Universiteit Gent
Faculteit Ingenieurswetenschappen
Vakgroep Telecommunicatie en
Informatieverwerking

Universitat Politècnica de València
Escuela Técnica Superior de Ingenieros de
Telecomunicación
Departamento de Comunicaciones

Optimized Information Processing in Resource-Constrained Vision Systems

From Low-Complexity Coding to Smart Sensor Networks

Marleen Morbee

Advisors:
Prof. Dr. Ir. Wilfried Philips (Ghent University)
Prof. Dr. José Prades Nebot (Universidad Politécnica de Valencia)
Prof. Dr. Hamid Aghajan (Stanford University)

Proefschrift tot het bekomen van de graad van
Doctor in de Ingenieurswetenschappen (Translation:
Doctor of Engineering)
Academiejaar 2010-2011

Tesis para obtener el título de
Doctora por la Universidad Politécnica de Valencia
Curso 2010-2011

PhD committee of Ghent University

Prof. Rik Van de Walle, chairman (Ghent University)
Prof. Richard Kleihorst, secretary (Vlaams Instituut voor Technologisch Onderzoek, Ghent University)
Prof. Wilfried Philips, advisor and member of the reading committee (Ghent University)
Prof. José Prades Nebot, advisor (Universidad Politécnica de Valencia)
Prof. Hamid Aghajan, advisor (Stanford University)
Prof. Peter Schelkens, member of the reading committee (Vrije Universiteit Brussel)
Dr. Ewout Vansteenkiste, member of the reading committee (Ghent University)
Prof. Ben Kröse (Universiteit van Amsterdam, Hogeschool van Amsterdam)
Prof. Joris Walraevens (Ghent University)

PhD committee of Universidad Politécnica de Valencia

Evaluadores del borrador de la tesis doctoral

Prof. Manuel Pérez Malumbres (Universidad Miguel Hernández)
Prof. Pedro Ángel Cuenca Castillo (Universidad de Castilla-La Mancha)
Prof. Francisco José Quiles Flor (Universidad de Castilla-La Mancha)

Miembros del tribunal

Prof. Rik Van de Walle, presidente (Ghent University)
Prof. Tinne Tuytelaars, secretario (Katholieke Universiteit Leuven)
Prof. Adrian Munteanu, vocal (Vrije Universiteit Brussel)
Dr. Gauthier Lafruit, vocal (Interuniversity Microelectronics Centre – IMEC)
Prof. Sabine Wittevrongel, vocal (Ghent University)
Prof. Ben Kröse, suplente (University of Amsterdam)
Prof. Peter Schelkens, suplente (Vrije Universiteit Brussel)

Affiliations

Ghent University
Faculty of Engineering
Department of Telecommunications and Information Processing (TELIN)
Research Group for Image Processing and Interpretation (IPI)
Interdisciplinary Institute for Broadband Technology (IBBT)
Sint-Pietersnieuwstraat 41
9000 Ghent
Belgium

Universidad Politécnica de Valencia
Escuela Técnica Superior de Ingenieros de Telecomunicación
Departamento de Comunicaciones
Image and Video Processing Group (GPIV)
Camino de Vera S/N
46071 Valencia
Spain

Stanford University
Electrical Engineering Department
Wireless Sensor Networks Lab
David Packard Building, Room 318
Stanford, CA 94305
USA

Acknowledgements

I would like to take this opportunity to thank everybody who has supported me to write the book you have in front of you. In particular, I would like to express my gratitude to:

- Prof. Wilfried Philips (Ghent University) for giving me the opportunity to conduct doctoral research at his lab in the exciting research domain of video processing. I would like to thank him for the many inspiring discussions, and for his critical yet very enlightening views on my research. I am very thankful to him for giving me the freedom to pursue my many goals and plans, while at the same time providing the technical and personal support I needed.

- Prof. José Prades Nebot (Universidad Politécnica de Valencia) for supervising my Master's thesis during my Erasmus year at the Universidad Politécnica de Valencia, which was the start of a very successful collaboration. This collaboration formed the foundation of this joint PhD. I would like to thank him for the intensive and instructive teamwork at the research lab in Valencia, and for his very dedicated support, also remotely, which has made this joint PhD come true. His many inputs have made an essential contribution to my PhD, especially to the Chapters 2 and 3.

- Prof. Hamid Aghajan (Stanford University) for giving me the opportunity to spend 8 months doing research at his lab in Stanford University. My stays as visiting researcher at Stanford University allowed me to broaden the scope of my research towards smart sensor networks, which was outside the area of expertise of the research labs at UGent and UPV. I would like to thank him for the very instructive and inspiring interaction during and after the months I was in Stanford, which has been crucial to the exploration of this new research track. The research done in collaboration with him is the subject of the second half of my PhD, and would have been impossible without his inputs and expertise.

- The members of the PhD committees of UGent and UPV, for their willingness to review the manuscript, for their positive feedback, and for the valuable suggestions they made, which enhanced the quality of this manuscript. I would like to thank prof. Rik Van de Walle (Ghent University), prof. Wilfried Philips (Ghent University), prof. José Prades Nebot (Universidad Politécnica de Valencia), prof. Hamid Aghajan (Stanford University), prof. Peter Schelkens (Vrije Universiteit Brussel), dr. Ewout Vansteenkiste (Ghent University), prof. Ben Kröse (Universiteit van Amsterdam, Hogeschool van Amsterdam), prof. Richard Kleihorst (Vlaamse Instelling voor Technologisch Onderzoek, Ghent University) and prof. Joris Walraevens (Ghent University) for being member of the UGent PhD committee. I would like to thank the members of the UPV PhD committee, prof. Rik Van de Walle (Ghent University), prof. Manuel Pérez Malumbres (Universidad Miguel Hernández), prof. Pedro Ángel Cuenca Castillo (Universidad de Castilla-La Mancha), prof. Francisco José Quiles Flor (Universidad de Castilla-La Mancha), prof. Tinne Tuytelaars (Katholieke Universiteit Leuven), dr. Gauthier Lafruit (Interuniversity Microelectronics Centre – IMEC), prof. Adrian Munteanu (Vrije Universiteit Brussel), prof. Sabine Wittevrongel (Ghent University), and the deputies prof. Luis Torres Urgell (Universitat Politècnica de Catalunya), prof. Ferran Marqués Acosta (Universitat Politècnica de Catalunya), prof. Philippe Salembier (Universitat Politècnica de Catalunya), prof. Ben Kröse (Universiteit van Amsterdam, Hogeschool van Amsterdam), and prof. Peter Schelkens (Vrije Universiteit Brussel).

- The members of the Image Processing and Interpretation research group at Ghent University for the interesting inputs at the IPI meetings and in the lab. I would like to thank those with whom I shared the computer room for the very friendly atmosphere, like Linda, Alessandro, Jan, Filip, Koen, Sebastian, Dirk, Seyfollah, Jonas, ... and all the others that frequently passed by to exchange some thoughts on research or simply to say hi, like Hiep, Bart, Johan, Ivana, Vedran, Jorge, Tijana, Werner, Ewout, Andrés, Joost, Stefaan, Ljubomir, Daniel, Ljiljana, Sergio, Peter, Danilo, Wenzhi, Cornel, Vincent, Rik, Mona, ... I would like to thank Ivana for her great after-work Pilates courses.

- Dr. Linda Tessens (Ghent University), for the very fruitful collaboration on a daily basis, and for always being there when I needed support, both as a colleague and as a friend.

- Prof. Aleksandra Pizurica (Ghent University), for her supervision and support at the start of my PhD.

- Prof. Richard Kleihorst (Vlaamse Instelling voor Technologisch Onderzoek, Ghent University), for his interesting inputs and ideas on sensor networks and hardware.

- Prof. Sidharta Gautama (Ghent University), Kris Bonnarens (Ghent University), An Van den broecke (Ghent University) and Axel Plas (IP HILLS nv) for their support in the patent application procedure.

- Annette, Patrick and Alice at the Department of Telecommunications and Information Processing of Ghent University, for their friendly assistance on administrative issues, always with enthusiasm and a smile.

- Philippe and Davy, system administrators of the Department of Telecommunications and Information Processing of Ghent University, for their outstanding computer skills and excellent advice on many topics, and for their never-ending willingness to help out.

- All the professors and researchers at the Department of Telecommunications and Information Processing of Ghent University for the technical interactions, but also for the nice chats, and exciting social events.

- All my colleagues at Hogeschool Gent. I would like to thank prof. Peter Veelaert, David, Dimitri, Jonas and Kristof for their support in constructing the real-time demonstrator discussed in Chapter 4.

- Antoni Roca (Universidad Politécnica de Valencia) and Jorge Quintanilla (Universidad Politécnica de Valencia) for the successful teamwork on distributed video coding, for their inputs on Chapter 3, and for the many interesting discussions.

- All my colleagues at the Image and Video Processing Group of the Universidad Politécnica de Valencia for their ideas on research topics, and for the personal and technical support. I would like to thank David, Toni, Pau, Jorge, Pieter, Sole, and all other professors and researchers for the very warm atmosphere at the lab and the many exciting social activities, which made the time I spent in Valencia unforgettable.

- Prof. Salvador Sales, Sandra Chirivella, Josep Antoni Claver Campillo, and Miguel Granell León from the Universidad Politécnica de Valencia for the support in drafting the joint PhD agreement, and during the submission and evaluation of this dissertation at the UPV.

- Prof. Edward J. Delp (Purdue University) for his useful comments on the distributed video coding and modulo-PCM coding research.

- All my colleagues at the Wireless Sensor Networks Lab at Stanford: Chen, Huang, Nan, Itai, Tommi, Stephan, ... for the nice exchanges of ideas during the group meetings and in the lab, but also for their friendly welcome and helpfulness.

- Dr. Vladan Velisavljevic (Deutsche Telekom Laboratories) for the nice collaboration and for the internship opportunity he offered me. I would also like to thank all other colleagues at the Deutsche Telekom Laboratories for the productive interactions, and for creating such a nice environment. I am grateful to Dr. Marta Mrak (University of Surrey) for her valuable inputs on the video retrieval project.

- My director Franco Cordera and my coaches Olivier Beaudoin and Annika Wahrenberg at the European Patent Office (EPO) in Munich, for their encouragement and support to finish my PhD while starting my job at the EPO.

- Marcos de la Torre (European Patent Office) for revising the Spanish summary.

- All those that I met on conferences, workshops, schools, lectures and during the time I was abroad, for the instructive interactions, and the many nice and inspiring moments.

- My family and friends for the relaxing and exciting moments outside of work, and for cheering and supporting me.

- My parents, my brother Joris, my sisters Lieve and Katrien, Sari, Linda and Paulus for their unconditional support which has been invaluable.

Sincere thanks to you all!

Marleen Morbee
March 2011

Samenvatting in het Nederlands –Summary in Dutch–

Visiesystemen zijn alomtegenwoordig geworden. Ze worden gebruikt voor verkeerstoezicht, ouderenzorg, videoconferenties, virtuele realiteit, bewaking, slimme kamers, domotica, sportanalyse, industriële veiligheid, medische zorg, enz. In de meeste visiesystemen worden de gegevens afkomstig van de visuele sensor(en) verwerkt alvorens ze te versturen, om zo te besparen op communicatiebandbreedte of om een hoger aantal beelden per seconde te verkrijgen. Het type dataverwerking dient nauwkeurig gekozen te worden in functie van de doeltoepassing, en rekening houdend met het beschikbare geheugen, de beschikbare rekenkracht, energiebronnen en bandbreedtebeperkingen.

In dit proefschrift onderzoeken we hoe een visiesysteem moet worden gebouwd, onder een gegeven set van praktische randvoorwaarden. Ten eerste moet dit systeem intelligent zijn zodat de juiste informatie gehaald wordt uit de videobron. Ten tweede moet dit intelligente visiesysteem bij het verwerken van de videogegevens zijn eigen beperkingen kennen, om zo tot het best mogelijke resultaat te komen dat binnen zijn mogelijkheden ligt. We bestuderen en verbeteren een brede waaier aan visiesystemen voor een verscheidenheid aan toepassingen, die met verschillende types beperkingen gepaard gaan.

Ten eerste stellen we een op Modulo-PCM gebaseerd coderingsalgoritme voor, voor toepassingen die codering met heel lage complexiteit vereisen en een aantal van de voordelige eigenschappen dienen te behouden van PCM-codering (directe verwerking, willekeurige toegang, schaalbaarheid van datasnelheden). Ons MPCM-coderingsschema combineert drie goed gekende, eenvoudige broncoderingsstrategieën: PCM, klassering en interpolatieve codering. De encoder analyseert eerst de signaalstatistieken op een zeer eenvoudige manier. Op basis van deze signaalstatistieken verwerpt de encoder simpelweg een aantal bits van elk beeldmonster. De MPCM-decoder recupereert de verwijderde bits van elk monster door gebruik te maken van de ontvangen bits en neveninformatie die gegenereerd wordt door eerder gedecodeerde signalen te interpoleren. Ons algoritme is in het bijzonder geschikt voor beeldcodering aangezien het grotere coderingsfouten introduceert in die regio's waar ze minder zichtbaar zijn (randen en texturen).

We ontwikkelen een model voor de coderingsvervorming geïntroduceerd door deze MPCM-coder. Gebruik makend van dit model analyseren we hoe de coderingsparameters gekozen moeten worden in functie van de doeldatasnelheid en de kwaliteit van de neveninformatie.

Experimentele resultaten bekomen bij het encoderen van verschillende digitale beelden tonen dat ons algoritme een betere objectieve en subjectieve performantie heeft dan PCM bij lage datasnelheden. Bij hoge datasnelheden leveren modulo-PCM en PCM gelijkaardige resultaten. Ons algoritme heeft een iets slechtere performantie wat betreft datasnelheid tegenover vervorming dan andere broncoderingstechnieken zoals MPCM-codering met neveninformatie of Wyner-Ziv-videocodering, maar het heeft het voordeel van een veel lagere rekencomplexiteit. Dit maakt ons algoritme heel nuttig in toepassingen die extreem eenvoudige encoders vereisen, zoals het encoderen van videosignalen van hoge-snelheidscamera's.

Ten tweede is het in sommige videotoepassingen wenselijk de complexiteit van de video-encoder te verminderen ten koste van een complexere decoder. Voorbeelden van zulke toepassingen zijn draadloze bewaking met laag energieverbruik, draadloze PC-camera's, multimedia-sensornetwerken, wegwerpcamera's, en camera's van mobiele telefoons. Gedistribueerde videocodering is een nieuw paradigma dat aan deze vereiste voldoet door het gebruik van intraframe-encodering en interframedecodering. Daardoor is het grootste deel van de rekenlast verschoven van de encoder naar de decoder, aangezien in dit geval de gedistribueerde videodecoders (en niet de encoders) de bewegingsschatting en de bewegingsgecompenseerde interpolatie uitvoeren. Twee theorema's uit de informatietheorie, namelijk het theorema van Slepian-Wolf voor verliesloze gedistribueerde broncodering en het theorema van Wyner-Ziv voor verlieshebbende broncodering met neveninformatie, suggereren dat zo een systeem met intraframe-encodering en interframedecodering de efficiëntie van een traditioneel interframecoderingssysteem zeer dicht kan benaderen.

Om een beter inzicht te krijgen in de werking van dit soort van coders, starten we met een diepgaande studie van de coderingsvervorming geïntroduceerd door Wyner-Ziv-videocoders die in het pixeldomein opereren. Ons model voor coderingsvervorming kan gebruikt worden om de optimale waarden te bepalen van coderingsparameters onder randvoorwaarden voor datasnelheid en vervorming. Als voorbeeld tonen we hoe ons model gebruikt kan worden om kwaliteitsfluctuaties te verminderen tussen verschillende beelden van de video.

Vele systemen maken gebruik van een terugkoppelingskanaal om een gepaste datasnelheid toe te kennen. Dit terugkoppelingskanaal is echter niet altijd beschikbaar, zoals dit het geval is in offline-codering of in unidirectionele toepassingen. We stellen een algoritme voor de toekenning van de datasnelheid voor dat toelaat om het terugkoppelingskanaal van het coderingsschema te verwijderen. Ons algoritme berekent het aantal bits om elk videobeeld te encoderen zonder de complexiteit van de encoder significant te verhogen. Experimentele resultaten tonen dat ons algoritme voor de toekenning van de datasnelheid goede schattingen levert van de datasnelheid, en dat de beeldkwaliteiten geproduceerd door ons algoritme heel dicht liggen bij die geproduceerd door een algoritme gebaseerd op een terugkoppelingskanaal.

Een algemeen doel in gedistribueerde videocodering is de complexiteit van de encoder zoveel mogelijk verminderen, maar dit gaat natuurlijk ten koste van meer complexiteit van de decoder. In dit opzicht nemen we waar dat deze toename van de complexiteit van de decoder excessief is, en dat bijgevolg de complexiteit van het totale coderingsproces veel hoger is dan in traditionele coderingsschema's. Om dit probleem aan te pakken, ontwikkelen we een methode die de complexiteit van de decoder drastisch vermindert. In deze methode gebruiken we een terugkoppelingskanaal om de datasnelheid toegekend door ons algoritme fijn af te regelen en om een nagenoeg optimale toekenning van datasnelheid te bereiken, terwijl we terzelfdertijd twee ongemakken van een terugkoppelingskanaal elimineren, i.e. zijn negatieve impact op vertraging en complexiteit van de decoder.

Ten derde bestuderen we in detail hoe een visiesysteem moet ontworpen worden voor de specifieke toepassing van 2D bezettingswaarneming. Een 2D bezettingskaart levert een bovenaanzicht van een scène met daarin mensen of objecten. Zulke kaarten zijn belangrijk in vele toepassingen zoals bewaking, slimme kamers, videoconferenties en sportanalyse. We stellen twee verschillende methodes voor. Met een eerste methode mikken we op het leveren van zeer nauwkeurige 2D bezettingskaarten. Daarvoor gebruiken een netwerk van slimme camera's, wat betekent dat in de camera's sterke verwerkingsmogelijkheden geïntegreerd zijn. Bijgevolg kunnen de camera's de videodata verwerken en comprimeren op een intelligente manier alvorens ze naar een basisstation voor centrale verwerking te sturen. Meer bepaald berekent elke camera een voorgrond/achtergrond-silhouet en transfereert dit silhouet naar een referentievlak, gebruik makend van zijn camerabeeld-vloerhomografieën. Deze grondbezettingen, berekend uit elk beeld, worden verstuurd naar een centraal verwerkingsstation. Aangezien de datahoeveelheid nodig om deze grondbezettingen voor te stellen klein is (veel kleiner dan de datahoeveelheid nodig voor een typisch natuurlijk beeld), is de vereiste bandbreedte eerder klein. In het basisstation worden de grondbezettingen van de camera's gefuseerd, gebruik makend van de Dempster-Shafer-theorie van bewijsvoering. De methode levert heel nauwkeurige resultaten voor bezettingsdetectie en presteert beter dan de andere state-of-the-art multicamera-methodes voor de berekening van 2D bezetting.

De eerste methode is zeer nauwkeurig, maar kan niet altijd gebruikt worden wegens praktische beperkingen. De belangrijkste bedenkingen zijn de mogelijkheid van inbreuk op de privacy, de hoge kostprijs, de dure veranderingen aan de infrastructuur, de verwerking met hoge complexiteit en het grote energieverbruik.

Rekening houdend met deze vereisten, stellen we een tweede nieuwe methode voor voor 2D bezettingswaarneming. In deze methode vervangen we de camera door een specifieker apparaat dat bestaat uit een lineair rooster van optische detectie-elementen (bv. fotodiodes), hetgeen we een lijnsensor zullen noemen. We stellen voor om meerdere van deze lijnsensoren te gebruiken om een nauwkeurige 2D bezettingskaart te berekenen. De lijnsensor is bijzonder geschikt voor deze toepassing wegens zijn lage prijs, zijn laag energieverbruik, zijn hoge datasnelheden, zijn grote bitdiepte en zijn privacy-vriendelijke aard. We stellen voor om de lijnsensor te gebruiken samen met een lichtintegrerend optisch systeem, dat ervoor zorgt dat elk detectie-element al het licht integreert binnen een bepaald bereik van invalshoeken. De scanlijn-uitvoer van meerdere lichtintegrerende lijnsensoren is heel geschikt als invoer voor een algoritme voor de berekening van de 2D bezetting. Bezettingsberekening met lichtintegrerende lijnsensoren levert nauwkeurige resultaten op die de resultaten bekomen met camera's heel dicht benaderen, vooral als de lijnsensoren de scène van opzij en niet van boven bekijken.

Ten vierde onderzoeken we hoe een visienetwerk kan omgaan met vele visietaken die gelijktijdig dienen uitgevoerd te worden, bv. het volgen van meerdere personen in een kamer. Het aantal en het type taken waarmee een netwerk kan omgaan is natuurlijk beperkt door de netwerkmiddelen. De belangrijkste beperkingen van een cameranetwerk zijn de beperkte rekenkracht van de camera's en de communicatiebeperkingen.

In een praktisch cameranetwerk belast met verschillende taken en met beperkte netwerkmiddelen, is het doel de beste totale taakperformantie te bereiken door de taken op een efficiënte manier te verdelen over de sensoren in overeenstemming met de gegeven beperkingen. Deze verdeling van taken over de sensoren heet taaktoekenning. In dit proefschrift stellen we een nieuwe, algemene oplossing voor taaktoekenning voor in praktische (i.e. met netwerkbeperkingen) visienetwerken met overlappende gezichtsvelden.

Dit raamwerk biedt de mogelijkheid om de kwaliteit te controleren waarmee de taken worden uitgevoerd, terwijl de taken over de camera's worden verdeeld volgens praktische criteria. In het bijzonder brengt dit raamwerk langs de ene kant kostfuncties met zich mee om de praktische criteria te modelleren, zoals bijvoorbeeld de beperkte rekenkracht van de camera's. Langs de andere kant gebruiken we functies van geschiktheidswaarden die aangeven hoe goed een verzameling van camera's een bepaalde taak kan uitvoeren, met als doel de kwaliteit van de uitgevoerde taken te controleren. De kost- en waardefuncties worden gecombineerd in een optimalisatieprobleem met randvoorwaarden, dat als oplossing de optimale verdeling van de taken over de camera's heeft.

Als demonstratietoepassing gebruiken we onze methode voor het beheren van de taken van het volgen van meerdere personen. We evalueren hoe de volgperformantie beïnvloed wordt door de bandbreedte en de rekenbeperkingen in het netwerk. We testen onze methode op extensieve echte data van verschillende omgevingen van cameranetwerken.

Samengevat zijn de belangrijkste bijdragen van deze dissertatie

- een coderingsalgoritme gebaseerd op modulo-PCM voor het coderen van beelden met heel lage complexiteit;
- een grondige studie en verbetering van gedistribueerde videocoderingsalgoritmes die in het beelddomein opereren;

- twee nieuwe visiesystemen om nauwkeurige 2D bezettingskaarten te berekenen;
- een raamwerk voor taaktoekenning voor intelligente visienetwerken.

Het onderzoek uitgevoerd tijdens dit doctoraat resulteerde in vijf internationale tijdschriftpublicaties (twee gepubliceerd, twee onder review, een in voorbereiding), waarvan drie als eerste auteur [Morbee et al., 2011, Prades-Nebot et al., 2010, Tessens et al., 2011, Morbee et al., 2010, Morbee et al., 2008a], twee (ingediende) octrooiaanvragen als eerste auteur [Morbee and Tessens, 2010, Morbee and Tessens, 2011], twee hoofdstukken in Lecture Notes of Computer Science waarvan een als eerste auteur [Lee et al., 2008, Morbee et al., 2007a] en twaalf publicaties op internationale conferenties, waarvan acht als eerste auteur [Morbee et al., 2009b, Morbee et al., 2009a, Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008, Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a, Morbee et al., 2006b].

Resumen en español –Summary in Spanish–

Los sistemas de visión se han vuelto omnipresentes. Se utilizan para control de tráfico, cuidado de ancianos, videoconferencia, realidad virtual, vigilancia, salas inteligentes, domótica, análisis deportivos, seguridad industrial, asistencia médica, etc. En la mayoría de los sistemas de visión, los datos procedentes de los sensores visuales se procesan antes de la transmisión con el fin de ahorrar ancho de banda o de incrementar las imágenes por segundo. El tipo de procesamiento de datos debe ser elegido cuidadosamente en función del objetivo de la aplicación y teniendo en cuenta la memoria disponible, la potencia de cálculo, los recursos energéticos y las limitaciones de ancho de banda.

En esta tesis se investiga cómo un sistema de visión debe ser construido teniendo en cuenta las limitaciones prácticas. En primer lugar, el sistema debe ser inteligente, de forma que se extraigan los datos apropiados de la fuente de vídeo. En segundo lugar, al procesar las señales de la fuente de vídeo este sistema de visión inteligente debe conocer sus propias limitaciones prácticas, y debería intentar lograr el mejor resultado dentro de sus posibilidades. Estudiamos y mejoramos una amplia gama de sistemas de visión para una variedad de aplicaciones, que conllevan diferentes tipos de limitaciones.

En primer lugar se presenta un algoritmo basado en la codificación módulo-PCM. Este algoritmo es muy útil para las aplicaciones que exigen una complejidad de codificación muy baja y que además necesitan conservar algunas de las ventajas de la codificación PCM (procesamiento directo, acceso aleatorio, tasa escalable). Nuestro sistema de codificación módulo-PCM combina tres estrategias de codificación conocidas: PCM, binning y codificación interpolativa. El codificador analiza primero las estadísticas de la señal de manera muy sencilla. Basándose en estas estadísticas, el codificador descarta un número de bits de cada muestra de la imagen. El decodificador módulo-PCM recupera los bits descartados de cada muestra utilizando los bits recibidos y la información lateral que se genera por interpolación de las señales decodificadas previas. Nuestro algoritmo es especialmente apropiado para la codificación de imágenes, ya que los errores de codificación que este algoritmo introduce son mayores en las regiones donde son menos visibles (los bordes y las zonas con texturas).

Desarrollamos un modelo para la distorsión de codificación introducida por

este codificador módulo-PCM. Utilizando este modelo, analizamos cómo los parámetros de codificación deben ser escogidos en función de la tasa deseada y de la calidad de la información lateral.

Los resultados experimentales obtenidos en la codificación de varias imágenes digitales muestran que nuestro algoritmo tiene un mejor rendimiento objetivo y subjetivo que PCM a tasas bajas. A tasas elevadas, módulo-PCM y PCM dan resultados similares. En cuanto a la relación tasa-distorsión, nuestro algoritmo tiene un rendimiento algo menor que otros tipos de codificación tales como la codificación módulo-PCM con información lateral o la codificación de vídeo Wyner-Ziv. Por otro lado, nuestro algoritmo tiene como ventaja una complejidad computacional mucho menor. Este hace que sea muy útil en aplicaciones que requieren codificadores extremadamente simples como por ejemplo la codificación de señales de cámaras de alta velocidad.

En segundo lugar, en algunas aplicaciones de vídeo es conveniente reducir la complejidad del codificador de vídeo a expensas de un decodificador más complejo. Ejemplos de este tipo de aplicaciones son la vigilancia con cámaras inalámbricas de bajo consumo, cámaras inalámbricas para PC, redes de sensores multimedia, cámaras desechables, y cámaras de teléfonos móviles. La codificación distribuida de vídeo es un nuevo paradigma que cumple este requisito mediante la codificación intra-frame y decodificación inter-frame. De esta forma la mayor parte de la carga de procesamiento se mueve del codificador al decodificador, ya que en este caso, los decodificadores distribuidos de vídeo (y no los codificadores) realizan la estimación de movimiento y la interpolación con compensación de movimiento. Dos teoremas de Teoría de la Información - el teorema de Slepian-Wolf para la codificación distribuida de fuente sin pérdidas y el teorema de Wyner-Ziv para la codificación de fuente con pérdidas con información lateral - sugieren que un sistema con codificación intra-frame y decodificación inter-frame puede acercarse a la eficiencia de un sistema de codificación tradicional inter-frame.

Para obtener una mejor comprensión del funcionamiento de este tipo de codificadores, comenzamos con un estudio en profundidad de la distorsión introducida por los codificadores de vídeo Wyner-Ziv actuando en el dominio del píxel. Nuestro modelo de distorsión se puede utilizar para determinar el valor óptimo de los parámetros de codificación bajo restricciones de tasa y distorsión. Como ejemplo mostramos cómo se puede utilizar nuestro modelo para reducir las fluctuaciones de calidad entre diferentes fotogramas del vídeo.

Muchos codificadores de vídeo Wyner-Ziv utilizan un canal de retorno para asignar una tasa adecuada. Sin embargo, este canal de retorno no siempre está disponible, como es el caso en la codificación offline o en aplicaciones unidireccionales. Se propone un algoritmo de asignación de tasa que permite eliminar el canal de retorno del sistema de codificación. Nuestro algoritmo calcula el número de bits para codificar cada fotograma de vídeo sin aumentar de manera significativa la complejidad del codificador. Los resultados experimentales muestran que nuestro algoritmo de asignación de tasa proporciona una buena estimación, y que la calidad de imagen proporcionada por nuestro algoritmo es bastante cercana a la proporcionada por un algoritmo con canal de retorno.

Uno de los objetivos generales de la codificación distribuida de vídeo es reducir la complejidad del codificador lo más posible, a expensas de un decodificador más complejo. En este contexto, se observa que el aumento de la complejidad del decodificador es excesivo, y por lo tanto la complejidad del proceso completo de codificación y decodificación es mucho mayor que en los sistemas tradicionales de codificación. Para superar este problema, desarrollamos un método que reduce drásticamente la complejidad del decodificador. En este método utilizamos un canal de retorno para ajustar la asignación de tasa que obtenemos con nuestro algoritmo, logrando una asignación de tasa muy cercana a la óptima. Al mismo tiempo se eliminan dos de los principales inconvenientes del canal de retorno: su impacto negativo en la latencia y la complejidad del decodificador.

En tercer lugar, se estudia en detalle cómo se debe diseñar un sistema de visión para la aplicación específica de detección de ocupación en 2D. Un mapa de ocupación en 2D proporciona una vista desde arriba de una escena que contiene a personas u objetos. Este tipo de mapas son importantes en muchas aplicaciones como vigilancia, habitaciones inteligentes, videoconferencias y análisis deportivos. Se presentan dos métodos diferentes. Con un primer método se pretende proporcionar mapas de ocupación en 2D muy precisos. Para ello, utilizamos una red de cámaras inteligentes, es decir, con potentes capacidades de procesamiento. En consecuencia, las cámaras pueden procesar y comprimir los datos de vídeo de manera inteligente antes de enviar esta información a la estación principal para el procesamiento central. En concreto, cada cámara calcula una silueta del primer plano y del plano de fondo, que transfiere a un plano de referencia utilizando transformaciones homográficas (el plano del suelo). Estas ocupaciones de suelo calculadas a partir de cada punto de vista se transmiten a una estación central de procesamiento. Puesto que la cantidad de datos necesarios para representar estas ocupaciones de suelo no es grande (mucho menor que con una imagen real), el ancho de banda requerido es más bien pequeño. En la estación de base, las ocupaciones de suelo de todas las cámaras se fusionan utilizando la teoría de la evidencia de Dempster-Shafer. El método da resultados de detección de ocupación muy precisos y supera los resultados del estado de la técnica en cálculo de mapas de ocupación con métodos multi-cámara.

Este primer método es muy preciso, pero no siempre se puede utilizar en la práctica. En particular, los principales problemas son la posibilidad de violación de la privacidad, coste elevado, costosas alteraciones de infraestructura, complejidad de procesamiento y alto consumo de energía.

Teniendo en cuenta estos requisitos, se presenta un segundo método nuevo para la detección de ocupación en 2D. En este método se sustituye la cámara por un dispositivo más específico que consiste en una línea de elementos ópticos (por ejemplo fotodiodos), que llamamos un sensor de línea. Proponemos el uso de múltiples de estos sensores de línea para calcular un mapa de ocupación en 2D preciso.

El sensor de línea es especialmente apropiado para esta aplicación gracias a su bajo precio, bajo consumo de energía, alta tasa de datos, alta profundidad de bits y el hecho de que no invade la privacidad. Proponemos usar el sensor de línea junto con un sistema de integración óptico, que garantiza que cada elemento del sensor de línea integre toda la luz dentro de un cierto rango de ángulos de incidencia. Las medidas de múltiples sensores de línea con sistema de integración óptica son muy adecuadas como entrada para un algoritmo de cálculo de la ocupación. El cálculo de un mapa de ocupación en 2D con múltiples sensores de línea da resultados precisos que se aproximan a los obtenidos con múltiples cámaras, sobre todo cuando los sensores de línea perciben la escena desde un lado y no desde arriba.

En cuarto lugar, se investiga cómo una red de visión puede manejar múltiples tareas de visión que deben llevarse a cabo simultáneamente, como por ejemplo el seguimiento de varias personas en una habitación. El número y el tipo de tareas que una red de cámaras puede manejar está por supuesto limitado por los recursos de la red. Las restricciones más importantes de la red de cámaras son la limitada potencia de cálculo de las cámaras y las limitaciones de comunicación.

En una red de cámaras práctica a cargo de múltiples tareas y con recursos de red limitados, el objetivo es lograr el mejor rendimiento mediante la distribución eficiente de las tareas entre los sensores de acuerdo con las restricciones dadas. Esta distribución de tareas entre los sensores se denomina asignación de tareas. En esta tesis, presentamos una nueva solución general para la asignación de tareas en la práctica (es decir, con restricciones de la red) para redes de visión con campos de visión solapados.

Este marco ofrece la posibilidad de controlar la calidad con que se realizan las tareas, mientras que se distribuyen las tareas entre las cámaras de acuerdo con criterios prácticos. En particular, este método supone, por una parte, funciones de coste para modelar los criterios prácticos, como por ejemplo la limitada potencia de cálculo de las cámaras. Por otra parte, utilizamos funciones de valor de idoneidad que indican con qué calidad un conjunto de cámaras puede realizar una tarea determinada, con el fin de controlar la calidad de las tareas ejecutadas. Las funciones de coste y de valor se combinan en un problema de optimización con restricciones, que tiene como solución la distribución óptima de las tareas entre las cámaras. Como prueba de concepto, utilizamos nuestro método para la gestión de múltiples tareas de seguimiento de personas. Evaluamos cómo la calidad del seguimiento está influenciada por el ancho de banda y la limitada potencia de cálculo de las cámaras en la red. Probamos nuestro método en una gran cantidad de datos reales que vienen de varios entornos donde instalamos una red de cámaras para observar la escena.

En resumen, las principales contribuciones de esta tesis son

- un algoritmo basado en módulo-PCM para la codificación con muy baja complejidad de imágenes;

- un estudio en profundidad y mejora de algoritmos de codificación distribuida de vídeo en el dominio del píxel;
- dos nuevos sistemas de visión para el cálculo preciso de mapas de ocupación en 2D;
- un sistema de asignación de tareas en redes de visión inteligentes.

La investigación llevada a cabo durante esta tesis resultó en cinco publicaciones en revistas internacionales (dos publicadas, dos en revisión, una en preparación) de las cuales tres como primer autor [Morbee et al., 2011, Prades-Nebot et al., 2010, Tessens et al., 2011, Morbee et al., 2010, Morbee et al., 2008a], dos solicitudes de patentes (registradas) como primer autor [Morbee and Tessens, 2010, Morbee and Tessens, 2011], dos capítulos en Lecture Notes of Computer Science de los cuales uno como primer autor [Lee et al., 2008, Morbee et al., 2007a], y doce publicaciones en congresos internacionales de las cuales ocho como primer autor [Morbee et al., 2009b, Morbee et al., 2009a, Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008, Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a, Morbee et al., 2006b].

Resum en valencià
–Summary in Valencian–

Els sistemes de visió s'han tornat omnipresents. S'utilitzen per al control del trànsit, tenir cura d'ancians, videoconferència, realitat virtual, vigilància, sales intel·ligents, domòtica, anàlisis esportives, seguretat industrial, assistència mèdica, etc. En la majoria dels sistemes de visió, les dades procedents dels sensors visuals es processen abans de la transmissió amb la finalitat d'estalviar ample de banda o d'incrementar les imatges per segon. El tipus de processament de dades ha de ser triat acuradament en funció de l'objectiu de l'aplicació i tenint en compte la memòria disponible, la potència de càlcul, els recursos energètics i les limitacions d'ample de banda.

En aquesta tesi s'investiga com un sistema de visió ha de ser construït tenint en compte les limitacions pràctiques. En primer lloc, el sistema ha de ser intel·ligent, de manera que s'extraguen les dades apropiades de la font de vídeo. En segon lloc, en processar els senyals de la font de vídeo, aquest sistema de visió intel·ligent ha de conèixer les seues pròpies limitacions pràctiques, i hauria d'intentar aconseguir el millor resultat dins de les seues possibilitats. Estudiem i millorem una àmplia gamma de sistemes de visió per a una varietat d'aplicacions, que comporten diversos tipus de limitacions.

En primer lloc, es presenta un algorisme basat en la codificació mòdul-PCM. Aquest algorisme és molt útil per a les aplicacions que exigeixen una complexitat de codificació molt baixa i que a més necessiten conservar alguns dels avantatges de la codificació PCM (processament directe, accés aleatori, taxa escalable). El nostre sistema de codificació mòdul-PCM combina tres estratègies de codificació conegudes: PCM, binning i codificació interpolativa. El codificador analitza primer les estadístiques del senyal de manera molt senzilla. Basant-se en aquestes estadístiques, el codificador descarta un nombre de bits de cada mostra de la imatge. El descodificador mòdul-PCM recupera els bits descartats de cada mostra utilitzant els bits rebuts i la informació lateral que es genera per interpolació dels senyals descodificats previs. El nostre algorisme és especialment apropiat per a la codificació d'imatges, ja que els errors de codificació que aquest algorisme introdueix són majors en les regions on són menys visibles (les vores i les zones amb textures).

Desenvolupem un model per a la distorsió de codificació introduïda per aquest

codificador mòdul-PCM. Utilitzant aquest model, analitzem com els paràmetres de codificació han de ser escollits en funció de la taxa desitjada i de la qualitat de la informació lateral.

Els resultats experimentals obtinguts en la codificació de diverses imatges digitals mostren que el nostre algorisme té un millor rendiment objectiu i subjectiu que PCM a taxes baixes. A taxes elevades, mòdul-PCM i PCM donen resultats similars. Quant a la relació taxa-distorsió, el nostre algorisme té un rendiment un poc menor que altres tipus de codificació, com ara la codificació de mòdul-PCM amb informació lateral o la codificació de vídeo Wyner-Ziv. D'altra banda, el nostre algorisme té com a avantatge una complexitat computacional molt menor. Això fa que siga molt útil en aplicacions que requereixen codificadors extremadament simples, com per exemple la codificació de senyals de càmeres d'alta velocitat.

En segon lloc, en algunes aplicacions de vídeo és convenient reduir la complexitat del codificador de vídeo a costa d'un descodificador més complex. Exemples d'aquest tipus d'aplicacions són la vigilància amb càmeres sense fils de baix consum, càmeres sense fils per a PC, xarxes de sensors multimèdia, càmeres d'un sol ús, i càmeres de telèfons mòbils. La codificació distribuïda de vídeo és un nou paradigma que compleix aquest requisit mitjançant la codificació intraquadre (intraframe) i descodificació interquadre (interframe). D'aquesta manera, la major part de la càrrega de processament es mou del codificador al descodificador, ja que en aquest cas, els descodificadors distribuïts de vídeo (i no els codificadors) realitzen l'estimació de moviment i la interpolació amb compensació de moviment. Dos teoremes de teoria de la informació –el teorema de Slepian-Wolf per a la codificació distribuïda de font sense pèrdues i el teorema de Wyner-Ziv per a la codificació de font amb pèrdues amb informació lateral– suggereixen que un sistema amb codificació intraquadre i descodificació interquadre pot acostar-se a l'eficiència d'un sistema de codificació tradicional interquadre.

Per a obtenir una millor comprensió del funcionament d'aquest tipus de codificadors, comencem amb un estudi en profunditat de la distorsió introduïda pels codificadors de vídeo Wyner-Ziv actuant en el domini del píxel. El nostre model de distorsió es pot utilitzar per a determinar el valor òptim dels paràmetres de codificació sota restriccions de taxa i distorsió. Com a exemple, mostrem com es pot utilitzar el nostre model per a reduir les fluctuacions de qualitat entre diferents fotogrames del vídeo.

Molts codificadors de vídeo Wyner-Ziv utilitzen un canal de tornada per a assignar una taxa adequada. No obstant això, aquest canal de tornada no sempre està disponible, com és el cas en la codificació fora de línia (offline) o en aplicacions unidireccionals. Es proposa un algorisme d'assignació de taxa que permet eliminar el canal de tornada del sistema de codificació. El nostre algorisme calcula el nombre de bits per a codificar cada fotograma de vídeo sense augmentar de manera significativa la complexitat del codificador. Els resultats experimentals mostren que el nostre algorisme d'assignació de taxa proporciona una bona estimació, i que la qualitat d'imatge proporcionada pel nostre algorisme és bastant propera a la proporcionada per un algorisme amb canal de tornada.

Un dels objectius generals de la codificació distribuïda de vídeo és reduir la complexitat del codificador tant com siga possible, a costa d'un descodificador més complex. En aquest context, s'observa que l'augment de la complexitat del descodificador és excessiu, i per tant la complexitat del procés complet de codificació i descodificació és molt major que en els sistemes tradicionals de codificació. Per a superar aquest problema, desenvolupem un mètode que redueix dràsticament la complexitat del descodificador. En aquest mètode, utilitzem un canal de tornada per a ajustar l'assignació de taxa que obtenim amb el nostre algorisme, i aconseguim una assignació de taxa molt propera a l'òptima. Al mateix temps, s'eliminen dos dels principals inconvenients del canal de tornada: l'impacte negatiu en la latència i la complexitat del descodificador.

