML Schema and there are numerous multimedia documents with XML-based annotations. Moreover, several software agents have specific XML-based parsers to interpret multimedia metadata. Consequently, to make the system practically usable, we allow communications expressed in XML format. To implement this within the PCMS, a conversion tool is needed to automatically generate the RDF triples out of the XML fragments. For this purpose, two major approaches exist: fixed XML to RDF mappings and ontology-dependent XML to RDF mappings. The former uses a mapping based on the XML schema and disregards the actual ontology [115, 127, 128]. The ontology-dependent mappings specify a XML to RDF mapping document that is specifically tailored for the actual ontology [103, 129–131]. They mostly differ in the way the mapping document is created. We utilize a generic XML to RDF convertor created in our research group [131]. This convertor uses an XML document as mapping document that defines specific mapping rules between an XML instance document and

[6] http://lucene.apache.org/java/docs/
[7] http://jena.sourceforge.net/

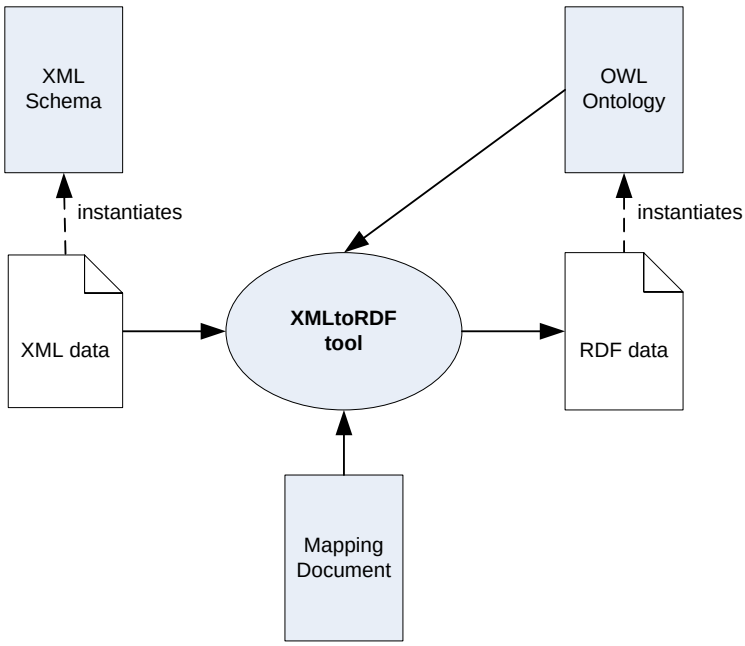


Figure 4.8: Working of the XML to RDF tool.

the resulting RDF document. A generic $XMLtoRDF$ tool takes this mapping document and the used ontology as input and then automatically transforms corresponding XML documents to RDF instances, as shown in Figure 4.8.

The XML-based mapping document is built from specific elements that form a mapping language and can be interpreted by the $XMLtoRDF$ tool. This language allows to create a simple mapping of XML nodes to corresponding OWL classes or properties. Conditional mappings are available in case a mapping not always holds. In that case, a condition can be made of XPATH (XML Path Language) or SPARQL ASK expressions. Finally, value processing is included that specifies different ways to infer the value of a resulting OWL property. These specific language constructs ease the development of such XML to RDF mappings.

4.7 Use Case Scenario

We have covered our metadata model, the layered semantic structure, mapping and rules, and the architectural design of the metadata service. Subsequently, in this section, we present how these technologies could be brought together to create a semantic PCMS. The evaluation of (semantic) multimedia management systems, with respect to retrieval of content, is a difficult task [132]. However, a common approach used in the field is to present a use case scenario and show the enhanced feasibility to solve it with the proposed system [98, 106, 108]. In this sense, it is important to present a general use case scenario, since otherwise a fair comparison cannot be given with related work. Therefore, this section elaborates on a use case scenario consisting of the upload of a resource, importing of the existing metadata fields, adding an annotation according to a different metadata format, and using high-level metadata to retrieve the content.

In the context of video surveillance, this use case scenario is translated to the capturing of video data that is inserted into the surveillance system. Different analysis modules add an annotation to the content (e.g., describing a detected person). During visualization, high-level metadata can be used to retrieve and visualize specific video feeds.

4.7.1 Solving the Use Case Scenario with the Proposed System

When a PeCMan user uploads a multimedia document, the related XML-based metadata is imported. This metadata could be stored in the multimedia file itself (e.g., JPEG), in which case an extractor retrieves the metadata from the file, or the XML fragments could be uploaded separately. Consider that a user has employed a face recognition tool on a picture to create an MPEG-7 annotation as output. The annotation, shown in XML format in Fragment 4.9, states that a specific person (John Smith) is depicted.

When this picture and its MPEG-7 annotation are uploaded to the system, the $XMLtoRDF$ tool is used to create corresponding RDF triples. Fragment 4.10 shows the RDF/N3 notation of an instance of the $PersonType$ class within the MPEG-7 ontology of Garcia et al. [115].

Now consider that a different user utilizes a DIG35 application to annotate a newly created picture in which he is depicted. The application provides the end-user with means to select a region in an image and to add an annotation. When the user selects a region, possible DIG35 metadata fields are

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <SpatialDecomposition>
          <StillRegion>
            <Semantic id="FormalAbstractionDescription">
              <SemanticBase xsi:type="AgentObjectType">
                <Agent xsi:type="PersonType">
                  <Name>
                    <GivenName>John</GivenName>
                    <FamilyName>Smith</FamilyName>
                  </Name>
                </Agent>
              </SemanticBase>
            </Semantic>
          </StillRegion>
        </SpatialDecomposition>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Fragment 4.9: Example of metadata expressed in MPEG-7 format.

```
@prefix mpeg7: <http://rhizomik.upf.edu/ontologies/2005/03/
    MPEG7-2001.owl#>.

<#mpeg7_0> a mpeg7:PersonType;
           mpeg7:Name [mpeg7:GivenName "John", mpeg7:
               FamilyName "Smith"] .
```

Fragment 4.10: RDF/N3 notation of the MPEG-7 description of a person.

presented and an XML annotation is created (e.g., the XML annotation shown in Fragment 4.11). Afterwards, the user uploads the content and metadata to the PCMS, in which the metadata is added to the metadata service.

Within the metadata service, this XML annotation is converted to RDF triples, by the $XMLtoRDF$ tool, and stored for future retrieval (the RDF/N3 annotation is given in Fragment 4.12).

Finally, the PCMS system offers the user the possibility to search for depicted persons through a Web interface. When invoked, the user fills in a name and family name. The Web service interprets this and constructs a SPARQL query solely based on PeCMan metadata as shown in Fragment 4.13. This is indeed a normal query that retrieves instances of the $ImageContentDescription$ class that describe a depicted person with name "John" and family name "Smith". The actual pictures can be retrieved and shown to the user since the metadata service internally stores a digital object identifier that relates the annotations to corresponding multimedia resources.

Since the semantic representations of our PeCMan metadata model and the underlying MPEG-7 and DIG35 standards are linked together, through the ontology mapping and rules as explained in Section 4.4.4, the system retrieves both pictures. For example, following the mapping and rules, the information from the DIG35 annotation of the second picture is linked to PeCMan triples within the RDF triple-store. The $Content$ class of DIG35 is matched with the $ImageContentDescription$ class of the PeCMan ontology and the $depictedItem$ property is deduced from the DIG35 constructs following the rule shown in Fragment 4.8. As a result, the software agent retrieves the two pictures depicting the requested person.

The use case presented here can be extended to deliver only those resources for which a user is granted access, or that are currently stored in some location, etc. The related work focuses on the (semantic) annotation of multimedia content and does not include metadata on the security or system level, hence it cannot enforce such restrictions. However, in the proposed system, all this metadata is part of the same search space, so metadata relevant for the working of a PCMS can be combined with the actual multimedia metadata. For example, to show only those resources that are shared with a user (with id $urn : pecman : someUserID$), the query of Fragment 4.13 is extended to check the sharing settings (see Fragment 4.14). The query finds all resources that are shared and for which the specific user has been given permission to view the resource.

```
<METADATA>
  <CONTENT_DESCRIPTION>
    <PERSON_NAME>
      <NAME_COMP type="Given">John</NAME_COMP>
      <NAME_COMP type="Family">Smith</NAME_COMP>
    </PERON_NAME>
  </CONTENT_DESCRIPTION>
</METADATA>
```

Fragment 4.11: Example of DIG35 metadata in XML format. It expresses that the person John Smith is depicted on the image.

```
@prefix dig35: <../DIG35/V0.2/DIG35.owl#>.

<#person_0> a dig35:Person;
            dig35:givenName "John";
            dig35:familyName "Smith".
<#personDescription_0>  a dig35:PersonDescription;
                        dig35:person <#person_0> .
<#content_0>  a dig35:Content;
              dig35:contentPersonDescription <#
                  personDescription_0> .
<#metadata_0> a dig35:Metadata;
              dig35:imageContent <#content_0> .
```

Fragment 4.12: Example of DIG35 metadata expressed in RDF/XML-format.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pecman: <../PecMan/V1.0/PDocument.owl#>

SELECT ?content
WHERE {   ?content rdf:type pecman:ImageContentDescription.
    ?content pecman:depictedItem ?person .
    ?person rdf:type pecman:Person.
    ?person pecman:name "John" .
    ?person pecman:familyName "Smith"
  }
```

Fragment 4.13: Example of SPARQL query to retrieve.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pecman: <../PecMan/V1.0/PDocument.owl#>

SELECT ?doc
WHERE {   ?doc rdf:type pecman:PecmanDocument.
    ?doc pecman:securityCentric ?security.
    ?security pecman:share ?shareLevel.
    ?shareLevel pecman:shareLevel "Shared".

    ?security pecman:permission ?policy.
    ?policy pecman:user "urn:pecman:someUserID".
    ?policy pecman:permission "View".

    ?doc pecman:userCentric ?userMetadata.
    ?userMetadata pecman:manualAnnotation ?imageMetadata
    ?imageMetadata pecman:imageContent ?content.
    ?content pecman:depictedItem ?person.
    ?person rdf:type pecman:Person.
    ?person pecman:name "John" .
    ?person pecman:familyName "Smith"
  }
```

Fragment 4.14: Example of SPARQL query to retrieve shared content.

4.7.2 Solving the Use Case Scenario with the Related Work

This section discusses whether the related work could be used to implement the use case scenario presented above (without the security restrictions). Note that the use case scenario consists of 2 steps. First resources with metadata, according to two different formats (MPEG-7 and DIG35), are inserted into the system. Secondly, by querying the high-level metadata model that is used within the PCMS, the according resources are retrieved.

Hunter used XSLT transformations to map concepts of different metadata standards on each other [102]. This means that XML fragments according to different metadata schemas need to be mapped upon each other using these transformations for each request. In the proposed system, these mappings are available as an OWL schema by itself. Hence, the relations between the different metadata schemas are part of the search space, so only one query is needed to retrieve the right results.

Cruz et al. use a proprietary mapping table that links an upper ontology to local RDF ontologies [103]. A query upon this upper ontology is therefore first translated, using the mapping table, to different queries on the local RDF ontologies. Moreover, according to their architecture, these queries are translated to XML queries that are executed upon the XML sources. This introduces overhead compared to the proposed system where only one query is executed on the RDF data to retrieve the same results.

Hunter and Little use the ABC ontology as a core ontology [104]. However, the ABC ontology is not suitable to relate different multimedia metadata formats (which describe multimedia and not that much existing concepts or events) to each other. In contrast, our system allows to relate those formats to each other by using the general user centric metadata within the top layer. In fact, the lower layer of our system could include the ABC ontology to link different domain-specific ontologies. These links can be regarded as a mapping ontology, which was described in Section 4.4.4.