En tercer lloc, s'estudia detalladament com s'ha de dissenyar un sistema de visió per a l'aplicació específica de detecció d'ocupació en 2D. Un mapa d'ocupació en 2D proporciona una vista des de dalt d'una escena que conté persones o objectes. Aquesta mena de mapes són importants en moltes aplicacions, com ara vigilància, habitacions intel·ligents, videoconferències i anàlisis esportives. Es presenten dos mètodes diferents. Amb un primer mètode es pretén proporcionar mapes d'ocupació en 2D molt precisos. Per a fer-ho, utilitzem una xarxa de càmeres intel·ligents, és a dir, amb capacitats de processament potents. En conseqüència, les càmeres poden processar i comprimir les dades de vídeo de manera intel·ligent abans d'enviar aquesta informació a l'estació principal per al processament central. En concret, cada càmera calcula una silueta del primer pla i del pla de fons, que transfereix a un pla de referència utilitzant transformacions homogràfiques (el pla del sòl). Aquestes ocupacions de sòl calculades a partir de cada punt de vista es transmeten a una estació central de processament, ja que la quantitat de dades necessàries per a representar aquestes ocupacions de sòl no és gran (molt menor que amb una imatge real), l'ample de banda requerit és més aviat menut. En l'estació de base, les ocupacions de sòl de totes les càmeres es fusionen utilitzant la teoria de l'evidència de Dempster-Shafer. El mètode dóna resultats de detecció d'ocupació molt precisos i supera els resultats de l'estat de la tècnica en càlcul de mapes d'ocupació amb mètodes multicàmera.

Aquest primer mètode és molt precís, però no sempre es pot utilitzar en la pràctica. En particular, els principals problemes són la possibilitat de violació de la privadesa, el cost elevat, les costoses alteracions d'infraestructura, la complexitat de processament i l'alt consum d'energia.

Tenint en compte aquests requisits, es presenta un segon mètode nou per a la detecció d'ocupació en 2D. En aquest mètode se substitueix la càmera per un dispositiu més específic que consisteix en una línia d'elements òptics (per exemple fotodíodes), que anomenem un sensor de línia. Proposem l'ús de múltiples d'aquests sensors de línia per a calcular un mapa d'ocupació en 2D precís.

El sensor de línia és especialment apropiat per a aquesta aplicació gràcies al seu baix preu, baix consum d'energia, alta taxa de dades, gran profunditat de bits, i pel fet que no envaeix la privadesa. Proposem usar el sensor de línia juntament amb el sistema d'integració òptic, que garanteix que cada element del sensor de línia reba (una integració de) tota la llum dins d'un cert rang d'angles d'incidència. Les mesures de múltiples sensors de línia amb sistema d'integració òptica són molt adequades com a entrada per a un algorisme de càlcul de l'ocupació. El càlcul d'un mapa d'ocupació en 2D amb múltiples sensors de línia dóna resultats precisos que s'aproximen als obtinguts amb múltiples càmeres, sobretot quan els sensors de línia perceben l'escena des d'un costat i no des de dalt.

En quart lloc, s'investiga com una xarxa de visió pot manejar múltiples tasques de visió que han de dur-se a terme simultàniament, com per exemple el seguiment de diverses persones en una habitació. El nombre i el tipus de tasques que una xarxa de càmeres pot manejar està per descomptat limitat pels recursos de la xarxa. Les restriccions més importants de la xarxa de càmeres són la limitada potència de càlcul de les càmeres i les limitacions de comunicació.

En una xarxa de càmeres pràctica a càrrec de múltiples tasques i amb recursos de xarxa limitats, l'objectiu és aconseguir el millor rendiment de tasques mitjançant la distribució eficient de les tasques entre els sensors d'acord amb les restriccions donades. Aquesta distribució de tasques entre els sensors es diu assignació de tasques. En aquesta tesi, presentem una nova solució general per a l'assignació de tasques en la pràctica (és a dir, amb restriccions de la xarxa) per a xarxes de visió amb camps de visió encavalcats.

Aquest marc ofereix la possibilitat de controlar la qualitat amb què es duen a terme les tasques, mentre que es distribueixen les tasques entre les càmeres d'acord amb criteris pràctics. En particular, aquest mètode suposa, d'una banda, funcions de cost per a modelar els criteris pràctics, com per exemple la limitada potència de càlcul de les càmeres. D'altra banda, utilitzem funcions de valor d'idoneïtat que indiquen amb quina qualitat un conjunt de càmeres pot fer una tasca determinada, amb la finalitat de controlar la qualitat de les tasques executades. Les funcions de cost i de valor es combinen en un problema d'optimització amb restriccions, que té com a solució la distribució òptima de les tasques entre les càmeres. Com una prova de concepte, utilitzem el nostre mètode per a la gestió de múltiples tasques de seguiment de persona. Avaluem com la qualitat del seguiment està influenciada per l'ample de banda i la limitada potència de càlcul de les càmeres en la xarxa. Provem el nostre mètode en una gran quantitat de dades reals que vénen de diversos entorns on instal·lem una xarxa de càmeres per a observar l'escena.

En resum, les principals contribucions d'aquesta tesi són

- un algorisme basat en mòdul-PCM per a la codificació d'imatges amb molt baixa complexitat;
- un estudi en profunditat i la millora d'algorismes de codificació distribuïda de vídeo en el domini del píxel;
- dos nous sistemes de visió per al càlcul precís de mapes d'ocupació en 2D;

- un sistema d'assignació de tasques en xarxes de visió intel·ligents.

La investigació duta a terme durant aquesta tesi va resultar en cinc publicacions en revistes internacionals (dues ja estan publicades, dues estan en revisió, una en preparació), de les quals tres com a primer autor [Morbee et al., 2011, Prades-Nebot et al., 2010, Tessens et al., 2011, Morbee et al., 2010, Morbee et al., 2008a], dues sol·licituds de patents (registrades) com a primer autor [Morbee and Tessens, 2010, Morbee and Tessens, 2011], dos capítols en Lecture Notes of Computer Science, dels quals un com a primer autor [Lee et al., 2008, Morbee et al., 2007a], i dotze publicacions en congressos internacionals, de les quals vuit com a primer autor [Morbee et al., 2009b, Morbee et al., 2009a, Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008, Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a, Morbee et al., 2006b].

Summary in English

Vision systems have become ubiquitous. They are used for traffic monitoring, elderly care, video conferencing, virtual reality, surveillance, smart rooms, home automation, sport games analysis, industrial safety, medical care etc. In most vision systems, the data coming from the visual sensor(s) is processed before transmission in order to save communication bandwidth or achieve higher frame rates. The type of data processing needs to be chosen carefully depending on the targeted application, and taking into account the available memory, computational power, energy resources and bandwidth constraints.

In this dissertation, we investigate how a vision system should be built under practical constraints. First, this system should be intelligent, such that the right data is extracted from the video source. Second, when processing video data this intelligent vision system should know its own practical limitations, and should try to achieve the best possible output result that lies within its capabilities. We study and improve a wide range of vision systems for a variety of applications, which go together with different types of constraints.

First, we present a modulo-PCM-based coding algorithm for applications that demand very low complexity coding and need to preserve some of the advantageous properties of PCM coding (direct processing, random access, rate scalability). Our modulo-PCM coding scheme combines three well-known, simple, source coding strategies: PCM, binning, and interpolative coding. The encoder first analyzes the signal statistics in a very simple way. Then, based on these signal statistics, the encoder simply discards a number of bits of each image sample. The modulo-PCM decoder recovers the removed bits of each sample by using its received bits and side information which is generated by interpolating previous decoded signals. Our algorithm is especially appropriate for image coding since it introduces larger coding errors in those regions where it is less visible (edges and textured regions).

We develop a model for the coding distortion introduced by this modulo-PCM coder. Using this model, we analyze how the coding parameters should be chosen as a function of the target rate and the quality of the side information.

Experimental results obtained in the encoding of several digital images show that our algorithm has a better objective and subjective performance than PCM at low rates. At high rates, Modulo-PCM and PCM provide similar results. Our algorithm has a worse rate-distortion performance than other source coding techniques

such as modulo-PCM coding with side information or Wyner-Ziv video coding, but it has the advantage of a much lower computational complexity (comparable to PCM). This makes our algorithm very useful in applications that require extremely simple encoders such as the encoding of video signals from high-speed cameras.

Second, in some video applications, it is desirable to reduce the complexity of the video encoder at the expense of a more complex decoder. Examples of such applications are wireless low-power surveillance, wireless PC cameras, multimedia sensor networks, disposable cameras, and mobile camera phones. Distributed video coding is a new paradigm that fulfills this requirement by performing intra-frame encoding and inter-frame decoding. Hence, most of the computational load is moved from the encoder to the decoder, since in this case the distributed video *de*coders (and not the *en*coders) perform motion estimation and motion compensated interpolation. Two theorems from information theory, namely the Slepian-Wolf theorem for lossless distributed source coding and the Wyner-Ziv theorem for lossy source coding with side information, suggest that such a system with intra-frame encoding and inter-frame decoding can come close to the efficiency of a traditional inter-frame encoding-decoding system.

To get a better insight into the functioning of this type of coders, we start with an in-depth study of the coding distortion introduced by pixel-domain Wyner-Ziv video coders. Our coding distortion model can be used to determine the optimal value of coding parameters under rate and distortion constraints. As an example, we show how our model can be used to reduce quality fluctuations between different frames of the video.

Many Wyner-Ziv video coders make use of a feedback channel to allocate an appropriate rate. However, this feedback channel is not always available, as is the case in offline coding or in unidirectional applications. We propose a rate allocation algorithm that allows to remove the feedback channel from the coding scheme. Our algorithm computes the number of bits to encode each video frame without significantly increasing the encoder complexity. Experimental results show that our rate allocation algorithm delivers good estimates of the rate, and that the frame qualities provided by our algorithm are quite close to the ones provided by a feedback channel-based algorithm.

A general aim in distributed video coding is to reduce the complexity of the encoder as much as possible, but this is of course at the expense of more decoder complexity. In this respect, we observe that this increase of the decoder complexity is excessive, and hence the complexity of the entire coding process is much higher than in traditional coding schemes. To overcome this problem, we develop a method that reduces the decoder complexity drastically. In this method, we utilize a feedback channel to fine-tune the rate allocation of our rate allocation algorithm and to achieve very near-to-optimal rate allocation, while we eliminate at the same time two main feedback channel inconveniences, i.e., its negative impact on latency and decoder complexity.

Third, we study in detail how a vision system for the specific application of 2D occupancy sensing should be designed. A 2D occupancy map provides an abstract top view of a scene containing people or objects. Such maps are important in many applications such as surveillance, smart rooms, video conferencing and sport games analysis. We present two different methods. With a first method we aim at providing very accurate 2D occupancy maps. For this, we use a network of *smart* cameras, which means that the cameras have strong on-board processing capabilities. Consequently, the cameras can process and compress the video data in an intelligent way before sending it to the base station for central processing. In particular, each camera calculates a foreground (FG)/background (BG) silhouette and transfers this silhouette to a reference plane using its camera image-floor homographies. These ground occupancies computed from each view are transmitted to a central processing station. Since the data amount needed to represent these ground occupancies is not large (much smaller than the data amount needed for a typical natural image), the required band width is rather small. At the base station, the ground occupancies from the cameras are fused using the Dempster-Shafer theory of evidence. The method yields very accurate occupancy detection results and outperforms the other state-of-the-art multi-camera 2D occupancy calculation methods.

This first method is very accurate but cannot always be used due to practical limitations. The major concerns are the possibility of privacy breach, the high cost price, the expensive alterations to the infrastructure, the high-complexity processing and the large power consumption.

Taking these requirements into consideration, we present a second novel method for 2D occupancy sensing. In this method, we replace the camera by a more specific device consisting of a linear array of optical sensing elements (e.g. photodiodes), which we call a line sensor. We propose to use multiple of these line sensors to calculate an accurate 2D occupancy map. The line sensor is particularly suited for this application due to its low price, its low-power consumption, its high data rates, its high bit depth and its privacy-friendly nature. We propose to use the line sensor together with a light-integrating optical system, which ensures that each sensing element integrates all light within a certain range of incidence angles. The scan line outputs from multiple light-integrating line sensors are very well suited as input for a 2D occupancy calculation algorithm. Occupancy calculation with light-integrating line sensors yields accurate results that approximate quite closely the results obtained with cameras, especially when the line sensors view the scene from aside and not from above.

Fourth, we investigate how a vision network can deal with many vision tasks that need to be performed simultaneously, e.g. the tracking of multiple persons in a room. The number and the type of tasks a camera network can handle is of course limited by the network resources. The most important camera network restrictions are the limited computational power of the cameras and the communication constraints.

In a practical multi-camera network charged with multiple tasks and with restricted network resources the aim is to achieve the best overall task performance by distributing the tasks in an efficient way among the sensors in accordance with the given restrictions. This distribution of tasks among the sensors is called task assignment. In this dissertation, we present a novel, general solution to task assignment in practical (i.e. with network restrictions) vision networks with overlapping fields of view.

This framework offers the possibility of controlling the quality with which tasks are performed, while distributing the tasks among the cameras according to practical criteria. In particular, this framework entails on the one hand *cost functions* to model the practical criteria, such as for example the limited computational power of the cameras. On the other hand, we use *suitability value functions* that indicate how well a set of cameras can perform a certain task, in order to monitor the quality of the executed tasks. The cost and value functions are combined in a constrained optimization problem, which has as solution the optimal distribution of the tasks over the cameras.

As a proof of concept, we use our method for the management of multiple person-tracking tasks. We evaluate how the tracking performance is influenced by bandwidth and computational constraints in the network. We test our method on extensive real data from different camera network environments.

To summarize, the main contributions of this dissertation are

- a modulo-PCM based coding algorithm for very low complexity coding of images;
- a thorough study and improvement of pixel-domain distributed video coding algorithms;
- two novel vision systems for calculating accurate 2D occupancy maps;
- a task assignment framework for intelligent vision networks.

The research performed during this PhD resulted in five international journal publications (two published, two under review, one in preparation) of which three as first author [Morbee et al., 2011, Prades-Nebot et al., 2010, Tessens et al., 2011, Morbee et al., 2010, Morbee et al., 2008a], two (submitted) patent applications as first author [Morbee and Tessens, 2010, Morbee and Tessens, 2011], two chapters in Lecture Notes of Computer Science of which one as first author [Lee et al., 2008, Morbee et al., 2007a], and twelve publications at international conferences of which eight as first author [Morbee et al., 2009b, Morbee et al., 2009a, Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008, Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a, Morbee et al., 2006b].

Table of Contents

1
Introduction

In recent years, vision systems have become ubiquitous. They are used for traffic monitoring, elderly care, video conferencing, virtual reality, surveillance, smart rooms, home automation, sport games analysis, industrial safety, medical care etc. In most vision systems, the data coming from the visual sensor(s) is processed before transmission in order to save communication bandwidth or achieve higher frame rates. The type of data processing needs to be chosen carefully, taking into account the available memory, computational power, energy resources and bandwidth constraints. In some applications, like mobile video telephony, high-speed cameras, disposable cameras or wireless sensor networks, a very simple and fast coder of the raw visual data might be the best choice. In other situations, like on industrial sites or for video surveillance, it is beneficial to analyse the video data and convert it into a semantic scene interpretation before transmission.

In this dissertation, we study a selection of intelligent vision systems. All these systems aim to process and interpret the image data in a smart way, while taking the specific demands and constraints of the targeted application into consideration. The proposed algorithms range from very simple, intelligent bit selection that can be used for fast compression in high-speed cameras, to high-level scene analysis with a network of smart cameras that have strong on-board processing capabilities. In the following chapters, the techniques will be discussed in detail and compared with the state-of-the-art.

In the next section, Section 1.1, we will start with an overview of the intelligent vision systems, which will be treated in detail in this dissertation in Chapters 2, 3, 4 and 5. Then, in Section 1.2, we will give a brief summary of the scientific publications that resulted from this dissertation. Finally, in Section 1.3, we give an outline of the remainder of this PhD.

1.1 Overview of the contributions

In this dissertation, we discuss in detail a series of intelligent vision systems. These systems reduce the amount of image and video data in a way that is as intelligent as the application and practical processing and communication constraints allow it to be. We have organized the algorithms according to their complexity level. We start from a very simple low-level coding technique and move on to high-level task assignment in smart camera networks. The techniques are briefly introduced in the following sections, Sections 1.1.1, 1.1.2, 1.1.3, and 1.1.4.

1.1.1 Very low complexity coding of images using Modulo-PCM

We start with a very basic processing algorithm for image data: an encoder that simply selects bits from the digital source and does not perform any numeric operation after the coding parameters have been assigned. This algorithm is very useful in applications with very limiting constraints on the encoder concerning computational power and/or energy. This situation occurs for instance in sensor networks [Xiong et al., 2004, Puri et al., 2006], or when the image signal must be acquired at a very high sampling rate, e.g. for the accurate representation of human motion [Kurita et al., 2005, Ueno et al., 2006], tomography [Tipnis et al., 2001], analysis of vocal fold vibration [Tao et al., 2007], recording of fast physical phenomena [Sekikawa and Kubono, 2007], and tracking [Muehlmann et al., 2004, Gemeiner et al., 2007]. For such applications, source coding algorithms must be extremely simple and, consequently, many times video must be recorded or transmitted in PCM (*raw video*). However, the high rate of raw video can make its transmission or storage difficult. Therefore, we propose a Modulo-PCM based coding algorithm, that preserves the advantageous properties of PCM coding (such as direct processing, random access and rate scalability), but achieves better rate-distortion performance. Our coding scheme combines three well-known simple coding techniques: PCM, binning and interpolative coding. The algorithm is especially appropriate for image coding since it introduces larger coding errors in those regions where it is less visible. Experimental results obtained in the encoding of several digital images show that our algorithm has a better objective and subjective performance than PCM at low rates. At high rates, Modulo-PCM and PCM provide similar results. Our algorithm performs slightly worse than other source coding techniques such as MPCM coding with side information or Wyner-Ziv video coding, but it has the advantage of a much lower computational complexity. This makes our algorithm very useful in applications that require extremely simple encoders such as the encoding of video signals from high-speed cameras.

1.1.2 Pixel-domain distributed video coding

Compared to Modulo-PCM, we slightly increase the complexity of the data processing unit, with the aim of getting a better rate-distortion performance. However, we still want low-complexity visual data processing to be key. Two theorems from information theory, namely the Slepian-Wolf theorem from 1973 [Slepian and Wolf, 1973] for lossless distributed source coding and the Wyner-Ziv theorem from 1976 [Wyner and Ziv, 1976] for lossy source coding with side information, are very useful in this respect. If we apply these principles to video coding, these theorems suggest that a system with intra-frame encoding and inter-frame decoding (called a distributed video coder) can come close to the efficiency of a traditional inter-frame encoding-decoding system. Hence, in distributed video coding (DVC) most of the computational load is moved from the encoder to the decoder, since in this case the distributed video *de*coders (and not the *en*coders) perform motion estimation and motion compensated interpolation.

By applying DVC techniques, we obtain a coder for vision systems with specific constraints: limited bandwidth, scarce energy resources, necessity of robustness against transmission errors, presence or absence of a feedback channel. In particular, we study and develop a pixel-domain distributed video coder. As a starting point, we use the distributed video coding scheme of [Ascenso et al., 2005a, Dalai et al., 2006a, Belkoura and Sikora, 2006b, Brites et al., 2006a, Trapanese et al., 2005,Morbee et al., 2007b,Ascenso et al., 2005b], as it is well-known in literature. Different aspects of this scheme are questioned and improved.

First, we get a better insight into the functioning of the developed pixel-domain distributed coder. We draw up a model for the distortion introduced by the coder and use this model to reduce quality fluctuations between different frames of the video. This study was performed in close collaboration with Antoni Roca.

Many Wyner-Ziv video coders make use of a feedback channel. However, this feedback channel is not always available, as is the case in offline coding or in unidirectional applications. An adequate rate allocation algorithm allows us to remove the feedback channel from the starting scheme.

Finally, due to the memory and energy limitations, a general aim in DVC is to reduce the complexity of the encoder as much as possible, but this is of course at the expense of more decoder complexity. In this respect, we observed that this increase of the decoder complexity is excessive, and hence the complexity of the entire coding process is much higher than in traditional coding schemes. To overcome this problem, we developed a method that reduces the decoder complexity drastically.

1.1.3 Vision systems for 2D occupancy sensing

We move from the pure information-theoretic approach of the previous methods towards more content- and application-aware data processing. We study in detail how a vision system for the specific application of 2D occupancy sensing should be designed. A 2D occupancy map provides an abstract top view of a scene containing people or objects. Such maps are important in many applications such as surveillance, smart rooms, video conferencing and sport games analysis. We present a novel method for calculating 2D occupancy maps with a set of calibrated and synchronized cameras. In particular, each camera calculates a foreground (FG)/background (BG) silhouette and transfers this silhouette to a reference plane using its camera image-floor homographies. These ground occupancies computed from each view are transmitted to a central processing station, where they are fused using the Dempster-Shafer theory of evidence. The method yields very accurate occupancy detection results and outperforms the state-of-the-art probabilistic occupancy map method [Fleuret et al., 2008] and fusion by summing [Delannay et al., 2009].

In a next step, we replace the camera by a more specific device consisting of a linear array of optical sensing elements (e.g. photodiodes), which we will call a line sensor. We propose to use multiple of these line sensors (each viewing a scene from a different direction) to calculate an accurate 2D occupancy map. The line sensor is particularly suited for this task due to its low price, its low-power consumption, its high data rates, its high bit depth and its privacy-friendly nature. We propose to use the line sensor together with a light-integrating optical system, which ensures that each sensing element integrates all light within a certain range of incidence angles. The data coming from the light-integrating line sensor will be called a scan line. These scan lines from multiple sensors are very well suited as input for a 2D occupancy calculation algorithm.

Occupancy calculation with light-integrating line sensors yields accurate results that approximate quite closely the results obtained with cameras, especially when the line sensors view the scene from aside and not from above. Additionally, the system with line sensors can profit from the more interesting characteristics of a line sensor compared to a regular camera, as will be discussed in detail in Chapter 4.

This research part has been performed in close collaboration with my colleague Linda Tessens and therefore the subject of this chapter is related to some concepts from her PhD thesis [Tessens, 2010]. However, in her work, the focus was mainly on the different fusion techniques that can be used in multi-camera systems to combine the single-view maps. One of these techniques will be used and presented in this work, but our emphasis will lie on the study of the usage of different sensor types (cameras, line sensors) and different data output types from these sensors (full images, scan lines from full images, scan lines from light-integrating line sensors). We make an overall comparison between the different systems in terms of the obtained occupancy map quality, the memory and computational requirements, the price of the system, its power consumption and its privacy-friendliness.

1.1.4 Task assignment in intelligent vision networks

In Chapter 5, we go even further and design a framework for an intelligent vision network that takes care of several tasks at the same time, e.g. tracking people, gathering scene activity statistics, detecting abnormal events, human motion analysis etc. The challenge of such an intelligent camera network is to appropriately allocate available network resources such that the best possible overall performance for the allotted tasks is achieved.

We propose a general framework for task assignment in vision networks, that can be applied to any combination of scene-related tasks. This framework offers the possibility of controlling the quality with which tasks are performed, while distributing the tasks among the cameras according to practical criteria. In particular, this framework entails on the one hand *cost functions* to model the practical criteria, such as for example the limited computational power of the cameras. On the other hand, we use *suitability value functions* that indicate how well a set of cameras can perform a certain task, in order to monitor the quality of the executed tasks. The cost and value functions are combined in a constrained optimization problem, which has as solution the optimal distribution of the tasks over the cameras.

As a proof of concept, we use our method for the management of multiple person-tracking tasks. We evaluate how the tracking performance is influenced by bandwidth and computational constraints in the network. We test our method on extensive real data from different camera network environments.

Some of the concepts of this chapter are related to the work described in the PhD of Linda Tessens [Tessens, 2010], with whom I worked together on this topic. In particular, the mentioned suitability value has been extensively treated in her PhD and will therefore not be discussed in detail in this dissertation. In the PhD of Linda Tessens [Tessens, 2010], however, the focus is on determining for *one* task which cameras are most suited to perform it. In this chapter, we go a step further. We investigate how we can efficiently distribute *a plurality* of tasks among the network cameras, taking into account that the camera network is constrained in terms of computational power and communication capabilities.

1.2 Summary of the scientific output

This dissertation resulted in

- five international journal publications (two published, two under review, one in preparation) of which three as first author [Morbee et al., 2011, Prades-Nebot et al., 2010, Tessens et al., 2011, Morbee et al., 2010, Morbee et al., 2008a],
- two (submitted) patent applications as first author [Morbee and Tessens, 2010, Morbee and Tessens, 2011],
- two chapters in Lecture Notes of Computer Science of which one as first author [Lee et al., 2008, Morbee et al., 2007a],
- twelve publications at international conferences of which eight as first author [Morbee et al., 2009b, Morbee et al., 2009a, Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008, Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a, Morbee et al., 2006b],
- a demonstrator showing real-time 2D occupancy sensing with a network of four cameras [Tessens and Morbee, 2010].

1.3 Outline

In Chapter 2, we study Modulo-PCM coding for very low complexity coding of images. Subsequently, we improve the rate-distortion performance of this Modulo-PCM coder in Chapter 3 by adding channel codes to the scheme. This leads to a slightly more complex coder, called a pixel-domain distributed video coder. For this coder, we propose a distortion model and rate allocation methods for feedback channel removal and decoder complexity reduction. In Chapter 4, we move from the information-theoretic data coding of Chapter 2 and Chapter 3 towards application-aware data processing. In particular, we develop vision systems for 2D occupancy sensing. In Chapter 5, we develop a general framework for an intelligent vision network that takes care of several tasks at the same time. In these systems, it is indispensable to appropriately allocate available network resources such that the best possible overall performance for the allotted tasks is achieved. The task assignment method of Chapter 5 formulates a solution to this and can be applied to any combination of scene-related tasks. Finally, the conclusions are formulated in Chapter 6.

2

Very Low Complexity Coding of Images Using Modulo-PCM

2.1 Introduction

Today, most signals of interest are efficiently transmitted or stored using a source coding algorithm. Among all the existing source coding algorithms, Pulse Code Modulation (PCM) is the simplest technique [Jayant and Noll, 1984]. Some PCM coders, e.g. A/D converters, use uniform quantization and fixed-length binary coding. Although this quantization/coding choice is generally not optimal in coding efficiency, it offers several advantages. First, the numeric equivalent of a sample codeword is proportional to the midpoint of the quantization interval that corresponds to the analog value of the sample. Hence, the PCM signal can be numerically processed (e.g., filtered, modulated) without a previous decoding (*direct processing property*). Second, since all the codewords have the same length, it is trivial to know the position of a sample codeword in the bit stream (*random access property*[1]). Third, if an embedded quantization scheme is used, then different rates and distortions can be achieved by simply discarding a fixed number of bits of each codeword (*rate scalability property*). In the rest of this chapter, the term PCM will refer to PCM coding with all the previously mentioned properties and

[1] In this chapter, the term *random access* refers to random access on a pixel level. Note that in video coding the term *random access* can also refer to other types of random access, such as for example random access on a macro block level or a frame level.

that dequantizes using the midpoint of the quantization intervals.[2]

Despite all these advantages, PCM is rarely used in the storing or transmission of signals in bandwidth-constrained applications due to its poor coding efficiency. For this reason, after the A/D conversion of the analog signal at a high rate R_0, the resulting PCM signal is usually compressed using a sophisticated and efficient source coding algorithm.

Apart from bandwidth efficiency, encoding simplicity is also an important factor to take into account in applications with very limiting constraints on computational power and/or energy. For instance, in sensor networks, energy-constrained and very simple processors must perform the acquisition, coding and transmission of the sensed signals [Xiong et al., 2004]. A simple encoder is also necessary when the signal must be acquired at a very high sampling rate since the average number of operations per sample that the encoder can perform is small. This happens in applications that require capturing video at very high frame rates. Examples of these applications are: the accurate representation of human motion [Kurita et al., 2005, Ueno et al., 2006], tomography [Tipnis et al., 2001], analysis of vocal fold vibration [Tao et al., 2007], recording of fast physical phenomena [Sekikawa and Kubono, 2007], analysis of movement in sports [Shum and Komura, 2005], and tracking [Gemeiner et al., 2007].

For these video applications, *high-speed cameras* that capture video at frame rates of 1,000,000 frames/second or more have been developed [Etoh et al., 2003, El-Desouki et al., 2009]. With such high frame rates, source coding algorithms must be extremely simple and, consequently, many times the video is recorded or transmitted using PCM (raw video). However, the high bit rate of raw video can make its transmission or storage difficult. Thus, the bus may not be fast enough to transfer the video out of the camera, or the writing speed of the storage device may not be high enough to save the video [Gemeiner et al., 2007].

In all these video applications, it would be interesting to be able to reduce the rate of the PCM signal *in a very simple way* without losing the advantages provided by PCM coding. Note that these constraints discard most of the existing image coding algorithms (e.g., predictive coding or transform coding) since they usually involve some numerical processing at the encoder and/or the use of variable length coding (which destroys the rate scalability and random access properties).

In this chapter, we propose a Modulo-PCM (MPCM) compression algorithm that reduces the rate of the PCM signals in a very simple way and preserves the random access and rate scalability properties of the PCM bit stream. In our algorithm, the encoder divides the PCM signal $x[n]$ into N signals $x_k[n]$ ($k \in \{0, \dots, N-1\}$) and removes the l_k least significant bits (LSBs) and the m_k most significant bits (MSBs) of each signal $x_k[n]$. The decoder recovers the bits that were removed

[2] Apart from using uniform quantization and fixed-length binary coding, a proper binary code must be chosen to fulfill the direct processing and rate scalability properties.

from each PCM codeword by using the received codeword and side information (SI) that is generated by interpolating previously decoded codewords. Note that our MPCM encoder simply removes a set of bits from each PCM codeword, and can further reduce the rate of a MPCM bit stream by removing more LSBs and/or MSBs from each MPCM codeword. Also note that our MPCM decoder can easily access the codeword of any sample if it knows the values of l_k and m_k used by the encoder. Hence, our algorithm is very simple and preserves the rate scalability and random access properties of PCM. To assign proper values to the coding parameters l_k and m_k, the MPCM encoder analyzes the statistics of the input signal. Although this analysis increases the complexity of the encoder, the overall complexity of our algorithm is much smaller than that of most source coding techniques.

Our algorithm is a hybrid coding technique that combines three simple coding strategies: PCM, binning [Cover and Thomas, 1991], and interpolative coding [Zeng and Venetsanopoulos, 1993, Bruckstein et al., 2003]. Our algorithm is also related to other techniques such as the MPCM-based coders described in [Ericson and Ramamoorthy, 1979, Ramamoorthy, 1981] and Pixel-Domain Wyner-Ziv (PDWZ) video coding [Aaron et al., 2002, Aaron et al., 2003, Ascenso et al., 2005b]. Even though these and other techniques have a higher coding efficiency than our algorithm, they are also more complex (as will be discussed in Section 2.6). Our algorithm aims to achieve the maximum coding efficiency while keeping a complexity similar to the complexity of PCM.

The rest of the chapter is organized as follows. In Section 2.2, we describe our coding algorithm and compare it with other related coding techniques. In Section 2.3, we develop a theoretical model of the distortion introduced by a MPCM coder as a function of the rate and the quality of the SI. In Section 2.4, we use this model to analyze how the optimum values of the coding parameters vary depending on the rate and the SI quality, and we propose a simple method to assign values to the coding parameters. In Section 2.5, we analyze the computational complexity of our algorithm. In Section 2.6, we experimentally test the efficiency of our algorithm for the compression of images and compare it to other source coding techniques. Finally, in Section 2.7, we summarize our results.

2.2 MPCM coding

In this section, we describe our MPCM coding algorithm and compare it with other techniques. First, we describe the MPCM coding of one-dimensional signals (Section 2.2.1). Then, we extend our algorithm to the encoding of images (Section 2.2.2). Finally, we comment on the similarities between our algorithm and other well-known low-complexity coding techniques (Section 2.2.3).

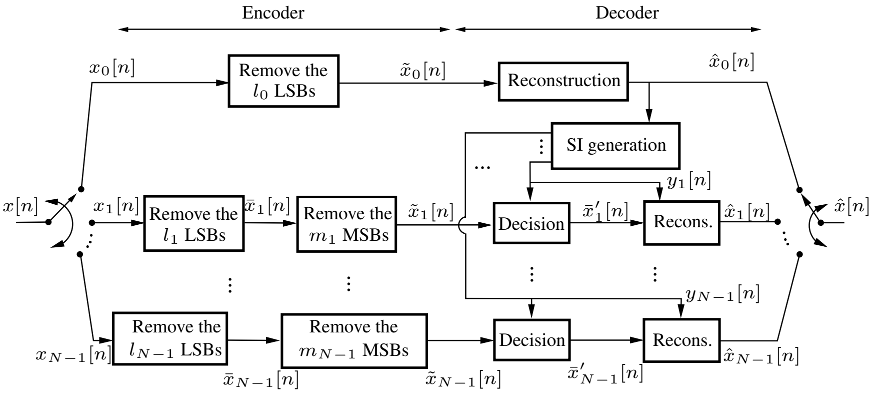
2.2.1 MPCM coding of one-dimensional signals

Let $x(t)$ be an analog signal whose amplitude values lie in $[A_{\min}, A_{\max}]$. Let $x[n]$ ($n \in \{0, 1, 2, ...\}$) be the digital signal that results from the PCM encoding of $x(t)$ by performing sampling and scalar quantization using a fixed-rate uniform quantizer of R_0 bits and step-size $\Delta_0 = (A_{\max} - A_{\min})/2^{R_0}$. In our algorithm, $x[n]$ is divided into N decimated signals $x_k[n]$

$$x_k[n] = x[nN + k], \qquad k \in \{0, \dots, N-1\} \tag{2.1}$$

which are encoded differently and with different accuracies (Figure 2.1). The signal $x_0[n]$, called *PCM signal*, is encoded by removing the l_0 LSBs of each codeword, which is equivalent to a PCM coding using $R_0 - l_0$ bits/sample. The signals $x_k[n]$ ($k \in \{1, \dots, N-1\}$), called *MPCM signals*, are encoded by removing the l_k LSBs and the m_k MSBs of each codeword of $x_k[n]$. The coding of a MPCM signal $x_k[n]$ is equivalent to a PCM coding at rate $R_0 - l_k$ followed by a modulo 2^{m_k} reduction of the resulting codeword [Ericson and Ramamoorthy, 1979]. At the decoder, the decoded PCM signal is used to generate SI for the decoding of the MPCM signals. The N signals $\tilde{x}_k[n]$ that result from this bit-removing process constitute the MPCM bit stream.

Figure 2.1 Block diagram of our MPCM coding algorithm.



The PCM signal should be encoded with a large number of bits in order to obtain a high quality decoded PCM signal and to generate accurate SI. In contrast to the PCM signal, MPCM signals are decoded with the help of SI, and hence, they will generally require a lower rate than the PCM signal.

Each codeword of $\tilde{x}_0[n]$ represents an interval of size $2^{l_0}\Delta_0$. Similarly, each codeword of $\bar{x}_k[n]$ that results from removing the l_k LSBs from $x_k[n]$ represents a quantization interval of size $2^{l_k}\Delta_0$. If $m_k > 0$, the MPCM encoder performs *binning* over the codewords of $\bar{x}_k$ [Cover and Thomas, 1991], i.e., each of the possible $2^{R_0-l_k-m_k}$ transmitted codewords $\tilde{x}_k$ represents a set (or *bin*) of 2^{m_k} codewords $\bar{x}_k$. Equivalently, each $\tilde{x}_k$ represents a set $\mathcal{M}_{k,u}$ of 2^{m_k} disjoint intervals $\mathcal{I}_{k,u,i}(i \in \{0, \ldots, 2^{m_k}-1\})$ of size $\Delta = 2^{l_k}\Delta_0 = (A_{\max} - A_{\min})/2^{R_0-l_k}$

$$\mathcal{M}_{k,u} = \{\mathcal{I}_{k,u,i} | i \in \{0, \ldots, 2^{m_k}-1\}\} \tag{2.2}$$

where u is the base-10 value of $\tilde{x}_k$ and $u \in \{0, 1, \ldots, 2^{R_0-l_k-m_k}\}$. Hence, binning reduces the rate at the expense of introducing decoding ambiguity since, for each received $\tilde{x}_k$, the decoder must decide which of the 2^{m_k} potential codewords $\bar{x}_k$ is the correct one. This is illustrated in Figure 2.2(a) and (b). In Figure 2.2, we left out the index n of $x_k[n]$ to keep the notations simple.

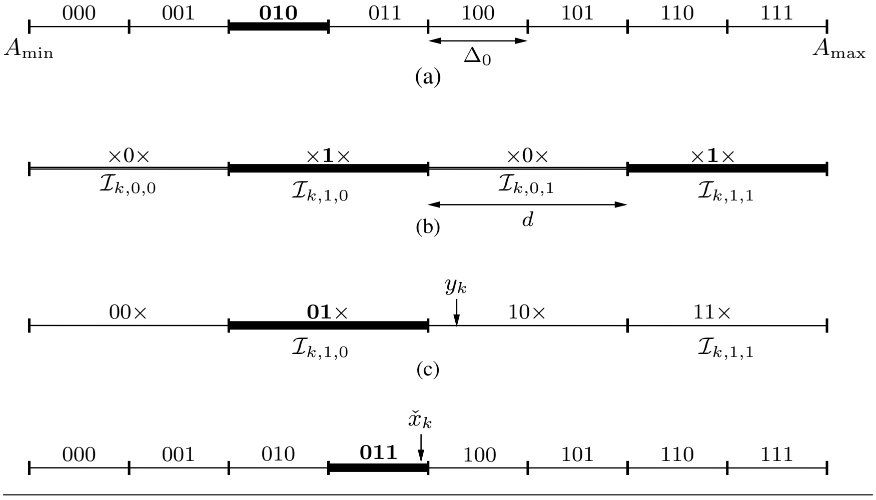
At the decoder, a PCM-coded signal is directly reconstructed from its received codeword $\tilde{x}_0[n]$ (Figure 2.1). The reconstructed value of each codeword is set to the midpoint of its quantization interval. The decoding of the k-th MPCM-coded signal is done by using its codeword $\tilde{x}_k[n]$ and its SI $y_k[n]$ (Figure 2.2). The SI $y_k[n]$ is obtained by interpolating the previously decoded $\hat{x}_0[n]$. The decoding of a MPCM codeword $x_k[n]$ is divided into two steps: *decision* and *reconstruction* (Figure 2.1).

In the decision step, the decoder uses $y_k[n]$ to select one of the 2^{m_k} codewords of the bin represented by $\tilde{x}_k[n]$, which yields the codeword $\bar{x}'_k[n]$ (Figure 2.1). How $\bar{x}'_k[n]$ is selected is explained further in this section (after Eq. 2.4). If the decision is correct, then $\bar{x}'_k[n] = \bar{x}_k[n]$ and the m_k MSBs of $x_k[n]$ are correctly recovered (Figure 2.2(c)). Otherwise, $\bar{x}_k[n]$ and $\bar{x}'_k[n]$ are different codewords, and the decoder incurs a *decision error*. The probability of decision error depends on m_k and the accuracy of the SI (i.e., the similarity between y_k and x_k). The more accurate the SI, the lower the probability of decision error. Additionally, the larger the m_k, the shorter the minimum distance between the codewords of the same bin; hence, the higher the probability of decision error (there is no decision error when $m_k = 0$).

In the reconstruction step, the aim is to recover the LSBs of $x_k[n]$. The decoder first estimates the value of $x_k[n]$ using its SI $y_k[n]$ and the quantization interval that corresponds to $\bar{x}'_k[n]$. How this estimate $\check{x}_k[n]$ is obtained will be explained in detail at the end of this section (Eqs. (2.5), (2.6), (2.7), (2.8)). Then, since the estimated value $\check{x}_k[n]$ is not quantized, it is quantized with the same quantizer used in the PCM coding of $x(t)$ in order to obtain $\hat{x}_k[n]$ (Figure 2.2(d)). If $\check{x}_k[n]$ belongs to the quantization interval of $x_k[n]$, then $\hat{x}_k[n] = x_k[n]$. Otherwise, not all the LSBs removed by the encoder are properly recovered (Figure 2.2(d)), and a *quantization error* is introduced. The quantization error in $x_k[n]$ depends on both l_k (the larger the l_k, the larger the Δ) and the accuracy of the SI. Finally, the N reconstructed signals $\{\hat{x}_k[n]\}$ are multiplexed to generate the decoded signal $\hat{x}[n]$.

To perform optimum decoding and to analyze the distortion introduced in

Figure 2.2 MPCM coding of $x_k = 010$ with $R_0 = 3$, $l_k = 1$ and $m_k = 1$. The symbol × represents the unknown bits after each step. The boldfaced codewords represent the codeword selected after each step. The marked intervals are the intervals represented by the codeword selected after each step. (a) PCM coding at $R = 3$ bits/sample. (b) MPCM encoding. (c) Decision between codewords $01\times$ and $11\times$ using y_k. (d) Reconstruction.