If the system of Hunter and Little is applied to find the resources that depict a person with given name, the following steps are needed. First a reasoner is used to find concepts within the participating ontologies that represent a person, or the name of a person. Next, these concepts need to be mapped upon the XML instances. To accomplish this an XPath query is needed to find the according XML element defined in the XML schema. Finally, an additional XPath expression is needed to find the actual value of that element within the XML instance, so that it can be used for retrieval.

Dönderler et al. presented a semantic video database management system [108]. A video annotation tool could be incorporated in their system that takes MPEG-7 or DIG35 metadata and stores this in the database according to the semantic model for future querying. A proprietary textual query language, called BilVideo, has been developed, which allows for example to retrieve video (fragments) that depict a certain person. Such a query is formulated in Fragment 4.15. The query is very simple since it is internally translated on semantic relations, which need to be explicitly defined in the database. Moreover, only relations between instances are possible. As a result, a query that is looking for all persons cannot be used to retrieve specific individuals.

```
1 select video from all
  where appear(somePerson);
```

Fragment 4.15: Example of BilVideo query to retrieve a video that depicts a specific person.

Additionally, since the database has a fixed semantic model to describe multimedia, including domain-specific ontologies would infer a new database structure, which makes the system not practical.

Asirelli et al. presented an infrastructure that they are planning to implement, so details are missing about the actual system [114]. Their system totally relies on MPEG-7 descriptors, stored in an XML database, no ways for translating other metadata formats to MPEG-7 descriptors are presented, which is necessary for retrieval across metadata formats. In this sense, their work resembles that of Petridis et al. [110], meaning they regard MPEG-7 descriptions as low-level features from which high-level semantic concepts are learned. Accordingly, the latter cannot work with other metadata formats.

Garcia et al. use XSD2OWL to automatically create an ontology based on an existing XML schema [115]. However, this tool is made to allow the automatic conversion from MPEG-7 XML schema to an OWL ontology (and only results for MPEG-7 conversions were presented). We noticed that this tool is not usable for the conversion of the DIG35 XML schema. For example, an XSD2OWL translation is used that translates an XML sequence to an intersection of classes (denoted by the $OWL : intersectionOf$ construct). Within the DIG35 schema the element $ContentDescription$ is defined as a sequence of different elements ($Caption$, $Location$, $Person$, $Thing$, $Comment$, etc.). Some of these would be translated to $owl : Class$ constructs, others to $owl : DatatypePropery$ constructs, which would invalidate the intersection.

Moreover, Garcia et al. did not define ways to map different ontologies on each other. By using the XSD2OWL conversions, making a mapping would only be possible if the different metadata schemes use the same names for the same concepts, which is obviously not the case as can be seen from Fragment 4.2 and Fragment 4.3.

Note that the use case scenario presented here is a very generic case study, which consists of importing metadata according to different specific multimedia metadata formats and using queries upon the proposed metadata model to retrieve the content. The actual implementation of this use case can vary; we have presented a use case of retrieving pictures that depict a certain person. Comparable studies can be made to retrieve content based on other metadata fields (finding pictures taken in a specific location, retrieve all pictures created by a specific person, retrieve pictures that correspond to a given set of tags, etc.). However, within the related work the same problems will reoccur, regardless the actual implementation of this use case.

4.8 Conclusions

In this chapter we firstly introduced the usage of metadata in the context of video surveillance systems. The shift towards distributed multi-camera surveillance systems requires a common format for information exchange between the different modules. We identified the issues that occur when using different XML-based metadata standards for this purpose. An overview of the related work in this field showed that Semantic Web technologies, like RDF or OWL, are currently considered for the representation of surveillance metadata. We explained how a surveillance system can be seen as a content management system, in which the content, being the captured images, is annotated conform a metadata standard to represent the detected events.

In the rest of the chapter we introduced the different components to create a semantic metadata service, tailored for personal content management systems. A layered metadata model has been created for use within this service. An upper layer consists of an OWL representation of a PCMS metadata model and allows for system, security, and user management. This ontology is linked to a lower layer, containing a pool of multimedia metadata ontologies that represent commonly accepted metadata formats. The first semantic representation of the DIG35 metadata standard was created to be included in the lower layer. Subsequently, the usage of mapping ontologies and inference rules was introduced to integrate this ontology in the layered metadata model. Additionally, the ar-

chitecture and practical implementation of a semantic metadata service was outlined. Finally, a general use case scenario of retrieving multimedia content based on annotations in different metadata formats was presented. This evaluation showed that the proposed architecture is better suited to deal with the case study than the related work, and that it allows to reduce the interoperability issues inherent to the use of different metadata standards.

In this field our contributions consist of the definition of an image metadata ontology representing the DIG35 standard. This work was input for the photo use case of the W3C Multimedia Semantics Incubator Group (MMSem). Additionally, a general ontology was defined suited for (personal) content management systems. Finally, we created a semantic metadata service. Our work in this area can be found in following publications:

1. Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. Personal Content Management System a Semantic Approach. *Visual Communication and Image Representation*, 47:131 – 144, February 2009
2. Davy Van Deursen, Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. XML to RDF Conversion: a Generic Approach. In *Proceedings of 4th International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS) 2008*, pages 138 – 144, 2008

Our work in the context of MPEG-21 has formed the base for this research. Studying MPEG-21, which goal it is to create an entire multimedia framework, the problems of using an XML-based metadata format became apparant. This work can be found in the following publications:

1. Davy De Schrijver, Chris Poppe, Sam Lerouge, Wesley De Neve, and Rik Van de Walle. MPEG-21 Bitstream Syntax Descriptions for Scalable Video Codecs. *Multimedia Systems*, 11:403–421, June 2006
2. Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Improved Background Mixture Models for Video Surveillance Applications. *Lecture Notes in Computer Science, 8th Asian Conference on Computer Vision(ACCV 2007)*, 4843:251–260, 2007
3. Chris Poppe, Saar De Zutter, Wesley De Neve, and Rik Van de Walle. Reconfigurable Multimedia: Putting the User in the Middle. In *Proceedings of COST298 conference: the Good, the Bad an the Unexpected*, pages 15–30, May 2007

4. M. Ransburg, H. Hellwagner, R. Cazoulat, B. Pellan, C. Concolato, S. De Zutter, C. Poppe, R. Van de Walle, and A Hutter. Dynamic and Distributed Adaptation of Scalable Multimedia Content in a Context-Aware Environment. In *Proceedings of the WiCon'06 conference*, 2006

5. Chris Poppe, Frederik De Keukelaere, Saar De Zutter, and Rik Van de Walle. Advanced Multimedia Systems using MPEG-21 Digital Item Processing. In *Proceedings of the Eight IEEE International Symposium on Multimedia*, pages 785–786, San Diego, December 2006

6. Chris Poppe, Ingo Wolf, Sven Wischnowsky, Saar De Zutter, and Rik Van de Walle. Licensing System in an online MPEG-21 Environment. In *Proceedings of the IADIS international conference WWW/internet 2006*, volume II, pages 315–319, Murcia, October 2006

7. Chris Poppe, Michael Ransburg, Saar De Zutter, and Rik Van de Walle. Interoperable Affective Context Collection using MPEG-21. In *Proceedings of the International Conference on Wireless Mobile and Multimedia Networks Proceedings*, volume II, pages 1603–1606, China, November 2006

8. Saar De Zutter, Frederik De Keukelaere, Chris Poppe, and Rik Van de Walle. Performance Analysis of MPEG-21 Technologies on Mobile Devices. In *Proceedings of SPIE-IST Electronic Imaging, Science and Technology*, volume 6074, page 12, San José, January 2006

Additionally, a number of contributions were made to the actual standardization of MPEG-21.

1. Chris Poppe, Saar De Zutter, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13965, Contribution to Utility Software for ISO/IEC 21000-10 DIP/AMD 1, MPEG-document, October 2006

2. Saar De Zutter, Chris Poppe, Davy De Schrijver, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13979, Update to Reference Software for Conformance to ISO/IEC 21000-10 DIP/AMD 1, MPEG-document, October 2006

3. Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13593, Contribution to Conformance Reference Software for ISO/IEC 21000-10 DIP AMD/1, MPEG-document, July 2006

4. Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13592, Contribution to Conformance for ISO/IEC 21000-10 DIP AMD/1, MPEG-document, July 2006

5. Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m12218, Contribution to WD of Amendment on C++ Binding, MPEG-document, July 2005

Using XML-based metadata standards lead to the problem of not being able to specifically define the semantics. Hence, our research shifted to the Semantic Web technologies. Future work in this area consists of an entire implementation of the semantic architecture, as described above, which would allow to make an exhaustive evalution of the system in terms of processing speed, memory consumption, scalability, and so on. Especially when employing such an architecture in a video surveillance system, the performance in terms of speed is of great importance. The Semantic Web technologies are still under development by the W3C and current research focuses on identification of image or video fragments, rule languages, refining OWL, etc. [11]. The ongoing activities will eventually increase the adoption and tool support of the Semantic Web technologies, which will benefit our work in this domain.

Chapter 5

Conclusions

All men by nature desire knwoledge.
– Aristotle.

The last decade, we have witnessed an explosion of video surveillance systems. Surveillance cameras are deployed in- and outdoor, in homes, public buildings and large industrial sites. It has been shown that human operators have a hard time to monitor several surveillance feeds over a longer period. Hence, the need for intelligent video analytics arose and together with the increase of cameras came the raise of efforts in the domain of automatic video analysis. This new branch within the computer vision research field has soon become a large research topic, steered by industrial and governmental funds.

Much research has been done on all aspects of an intelligent video surveillance system. Algorithms were developed to detect and segment moving objects like people and cars. Tracking algorithms find the temporal dependencies between objects in different frames or camera views. Trajectories and body shape are used to determine and predict the behaviour of detected objects. Face and licence plate recognition are examples of more advanced techniques working on the segmented images.

Almost all of the aforementioned use cases require a **fast** and**accurate** initial segmentation of moving objects. Although this is the first step, it is also one of the most difficult ones due to it's dependence on the statistics of the captured video data. The major problems in accurately detecting and segmenting moving objects are noise, shadows, lighting changes, background movement and so on. Additionally, occlusion, differences in size, speed, and shape of the moving objects, stopped objects or moved background objects make it difficult

to create a reliable moving object detection system.

When object detection is incorporated in large-scale video surveillance systems, means are necessary to define how the objects, or other events of interest, can be signalled, communicated and stored. The usage of standardized (XML-based) metadata has been proposed to create interoperability within one system. However, when trying to combine different surveillance systems, or modules of different vendors, the usage of different XML-based standards introduces again interoperability issues. Different standards generally use different constructs to represent the same concepts.

Our research is situated within the context of an intelligent surveillance system, where we focus on moving object detection and the usage of different XML-based metadata standards. In this dissertation, we presented two novel background subtraction techniques to detect moving objects when using a static camera. The first technique works on uncompressed video data, while the second analyses video sequences compressed according to a recent video standard (H.264/AVC). In the latter, we use features available in the H.264/AVC-compressed video stream to detect the moving objects. Lastly, we investigated the problems that occur when trying to describe these detected objects by using standardized XML-based metadata formats.

In Chapter 2, we presented a robust moving object detection technique. The proposed system is a multi-modal spatio-temporal background subtraction technique that tries to find pixels corresponding to moving objects. The temporal background subtraction system uses a weighted mixture of models and is combined with a fast spatial image segmentation technique. The models consist of an average, an upper and lower threshold, a maximum difference with the last background value, and an illumination allowance based on photon noise statistics of the image. The problem of gradual illumination changes is well-known and a number of methods exist in the literature to deal with it, but at the cost of high additional complexity. By incorporating the difference with the last background value, we presented a very fast and efficient technique to deal with the problem of gradual illumination changes. New pixel values are compared with the models to find a match. Consequently a decision (classification as foreground or background) is made based on the weights of the models. The parameters of the models are continuously updated with new information to follow the dynamics of the surveilled environment. Finally, a shadow removal scheme has been included to improve the detection results.

The processing speed of video analytics algorithms is very important, especially in the context of a video surveillance system. Therefore, we presented

an intelligent analysis mask that allows to reduce the number of pixels that need to be analysed for each frame. Additionally, a fast image segmentation is introduced that applies edge detection to find segments in the images. Photon noise statistics are incorporated to remove noisy edges and holes in the edges are filled. Lastly, this segmentation is combined with the background subtraction.

We have presented an extensive evaluation of our system and compared it with related work in the field. Additionally, we have shown both objective and subjective results and have shown that our system can be used for real-time applications.

Most video surveillance systems incorporate an aspect of digital video and rely on advanced video coding formats for the efficient representation, transmission, and/or storage of digital video content. Hence a decoding step is needed, before algorithms for video analytics can be applied. Many research efforts have been done to apply those algorithms directly on the compressed video sequences. As such, not relying on the pixel information, but on the features created during the encoding process. In this dissertation, we focused on the H.264/AVC specification for digital video coding, which has become the de facto benchmark in the domain of digital video coding with respect to compression.

H.264/AVC is a rather new video standard, but the first cameras supporting this video format have already hit the surveillance market. A number of approaches exist to detect moving objects in the H.264/AVC compressed domain. One common aspect of these systems is the usage of motion vector information available in the compressed bitstreams. However, motion vectors are created from a coding point of view and, hence, do not necessarily represent the actual motion in a video sequence. As a result, these algorithms spend a large amount of processing to remove noisy motion vectors.

In Chapter 3, we presented our moving object detection technique, working in the H.264/AVC compressed domain, which disregards the motion vector information entirely. The proposed system relies on the degree of compression that an encoder can achieve for different parts of the images. We made the assumption that an encoder can compress parts of the background well, while it has more difficulties to compress parts of moving objects. To show that our assumption holds, we presented an analysis of the bit usage for different macroblocks in the compressed bitstream. These observations were used to create a compressed domain technique that finds macroblocks assumed to correspond with moving objects. During execution, a background model is created based

on the sizes (in bits) of the macroblocks. For new images, the sizes of the macroblocks are compared with this model to detect unusual large sizes. The corresponding macroblocks are assumed to correspond to moving objects and are spatially and temporally filtered. However, the usage of macroblocks can be too coarse, since these consist of 16×16 pixels. Consequently, we refined the algorithm up to blocks of 4×4 pixels by analysing sub-macroblocks of the detected macroblocks that lie on edges of moving objects.

As in Chapter 2, an extensive comparison was made with related work in this field. We have shown that our system outperforms other moving object detection techniques that rely on motion vector information. Different challenging surveillance sequences were used for this comparison. Additionally, the features being used, namely the amount of compression for blocks in a frame, can be extracted from a compressed video very fast. As a result, we largely outperform the related work in terms of execution speeds. We can detect moving objects up to 20 times faster than the related work in this domain.

Lastly, we evaluated the influence of the encoder configuration on the detection performance of the algorithm. This is a study that is neglected by most of the related work in this domain. We showed that different encoder configurations, resulting in different compressed video streams, have an influence on the performance of the algorithm. However, it is shown that the proposed system is less dependent on different configurations compared to the motion vector-based approaches.

Future work in the context of the detection of moving objects would exist of analysing the usability of the presented algorithms on larger datasets. It has been shown that, compared to related work, our algorithms perform well on different sequences. However, the actual performance is still very dependent on the actual sequence. Moreover, the tests are generally performed on short sequences, which are assumed to be representative for real surveillance scenarios. Such short sequences are indeed usefull for comparisons with other techniques and are commonly used in the domain. Nevertheless, it would be interesting to see how the algorithms perform when monitoring an environment for several hours or days.

The presented algorithms detect pixels that are assumed to correspond to moving objects. To create an entire video surveillance system, other modules need to be added to group the detected pixels into objects. Additionally, tracking of these objects is indispensable for most surveillance applications. This is out of the scope of this dissertation, however it is our belief that these modules could be used to further enhance the detection of objects. If detected objects are

tracked over consecutive frames, information can be fused to improve the accuracy of the detection. Moreover, feedback of the user, meaning the operator of the surveillance system, could also be used to improve the detection algorithms. This feedback, on all levels of the surveillance system, could be used to increase the detection and to adjust the initial parameters and thresholds. It is our belief that research on which feedback is usefull and how it can be used to fine-tune the system would be of great value for the video surveillance community.

Future work, more specifically for the compressed-domain object detection, would exist of extending the algorithm to include other profiles of the H.264/AVC specification. Additionally, it would be interesting to see how our algorithm would perform on older video formats (MPEG-2 or MPEG-4) and more recent ones (SVC).

Finally, the combination of pixel-domain and compressed-domain approaches would also be an interesting research topic. The fast detection in the compressed domain might be used as input of techniques working in the pixel domain. As such, the block-based detection can be refined to the pixel level.

A last aspect that we have covered in this dissertation is how to deal with interoperability issues occurring when using different XML-based metadata standards. In Chapter 4, we briefly discussed how this problem arises in the context of video surveillance systems. However, the main topic of the chapter is how to solve those issues when applied to personal content management systems. We introduced how these problems are manifested in the context of personal content management. In order to deal with these issues, we presented a novel layered architecture based on Semantic Web technologies. Different metadata standards are combined in the lower layer, while the higher layer consists of specific ontologies representing metadata that is relevant for personal content management systems. The metadata standards and relations between them are represented as OWL ontologies and we have created the first formal representation (using OWL) of the DIG35 image metadata standard. We showed how it is incorporated in the framework and how it can be combined with other metadata standards, like MPEG-7. For evaluation purposes, we presented a general use case scenario consisting of the upload of a resource, importing of the existing metadata fields, adding an annotation according to a different metadata format, and using high-level metadata to retrieve the content. We showed how our proposed system is more suitable to solve this use case than related work in the field.

As explained in Chapter 4, future work in this area consists of an entire imple-

mentation of the semantic architecture, which would allow to make an exhaustive evalution of the system in terms of processing speed, memory consumption, scalability, and so on. When looking at video surveillance systems, the practical aspects of such a semantic architecture need to be revised, since the performance in terms of speed is of greater importance.

To conclude, we have presented novel techniques to detect moving objects, both in pixel and H.264/AVC-compressed domain, at high speeds and with good accuracy. Finally, we introduced the usage of Semantic Web technologies to deal with interoperability issues when using different XML-based metadata standards and presented the different benefits these have.

We hope to have convinced the reader that our work is relevant for current and future multimedia systems and can be applied in a wide range of applications, being video surveillance or content management. It is our belief that H.264/AVC will become the dominating video codec in video surveillance systems in the coming years. Additionally, the research in Semantic Web technologies is only just starting to bloom and a wide-spread adoption and tool support can be predicted in the near future. As such, we believe that the value and applicability of our work will only increase in the coming years.

Appendix A

Overview of the H.264/AVC standard

With the standardization of H.264/AVC by MPEG and VCEG, a new video codec was introduced to the video surveillance market. This appendix presents a number of concepts of H.264/AVC that are important in the context of Chapter 3.

Pictures

An H.264/AVC encoded video sequence consists of a sequence of coded pictures. In this chapter we assume the coded picture to represent an entire (progressive) frame. H.264/AVC uses default the YCbCr (Y for luma, representing the brightness, Cb and Cr for chroma) color space and applies 4:2:0 sampling with 8 bits of precision per sample. This means that the chroma components have one fourth of the number of samples of the luma component, which is in accordance with the fact that the human visual system is more sensitive to luma then to chroma. The picture is partitioned into macroblocks (MBs) each covering a rectangular area of 16×16 samples of the luma component and 8×8 samples of each of the two chroma components.

Slices

Slices are sequences of macroblocks which are processed in raster scan order. A picture may be split into one or several slices as shown in Figure A.1 and is therefore a collection of one or more slices in H.264/AVC. Slices are

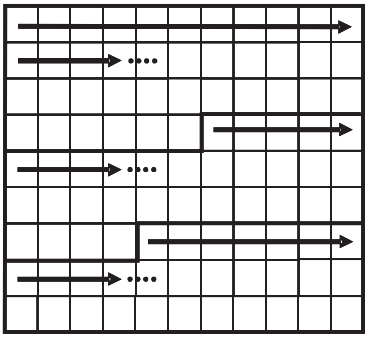


Figure A.1: The concept of slices.

self-contained in the sense that their syntax elements can be parsed from the bitstream and the values of the samples in the area of the picture that the slice represents can be correctly decoded without use of data from other slices. All luma and chroma samples of a MB are either spatially (intra) or temporally (inter) predicted, and the resulting prediction residual is transform coded. The ways a MB can be coded depends on the coding type of the slice. A slice can be coded using different coding types of which the following are the most common:

I slice A slice in which all macroblocks are coded using intra prediction.

P slice These slices include the coding types of the I slices, alternatively, the MBs can be temporally predicted. In this case inter prediction is applied with only one reference picture.

B slice These slices can contain the coding types of the P slices and some MBs can be inter predicted using one or two reference pictures.

Figure A.2 shows a simplified illustration of the syntax of a slice within a H.264/AVC encoded bitstream. The slice consists of a header and a series of coded MBs. A coded MB partly consists of syntax elements (representing e.g., the type, partitioning, prediction modes, and information regarding the MVs). The rest of the MB contains the coded transform coefficients after prediction and compensation.

In this chapter, we assume that a picture consists of only one slice, containing the entire picture. Hence we will use the term I, P, and B pictures.

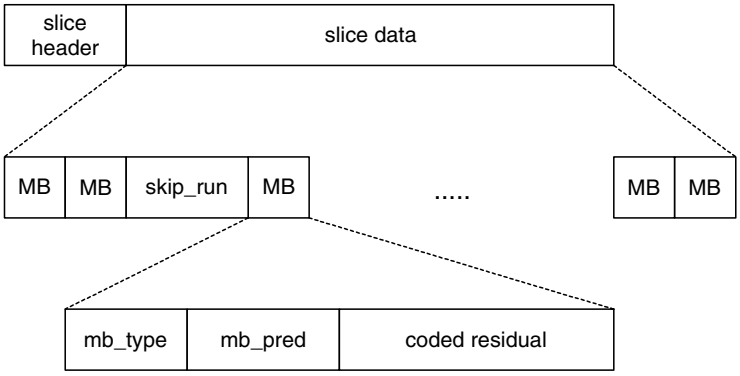


Figure A.2: Syntax of a slice.

When an I picture is lost, e.g., due to transmission errors, the reconstructed video quality is very bad as all the subsequent frames are encoded depending on the I picture. To reduce the propagated error, it is possible to randomly insert intra-coded MBs in the bitstream. (In the H.264/AVC reference software this is denoted by an encoding parameter called $RandomIntraMBRefresh$). In this dissertation we neglect the occurrence of errors or frame drops in the video stream. As such, $RandomIntraMBRefresh$ is set to 0, meaning no random Intra-coded MBs are inserted in the bitstream.

Intra Prediction

Intra prediction, in H.264/AVC, is conducted in the spatial domain. To predict a block, neighbouring samples of previously-coded blocks which are to the left and/or above the block, are used.

Two main types of Intra coding exist: Intra-4×4 and Intra-16×16. When Intra-4×4 is used, each 4×4 block of luminance samples of a MB is predicted separately using one of nine prediction modes. This mode is well suited for coding parts of pictures with much details. For Intra-16×16, four uniform prediction modes are defined to predict the luma component of a whole MB. Homogeneous areas of a picture are best coded with this mode. Figure A.3 shows the first four Intra-4×4 prediction modes.