(d)

MPCM coding, we need to assume some statistical models. Since, in this chapter, we propose the use of MPCM for the coding of images, we have used statistical models that are appropriate for this type of signals. We model the amplitude of $x_k[n]$ and $y_k[n]$ as realizations of two stationary random processes. Although $x_k[n]$ and $y_k[n]$ are discrete-amplitude signals, for the sake of simplicity, we model them using two continuous random variables X and Y, respectively. To model the relation between X and Y, we use the additive model $Y = X + Z$ where Z is independent of X, and hence $f_{Y|X}(y|x) = f_Z(y - x)$. The reasonable assumption here is that the value of the prediction residual Z does not depend on the value of X. Since the pixel values of images do not follow any specific distribution, we assume that X is uniformly distributed in $[A_{\min}, A_{\max}]$. Since prediction residuals tend to follow a Laplacian distribution in image and video coding [Netravali and Limb, 1980], we consider that Z is Laplacian [Jayant and Noll, 1984] with probability density function (pdf)

$$f_Z(z) = \frac{\alpha}{2} e^{-\alpha|z|} \tag{2.3}$$

where $\alpha = \sqrt{2}/\sigma_z$ and σ_z is the standard deviation of Z. Finally, we also assume that $\sigma_x^2 \gg \sigma_z^2$, and as $f_Y(y) = f_X(y) * f_Z(y)$ and X is uniformly distributed, then

$f_Y(y) \approx f_X(y)$. Consequently,

$$\begin{aligned} f_{X|Y}(x|y) &\approx f_{Y|X}(y|x) \\ &= \frac{\alpha}{2} e^{-\alpha|x-y|}. \end{aligned} \tag{2.4}$$

In the decoding of a received codeword $\tilde{x}_k[n]$, the decoder must select one among all the codewords $\{\tilde{x}_k[n]\}$ that are in the bin of $\tilde{x}_k[n]$ or, equivalently, one of the intervals in its corresponding set $\mathcal{M}_{k,u}$. The optimum decision is to select the interval in $\mathcal{M}_{k,u}$ that maximizes the probability of $X \in \mathcal{I}_{k,u,i}$ ($u = (\tilde{x}_k[n])_{10}$) given its SI $Y = y_k[n]$. As we assume that $f_{X|Y}(x|y)$ is symmetric and unimodal, the optimum decision is to select the interval closest to $y_k[n]$.

With respect to the reconstruction of $x[n]$, first we estimate the original continuous value of $x_k[n]$ given its selected interval $\mathcal{I}_{k,u,i^*}[n]$ and its SI $y[n]$. The minimum mean squared error (MMSE) estimate of X when $Y = y$ and $X \in [a, b]$ is given by

$$\check{x} = \mathbb{E}\left[X|Y = y; a \leq X < b\right] \tag{2.5}$$

which using (2.4) provides after some calculations:

$$\check{x} = \begin{cases} a + \dfrac{\Delta}{1 - e^{\alpha\Delta}} + \dfrac{1}{\alpha}, & y < a \\ y + \dfrac{e^{-\alpha\gamma}\left(\gamma + \frac{1}{\alpha}\right) - e^{-\alpha\delta}\left(\delta + \frac{1}{\alpha}\right)}{2 - e^{-\alpha\gamma} - e^{-\alpha\delta}} & a \leq y \leq b \\ b - \dfrac{\Delta}{1 - e^{\alpha\Delta}} - \dfrac{1}{\alpha} & y > b \end{cases} \tag{2.6}$$

where $\gamma \triangleq y - a$, $\delta \triangleq b - y$ and $\Delta = b - a$ [Kubasov et al., 2007]. The MMSE estimator (2.6) depends on α and is a nonlinear function. Instead of a MMSE estimation of x, the decoder can perform a Maximum a Posteriori (MAP) estimation:

$$\check{x} = \arg\max_x f(x|y, a \leq X < b) \tag{2.7}$$

where we have removed the subscripts in the pdf for the sake of clarity. Since we assume that $f_{X|Y}(x|y)$ is symmetric and unimodal, (2.7) is simply the clipping function

$$\check{x} = \begin{cases} a, & y < a \\ y, & a \leq y \leq b \\ b, & y > b \end{cases} \tag{2.8}$$

which is simpler than (2.6) and does not depend on α. Note that (2.6) tends to (2.8) when $\alpha \to \infty$. Hence, the MAP and the MMSE reconstruction functions are very close when the quality of the SI is high. Also note that (2.6) transforms into $\check{x} =$

$(a+b)/2$ when $\alpha \to 0$, i.e., the MMSE reconstruction is a midpoint reconstruction when the knowledge of the SI does not help in the reconstruction [Kubasov et al., 2007].

2.2.2 MPCM coding of images

In this section, we extend our algorithm to the coding of images. Although we describe the coding of monochromatic images, our algorithm can be extended to color images by applying the algorithm to each component separately. Let $x[n_1, n_2]$ be a monochromatic digital image of R_0 bits per pixel (bpp). Our algorithm first divides the image into N decimated images, and then it encodes one of the resulting images using PCM and encodes the rest of them using MPCM. A possible strategy is to divide $x[n_1, n_2]$ into four ($N = 4$) images $x_{0,0}[n_1, n_2]$, $x_{0,1}[n_1, n_2]$, $x_{1,0}[n_1, n_2]$, and $x_{1,1}[n_1, n_2]$ such that

$$x_{p,q}[n_1, n_2] = x[2n_1 + p, 2n_2 + q]. \tag{2.9}$$

with p and $q \in [0, 1]$. Note, however, that the algorithm can be easily adapted to other integer values of N. Subsequently, the encoder removes the $l_{p,q}$ LSBs and the $m_{p,q}$ MSBs (with $m_{0,0} = 0$) from each codeword in $x_{p,q}[n, m]$. At the decoder, $x_{0,0}[n_1, n_2]$ is reconstructed from its codewords while each of the rest of the decimated images $x_{p,q}$ is decoded using their corresponding received codewords $\tilde{x}_{p,q}$ and their SI images $y_{p,q}$. The three SI images are generated using an image interpolation algorithm over the reconstructed image $\hat{x}_{0,0}[n_1, n_2]$. There exists a large variety of image interpolation algorithms that could be used in the generation of the SI images [Wolberg, 1990]. We propose the use of *bilinear interpolation* since it represents a good trade-off between simplicity and interpolation accuracy [Wolberg, 1990]. Other under-sampling schemes and interpolation algorithms could be used.

The visibility of a coding error in a pixel depends on both the error amplitude and the local structure of the image in that pixel. Thus, coding errors are more visible in those regions where the luminance changes smoothly than in edges and textured areas. The error introduced by a MPCM coder in a pixel depends on the accuracy of its SI, and, therefore, on the interpolation error. Image interpolation algorithms introduce large errors in edges and textured regions and small errors in smooth regions. Consequently, the spatial distribution of the error introduced by a MPCM coder reduces the visibility of the error. Since the coding error in PCM does not depend on the local structure of the image, the error is more visible in PCM than in MPCM. This will be illustrated by the results presented in Section 2.7.

2.2.3 Relation with other coding techniques

In this section we discuss the relation between our algorithm and other source coding techniques. Our algorithm combines PCM coding [Jayant and Noll, 1984], binning [Comer et al., 1996], and interpolative coding [Zeng and Venetsanopoulos, 1993, Bruckstein et al., 2003]. In fact, for specific values of the coding parameters, our coder behaves as a PCM coder or an interpolative coder.

Our MPCM coder behaves as a PCM coder when binning is not used ($m_k = 0$), and the same number of LSBs are removed in all the samples. In this case, each sample must be decoded independently since all the samples have the same coding distortion (the SI would not provide any help).

In interpolative coding [Zeng and Venetsanopoulos, 1993, Bruckstein et al., 2003], a down-sampled version of the input signal is encoded using a source coding technique. The decoder first decodes the down-sampled signal and then interpolates the result. Note that when $l_k + m_k = R_0$, the MPCM coder behaves as an interpolative coder: the encoder only encodes and transmits the PCM signal $x_0[n]$, which is reconstructed and interpolated at the decoder.

When $m_k = 0$ but not all the values of l_k are equal, our MPCM encoder behaves as a PCM encoder that uses several quantization step-sizes. At the decoder, those samples that have been encoded with less distortion are used to generate SI, which helps in the reconstruction of the rest of the samples. Therefore, in these cases, our algorithm combines PCM with unequal quantization and interpolative coding.

Our algorithm is also related to the MPCM-based coders of Ericson and Ramamoorthy [Ericson and Ramamoorthy, 1979, Ramamoorthy, 1981]. In the MPCM coder proposed in [Ericson and Ramamoorthy, 1979], after sampling the input analog signal, the amplitude of *each* sample is encoded by performing a modulo operation followed by a fixed-rate uniform quantization [Ericson and Ramamoorthy, 1979]. The modulo operation introduces ambiguity since each transmitted codeword represents several quantization intervals. For each received codeword $\tilde{x}[n]$, the decoder first decides which quantization interval corresponds to $\tilde{x}[n]$ taking into account previously decoded codewords. Then, the decoder sets the reconstructed value to the midpoint of the selected interval.

The closed loop architecture of the decoder makes it possible for a decision error to propagate, which significantly reduces the coding performance. To solve this problem, a MPCM coder with SI (MPCMSI) is proposed in [Ramamoorthy, 1981]. In this coder, a SI signal is transmitted together with the MPCM signal in order to prevent decision errors. The SI is generated by quantizing each sample of the input signal with a fixed-rate uniform quantizer and then by encoding the quantization indexes using DPCM and entropy coding. This hybrid MPCM-DPCM strategy has a high coding efficiency but also a high computational complexity. The coding efficiency can be further improved by adapting the coding parameters to the varying properties of the input signal [Ramamoorthy, 1981].

Our algorithm is also a hybrid coding technique. However, in contrast to MPCMSI, our algorithm only involves very simple techniques (PCM, interpolative coding, and binning) that do not require numeric processing or entropy coding at the encoder (only a reduced number of operations are necessary to assign proper values to the coding parameters).

Our algorithm is also related to Pixel-Domain Wyner-Ziv (PDWZ) coding algorithms [Aaron et al., 2002, Aaron et al., 2003, Ascenso et al., 2005b], which will be discussed in detail in Chapter 3. In these coders, the video frames are organized into key frames (K-frames) and Wyner-Ziv frames (WZ-frames). Each key frame is coded using a conventional intra-frame coder. In each Wyner-Ziv frame, the encoder first quantizes the pixel values of the frame. Then, the quantization indexes are encoded using a Slepian-Wolf encoder that transmits a proper number of parity bits [Aaron et al., 2002]. At the decoder, the Slepian-Wolf decoder obtains the quantization indexes from the received parity bits and a SI obtained by extrapolating or interpolating previously decoded frames. Finally, the decoder reconstructs the pixel values using the quantization indexes and the SI.

In PDWZ video coding, the encoders are simple systems but the decoders are complex due to the use of sophisticated channel decoding and SI generation techniques. Additionally, the number of parity bits to transmit in each Wyner-Ziv frame is usually determined by iteratively asking the encoder for parity bits until a correct decoding is achieved [Aaron et al., 2002, Aaron et al., 2003]. This way, the encoder can determine the minimum number of parity bits that are necessary to achieve a correct decoding without analyzing the statistics of the video signal. This rate allocation strategy imposes several constraints on the applications. First, a feedback channel must be used so that the decoder can request bits to the encoder, which inhibits its use in one directional applications. Second, several decodings must be performed for each encoding, and since decoding is a complex operation, the overall number of operations is very high. Third, the use of the feedback channel greatly increases the coding latency. All these factors prevent the use of this coding technique in high-speed video cameras. Note that, in Chapter 3, we will present solutions to the above mentioned problems, but these solutions require extra operations at the encoder and/or do not simplify the decoding process to the extent needed for high-speed video cameras.

Similarly to PDWZ video coding, the algorithm presented in this chapter also tries to recover the quantization indexes of the original signal using SI. In our MPCM algorithm, however, no sophisticated channel code is used and errors in the recovery of the quantization indexes are permitted. Thus, our algorithm achieves both very low latency and complexity (in both encoding *and* decoding) at the expense of a lower coding efficiency than PDWZ coding.

2.3 Distortion model of the MPCM coder

In this section, we derive an approximation for the distortion of our MPCM coder using MAP reconstruction as a function of its coding parameters. First, we obtain an expression for the distortion introduced in each MPCM signal (Section 2.3.1). Then, in Section 2.3.2, we obtain a distortion model for the overall distortion of our MPCM coder with MAP reconstruction using the model of Section 2.3.1 for the MPCM signals and a high-resolution distortion model for the PCM signal [Gray and Neuhoff, 1998].

2.3.1 Distortion model of the MPCM signal

2.3.1.1 Theoretical expression

Let us assume that $x[n]$ is a signal of which the amplitude is uniformly distributed in $[A_{\min}, A_{\max}]$ and that it has been encoded using PCM with R_0 bits/sample and a quantization step $\Delta_0 = (A_{\max} - A_{\min})/2^{R_0}$. Let us also assume that we encode and decode $x[n]$ with the MPCM algorithm of Section 2.2.1. In this section, we will study the distortion introduced in the MPCM signals $x_k[n]$ ($k = 1, \ldots, N-1$) when MAP reconstruction is used. In the remainder of this section, we will leave out the indexes k and n to keep the notations simple.

In the encoding of a codeword x, the l LSBs and the m MSBs of x are removed and the remaining bits $\tilde{x}$ are transmitted to the decoder. Decoding of a received codeword $\tilde{x}$ is performed in two steps. First, the decoder estimates which interval $\mathcal{I}_{u,i} \in \mathcal{M}_u$ (with $i \in \{0, \ldots, 2^m - 1\}$) contains x using $\tilde{x}$ and its SI y. Then, MAP reconstruction is performed using $\mathcal{I}_{u,i}$ and y. In the following analysis, we make use of the statistical assumptions of Section 2.2.1.

As a measure of the coding distortion, we consider the Mean Square Error (MSE)

$$D_{\mathrm{MPCM}} = \mathbb{E}\left[(X - \hat{X}(X,Y))^2\right] \tag{2.10}$$

where $\hat{X}(X,Y)$ is the random variable that represents the reconstructed value and the expectation $\mathbb{E}$ is over X and Y. In this analysis, X and $\hat{X}$ are treated as *continuous* random variables. Therefore, the reconstruction does not include the final quantization process (Section 2.2.1).[3] As X is uniformly distributed in $[A_{\min}, A_{\max}]$, the MSE is

$$D_{\mathrm{MPCM}} = \frac{1}{r} \int_{A_{\min}}^{A_{\max}} \int_{-\infty}^{\infty} (x - \hat{x})^2 f_{Y|X}(y|x) dy\, dx \tag{2.11}$$

[3] In this section, we assume that the amplitude of the signal to encode is continuous, so that the decoder also provides a continuous-amplitude signal. In Section 2.2.1, however, the input signal is digital (a PCM signal), and for this reason, the decoder also provides a digital signal.

where $r = A_{\max} - A_{\min}$. If we divide the integration domain for x into $L = 2^{R_0 - l}$ quantization intervals of size $\Delta = \Delta_0 2^l$, we obtain

$$D_{\text{MPCM}} = \frac{1}{r} \sum_{w=0}^{L-1} \int_{A_{\min}+w\Delta}^{A_{\min}+(w+1)\Delta} \int_{-\infty}^{\infty} (x - \hat{x})^2 f_{Y|X}(y|x) dy\, dx. \tag{2.12}$$

The w-th quantization interval $[A_{\min} + w\Delta, A_{\min} + (w+1)\Delta]$ can be rewritten as $\mathcal{I}_{u,i}$ where

$$\begin{aligned} u &= w \mod M \\ i &= \left\lfloor \frac{w}{M} \right\rfloor \\ w &= Mi + u \end{aligned}$$

with $M = 2^{R_0 - l - m}$. All x values in $\mathcal{I}_{u,i}$ are encoded with the same codeword $\tilde{x}$. This codeword $\tilde{x}$ represents a set $\mathcal{M}_u$ (u is the base-10 value of $\tilde{x}$) of 2^m intervals $\{\mathcal{I}_{u,i}\}$; i indicates the specific interval of $\mathcal{M}_u$ where x lies. Thus, (2.12) can be rewritten as

$$D_{\text{MPCM}} = \frac{1}{r} \sum_{u=0}^{M-1} \sum_{i=0}^{2^m-1} \int_{\mathcal{I}_{u,i}} \int_{-\infty}^{\infty} (x - \hat{x})^2 f_{Y|X}(y|x) dy\, dx. \tag{2.13}$$

After receiving the codeword $\tilde{x}$, the decoder must decide which interval $\mathcal{I}_{u,i}$ in $\mathcal{M}_u$ is the correct one using the SI y of $\tilde{x}$. Each $\mathcal{I}_{u,i}$ has 2^m *decision intervals* $\mathcal{L}_{u,j}$ ($j = 0, \ldots, 2^m - 1$), such that if $y \in \mathcal{L}_{u,j}$, then the decoder decides that $x \in \mathcal{I}_{u,j}$ (Figure (2.3)). If in (2.13), we divide the integration domain in y into its 2^m decision intervals, we obtain

$$D_{\text{MPCM}} = \frac{1}{L} \sum_{u=0}^{M-1} \sum_{i=0}^{2^m-1} \sum_{j=0}^{2^m-1} D_{u,i,j} \tag{2.14}$$

where

$$D_{u,i,j} = \frac{1}{\Delta} \int_{x \in \mathcal{I}_{u,i}} \int_{y \in \mathcal{L}_{u,j}} (x - \hat{x})^2 f_{Y|X}(y|x)\, dy\, dx. \tag{2.15}$$

If $f_{Y|X}(y|x) = \alpha/2 \exp(-\alpha|x - y|)$, then for $i = j$ $D_{u,i,i}$ can be approximated (see Appendix A.1) through

$$D_{u,i,i} \approx \frac{2}{\alpha^2} \left(1 + e^{-\alpha\Delta}\right) + \frac{4}{\alpha^3} \left(e^{-\alpha\Delta} - 1\right) \tag{2.16}$$

while $D_{u,i,j}$ with $j \neq i$ can be approximated (see Appendix A.2) through

$$D_{u,i,j} \approx (j-i)^2 d^2 \sinh \frac{\alpha d}{2} e^{-\alpha|j-i|d} \tag{2.17}$$

where $d = (A_{\max} - A_{\min})/2^m$. Since the approximations in (2.16) and (2.17) do not depend on u, in the following, we will denote $D_{u,i,j}$ by simply $D_{i,j}$. Hence,

$$D_{\text{MPCM}} \approx \frac{M}{L} \sum_{i=0}^{2^m-1} \sum_{j=0}^{2^m-1} D_{i,j} \tag{2.18}$$

Moreover, since all the terms $D_{i,j}$ with the same $|i-j|$ are equal, (2.18) can be rewritten as

$$D_{\text{MPCM}} \approx \frac{M}{L} \sum_{v=0}^{2^m-1} N_v D_v \tag{2.19}$$

where $D_v = D_{i,j}$ when $v = |i-j|$ and N_v is the number of terms $D_{i,j}$ with $v = |j-i|$ in the two summations of (2.18). It is straightforward to show that

$$N_v = \begin{cases} 2^m, & v = 0 \\ 2\left(2^m - v\right), & 1 \leq v \leq 2^m - 1 \end{cases} \tag{2.20}$$

Finally, taking into account (2.16), (2.17), (2.19), and (2.20), we obtain

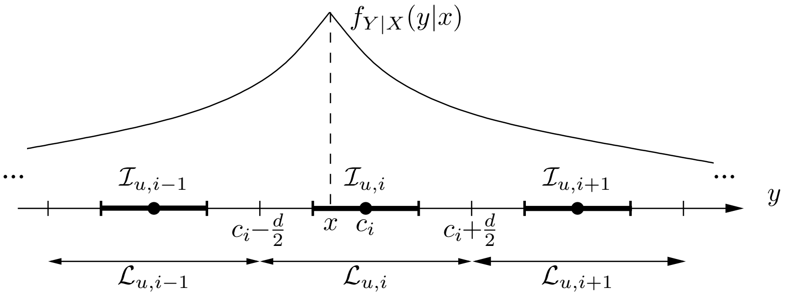
$$\begin{aligned} D_{\text{MPCM}} &\approx \frac{2}{\alpha^2}\left(1 + e^{-\alpha\Delta}\right) + \frac{4}{\alpha^3}\left(e^{-\alpha\Delta} - 1\right) \\ &+ 2\,d^2 \sinh \frac{\alpha d}{2} \sum_{v=1}^{2^m-1} \left(1 - \frac{v}{2^m}\right) e^{-\alpha v d}\, v^2 \end{aligned} \tag{2.21}$$

where the last term of (2.21) equals zero when $m = 0$.

2.3.1.2 Simulations

In this section, we compare the theoretical distortion provided by expression (2.21) (Section 2.3.1) with results obtained from simulations. This way, we test the validity of some of the assumptions we made for the derivation of the theoretical expression. To perform the simulations, we generated a sequence $x[n]$ of 10000 samples drawn from a uniform distribution in [0, 255]. SI for $x[n]$ was generated by adding a sequence $z[n]$ with values drawn from a Laplacian distribution to $x[n]$. To study the influence of the SI accuracy, three different SI sequences were generated, each with a different value of σ_z^2 ($\sigma_x^2/\sigma_z^2 = 1, 100, \text{and } 10000$). The encoding of $x[n]$ was done by uniformly quantizing each sample with $R_0 = 8$ bits

Figure 2.3 Partition of the integration domain in y into the decision intervals $\mathcal{L}_{u,i}$ and their associated quantization intervals $\mathcal{I}_{u,i}$.



and then removing l LSBs and m MSBs from each resulting codeword. For each rate R_{MPCM} of the MPCM signal, the possible pairs (l, m) were those that fulfill

$$R_{\mathrm{MPCM}} = R_0 - l - m, \qquad 0 \le l, m \le R_0. \tag{2.22}$$

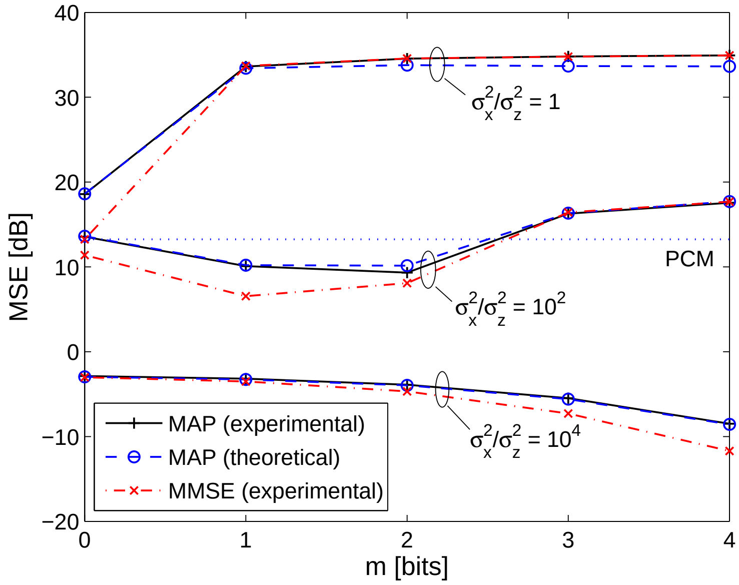
The decoding of each codeword was done by performing decision and reconstruction using the sample of the SI that corresponds to the codeword. Two types of reconstruction were used: MAP and MMSE (with $\alpha = \sqrt{2}/\sigma_z$).

Figures 2.4 and 2.5 show the MSE (in dB) of the decoded MPCM signal as a function of m for $R_{\mathrm{MPCM}} = 4$ bits/sample and $R_{\mathrm{MPCM}} = 2$ bits/sample (each value of m has a corresponding value of $l = R_0 - R_{\mathrm{MPCM}} - m$). Specifically, these figures show the curves obtained: from MPCM encodings of $x[n]$ using both MAP and MMSE reconstruction, from PCM encodings of $x[n]$, and from the theoretical expression (2.21).

Note that in Figures 2.4 and 2.5, the lower σ_z, the higher the value of the optimum m (m^*) since a larger number of MSBs can be removed with a negligible number of decision errors. MMSE reconstruction provides better results than MAP reconstruction and, the closer m is to m^*, the larger the gain of MMSE is with respect to MAP reconstruction. The larger σ_z, the more decision errors will occur, and hence, the larger the contribution of terms $D_{u,i,j}$ with $i \neq j$ in (2.14). Since for these terms the expression in (2.17) is a rough approximation (see Appendix A.2), the difference between the theoretical and the simulated distortion will be larger for larger σ_z.

Figures 2.4 and 2.5 also show that MPCM with MMSE reconstruction and the optimum coding parameters never suffers a loss with respect to PCM. When $\sigma_x^2/\sigma_z^2 = 1$, PCM and MPCM with MMSE reconstruction and $m = 0$ provide almost the same results because both coders essentially behave the same way. MPCM with MAP reconstruction and the optimum m value outperforms PCM when the SI is above a certain threshold of accuracy (below this threshold, mid-

Figure 2.4 Comparison between the theoretical and the simulated performance of a MPCM coder with $R = 4$ bits/sample.



point reconstruction performed by PCM is closer to the optimum (MMSE) than MAP reconstruction). The value of this threshold depends on R_{MPCM}: the lower R_{MPCM}, the lower the necessary quality of the SI so that MPCM with MAP reconstruction outperforms PCM. The reason for this is that the loss of quality due to the requantization performed by PCM (in passing from R_0 to R) must be compensated in part by exploiting the SI in MPCM; therefore, the higher the compression factor, the lower the accuracy of the SI needed to compensate for the loss of quality suffered by PCM.

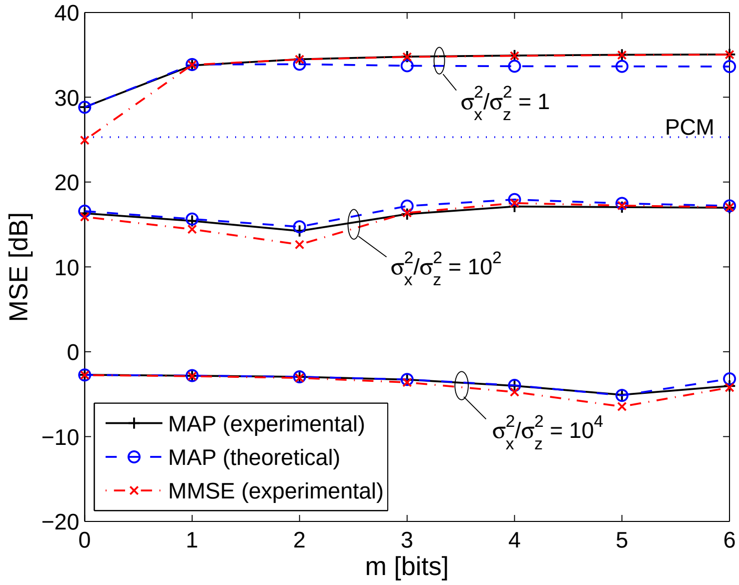
2.3.2 Overall distortion of the MPCM coder

The distortion of an MPCM coder is the average of the distortion of the PCM sequence and the distortions of the $N - 1$ MPCM sequences:

$$D = \frac{1}{N}\left(D_{\text{PCM}} + \sum_{k=1}^{N-1} D_{\text{MPCM},k}\right). \tag{2.23}$$

With respect to D_{PCM}, we assume that $R_0 - l_0$ is large enough to use the

Figure 2.5 Comparison between the theoretical and the simulated performance of a MPCM coder with $R = 2$ bits/sample.



high-resolution expression $D_{\mathrm{PCM}} \approx \Delta^2/12$ [Gray and Neuhoff, 1998], so

$$D_{\mathrm{PCM}} = 2^{2(l_0 - R_0)} \frac{r^2}{12}. \tag{2.24}$$

The distortion $D_{\mathrm{MPCM},k}$ is obtained by substituting $d = 2^{R_0 - m_k}\Delta_0$, $\Delta = 2^{l_k}\Delta_0$, and $\alpha = \alpha_k$ in (2.21). It should be pointed out that, to obtain a simple expression for D, we have assumed that α_k is independent of l_0. Nevertheless, in a practical MPCM coder, $y_k[n]$ is interpolated from $\hat{x}_0[n]$, and therefore, α_k depends on l_0. When the value of l_0 is small, this dependence can be neglected, and hence, α_k is mainly determined by the degree of correlation of the original signal. The dependence of α_k on l_0 increases when l_0 increases, and when l_0 is very close to R_0 (e.g., $l_0 = R_0 - 1$ bits), α_k mainly depends on l_0 and is approximately independent of the correlation degree of $x[n]$.

Finally, in a MPCM coder, the average distortion is

$$\begin{aligned} D &\approx \frac{1}{N}\left[2^{2(l_0-R_0)}\frac{r^2}{12}+\sum_{k=1}^{N-1}[\frac{2}{\alpha_k^2}\left(1-\frac{2}{\alpha_k}+e^{-\alpha_k r 2^{l_k-R_0}}\left(1+\frac{2}{\alpha_k}\right)\right)\right. \\ &+ \left. r^2\,2^{1-2m_k}\sinh\left(\frac{\alpha_k r}{2^{m_k+1}}\right)\sum_{v=1}^{2^{m_k}-1}v^2\left(1-\frac{v}{2^{m_k}}\right)e^{-\alpha_k v r 2^{-m_k}}]\right] \end{aligned} \qquad (2.25)$$

and the average rate R is

$$R = R_0 - \frac{1}{N}\left(l_0+\sum_{k=1}^{N-1}(l_k+m_k)\right) \quad \text{bits/sample.} \qquad (2.26)$$

Expressions (2.25) and (2.26) provide the theoretical distortion/rate pair (D, R) for each set of integer parameters $\{l_k\}$ and $\{m_k\}$. These parameters should fulfill the constraints:

$$\begin{aligned} 0 \le l_0 &\le R_0, & & (2.27)\\ 0 \le l_k &\le R_0, & 1 \le k \le N-1 \quad & (2.28)\\ 0 \le m_k &\le R_0, & 0 \le k \le N-1 \quad & (2.29)\\ 0 \le l_k + m_k &\le R_0, & 1 \le k \le N-1. \quad & (2.30) \end{aligned}$$

For a target rate R, the optimum parameters $\{l_k^*\}$ and $\{m_k^*\}$ are those that minimize the overall distortion (2.25) and fulfill the rate constraints (2.26), (2.27), (2.28), (2.29), and (2.30).

2.4 Optimum coding parameters

In this section, we first use the distortion model of Section 2.3.2 to analyze how the optimum values of the coding parameters vary depending on the rate and the accuracy of the SI (Section 2.4.1). Then, we propose the use of an assignment table so that a MPCM coder can assign proper values to the coding parameters as a function of R and the accuracy of the SI (Section 2.4.2).

2.4.1 Theoretical optimum coding parameters

Table 2.1 shows the theoretical optimum coding parameters and the corresponding decoding quality for a signal that is uniformly distributed in $[0, 255]$ using a MPCM coder at different rates and SI qualities. The MPCM coder uses $N = 4$ and MAP reconstruction (even in those encodings that correspond to PCM encodings).

By varying the quality of the SI used in the decoding phase, we simulate different degrees of correlation that can exist between samples in a signal. To measure the quality of the decoded signal, we use the Peak Signal to Noise Ratio (PSNR) defined as

$$\mathrm{PSNR} = 10\log\frac{255^2}{\mathrm{MSE}} \quad [\mathrm{dB}] \tag{2.31}$$

where MSE is the mean squared *decoding error*. The PSNR is an objective and simple quality measure that is commonly used in image and video coding [Winkler, 2006]. Similarly, to measure the quality of the SI, we will use

$$\mathrm{PSNR}_{\mathrm{SI}} = 10\log\frac{255^2}{\mathrm{MSE}_{\mathrm{SI}}} \quad [\mathrm{dB}] \tag{2.32}$$

where $\mathrm{MSE}_{\mathrm{SI}}$ is the mean squared *interpolation error*. In all our simulations, we assumed that all three SI signals had the same $\mathrm{PSNR}_{\mathrm{SI}}$, and, therefore, we only considered those sets of parameter values where $l_1 = l_2 = l_3$ and $m_1 = m_2 = m_3$. Hence, only three coding parameters had to be chosen in each encoding: l_0, l_1, and m_1.

Note in Table 2.1 that, at $R = 6$ and $R = 4$ bits/sample, the optimum parameters involve the use of binning (i.e., $m_1 > 0$ and $m_1 + l_1 < R_0$) when $\mathrm{PSNR}_{\mathrm{SI}}$ is above a threshold $\mathrm{PSNR}_{\mathrm{SI,B}}(R)$. When $\mathrm{PSNR}_{\mathrm{SI}}$ is below $\mathrm{PSNR}_{\mathrm{SI,B}}(R)$, the decrease in Δ provided by binning does not compensate the large impact that decision errors have on the distortion. The lower the R, the lower the $\mathrm{PSNR}_{\mathrm{SI,B}}(R)$ ($\mathrm{PSNR}_{\mathrm{SI,B}}(6) = 25$ dB and $\mathrm{PSNR}_{\mathrm{SI,B}}(4) = 22.5$ dB). At $R = 2$ bits/sample, binning is included in the optimum assignments when 20 dB < $\mathrm{PSNR}_{\mathrm{SI}} \leq 37.5$ dB. At $R = 1$ bit/sample, the optimum strategy never involves the use of binning regardless of the value of $\mathrm{PSNR}_{\mathrm{SI}}$.

Interpolative coding (i.e., when $l_1 + m_1 = R_0$) is the optimum technique in some of the encodings with $R = 1$ or $R = 2$ bits/sample. At these small rates, there is a small number of bits for encoding both the PCM signal and the MPCM signals. Hence, if $\mathrm{PSNR}_{\mathrm{SI}}$ is above a certain value $\mathrm{PSNR}_{\mathrm{SI,I}}(R)$, it is better to spend all the available bits in the encoding of only the PCM signal and decoding each MPCM signal using only its corresponding SI. The lower the R, the lower the $\mathrm{PSNR}_{\mathrm{SI,I}}(R)$ ($\mathrm{PSNR}_{\mathrm{SI,I}}(2) = 37.5$ dB and $\mathrm{PSNR}_{\mathrm{SI,I}}(1) = 15$ dB). When $R > 2$ bits/sample, interpolative coding is never the optimum coding technique irrespectively of the $\mathrm{PSNR}_{\mathrm{SI}}$ value.

At a given rate R, the optimum MPCM encoder acts as a PCM encoder (i.e., $l_0 = l_1$ and $m_1 = 0$) when $\mathrm{PSNR}_{\mathrm{SI}}$ is equal or below a certain threshold $\mathrm{PSNR}_{\mathrm{SI,P}}(R)$. The lower the R, the lower $\mathrm{PSNR}_{\mathrm{SI,P}}$ ($\mathrm{PSNR}_{\mathrm{SI,P}}(6) = 25$ dB, $\mathrm{PSNR}_{\mathrm{SI,P}}(4) = 22.5$ dB, $\mathrm{PSNR}_{\mathrm{SI,P}}(2) = 20$ dB, and $\mathrm{PSNR}_{\mathrm{SI,P}}(1) = 15$ dB). In these cases, the help provided by the SI does not compensate for the loss suffered by encoding each MPCM signal with fewer bits than the PCM signal. In fact,

Table 2.1 Theoretical optimum coding parameters and maximum PSNR of a MPCM coder ($N = 4$, MAP reconstruction, $l_1 = l_2 = l_3$, and $m_1 = m_2 = m_3$) for several values of $\mathrm{PSNR}_{\mathrm{SI}}$ and R.

$\mathrm{PSNR}_{\mathrm{SI}}$ [dB]	Optimum coding parameters and maximum PSNR							
	$R = 6$ bits/sample		$R = 4$ bits/sample		$R = 2$ bits/sample		$R = 1$ bits/sample	
	(l_0, l_1, m_1)	PSNR	(l_0, l_1, m_1)	PSNR	(l_0, l_1, m_1)	PSNR	(l_0, l_1, m_1)	PSNR
40.0	(2,0,2)	50.97	(1,2,3)	45.93	(0,8,0)	41.29	(4,8,0)	38,09
37.5	(2,0,2)	50.83	(1,2,3)	43.63	(3,4,3)	38.95	(4,8,0)	36.73
35.0	(2,0,2)	49.98	(1,3,2)	40.71	(3,5,2)	36.75	(4,8,0)	35.05
32.5	(2,1,1)	47.63	(1,3,2)	38.98	(3,5,2)	34.56	(4,8,0)	33.11
30.0	(2,1,1)	47.16	(4,3,1)	36.78	(3,5,2)	31.96	(4,8,0)	30.98
27.5	(2,1,1)	43.99	(4,3,1)	36.00	(3,6,1)	29.76	(4,8,0)	28.74
25.0	(2,2,0)	42.56	(4,3,1)	33.74	(3,6,1)	27.40	(4,8,0)	26.45
22.5	(2,2,0)	42.38	(4,4,0)	31.92	(3,6,1)	24.76	(4,8,0)	24.14
20.0	(2,2,0)	42.23	(4,4,0)	31.42	(6,6,0)	23.10	(4,8,0)	21.86

in these cases, midpoint reconstruction performs better than MAP reconstruction. Consequently, at a rate R, a MPCM coder should use conventional PCM when $\mathrm{PSNR}_{\mathrm{SI}} \leq \mathrm{PSNR}_{\mathrm{SI,P}}(R)$.

2.4.2 Assignment table

In order to assign values to the coding parameters, we propose the use of an *assignment table* that provides the values of l_0, l_1, and m_1 as a function of R and $\mathrm{PSNR}_{\mathrm{SI}}$. To build this table, we consider the typical range of $\mathrm{PSNR}_{\mathrm{SI}}$ values that have the type of signals that we want to encode, and we divide this range into intervals of the same width. Then, we assign proper values of l_0, l_1, and m_1 to each entry of the table.

Table 2.2 shows one possible assignment table that is useful for the encoding of images. This table was obtained by first dividing the range of $\mathrm{PSNR}_{\mathrm{SI}}$ values [21 dB, 37 dB] into intervals of 2 dB. This range was chosen because we found that the $\mathrm{PSNR}_{\mathrm{SI}}$ (with $l_0 = 0$) of 25 typical images of 512×512 pixels are in this range. Then, for each rate and midpoint of each $\mathrm{PSNR}_{\mathrm{SI}}$ interval, we found the set of coding parameters that provide the lowest theoretical distortion (according to (2.25)). Finally, PCM is chosen for those rates and $\mathrm{PSNR}_{\mathrm{SI}}$ intervals in which PCM theoretically outperforms MPCM with MAP reconstruction.

Table 2.2 Assignment of values to the coding parameters (l_0, l_1, m_1) of a MPCM coder ($N = 4$, $l_1 = l_2 = l_3$, and $m_1 = m_2 = m_3$) as a function of $\mathrm{PSNR}_{\mathrm{SI}}$ and R.

$\mathrm{PSNR}_{\mathrm{SI}}$ [dB]	Rate [bits/sample]						
	7	6	5	4	3	2	1
$(35, 37]$	(1,0,1)	(2,0,2)	(3,1,2)	(1,3,2)	(2,3,3)	(3,5,2)	(4,8,0)
$(33, 35]$	(1,0,1)	(2,0,2)	(3,1,2)	(1,3,2)	(2,4,2)	(3,5,2)	(4,8,0)
$(31, 33]$	(1,0,1)	(2,1,1)	(3,2,1)	(1,3,2)	(2,4,2)	(3,5,2)	(4,8,0)
$(29, 31]$	(1,1,0)	(2,2,0)	(3,2,1)	(4,3,1)	(2,4,2)	(3,6,1)	(4,8,0)
$(27, 29]$	(1,1,0)	(2,2,0)	(3,3,0)	(4,3,1)	(2,5,1)	(3,6,1)	(4,8,0)
$(25, 27]$	(1,1,0)	(2,2,0)	(3,3,0)	(4,4,0)	(5,4,1)	(3,6,1)	(4,8,0)
$(23, 25]$	(1,1,0)	(2,2,0)	(3,3,0)	(4,4,0)	(5,5,0)	(3,6,1)	(4,8,0)
$(21, 23]$	(1,1,0)	(2,2,0)	(3,3,0)	(4,4,0)	(5,5,0)	(3,7,0)	(4,8,0)

To obtain an assignment from Table 2.2, the encoder must know the $\mathrm{PSNR}_{\mathrm{SI}}$, and, hence, it must compute the $\mathrm{MSE}_{\mathrm{SI}}$. However, this cannot be done since the $\mathrm{MSE}_{\mathrm{SI}}$ depends on the value of l_0, which is still unknown. To solve this problem, we propose assuming that $l_0 = 0$ in the computation of $\mathrm{MSE}_{\mathrm{SI}}$ (i.e., to directly interpolate the original PCM signal).

The increase in complexity due to the computation of the $\mathrm{MSE}_{\mathrm{SI}}$ can make our algorithm useless for the high-speed video applications considered in this chapter. To alleviate this problem, we propose estimating the $\mathrm{MSE}_{\mathrm{SI}}$ by considering only a fraction of the samples of the MPCM signals.[4] Since the error in estimating the $\mathrm{MSE}_{\mathrm{SI}}$ depends on the number of samples considered, there is a trade-off between the accuracy of the estimated $\mathrm{MSE}_{\mathrm{SI}}$ and the encoder complexity. This trade-off and the quality of the estimation of $\mathrm{MSE}_{\mathrm{SI}}$ as a function of the fraction of samples used will be studied in detail in Section 2.6.2.

From the estimated $\mathrm{MSE}_{\mathrm{SI}}$, we obtain the $\mathrm{PSNR}_{\mathrm{SI}}$ through (2.32). To simplify this step (which involves the computation of a logarithm), we propose the use of a table that provides the $\mathrm{PSNR}_{\mathrm{SI}}$ for a set of $\mathrm{MSE}_{\mathrm{SI}}$ values. This table must also be present at the decoder. This way, if the encoder transmits the table index, the decoder can obtain an approximate value for the $\mathrm{PSNR}_{\mathrm{SI}}$, and from this, a value for α which is necessary when MMSE reconstruction is performed.

The estimated $\mathrm{PSNR}_{\mathrm{SI}}$ should be decreased before using it in the assignment table. There are two reasons for this correction. First, since the $\mathrm{PSNR}_{\mathrm{SI}}$ is computed from a fraction of the interpolation error samples, the estimated $\mathrm{PSNR}_{\mathrm{SI}}$ will fluctuate around its true value. However, an overestimation of the $\mathrm{PSNR}_{\mathrm{SI}}$ is generally worse than an underestimation. The reason is that, due to the large impact of decision errors, the decrease in coding efficiency is larger when binning is performed in poorly correlated signals than when binning is *not* performed in highly correlated signals. By reducing the estimated $\mathrm{PSNR}_{\mathrm{SI}}$, the probability of using binning in poorly correlated signals is reduced. Second, even if all the samples are used in the computation of $\mathrm{PSNR}_{\mathrm{SI}}$, the value obtained at the encoder will be higher than the real one since the decoder generates the SI from the decoded PCM signal. How large this decrease in PSNR should be will be explained in Section 2.6.2.