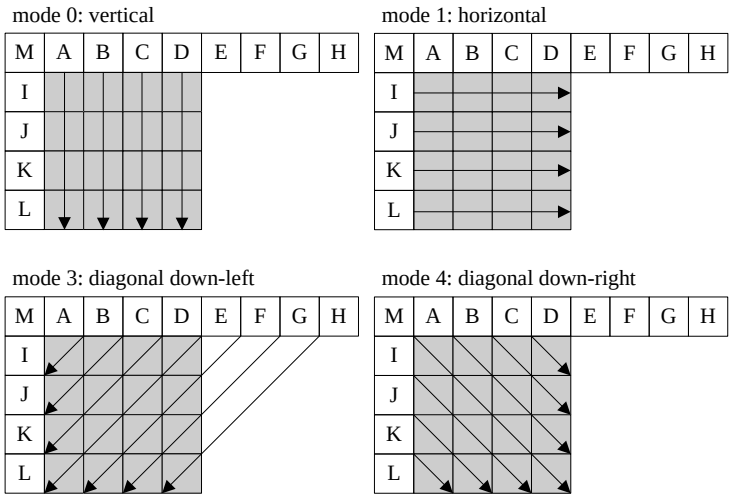


Figure A.3: Four of the nine Intra-4×4 prediction modes in H.264/AVC.

Inter Prediction

MBs which are inter predicted are assigned a MB type. This type denotes a specific partitioning of the 16×16 block. The partitions can have a size of 16×16, 16×8, 8×16, or 8×8 pixels. If the latter partitioning is chosen the 8×8 blocks can be further partitioned in partitions of 8×4, 4×8, or 4×4 pixels. For each partition a motion vector (MV) is created to denote the best match of the current partition with a block of corresponding size in a reference picture. The MV components are differentially coded using either median or directional prediction from neighbouring blocks.

A MB can also be coded as a skipped MB, called a P_Skip MB. In this case no prediction residual is coded, nor a MV or reference picture is signalled. The MV, used for reconstructing such a P_Skip MB, is calculated based on the MVs of neighbouring blocks. These types of MBs are very useful when large areas of no change between pictures occur in the sequence, since this can be coded with very few bits.

Transform and Quantization

H.264/AVC applies transform coding to the prediction residual. The transform coding happens on 4×4 blocks and consists of a separable integer transform.

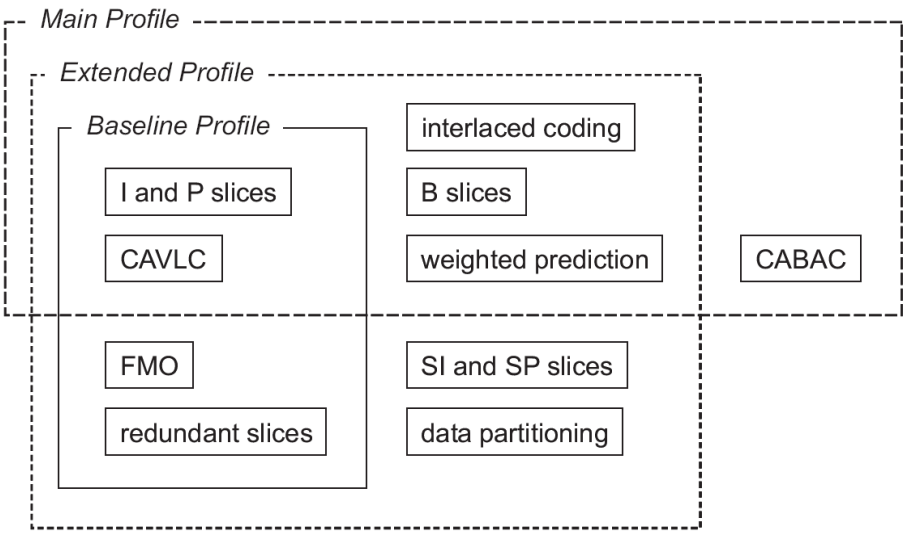


Figure A.4: Profiles in H.264/AVC.

After the transformation, H.264/AVC applies zig-zag scanning, scaling and rounding as a quantization step. The latter is determined by a quantization parameter (QP), which influences the amount of compression.

Entropy Coding

For transmitting the quantized transform coefficients, two entropy coding methods are defined: Context-Adaptive Variable Length Coding (CAVLC) and Context-based Adaptive Binary Arithmetic Coding (CABAC). Both methods use statistics about previously coded symbols as a basis for the prediction for the current symbol (context).

Profiles

As in most other specifications for digital video coding, H.264/AVC uses the concept of profiles and levels. A profile defines a set of coding tools or algorithms that may be used to generate a compliant bitstream whereas a level imposes constraints on certain key parameters such as bit rate, resolution, or frame rate. Three profiles were defined in the first version of the H.264/AVC specification, each targeting a different range of applications.

Baseline Profile The Baseline Profile tries to minimize the complexity of the coding tools while including many tools for robustness as it is intended to be used in a broad range of transmission networks where the quality of service is not guaranteed. It targets wireless communication and low-delay (interactive) video applications such as video conferencing.

Main Profile This profile focuses on high compression efficiency. The target applications are in the domain of entertainment such as digital video broadcast and storage (e.g., on next-generation DVDs).

Extended Profile The Extended Profile is defined to be used in streaming scenarios as it adds tools for an enhanced coding efficiency and for better network robustness to the set of tools of the Baseline Profile. In fact, the Extended Profile is a superset of the Baseline Profile.

The relationship between these three profiles and the coding tools of H.264/AVC is shown in Figure A.4.

The first amendment of H.264/AVC (FRExt) defines four additional profiles which are all supersets of the Main Profile. These four profiles are the High Profile, the High 10 Profile, the High 4:2:2 Profile, and the High 4:4:4 Profile. More information about these profiles and the tools they include can be found in [133, 134].

Appendix B

Encoder settings

This appendix covers the parameter settings of the codecs used in the tests, as described in Chapter 3.

H.264/AVC JM 12.4

The parameter values, as processed by the H.264/AVC JM 12.4 encoder, are described in Table B.1 and Table B.2 . Zero-valued parameters are omitted.

Table B.1: Parameter settings for the H.264/AVC JM 12.4 encoder.

Parameter	Value
InputFile	"../PetsD2TeC2.yuv"
FramesToBeEncoded	2800
FrameRate	30
SourceWidth	384
SourceHeight	288
TraceFile	"trace_enc.txt"
ReconFile	"PetsD2TeC2_rec.yuv"
OutputFile	"PetsD2TeC2_cbr100.264"
ProfileIDC	66
LevelIDC	40
IntraPeriod	16

Table B.2: Parameter settings for the H.264/AVC JM 12.4 encoder.

Parameter	Value
AdaptiveIntraPeriod	1
AdaptiveIDRPeriod	1
QPISlice	28
QPPSlice	28
SearchRange	32
MEDistortionHPel	2
MEDistortionQPel	2
MDDistortion	2
ChromaMCBuffer	1
NumberReferenceFrames	5
Log2MaxPOCLsbMinus4	-1
InterSearch16x16	1
InterSearch16x8	1
InterSearch8x16	1
InterSearch8x8	1
InterSearch8x4	1
InterSearch4x8	1
InterSearch4x4	1
EnableIPCM	1
RDPSliceWeightOnly	1
RestrictSearchRange	2
RDOptimization	1
FastCrIntraDecision	1
RateControlEnable	1
Bitrate	100000
BasicUnit	12
AdaptiveRounding	1
SearchMode	1
UMHexDSR	1
UMHexScale	3

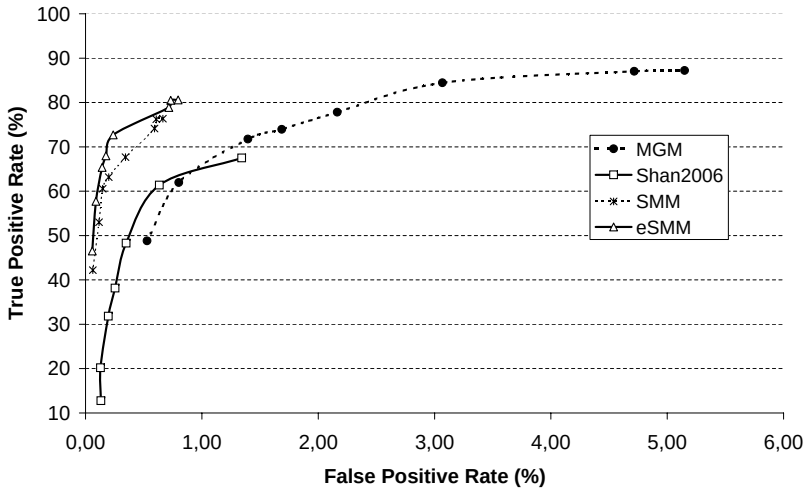
Appendix C

ROC vs. Precision-Recall

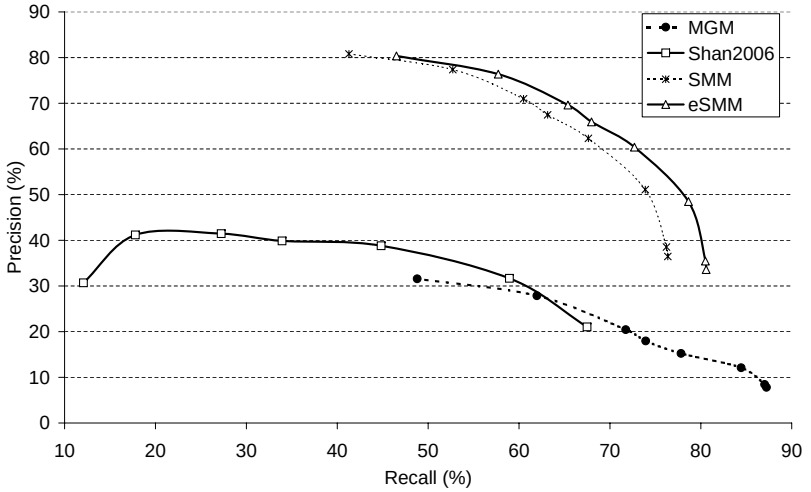
In machine learning, current research has shifted away from simply presenting accuracy results when performing an empirical validation of new algorithms. This is especially true when evaluating algorithms that output probabilities of class values. Provost et al. have argued that simply using accuracy results can be misleading, as one can often use thresholds on the output of an algorithm to alter its performance [135]. They recommended using Receiver Operator Characteristic (ROC) curves when evaluating binary decision problems. However, ROC curves can present an overly optimistic view of an algorithm's performance if there is a large skew in the class distribution. Precision-Recall curves, often used in Information Retrieval [136], have been cited as an alternative to ROC curves for tasks with a large skew in the class distribution [137, 138]. An important difference between ROC space and PR space is the visual representation of the curves.

When looking at video surveilance systems, more specifically at object detection techniques, the number of negative examples (the pixels representing background regions) greatly exceeds the number of positives instances (the pixels representing foreground objects). Consequently, a change in the number of false positives generally only leads to a small change in the false positive rate used in ROC analysis. Precision, on the other hand, uses false positives together with true positives rather than true negatives. As such, a change in the number of false positives has a larger effect on the performance results of the algorithms.

Figure C.1 repeats the ROC graphs of Figure 2.9(a) and, additionally, shows the Precision-Recall curves for the same values.



(a)



(b)

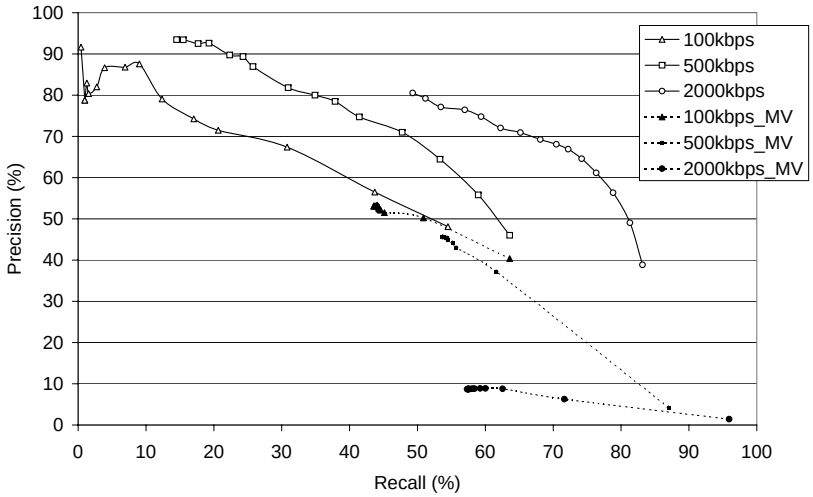
Figure C.1: Comparison of ROC (a) and Precision-Recall (b) graph of the output of pixel-domain algorithms for the PetsD2TeC2 sequence.

From this figure it is clear that the relations between the different graphs remain the same. If one technique works better than another it will be visible in both ROC and Precision-Recall curves. The superior working of one technique over another can be seen by looking at the dominance of one curve to another curve. We use the definition for the term dominance of Provost et al. [135], saying that one curve dominates another if the latter curve is always equal to or below the former curve. With this definition Davis and Goadrich evaluated the relationship between ROC and Precision-Recall graphs [139] and they have proven the following theorem.

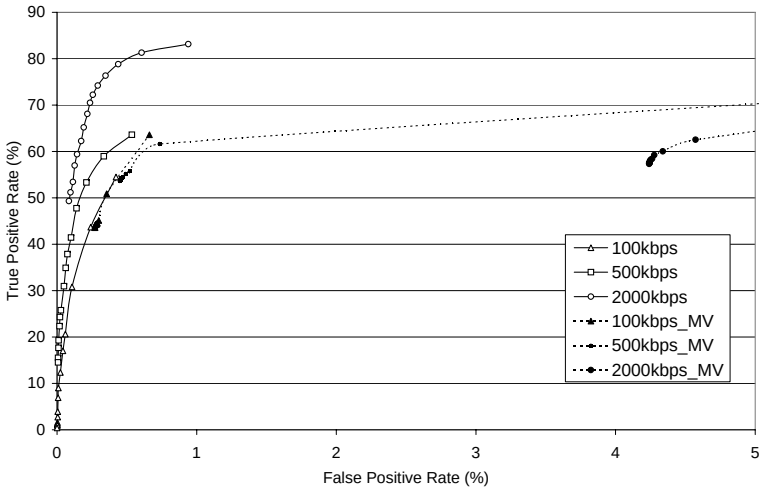
Theorem *Curve A dominates curve B in ROC space if and only if curve A dominates curve B in precision recall space.*

This can also be seen in Figure C.1 which repeats the Precision-Recall curves of Figure 3.12(a) and, additionally, shows the ROC graphs for the same values.

The performances of the algorithms shown in the Precision-Recall graphs appear to be comparable in ROC space. However, in Precision-Recall space we can see more variations in the precision values than in the False Positive Rate in ROC space. The latter is determined according to a fixed number of real background pixels, hence less variation appears for different thresholds. Nevertheless, this does not stop us to make decisions about how one algorithm works better than the other. Here we have shown that the usage of ROC curves and Precision-Recall graphs leads to the same results when comparing different algorithms to eachother. Our study of the scientific work in pixel-domain object detection showed that the usage of ROC curves is omni-present in that domain. Surprisingly, when looking at compressed-domain object detection techniques, it was noticed that Precision-Recall curves are preferred. Hence, in Chapter 1 we use ROC curves for the evaluation of our pixel-domain algorithm, whereas, in Chapter 3, precision and recall are used. It is our believe that the usage of according evalution methods in each domain will encourage other authors to compare their system with ours.



(a)



(b)

Figure C.2: Comparison of Precision-Recall (a) and ROC (b) graph for fixed bitrate versions of (a) the PetsD2TeC2 sequence and (b) the Indoor sequence.

Appendix D

PeCMan OWL schema

This appendix presents the PeCMan ontology. This ontology has been created as a collection of OWL schemas. Only a number of schemas are incorporated in this appendix, due to the size of the ontology, the ontology exists of 25 OWL schemas, each containing numerous classes and properties.

The OWL schema denoting a PeCMan document is given in Listing D.1. The way that $ImageMetadata$ is related to the PeCMan document can be found in Listing D.2 and Listing D.3. The latter shows that the $ImageMetadata$ class is a subclass of $ManualAnnotation$ as explained in Chapter 4. As can be seen in the listings, different concepts of our metadata model are represented as OWL classes and the relationships between them are represented as OWL $ObjectProperties$.

Fragment D.1: OWL schema describing the PeCMan document.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF[
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY pcms "http://multimedialab.elis.ugent.be/users/gmartens/
    Ontologies/Pecman/v1.0/">
<!ENTITY ucentric "&pcms;UserCentric.owl#">
<!ENTITY syscentric "&pcms;SystemCentric.owl#">
<!ENTITY seccentric "&pcms;SecurityCentric.owl#">
]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="&xsd;"
  xmlns:rdfs="&rdfs;"
  xmlns:owl="&owl;"
  xmlns="&pcms;PDocument.owl#"
```

```
xml:base="&pcms;PDocument.owl"
xmlns:ucentric="&ucentric;"
xmlns:syscentric="&syscentric;"
xmlns:seccentric="&seccentric;">

<owl:Ontology rdf:about="">
  <rdfs:comment xml:lang="en">
    Ontology for describing the metadata related to
    a PeCMan Document
  </rdfs:comment>
  <owl:imports rdf:resource="&ucentric;"/>
  <owl:imports rdf:resource="&syscentric;"/>
  <owl:imports rdf:resource="&seccentric;"/>
</owl:Ontology>

<!-- ***********************************************-->
<!-- Definition of the PecmanDocument class -->
<!-- ***********************************************-->

<owl:Class rdf:ID="PecmanDocument">
  <rdfs:label>Pecman Document</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#userCentric"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        1
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#systemCentric"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        1
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#securityCentric"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        1
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- *******************************************************-->
<!-- Property definitions of the PecmanDocument class -->
<!-- *******************************************************-->

<owl:ObjectProperty rdf:ID="userCentric">
  <rdfs:domain rdf:resource="#PecmanDocument"/>
  <rdfs:range rdf:resource="&ucentric;UserCentric"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="systemCentric">
```

```
      <rdfs:domain rdf:resource="#PecmanDocument"/>
      <rdfs:range rdf:resource="&syscentric;SystemCentric"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="securityCentric">
      <rdfs:domain rdf:resource="#PecmanDocument"/>
      <rdfs:range rdf:resource="&seccentric;SecurityCentric"/>
    </owl:ObjectProperty>
  </rdf:RDF>
```

Fragment D.2: OWL schema describing user-centric metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE rdf:RDF[
 <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
 <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
 <!ENTITY owl "http://www.w3.org/2002/07/owl#">
 <!ENTITY pecm "http://multimedialab.elis.ugent.be/users/gmartens/
     Ontologies/Pecman/v1.0/">
 <!ENTITY log "&pcms;Log.owl#">
 <!ENTITY annot "&pcms;Annotation.owl#">
 <!ENTITY pdoc "&pcms;PDocument.owl#">
 ]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="&xsd;"
    xmlns:rdfs="&rdfs;"
    xmlns:owl="&owl;"
    xmlns="&pcms;UserCentric.owl#"
    xml:base="&pcms;UserCentric.owl"
    xmlns:log="&log;"
    xmlns:annot="&annot;"
    xmlns:pdoc="&pdoc;">

    <owl:Ontology rdf:about="">
        <rdfs:comment xml:lang="en"> Pecman Ontology for describing
             user centric metadata </rdfs:comment>
        <owl:versionInfo rdf:datatype="&xsd;string">version 0.1</
            owl:versionInfo>
        <owl:imports rdf:resource="&log;"/>
        <owl:imports rdf:resource="&annot;"/>
        <owl:imports rdf:resource="&pdoc;"/>
    </owl:Ontology>

    <!-- ************************************-->
    <!-- Definition of the UserCentric class -->
    <!-- ************************************-->

    <owl:Class rdf:ID="UserCentric">
        <rdfs:label>User centric</rdfs:label>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#thumbnail"/>
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger">1</owl:maxCardinality>
```

```
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#manualAnnotation"/>
                <owl:cardinality rdf:datatype="&xsd;
                    nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- ********************************************** -->
    <!-- Property definitions of the UserCentric class -->
    <!-- ********************************************** -->

    <owl:ObjectProperty rdf:ID="thumbnail">
        <rdfs:domain rdf:resource="#UserCentric"/>
        <rdfs:range rdf:resource="&pdoc;PecmanDocument"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="manualAnnotation">
        <rdfs:domain rdf:resource="#UserCentric"/>
        <rdfs:range rdf:resource="&annot;ManualAnnotation"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="automaticAnnotation">
        <rdfs:domain rdf:resource="#UserCentric"/>
        <rdfs:range rdf:resource="&annot;AutomaticAnnotation"/>
    </owl:ObjectProperty>
</rdf:RDF>
```

Fragment D.3: OWL schema describing image metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF[
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY pcms "http://multimedialab.elis.ugent.be/users/gmartens/
    Ontologies/Pecman/v1.0/">
<!ENTITY basic "&pcms;BasicImageParam.owl#" >
<!ENTITY creat "&pcms;ImageCreation.owl#" >
<!ENTITY cont "&pcms;Content.owl#">
<!ENTITY annot "&pcms;Annotation.owl#">
]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="&xsd;"
    xmlns:rdfs="&rdfs;"
    xmlns:owl="&owl;"
    xmlns:basic="&basic;"
    xmlns:creat="&creat;"
    xmlns:cont="&cont;"
    xmlns:annot="&annot;"
    xmlns="&pcms;ImageMetadata.owl#"
```

```
    xml:base="&pcms;ImageMetadata.owl">

    <owl:Ontology rdf:about="">
        <rdfs:comment xml:lang="en"> Pecman Ontology for describing
             metadata for digital images </rdfs:comment>
        <owl:versionInfo rdf:datatype="&xsd;string">version 0.2</
            owl:versionInfo>
        <owl:imports rdf:resource="&basic;"/>
        <owl:imports rdf:resource="&creat;"/>
        <owl:imports rdf:resource="&cont;"/>
        <owl:imports rdf:resource="&annot;"/>
    </owl:Ontology>

    <!-- ************************************** -->
    <!-- Definition of the ImageMetadata class -->
    <!-- ************************************** -->

    <owl:Class rdf:ID="ImageMetadata">
        <rdfs:subClassOf rdf:resource="&annot;ManualAnnotation"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#basicParam"/>
                <owl:cardinality rdf:datatype="&xsd;
                    nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#imageCreation"/>
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger">1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- ****************************************************-->
    <!-- Property definitions of the ImageMetadata class -->
    <!-- ****************************************************-->