2.5 Computational Complexity

In this section, we analyze the computational complexity of our MPCM coder in terms of the number of arithmetic operations, the memory consumption, and the access to the data. Since, in this chapter, the main application of our algorithm is the coding of digital images, we focus on the algorithm described in Section 2.2.2. We analyze the encoder and the decoder separately since they have different complexities.

[4] Since the $\mathrm{MSE}_{\mathrm{SI}}$ is an estimate of the variance of the interpolation error using the method of moments, the $\mathrm{MSE}_{\mathrm{SI}}$ estimated using a fraction of the samples is a random variable of which the mean is the true variance and of which the variance depends on the number of samples considered in the computation [Kay, 1993].

2.5.1 Encoder

The MPCM encoding of an image of w pixels (width) × h pixels (height) has two stages: the assignment of values to the coding parameters and the encoding process. With respect to the assignment, it basically involves the computation of $\mathrm{MSE}_{\mathrm{SI}}$. As proposed in Section 2.4.2, to reduce the number of operations necessary to compute $\mathrm{MSE}_{\mathrm{SI}}$, we evaluate the interpolation error in one out of s pixels in each dimension of each of the three MPCM subimages. This way, the number of operations necessary to compute $\mathrm{MSE}_{\mathrm{SI}}$ is reduced by a factor s^2. Since we use bilinear interpolation to generate the SIs, the interpolation of each pixel of $x_{0,1}$ and $x_{1,0}$ requires the average of two pixels of $x_{0,0}$ (i.e., to perform one addition and one division by 2). And the interpolation of each pixel of $x_{1,1}$ requires the average of four pixels of $x_{0,0}$ (i.e., three additions and one division by 4). Hence, the bilinear interpolation requires $5wh/(4s^2)$ additions and $3wh/(4s^2)$ divisions. The computation of the interpolation error samples, the squaring of each error sample, and the summation of all the error samples require $3wh/(4s^2)$ additions, $3wh/(4s^2)$ subtractions, and $3wh/(4s^2)$ multiplications, respectively. All these arithmetic operations are integer operations.

With respect to the encoding process, no arithmetic operation is necessary since it only involves removing a different number of bits from the value of each pixel depending on its position in the image (if the row/column of the pixel is even/odd). Table 2.3 shows the number of operations per pixel that is required to encode an image. Note that even with small values of s, the average total number of operations per pixel is very small (less than 1 operation per pixel if $s > 2$).

Table 2.3 Number of operations per image pixel required in the MPCM encoding and decoding (using MAP reconstruction) of an image.

Operation	Average number of operations per pixel	
	Encoder	Decoder
Additions	$\frac{2}{s^2}$	$\frac{3m_1+6}{4}$
Subtractions	$\frac{3}{4s^2}$	$\frac{3m_1+6}{4}$
Multiplications	$\frac{3}{4s^2}$	0
Divisions	$\frac{3}{4s^2}$	$\frac{3}{4}$
Total	$\frac{17}{4s^2}$	$\frac{15+6m_1}{4}$

The two-stage process (assignment/encoding) must be done sequentially (the encoding cannot be started before the encoding parameters are known). With respect to the memory consumption, wh memory positions are necessary to hold the digital image while the $\mathrm{MSE}_{\mathrm{SI}}$ is computed. If the application requires to perform the entire encoding in a single stage, the encoder can compute the coding parameters of a frame and, at the same time, encode the frame using the coding parameters of the previous frame. For the encoding of the first frame, we can use a default set of values for the coding parameters. By doing that, the wh memory positions are no longer necessary and the coding of a pixel can be done as soon as the A/D converter of the camera provides its PCM codeword.

2.5.2 Decoder

Apart from decoding the values of l_0, l_1, and m_1 (and decoding $\mathrm{PSNR}_{\mathrm{SI}}$ and obtaining α if MMSE reconstruction is used), the decoder has three stages: the decoding of the PCM image, the generation of the SIs, and the decoding of the three MPCM subimages. Since bilinear interpolation involves very local processing, the three stages can be done almost simultaneously and the memory usage is very small. Specifically, the maximum decoding latency is $w + 1$ pixels and a maximum number of $w + 2$ positions of memory are required.

The decoding of the PCM subimage involves adding $2^{l_0 - 1}\Delta_0$ to each codeword (i.e., $wh/4$ additions). With respect to the generation of the SIs, the bilinear interpolation of the MPCM subimages requires $5wh/4$ additions and $3wh/4$ divisions. The decoding of the MPCM subimages involves the decision and the reconstruction processes. In the decision of each pixel of the MPCM subimages, the decoder has to select one of the 2^{m_1} decision intervals (see Figure 2.3). This can be done by performing m_1 comparisons between the SI and the limits that separate the decision intervals. To compute each limit, one addition or subtraction, depending on if the SI was on the right or on the left of the previous limit, is necessary. Finally, the MAP reconstruction of each pixel of the MPCM images involves two comparisons.

Table 2.3 shows the average number of operations per image pixel that is necessary to decode an image when $l_0 \neq l_1$. In this table, we have assumed that a comparison is equivalent to a subtraction, and that in the computation of the limits between the decision intervals, all the operations are additions. When $l_0 = l_1$ and $m_1 = 0$, then the decoder only needs to perform one addition per pixel (midpoint reconstruction). Note that the decoder performs more operations than the encoder when $s > 1$.

When MMSE reconstruction is used, the number of operations increases considerably mainly because of the computation of the terms $e^{-\alpha\gamma}$ and $e^{-\alpha\delta}$ in (2.6). Hence, MMSE reconstruction should be avoided when very low decoding complexity is also required.

2.6 Experimental results

In this section, we have experimentally assessed the efficiency of our algorithm for coding digital images. In our experiments, we used ten gray-scale images (512×512 pixels and 8 bit planes) with different degrees of spatial correlation. Our image coder is the MPCM algorithm described in Section 2.2.2. As in Section 2.4, we restricted the parameter assignments to those that fulfill $l_1 = l_2 = l_3$ and $m_1 = m_2 = m_3$. The experimental results allow us to assess the accuracy of the distortion model of Section 2.3 (Section 2.6.1), to show the coding efficiency of our algorithm as a function of its computational complexity (Section 2.6.2), and to compare our algorithm with other coding strategies (Section 2.6.3).

2.6.1 Experimental optimum parameters

We encoded each of the ten images using MPCM with all the possible parameter assignments that provide an average rate of 1, 2, 4, and 6 bpp. In all these encodings, MAP reconstruction was used (even in those assignments that correspond to PCM encodings). The optimum parameter values and the maximum PSNR of the MPCM coder for each image and rate are shown in Table 2.4. Table 2.4 also shows the PSNR of the SI when $l_0 = 0$ for each image.

Tables 2.1 and 2.4 allow us to compare the theoretical and real performance of a MPCM image coder. The differences between the values in the two tables are due to the hypotheses that we have assumed in the theoretical analysis but that are not fulfilled in practice. In particular, in practice: the value of $\mathrm{PSNR}_{\mathrm{SI}}$ depends on l_0 and is generally different for each MPCM signal, the interpolation error is only *approximately* Laplacian, and the luminance values are not uniformly distributed.

Despite the differences between the theoretical and the experimental results, the theoretical optimum parameter values are similar to the experimental ones, and the evolution of the optimum parameter values with $\mathrm{PSNR}_{\mathrm{SI}}$ and R follow the same *tendencies* in both the tables. However, for each rate, the interval of $\mathrm{PSNR}_{\mathrm{SI}}$ values in which the optimum coding uses binning is smaller in Table 2.4 than in Table 2.1. This indicates that our distortion model underestimates the impact that the decision errors have on the quality of the decoded images. This fact, together with the two reasons that we have already mentioned in Section 2.4.2, justify that the estimated $\mathrm{PSNR}_{\mathrm{SI}}$ is decreased before performing the assignment.

As we have already noted in Section 2.4.1, PCM outperforms MPCM with MAP reconstruction at each rate when $\mathrm{PSNR}_{\mathrm{SI}}$ is below a certain value. For instance, at $R = 4$ bpp, PCM provides a higher PSNR than MPCM with MAP reconstruction for the six images with the smallest $\mathrm{PSNR}_{\mathrm{SI}}$. As we proposed in

Table 2.4 Optimum coding parameters and maximum PSNR (in dB) of a MPCM coder (MAP reconstruction, $l_1 = l_2 = l_3$, and $m_1 = m_2 = m_3$) for twelve gray-scale images and four rates. For each image, the PSNR of the SI when $l_0 = 0$ ($\mathrm{PSNR}_{\mathrm{SI}}$) is also shown.

Image	$\mathrm{PSNR}_{\mathrm{SI}}$ [dB]	Optimum parameters and maximum PSNR			
		$R = 6$ bpp		$R = 4$ bpp	
		(l_0, l_1, m_1)	PSNR	(l_0, l_1, m_1)	PSNR
Zelda	35.40	(2,1,1)	48.08	(1,3,2)	40.01
Lena	32.22	(2,1,1)	47.56	(1,4,1)	37.70
Clown	30.58	(2,2,0)	43.54	(1,4,1)	35.02
Peppers	30.47	(2,2,0)	44.62	(4,4,0)	34.90
Couple	28.45	(2,2,0)	44.40	(4,4,0)	33.93
Man	27.09	(2,2,0)	44.29	(4,4,0)	33.38
Aerial	25.26	(2,2,0)	44.64	(4,4,0)	33.48
Bridge	24.54	(2,2,0)	44.32	(4,4,0)	32.63
Barbara	23.77	(2,2,0)	44.65	(4,4,0)	33.67
Baboon	21.43	(2,2,0)	43.84	(4,4,0)	32.20

Image	$\mathrm{PSNR}_{\mathrm{SI}}$ [dB]	Optimum parameters and maximum PSNR			
		$R = 2$ bpp		$R = 1$ bpp	
		(l_0, l_1, m_1)	PSNR	(l_0, l_1, m_1)	PSNR
Zelda	35.40	(0,8,0)	36.61	(4,8,0)	33.84
Lena	32.22	(3,6,1)	33.74	(4,8,0)	31.79
Clown	30.58	(3,6,1)	32.16	(4,8,0)	30.62
Peppers	30.47	(3,7,0)	32.16	(4,8,0)	30.59
Couple	28.45	(3,7,0)	29.63	(4,8,0)	28.45
Man	27.09	(3,7,0)	28.82	(4,8,0)	27.70
Aerial	25.26	(3,7,0)	27,74	(4,8,0)	26.46
Bridge	24.54	(3,7,0)	26.51	(4,8,0)	25.48
Barbara	23.77	(3,7,0)	26.63	(4,8,0)	24.74
Baboon	21.43	(3,7,0)	24.22	(4,8,0)	22.52

Section 2.4.2, this justifies that our coder uses midpoint reconstruction when it acts as a PCM coder, and MAP or MMSE reconstruction in the rest of the cases.

We also obtained the optimum coding parameters for the images and the rates of Table 2.4 using a MPCM coder with MMSE reconstruction. The α value necessary to perform MMSE reconstruction was obtained by using $\mathrm{MSE_{SI}}$ as the estimate of the interpolation error of each image ($\alpha = \sqrt{2}/\sqrt{\mathrm{MSE_{SI}}}$). The obtained optimum assignments were the same as in Table 2.4 except for one of the 40 encodings. Hence, since the type of reconstruction (MAP or MMSE) does not change the optimum assignment significantly, the assignment table of Section 2.4.2 could also be used for MPCM coding with MMSE reconstruction.

2.6.2 Coding efficiency and complexity

Figures 2.6, 2.7 and 2.8 show the PSNR of six images (*Zelda*, *Lena*, *Peppers*, *Man*, *Barbara*, *Baboon*) as a function of the rate R using four different coding strategies: MPCM with MMSE reconstruction and an optimal assignment (MPCM-MMSE-OA); MPCM with MAP reconstruction and an optimal assignment (MPCM-MAP-OA); MPCM with MAP reconstruction and the assignment provided by Table 2.2 (MPCM-MAP); and PCM coding. In the MPCM encodings, midpoint reconstruction is used when $m_1 = 0$ and $l_0 = l_1$. For each rate, the optimal assignments were found by encoding the images at all the possible assignments and choosing the optimal one. In those encodings that needed to estimate $\mathrm{PSNR_{SI}}$, all the error pixels were considered in the computation of the $\mathrm{MSE_{SI}}$ (i.e., $s = 1$). As proposed in Sections 2.4.2 and 2.6.1, we decreased the estimated $\mathrm{PSNR_{SI}}$ before obtaining the assignment from Table 2.2. Specifically, we subtracted 2.5 dB from the estimated $\mathrm{PSNR_{SI}}$ since we experimentally tested that this generally provides good assignments when Table 2.2 is used.

As expected, MPCM-MMSE-OA provides the best results of all the strategies considered. The performance gain of MPCM-MMSE-OA with respect to PCM decreases when R increases and when the degree of spatial correlation decreases. MPCM-MMSE-OA performs slightly better than MPCM-MAP-OA. Consequently, the use of MMSE reconstruction is not justified except for applications where decoding complexity is not an issue.

The loss in performance incurred by the use of Table 2.2 (instead of the optimal assignment) is not high in most cases. One of the highest losses is produced in *Peppers* at $R = 4$ bpp (a loss of 1.49 dB). While the ideal assignment for this encoding is (4,4,0), the assignment obtained from Table 2.2 after the $\mathrm{PSNR_{SI}}$ correction is (4,3,1). Similar incorrect assignments occur in other images especially at 3 and 4 bpp. Nevertheless, in most cases MPCM-MAP performs better than or the same as PCM, with large gains at 1 and 2 bpp. This shows that the use of an assignment table allows a practical MPCM coding of images.

Figure 2.6 PSNR as a function of the rate for the images *Zelda* and *Lena* coded using MPCM-MMSE with optimal assignment, MPCM-MAP with optimal assignment, MPCM-MAP with assignment from Table 2.2, and PCM.

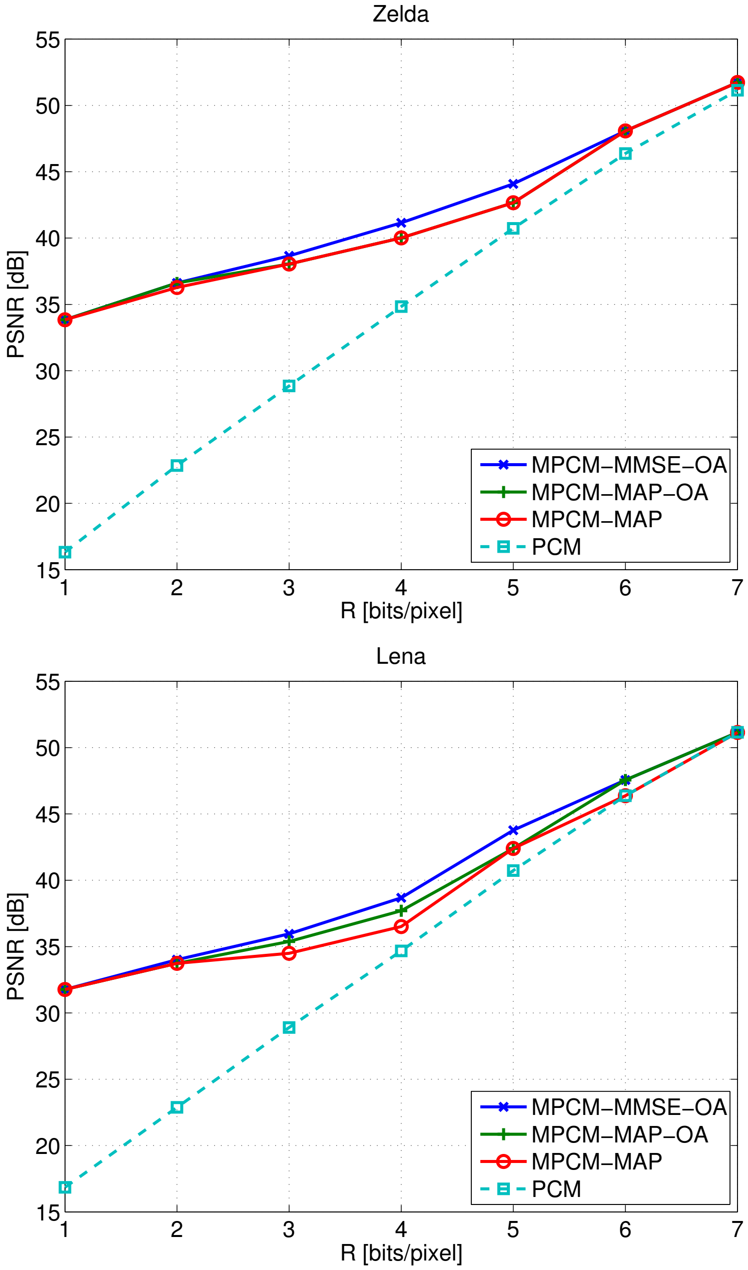


Figure 2.7 PSNR as a function of the rate for the images *Peppers* and *Man* coded using MPCM-MMSE with optimal assignment, MPCM-MAP with optimal assignment, MPCM-MAP with assignment from Table 2.2, and PCM.

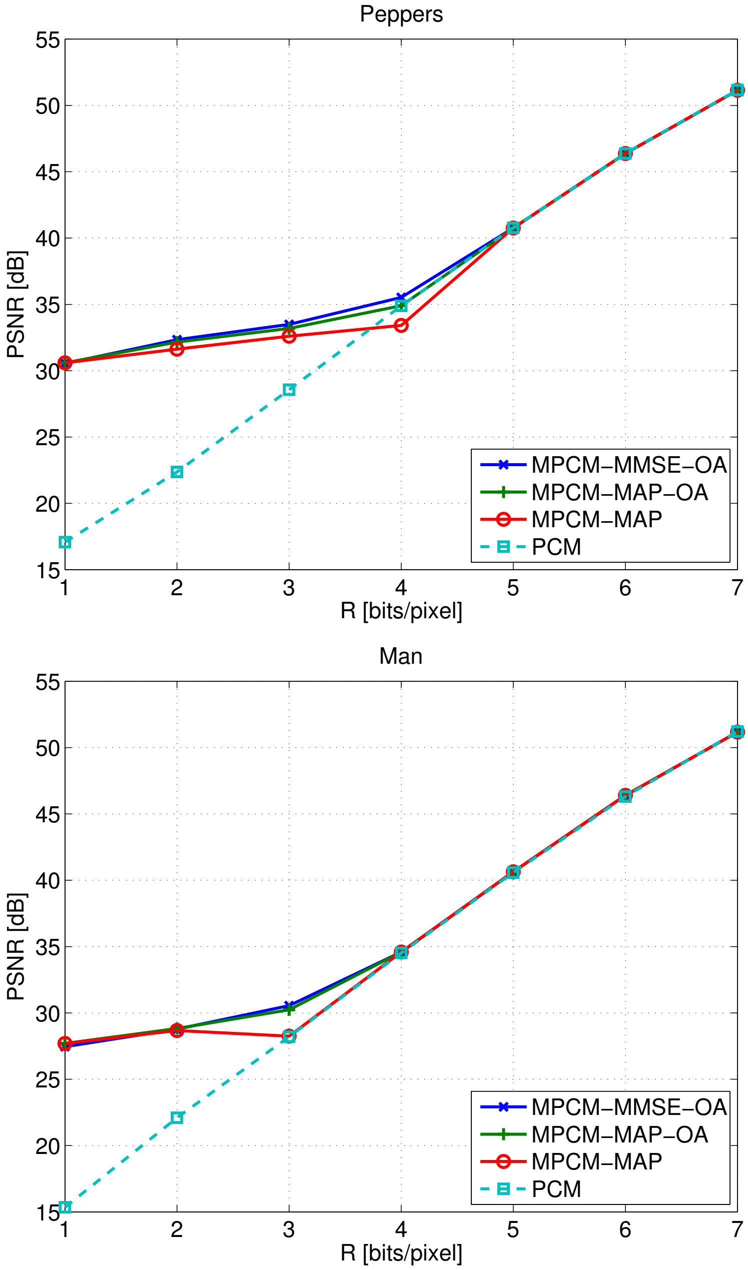
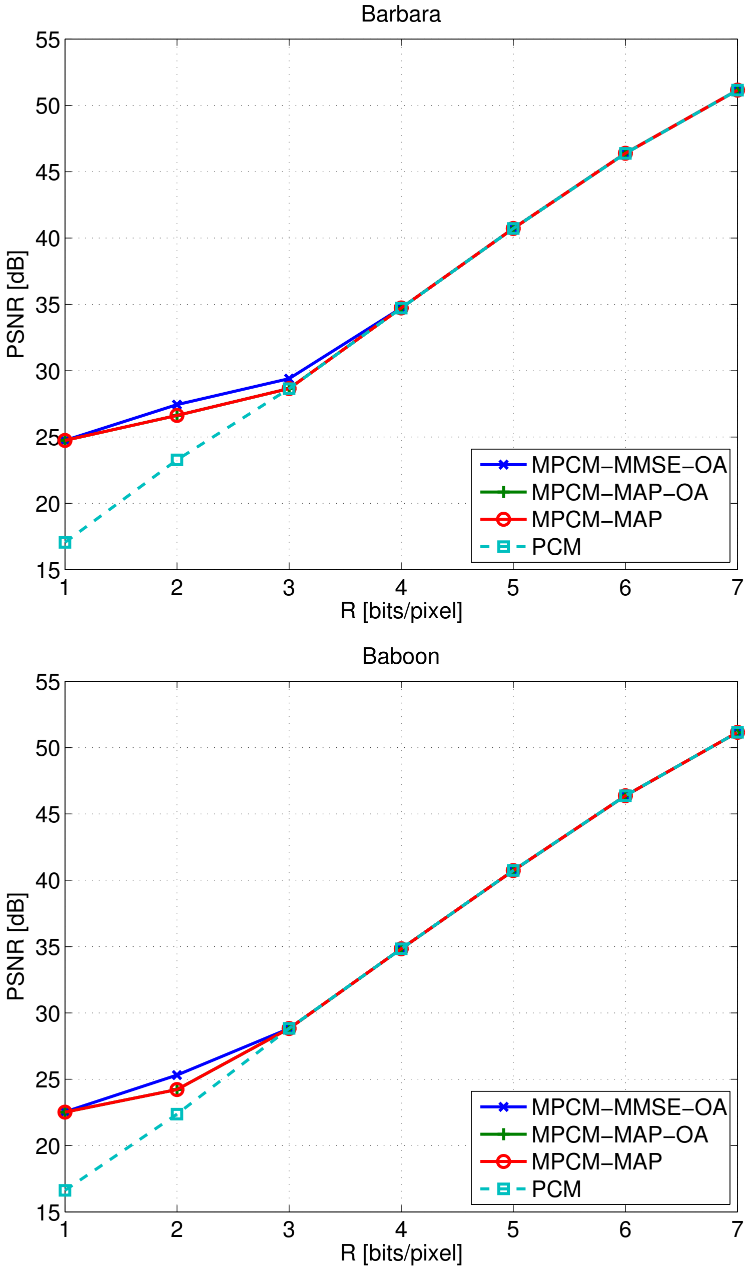


Figure 2.8 PSNR as a function of the rate for the images *Barbara* and *Baboon* coded using MPCM-MMSE with optimal assignment, MPCM-MAP with optimal assignment, MPCM-MAP with assignment from Table 2.2, and PCM.



To use the assignment table in a MPCM coder, the $\mathrm{MSE}_{\mathrm{SI}}$ must be computed. If we use all samples for this, the calculation of $\mathrm{MSE}_{\mathrm{SI}}$ represents performing 4.25 arithmetic operations per pixels at the encoder. Those video applications that cannot afford this encoding complexity can compute the $\mathrm{MSE}_{\mathrm{SI}}$ using a fraction of the pixels (Section 2.4.2). However, this complexity reduction is achieved at the expense of introducing errors in the $\mathrm{MSE}_{\mathrm{SI}}$ computation. To see how these errors influence the coding efficiency, Figure 2.9 shows the average loss (in PSNR) of the ten images with respect to an ideal assignment for each rate and s value. Note that, except at 3 bpp, the losses are not large even when $s = 32$. At 3 bpp, however, the loss is significant even with $s = 1$, which indicates that Table 2.2 provides bad assignments at this rate. Additionally, at 3 bpp, the loss is much more sensitive to errors in the estimation of $\mathrm{MSE}_{\mathrm{SI}}$ than at the other rates. Note that at 4 bpp, the value $s = 1$ provides the highest PSNR loss.

The above results suggest that the coding efficiency could be improved by using an assignment table different from Table 2.2, which was derived using our distortion model. Table 2.4 shows that, in the encoding of the ten digital images, for each R, the optimum assignments mainly depend on the $\mathrm{PSNR}_{\mathrm{SI}}$. Consequently, by encoding a larger set of images and studying how the PSNR varies as a function of R and $\mathrm{PSNR}_{\mathrm{SI}}$, a better assignment table could be obtained (especially for 3 and 4 bpp).

To show the trade-off between coding efficiency and computational complexity, we averaged the PSNR losses shown in Figure 2.9 for each value of s. The results are shown in Table 2.5, which also shows the number of operations per pixel that the *encoder* must perform for each value of s. With $s = 16$, the average decrease in PSNR with respect to $s = 1$ is 0.05 dB; however, 256 times fewer operations are required. Hence, we suffer a negligible average loss in efficiency when the encoder performs only 0.017 operations per pixel. Even with simpler encoders, the average losses are not significant: 0.1 dB with 0.004 operations per pixel, and 0.18 dB with 0.001 operations per pixel.

Table 2.5 Average loss in PSNR and average number of operations per image pixel required in the MPCM-MAP encoding of an image.

s	Average loss [dB]	Average # ops. per pixel
1	0.159	4.25
2	0.152	1.062
4	0.177	0.266
8	0.188	0.066
16	0.209	0.017
32	0.258	0.004
64	0.340	0.001

Figure 2.9 Average loss in PSNR with respect to an ideal assignment for different values of s.

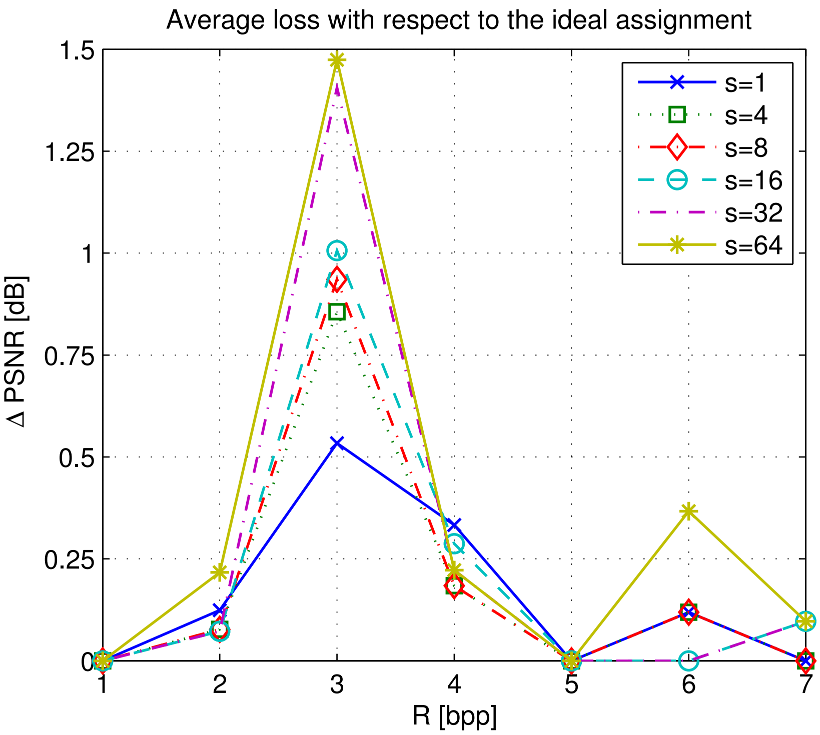
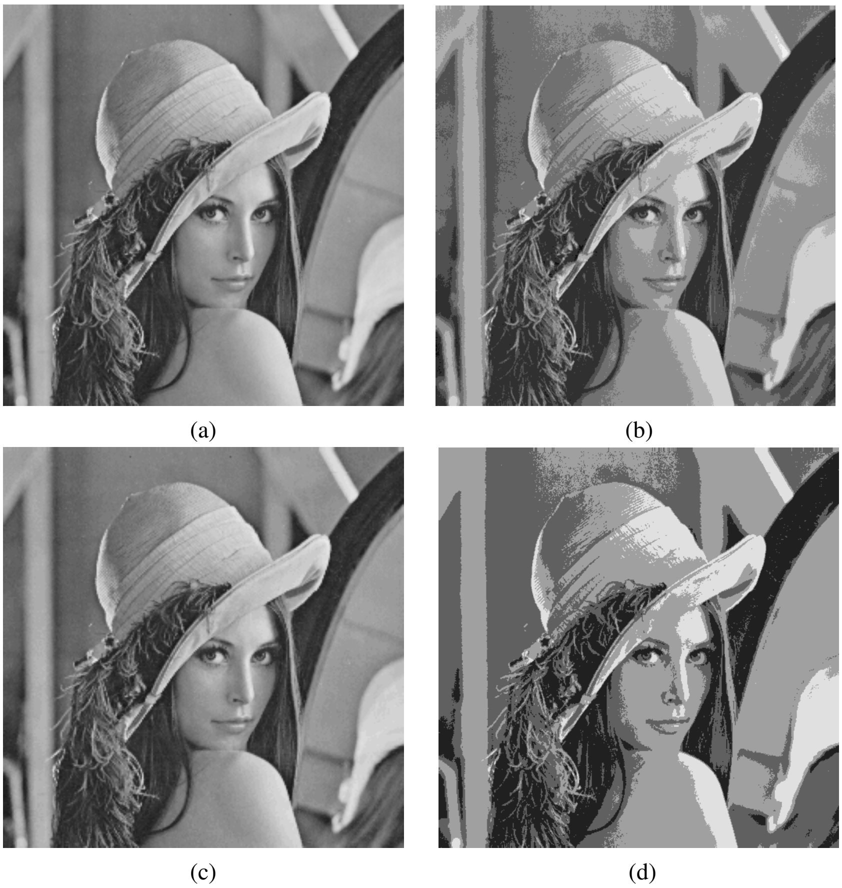


Figure 2.10 shows the image *Lena* encoded at 2 bpp and 3 bpp using: MPCM-MAP with $s = 32$, and PCM. In the images coded using MPCM, the degradation comes mostly from the under-sampling and interpolation performed (the reduction of the edge sharpness and stair-case effect in slanted edges). In the images encoded with PCM, luminance is poorly represented and there is also *false contouring* [Jain, 1989]. Note that the difference in quality, both objective and subjective, of MPCM with respect to PCM increases as the rate decreases (the difference is even greater at 1 bpp).

Figure 2.10 Coding of the image *Lena* using: (a) MPCM-MAP ($s = 32$) at 3 bpp, (b) PCM at 3 bpp, (c) MPCM-MAP ($s = 32$) at 2 bpp, and (d) PCM at 2 bpp.



(a)

(b)

(c)

(d)

2.6.3 Comparison with other techniques

In this section, we compare our coder with two coding techniques: MPCMSI and PDWZ. A brief description of these two techniques can be found in Section 2.2.3. We have encoded two typical images (*Lena* and *Barbara*) using our MPCM-MAP algorithm (with $s = 32$), MPCMSI and PDWZ. The results are shown in Figure 2.11, where the performance of a PCM coder is also plotted for comparison.

We adapted the MPCMSI speech coder described in [Ramamoorthy, 1981] for the encoding of digital images. In our MPCMSI image coder, the generation of the MPCM and the SI signals is done by removing bits from the pixel values of the input image. The DPCM coding of the SI uses *planar prediction* [Jayant and Noll, 1984] (more sophisticated prediction can be used at the expense of a higher coding complexity). The prediction error is encoded using a Huffman code that is optimal for the statistics of the prediction error of each encoding. Each image was encoded using all the possible combinations of the coding parameters. The concave hull of all the (R, PSNR) points obtained in the encoding of each image is shown in Figure 2.11. This curve represents the performance of the MPCMSI coder when an optimum assignment and Huffman coding is done.

Figure 2.11 shows that MPCMSI outperforms our algorithm except for low rates. However, our encoding algorithm is much less complex than the MPCMSI encoder. Our MPCMSI implementation encodes each pixel with six integer operations (two additions, three subtractions, and one multiplication). Our encoder requires $4.25/s^2$ integer operations per pixel to estimate the image statistics and zero operations for the encoding. If we assume that both MPCMSI and our algorithm estimate the image statistics with the same number of operations, then our algorithm performs approximately $1.4s^2$ times fewer operations than MPCMSI when $s \geq 4$ (e.g., 1446 times fewer operations when $s = 32$). In MPCMSI, apart from the DPCM decoding, we have to perform the final PCM reconstruction. Hence, the decoding complexity of MPCMSI is 7 operations per pixel. In our MPCM-MAP decoder, $(15 + 6m_1)/4$ operations per pixel are necessary when $l_0 \neq l_1$ and 1 operation per pixel when $l_0 = l_1$. Since $m_1 \leq 3$ in all the possible assignments of Table 2.2, our MPCM-MAP decoder performs a number of operations that is similar or less than the number of operations performed by a MPCMSI decoder.

The high coding efficiency of MPCMSI is mainly due to the Huffman coding of the prediction error. However, the use of Huffman coding (or any other entropy coding technique) destroys the random access and scalability properties of the original PCM bit stream. Both properties are still present in the bit stream generated by our algorithm.

We implemented a PDWZ image coder that resembles the architecture of most PDWZ video coders [Aaron et al., 2002, Aaron et al., 2003, Ascenso et al., 2005b]. Our PDWZ coder divides the image into two parts: the key subimage (K-subimage), which is equal to $x_{0,0}[n_1, n_2]$, and the Wyner-Ziv subimage

(WZ-subimage) which comprises the rest of pixels. The K-subimage is encoded using PCM at a rate R_{PCM}. To encode the WZ-subimage, its M_{WZ} most significant bit planes are extracted and encoded independently. The encoding, transmission, and decoding of the bit planes is done in order of significance (the most significant bit planes are transmitted and decoded first). Each bit plane is encoded using a Slepian-Wolf coder based on a turbo code [Lin and Costello, 2004]. The turbo encoder is composed of two recursive systematic convolutional (RSC) encoders of rate 1/2 with generator matrix $\left[1 \; \frac{1+D+D^3+D^4}{1+D^3+D^4}\right]$ [Lin and Costello, 2004]. In the encoding of a bit plane, the turbo encoder generates all the parity bits for the bit plane, saves these bits in a buffer, and divides them into parity bit sets (64 sets in our implementation). Then, the encoder transmits one set of parity bits from the buffer. If after decoding the bit plane, the decoder detects that the residual bit error probability (BER) is above 10^{-3}, it requests an additional set of parity bits through the feedback channel. This transmission-request process is repeated until the BER is below 10^{-3}. The PDWZ decoder first decodes the K-subimage and performs up-sampling using bilinear interpolation. Then, the interpolated pixels act as SI for the decoding of the received parity bits of each bit plane. After decoding all the transmitted bit planes, MAP reconstruction is performed.

As in the case of the MPCMSI coder, each image was encoded with our PDWZ coder using all the possible values of the two coding parameters (R_{PCM} and M_{WZ}). The concave hulls of all the (R, PSNR) points obtained in the encoding of the *Lena* and *Barbara* images are shown in Figure 2.11. This figure shows that PDWZ performs better than or equal to our coder. However, the better performance of the PDWZ coder comes at the expense of a larger latency and complexity, mainly due to the turbo decoding of the bit planes and the rate allocation. The turbo encoding involves one interleaving process and two RSC encodings for each transmitted bit plane. Since an RSC encoder is a *finite state machine*, the RSC encoding can be done by using a look-up table that provides the next state and output parity bit for each present state and input bit. The turbo decoding is an iterative and much more complex process in which, apart from the interleaving and deinterleaving processes, two MAP decodings are performed in each iteration [Lin and Costello, 2004]. With our turbo code, even fast MAP decoding algorithms require more than 200 operations per bit in each iteration [Robertson et al., 1995]. Since the decoder performs several iterations (18 in our implementation) the decoding of *one* set of parity bits requires many more operations than the decoding of the entire image in MPCMSI and our algorithm.

In general, the decoder requests the transmission of several sets of parity bits from the encoder to correctly decode each bit plane. Also, several bit planes can be transmitted. For instance, to obtain the point (2.29 bpp, 30.35 dB) in the image *Barbara*, the encoder attended: 13 requests for the first bit plane, 13 requests for the second bit plane, and 26 requests for the third bit plane. Hence, 55 turbo

Figure 2.11 Comparison between the coding efficiency of MPCMSI, PDWZ, PCM, and our algorithm in the encoding of the images *Lena* and *Barbara*.

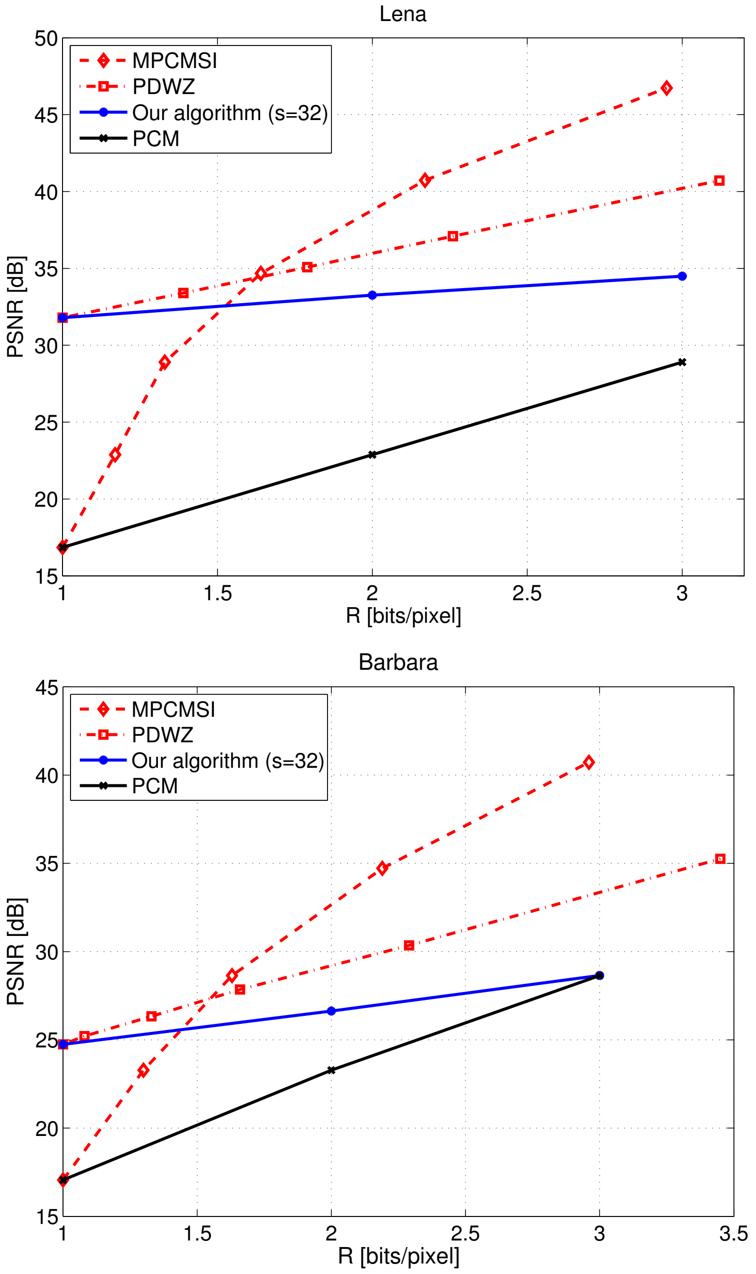


Table 2.6 PSNR (in dB) and corresponding rate (in bpp) for the image *Lena* when coded according to the JPEG coding standard.

PSNR (dB)	R (bpp)
24.83	0.142
29.94	0.230
35.00	0.539
40.14	1.601

decodings were necessary. Thus, even though the PDWZ encoder is simple, the complexity of the PDWZ decoder and the overall coding latency is very high.

Although the results in this section show that our MPCMSI and PDWZ implementations are generally more efficient than our MPCM algorithm, it should be taken into account that these results represent the optimal performance of our MPCMSI and PDWZ implementations. In practice, these coders should analyze the image statistics in order to properly assign values to their coding parameters. This analysis would increase the complexity of the encoders even more. Moreover, errors in the assignments would decrease their coding efficiency with respect to the results shown in this section.

To give an idea of how the low-complexity coders perform in terms of rate-distortion compared to a conventional JPEG coder, we indicated in Table 2.6 the number of bits per pixel (bpp) that are needed to code the image *Lena* (512×512 pixels, 8 bpp) with a quality of approximately 25, 30, 35 and 40 dB[5]. However, it should be noted that the complexity of JPEG coding is much higher than the complexity of the MPCMSI and PDWZ coding described in this section. As discussed above, the MPCMSI and PDWZ algorithms are on their turn much more complex than our modulo-PCM algorithm.

It is also interesting to compare the number of frames per second (fps) that can be coded using these three algorithms (our algorithm, MPCMSI and PDWZ). To make these measures, we coded the image Lena (512×512 pixels, 8 bpp) thousand times with each algorithm on a computer with an Intel Core 2 Duo (2.13 GHz) processor, and measured the execution times. In all these cases, only the time for the coding was considered (i. e., we assume that the values of the coding parameters are available)[6].

Our algorithm was able to code at 306,7 fps. The use of different values of the

[5] These results are obtained using the JPEG implementation of the OpenCV library.

[6] Note that these execution times are obtained without any type of optimization in the compilation of the code for the three algorithms. Hence, these execution times are especially interesting to see how the three coders behave with respect to each other. If one would like to compare these execution times with the execution times of other coders that use optimized code, similar optimization techniques should be used for the proposed coder to make a fair comparison.

coding parameters does not have an appreciable effect on the coding times. When using the MPCMSI algorithm, frame rates of 64,85 fps are obtained. The frame rate increases to 74,07 fps when the MPCMSI coder only codes the SI (i. e., when the MPCMSI behaves as a DPCM coder with entropy coding). The execution times for PDWZ video coding are much larger and depend on the number of transmitted bit planes per image. In particular, the frame rate that can be achieved with the PDWZ coder is 17.34 fps when one bit plane is transmitted, 13.59 fps when two bit planes are transmitted, and 10.82 fps when three bit planes are transmitted.

2.7 Conclusion

In this chapter, we have presented a very simple MPCM coding algorithm that reduces the rate of the PCM signals while still preserving the advantageous properties of PCM (random access and rate scalability). After analyzing the signal statistics and assigning proper values to the coding parameters, the encoder simply discards a specified number of LSBs and MSBs of each sample. The decoder attempts to recover the removed bits by using the received bits and a SI that is generated using interpolation.

Experimental results obtained in the encoding of several digital images show that our algorithm has a better objective and subjective performance than PCM at low rates. At high rates, Modulo-PCM and PCM provide similar results. Our algorithm has a worse rate-distortion performance than other source coding techniques such as MPCM coding with side information or Wyner-Ziv video coding, but it has the advantage of a much lower computational complexity (comparable to PCM). This makes our algorithm very useful in applications that require extremely simple encoders such as the encoding of video signals from high-speed cameras.

The work described in this chapter has led to one journal publication, which is currently under review [Prades-Nebot et al., 2010].

3
Pixel-Domain Distributed Video Coding

3.1 Introduction

In some video applications, it is desirable to reduce the complexity of the video encoder at the expense of a more complex decoder. Examples of such applications are wireless low-power surveillance, wireless PC cameras, multimedia sensor networks, disposable cameras, and mobile camera phones. Distributed video coding is a new paradigm that fulfills this requirement by performing intra-frame encoding and inter-frame decoding [Puri and Ramchandran, 2002]. Hence, most of the computational load is moved from the encoder to the decoder, since in this case the distributed video *de*coders (and not the *en*coders) perform motion estimation and motion compensated interpolation. Two theorems from information theory, namely the Slepian-Wolf (SW) theorem [Slepian and Wolf, 1973] for lossless distributed source coding and the Wyner-Ziv (WZ) theorem [Wyner and Ziv, 1976] for lossy source coding with side information, suggest that such a system with intra-frame encoding and inter-frame decoding can come close to the efficiency of a traditional inter-frame encoding-decoding system.

The most common practical distributed video coders are Wyner-Ziv video coders implemented with error correcting codes such as

- syndrome codes [Puri and Ramchandran, 2002, Xu and Xiong, 2006, Xu and Xiong, 2004],
- turbo codes [Aaron and Girod, 2002, Aaron et al., 2004, Aaron et al., 2002, Ascenso et al., 2005a, Ascenso et al., 2005b, Belkoura and Sikora, 2006a, Belkoura and Sikora, 2006b, Brites et al., 2006a, Brites et al., 2006c, Brites et al., 2006b, Dalai et al., 2006a, Morbee et al., 2007b, Roca et al., 2007, Trapanese et al., 2005, Tagliasacchi et al., 2006] and
- low-density parity-check (LDPC) codes [Liveris et al., 2002, Westerlaken et al., 2006, Xu and Xiong, 2006, Xu and Xiong, 2004, Cheung and Ortega, 2006, Liu et al., 2006].

Some proposed coding schemes apply Wyner-Ziv coding to the pixel values of the video signal and are therefore called pixel-domain Wyner-Ziv (PDWZ) video coders [Aaron et al., 2002, Ascenso et al., 2005a, Ascenso et al., 2005b, Belkoura and Sikora, 2006a, Belkoura and Sikora, 2006b, Brites et al., 2006a, Brites et al., 2006c,Dalai et al., 2006a,Morbee et al., 2007b,Roca et al., 2007,Tagliasacchi et al., 2006,Trapanese et al., 2005]. Other approaches exploit the statistical dependencies within a frame by applying an image transform and are categorized as transform-domain Wyner-Ziv video coders [Xu and Xiong, 2006, Xu and Xiong, 2004, Liu et al., 2006, Aaron et al., 2004, Brites et al., 2006b, Cheung and Ortega, 2006, Puri and Ramchandran, 2002].

In this chapter, we focus on the turbo code-based pixel-domain Wyner-Ziv video coding architecture, as it is well known in literature [Ascenso et al., 2005a, Dalai et al., 2006a, Belkoura and Sikora, 2006b, Brites et al., 2006a, Trapanese et al., 2005, Morbee et al., 2007b, Ascenso et al., 2005b]. This coder is described in detail in Section 3.2.

To get a better insight into the functioning of this coder, we start with an in-depth study of the coding distortion introduced by pixel-domain Wyner-Ziv video coders. In particular, we present a model of the coding distortion, which can be used to determine the optimal value of coding parameters under certain coding constraints. As an example, we show how our model can be used to select the quantization step size of each video frame so that a target distortion can approximately be met. This distortion analysis is discussed in Section 3.3. The research work described in this section was performed in close collaboration with Antoni Roca.

Subsequently, the shortcomings of the pixel-domain Wyner-Ziv video coder are identified and tackled. One of the most difficult tasks in Wyner-Ziv video coding is allocating a proper number of bits to encode each video frame. This is mainly because, firstly, the encoder does not have access to the motion estimation information, since the motion estimation and compensation is performed at the decoder. Secondly, small variations in the allocated number of bits can cause large changes in distortion because of the threshold effect of the channel codes used in distributed video coders [Gunduz and Erkip, 2006]. Most Wyner-Ziv video coders solve this problem by using a feedback channel (FBC), which allows the decoder to request additional bits from the encoder when needed. Although this way an optimal rate is allocated, it is not a valid solution in unidirectional and offline applications, and increases the decoder complexity and latency [Brites et al., 2006a].

In Section 3.4 and Section 3.5, these problems are studied and solutions are formulated. Firstly, in Section 3.4 we propose a rate allocation (RA) algorithm for pixel-domain distributed video (PDDV) coders that do not use a feedback channel. Our algorithm computes the number of bits to encode each video frame without significantly increasing the encoder complexity. The experimental results show that the rate allocation algorithm delivers good estimates of the rate and the frame qualities provided by our algorithm are quite close to the ones provided by a feedback channel-based algorithm.

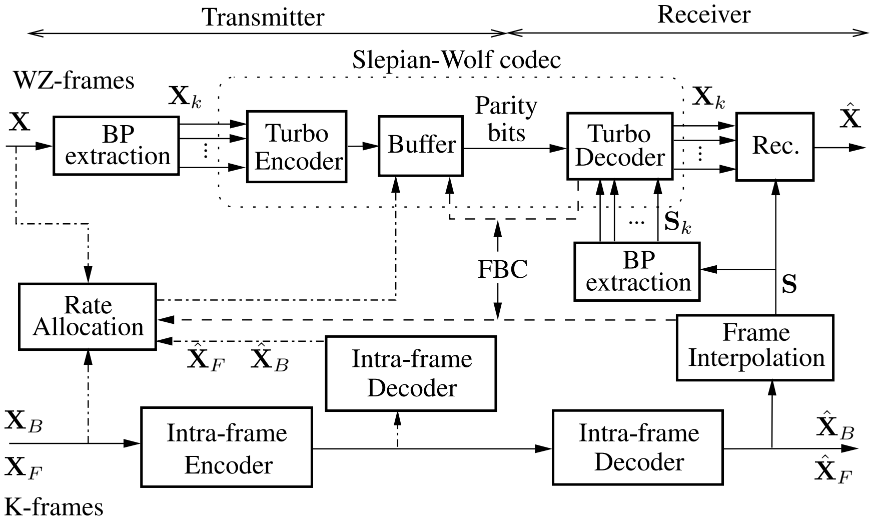
Secondly, in Section 3.5, we study a pixel-domain distributed video coder with feedback channel. This feedback channel allows us to allocate an optimal rate but has several drawbacks:

- Due to the multiple bit requests (and the corresponding multiple decodings) the computational complexity of the decoder increases significantly. In [Belkoura and Sikora, 2006a], it is shown that the overall workload in WZ video coding often exceeds that of conventional coders, such as H.264.

- The feedback channel introduces latency because each bit request causes an additional delay in the decoding of each frame [Brites et al., 2006a].

To overcome these feedback channel problems, we propose a rate allocation algorithm for pixel-domain Wyner-Ziv video coders. This algorithm reduces the number of bit requests from the decoder over the feedback channel and simultaneously keeps the computational load for the encoder low. The final aim is to reduce the decoder complexity and the latency to a minimum, while maintaining very near-to-optimal rate-distortion (RD) performance. This method is related to Section 3.4, where we study a rate allocation algorithm for pixel-domain Wyner-Ziv video coders without feedback channel. However, in Section 3.4 we focus on the pixel-domain Wyner-Ziv video coder with feedback channel. We utilize this feedback channel to improve the rate allocation and to achieve very near-to-optimal rate allocation while at the same time eliminating the main feedback channel inconveniences. Moreover, in Section 3.4, the estimation of the encoding rate is based on experimentally obtained performance graphs of the turbo codes, while in

Figure 3.1 General block diagram of a scalable pixel-domain Wyner-Ziv video coder.



Section 3.4 we derive expressions for the encoding rate founded on information theory concepts.

The chapter is organized as follows. In Section 3.2, we study the basics of pixel-domain Wyner-Ziv video coding. In Section 3.3, we propose a model for the distortion introduced by this coder, and use it to minimize the quality fluctuations of decoded frames. Subsequently, in Section 3.4, we explain how the feedback channel can be removed from the scheme, and which influence this has on the rate-distortion performance. In Section 3.5, we study the pixel-domain Wyner-Ziv coder with feedback channel, and show how we can eliminate the main feedback channel inconveniences, i.e., its negative impact on latency and decoder complexity. Finally, the conclusions are drawn in Section 3.6.

3.2 Pixel-domain Wyner-Ziv video coder

In this section, we describe in detail the turbo-code based scalable pixel-domain Wyner-Ziv video coder that will be used in the analysis of the subsequent sections.

3.2.1 General scheme

In this section, we review the basics of pixel-domain Wyner-Ziv video coding. In Wyner-Ziv video coding, the frames are organized into key(K) frames and

Wyner-Ziv frames. The key frames are coded using a conventional intra-frame coder. The Wyner-Ziv frames are coded using the Wyner-Ziv paradigm, i.e., they are intra-frame encoded, but they are conditionally decoded using side information (Figure 3.1). In most Wyner-Ziv video coders, the odd frames are encoded as key frames, and the even frames are encoded as Wyner-Ziv frames [Aaron et al., 2002, Ascenso et al., 2005a]. Coding and decoding is done unsequentially in such a way that, before decoding the Wyner-Ziv frame $\mathbf{X}$, the preceding and succeeding key frames ($\mathbf{X}_{\mathrm{B}}$ and $\mathbf{X}_{\mathrm{F}}$) have already been transmitted and decoded. Thus, the receiver can obtain a good approximation $\mathbf{S}$ of $\mathbf{X}$ by interpolating its two closest decoded frames ($\hat{\mathbf{X}}_{\mathrm{B}}$ and $\hat{\mathbf{X}}_{\mathrm{F}}$). $\mathbf{S}$ is used as part of the side information to conditionally decode $\mathbf{X}$, as will be explained below.

In this chapter, we focus on the practical pixel-domain Wyner-Ziv video coder depicted in Figure 3.1 [Ascenso et al., 2005a, Brites et al., 2006a, Morbee et al., 2007b]. In the following sections, we will study this coder in detail. In Section 3.2.1.1, we discuss the transmitter side of this coder. In Section 3.2.1.2, we explain the receiver side of this coder.

3.2.1.1 Encoder

At the transmitter side of the scheme depicted in Figure 3.1, we first extract the M bit planes (BPs) $\mathbf{X}_k$ ($1 \leq k \leq M$) from the Wyner-Ziv frame $\mathbf{X}$. M is the number of bits by which the pixel values of $\mathbf{X}$ are represented. Subsequently, only the m most significant bit planes $\mathbf{X}_k$ ($1 \leq k \leq m, 1 \leq m \leq M$) are encoded independently of each other by a Slepian-Wolf coder [Slepian and Wolf, 1973, Ascenso et al., 2005a, Ascenso et al., 2005b, Tagliasacchi et al., 2006, Brites et al., 2006c, Dalai et al., 2006b, Morbee et al., 2007b]. The other bit planes $\mathbf{X}_k$ ($m + 1 \leq k \leq M$) are not encoded and are simply discarded. These discarded bit planes can be recovered at the decoder by using the side information and other previously decoded bit planes, as will be explained later by Eq. 3.2. Discarding bit planes is a first way of achieving compression with this coder.

The higher m, the higher the encoding rate, but the lower the distortion. The value of parameter m can be fixed along the sequence [Ascenso et al., 2005a, Ascenso et al., 2005b, Tagliasacchi et al., 2006, Brites et al., 2006c, Dalai et al., 2006b, Morbee et al., 2007b] or can be adaptively changed to fulfil the coding constraints [Roca et al., 2007]. The transmission and decoding of bit planes is done in order of significance (the most significant bit planes are transmitted and decoded first). The Slepian-Wolf coding is implemented with efficient channel codes that yield the parity bits of $\mathbf{X}_k$, which are *partially* transmitted over the channel. Transmitting only part of the parity bits generated for a bit plane is a second way of achieving compression with this coder. Let us denote the parity bits transmitted for bit plane $\mathbf{X}_k$ by PB_k. To determine the number of parity bits of a bit plane $\mathbf{X}_k$ to be transmitted (or in other words, the number of bits included in PB_k), a rate-adaptive channel coder together with a feedback channel and/or an adequate rate allocation algorithm is used. This will be discussed in detail in Sections 3.4.1, 3.4.2, 3.5.1 and 3.5.2.

The coder depicted in Figure 3.1 is a *scalable* coder. Indeed, if the m most significant bit planes are encoded (independently of each other) by the Slepian-Wolf coder, $m+1$ different decodable bit streams $\{\mathrm{BS}_0, \ldots, \mathrm{BS}_k, \ldots, \mathrm{BS}_m\}$ can be generated for each Wyner-Ziv frame $\mathbf{X}$, where each possible bit stream BS_k contains the parity bits of the k most significant bit planes:

$$\mathrm{BS}_k = \{\mathrm{PB}_1, \ldots, \mathrm{PB}_k\} \tag{3.1}$$

When $k = 0$, no bit plane is transmitted (BS_0 does not contain any parity bits) and each decoded frame $\mathbf{X}$ is equal to $\mathbf{S}$. Consequently, with the *scalable* coder [Ascenso et al., 2005a, Brites et al., 2006a, Morbee et al., 2007b] shown in Figure 3.1, $m+1$ different rate-distortion points are possible.

In *non-scalable* pixel-domain Wyner-Ziv video coders [Aaron et al., 2002], each Wyner-Ziv frame is quantized using a uniform quantizer and the Slepian-Wolf coder directly encodes the quantization indexes. In these coders, once the quantizer step size Δ of $\mathbf{X}$ has been set, only *one* rate-distortion point is possible. The scalable coders have the advantages that the rate can be flexibly adapted and that the rate control is easier than in the non-scalable case.

3.2.1.2 Decoder

At the receiver side of the scheme depicted in Figure 3.1, the Slepian-Wolf decoder obtains the original bit plane $\mathbf{X}_k$ from the transmitted parity bits PB_k, the corresponding bit plane $\mathbf{S}_k$ extracted from the interpolated frame $\mathbf{S}$, and the previously decoded bit planes $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$. It is interesting to notice that $\mathbf{S}_k$ can be considered the result of transmitting $\mathbf{X}_k$ through a noisy *virtual channel*. The Slepian-Wolf decoder is a channel decoder that recovers $\mathbf{X}_k$ from its noisy version $\mathbf{S}_k$.

Finally, the decoder obtains the reconstruction $\hat{X}$ of each pixel $X \in \mathbf{X}$ by using the decoded bits $X_k \in \mathbf{X}_k$ ($k = 1, \ldots, m$) and the corresponding pixel S of the interpolated frame $\mathbf{S}$ through

$$\hat{X} = \begin{cases} X_\mathrm{L}, & S < X_\mathrm{L} \\ S, & X_\mathrm{L} \leq S \leq X_\mathrm{R} \\ X_\mathrm{R}, & S > X_\mathrm{R} \end{cases} \tag{3.2}$$

with

$$X_{\mathrm{L}} = \sum_{i=1}^{m} X_i 2^{M-i} \quad \text{and} \quad X_{\mathrm{R}} = X_{\mathrm{L}} + 2^{M-m} - 1. \tag{3.3}$$

Note that the collection of decoded bit planes $\{\mathbf{X}_1, \ldots, \mathbf{X}_k\}$ constitutes a quantized version of $\mathbf{X}$ using a uniform quantizer of k bits with step size $\Delta = 2^{M-k} - 1$. The larger k, the smaller Δ and the lower the distortion of the decoded video. On the other hand, for each decoded bit plane $\mathbf{X}_k$ the parity bits PB_k were transmitted. Hence, the larger k, the larger the rate R of the encoded bit stream ($\mathrm{BS}_k = \{\mathrm{PB}_1, \ldots, \mathrm{PB}_k\}$).

3.2.2 Turbo code-based coder

The Slepian-Wolf coder studied in this chapter is implemented with turbo codes (TC) (see Figure 3.1).

The bit planes $\mathbf{X}_k$ ($k = 1, \ldots, m$) are turbo-encoded using a turbo encoder (see Figure 3.1). This turbo encoder yields the parity bits of $\mathbf{X}_k$ [Rowitch and Milstein, 2000]. These parity bits are only partially transmitted, and the parity bits for bit plane $\mathbf{X}_k$ that are transmitted are denoted by PB_k, as explained in Section 3.2.1.1. The number of parity bits to transmit, is determined by communication with the turbo decoder through a feedback channel. The use of this feedback channel can be combined with an adequate rate allocation algorithm, or the feedback channel can even be replaced entirely by a rate allocation algorithm. This rate allocation problem will be the subject of Sections 3.4 and 3.5.

The parity bits PB_k are then transmitted. The turbo decoder obtains the original bit plane $\mathbf{X}_k$ from the transmitted parity bits PB_k, the corresponding bit plane $\mathbf{S}_k$ extracted from the interpolated frame $\mathbf{S}$, and the previously decoded bit planes $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$, as will be explained in the remainder of this section. Since $\mathbf{S}_k$ is considered the result of transmitting $\mathbf{X}_k$ through a noisy *virtual channel* (see Section 3.2.1.1), the turbo decoder recovers $\mathbf{X}_k$ from its noisy version $\mathbf{S}_k$. The virtual channel is assumed to be symmetric and the symbols of the bit planes are binary, so the virtual channel is modelled as a binary symmetric channel.

For a correct and efficient turbo decoding of a bit plane $\mathbf{X}_k$ it is essential to know the error probability for each bit of $\mathbf{S}_k$, i.e. the probability that this bit of bit plane $\mathbf{S}_k$ differs from the corresponding bit of $\mathbf{X}_k$ [Rowitch and Milstein, 2000]. Hence, to decode the k^{th} transmitted bit plane $\mathbf{X}_k$ of a Wyner-Ziv frame $\mathbf{X}$, the turbo decoder needs to compute the error probability of each bit of the bit plane $\mathbf{S}_k$. The way to do this is related to the method proposed in [Belkoura and Sikora, 2006a]. Apart from the received parity bits and the interpolated frame $\mathbf{S}$, we also take into account the information provided by the previously decoded bit planes $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$ of $\mathbf{X}$, as is done in [Xu and Xiong, 2006, Xu and Xiong, 2004]. In order to efficiently combine all the available pieces of information for the computation of the error probability of each bit of the bit plane $\mathbf{S}_k$, we need to statistically model the difference between an original pixel value and its corresponding side information value, which is called the *correlation noise*. The correlation noise frame $\mathbf{U}$ contains the correlation noise values for the whole image, i.e. $\mathbf{U} = \mathbf{X} - \mathbf{S}$ [Cheung et al., 2005]. As in [Aaron et al., 2002, Ascenso et al., 2005a, Morbee et al., 2007b], we assume that the value of a correlation noise pixel U of $\mathbf{U}$ follows a Laplacian distribution with a probability density function (pdf)

$$p(U) = P(X|S) = \frac{\alpha}{2}e^{(-\alpha|U|)} \tag{3.4}$$

where $\alpha = \sqrt{2}/\sigma$ and σ is the standard deviation of the correlation noise frame $\mathbf{U}$. From the $k-1$ most significant bits $\{X_1, \ldots, X_{k-1}\}$ of $X \in \mathbf{X}$ that have already been transmitted and error-freely decoded, the decoder knows that X lies in the quantization interval $[X_\mathrm{L}, X_\mathrm{R}]$ where X_L and X_R are as in (3.3) with $m = k-1$. Hence, the conditional probability density function of X given S and $X_\mathrm{L} \leq X \leq X_\mathrm{R}$ is

$$p_{\mathrm{dec}}(X|S, X_\mathrm{L} \leq X \leq X_\mathrm{R}) = \begin{cases} \dfrac{\frac{\alpha}{2}e^{-\alpha|X-S|}}{\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R}|S)} & \text{if } X_\mathrm{L} \leq X \leq X_\mathrm{R} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where the probability $\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R}|S)$ can be computed by integrating (3.21)

$$\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R}|S) = \int_{X_\mathrm{L}}^{X_\mathrm{R}} \frac{\alpha}{2}e^{-\alpha|X-S|}\,dX. \tag{3.6}$$

To derive the error probability of the k^{th} bit S_k of the pixel value S, we first observe that the decoded bit X_k will further shrink the quantization interval of X in such a way that

$$\begin{cases} X \in [X_\mathrm{L}, X_\mathrm{C}] & \text{if } X_k = 0 \\ X \in [X_\mathrm{C}+1, X_\mathrm{R}] & \text{if } X_k = 1 \end{cases} \tag{3.7}$$

where

$$X_\mathrm{C} = \left\lfloor \frac{X_\mathrm{L} + X_\mathrm{R}}{2} \right\rfloor \tag{3.8}$$

with $\lfloor y \rfloor$ denoting the floor function that returns the highest integer less than or equal to y. For the pixel value X from which the bit X_k needs to be decoded, the values X_L, X_R, and X_C can be computed from the previously decoded bits $\{X_1, \ldots, X_{k-1}\}$ using (3.3) with $m = k-1$ and (3.8). The estimate $X_k = S_k$ is erroneous if $S_k = 0$ and $X \in [X_\mathrm{C}+1, X_\mathrm{R}]$ or if $S_k = 1$ and $X \in [X_\mathrm{L}, X_\mathrm{C}]$.

Hence, the error probability of the k^{th} bit of S is estimated through

$$P_{\text{e}}(S_k) = \begin{cases} \displaystyle\int_{X_{\text{C}}+0.5}^{X_{\text{R}}} p_{\text{dec}}(X|S, X_{\text{L}} \leq X \leq X_{\text{R}})\, dX & \text{if } S_k = 0, \\ \displaystyle\int_{X_{\text{L}}}^{X_{\text{C}}+0.5} p_{\text{dec}}(X|S, X_{\text{L}} \leq X \leq X_{\text{R}})\, dX & \text{if } S_k = 1. \end{cases} \tag{3.9}$$

Note that the integration intervals are extended by 0.5 in order to cover the whole interval $[X_{\text{L}}, X_{\text{R}}]$. For the first bit plane $\mathbf{X}_1$, no previous bit planes have been transmitted and decoded and, consequently, $X_{\text{L}} = 0$, $X_{\text{R}} = 255$, and $X_{\text{C}} = 127$ for all the pixels.

Knowing the error probability $P_{\text{e}}(S_k)$ of each bit of the side information bit plane $\mathbf{S}_k$, the turbo decoder can obtain the turbo decoded bit plane $\mathbf{X}_k$ by *correcting* bit plane $\mathbf{S}_k$ with the parity bits PB_k [Rowitch and Milstein, 2000].

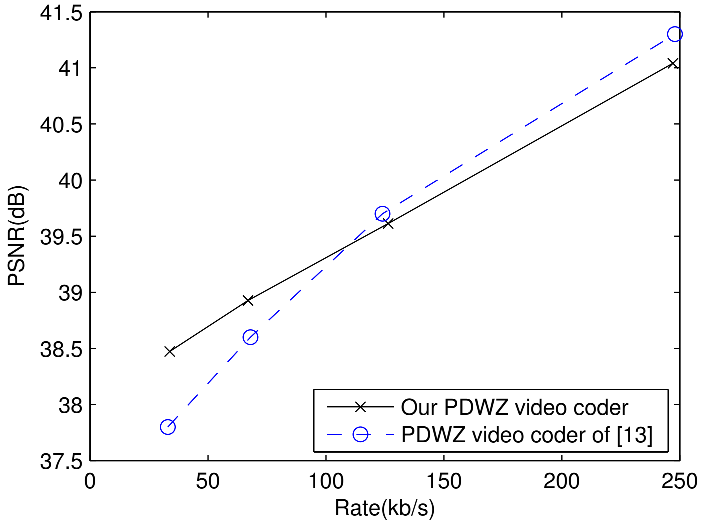
3.2.3 Comparison with literature and visual result

The described pixel-domain Wyner-Ziv video coder shares its main characteristics with the coders proposed in [Dalai et al., 2006a, Ascenso et al., 2005a, Ascenso et al., 2005b, Brites et al., 2006a, Trapanese et al., 2005]. This turbo code-based pixel domain Wyner-Ziv coding scheme is one of the most studied distributed video codecs, because of its simple and low complexity encoder architecture.

Hence, the efficiency of our coder is expected to be close to that of the coders of [Dalai et al., 2006a, Ascenso et al., 2005a, Ascenso et al., 2005b, Brites et al., 2006a, Trapanese et al., 2005]. We have verified that this holds for the test sequences for which the coding efficiency of [Dalai et al., 2006a, Ascenso et al., 2005a, Ascenso et al., 2005b, Brites et al., 2006a, Trapanese et al., 2005] has been published.

As an illustration, we have plotted the rate-distortion performance of our coder and the coder of [Dalai et al., 2006a] for the first 100 frames of the test video sequence *Foreman* in Figure 3.2. The resolution of the sequence is QCIF (176×144 pixels/frame) and the frame rate is 30 frames/s. In this experiment, only the luminance of the Wyner-Ziv frames is considered. The Wyner-Ziv frame rate is 15 frames/s, i.e. one out of two frames is a Wyner-Ziv frame, the other one is a key frame. To obtain the data shown in Figure 3.2, both coders used the same quantization parameters. In particular, the key frames are losslessly coded and each rate-distortion point corresponds to a fixed number of bit planes m sent for the encoding of the Wyner-Ziv frames ($m = 1, \ldots, 4$). Hence, Figure 3.2 provides a fair comparison between the coding efficiency of both coders. For the other sequences, the plots look similar. For the sake of conciseness, they are not shown here. For a comparison of the performance of our pixel-domain Wyner-Ziv video

Figure 3.2 Comparison between the rate-distortion performance of the pixel-domain Wyner-Ziv video coder of [Dalai et al., 2006a] and our pixel-domain Wyner-Ziv video coder for the first 100 frames of the Foreman sequence (QCIF, 30 frames/s). The key frames are losslessly coded.

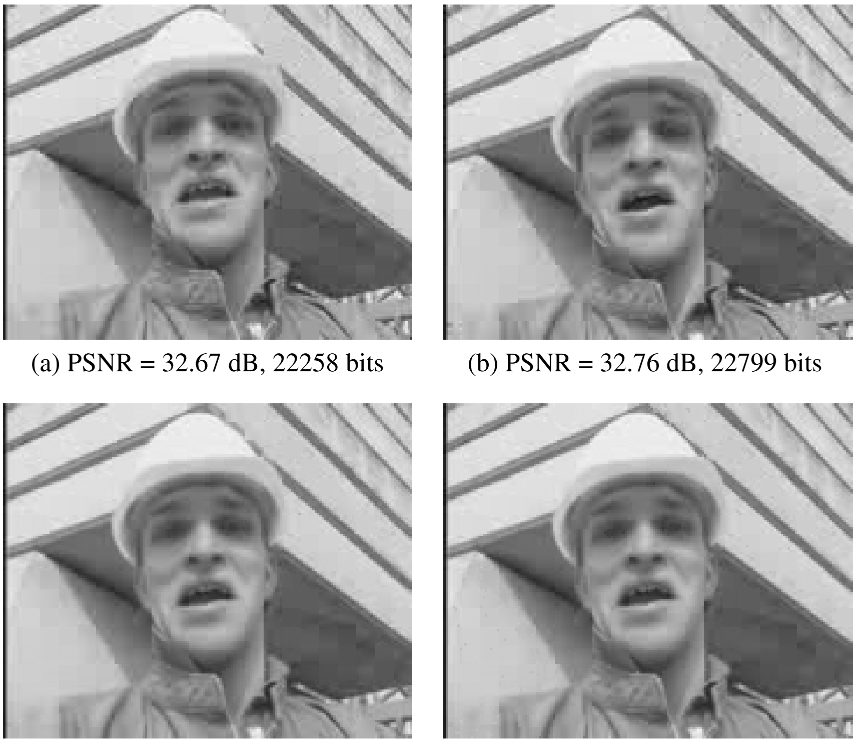


coder with existing conventional coding schemes we refer the reader to [Dalai et al., 2006a, Ascenso et al., 2005a, Ascenso et al., 2005b].

In Figure 3.3, we present a visual result of our pixel-domain Wyner-Ziv video codec. For this experiment, the key frames were coded using H.263+ with quantization parameter $QP = 10$. We show for Wyner-Ziv frame $\mathbf{X}$ number 70 of the sequence *Foreman* (QCIF, 30 frames/s) its two adjacent decoded key frames $\hat{\mathbf{X}}_{\mathrm{B}}$ and $\hat{\mathbf{X}}_{\mathrm{F}}$ (coded using H.263+ with $QP = 10$), the interpolated frame $\mathbf{S}$ after motion estimation and motion compensated interpolation at the decoder (using the method of [Ascenso et al., 2005a]), and the final reconstructed Wyner-Ziv frame $\hat{\mathbf{X}}$ when two bit planes are transmitted (or in other words $m = 2$). Below the decoded frames the PSNR of the frame and the number of bits dedicated to the encoding of the frame are indicated.

We observe that the quality of the decoded Wyner-Ziv frame is better than the quality of its adjacent decoded key frames, while the number of bits used for the encoding of the Wyner-Ziv frame is lower. This is possible, since the Wyner-Ziv frame exploits the temporal correlation in this video, and hence can be encoded with fewer bits for an equal or even better quality. Note that it is not desirable that the image quality fluctuates too much between the frames of a video. The problem of quality fluctuation will therefore be tackled in the next section, where we give

Figure 3.3 For Wyner-Ziv frame $\mathbf{X}$ number 70 of the sequence *Foreman*: (a) its preceding decoded key frame $\hat{\mathbf{X}}_{\mathrm{B}}$, (b) its succeeding decoded key frame $\hat{\mathbf{X}}_{\mathrm{F}}$, (c) the interpolated frame $\mathbf{S}$ after motion estimation and motion compensated interpolation at the decoder, and (d) the reconstructed Wyner-Ziv frame $\hat{\mathbf{X}}$ for $m = 2$. The key frames are intra-coded with H.263+ ($QP = 10$).



(a) PSNR = 32.67 dB, 22258 bits

(b) PSNR = 32.76 dB, 22799 bits

(c) PSNR = 31.82 dB

(d) PSNR = 33.04 dB, 11880 bits

a detailed study of the partition of quality between key frames and Wyner-Ziv frames.

3.3 Distortion analysis

In this section, we analyze the distortion introduced by a pixel-domain Wyner-Ziv video coder. In Section 3.3.1, we present a distortion model, which provides the coding distortion of a frame $\mathbf{X}$ as a function of the quantization step value (Δ) and a parameter α that depends on the accuracy of the frame that is used to conditionally decode $\mathbf{X}$ at the decoder. In Section 3.3.2, we show how the model can be used by pixel-domain Wyner-Ziv coders to adaptively set Δ in distortion-constrained encodings. In Section 3.3.3, the experimental results are shown and discussed. The research work discussed in this section was performed in close collaboration with Antoni Roca.

3.3.1 Distortion model

Let X and S be continuous and correlated random variables representing the signal to be encoded and the SI, respectively. Let U be the *correlation noise*, *i.e.*, $S = U + X$ with U and X being independent. Let x, s and u be realizations (random variates) of the random variables X, S and U respectively. We assume that X is distributed in $[x_{\min}, x_{\max}]$ and that a uniform quantizer with N decision intervals $[x_n, x_{n+1}]$ ($n = 0, \ldots, N-1$) of length Δ is used ($\Delta = (x_{\max} - x_{\min})/N$). Moreover, we assume U follows a Laplacian distribution with a probability density function (pdf) $f_U(u) = \alpha/2 \exp(-\alpha|u|)$, where $\alpha = \sqrt{2}/\sigma$ and σ is the standard deviation of U. As in [Ascenso et al., 2005b, Sun and Li, 2005] and Section 2.2.1 (Eq. 2.8), the reconstruction $\hat{x}$ of x is obtained through

$$\hat{x}(s, x_n, x_{n+1}) = \begin{cases} x_n & \text{if } s < x_n \\ s & \text{if } x_n \leq s \leq x_{n+1} \\ x_{n+1} & \text{if } s > x_{n+1} \end{cases} \tag{3.10}$$

where $[x_n, x_{n+1}]$ is the quantization interval that x belongs to. Note that, as in Section 2.2.1, this reconstruction function provides worse estimates than the minimum-mean-squared-error (MMSE) estimate, but this loss in performance is small except when σ^2 is large or when Δ is small. However, function (3.10) requires less computations than the MMSE estimate.

The quadratic distortion introduced in the encoding of a certain value x of X

using S as side information at the decoder is

$$D_{\mathrm{WZ}}(x) = \int_{-\infty}^{\infty} (x - \hat{x})^2 \, f_{S|X}(s|x) \, ds \tag{3.11}$$

where $f_{S|X}(s|x)$ is the conditional probability density function of S given X. As U is an additive noise, then $f_{S|X}(s|x) = f_U(s - x)$, and as U follows a Laplacian distribution, then

$$f_{S|X}(s|x) = \frac{\alpha}{2} \, e^{-\alpha|x-s|}. \tag{3.12}$$

By substituting (3.12) and (3.10) into (3.11) and solving the integral, we obtain

$$\begin{aligned} D_{\mathrm{WZ}}(x) &= \frac{2}{\alpha^2} + e^{-\alpha(x-x_n)} \left(\frac{1}{\alpha}(x_n - x) - \frac{1}{\alpha^2} \right) \\ &+ e^{-\alpha(x_{n+1}-x)} \left(\frac{1}{\alpha}(x - x_{n+1}) - \frac{1}{\alpha^2} \right) \end{aligned} \tag{3.13}$$

where $[x_n, x_{n+1}]$ is the quantization interval that x belongs to.

From (3.13), we can compute the average quadratic distortion D_{WZ} introduced in the encoding of X through

$$D_{\mathrm{WZ}} = \int_{-\infty}^{\infty} D_{\mathrm{WZ}}(x) \, f_X(x) \, dx \tag{3.14}$$

where $f_X(x)$ is the probability density function of X. By taking into account that the quantizer has N intervals, we obtain

$$D_{\mathrm{WZ}} = \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} D_{\mathrm{WZ}}(x) \, f_X(x) \, dx. \tag{3.15}$$

As in the case of images, the pixels values do not follow any statistical model, we assume X is uniformly distributed in $[x_{\min}, x_{\max}]$, and hence

$$D_{\mathrm{WZ}} = \frac{1}{x_{\max} - x_{\min}} \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} D_{\mathrm{WZ}}(x) \, dx \tag{3.16}$$

and as the quantizer is uniform, the integral in (3.16) has the same value in all the intervals and hence

$$D_{\mathrm{WZ}} = \frac{N}{x_{\max} - x_{\min}} \int_{x_n}^{x_{n+1}} D_{\mathrm{WZ}}(x) \, dx. \tag{3.17}$$

Finally, by substituting (3.13) into (3.17), and solving the integral in (3.17) we obtain

$$D_{\mathrm{WZ}} = \frac{2}{\alpha^2}\left(1 + e^{-\alpha\Delta}\right) + \frac{4}{\alpha^3\Delta}\left(e^{-\alpha\Delta} - 1\right). \tag{3.18}$$

It is interesting to note at this point that the calculation of D_{WZ} is related to the calculation of $D_{u,i,i}$ in Section 2.2.1 (see also Appendix A.1). Indeed, we assume that the number of transmitted parity bits is high enough to completely avoid turbo decoding errors, and hence no decision errors (see Section 2.2.1) can occur. As a consequence, since we made the same assumptions for the correlation noise U and the original signal X as in Section 2.2.1, the distortion introduced by the pixel-domain Wyner-Ziv coder is the same as for our modulo-PCM coder of Chapter 2 when no decision error can occur. This is the case when no binning is used ($m = 0$). For $m = 0$, the last part of (2.21) equals zero (as mentioned in Section 2.2.1), and the distortion of the modulo-PCM coder equals the distortion of the pixel-domain Wyner-Ziv video coder.

When using (3.18) to obtain the distortion in distributed video coders, the limitations of the assumed hypotheses must be taken into account. First, the pixel values of frames are discrete-amplitude values rather than continuous-amplitude values. Second, pixel values are clipped to an interval; however, to derive our model, we have assumed that the SI pixel values can have any real value. Third, the pixel value distribution in practice can be far from the uniform distribution assumed to derive (3.18). A distribution different from the uniform distribution could be used if the pixel amplitude distribution is measured. Notice, however, that this would increase the complexity of the encoder. Finally, the correlation noise distribution is, in general, more peaked and has longer tails than the assumed Laplacian distribution.

3.3.2 Frame-adaptive Δ-selection algorithm

In a pixel-domain Wyner-Ziv video coder, a quantization parameter (QP) has to be provided for both the key frames and the Wyner-Ziv frames. For the key frames, a transform-based intra-frame coder is usually used, and the quantization step size is determined by the quantization parameter. For the Wyner-Ziv frames, $M + 1$ different quantization step sizes Δ are possible. The quantization parameter has to be adaptively set in order to fulfill rate, distortion or delay constraints and to improve coding efficiency. However, in most pixel-domain Wyner-Ziv algorithms so far, all the key frames and Wyner-Ziv frames are encoded using the same QP and Δ values, respectively [Aaron et al., 2002, Ascenso et al., 2005a, Tagliasacchi et al., 2006, Brites et al., 2006c].

To select the proper QP for a given Δ, some of these algorithms encode the sequence with several QP values and then select the one that provides the lowest quality fluctuations. In the following, we call this off-line strategy the *constant* Δ *algorithm*. This algorithm does not adapt to variations in the accuracy of the side information and cannot obtain the QP and Δ values in real time.

In this section, we present an algorithm to adaptively select the quantization parameter for both key frames and Wyner-Ziv frames when a target distortion must be met. This is important since the use of our algorithm can allow a pixel-domain Wyner-Ziv video coder to fulfill a distortion constraint at the expense of a slight increase in the complexity of its encoder. Our algorithm can be used in both non-scalable and scalable pixel-domain Wyner-Ziv video coders. In non-scalable pixel-domain Wyner-Ziv video coders [Aaron et al., 2002], our algorithm provides the quantizer Δ of each Wyner-Ziv frame. In scalable pixel-domain Wyner-Ziv video coders [Ascenso et al., 2005a, Brites et al., 2006a, Morbee et al., 2007b], the Δ provided by our algorithm determines the number m of bit planes to transmit.

To determine the QP parameter of key frames, we compute, after encoding a key frame, its distortion D_{K} and compare it to the target distortion D_t. If $|D_{\mathrm{K}}-D_t|$ is below a threshold T, then the same QP value is used in the next key frame. Otherwise, we change the QP value for the next key frame in such a way that if $D_{\mathrm{K}} > D_t$, then QP = QP - 1 and if $D_{\mathrm{K}} < D_t$, then QP = QP + 1. To encode the first key frame, we use a default QP$_0$ value.

To select the proper quantization parameter QP (or equivalently, the proper quantization step size Δ) for a Wyner-Ziv frame $\mathbf{X}$, we use the distortion model of Section 3.3.1. According to (3.18), the coding distortion D_{WZ} depends on α and Δ. However, α cannot be computed at the encoder since $\mathbf{S}$ is not available there[1]. Therefore, a simple estimate $\hat{\alpha}$ of α must be first computed. Several methods to estimate α have been proposed in the literature [Morbee et al., 2007b, Brites et al., 2006c]. Then, the distortion of $\mathbf{X}$ for $\Delta_v = 2^{M-v}-1$, denoted $D_{\mathrm{WZ}}^{(v)}$, is computed for $v = 0, \ldots, M$. Finally, the optimum Δ value for $\mathbf{X}$ is chosen. Therefore, the following steps are performed by our algorithm:

1. Compute $\hat{\alpha}$ and set $D_{\mathrm{WZ}}^{(0)}$ to $2/\hat{\alpha}^2$.
2. For $v = 1, \ldots, M$, compute $D_{\mathrm{WZ}}^{(v)}$ using (3.18) with $\alpha = \hat{\alpha}$ and $\Delta = \Delta_v$.
3. Set m to the v value such that $|D_{\mathrm{WZ}}^{(v)} - D_t|$ is minimum.
4. Set the optimum Δ to $2^{M-m} - 1$

In practice, the $\hat{\alpha}$ estimates can exhibit a bias. In this case, the criterion for selecting the optimal m value (step 3), can be modified in order to reduce the effect of

[1] Note that of course $\mathbf{S}$ *could* be computed at the encoder for the purpose of estimating α, but this would increase the encoder complexity substantially and hence we would deviate from the simple encoder-complex decoder principle we are targeting at by using the distributed coding paradigm. The more source correlation estimation is performed at the encoder, the more we go back to the traditional coding schemes, where the core part of the source correlation estimation is performed at the encoder.

the bias. This is illustrated in Section 3.3.3, where we set m to the maximal v such that $D_{\mathrm{WZ}}^{(v)} \geq D_t$. When $D_{\mathrm{WZ}}^{(0)} \leq D_t$, then m was set to 0 ($\Delta = 255$).