    <owl:ObjectProperty rdf:ID="basicParam">
        <rdfs:domain rdf:resource="#ImageMetadata"/>
        <rdfs:range rdf:resource="&basic;BasicParam"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="imageCreation">
        <rdfs:domain rdf:resource="#ImageMetadata"/>
        <rdfs:range rdf:resource="&creat;ImageCreation"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="imageContent">
        <rdfs:domain rdf:resource="#ImageMetadata"/>
        <rdfs:range rdf:resource="&cont;ImageContentDescription"/>
    </owl:ObjectProperty>
</rdf:RDF>
```

Listing D.4 holds the OWL schema describing basic image parameters. This listing shows the usage of OWL $DataTypeProperties$ for image parameters like the coding format and image width or height.

Fragment D.4: OWL schema holding basic image parameters.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF[
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY pcms "http://multimedialab.elis.ugent.be/users/gmartens/
    Ontologies/Pecman/v1.0/">
]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="&xsd;"
    xmlns:rdfs="&rdfs;"
    xmlns:owl="&owl;"
    xml:base="&pcms;BasicImageParam.owl"
    xmlns="&pcms;BasicImageParam.owl#">

    <owl:Ontology rdf:about="">
        <rdfs:comment xml:lang="en"> PecMan Basic Image Parameter
            Metadata </rdfs:comment>
        <owl:versionInfo rdf:datatype="&xsd;string">version 0.1</
            owl:versionInfo>
    </owl:Ontology>

    <!-- ************************************** -->
    <!-- Definition of the BasicParam class -->
    <!-- ************************************** -->

    <owl:Class rdf:ID="BasicParam">
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#fileName"/>
                <owl:cardinality rdf:datatype="&xsd;
                    nonNegativeInteger">
                    1
                </owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#codingFormat"/>
                <owl:cardinality rdf:datatype="&xsd;
                    nonNegativeInteger">
                    1
                </owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#imageWidth"/>
```

```
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger"
                >1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#imageHeight"/>
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger"
                >1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- ****************************************************-->
    <!-- Property definitions of the BasicParam class -->
    <!-- ****************************************************-->

    <!-- File and coding format-->
    <owl:DatatypeProperty rdf:ID="fileName">
        <rdfs:domain rdf:resource="#BasicParam"/>
        <rdfs:range rdf:resource="&xsd;anyURI"/>
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="codingFormat">
        <rdfs:domain rdf:resource="#BasicParam"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- image size  properties-->
    <owl:DatatypeProperty rdf:ID="imageWidth">
        <rdfs:comment>image width in pixels</rdfs:comment>
        <rdfs:domain rdf:resource="#BasicParam"/>
        <rdfs:range rdf:resource="&xsd;positiveInteger"/>
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="imageHeight">
        <rdfs:comment>image height in pixels</rdfs:comment>
        <rdfs:domain rdf:resource="#BasicParam"/>
        <rdfs:range rdf:resource="&xsd;positiveInteger"/>
    </owl:DatatypeProperty>
</rdf:RDF>
```

Classes and constructs for describing the content of a image, meaning what is depicted on the picture, are contained in the $Content$ OWL schema, as shown in Listing D.5. A $Person$ is a subclass of the $Thing$ class, so through the $depictedItem$ property we can describe a person that is depicted on an image. The definition of the $Person$ class can be seen in Listing D.6.

Fragment D.5: OWL schema holding content-related metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE rdf:RDF[
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY pcms "http://multimedialab.elis.ugent.be/users/gmartens/
    Ontologies/Pecman/v1.0/">
<!ENTITY ev "&pcms;Event.owl#" >
<!ENTITY tang "&pcms;TangibleThing.owl#">
<!ENTITY log  "&pcms;Log.owl#">
<!ENTITY rat  "&pcms;Rating.owl#">
<!ENTITY pos  "&pcms;Position.owl#">
]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="&xsd;"
    xmlns:rdfs="&rdfs;"
    xmlns:owl="&owl;"
    xmlns="&pcms;Content.owl#"
    xml:base="&pcms;Content.owl"
    xmlns:ev="&ev;"
    xmlns:tang="&tang;"
    xmlns:log="&log;"
    xmlns:rat="&rat;"
    xmlns:pos="&pos;">

    <owl:Ontology rdf:about="">
        <rdfs:comment xml:lang="en"> Pecman: Ontology for
            describing image content </rdfs:comment>
        <owl:imports rdf:resource="&ev;"/>
        <owl:imports rdf:resource="&pos;"/>
        <owl:imports rdf:resource="&tang;"/>
        <owl:imports rdf:resource="&log;"/>
        <owl:imports rdf:resource="&rat;"/>
    </owl:Ontology>

    <!-- ***************************************************** -->
    <!-- Definition of the ImageContentDescription class   -->
    <!-- ***************************************************** -->

    <owl:Class rdf:ID="ImageContentDescription">
        <rdfs:label>Image content description</rdfs:label>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#position"/>
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger">
                    1
                </owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!--
        ***************************************************************
         -->
    <!-- Property definitions of the ImageContentDescription class
        -->
```

```
    <!--
        ************************************************************
         -->

    <owl:DatatypeProperty rdf:ID="comment">
        <rdfs:domain rdf:resource="#ImageContentDescription"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <owl:ObjectProperty rdf:ID="depictedItem">
        <rdfs:domain rdf:resource="#ImageContentDescription"/>
        <rdfs:range rdf:resource="&tang;TangibleThing"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="depictedEvent">
        <rdfs:domain rdf:resource="#ImageContentDescription"/>
        <rdfs:range rdf:resource="&ev;Event"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="rating">
        <rdfs:domain rdf:resource="#ImageContentDescription"/>
        <rdfs:range rdf:resource="&rat;Rating"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="position">
        <rdfs:domain rdf:resource="#ImageContentDescription"/>
        <rdfs:range rdf:resource="&pos;Position"/>
    </owl:ObjectProperty>
</rdf:RDF>
```

Fragment D.6: OWL schema describing a person.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF[
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY pcms "http://multimedialab.elis.ugent.be/users/gmartens/
    Ontologies/Pecman/v1.0/">
<!ENTITY addr "&pcms;Address.owl#" >
<!ENTITY phone "&pcms;PhoneNumber.owl#" >
<!ENTITY org "&pcms;Organization.owl#" >
<!ENTITY tang "&pcms;TangibleThing.owl#">
<!ENTITY log "&pcms;Log.owl#">
<!ENTITY ipr "&pcms;IPR.owl#">
<!ENTITY puser "&pcms;PUser.owl#">

]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="&xsd;"
    xmlns:rdfs="&rdfs;"
    xmlns:owl="&owl;"
    xmlns="&pcms;Person.owl#"
    xml:base="&pcms;Person.owl"
```

```
xmlns:addr="&addr;"
xmlns:phone="&phone;"
xmlns:org="&org;"
xmlns:tang="&tang;"
xmlns:log="&log;"
xmlns:ipr="&ipr;"
xmlns:puser="&puser;">

<!-- ==============  -->
<!-- PERSON ontology -->
<!-- ==============  -->

<owl:Ontology rdf:about="">
    <rdfs:comment xml:lang="en"> Pecman ontology for describing
         a person </rdfs:comment>
    <owl:versionInfo rdf:datatype="&xsd;string">version 0.2</
        owl:versionInfo>
    <owl:imports rdf:resource="&tang;"/>
    <owl:imports rdf:resource="&addr;"/>
    <owl:imports rdf:resource="&org;"/>
    <owl:imports rdf:resource="&phone;"/>
    <owl:imports rdf:resource="&tang;"/>
    <owl:imports rdf:resource="&puser;"/>
    <owl:imports rdf:resource="&ipr;"/>
</owl:Ontology>

<!-- ********************************** -->
<!-- Definition of the Person class    -->
<!-- ********************************** -->

<owl:Class rdf:ID="Person">
    <rdfs:label>Person</rdfs:label>
    <rdfs:subClassOf rdf:resource="&tang;TangibleThing"/>
    <rdfs:subClassOf rdf:resource="&ipr;IPRClaimer"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#birthDate"/>
            <owl:maxCardinality rdf:datatype="&xsd;
                nonNegativeInteger"
            >1</owl:maxCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#organizationMember"/
                >
            <owl:maxCardinality rdf:datatype="&xsd;
                nonNegativeInteger"
            >1</owl:maxCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#address"/>
            <owl:maxCardinality rdf:datatype="&xsd;
                nonNegativeInteger"
            >1</owl:maxCardinality>
```

```
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#familyName"/>
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger"
                >1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#userReference"/>
                <owl:maxCardinality rdf:datatype="&xsd;
                    nonNegativeInteger"
                >1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>

    </owl:Class>

    <!-- ******************************************** -->
    <!-- Property definitions of the Person class   -->
    <!-- ******************************************** -->

    <!--  name properties -->
    <owl:DatatypeProperty rdf:ID="nickName">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="&xsd;string"/>
        </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="familyName">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="&xsd;string"/>
        </owl:DatatypeProperty>

    <!-- contact information -->
    <owl:ObjectProperty rdf:ID="address">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="&addr;Address"/>
    </owl:ObjectProperty>

    <owl:ObjectProperty rdf:ID="phone">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="&phone;PhoneNumber"/>
    </owl:ObjectProperty>

    <owl:DatatypeProperty rdf:ID="email">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="&xsd;string"/>
        </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="web">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="&xsd;string"/>
        </owl:DatatypeProperty>
```

```
        <!-- birth date -->
130     <owl:DatatypeProperty rdf:ID="birthDate">
            <rdf:type rdf:resource="&owl;FunctionalProperty"/>
            <rdfs:domain rdf:resource="#Person"/>
            <rdfs:range rdf:resource="&xsd;date"/>
            </owl:DatatypeProperty>
135
        <!--  organization    -->
        <owl:ObjectProperty rdf:ID="organizationMember">
            <rdfs:domain rdf:resource="#Person"/>
            <rdfs:range rdf:resource="&org;Organization"/>
140     </owl:ObjectProperty>

        <!--  Pecman-user reference    -->
        <owl:ObjectProperty rdf:ID="userReference">
            <rdfs:domain rdf:resource="#Person"/>
145         <rdfs:range rdf:resource="&puser;PecmanUser"/>
        </owl:ObjectProperty>
    </rdf:RDF>
```

References

[1] A. Hildebrand. Megapixel Cameras used in Smart CCTV Project, June 2008. Available on `http://www.info4security.com/story.asp?sectioncode=9&storycode=4119495`.

[2] IMS Research. IMS Research Cuts 2008 Forecast for US Network Video Surveillance Market, 2008. Available on `http://www.marketresearchworld.net/`.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):334–352, 2004.

[4] A. Mittal and D. Huttenlocher. Scene Modelling for Wide Area Surveillance and Image Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 160–167, 2000.

[5] F. Tiburzi and J. Bescos. Camera Motion Analysis in on-line MPEG Sequences. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, pages 42–46, 2007.

[6] A. Dick and M.J. Brooks. Issues in Automated Visual Surveillance. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, pages 195–204, 2003.

[7] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):34–77, 1994.

[8] H. Li, S. Lin, and Y. Zhang. Combining Template Matching and Model Fitting for Human Body Segmentation and Tracking with Applications to Sports Training. In *Proceedings of the Image Analysis and Recognition*, pages 823–831, 2006.

[9] O. Javed, K. Shafique, and M. Shah. A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information. In *Proceedings of the Workshop on Motion and Video Computing*, pages 22–27, 2002.

[10] T. Wiegand, G. Sullivan, G. Bjϕntegaard, and G. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[11] W3C Semantic Web Activity. Available on `http://www.w3.org/2001/sw/`.

[12] P. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, 10 february 2004. Available on `http://www.w3.org/TR/2004/REC-owl-semantics-20040210/`.

[13] Chris Poppe, Sarah De Bruyne, Tom Paridaens, Peter Lambert, and Rik Van de Walle. Moving Object Detection in the H.264/AVC Compressed Domain for Video Surveillance Applications. *Visual Communication and Image Representation*.

[14] Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. Personal Content Management System a Semantic Approach. *Visual Communication and Image Representation*, 47:131 – 144, February 2009.

[15] Chris Poppe, Gaëtan Martens, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Robust spatio-temporal multimodal background subtraction for video surveillance. *Optical Engineering*, 47:110101, October 2008.

[16] Davy De Schrijver, Chris Poppe, Sam Lerouge, Wesley De Neve, and Rik Van de Walle. MPEG-21 Bitstream Syntax Descriptions for Scalable Video Codecs. *Multimedia Systems*, 11:403–421, June 2006.

[17] Sarah De Bruyne, Chris Poppe, Steven Verstockt, Peter Lambert, and Rik Van de Walle. Estimating Motion Reliability to Improve Moving Object Detection in the H.264/AVC Domain. In *To be published in Proceedings of The International Conference on Multimedia and Expo (ICME) 2009*, June 2009.