3.3.3 Experimental Results

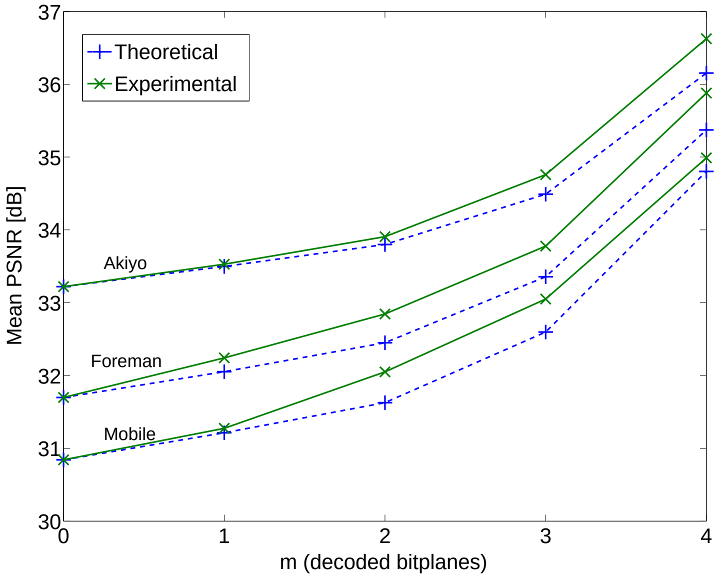
We experimentally tested the validity of the distortion model presented in Section 3.3.1 and the Δ-selection algorithm presented in Section 3.3.2. To obtain experimental results, we used a pixel-domain Wyner-Ziv video coder with the structure shown in Figure 3.1. The odd frames are encoded as key frames and the even frames are encoded as Wyner-Ziv frames [Aaron et al., 2002, Ascenso et al., 2005a, Ascenso et al., 2005b, Tagliasacchi et al., 2006, Brites et al., 2006c, Dalai et al., 2006b, Morbee et al., 2007b]. As in [Tagliasacchi et al., 2006, Brites et al., 2006c], key frames are encoded as intra-frames using a standard intra H.263+ coder. The Slepian-Wolf coder uses a rate-compatible turbo coder with a puncturing period of 32. The turbo coder is composed of two identical constituent convolutional encoders of rate 1/2 with generator polynomials $(1, 33/23)$ in octal form. The decoder uses the interpolation tools described in [Ascenso et al., 2005a] to generate the SI. Reconstruction is done using reconstruction function (3.10). The test sequences have a QCIF resolution (176×144 pixels/frame, 30 frames/second) and for the encoding only the luminance component was considered. The coding efficiency of this algorithm is similar to the one in [Dalai et al., 2006b] when they operate with the same quantization parameter values.

To test the validity of the distortion model of Section 3.3.1, we encoded the first 299 frames of the *Akiyo*, *Foreman*, and *Mobile* sequences using our pixel-domain Wyner-Ziv video coder. In each sequence, the H.263+ quantization parameter QP was set so that the mean PSNR of key frames was close to 33 dB. For each sequence, the four (and not all eight for the sake of clarity) most significant bit planes of the Wyner-Ziv frames were encoded. Thus, five bit streams $\{\mathrm{BS}_0, \ldots, \mathrm{BS}_4\}$ were generated and decoded. Finally, the PSNR values (in dB) of the Wyner-Ziv frames corresponding to each bit stream BS_m were computed and averaged. The resulting mean PSNR values for each sequence and m are shown in Figure 3.4.

For each video sequence and m value, the theoretical mean PSNR value was also computed. To do this, in each Wyner-Ziv frame, the theoretical distortion of each bit stream BS_m was computed by substituting $\Delta = 2^{8-m} - 1$ and $\alpha = \sqrt{2/\mathrm{MSE}}$ in (3.18), where MSE is the mean squared error between $\mathbf{X}$ and its interpolated frame $\mathbf{S}$. The theoretical distortion of BS_0 was set to its MSE value. Finally, the PSNR (in dB) of the Wyner-Ziv frames were computed and averaged for each m, and the results are shown in Figure 3.4.

Figure 3.4 shows that both the theoretical and the experimental curves follow the same main trends. Note that the theoretical mean PSNR is lower than the experimental mean PSNR. The main reason for this discrepancy is that, as we have already mentioned, the correlation noise distribution has larger tails than the Laplacian distribution, which provides larger distortion reductions than those predicted theoretically.

Figure 3.4 Theoretical and experimental mean PSNR of the Wyner-Ziv frames



To test the efficiency of our Δ-selection algorithm, we encoded 299 frames of several QCIF sequences using the algorithm of Section 3.3.2 with two different target PSNR (PSNR$_\mathrm{t}$) values: 30 dB and 36 dB. For the key frames, QP$_0$ was set to 10 in all the encodings, and T was set to 0.25.

For each Wyner-Ziv frame $\mathbf{X}$, $\hat{\alpha}$ needs to be estimated. Different approaches can be followed depending on how complex the estimation can be and how accurate the estimate should be.

In this section, $\hat{\alpha}$ was set to

$$\hat{\alpha} = \sqrt{2/\mathrm{MSE}'} \tag{3.19}$$

where MSE' is the mean square error between $\mathbf{X}$ and the average of its two closest decoded key frames.

This estimate increases the original encoder complexity. The complexity increase is mainly caused by the fact that key frames have to be decoded also at the encoder side. However, it should be noted that even with this increased complexity, the encoder is still much simpler than the encoders of traditional coding schemes (that perform e.g. motion estimation, that is much more computationally demanding than intra-frame coding). The advantage of this α-estimation approach is that it gives good α-estimates, especially when there is little movement in the sequence. Hence, this α-estimate allows us to assess well the validity of the distortion model and the frame-adaptive Δ-algorithm, since it is an estimate that can be made at the encoder (as it is not too complex, e.g. no motion estimation is needed), and that is at the same time an acceptably accurate estimate of the real α (such that it does not introduce too much noise in the distortion estimation).

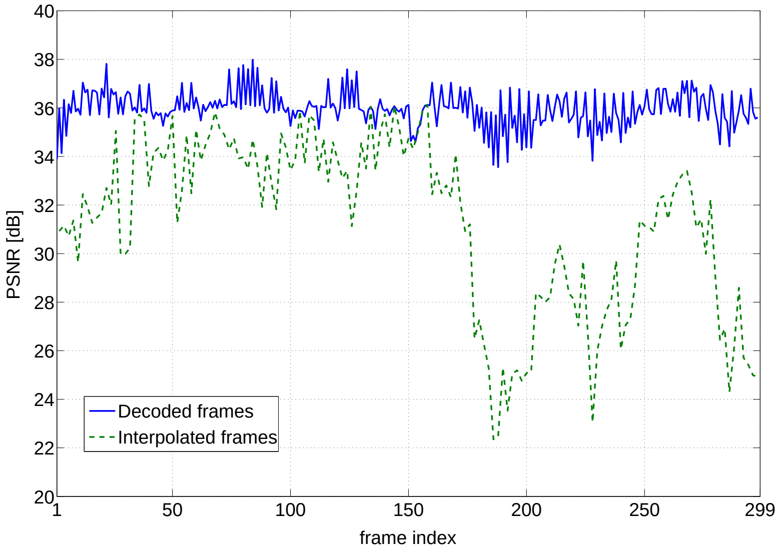
As shown above, in each Wyner-Ziv frame, our model predicts a distortion that is generally larger than the real distortion. This bias is reinforced by the fact that $\hat{\alpha}$ is an underestimate of α in most frames. Because of this bias, the algorithm of Section 3.3.2 tends to provide Wyner-Ziv frames with a distortion that is lower than D_t. As explained in Section 3.3.2 , to reduce the bias, we set m to the maximal v such that $D_{\mathrm{WZ}}^{(v)} \geq D_t$. When $D_{\mathrm{WZ}}^{(0)} \leq D_t$, then m was set to 0 ($\Delta = 255$).

Figure 3.5 shows the PSNR of each decoded frame (full line) of the sequence Carphone when $\mathrm{PSNR_t}$= 36 dB. In this figure, we can clearly observe that the quality fluctuations are very small between the different decoded frames. Moreover, the PSNR of the frames is close to $\mathrm{PSNR_t}$. Figure 3.5 also shows the PSNR of each interpolated frame $\mathbf{S}$ (dashed line). As mentioned at the beginning of this section, we used the interpolation method of [Ascenso et al., 2005a] for the generation of the side information $\mathbf{S}$. We can see that, despite the large variations in the PSNR of the interpolated frames, our algorithm selected the Δ value of each Wyner-Ziv frame so that the PSNR of this Wyner-Ziv frame was close to $\mathrm{PSNR_t}$.

Table 3.1 shows the mean PSNR (in dB) of key frames and Wyner-Ziv frames obtained after encoding several QCIF video sequences using our Δ-selection algorithm with two $\mathrm{PSNR_t}$ values (30 dB and 36 dB). Table 3.1 also shows the mean rate R (in kbps) of the Wyner-Ziv frames. As in [Aaron et al., 2002, Ascenso et al., 2005a, Ascenso et al., 2005b], the mean rate values were computed considering that the Wyner-Ziv frame rate was 15 frames/s. Note that the mean PSNR values of key frames are closer to $\mathrm{PSNR_t}$ than the mean PSNR of Wyner-Ziv frames. There are two main reasons for this. First, there are 31 different values of QP for key frames but only five different Δ values for Wyner-Ziv frames, and hence, quality can be set in a more precise way in key frames than in Wyner-Ziv frames. Second, the encoder can *exactly* compute the distortion introduced in the encoding of each key frame and set QP accordingly. The distortion of Wyner-Ziv frames, however, can only be estimated.

We compared the constant Δ algorithm and our algorithm by encoding several sequences with our pixel-domain Wyner-Ziv video coder using both strategies. Our algorithm provided PSNR values closer to the $\mathrm{PSNR_t}$ than the constant Δ

Figure 3.5 PSNR of the decoded frames (full line) and the interpolated frames (dashed line) of Carphone using our Δ-selection algorithm with PSNR$_t$=36 dB.



algorithm in most encodings. For instance, in the encoding of Carphone with PSNR$_t$=36 dB, the average absolute difference between the PSNR of each frame and PSNR$_t$ was 0.91 dB in the constant Δ algorithm but 0.53 dB in our algorithm. Therefore, despite the errors in estimating α, our algorithm better approaches the target PSNR without encoding the video sequence several times (as in the constant Δ algorithm).

3.4 Feedback channel removal

In this section, we study how we can remove the feedback channel from the scheme depicted in Figure 3.1. Therefore, we need to estimate the number of parity bits to be transmitted for each bit plane, such that this bit plane can be error-freely decoded at the decoder side. Deciding on the number of bits to sent, is called rate allocation. In Section 3.4.1, we discuss the rate allocation problem for pixel-domain Wyner-Ziv video coders. The most common way to allocate the rate at the pixel-domain Wyner-Ziv encoder is via feedback from the decoder to the encoder over a feedback channel. To remove this feedback channel, we need an algorithm

Table 3.1 The mean PSNR (in dB) of key (K) frames and Wyner-Ziv (WZ) frames, and the mean rate (in kbps) of Wyner-Ziv frames obtained using our Δ-selection algorithm with two target PSNR values (30 dB and 36 dB).

Video sequence	PSNR_t = 30 dB			PSNR_t = 36 dB		
	K-frames	WZ-frames		K-frames	WZ-frames	
	PSNR	PSNR	R	PSNR	PSNR	R
Carphone	30.1	30.5	140	36.0	36.0	419
Foreman	30.1	30.8	129	35.9	36.5	347
Mobile	30.1	30.0	113	35.8	35.8	440
Silent	30.0	30.2	119	36.0	36.6	292

that estimates at the encoder the rate without the need for communication with the decoder. We developed a method for this. This novel rate allocation algorithm is presented in Section 3.4.2. This method allows to remove the feedback channel from the scheme depicted in Figure 3.1. Finally, in Section 3.5.3, we assess how well the rate is allocated with our rate allocation method without feedback channel. We compare our rate estimations with the rate estimations obtained when using a feedback channel.

3.4.1 Rate allocation problem for pixel-domain distributed video coders

In pixel-domain Wyner-Ziv video coders, the optimum rate R^* is the *minimum* rate necessary to decode the bit planes $\mathbf{X}_k$ $(k = 1, \ldots, m)$ without bit errors[2]. The use of a rate higher than R^* does not lead to a reduction in distortion, but only to an unnecessary bit expense. On the other hand, encoding with a rate lower than R^* can cause the introduction of a large number of errors in the decoding of $\mathbf{X}_k$, which can greatly increase the distortion. This is because if the channel quality falls under a certain threshold (or, in other words the number of bit errors is too large compared to the number of received parity bits), channel codes are no longer capable of correcting the errors reliably, which leads to maximal distortion. This phenomenon is often called the threshold effect of a channel code [Gunduz and Erkip, 2006].

A common approach to estimate the optimum rate R^* in pixel-domain Wyner-Ziv video coding is the use of a feedback channel in combination with a rate-compatible punctured turbo code [Rowitch and Milstein, 2000]. In this configuration, the turbo encoder generates all the parity bits for the bit planes to be encoded,

[2] In practical pixel-domain distributed video coding, Slepian-Wolf decoders are allowed to introduce a certain small number of errors [Aaron et al., 2002, Ascenso et al., 2005a]

saves these bits in a buffer (see Figure 3.1), and divides them into parity bit sets. The size of a parity bit set is N/T_{punc}, where T_{punc} is the puncturing period of the rate-compatible punctured turbo code and N is the number of pixels in each frame. To determine the adequate number of parity bit sets to send for a certain bit plane $\mathbf{X}_k$, the encoder first transmits one parity bit set from the buffer. Then, if the decoder detects that the residual error probability Q_k is above a threshold t [Morbee et al., 2007b], it requests an additional parity bit set from the buffer through the feedback channel. This transmission-request process is repeated until $Q_k < t$. If we denote by K_k the number of transmitted parity bit sets, then the encoding rate R_k for bit plane $\mathbf{X}_k$ is

$$R_k = r\, K_k\, \frac{N}{T_{\text{punc}}}, \tag{3.20}$$

with r being the frame rate of the video.

However, although the feedback channel allows the system to allocate an optimal rate, this feedback channel cannot be implemented in offline applications or in those applications where communication from the decoder to the encoder is not possible. In those applications, an appropriate rate allocation algorithm at the encoder can take over its role. In the following section, we will propose such a rate allocation algorithm that allows us to suppress the feedback channel.

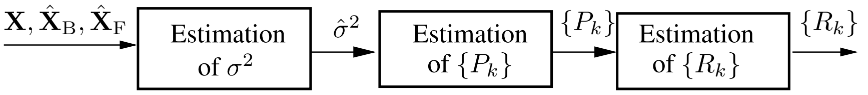
3.4.2 Rate allocation for feedback channel removal

In this section, we present a novel algorithm that estimates the rate at the encoder without the need for communication with the decoder. This method allows to remove the feedback channel from the scheme depicted in Figure 3.1.

The main idea of the proposed method is to estimate at the encoder side, for each bit plane of the Wyner-Ziv frames, the optimal (i.e. the minimal required) number of parity bits for a given residual error probability. In this respect, it is important to note that the proposed algorithm should avoid underestimation of the optimal number of parity bits. Indeed, as discussed in Section 3.4.1, if the rate is underestimated, the decoding of the bit planes of the frames will not be error-free and this will lead to a large increase in distortion. Of course, it is also preferable not to overestimate the rate, but this is less crucial, as explained in Section 3.4.1. Rate overestimation only leads to an unnecessary bit expense but not to a large image degradation. Hence, if the amount of overestimation is not large, this will not significantly influence the rate-distortion performance of the coder.

As in Section 3.3.1, let us denote by $\mathbf{U}$ the difference between the original frame and the side information frame: $\mathbf{U} = \mathbf{X} - \mathbf{S}$, and let us denote by x, s and u the realizations of the random variables X, S and U (X, S and U are pixel values of the images $\mathbf{X}, \mathbf{S}$ and $\mathbf{U}$). As in Section 3.3.1 and [Aaron et al., 2002, Ascenso

Figure 3.6 Rate allocation module at the encoder.



et al., 2005a,Morbee et al., 2007b], we assume that a pixel value $U \in \mathbf{U}$ follows a Laplacian distribution with probability density function (pdf)

$$f_U(u) = \frac{\alpha}{2} e^{(-\alpha|u|)} \tag{3.21}$$

where $\alpha = \sqrt{2}/\sigma$ and σ is the standard deviation of the difference frame $\mathbf{U}$.

Every bit plane of a Wyner-Ziv frame $\mathbf{X}$ is separately encoded. As will be explained in Section 3.4.2.2, the probability that a bit of the corresponding side information is erroneous, is different for each bit plane. Therefore, a different encoding rate R_k is allocated to each bit plane $\mathbf{X}_k$. As the virtual channel is assumed to be a binary symmetric channel (see Section 3.2.1.1), to obtain R_k, we need to know the bit error probability P_k of each bit plane $\mathbf{X}_k$. To calculate this probability, we first make an estimate $\hat{\sigma}^2$ of the parameter σ^2 (Section 3.4.2.1). Then, for each bit plane $\mathbf{X}_k$, we use $\hat{\sigma}$ to estimate P_k (Section 3.4.2.2). Once P_k is estimated, we can determine the encoding rate R_k for bit plane $\mathbf{X}_k$ by taking into account the error correcting capacity of the turbo code (Section 3.5.2.2). In Figure 3.6, a block diagram of the rate allocation module is depicted.

Although we aim at an overestimation of the rate, this is not always achieved. Therefore, once the parity bits have been decoded, the residual error probability Q_k is estimated at the decoder ($\hat{Q}_k$) (Section 3.4.2.4). If $\hat{Q}_k$ is above a threshold t, the parity bits of the considered bit plane are discarded and the frame is reconstructed with the available previously decoded bit planes. This way, we prevent an increase in the distortion caused by an excessive number of errors in a decoded bit plane. In the following, we explain each step of our rate allocation algorithm in more detail.

3.4.2.1 Estimation of σ^2

We adopt the same approach as for the estimation of α in Section 3.3.3 (Eq. 3.19). Hence, since $\sigma = \sqrt{2}/\alpha$ and with $\hat{\alpha}$ as in Eq. 3.19, $\hat{\sigma}^2$ is the mean squared error (MSE) between the current Wyner-Ziv frame and the average of the two closest decoded key frames $\hat{\mathbf{X}}_\mathrm{B}$ and $\hat{\mathbf{X}}_\mathrm{F}$:

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{(v,w) \in \mathbf{X}} \Big(\mathbf{X}(v,w) - \frac{\hat{\mathbf{X}}_\mathrm{B}(v,w) + \hat{\mathbf{X}}_\mathrm{F}(v,w)}{2} \Big)^2 \tag{3.22}$$

with N denoting the number of pixels in each frame. The decoded frames $\hat{\mathbf{X}}_{\mathrm{B}}$ and $\hat{\mathbf{X}}_{\mathrm{F}}$ are obtained by the intra-frame decoding unit at the encoder site (see Figure 3.1). We choose this σ-estimation approach because this method yields a good trade-off between calculation complexity and estimation accuracy, as we explained in Section 3.3.3.

In general, the resulting $\hat{\sigma}^2$ is an overestimate of the real σ^2 since it is expected that the motion compensated interpolation performed at the decoder to obtain the side information will be more accurate than the simple averaging of the two closest decoded key frames. This overestimation is exactly what is required for our purpose, since we prefer an overestimation of the encoding rate to an underestimation, as explained above.

3.4.2.2 Estimation of the error probabilities $\{P_k\}$

Let us assume that the most significant $k-1$ bits of the pixel value $X \in \mathbf{X}$ have already been decoded without errors. Hence, both the encoder and the decoder know from $\{X_1, \ldots, X_{k-1}\}$ that X is in the interval $[X_{\mathrm{L}}, X_{\mathrm{R}}]$ where X_{L} and X_{R} are as in (3.3) with $m = k-1$. At the encoder, the bit value X_k shrinks this interval in such way that $X \in [X_{\mathrm{L}}, X_{\mathrm{C}}]$ if $X_k = 0$, and $X \in [X_{\mathrm{C}}+1, X_{\mathrm{R}}]$ if $X_k = 1$ with

$$X_{\mathrm{C}} = \left\lfloor \frac{X_{\mathrm{L}} + X_{\mathrm{R}}}{2} \right\rfloor. \tag{3.23}$$

An error in X_k occurs if $X \in [X_{\mathrm{L}}, X_{\mathrm{C}}]$ and $S \in [X_{\mathrm{C}}+1, X_{\mathrm{R}}]$ or if $X \in [X_{\mathrm{C}}+1, X_{\mathrm{R}}]$ and $S \in [X_{\mathrm{L}}, X_{\mathrm{C}}]$. By assuming a Laplacian probability density function for the difference between the original frame and the side information, the conditional probability density function of S given X and $X_{\mathrm{L}} \le S \le X_{\mathrm{R}}$ is

$$p(S|X, X_{\mathrm{L}} \le S \le X_{\mathrm{R}}) = \begin{cases} \dfrac{\frac{\alpha}{2} e^{-\alpha|X-S|}}{\mathrm{P}(X_{\mathrm{L}} \le S \le X_{\mathrm{R}}|X)} & \text{if } X_{\mathrm{L}} \le S \le X_{\mathrm{R}} \\ 0 & \text{otherwise} \end{cases}. \tag{3.24}$$

From (3.24), the error probability of bit value X_k of pixel value X is estimated through

$$P_{\mathrm{e}}(X_k) = \begin{cases} \displaystyle\int_{X_c+0.5}^{X_{\mathrm{R}}} p(S|X, X_{\mathrm{L}} \le S \le X_{\mathrm{R}})\, dS & \text{if } X_k = 0 \\ \displaystyle\int_{X_{\mathrm{L}}}^{X_c+0.5} p(S|X, X_{\mathrm{L}} \le S \le X_{\mathrm{R}})\, dS & \text{if } X_k = 1 \end{cases} \tag{3.25}$$

Note that the integration intervals are extended by 0.5 in order to cover the whole interval $[X_{\mathrm{L}}, X_{\mathrm{R}}]$. For the first bit plane $\mathbf{X}_1$, no previous bit planes have been transmitted and decoded and, consequently, $X_{\mathrm{L}} = 0$, $X_{\mathrm{R}} = 255$, and $X_{\mathrm{C}} = 127$ for all the pixels.

Finally, we estimate the average error probability P_k for the entire bit plane $\mathbf{X}_k$. Therefore, we take into account the histogram of the frame $H(X)$, which provides the relative frequency of occurrence for each pixel value X. P_k is then estimated through

$$P_k = \sum_{X=0}^{2^M-1} H(X) P_{\mathrm{e}}(X_k). \tag{3.26}$$

3.4.2.3 Estimation of the encoding rates $\{R_k\}$

Once P_k is estimated, we choose the corresponding encoding rate R_k that enables us to decode the estimated number of errors with a residual error probability Q_k below a threshold t ($Q_k < t$). The calculation of Q_k is explained in Section 3.4.2.4. To estimate R_k, we need to express the residual error probability Q_k as a function of input error probability P_k and the number of parity bit sets K_k [Morbee et al., 2007b]. We estimate these functions experimentally by averaging simulation results over a large set of video sequences with a wide variety of properties. Using these experimental functions and knowing P_k and the threshold t, we estimate the adequate number of parity bit sets K_k. Finally, we obtain R_k from K_k through (3.20), with r the frame rate, T_{punc} the puncturing period and N the number of pixels in each frame.

3.4.2.4 Estimation of the residual error probabilities $\{Q_k\}$

If the rate allocated to encode a bit plane is too low, the decoded bit plane can contain such a large number of errors that the quality of the reconstructed frame is worse than the quality of the side information. To prevent this situation, we need to know the residual error probability Q_k of each bit plane at the decoder. We estimate Q_k as [Hoeher et al., 2000]

$$\hat{Q}_k = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{1 + e^{|L_n|}} \tag{3.27}$$

where N is the number of pixels in each frame and L_n the log-likelihood ratio of the n^{th} bit in the considered bit plane $\mathbf{X}_k$ [Hoeher et al., 2000]. If $\hat{Q}_k$ is above a certain threshold ($\hat{Q}_k > t$), the decoded bit planes are discarded and the frame is reconstructed with the available previously error-freely decoded bit planes.

3.4.3 Experimental Results

In this section, we experimentally study the accuracy of our rate allocation (RA) algorithm when it is used in a pixel-domain distributed video coder (PDDV) without feedback channel (RA-PDDV coder) and compare it with the rate allocations provided by the same coder using a feedback channel (FBC-PDDV coder).

Table 3.2 Percentage of frames that differ by ΔR from the rate of the feedback channel (for the first bit plane). The key frames are losslessly transmitted.

Video sequence	% of frames with ΔR				
	≤-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	≥+24 kb/s
Akiyo	0	0	100	0	0
Carphone	0	5.4	42.6	40.5	11.5
Foreman	1.0	2.5	25.8	30.8	39.9
Salesman	0	0	93.9	6.1	0
Mobile	0	2.0	38.5	58.1	1.4

The pixel-domain distributed video coder used in the experiments first decomposes each Wyner-Ziv frame into its 8 bit planes. Then, the m most significant bit planes are separately encoded by using a rate-compatible punctured turbo code; the other bit planes are discarded. In our experiments, m is chosen to be 3. The turbo coder is composed of two identical constituent convolutional encoders of rate 1/2 with generator polynomials $(1, 33/31)$ in octal form. The puncturing period was set to 32 which allowed our rate allocation algorithm to allocate parity bit multiples of $N/32$ bits to each bit plane, where N is the number of pixels in each frame. The key frames were either losslessly transmitted or intra-coded using H.263 with quantization parameter QP. The interpolated frame was generated at the decoder with the interpolation tools described in [Ascenso et al., 2005a].

We encoded several test QCIF sequences $(176 \times 144 \text{ pixels/frame}, 30 \text{ frames/s})$ with two rate allocation strategies: our rate allocation algorithm and the allocations provided by the FBC-PDDV coder. The threshold t for Q_k (feedback channel) and for $\hat{Q}_k$ (our rate allocation approach) is set to $\frac{1}{N}$, where N is the number of pixels in each frame.

Tables 3.2 and 3.3 show the difference between the rate allocation (in kb/s) provided by our algorithm and the rate allocation using the feedback channel when encoding the first bit plane of each frame. More specifically, the percentage of frames with a difference in rate of ΔR kb/s is shown. In Table 3.2 the key frames are losslessly coded while in Table 3.3 the key frames are intra-coded with H.263

Table 3.3 Percentage of frames that differ by ΔR from the rate of the feedback channel (for the first bit plane). The key frames are intra-coded with H.263 ($QP = 10$).

Video sequence	% of frames with ΔR				
	≤-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	≥+24 kb/s
Akiyo	0	0	47.3	52.7	0
Carphone	0	8.1	39.9	43.9	8.1
Foreman	2.5	4.0	41.4	21.7	30.3
Salesman	0	0	66.2	33.8	0
Mobile	0	0	60.8	36.5	2.7

Table 3.4 Percentage of frames that differ by ΔR from the rate of the feedback channel (for the second bit plane). The key frames are losslessly transmitted.

Video sequence	% of frames with ΔR				
	≤-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	≥+24 kb/s
Akiyo	0	0.7	87.2	10.1	2.0
Carphone	0	0.7	14.9	37.8	46.6
Foreman	0	0	19.2	24.8	56.1
Salesman	0	0.7	43.9	44.6	10.8
Mobile	0	1.4	27.0	37.8	33.8

Table 3.5 Percentage of frames that differ by ΔR from the rate of the feedback channel (for the second bit plane). The key frames are intra-coded with H.263 ($QP = 10$).

Video sequence	% of frames with ΔR				
	$\leq$-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	$\geq$+24 kb/s
Akiyo	0	0.7	0	93.9	5.4
Carphone	0	0	20.3	49.3	30.4
Foreman	0	1.5	26.3	21.2	51.0
Salesman	0	25.7	74.3	0	0
Mobile	0	0.7	31.8	37.8	29.7

Table 3.6 Percentage of frames that differ by ΔR from the rate of the feedback channel (for the third bit plane). The key frames are losslessly transmitted.

Video sequence	% of frames with ΔR				
	$\leq$-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	$\geq$+24 kb/s
Akiyo	0.7	74.3	17.6	5.4	2.0
Carphone	0	4.1	31.1	14.9	50.0
Foreman	0	0.5	17.7	8.6	73.2
Salesman	0.7	25.0	39.9	24.3	10.1
Mobile	0	0.7	16.2	27.7	55.4

Table 3.7 Percentage of frames that differ by ΔR from the rate of the feedback channel (for the third bit plane). The key frames are intra-coded with H.263 ($QP = 10$).

Video sequence	% of frames with ΔR				
	≤-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	≥+24 kb/s
Akiyo	0	0	100	0	0
Carphone	0	2.0	21.0	27.7	49.3
Foreman	0	0	4.0	19.7	76.3
Salesman	0	0	0	37.2	62.8
Mobile	0	0	2.7	18.9	78.4

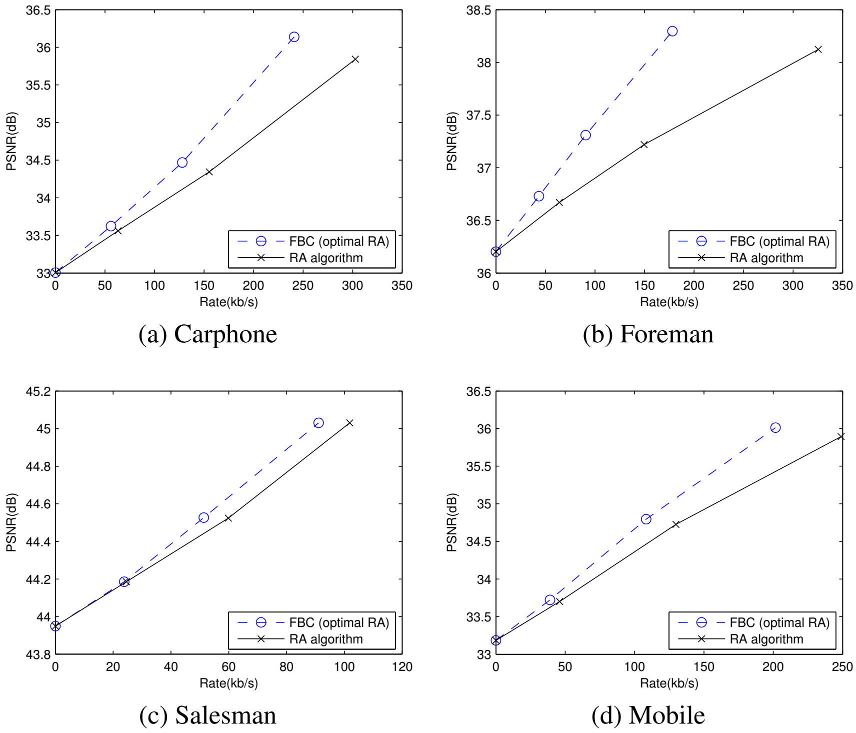
and $QP = 10$.

For the lossless case the ideal rate is allocated in between 25% and 100% of the frames (depending on the sequence). In this respect, it is important to notice that when the allocated rate is not the ideal rate, the rate is mainly overestimated, and for only very few frames the rate is underestimated. This is especially due to the fact that $\hat{\sigma}^2$ is too high (as explained in Section 3.4.2.1), which causes an overestimation of the corresponding P_k (see Section 3.4.2.2) and R_k (see Section 3.5.2.2). Rate overestimation (as opposed to rate underestimation) is particularly beneficial for our purpose, as rate underestimation leads to a large increase in distortion due to non-error-free decoding of bit planes (as explained in Section 3.4.2). Rate overestimation means a usage of bits without a corresponding increase in image quality (see Section 3.4.1), but this is less disturbing, especially since we can observe that the number of lost bits is small. In the rare cases where the rate is underestimated, we detect this based on the residual error probability estimate $\hat{Q}_k$ (Eq. 3.27). In this case, the decoded bit plane is discarded, and the frame is reconstructed based on the available error-freely decoded bit planes, as explained in Section 3.4.2.4. In sequences with little motion (*Salesman*, *Akiyo*), we allocate a more appropriate rate since the estimate $\hat{\sigma}^2$ is more accurate in this case.

The results for the case of lossy coding of the key frames are a little worse but similar. Also here, non-optimal rate allocations are nearly always overestimations.

Tables 3.4, 3.5, 3.6 and 3.7 also show the difference between the rate allocation (in kb/s) provided by our algorithm and the rate allocation using the feedback channel but now for the second and third bit plane of each frame. In Tables 3.4 and 3.6 the key frames are losslessly coded while in Tables 3.5 and 3.7 the key frames are intra-coded with H.263 and $QP = 10$. We observe that the inaccuracy of the rate allocation increases when the bit planes are less significant.

Figure 3.7 Rate-distortion performance of our rate allocation algorithm for the sequences (a) *Carphone*, (b) *Foreman*, (c) *Salesman* and (d) *Mobile*. Compared is the rate-distortion performance for the case of optimal rate allocation. The key frames are losslessly transmitted.



(a) Carphone

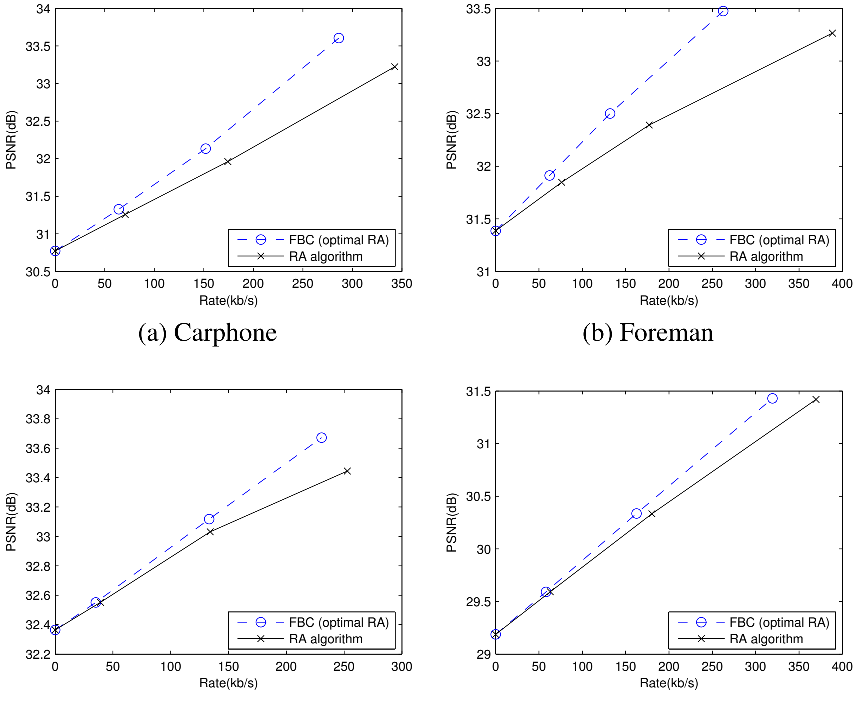
(b) Foreman

(c) Salesman

(d) Mobile

In Figures 3.7 and 3.8, we show the rate-distortion curves of *Carphone*, *Foreman*, *Salesman*, and *Mobile* for the RA-PDDV coder, and we compare them with the corresponding rate-distortion curves when, for the given puncturing period, an optimal rate is allocated (FBC-PDDV coder). In Figure 3.7 the key frames are losslessly coded while in Figure 3.8 the key frames are intra-coded with H.263 and $QP = 10$. The value of the PSNR at rate 0 shows the average quality of the interpolated frame $\mathbf{S}$. For both lossless and lossy coding of the key frames, we observe that the loss in image quality (expressed in PSNR) of the RA-PDDV coder when compared to the FBC-PDDV coder is very small (between 0 and 0.2 dB) for low rates up to 100 kb/s. The difference in image quality increases with higher rates to an extent that varies from sequence to sequence. We observe the largest loss in quality for the sequence *Foreman*, where the loss is around 1 dB for a rate of 175 kb/s. This is due to the motion in this sequence, which makes the es-

Figure 3.8 Rate-distortion performance of our rate allocation algorithm for the sequences (a) *Carphone*, (b) *Foreman*, (c) *Salesman* and (d) *Mobile*. Compared is the rate-distortion performance for the case of optimal rate allocation. The key frames are intra-coded with H.263 ($QP = 10$).



(a) Carphone

(b) Foreman

(c) Salesman

(d) Mobile

timation of σ (Eq. 3.22) less accurate and hence the estimation of the rate (which depends on σ) becomes less precise. The acceptability of this performance loss is application-dependent. Especially in applications where a feedback channel is difficult to implement or even impossible, the loss in rate-distortion performance incurred by our rate allocation algorithm will be preferred to the necessity of a feedback channel.

3.5 Reduction of decoder complexity and latency

In the previous section, we studied a pixel-domain distributed video coder without feedback channel, and we showed the consequences a feedback channel removal has on the rate-distortion performance of the pixel-domain distributed video coder.

In this section, we opt for a different approach. Instead of removing the feedback channel, we overcome two of its main inconveniences. We start from the pixel-domain distributed video coder with feedback channel (Figure 3.1) such that we can achieve an optimal rate-distortion performance, and we remove two important drawbacks of this pixel-domain distributed video coder with feedback channel: the excessive computational complexity of the decoder, and the latency due to the multiple bit requests (see Section 3.1).

To overcome these feedback channel problems, we propose a rate allocation algorithm for pixel-domain Wyner-Ziv video coders. This algorithm reduces the number of bit requests from the decoder over the feedback channel and simultaneously keeps the computational load for the encoder low. The final aim is to reduce the decoder complexity and the latency to a minimum, while maintaining very near-to-optimal rate-distortion (RD) performance. This method is related to the method of Section 3.4. However, in this section we focus on the pixel-domain Wyner-Ziv video coder *with* feedback channel. We utilize this feedback channel to improve the rate allocation. At the same time we eliminate two bothersome feedback channel inconveniences. Moreover, the estimation of the encoding rate of Section 3.4 was based on experimentally obtained performance graphs of the turbo codes, while in this section we derive expressions for the encoding rate founded on information theory concepts.

3.5.1 Decoder complexity and latency problem

A common rate allocation solution adopted in Wyner-Ziv video coders is the use of a feedback channel and a rate-compatible punctured turbo code [Rowitch and Milstein, 2000]. The functioning of this feedback channel was described in detail in Section 3.4.1.

However, this feedback channel solution has several drawbacks. Firstly, the transmission-request process increases the decoder complexity drastically since multiple parity bit decodings have to be performed for each bit plane of the Wyner-Ziv frame. More specifically, when we denote by $O_{\text{dec},k}$ the number of operations needed for the turbo decoding of the k^{th} bit plane, then the number of operations O_{dec} for the decoding of a Wyner-Ziv frame is [Belkoura and Sikora, 2006a]

$$O_{\text{dec}} = \sum_{k=1}^{m} O_{\text{dec},k} = \sum_{k=1}^{m} 2P_{\text{TC}}(W_k + 1), \tag{3.28}$$

where W_k is the number of bit requests for the decoding of the k^{th} bit plane and P_{TC} is a variable combining the parameters of the rate-compatible punctured turbo code. These parameters are discussed in detail in [Belkoura and Sikora, 2006a], and are therefore not treated here. In our setup, P_{TC} is fixed for all the decodings and is independent of W_k, so the decoder complexity depends on the number of bit requests needed for the decoding of the bit planes through the factor $\sum_{k=1}^{m}(W_k + 1)$. W_k is determined mainly by the correlation between the interpolated frame $\mathbf{S}$ and the Wyner-Ziv frame $\mathbf{X}$ for the k^{th} bit plane; this correlation is usually high for the most significant bit plane and decreases for less significant bit planes.

Note that after each bit request a parity bit set is sent from the encoder to the decoder (see Section 3.4.1). Hence, the more parity bits are needed, the more bit requests will be done. Thus, we can say that the decoder complexity depends on the number of parity bit sets needed for each bit plane, which is equivalent with Eq. 3.28, where we say that the decoder complexity depends on the number of bit requests needed for each bit plane.

Secondly, the feedback channel increases the coding latency [Brites et al., 2006a]. In fact, after sending a parity bit set, the encoder has to wait for an answer from the decoder before it can send more bits from the buffer (see Section 3.4.1). Hence, the round trip delay per bit request depends on the time needed for one turbo decoding. In particular, let us denote by $L_{\text{dec},k}$ the latency for the decoding of the k^{th} bit plane. Then, if we make abstraction of delays introduced by networking effects[3], the total latency for the coding of a Wyner-Ziv frame can be expressed as

$$L_{\text{dec}} = \sum_{k=1}^{m} L_{\text{dec},k} = \sum_{k=1}^{m} \frac{2P_{\text{TC}}(W_k + 1)}{v} \tag{3.29}$$

where W_k and P_{TC} are the same as in (3.28) and v is the processor speed (in operations/s). In our setup, P_{TC} and v are fixed for all the decodings and are independent of W_k, so the total latency depends on the number of bit requests needed for the decoding of the bit planes through the factor $\sum_{k=1}^{m}(W_k + 1)$.

As shown by (3.28) and (3.29), both the decoder complexity and the latency can be reduced by minimizing the number of bit requests, or more specifically, by reducing the factor $\sum_{k=1}^{m}(W_k + 1)$. Note that this factor yields a relative reduction of decoder complexity and latency, which is independent of the specific implementation parameters of the coder, such as P_{TC} and v. In the following section, we propose a novel rate allocation algorithm for pixel-domain Wyner-Ziv video coders with feedback channel, which provides an estimate of the optimal number of parity bit sets that have to be transmitted, thereby reducing the number of bit requests to a minimum.

[3] Additional delays are possible due to networking effects. The study of these networking effects depends on the application and the network setup, and falls out of the scope of this PhD. To make abstraction of these effects, we assume in the remainder of this chapter that the network is a *perfect* network, and consequently, the delay introduced by networking effects on the transmission of the bits is set to 0.

3.5.2 Rate allocation for complexity and latency reduction

In the proposed method, we first estimate at the encoder side, for each bit plane of the Wyner-Ziv frames, the optimal (i.e. the minimal required) number of parity bits that allows us to decode the considered bit plane (or more specifically, decode this bit plane such that the residual error probability is below a threshold). Then, this optimal number of parity bits for a certain bit plane is transmitted from the encoder to the decoder.

The decoder tries to decode the considered bit plane with the received parity bits. If the decoder is able to decode this bit plane, the process ends for this bit plane, and the transmission of a next bit plane can start. If, however, the decoder is not yet able to decode this bit plane with these first parity bits, the decoder can request an extra parity bit set from the encoder. This process of requesting an extra parity bit set is repeated until the bit plane can be correctly decoded. In other words, this process of requesting parity bit sets is equal to the process described in Section 3.4.1. However, the number of bit requests needed to decode the bit plane will be significantly reduced compared to the case described in Section 3.4.1. Indeed, since in the first step, an estimated optimal number of parity bits was sent (and not just one parity bit set unit, as was the case in Section 3.4.1), none or only a few bit requests will be needed during the coding process.

The proposed approach attempts to avoid overestimation of the optimal number of parity bits. This is an important aspect because if more bits than needed are sent in the first step, there will not be a decrease in distortion but only an unnecessary bit expense (as explained in Section 3.4.1).

Every bit plane of a Wyner-Ziv frame $\mathbf{X}$ is separately encoded. As explained in Section 3.4.2.2, the probability that a bit of the corresponding side information is erroneous, is different for each bit plane. Therefore, a different encoding rate R_k is allocated to each bit plane $\mathbf{X}_k$. A lower bound of the appropriate encoding rate R_k is estimated based on the adopted Laplacian correlation model (3.21) and the entropy of $\mathbf{X}_k$ conditional on the interpolated frame $\mathbf{S}$ and the previously decoded bit planes $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$.

Hence, our algorithm consists of two steps. Firstly, we make an estimate $\hat{\sigma}^2$ of the parameter σ^2 of the Laplacian model (Section 3.5.2.1). Secondly, for each bit plane $\mathbf{X}_k$, we use $\hat{\sigma}^2$ to estimate a lower bound of the encoding rate R_k for bit plane $\mathbf{X}_k$ by means of the conditional entropy (Section 3.5.2.2). In the following, we explain both steps of our rate allocation algorithm in more detail.

3.5.2.1 Estimation of σ^2

The true value of σ^2 can only be obtained by combining information that is only available at the encoder (the original frame $\mathbf{X}$) and information that is only available at the decoder (the interpolated frame $\mathbf{S}$). The encoder could obtain $\mathbf{S}$ by motion compensated interpolation, but, of course, that would heavily increase the encoder complexity which is undesirable in Wyner-Ziv video coding. Note in this respect that motion estimation is much more computationally demanding than intra-frame decoding of the key frames, which we will perform to obtain an accurate $\hat{\sigma}^2_{\mathrm{enc}}$ (as in Section 3.4.2.1, and see Section 3.5.2.1).

Thus, neither the decoder, nor the encoder can obtain the true value of σ^2. In [Cheung et al., 2005], the authors propose to estimate the variance by interchanging image samples between encoder and decoder. That way, the coder can estimate the variance with high precision, but this information exchange requires a large amount of feedback channel communication, and consequently, a significant additional overhead and delay.

In our approach, however, we would like to keep the feedback channel overhead and delay as low as possible. Therefore, our idea is to estimate σ^2 separately at the encoder and at the decoder side. In a next step we then combine the two estimates via the feedback channel . More specifically, we will transmit for each frame an estimate $\hat{\sigma}^2_{\mathrm{dec}}$ of σ^2 made at the decoder to the encoder through the feedback channel, so that both estimates are available at the encoder side. Transmitting $\hat{\sigma}^2_{\mathrm{dec}}$ introduces an overhead and a round trip delay. However, this overhead is negligible compared to the total bit rate spent[4]. The small latency (merely caused by the time needed for the calculation of the decoder estimate[5], since we assume a perfect network) is well compensated for by the reduction of the number of bit requests, which we will discuss in detail in Section 3.5.3. By combining $\hat{\sigma}^2_{\mathrm{dec}}$ with the estimate $\hat{\sigma}^2_{\mathrm{enc}}$ at the encoder, the risk of overestimating σ^2 is reduced.

Encoder estimate $\hat{\sigma}^2_{\mathrm{enc}}$

For the encoder estimate, we adopt the same approach as discussed in Section 3.4.2.1, where we described the estimation of σ^2 at the encoder in the case of a pixel-domain Wyner-Ziv video coder without feedback channel (see Eq. 3.22). We will denote this estimate by $\hat{\sigma}^2_{\mathrm{enc}}$.

Decoder estimate $\hat{\sigma}^2_{\mathrm{dec}}$

At the decoder, motion compensated interpolation is performed on a block-basis in order to generate the interpolated frame $\mathbf{S}$ [Ascenso et al., 2005a]. During the interpolation process of a block of the frame $\mathbf{X}$, the best matching blocks in $\hat{\mathbf{X}}_{\mathrm{B}}$ and $\hat{\mathbf{X}}_{\mathrm{F}}$ are searched using a minimum MSE criterion. Assuming linear

[4] If we represent $\hat{\sigma}^2_{\mathrm{dec}}$ by 8 bits, which are sent for every Wyner-Ziv frame, of which there are typically 15 frames/second (as in e.g. Section 3.5.3), then this is an overhead of 0.120 kbits/second.

[5] This calculation is much less complex than the number of calculations needed for the total decoding in a pixel-domain Wyner-Ziv coder, as discussed in Section 2.6.3. The exact number of operations needed for this σ^2-estimate, can be deduced from the discussion in Section 2.4.2 and Section 2.6.2. Additionally, the calculation of this σ^2-estimate can be simplified with similar techniques as described in Section 2.4.2 and Section 2.6.2.

motion between $\hat{\mathbf{X}}_{\text{B}}$ and $\hat{\mathbf{X}}_{\text{F}}$, we generate the interpolated pixels that constitute the frame $\mathbf{S}$ by taking the average of the corresponding (i.e., as matched by the motion estimation) pixels in the key frames [Ascenso et al., 2005a]. Or in other words, if we assume linear motion, the pixel values of the two key frames contribute equally to the pixel values of $\mathbf{S}$. Then, the estimate of the variance between the original frame $\mathbf{X}$ and the interpolated frame $\mathbf{S}$ is [Brites et al., 2006c]:

$$\hat{\sigma}^2_{\text{dec}} = \frac{1}{4}\frac{1}{N} \sum_{(v,w)\in\mathbf{S}} \Big(\hat{\mathbf{X}}_{\text{B}}(v-dv, w-dw) - \hat{\mathbf{X}}_{\text{F}}(v+dv, w+dw)\Big)^2 \quad (3.30)$$

where (v, w) corresponds to the pixel location in $\mathbf{S}$ and $(2dv, 2dw)$ denotes the motion vector between the corresponding pixels $(v-dv, w-dw)$ and $(v+dv, w+dw)$ in $\hat{\mathbf{X}}_{\text{B}}$ and $\hat{\mathbf{X}}_{\text{F}}$, respectively.

Combining $\hat{\sigma}^2_{\text{enc}}$ and $\hat{\sigma}^2_{\text{dec}}$

The main advantage of estimating σ^2 at the encoder, is that we have the original image at our disposal. The main disadvantage of estimating σ^2 at the encoder is that we do not have the motion information available, since the motion estimation is performed at the decoder. Hence, the encoder estimate will be accurate for video sequences with little motion. For sequences with large motion, this estimate will be in most cases an overestimate.

The decoder, on the other hand, has the motion information available, but does not dispose of the original frame. In this respect, we experimentally tested that in the case of high-quality intra-coding of the key frames (i.e. small quantization parameter QP), the decoder estimate will be more accurate than the encoder estimate if there is a lot of movement in the scene (e.g. for sequences like *Foreman*). However, in the case of low-quality intra-coding of the key frames (i.e. large quantization parameter QP), the decoder estimate will be in most cases less accurate than the encoder estimate, even if there is a lot of motion in the scene. Hence, for large QP, it is more important to know the original frame (available at the encoder) than the motion information (available at the decoder) to estimate σ^2. We also observed that for large QP - which is when the decoder estimate is less accurate than the encoder estimate, as just explained - the decoder estimate is merely an underestimate. This is logical, since if QP is large, the decoder estimate does not proportionally increase with the distortion introduced by the low-quality coding of the key frames, since this distortion is not explicitly part of the decoder estimate. At the encoder, this distortion can be easily estimated since we dispose of both the decoded key frames and the original frame.

Taking these results into consideration, we determine the estimate of σ^2 that we need for our purpose. In this respect, it is important to note that we want to avoid overestimating the optimal number of parity bits. Hence, we need to avoid overestimating σ^2. Therefore, we propose

$$\hat{\sigma}^2 \;=\; \min(\hat{\sigma}^2_{\text{enc}},\, \hat{\sigma}^2_{\text{dec}}). \tag{3.31}$$

Experimental results on 10 test sequences show that, in 97% of the cases, $\hat{\sigma}^2 \leq \sigma^2$, which is exactly what is required for our purpose. In most of the cases, this σ^2 will be the decoder estimate since this estimate tends to be the lowest one. However, in some cases, for example for the sequence *Mobile* for small QP, the encoder estimate is for some frames the lowest estimate. Note that, to avoid unnecessary computations at the encoder, one could consider not to calculate the estimate of σ^2 at the encoder, since for most of the frames, we can rely on the decoder estimate. In this respect, we have ascertained that leaving out the encoder estimate only has a small influence on the experimental results presented in Section 3.5.3.

3.5.2.2 Estimation of the encoding rates $\{R_k\}$

The estimation of the encoding rates R_k for the bit planes $\mathbf{X}_k$ is related to the algorithm described in Section 3.2.2 to estimate the error probabilities of the bits of the bit planes $\mathbf{S}_k$ (extracted from $\mathbf{S}$) at the decoder. Note that at the encoder, we know all the bit planes of frame $\mathbf{X}$ but not the corresponding interpolated frame $\mathbf{S}$; at the decoder, however, we know $\mathbf{S}$ but only the previously decoded bit planes of $\mathbf{X}$. More specifically, to estimate the required number of bits to encode a bit X_k of the k^{th} bit plane $\mathbf{X}_k$, we observe that when encoding the k^{th} bit of a pixel $X \in \mathbf{X}$ the most significant $k-1$ bits of this pixel X have already been decoded without errors. Hence, the decoder is aware of $\{X_1, \ldots, X_{k-1}\}$ and the corresponding pixel S of the interpolated frame $\mathbf{S}$. Consequently, the minimum number of bits $B(X_k)$ to encode a bit X_k of bit plane $\mathbf{X}_k$ is the entropy of X_k conditional on S and the previously decoded bits $\{X_1, \ldots, X_{k-1}\}$:

$$B(X_k) = H(X_k|S, X_1, \ldots, X_{k-1}) \tag{3.32}$$

Applying the chain rule, we derive

$$B(X_k) = H(X_1, \ldots, X_k|S) - \sum_{i=1}^{k-1} B(X_i) \tag{3.33}$$

and further,

$$\begin{aligned} B(X_k) &= \sum_{s=0}^{255} f_S(s) H(X_1, \ldots, X_k | S = s) - \sum_{i=1}^{k-1} B(X_i) \qquad (3.34) \\ &= -\sum_{s=0}^{255} f_S(s) \sum_{x_1=0}^{1} \ldots \sum_{x_k=0}^{1} \mathrm{P}(X_1 = x_1, \ldots, X_k = x_k | S = s) \\ & \qquad \log_2 \mathrm{P}(X_1 = x_1, \ldots, X_k = x_k | S = s) - \sum_{i=1}^{k-1} B(X_i) \qquad (3.35) \end{aligned}$$

where $f_S(s)$ is the probability density function (pdf) of $S \in \mathbf{S}$. As the interpolated frame $\mathbf{S}$ is not available at the encoder, we use instead of $f_S(s)$ the probability density function of X, $f_X(x)$, since both probability density functions can be considered very similar[6] By s and x we denoted the possible outcomes of X and S which are $\in \{0, \ldots, 255\}$ and by $x_1, \ldots, x_k$ we denoted the possible outcomes of $X_1, \ldots, X_k$ which are $\in \{0, 1\}$. In practice, we estimate $f_X(x)$ through the histogram of the Wyner-Ziv frame $\mathbf{X}$. $\mathrm{P}(X_1, \ldots, X_k | S)$ can be computed from the assumed Laplacian probability density function of U (3.21) with the estimated parameter $\hat{\sigma}^2$ (3.31). More concretely,

$$\mathrm{P}(X_1, \ldots, X_k | S) = \mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R} | S) \qquad (3.36)$$

where $\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R} | S)$ is as in (3.6) with X_L and X_R as in (3.3) (with $m = k$). By using (3.35), we can now compute $B(X_k)$ by calculating recursively $B(X_i)$ $(i = 1, \ldots, k-1)$ starting from $i = 1$.

Finally, the minimum encoding rate R_k for bit plane $\mathbf{X}_k$ is

$$R_k = r \; N \; B(X_k) \qquad (3.37)$$

with r being the frame rate of the video and N being the number of pixels in each frame.

By using (3.20), the number of parity bit sets to be transmitted K_k is estimated through

$$K_k = \left\lceil \frac{R_k}{N / T_\mathrm{punc}} \right\rceil + 1 \qquad (3.38)$$

where T_punc is the puncturing period of the rate-compatible punctured turbo code

[6] Since motion-compensated filtering can be seen as a low-pass filter, the probability density function of S could be estimated more accurately as a convolution of the probability density function of X with a low-pass filter. However, since $f_X(x)$ and $f_S(s)$ are already very similar, the influence of the use of this extra filter is negligible.

and $\lceil y \rceil$ denotes the ceiling function that returns the smallest integer not less than y. The last term of the sum is a rate margin which is applied to compensate for the sub-optimality of the adopted turbo code.

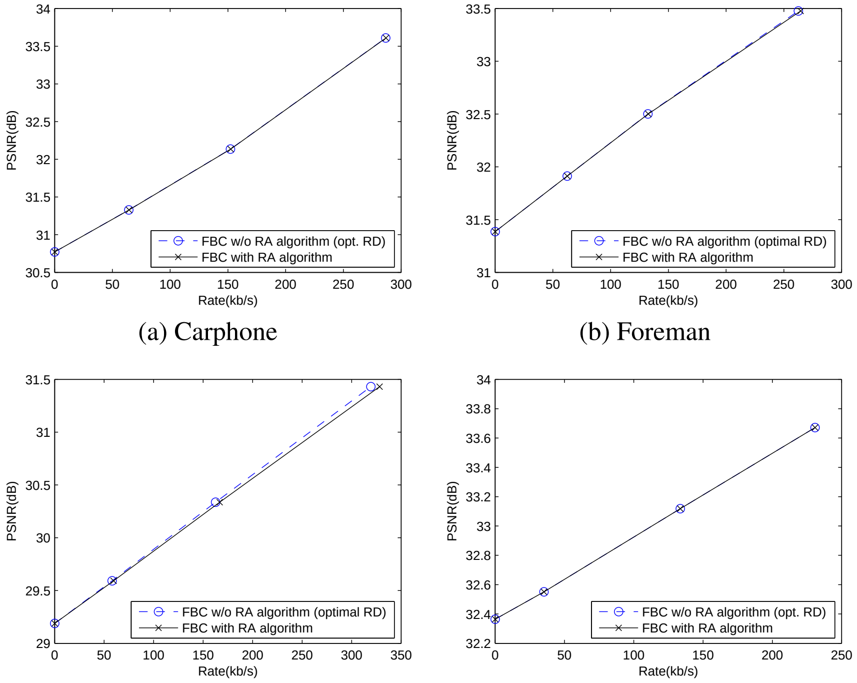
3.5.3 Experimental results and discussion

In this section, we first experimentally study the rate-distortion performance of a pixel-domain Wyner-Ziv (PDWZ) video coder with feedback channel that allocates bits with our rate allocation (RA) algorithm (RA-PDWZ video coder). We compare its performance to the same pixel-domain Wyner-Ziv video coder with feedback channel that does not use our rate allocation algorithm. In the latter case, the rate is allocated through bit requests over the feedback channel without a priori estimation of the rate at the encoder. Hence, with this coder an optimal rate-distortion performance is achieved, but the number of bit requests over the feedback channel is much larger. We will call this coder the noRA-PDWZ video coder, i.e. the optimal rate-distortion coder without rate allocation algorithm at the encoder. The advantage of our RA-PDWZ video coder compared to the noRA-PDWZ video coder is that it has a smaller latency and decoder complexity since the number of bit requests over the feedback channel is diminished. Nevertheless, we can achieve a very-near-to-optimal rate-distortion performance with our RA-PDWZ video coder, as will be discussed in this section. The latency and decoder complexity of the RA-PDWZ video coder and the noRA-PDWZ video coder is discussed after the rate-distortion performance. In particular, we compare the number of bit requests from the decoder over the feedback channel of the RA-PDWZ video coder with the number of bit requests of the noRA-PDWZ video coder.

The pixel-domain Wyner-Ziv video coder used in the experiments, first decomposes each Wyner-Ziv frame into its 8 bit planes. Then, the m most significant bit planes are separately encoded by using a rate-compatible punctured turbo code; the other bit planes are discarded. In our experiments, m is chosen to be $\in \{0, \ldots, 3\}$. The turbo coder is composed of two identical constituent convolutional encoders of rate 1/2 with generator polynomials $(1, 33/31)$ in octal form. The puncturing period was set to 32 which allowed our rate allocation algorithm to allocate parity bit multiples of $N/32$ bits to each bit plane, where N is the number of pixels in each frame. The key frames were intra-coded using H.263+ [International Telecommunication Union, 1998] with quantization parameter QP. We used the H.263+ software implementation of the University of British Columbia (UBC) (Version 3). The interpolated frames were generated at the decoder with the interpolation tools described in [Ascenso et al., 2005a]. The threshold t for Q_k was set to 10^{-3}.

To assess the efficiency of our rate allocation algorithm, we encoded several test sequences (QCIF, 30 frames/s) with the described RA-PDWZ video coder. For

Figure 3.9 Rate-distortion performance of our RA-PDWZ video coder for the sequences (a) *Carphone*, (b) *Foreman*, (c) *Mobile* and (d) *Salesman*. Shown is also the rate-distortion performance for the case of a noRA-PDWZ video coder (optimal rate-distortion performance, but without latency and decoder complexity reduction). The key frames are intra-coded with H.263+ ($QP = 10$).



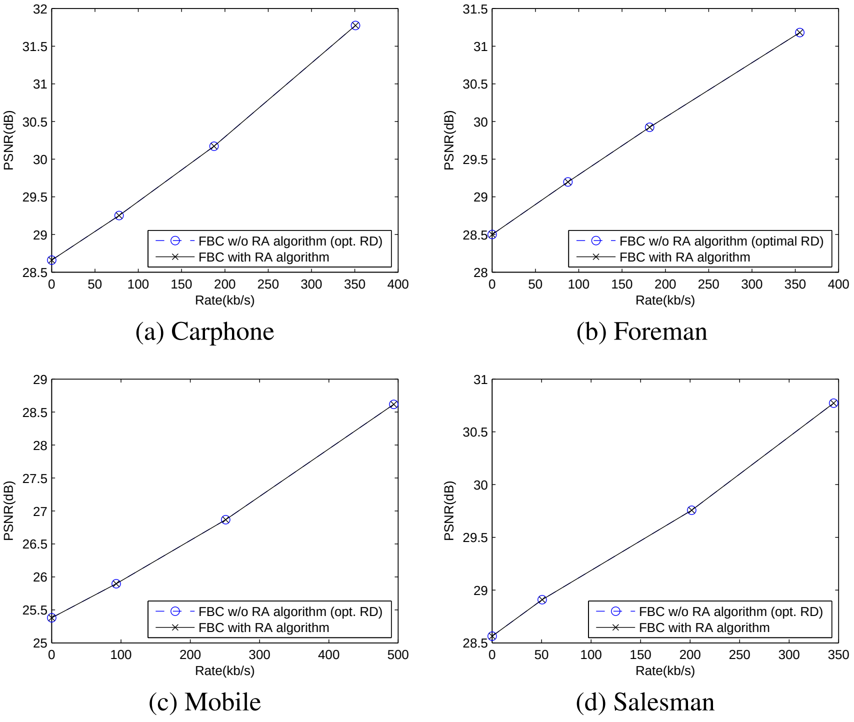
(a) Carphone

(b) Foreman

(c) Mobile

(d) Salesman

Figure 3.10 Rate-distortion performance of our RA-PDWZ video coder for the sequences (a) *Carphone*, (b) *Foreman*, (c) *Mobile* and (d) *Salesman*. Shown is also the rate-distortion performance for the case of a noRA-PDWZ video coder (optimal rate-distortion performance, but without latency and decoder complexity reduction). The key frames are intra-coded with H.263+ ($QP = 20$).



(a) Carphone

(b) Foreman

(c) Mobile

(d) Salesman

Table 3.8 Comparison of the average number of bit requests for the encoding of the k^{th} bit plane ($k = 1 \ldots 3$) between a pixel-domain Wyner-Ziv video coder *with* ($W_{k,\mathrm{RA}}$) and *without* ($W_{k,\mathrm{opt}}$) our rate allocation algorithm. f is the bit request reduction ratio (see (3.39)). The key frames are intra-coded with H.263+ ($QP = 10$).

Video sequence	$W_{k,\mathrm{opt}}$			$W_{k,\mathrm{RA}}$			f
	BP 1	BP 2	BP 3	BP 1	BP 2	BP 3	
Akiyo	1.49	3.00	7.00	0.49	1.04	4.03	1.69
Carphone	4.41	6.41	10.32	1.65	2.28	3.73	2.26
Coast	2.70	4.72	10.67	0.28	1.17	2.72	2.94
Container	8.07	1.14	6.31	3.76	0.14	3.47	1.79
Foreman	4.25	4.88	9.97	1.25	1.23	1.70	3.08
Hall	3.25	2.81	6.40	0.75	0.93	2.97	2.02
Mobile	3.89	7.80	12.20	0.02	0.16	0.17	8.04
Mother&D.	4.41	2.84	7.39	1.08	1.51	4.32	1.78
Salesman	1.97	7.28	7.19	0.92	4.49	3.82	1.59
Tennis	12.20	2.64	6.07	3.13	1.05	1.97	2.61

Table 3.9 Comparison of the average number of bit requests for the encoding of the k^{th} bit plane ($k = 1 \ldots 3$) between a pixel-domain Wyner-Ziv video coder *with* ($W_{k,\text{RA}}$) and *without* ($W_{k,\text{opt}}$) our rate allocation algorithm. f is the bit request reduction ratio (see (3.39)). The key frames are intra-coded with H.263+ ($QP = 20$).

Video sequence	$W_{k,\text{opt}}$			$W_{k,\text{RA}}$			f
	BP 1	BP 2	BP 3	BP 1	BP 2	BP 3	
Akiyo	3.00	5.99	10.27	2.00	3.86	7.11	1.39
Carphone	5.55	8.23	12.74	2.65	3.90	5.85	1.92
Coast	4.34	7.57	15.68	1.61	3.57	7.16	1.99
Container	11.52	2.63	10.72	6.91	1.63	7.82	1.44
Foreman	6.36	6.94	13.60	3.28	3.07	4.83	2.11
Hall	5.95	4.64	10.57	3.31	2.71	6.94	1.51
Mobile	6.84	12.30	19.42	2.09	2.89	4.75	3.27
Mother&D.	7.13	4.52	10.82	3.73	3.18	7.77	1.44
Salesman	3.26	11.72	11.06	2.22	8.84	7.56	1.34
Tennis	14.64	3.94	9.10	5.18	2.30	4.70	2.02

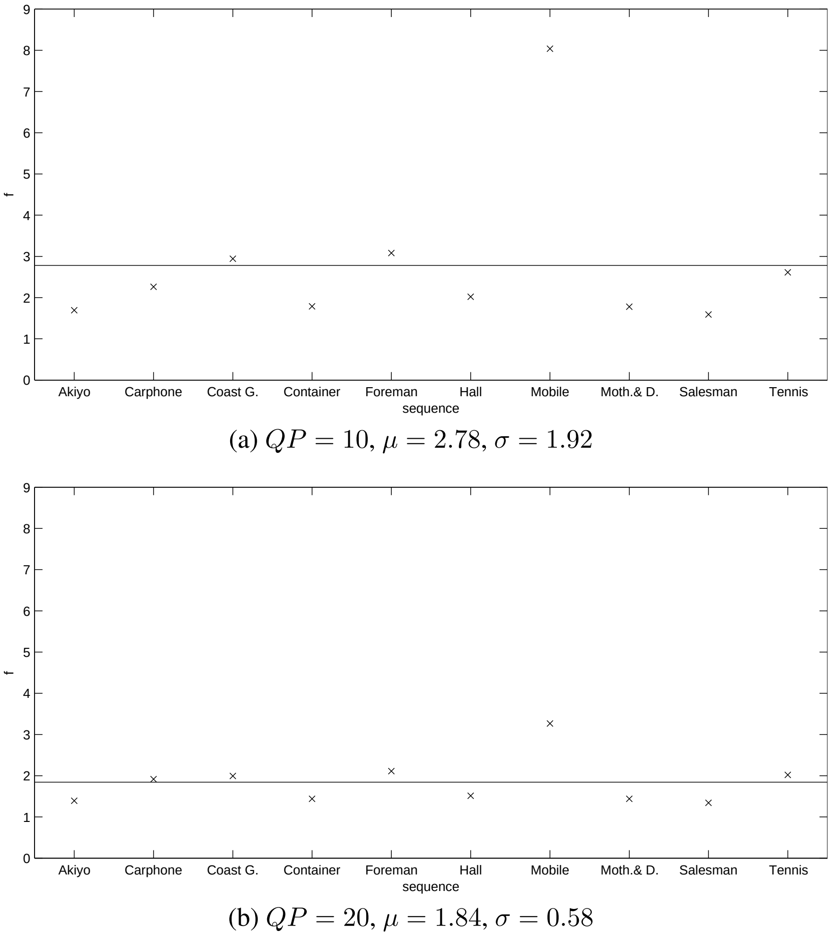
the plots, we only include the rate and distortion of the luminance of the Wyner-Ziv frames. The Wyner-Ziv frame rate is 15 frames/s. In Figures 3.9 and 3.10, we show the rate-distortion curves of *Carphone*, *Foreman*, *Mobile*, and *Salesman* when coded with the RA-PDWZ video coder, and we compare them with the corresponding rate-distortion curves when, for the given puncturing period, an optimal rate is allocated. This comparison is done for two different QP-values for the encoding of the key frames: $QP = 10$ (Figure 3.9) and $QP = 20$ (Figure 3.10). The value of the PSNR at rate 0 ($m = 0$) shows the average quality of the interpolated frame $\mathbf{S}$, as generated by the interpolation method of [Ascenso et al., 2005a]. Indeed, since $m = 0$ no Wyner-Ziv bit planes are sent, and the decoded frame is equal to the side information. The rate-distortion points at higher rates correspond to an increasing number of bit planes sent, more specifically, $m = 1, 2$ and 3. We observe that for all the sequences the rate-distortion performance of our RA-PDWZ video coder is very close to the optimal one, as obtained by the noRA-PDWZ video coder without latency and decoder complexity reduction.

Tables 3.8 and 3.9 show for ten test video sequences the average number of bit requests needed to decode the k^{th} bit plane ($k = 1, \ldots, 3$) for the noRA-PDWZ video coder (denoted $W_{k,\text{opt}}$, coder with optimal rate-distortion performance but without latency and decoder complexity reduction algorithm) and for our RA-PDWZ video coder (denoted $W_{k,\text{RA}}$). In Table 3.8 the key frames are intra-coded with $QP = 10$, while in Table 3.9 the key frames are intra-coded with $QP = 20$. We observe that with our rate allocation algorithm the number of bit requests is reduced significantly. Tables 3.8 and 3.9 also show for each sequence the average bit request reduction ratio f, with

$$f = \frac{\sum_{k=1}^{m}(W_{k,\text{opt}} + 1)}{\sum_{k=1}^{m}(W_{k,\text{RA}} + 1)}. \tag{3.39}$$

In Figure 3.11, we plot for the 10 test sequences (and for $QP = 10$ and $QP = 20$) these average bit request reduction ratios f. In particular, we show how for each sequence this average bit request reduction ratio compares to the mean value μ over all the sequences. We also indicated the standard deviation σ. As can be expected, we observe for both $QP = 10$ and $QP = 20$ higher bit request reduction ratios (around and above μ) for sequences that need a higher number of parity bit sets, i.e. when the amount of correlation noise (difference between the pixel values of the original frame $\mathbf{X}$ and the corresponding pixel values of the side information $\mathbf{S}$, see Section 3.2.2) is larger. This is mostly the case for sequences that contain a lot of motion and camera movement (e.g. *Carphone*, *Coast*, *Foreman*, *Mobile*, *Tennis*). Indeed, in these sequences the motion compensated interpolation between the two adjacent key frames is more difficult and yields a worse estimate of the frame to be encoded than in the case of sequences with less motion and recorded

Figure 3.11 Average bit request reduction ratio f for the 10 test sequences. The key frames are intra-coded using H.263+ with (a) $QP = 10$ and (b) $QP = 20$. The mean value over all the sequences is denoted by μ and is indicated with the solid horizontal line. The standard deviation is denoted by σ.



(a) $QP = 10$, $\mu = 2.78$, $\sigma = 1.92$

(b) $QP = 20$, $\mu = 1.84$, $\sigma = 0.58$

with a static camera (e.g. *Akiyo*, *Container*, *Hall*, *Mother&Daughter*, *Salesman*).

Moreover, we observe that, even though the number of parity bits sets is generally higher for $QP = 20$ than for $QP = 10$, lower bit request reduction ratios are achieved for $QP = 20$ than for $QP = 10$. This is due to the fact that the rate is more significantly underestimated for $QP = 20$ than for $QP = 10$. The reason for this is as follows. As explained in Section 3.5.2, the rate allocation is based on two estimates of the variance, one made at the encoder (Section 3.5.2.1), and one made at the decoder (Section 3.5.2.1). For the encoder estimate the intra-coding noise is taken into account (the original frame intervenes in the estimate), while in the decoder estimate this error is not incorporated (the subtracted frames are both coded). Since for most of the frames the decoder estimate is the final estimate, the rate allocation is for many frames done without taking into account the distortion introduced by the intra-coder, which results in an underestimation of the rate. If the rate is underestimated, additional parity bit sets are requested by the decoder over the feedback channel to reach the optimal rate, or in other words, the number of bit requests is increased. The impact of this effect is obviously more significant for a higher value of the quantization parameter QP, and therefore lower bit request reduction ratios are achieved for $QP = 20$ than for $QP = 10$. In order to refine the rate allocation for higher values of QP, the influence of the intra-coding noise should be incorporated in the decoder estimate of the variance. This is a matter for further investigation. Note, however, that in any case a certain amount of underestimation of the rate should be maintained, since this assures an optimal rate-distortion performance (as explained in Section 3.5.2).

Nevertheless, in general we observe that significant bit request reduction ratios f are achieved. According to (3.28) and (3.29), the decoder complexity and the latency decrease by the same ratio f. The reduction of the latency is especially crucial when putting the discussed Wyner-Ziv video coding scheme into practice, since then the large delays of the feedback channel approach without rate allocation are unacceptable. Note that as a counterpart our rate allocation algorithm has made the encoder a bit more complex. More specifically, the main factors that influence the encoder burden are: the decoding of the key frames, the estimation of the correlation noise variance and the estimation of the bit rate for each bit plane. Nevertheless, the total computational load of the encoder is still small in comparison with the complexity of conventional encoders.

3.6 Conclusion

In contrast to conventional video coding, Wyner-Ziv video coders perform simple intra-frame encoding and complex inter-frame decoding. This feature makes this type of coding suitable for applications that require low-complexity encoders. In this chapter, we studied and improved a scalable turbo-code based pixel-domain distributed video coder. Our contributions can be summarized as follows:

First, we presented a model for the distortion introduced in pixel-domain Wyner-Ziv video coders. The model can be used to help coders of this type to fulfill coding constraints and improve their efficiency. As an application example of our model, we used it to design an algorithm to select the quantization parameter of each frame in distortion-constrained encodings. Despite the restricted capability of Wyner-Ziv video encoders to accurately estimate the model parameter, experimental results show that our model allows us to approach the target distortion. In our algorithm, the rate of Wyner-Ziv frames was freely chosen to fulfill distortion constraints.

Second, we proposed a rate allocation algorithm for rate-compatible, turbo code-based pixel-domain distributed video coders. Without complicating the encoder, the algorithm estimates the appropriate number of bits for each frame. The proposed pixel-based rate allocation algorithm delivers more accurate estimates of the encoding rate than the frame-based approach. This pixel-based rate allocation algorithm allows to remove the feedback channel from the traditional scheme, with only a small loss in rate-distortion performance, especially for low rates.

Third, we described a method to reduce the computational decoder complexity and latency of rate-compatible, turbo code-based pixel-domain Wyner-Ziv video coders with feedback channel. The algorithm estimates the appropriate number of bits for each frame without complicating the encoder. By sending in a first step this estimated number of parity bits, the number of bit requests from the decoder over the feedback channel (in order to reach the final number of parity bits needed to error-freely decode the frame) is considerably reduced compared to the case where no rate estimation algorithm is used. Experimental results on several test sequences show that the decoder complexity and the latency are diminished by a significant factor, while a very near-to-optimal rate-distortion performance is preserved.

The research work on distributed video coding was published as one journal article [Morbee et al., 2008a], one chapter in Lecture Notes on Computer Science [Morbee et al., 2007a], and several international conference publications [Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a].

4
Vision Systems for 2D Occupancy Sensing

4.1 Introduction

A 2D occupancy map provides an abstract top view of a scene containing people or objects. Such maps are important in many applications such as surveillance, smart rooms, video conferencing and sport games analysis. The 2D occupancy sensing systems that are proposed in research are based on a single-camera setup, a multi-camera setup [Delannay et al., 2009, Fleuret et al., 2008, Alahi et al., 2009], pressure sensitive carpets [Clos et al., 2004, Federspiel and Michael, 2005], passive infrared (PIR) sensors [Elwell, 2009, Zhevelev et al., 2009], or active radar/ultrasound/radio beacons [Bahl and Padmanabhan, 2000, McCarthy and Muller, 2005] (or a combination of the previous, e.g. infrared and ultrasonic [Fowler, 2000, Myron et al., 1997, Elwell, 2009]).

Single-camera systems often provide poor occupancy results. For example, if the camera is mounted such that the scene is observed from the side, the occupancy results lack accuracy in the viewing direction of the camera. Moreover, occlusion often makes it impossible to accurately detect target occupancy. If the camera is mounted overhead, occlusion is not a problem in the center of the captured view, but remains problematic near the borders of the captured images. Furthermore, objects that do not appear at the center of the image but at the borders, are not captured from right above. Hence, instead of capturing a top-view of the object, a smeared-out top-view is grabbed. This deformation deteriorates occupancy detection results.

Compared to single-camera systems, camera networks, offer an attractive non-intrusive and flexible tool for occupancy mapping. In recent years, foreground silhouettes in multiple camera views have been increasingly used to estimate the probability of ground occupancy, because the silhouettes contain the most important information needed for occupancy sensing. Moreover, these silhouette-based approaches are popular because of their simplicity and computational efficiency. Two basic approaches exist. Bottom-up methods transfer the foreground silhouettes from the different camera images to a common reference plane using camera image-floor homographies [Delannay et al., 2009]. Top-down approaches extract occupied ground positions by comparing a generative model of the objects in the scene with the actual foreground silhouettes observed in the camera views [Fleuret et al., 2008, Alahi et al., 2009]. Until now, for both approaches the mathematical laws for the fusion of data from different cameras have not been considered explicitly. In this chapter, more specifically in Section 4.2, we will describe a novel method for calculating occupancy maps with multiple cameras. In particular, we focus on this data fusion aspect within a bottom-up method and show that Dempster-Shafer based fusion of camera information leads to significantly more accurate occupancy maps.

The position measurements from these multi-camera systems are more accurate than can be achieved with active radio, ultrasound and radar technologies. Moreover, these active sensor systems need extensive noise cancellation techniques and heavy processing because of the use of direction of arrival methods [Schiele and Crowley, 1994]. Camera based systems, however, suffer from other disadvantages. A first concern is the possibility of privacy breach. This is halting the deployment of camera networks into enterprises, shopping malls, streets, elderly homes, private houses, etc. Additionally, multi-camera setups, as well as pressure sensitive carpets, require expensive alterations to the infrastructure and wiring for power and data lines. Regular cameras are expensive because of the high-complexity processing involved in analyzing real-time video signals. Also, their power consumption might make battery operation difficult. This makes hook-up to grid power or power over Ethernet desirable, which requires changes to existing infrastructure.

To overcome these shortcomings of a multi-camera based setup, we propose in Section 4.3 the use of multiple light-integrating line sensors for 2D occupancy sensing. Light-integrating line sensors do not record photographic images from the scene they observe, which rules out privacy problems. They are cheap (around 4 euros) and consume very little power (about a factor 100 less power-consuming than a regular camera sensor). The cheap and low-power nature of line sensors allows to use many of them in one setup, which gives the possibility of achieving higher accuracies than with a small number of sensors. Through this, they can outperform other (more expensive or more power-consuming) types of sensors for the same cost or power budget. The low-power property also allows the line sensor to be battery operated. This removes the need for wiring, which makes a setup with multiple line sensors easy to construct and which minimizes its aesthetic impact. Moreover, a line array is made up of a limited number of sensing elements, making data processing much less computationally expensive. Because individual sensing elements of line sensors are large, they are very light sensitive, and because of their high bit depth, their output is accurate.

The work presented in this chapter has been performed in collaboration with my colleague Linda Tessens and therefore the subject of this chapter is related to some concepts from her PhD thesis [Tessens, 2010]. However, in her work, the focus was mainly on the various fusion techniques that can be used in multi-camera systems to combine the single-view maps. One of these techniques will be used and presented in this work, but our emphasis will lie on the study of the usage of different sensor types (cameras, line sensors) and different data output types from these sensors (full images, scan lines from full images, scan lines from light-integrating line sensors). We make an overall comparison between the different systems in terms of the obtained occupancy map quality, the memory and computational requirements, the price of the system, its power consumption and its privacy-friendliness.

The remainder of this chapter is organized as follows. In Section 4.2, we present a novel method for calculating occupancy maps with a set of calibrated and synchronized cameras. In particular, we propose Dempster-Shafer based fusion of the ground occupancies computed from each view. Then, in Section 4.3, we describe a system and method for 2D occupancy sensing with light-integrating line sensors. Finally, in Section 4.4, we compare the performance of the proposed methods with the state-of-the-art occupancy calculation methods. We will show that our Dempster-Shafer based multi-view occupancy map method yields significantly more accurate occupancy maps than the other methods from literature based on data from multiple cameras. For the basket ball dataset of [De Vleeschouwer and Delannay, 2009], the total mass of occupancy evidence (or probability) as obtained with our methods is up to 8 times more concentrated around the ground truth player positions than for the methods of [Delannay et al., 2009] and [Fleuret et al., 2008]. The method based on data from light-integrating line sensors has the advantages as described above (and also further in this chapter in Table 4.1). These advantages come, however, at the expense of a small performance loss in terms of occupancy map accuracy compared to some of the state-of-the-art multi-camera based methods. This will be discussed in detail in Section 4.4.

4.2 Multi-view occupancy mapping

In this section, we present a novel method for calculating discrete occupancy maps with a set of calibrated and synchronized cameras. The discrete occupancy mapping problem can be formulated as follows. Let the ground plane of the observed scene be discretized in resolution cells $\mathbf{x}$. The goal of occupancy sensing is to assign to each cell a value which expresses the probability that the cell is occupied by a *foreground* object. Foreground objects are objects of interest. In typical applications such objects are persons, cars, luggage, etc. The parts that are not of interest make up the *background*.

To calculate the occupancy map, we first calculate for each view a ground occupancy from the foreground silhouette by image-floor homography mapping. Then, we fuse these ground occupancies from different views. The novelty of our method is that we propose a particular type of fusion, namely Dempster-Shafer based fusion of these ground occupancies computed from each view. This method allows to achieve higher occupancy accuracies than the state-of-the-art.

In Section 4.2.1, we start with an overview of the related work. Then, in Section 4.2.2, we introduce the basics of the Dempster-Shafer theory of evidence. Finally, in Section 4.2.3, we describe our Dempster-Shafer based approach for occupancy map calculation.

4.2.1 Related work

Recent multi-view camera systems for occupancy sensing often make use of foreground silhouettes from the multiple views to obtain an accurate 2D occupancy map. In the probabilistic occupancy map (POM) method of [Fleuret et al., 2008], a top-down approach is followed. For each view the conditional probability distribution of the observed background subtraction image given the true object positions is a function of a distance measure between the background subtraction image and the image obtained from a generative model. Information from different views is fused by multiplying these conditional probability distributions.