[18] Chris Poppe, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Effect of H.264/AVC Compression on Object Detection for Video Surveillance. In *Proceedings of The International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2009*, May 2009.

[19] Davy Van Deursen, Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. XML to RDF Conversion: a Generic Approach. In *Proceedings of 4th International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS) 2008*, pages 138 – 144, 2008.

[20] Gaëtan Martens, Chris Poppe, Peter Lambert, and Rik Van de Walle. Unsupervised Texture Segmentation and Labeling using Biologically Inspired Features. In *Proceedings of MMSP2008*, pages 159–164, 2008.

[21] Chris Poppe, Gaëtan Martens, Sarah De Bruyne, Peter Lambert, and Rik Van de Walle. Dealing with Gradual Lighting Changes in Video Surveillance for Indoor Environments. In *Proceedings of 15th World Congress on Intelligent Transport Systems*, November 2008.

[22] Chris Poppe, Sarah De Bruyne, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Intelligent Preprocessing for Fast Moving Object Detection. In *Proceedings of SPIE Security and Defense*, volume 6978, March 2008.

[23] Chris Poppe, Frederik De Keukelaere, Saar De Zutter, Sarah De Bruyne, Wesley De Neve, and Rik Van de Walle. Predictable Processing of Multimedia Content, Using MPEG-21 Digital Item Processing. In *Proceedings of the 8th Pacific Rim Conference on Multimedia*, volume 4810, pages 549 – 558, December 2007.

[24] Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Improved Background Mixture Models for Video Surveillance Applications. *Lecture Notes in Computer Science, 8th Asian Conference on Computer Vision(ACCV 2007)*, 4843:251–260, 2007.

[25] Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Dealing with Quick Illumination Changes when using Background Mixture Models. In *Proceedings 8th Asian Conference on Computer Vision*, volume 4843, Tokyo, November 2007.

[26] Chris Poppe, Gaëtan Martens, Peter Lambert, and Rik Van de Walle. Mixture Models Based Background Subtraction for Video Surveillance Applications. In *Lecture Notes in Computer Science: Proceedings 12th International Conference on Computer Analysis of Images and Patterns*, volume 4673, pages 28–35, August 2007.

[27] Chris Poppe, Saar De Zutter, Wesley De Neve, and Rik Van de Walle. Reconfigurable Multimedia: Putting the User in the Middle. In *Proceedings of COST298 conference: the Good, the Bad an the Unexpected*, pages 15–30, May 2007.

[28] Gaëtan Martens, Chris Poppe, and Rik Van de Walle. Enhanced Grating Cell Features for Unsupervised Texture Segmentation. In *Proceedings of Performance Evaluation for Computer Vision: 31ste AAPR/OAGM Workshop*, pages 9–16, Wien, May 2007.

[29] M. Ransburg, H. Hellwagner, R. Cazoulat, B. Pellan, C. Concolato, S. De Zutter, C. Poppe, R. Van de Walle, and A Hutter. Dynamic and Distributed Adaptation of Scalable Multimedia Content in a Context-Aware Environment. In *Proceedings of the WiCon'06 conference*, 2006.

[30] Chris Poppe, Frederik De Keukelaere, Saar De Zutter, and Rik Van de Walle. Advanced Multimedia Systems using MPEG-21 Digital Item Processing. In *Proceedings of the Eight IEEE International Symposium on Multimedia*, pages 785–786, San Diego, December 2006.

[31] Chris Poppe, Ingo Wolf, Sven Wischnowsky, Saar De Zutter, and Rik Van de Walle. Licensing System in an online MPEG-21 Environment. In *Proceedings of the IADIS international conference WWW/internet 2006*, volume II, pages 315–319, Murcia, October 2006.

[32] Chris Poppe, Michael Ransburg, Saar De Zutter, and Rik Van de Walle. Interoperable Affective Context Collection using MPEG-21. In *Proceedings of the International Conference on Wireless Mobile and Multimedia Networks Proceedings*, volume II, pages 1603–1606, China, November 2006.

[33] Saar De Zutter, Frederik De Keukelaere, Chris Poppe, and Rik Van de Walle. Performance Analysis of MPEG-21 Technologies on Mobile Devices. In *Proceedings of SPIE-IST Electronic Imaging, Science and Technology*, volume 6074, page 12, San José, January 2006.

[34] Chris Poppe, Saar De Zutter, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13965, Contribution to Utility Software for ISO/IEC 21000-10 DIP/AMD 1, MPEG-document, October 2006.

[35] Saar De Zutter, Chris Poppe, Davy De Schrijver, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13979, Update to Reference Software for Conformance to ISO/IEC 21000-10 DIP/AMD 1, MPEG-document, October 2006.

[36] Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13593, Contribution to Conformance Reference Software for ISO/IEC 21000-10 DIP AMD/1, MPEG-document, July 2006.

[37] Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m13592, Contribution to Conformance for ISO/IEC 21000-10 DIP AMD/1, MPEG-document, July 2006.

[38] Saar De Zutter, Chris Poppe, Frederik De Keukelaere, and Rik Van de Walle. ISO/IEC JTC1/SC29/WG11 m12218, Contribution to WD of Amendment on C++ Binding, MPEG-document, July 2005.

[39] B.P.L. Lo and S.A. Velastin. Automatic Congestion Detection System for Underground Platforms. In *Proceedings of the 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 158–161, 2000.

[40] M. Piccardi R. Cucchiara, C. Grana and A. Prati. Detecting Moving Objects, Ghosts and Shadows in Video Streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

[41] O. Munkelt C. Ridder and H. Kirchner. Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199, 1995.

[42] T. Darrell C.R. Wren, A. Azarbayejani and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[43] C. Stauffer and W. E. L. Grimson. Learning Patterns of Activity using Real-Time Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. Available on `citeseer.ist.psu.edu/article/stauffer00learning.html`.

[44] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and Practice of Background Maintenance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 255–261, 1999.

[45] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-Time Surveillance of People and their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[46] J. Wu and M. Trivedi. Performance Characterization for Gaussian Mixture Model based Motion Detection Algorithms. In *Proceedings of the IEEE International Conference on Image Processing*, pages 97–100, 2005.

[47] D. Lee. Online Adaptive Gaussian Mixture Learning for Video Applications. In *Lecture Notes in Computer Science, Statistical Methods in Video Processing*, pages 105–116, 2004.

[48] Y. Zhang, Z. Liang, Z. Hou, H. Wang, and M. Tan. An Adaptive Mixture Gaussian Background Model with Online Background Reconstruction and Adjustable Foreground Mergence Time for Motion Segmentation. In *Proceedings of the IEEE International Conference on Industrial Technology*, pages 23–27, 2005.

[49] Z. Zivkovic. Improved adaptive Gaussian Mixture Model for Background Subtraction. In *Proceedings of the International Conference on Pattern Recognition*, pages 28–31, 2004.

[50] Yong Shan and Runsheng Wang. Improved Algorithms for Motion Detection and Tracking. *Optical Engineering*, 45(6):067201, 2006.

[51] H. Yang, Y. Tan, and J. Tian J. Liu. Accurate Dynamic Scene Model for Moving Object Detection. In *Proceedings of the IEEE International Conference on Image Processing*, pages 157–160, 2007.

[52] Y. Tian, M. Lu, and A. Hampapur. Robust and Efficient Foreground Analysis for Real-Time Video Surveillance. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1182–1187, 2002.

[53] A. Prati, I. Mikic, M.M. Trivedi, and R. Cucchiara. Detecting Moving Shadows: Algorithms and Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, 2003.

[54] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati. The Sakbot System for Moving Object Detection and Tracking. In *Proceedings of Video-Based Surveillance Systems - Computer Vision and Distributed Processing*, pages 145–157, 2001.

[55] M. Yokoyama and T. Poggio. A Contour-Based Moving Object Detection and Tracking. In *Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 271–276, 2005.

[56] Y. Hwang, K. Jun-Sik, and K.In-So. Sensor Noise Modeling using the Skellam Distribution: Application to the Color Edge Detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[57] I. Young, J. Gerbrands, and L. van Vliet. Fundamentals of Image Processing, 1995. Available on `citeseer.ist.psu.edu/young95fundamentals.html`.

[58] J.G. Skellam. The Frequency Distribution of the Difference between two Poisson Variates Belonging to Different Populations. *Journal of the Royal Statistical Society: Series A*, 109(3):296, 1946.

[59] M. Abramowitz and I.A. Stegun. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 1973.

[60] E.R. Dougherty. Probability and Statistics for the Engineering, Computing and Physical Sciences, 1990.

[61] Q. Zang and R. Klette. Parameter Analysis for Mixture of Gaussians Model. In *Communication and Information Technology Research - Technical Report - 188*, 2006.

[62] S. Cheung and C. Kamath. Robust Techniques for Background Subtraction in Urban Traffic Video. *Visual Communications and Image Processing*, pages 881–892, 2004.

[63] L.M. Brown, A.W. Senior, Y. Tian, J. Connell, A. Hampapur, C. Shu, H. Merkl, and M. Lu. Performance Evaluation of Surveillance Systems Under Varying Conditions. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005. Available on `http://www.research.ibm.com/peoplevision/performanceevaluation.html`.

[64] Qi Zang and Reinhard Klette. Evaluation of an Adaptive Composite Gaussian Model in Video Surveillance. In *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, pages 165–172, 2003.

[65] S. Lu, J. Zhang, and D. Feng. An Efficient Method for Detecting Ghost and Left Objects in Surveillance Video. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 540–545, 2007.

[66] P. Rosin and T. Ellis. Image Difference Threshold Strategies and Shadow Detection. In *Proceedings of the British conference on Machine vision*, pages 347–356, 1995.

[67] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[68] J. De Bock, R. Pires, P. De Smet, and W. Philips. A Fast Dynamic Border Linking Algorithm for Region Merging. In *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 232–241, 2006.

[69] P. Lambert, W. De Neve, P. De Neve, I. Moerman, P. De Meester, and R. Van de Walle. Rate-Distortion Performance of H.264/AVC Compared to State-Of-The-Art Video Codecs. 16(1):134–140, 2006.

[70] S. Kwon, A. Tamhamkar, and K. R. Rao. Overview of H.264/MPEG-4 part 10. *Journal of Visual Communication and Image Representation*, 17:186–216, 2006.

[71] H. Wang, A. Divakaran, A. Vetro, S. Chang, and H. Sun. Survey on Compressed-Domain Features used in Video/Audio Indexing and Analysis. 14(2):150–183, 2003.

[72] H. Zen, T. Hasegawa, and S. Ozawa. Moving Object Detection from MPEG Coded Picture. In *Proceedings of the International Conference on Image Processing*, pages 25–29, 1999.

[73] M.L. Jamrozik and M. H. Hayes. A Compressed Domain Video Object Segmentation System. In *Proceedings of the International Conference on Image Processing*, pages 113–116, 2002.

[74] L. Long, F. Xingle, J. Ruirui, and D. Yi. A Moving Object Segmentation in MPEG Compressed Domain Based on Motion Vectors and DCT Coefficients. In *Proceedings of the Congress on Image and Signal Processing*, pages 605–609, 2008.

[75] V. Thilak and C. Creusere. Tracking of Extended Size Targets in H.264 Compressed Video using the Probabilistic Data Association Filter. In *Proceedings of the European Signal Processing Conference*, pages 281–284, 2004.

[76] W. Zeng, J. Du, W. Gao, and Q. Huang. Robust Moving Object Segmentation on H.264 Compressed Video using the Block-Based MRF Model. *Real-Time Imaging*, 11:290–299, 2005.

[77] G. Yang, S. Yu, and Z. Zhang. Robust Moving Object Segmentation in the Compressed Domain for H.264 Video Stream. In *Proceedings of the Picture Coding Symposium*, 2006.

[78] Z. Liu, Z. Zhang, and L. Shen. Moving Object Segmentation in the H.264 Compressed Domain. *Optical Engineering*, 46(1):017003, 2007.

[79] S. Cheung and C. Kamath. Robust Techniques for Background Subtraction in Urban Traffic Video. In *Proceedings of Visual Communications and Image Processing*, pages 881–892, 2004.

[80] L.M. Brown, A.W. Senior, Y. Tian, J. Connell, A. Hampapur, C. Shu, H. Merkl, and M. Lu. Performance Evaluation of Surveillance Systems Under Varying Conditions. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005. Available on `http://www.research.ibm.com/peoplevision/performanceevaluation.html`.

[81] I. E. G. Richardson. H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia. 2003.

[82] Chris Poppe and Rik Van de Walle. High-speed Moving Object Detection in H.264/AVC Compressed Domain. In *Proceedings 9th FirW Phd Symposium*, pages 204–205, December 2008.

[83] J. Black, D. Makris, and T. Ellis. Hierarchical Database for a Multi-Camera Surveillance System. *Pattern Analysis and Applications*, pages 430–446, 2005.

[84] D.C. Fallside and P. Walmsley. XML Schema part 0: Primer (second edition). W3C Recommendation, W3C, October 2004.

[85] R.G.J. Wijnhoven and E.G.T. Jaspersand P.H.N. de With. Flexible surveillance system architecture for prototyping video content analysis algorithms. In *Proc. SPIE Electronic Imaging*, volume 6073, 2006.

[86] MPEG-7 Overview, International Organization for Standardisation, Klagenfurt ISO/IEC JTC1/SC29/WG11, July 2002.

[87] K. Zerzour, G. Frazier, and F. Marir. VIGILANT: A Semantic Model for Content and Event Based Indexing and Retrieval of Surveillance Video. In *Proceedings of International Workshop onKnowledge Representation meets Databases*, pages 143–154, 2000.

[88] D.P. Young and J.M. Ferryman. PETS Metrics: On-Line Performance Evaluation Service. In *Proceedings of Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 317– 324, 2005.

[89] T. List and R.B. Fisher. CVML - An XML-Based Computer Vision Markup Language. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 789–792, 2004.

[90] J. Annesley, A. Colombo, J. Orwell, and S. Velastin. A Profile of MPEG-7 for Visual Surveillance. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 482–487, 2007.

[91] J. Clark. XSL Transformations: XSLT (Version 1.0). W3C Recommendation, W3C, November 1999.

[92] R. Nevatia, J. Hobbs, and R. Bolles. An Ontology for Video Event Representation. In *Proceedings of Computer Vision and Pattern Recognition Workshop*, pages 119–129, 2004.

[93] A.R.J. Francois, R.Nevatia, J. Hobbs, and R. Bolles. VERL: An Ontology Framework for Representing and Annotating Video Events. *IEEE Multimedia*, pages 76–78, 2005.

[94] D. Gussow. New Lens on War, St. Petersburgh Times Online, 2004. Available on `http://www.sptimes.com/`.

[95] Photo Metadata, White Paper, International Press Telecommunications Council (IPTC), 2007.

[96] I. Burnett, F. Pereira, R. Van de Walle, and R. Koenen. The MPEG-21 Book, Whiley, New Yersey, 2006.

[97] I3A DIG35 Initiative Group. Available on `http://www.i3a.org/technologies/metadata/`.

[98] R. Arndt, R. Troncy, S. Staab, and M. Vacura L. Hardman. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In *Lecture Notes of Computer Science: The Semantic Web*, volume 4825, pages 30–43, 2007.

[99] C. Poppe, G. Martens, P. Lambert, and R. Van de Walle. Mixture Models Based Background Substraction for Video Surveillance Applications. *Lecture Notes in Computer Sciences, Computer Analysis of Images and Patterns*, 4673:28–35, 2007.

[100] W3C Multimedia Semantics Incubator Group. Available on `http://www.w3.org/2005/Incubator/mmsem/`.

[101] F. Manola and E. Miller. RDF Primer. W3C Recommendation, W3C, February 2004.

[102] J. Hunter. Adding Multimedia to the Semantic Web - Building an 7 ontology. In *Proceedings of the First Semantic Web Working Symposium (SWWS)*, pages 261–281, 2001.

[103] I.F. Cruz, H. Xiao, and F. Hsu. An Ontology-Based Framework for XML Semantic Integration. In *Proceedings of International Database Engineering and Applications Symposium*, pages 217–226, 2004.

[104] J. Hunter and S. Little. A Framework to enable the Semantic Inferencing and Querying of Multimedia Content. *International Journal of Web Engineering and Technology*, 2:264–286, 2005.

[105] C. Lagoze and J. Hunter. The ABC Ontology and Model. *Journal of Digital Information*, 2(2), 2001.

[106] D. Vallet, P. Castells, M. Fernandez, P. Mylonas, and Y. Avrithis. Personalized Content Retrieval in Context Using Ontological Knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 17:336–346, 2007.

[107] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, C. Ferrario, R. Gangemi, N. Oltramari, and A. Schneider. The WonderWeb Library of Foundational Ontologies (WFOL). Technical report, WonderWeb Deliverable 17. 2002.

[108] M. E. Dönderler, E. Saykol, U. Arslan, Ö. Ulusoy, and U. Güdükbay. BilVideo: Design and Implementation of a Video Database Management System. *Multimedia Tools and Applications*, 27:79–104, 2005.

[109] P. Schallauer, W. Bailer, and G. Thallinger. A Description Infrastructure for Audiovisual Media Processing Systems Based on MPEG-7. *Journal of Universal Knowledge Management*, 1:26–35, 2006.

[110] K. Petridis, S. Bloehdorn, C. Saathoff, N. Simou, S. Dasiopoulou, V. Tzouvaras, S. Handschuh, Y. Avrithis, Y. Kompatsiaris, and S. Staab. Knowledge Representation and Semantic Annotation of Multimedia Content. *IEE Proceedings Vision, Image and Signal Processing*, 153(3):255–262, 2006.

[111] N. Simou, V. Tzouvaras, Y. Avrithis, G. Stamou, and S. Kollias. A Visual Descriptor Ontology of Multimedia Reasoning. In *Proc. Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[112] ISO/IEC 15938-3 FCD Information Technology - Multimedia Content Description Interface - Part 3:Visual, March 2001, Singapore.

[113] P. Salembier and J.R. Smith. MPEG-7 Multimedia Description Schemes. *IEEE Trans. Circuits and Systems for Video Technology*, 11:748–759, 2001.

[114] P. Asirelli, S. Little, M. Marinelli, and O. Salvetti. MultiMedia Metadata Management: a Proposal for an Infrastructure. In *Proc. Semantic Web Applications and Perspectives (SWAP)*, pages 255–261, 2006.

[115] R. Garcia and O. Celma. Semantic Integration and Retrieval of Multimedia Metadata. In *Proc. 5th Knowledge Markup and Semantic Annotation Workshop*, pages 69–80, 2006.

[116] V. Tzouvaras, R. Troncy, and J. Z. Pan. Multimedia Annotation Interoperability Framework, W3C Incubator Group Report 24 July 2007. Available on `http://www.w3.org/2005/Incubator/mmsem/XGR-interoperability/`.

[117] N. Sluijs, K. Vlaeminck, T. Wauters, B. Dhoedt, F. De Turck, and P. Demeester. Caching strategies for Personal Content Storage Grids. In *Proc. the International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 396–404, 2007.

[118] X. Wang, T. DeMartini, B. Wragg, M. Paramasivam, and C. Barlas. The MPEG-21 Rights Expression Language and Rights Date Dictionary. *IEEE Transactions on Multimedia, International Conference on Parallel and Distributed Processing Techniques and Applications*, 7(3):408–417, 2005.

[119] C. Poppe, I. Wolf, S. Wischnowsky, S. De Zutter, and R. Van de Walle. Licensing System in an online MPEG-21 Environment. In *Proc. the International Conference WWW/Internet*, pages 315–319, 2006.

[120] J. S. Hare, P. A.S. Sinclair, P. H. Lewis, K. Martinez, P. G.B. Enser, and C. J. Sandom. Bridging the Semantic Gap in Multimedia Information Retrieval. In *Proceedings of the third European Semantic Web Conference Workshop on Mastering the Gap: From Information Extraction to Semantic Representation*, 2006.

[121] M. Hausenblas, S. Boll, T. Burger, O. Celma, C. Halaschek-Wiener, E. Mannens, and R. Troncy. Multimedia Vocabularies on the Semantic Web, W3C Incubator Group Report 24 July 2007. Available on `http://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/`.

[122] R. Troncy, W. Bailer, M. Hausenblas, P. Hofmair, and R. Schlatte. Enabling Multimedia Metadata Interoperability by Defining Formal Semantics of MPEG-7 Profiles. In *Lecture Notes in Computer Science*, volume 4306, pages 41–55, 2006.

[123] Describing and Retrieving Photos using RDF and HTTP, W3C Note, 2002. Available on `http://www.w3.org/TR/photo-rdf/`.

[124] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language - Combining OWL and RuleML, W3C Member Submission, 21 May 2004. Available on `http://www.w3.org/Submission/SWRL/`.

[125] BRICKS - Building Resources for Integrated Cultural Knowledge Services, 2007. Available on `http://www.brickscommunity.org/prj`.

[126] SPARQL Query Language for RDF, W3C Recommendation 15 January 2008. Available on `http://www.w3.org/TR/rdf-sparql-query/`.

[127] S. Battle. Gloze, XML to RDF and back again. In *Proceedings of First Jena User Conference*, 2006.

[128] M. Ferdinand, C. Zirpins, and D. Trastour. Lifting XML Schema to OWL. In *Proc. Web Engineering - 4th International Conference*, pages 354–358, 2004.

[129] M. Klein. Interpreting XML Documents via an RDF Schema Ontology. In *Proc. DEXA Workshop*, pages 889–894, 2002.

[130] B. Amann, C. Beeri, I. Funulaki, and M. Scholl. Ontology-Based Integration of XML Web Resources. In *Proc. the First International Semantic Web Conference on The Semantic Web*, pages 117–131, 2002.

[131] D. Van Deursen, C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. XML to RDF conversion: a Generic Approach. In *Proc. the 4th International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, 2008.

[132] R. Wilkinson and M. Wu. Evaluation Experiments and Experience from the Perspective of Interactive Information Retrieval. In *Proc. 3rd Workshop Empirical Evaluation Adaptive Systems*, pages 221–230, 2004.

[133] D. Marpe, S. Gordon, and T. Wiegand. H.264/MPEG4-AVC Fidelity Range Extensions:Tools, Profiles, Performance, and Application Areas. In *Proceedings of IEEE International Conference on Image Processing (ICIP 2005)*, pages 593–596, September 2005.

[134] G. J. Sullivan, P. Topiwala, and A. Luthra. The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. In *Proceedings of SPIE annual meeting 2004: Signal and Image Processing and Sensors*, pages 454–474, August 2004.

[135] F. Provost, T. Fawcett, and R. Kohavi. The Case against Accuracy Estimation for Comparing Induction Algorithms. In *The 15th International Conference on Machine Learning*, pages 445–453, San Francisco, CA, 1998.

[136] V. Raghavan, P. Bollmann, and G. S. Jung. A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. *ACM Transactions on Information Systems*, 7:205–229, 1989.

[137] J. Bockhorst and M. Craven. Markov Networks for Detecting Overlapping Elements in Sequence Data. *Advances in Neural Information Processing Systems*, 17:193–200, 2005.

[138] S. Kok and P. Domingos. Learning the Structure of Markov Logic Networks. In *22nd International Conference on Machine Learning*, pages 441–448, 2005.

[139] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC Curves. Technical report, University of Wisconsin - Madison Computer Science Departme, 2006.