In the bottom-up method from [Delannay et al., 2009], each camera produces a confidence value for the occupancy of each ground position by back-projecting the foreground silhouettes to a common reference plane using camera image-floor homographies. The aggregated ground occupancy map is obtained by summing the camera confidences and by normalizing by the number of cameras that actually view a particular ground position.

In this section, unlike the summing [Delannay et al., 2009] and POM [Fleuret et al., 2008] fusion strategy, we use Dempster-Shafer (DS) based fusion to exploit the fact that if a hypothesis of (non-)occupancy is corroborated by different cameras, a higher belief should be assigned to it. Moreover, the DS theory of evidence allows distinguishing between equal probability of occupancy and non-occupancy, and lack of knowledge, e.g. when an object is outside a camera's viewing range. More specifically, in our method the cameras are considered independent sources of information of which the data about the (non-)occupancy of ground positions can be opportunistically fused using the DS rule of combination [Dempster, 1968]. In Section 4.2.2, we first briefly introduce the Dempster-Shafer theory of evidence.

4.2.2 Dempster-Shafer theory of evidence

The DS theory of evidence provides a theoretical basis to combine evidence from different sources to arrive at a degree of belief in a number of propositions. Formally, an exhaustive set of mutually exclusive propositions constitutes a frame of discernment Ω. The subsets A of Ω are called propositions, the singleton subsets ω of Ω are elementary propositions and the power set, denoted as 2^Ω, is the set of all possible subsets A of Ω. A basic belief assignment (BBA) is a mapping m from 2^Ω to $[0, 1] \subset \mathbb{R}$ such that $\sum_{A \subseteq \Omega} m(A) = 1$ and $m(\emptyset) = 0$. $m(A)$ expresses how much an agent believes in proposition A alone, with no further assumption about any proper subset E of A ($E \subset A$). A particular instance of a BBA is called a body of evidence. The basic probability allotted to Ω is a measure of the belief that has not been assigned to any of the proper subsets of Ω. It can be interpreted as the remaining uncertainty about the propositions. Complete ignorance is represented by $m(\Omega) = 1$. Note that such a measure is absent when evidence for a proposition is gathered in a purely probabilistic manner: when a proposition is true with probability P, a probability of $1 - P$ must be assigned to its negation.

Assume two pieces of evidence give rise to two bodies of evidence m_1 and m_2. These provide different assessments for the propositions in the same frame of discernment. To aggregate the information from these two sources, we need a rule of combination. The best known and most common combination rule is Dempster's rule of combination:

$$m_1 \oplus m_2(C) = \begin{cases} \displaystyle\sum_{A,B|A\cap B=C} \frac{m_1(A)m_2(B)}{1-K} & \text{if } C \neq \emptyset \\ 0 & \text{if } C = \emptyset. \end{cases} \tag{4.1}$$

where $C \subseteq \Omega$ and K is the amount of conflict between the two bodies of evidence, measured by

$$K = \sum_{A,B|A\cap B=\emptyset} m_1(A)m_2(B). \tag{4.2}$$

The denominator in Eq. 4.1 is a normalizing factor. This rule leads to a specialization of the basic belief: each time a new piece of information is accepted, the basic belief assigned to a proposition A is distributed over the subsets of A [Denoeux, 2008].

Dempster's rule assumes that m_1 and m_2 are distinct, i.e. that the sources that produced the evidence are uncorrelated. In [Denoeux, 2008] a cautious conjunctive rule is proposed to combine bodies of evidence that are not distinct. We refer the reader to [Denoeux, 2008] for a formal definition of this rule. It is derived from the principle of least commitment: of all bodies of evidence that could result from the combination of the inputs m_1 and m_2, the least informative one is chosen. Note that as a consequence, if the bodies of evidence *are* distinct and they are combined using the cautious rule, the result will be less informative than if Dempster's rule is used. In the next section, we will go into details about how we use the Dempster-Shafer theory of evidence in the calculation of occupancy maps.

4.2.3 Evidential multi-view occupancy maps

Consider a network of N cameras. To obtain a discrete occupancy map of the scene, the ground plane of the observed scene is divided into resolution cells $\mathbf{x}$. We wish to assign a real value to each cell that expresses our confidence that the cell is occupied by an object of interest. In typical applications such objects are persons, vehicles, etc. The discretization resolution is chosen such that the area covered by one cell is (typically a lot) smaller than the average area occupied by a person or another object of interest. Why this choice is preferable, will be explained later in this section when we discuss the practical implementation of our algorithm.

As explained in Section 4.2.2, in the DS theory of evidence a basic belief assignment or BBA m is a mapping that assigns to each subset A of a frame of discernment θ a belief $m(A) \in [0, 1]$. The basic belief assigned to a hypothesis expresses how much evidence supports it. In our method, for each cell $\mathbf{x}$ the mutually exclusive and exhaustive hypotheses that $\mathbf{x}$ is either occupied ($\{occ_\mathbf{x}\}$) or not ($\{nocc_\mathbf{x}\}$) constitute the frame of discernment $\theta_\mathbf{x} = \{occ_\mathbf{x}, nocc_\mathbf{x}\}$ [Dempster, 1968]. The information from each view n, $1 \leq n \leq N$, is considered a distinct piece of evidence and we denote the BBA representing this evidence by m_n. We now explain how we define the BBA in our method.

Let H be the height of a typical person and consider a rectangular cuboid with cell $\mathbf{x}$ as base and height H. If this cuboid lies completely outside the viewing frustum of camera n, this camera cannot provide any information about the occupancy

Figure 4.1 An example of a region $R^n_{\mathbf{x}}$ with $H = 2m$ is marked with a white line on the player of the dark team at the front right in the image. The 3D box with as ground plane the cell $\mathbf{x}$ ($\mathbf{x}$ has an area of $(0.02m)^2$) is so thin, that it is reprojected onto a line-shaped region $R^n_{\mathbf{x}}$.

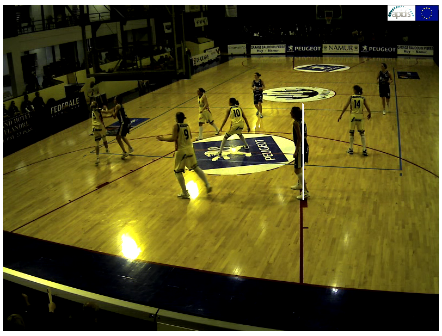
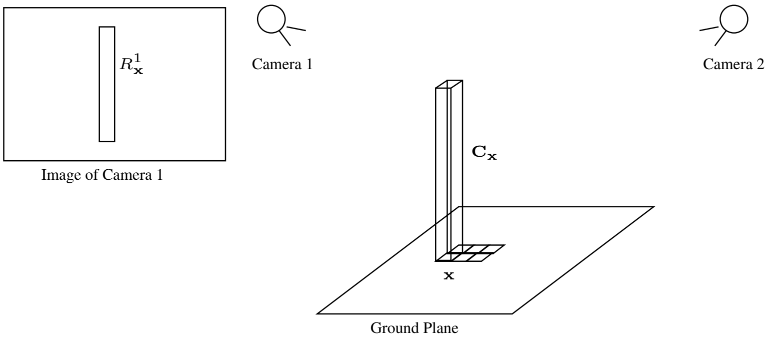


Figure 4.2 The projection of a rectangular cuboid $\mathbf{C}_{\mathbf{x}}$ with height H and cell $\mathbf{x}$ as base into camera view 1 defines an image region $R^1_{\mathbf{x}}$.



of $\mathbf{x}$. The BBA is then $m_n(\{occ_{\mathbf{x}}\}) = 0$, $m_n(\{nocc_{\mathbf{x}}\}) = 0$ and $m_n(\theta_{\mathbf{x}}) = 1$.

Otherwise, the projection of this cuboid into camera view n defines an image region $R^n_{\mathbf{x}}$. An example of such a region is marked by the white line in Fig. 4.1. Fig. 4.2 illustrates the introduced notations. Note that for a ground plane grid with high resolution (or in other words, for cells $\mathbf{x}$ with a small area) the 3D box with as ground plane the cell $\mathbf{x}$ is reprojected onto a vertical area in the images that has a width smaller than the width of a pixel, such that $R^n_{\mathbf{x}}$ is a line-shaped region. This is the case in Fig. 4.1. This aspect will return later this section, when we explain the equivalent procedure for the calculation of $m_n(\{nocc_{\mathbf{x}}\})$.

We gather evidence about the (non-)occupancy of the cells by independently segmenting each view into background and foreground. To this end, we use an algorithm based on mixture of Gaussians that yields binary foreground (FG) silhouette images F_n for each view n [Stauffer and Grimson, 2000]. This binary FG silhouette assumes value 1 for a foreground pixel and value 0 for a background pixel. Then, we determine in each region $R^n_{\mathbf{x}}$ the fraction of background pixels $b^n_{\mathbf{x}}$ and of foreground pixels $f^n_{\mathbf{x}}$. Of course $b^n_{\mathbf{x}} + f^n_{\mathbf{x}} = 1$.

The evidence $m_n(\{nocc_{\mathbf{x}}\})$ of camera n for the hypothesis $\{nocc_{\mathbf{x}}\}$ is

$$m_n(\{nocc_{\mathbf{x}}\}) = b^n_{\mathbf{x}}. \tag{4.3}$$

Later this section, we will describe the calculation of the evidence $m_n(\{occ_{\mathbf{x}}\})$, which will be more complicated than the calculation of the evidence $m_n(\{nocc_{\mathbf{x}}\})$.

Before that, we would like to describe first an equivalent procedure to obtain $m_n(\{nocc_{\mathbf{x}}\})$. This procedure helps to clarify the method and shows how we implement this method in practice. This equivalent procedure is related to the method followed by [Delannay et al., 2009]. First, as in the previous procedure we perform change detection independently for each single-view n, $1 \leq n \leq N$ which yields the binary foreground/background masks F_n. These FG silhouette images F_n are mapped through a combination of homographies to the transformed view $\check{F}_n$ such that

1. verticality is preserved when projecting from 3D space to $\check{F}_n$ (*verticality property*), and

2. the ratio of heights between objects in 3D space and their projections in $\check{F}_n$ remains the same when the feet of those objects are projected on the same horizontal line in $\check{F}_n$ (*ratio of heights property*).[1]

This transformation allows to simplify the computation of $m_n(\{nocc_{\mathbf{x}}\})$ associated with view n. In particular, because of the verticality property a line perpendicular to the ground plane in 3D space will be reprojected onto a vertical line in $\check{F}_n$. Moreover, if we assume that the size of the resolution cells is such that a rectangular cuboid with cell $\mathbf{x}$ as base is reprojected onto a rectangular area (which will be vertical, due to the verticality property) with width smaller than the width of a pixel[2] (as illustrated by Figure 4.1), and since in the image we cannot sample denser than the width of a pixel, integration along a rectangular cuboid with cell $\mathbf{x}$ as base can be approximated by integrating along a vertical line in the BG/FG image.

Let us now consider the height H of an average person in 3D space. An upright 3D box of this height H is projected onto a segment of length h (a vertical line of h pixels) in $\check{F}_n$. An example of such a segment is shown in Figure 4.1. This segment

[1] For top views, the principal axis is set perpendicular to the ground and a polar mapping is performed to achieve the same properties [Delannay et al., 2009].

[2] This will be the case when the discretization resolution is chosen such that the area covered by one cell is a lot smaller than the average area occupied by a person or object of interest.

will be *vertical* in $\check{F}_n$ due to the verticality property described above. Depending on where the 3D box is positioned in the scene, the length h will be different. For nearby objects the length h will be larger, for objects further away from the camera, the length h will be smaller. However, due to the ratio of heights property described above, the length of this segment h is equal along a horizontal line in the image $\check{F}_n$. Hence, h will be only dependent on y, i.e. $h(y)$. For upright objects of size H of which the feet are projected on a vertical line at the upper part of $\check{F}_n$, their length in the image $\check{F}_n$ will be smaller than for objects of size H of which the feet are projected on a vertical line at the lower part of $\check{F}_n$.

Hence, since

- the integration along a rectangular cuboid with cell $\mathbf{x}$ as base can be approximated by integrating along a vertical line in the BG/FG image, and
- since the height H of an average person in 3D space corresponds to a segment of length $h(y)$ in $\check{F}_n$,

we can integrate occupancies along the height H in 3D space by calculating an image for which the value at any point (x, y) is the (normalized) sum of all the pixels on and above (up to the height $h(y)$) the position (x, y) in the image $\check{F}_n$. We will denote this image by $\dot{F}_n$. More specifically,

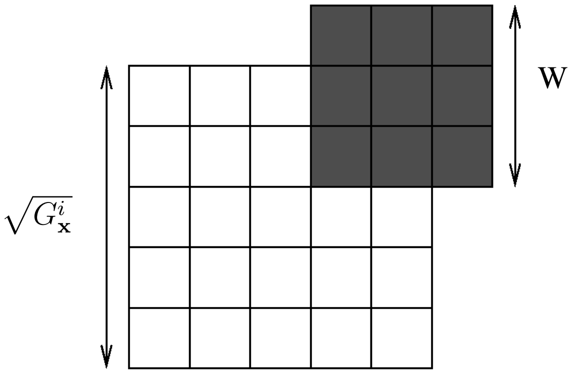
$$\dot{F}_n(x, y) = \sum_{y \leq y' \leq y+h(y)} \frac{\check{F}_n(x, y')}{h(y)}. \tag{4.4}$$

$\dot{F}_n(x, y)$ is the integral image of the binary image $\check{F}_n$ along a vertical segment $h(y)$. Calculating this integral image is the equivalent of integrating occupancies along the height H in 3D space. Note that, in Eq. 4.4, the sum over the segment with length $h(y)$ to obtain a pixel value of the integral image, is normalized with the segment length $h(y)$. This is done in order not to favor nearby objects. This way, $\dot{F}_n$ is a floating point image with values between 0 and 1. Finally, $m_n(\{nocc_{\mathbf{x}}\})$ is obtained by re-projecting the complement $\bar{F}_n$ of the integral image $\dot{F}_n$ to the ground plane by homography warping [Delannay et al., 2009].

For $m_n(\{occ_{\mathbf{x}}\})$ the situation is more complicated. Because of the limited resolution of the cameras, different cells $\mathbf{x}$ and $\mathbf{x}'$ may give rise to completely coinciding regions $R^n_{\mathbf{x}}$ and $R^n_{\mathbf{x}'}$. Let $G^n_{\mathbf{x}}$ be the number of cells sharing the same region $R^n_{\mathbf{x}}$ as the cell $\mathbf{x}$. If $G^n_{\mathbf{x}} > 1$, the evidence of occupancy collected in $R^n_{\mathbf{x}}$ may be attributable to a person occupying only part of the cells with coinciding $R^n_{\mathbf{x}}$. Because of the reprojection geometry, these $G^n_{\mathbf{x}}$ positions will be approximately laid out in a trapezoid, which we approximate by a square $\mathbf{S}$ with side length $\sqrt{G^n_{\mathbf{x}}}$.

Assuming a person occupies a square of W^2 cells, this person can be in $(\sqrt{G^n_{\mathbf{x}}} + W - 1)^2$ different positions with respect to the square $\mathbf{S}$ (see Fig. 4.3).

Figure 4.3 Example of a square approximation $\mathbf{S}$ of $G^n_{\mathbf{x}} = 25$ resolution cells $\mathbf{x}$ for which the cuboid $\mathbf{C}_{\mathbf{x}}$ is projected onto the same region $R^n_{\mathbf{x}}$. A person, represented here by the gray square with $W^2 = 9$ resolution cells, can assume $(\sqrt{G^n_{\mathbf{x}}}+W-1)^2$ different positions such that it overlaps with $\mathbf{S}$. Hence, if $R^n_{\mathbf{x}}$ is completely part of the foreground, there is a probability of $W^2/(\sqrt{G^n_{\mathbf{x}}} + W - 1)^2$ that a particular cell is actually occupied by a foreground object.



A particular cell $\mathbf{x}$ in the square $\mathbf{S}$ is only occupied in W^2 of all these positions. Hence, the evidence of occupancy $m_n(\{occ_{\mathbf{x}}\})$ is scaled with

$$g^n_{\mathbf{x}} = W^2/(\sqrt{G^n_{\mathbf{x}}} + W - 1)^2 \tag{4.5}$$

and we define

$$m_n(\{occ_{\mathbf{x}}\}) = g^n_{\mathbf{x}} f^n_{\mathbf{x}}. \tag{4.6}$$

A similar reasoning holds for the equivalent practical procedure. There, $m_n(\{occ_{\mathbf{x}}\})$ is obtained by re-projecting the integral BG image $\dot{F}_n$ to the ground plane by homography warping and then scaling the value obtained in the position $\mathbf{x}$ by $g^n_{\mathbf{x}}$.

When $m_n(\{occ_{\mathbf{x}}\})$ and $m_n(\{nocc_{\mathbf{x}}\})$ are known, $m_n(\theta_{\mathbf{x}})$ can be calculated as

$$m_n(\theta_{\mathbf{x}}) = 1 - m_n(\{occ_{\mathbf{x}}\}) - m_n(\{nocc_{\mathbf{x}}\}). \tag{4.7}$$

The pieces of evidence collected by the N views about each cell $\mathbf{x}$ are fused using Dempster's rule of combination (see Eq. 4.1 in Section 4.2.2). The denominator in Eq. 4.1 is a normalizing factor. As mentioned in Section 4.2.2, Dempster's rule assumes that the sources that produced the evidence are uncorrelated. This is a valid assumption in a scenario in which the cameras are not mounted very closely to each other. This is the case for the experimental setups discussed in Section 4.4, and hence Dempster's rule can be applied in these scenarios. If this assumption is not valid, the cautious conjunctive rule [Denoeux, 2008] should be used instead, as discussed in Section 4.2.2.

This fusion process must be performed for each resolution cell in the occupancy map. We denote the fused evidence of occupancy for all occupancy map cells by $m(\{occ\})$.

It is interesting to note at this point that we could also use an approximated version of the foreground silhouettes of the cameras to calculate an occupancy map. To save memory and computations, one can calculate the FG/BG segmentation for a reduced version of an image, instead of for the full image. This BG/FG segmentation on a reduced image version is then used to approximate the BG/FG segmentation of the full image.

In particular, we observe that for the calculation of a *two-dimensional* occupancy map the observations along the *vertical direction* might be combined without loosing too much 2D occupancy map information. Therefore, we study the use of a horizontal scan line as a reduced version of the image. A horizontal scan line of an observed image $I(x, y)$ is its Radon transform along the vertical grid direction. This is equivalent to integrating the intensity function $I(x, y)$ over vertical lines to obtain the horizontal scan line, $H(x)$:

$$H(x) = \sum_{y=1}^{l} I(x, y) \tag{4.8}$$

where l is the image height.[3]

After calculating this horizontal scan line, we perform FG/BG segmentation on this horizontal scan line. We denote the background/foreground mask of H by F_H. How FG/BG segmentation on scan lines is implemented will be explained in Section 4.3.3. The foreground/background mask F of the original image can be approximated by $\hat{F}$, where $\hat{F}$ is obtained as

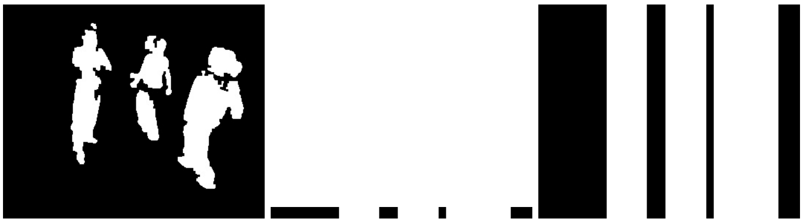
$$\hat{F}(x, y) = F_H(x). \tag{4.9}$$

In Figure 4.4, we show an example of FG/BG segmentation on a full image (Figure 4.4a), scan line FG/BG segmentation (Figure 4.4b) and the approximated full image FG/BG segmentation from scan line FG/BG segmentation (Figure 4.4c). This approximated FG/BG segmentation is obtained by column-wise extending the scan line pixels, as expressed by Eq. 4.9.

Based on these approximated FG/BG masks, we can - in the same way as described earlier in this section based on FG/BG segmentations on full images - calculate a Dempster-Shafer based occupancy map. The quality of the thus obtained occupancy map depends on the quality of the foreground/background segmenta-

[3] Note that Section 4.3 shows how this Radon transform is implemented optically using light-integrating line sensors.

Figure 4.4 (a) BG/FG segmentation on full images ($F(x, y)$), (b) scan line FG/BG segmentation ($F_H(x)$) and (c) approximated full image FG/BG segmentation from scan line FG/BG segmentation ($\hat{F}(x, y)$).



(a) (b) (c)

tion $\hat{F}$. The loss in occupancy map accuracy due to the FG/BG segmentation approximation will be studied in detail in Section 4.4.

Note that the quality of the FG/BG segmentation approximation itself, as compared to the FG/BG segmentation of the full image, is influenced by the camera set-up and it is better when the objects appear large in the camera image. These issues are discussed in Section 4.5 of the PhD thesis of Linda Tessens [Tessens, 2010], to which we refer the reader for more details on this subject.

4.3 Multiple light-integrating line sensors for occupancy sensing

In this section, we will describe the novel system and method for 2D occupancy sensing based on multiple light-integrating line sensors. First, we summarize briefly the differences between this system and other state-of-the-art 2D occupancy systems. Then, in Section 4.3.2, we give an overview of the system setup. Finally, in the Sections 4.3.3 and 4.3.4, we explain how the 2D occupancy maps are obtained based on the data coming from multiple light-integrating line sensors.

4.3.1 Related work

In most cases, occupancy systems provide input to other algorithms implementing the final application. This means that in order to be useful in an economical sense, their price should be far below the cost savings that their usage can bring in the main application. An example of such an application is a smart building system that switches lights on or off to save energy as a function of person position and activity. This low-cost requirement puts a heavy burden on the bill of materials of the complete sensor system. Moreover, apart from being cheap, it is desirable that a sensor based occupancy system is low-power to avoid hook-up to grid power or power over Ethernet, privacy-friendly and non-intrusive. For some applications, such as sport games analysis or surveillance, occupancy accuracy is also key to make the system successful in practice. Taking these requirements into consideration, we present a novel method for 2D occupancy sensing based on multiple light-integrating line sensors.

In Table 4.1, we schematically summarize the comparison between this novel system and other non-intrusive 2D occupancy sensing systems (including the system based on multiple cameras) that were discussed in Section 4.1.

4.3.2 System overview

In some vision applications observations along one direction can be reduced to a single observation because what one measures has a repetitive structure along this direction. An example is the measurement of the 2D occupancy of a room: to a certain degree it does not matter if this is measured at knee-height or at shoulder-height, as people mostly stand upright. Additionally, it is possible to combine several observations along this direction[4].How we will do this will be explained in detail further this section.

A central role in this research work is played by a device consisting of a linear array of optical sensing elements (e.g. photo diodes), as described in [Inada, 2006, Kawamoto and Narabu, 1999]. In the following, this device is denoted as line sensor. Figure 4.5 shows a schematic frontal view of a line sensor. A line sensor is composed of a number of sensing elements that are positioned in a 1D array. Each sensing element or pixel can sense light independently of the other sensing elements, like it is the case for the pixels of a 2D image sensor array. The output of a line sensor is a 1D array of pixel values (while a camera has a 2D array of pixel values as output). Nowadays, the line sensor is especially known for its use in combination with an optical system in image or object scanners that capture two-dimensional (2D) images by moving the object or the sensor (and optical system) perpendicularly to the scan line [Inada, 2006, Baird et al., 1992, Nishiyama, 2004, Kawamoto and Narabu, 1999].

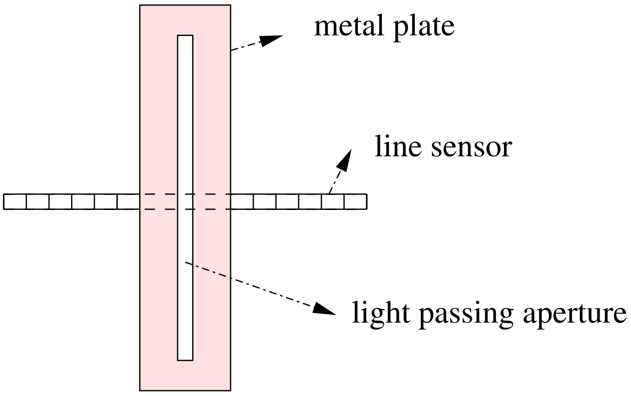
Since a line sensor has only a linear array of sensing elements (as opposed to a camera that has a 2D array of sensing elements), it is especially useful in vision applications where observations along one direction can be reduced to a single observation. We propose to use the line sensor to simplify this type of vision applications at the device level and in terms of processing requirements. The calculation of 2D occupancy maps of 3D scenes is an example of such an

[4] In practice, of course it will only be possible to combine observations along a direction over a certain finite length. This will also be the case for the system we describe in this section.

Table 4.1 Comparison between the system with multiple light-integrating line sensors and other 2D occupancy sensing systems.

System	Advantages of proposed setup
One camera (overhead or side view)	Cheaper, much less power-consuming, simpler to construct, infrastructure with no wiring and aesthetic impact, no possibility of privacy breach, higher data rates, more sensitive pixels, less expensive image processing, and no suffer from lack of accuracy due to uncertainty along the viewing direction (side view), deformation when moving away from the center of the captured images (overhead), or occlusions (side view and overhead) since we combine multiple sensors.
Multi-camera setup	Cheaper, much less power-consuming, simpler to construct, infrastructure with no wiring and aesthetic impact, no possibility of privacy breach, higher data rates, more sensitive pixels, and less expensive image processing.
Pressure sensitive carpets	Cheaper, much less power-consuming, simpler to construct, and infrastructure with no wiring and aesthetic impact.
Active radar, ultra-sound, or radio beacons (or a combination of these types of beacons)	No need for extensive noise cancellation techniques and heavy processing because no direction of arrival methods must be used, more accurate output magnitudes and larger angular resolution, and much less power-consuming.
PIR sensors	Output magnitudes more accurate (not binary (motion/no motion) but values within a range of 0-32000), much larger angular resolution (each pixel only senses light within a (small) range of incidence angles, and not within the whole sensing frustum). With the more accurate outputs and the larger angular resolution, higher occupancy accuracies can be achieved with the same number of sensors, or alternatively, fewer sensors are needed to achieve the same accuracy.

Figure 4.5 Front view of a line sensor and a slit.



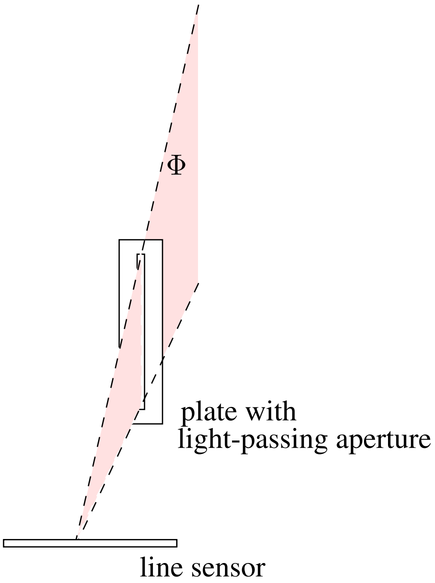
application.

To incorporate more observations along a certain direction we propose to use the line sensor together with an optical system that ensures that each sensing element of the sensor senses light from rays with incidence angles subject to specific geometric constraints. A simple example of such a light-integrating optical system is a slit perpendicular to the line sensor, which is shown in Figure 4.5. As shown in Figure 4.6, in this case each sensing element of the line sensor senses the light coming from a frustum centered around a two dimensional plane Φ. This plane Φ contains the considered light sensing element and the slit, and is oriented along the viewing direction. An optical system consisting of a slit is more attractive than a lens because it is easier and cheaper to produce and mount and still allows enough light to fall onto the sensor to produce useful output data. In this respect, note that the sensing elements of a line sensor are more light sensitive than for instance those of a regular camera, due to the large size of the line sensor pixels.

In the remainder of this chapter, the line sensor in combination with a light-integrating optical system will be denoted as light-integrating line sensor. To model the light-integrating line sensor system consisting of a line sensor and a slit (see Figure 4.5, light passing aperture), we use concepts from projective camera geometry [Hartley and Zisserman, 2004], but we need to adapt the formulations from camera geometry to this particular setup. First, as the slit integrates all incoming visible light along the slit direction and since the line sensor has only a 1D array of sensing elements, the projection matrices for this setup will be different from the projection matrices of a pinhole camera [Hartley and Zisserman, 2004]. In particular, the projection matrices will be 2×4 matrices instead of the typical 3×4 matrices in the camera case (for homogeneous coordinates) [Hartley and Zisserman, 2004].

We will clarify this with an example configuration, which is illustrated in Fig-

Figure 4.6 Illustration of the light-integrating capacity of a line sensor combined with a light-passing aperture. In this figure we show a particular example in which the light-integrating line sensor consists of a line sensor and a slit. In this case, the line sensor senses the light coming from a frustum centered around the two dimensional plane Φ. This plane Φ contains a light sensing element of the line sensor and the slit, and is oriented along the viewing direction.



ure 4.7. Assume that we have a pinhole camera with projection matrix $\mathcal{P}_{\text{cam}}$:

$$\mathcal{P}_{\text{cam}} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tag{4.10}$$

Let us use the notation $\mathbf{Y}$ for the point in the 3D world represented by the homogeneous 4-vector $[Y_1\ Y_2\ Y_3\ 1]^T$ and the notation $\mathbf{y}$ for the image point represented by the homogeneous 3-vector $[y_1\ y_2\ y_3]^T$. Then, the following equation holds

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ 1 \end{bmatrix} \tag{4.11}$$

Or more compactly,

$$\mathbf{y} = \mathcal{P}_{\text{cam}} \mathbf{Y}. \tag{4.12}$$

Hence, this projection matrix $\mathcal{P}_{\text{cam}}$ describes the mapping of a pinhole camera from 3D points in the world to 2D points in an image [Hartley and Zisserman, 2004].

The image plane of this camera is denoted by Ψ and the center of projection (called the camera center) is denoted by $\mathbf{C}$ (see Figure 4.7). As is common practice in the field of camera geometry, this image plane is assumed to be in front of the camera center in Figure 4.7. Let us now consider a line sensor $\mathbf{L}$ for which the array of line sensing elements coincides with one of the horizontal lines of the 2D array of sensing elements of the camera. The coordinates of the line sensor are denoted by y, which represents the homogeneous 2-vector $[y_1\ y_3]^T$. Additionally, assume that the slit $\mathbf{S}$ belonging to the line sensor is positioned perpendicularly to the line sensor array, through the optical center $\mathbf{C}$ of the considered camera and parallel to the camera sensor array. From Eq. 4.11, we can derive

$$\begin{bmatrix} y_1 \\ y_3 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ 1 \end{bmatrix}. \tag{4.13}$$

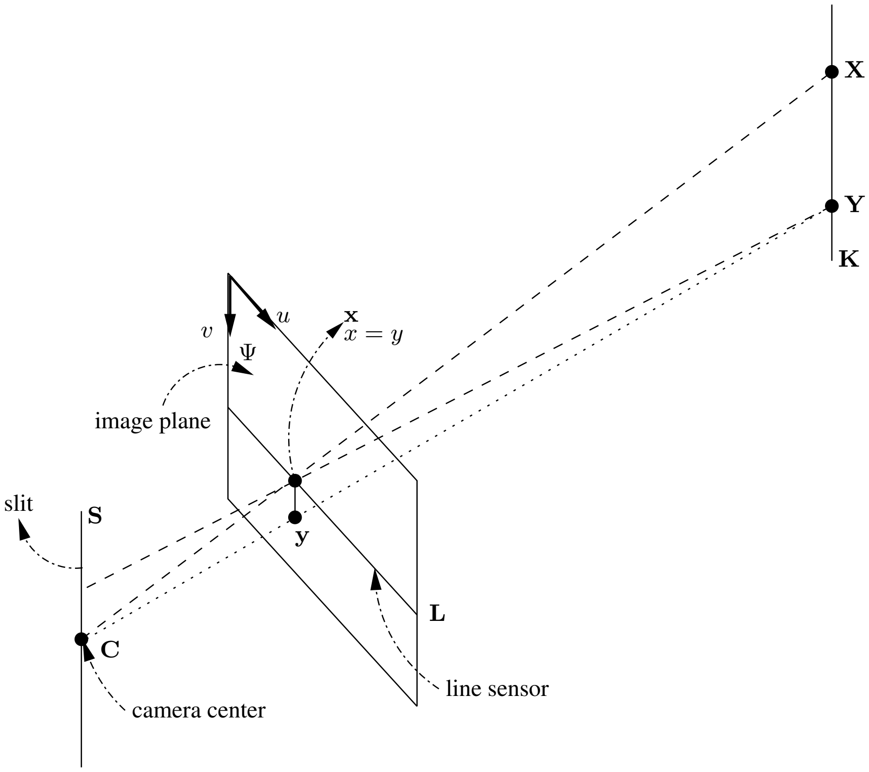
Or more compactly,

$$\mathbf{y} = \mathcal{P}_{\text{line}}\mathbf{Y}. \tag{4.14}$$

Hence, this projection matrix $\mathcal{P}_{\text{line}}$ for the light-integrating line sensor describes the mapping of the light-integrating line sensor from 3D points in the world to 1D points on a line. For the configuration of Figure 4.7, the projection matrix $\mathcal{P}_{\text{line}}$ for the line sensor can be obtained from the projection matrix $\mathcal{P}_{\text{cam}}$ by removing the second row of the matrix. This is logical since the light-integrating sensor in this configuration combines all observations along the v-direction of the image coordinate system (see Figure 4.7) into one observation.

The data coming from the light-integrating line sensor (a 1D array of pixel values) will be called a scan line. Note that a scan line can also be obtained from a camera image by summing up the values from the pixels in the camera sensor lying along parallel lines in the sensor array (see Eq. 4.8). Indeed, the integration over a certain direction is then executed by means of a sum calculation, while with the light-integrating line sensor this integration is performed optically. The direction of these parallel lines along which the summing is performed corresponds to the direction of the slit with respect to the line sensor. In Eq. 4.8, the pixel values are summed up in the vertical image direction, with the aim of reducing the memory and computational requirements for the occupancy calculation. The latter strategy corresponds to the light-integrating line sensor setup of Figure 4.5, 4.6 and 4.7. However, the advantage of using a light-integrating line sensor for this purpose is that the calculation of 4.8 is performed optically without any need

Figure 4.7 Geometry of line sensor with slit vs. pinhole camera geometry.



for processing/memory capacity and energy, and that we can profit from the more interesting characteristics of a line sensor compared to a regular camera: its lower price (around 4 euros), its lower power consumption (around 8 mWatt), its higher data rates, its higher bit depth and its privacy-friendly nature. We propose to use multiple of these line sensors (each viewing a scene from a different direction) to calculate an accurate 2D occupancy map.

In Section 4.3.3, we present foreground (FG)/background (BG) subtraction algorithms for scan lines, i.e. we determine the probability that a pixel of the line sensor is a foreground pixel. To test the efficiency of foreground FG/BG detection with scan lines, we compare the performance, computational complexity and memory requirements of this method with foreground FG/BG subtraction algorithms on images. In Section 4.3.4, we explain how we obtain a 2D occupancy map by fusing the FG/BG scan lines from multiple line sensors.

4.3.3 Foreground detection on scan lines

4.3.3.1 Related work

To automatically detect which pixels in an image belong to the foreground, the research community has developed a broad range of moving object detection algorithms, ranging from simple static background subtraction to complex techniques based on statistical background modeling.

Depending on the complexity of these algorithms, their output quality as well as the requirements they put on computational and memory resources varies greatly. A static background model is computationally very cheap and requires only one frame of memory. Unfortunately, this model cannot cope with lighting changes, changes in the background, etc. and thus often produces very poor results that are unusable outside of a controlled laboratory environment. More advanced techniques build statistical models of the background, e.g. using mixtures of Gaussian distributions [Stauffer and Grimson, 2000] or Bayesian classification of feature vectors [Li et al., 2003]. To further increase robustness, the incorporation of gradient information has been suggested [Javed et al., 2002, Tian et al., 2005]. These methods are able to accurately detect foreground regions even in the case of very complex backgrounds. This performance comes at the price of higher computational and memory costs. For example, with K the number of mixture components (usually three to seven), mixture of Gaussians background modeling requires $4K + 11$ arithmetical operations per pixel for every update of the background model and at least $2K$ frames of memory storage [Stauffer and Grimson, 2000]. Some methods aim at providing high quality foreground detection at low computational and memory cost, e.g. [Petrovic et al., 2009]. This method requires only two frames of memory storage and six arithmetic operations per pixel per background model update, without a significant loss in performance. [Zivkovic et al., 2008] and [Casares and Velipasalar, 2008] are other examples of light-weight foreground detection algorithms.

Foreground detection on a scan line is not fundamentally different from foreground detection on an entire image. It is therefore possible to use an existing algorithm and adapt its parameters to make it suitable for scan lines. We will describe our novel method for FG/BG detection on scan lines in Section 4.3.3.2. We outline its computational and memory requirements in Sections 4.3.3.3 and 4.3.3.4.

4.3.3.2 Method

The data coming from the light-integrating line sensor is called a scan line and is denoted by $H(x)$, with $x \in [1, \ldots, w_{\text{line}}]$ and w_{line} being the number of light sensing elements of the line sensor. The background B models what value the scan line would assume in the absence of moving objects in the scene. The binary foreground mask F assumes value 1 where the scan line value differs considerably from the current background model, and 0 elsewhere. We denote the background model and the foreground mask of $H(x)$ by respectively B_H and F_H.

As explained above, FG/BG segmentation on scan lines is not substantially different from FG/BG segmentation on full images, and hence we can develop our method starting from an existing FG/BG segmentation method for full images. As a proof of concept, we show this for the method of [Petrovic et al., 2009]. This method has been designed especially for low memory and computation cost, and is therefore appropriate to adapt for a low memory and low computational power platform such as a line sensor.

We refer to [Petrovic et al., 2009] for a detailed account of this method. Summarized, its background model consists of two components: a long-term background B^{long} which adapts slowly to scene changes, and a short-term background B^{short} which is used to detect the image areas where motion appears. We denote an image on time instant t by $I(x, y, t)$. At time step $t + 1$, the long-term background model is obtained as

$$B^{\text{long}}(x, y, t+1) = \alpha(x, y, t)I(x, y, t) + (1 - \alpha(x, y, t))B^{\text{long}}(x, y, t) \quad (4.15)$$

with the learning rate[5]

$$\alpha(x, y, t) = (1 - F(x, y, t))\alpha_I^{\text{long}}. \quad (4.16)$$

The short-term background is[6]

$$B^{\text{short}}(x, y, t+1) = \alpha_I^{\text{short}} I(x, y, t) + (1 - \alpha_I^{\text{short}})B^{\text{short}}(x, y, t). \quad (4.17)$$

The long-term foreground mask is the result of hysteresis thresholding:

$$F^{\text{long}}(t) = \text{hyst}(|I(t) - B(t)^{\text{long}}|), \quad (4.18)$$

which means that $F^{\text{long}}(x, y, t)$ assumes value 0 if the difference $|I(x, y, t) - B_{\text{long}}(x, y, t)|$ is strictly smaller than a low threshold $T_{I,L}$, value 1 if it is strictly bigger than a second higher threshold $T_{I,H}$, and if it lies between $T_{I,L}$ and $T_{I,H}$, the pixel (x, y) is assumed to be part of the foreground only if some neighborhood pixels have $|I(x, y, t) - B^{\text{long}}(x, y, t)| > T_{I,H}$.

[5] α_I^{long} was set to 0.01 in our experiments.

[6] α_I^{short} was set to 0.1 in our experiments.

The short-term foreground mask is obtained through simple thresholding:

$$F^{\mathrm{short}}(x,y,t) = \begin{cases} 1 & \text{if } |I(x,y,t) - B^{\mathrm{short}}(x,y,t)| > T_{I,L}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.19}$$

The foreground mask $F^{\mathrm{long}}(t)$ obtained from the long-term background is validated by the motion detection mask $F^{\mathrm{short}}(t)$ in order to construct the final foreground mask $F(t)$:

$$F(x,y,t) = F^{\mathrm{long}}(x,y,t)F^{\mathrm{short}}(x,y,t). \tag{4.20}$$

Because the horizontal scan line background model $B_H(x)$ will have a similar temporal behavior as $B_I(x,y)$, the learning rates for the scan line α_H^{long} and $\alpha_H^{\mathrm{short}}$ can be kept equal to the image learning rates α_I^{long} and $\alpha_I^{\mathrm{short}}$. However, the differences $|H(x) - B_H^{\mathrm{X}}(x)|$, $\mathrm{X} \in \{\mathrm{short}, \mathrm{long}\}$, will be close to an accumulation of the differences $|I(x,y) - B_I^{\mathrm{X}}(x,y)|$, and will thus have to be compared to a threshold $T_{H,L} \approx G_H T_{I,L}$ and $T_{H,H} \approx G_H T_{I,H}$, with G_H a parameter that depends on the height of the slit and the sensitivity of the line sensor pixels.

As mentioned in Section 4.2.3, a scan line can also be obtained from an image coming from a camera. A horizontal scan line of an observed image $I(x,y)$ is in particular its Radon transform along the vertical grid directions. This is equivalent to integrating the intensity function $I(x,y)$ over vertical lines to obtain the horizontal scan line, $H(x)$:

$$H(x) = \sum_{y=1}^{l} I(x,y) \tag{4.21}$$

where l is the image height. In this case, the parameter G_H is determined depending on the percentage of true positives (correctly classified foreground pixels) we would like to detect, and the number of false negatives (pixels that are mistakenly classified as background pixels) we allow. If we value detecting all true positives, G_H should be low. If we wish to avoid this, G_H should be chosen high, at the cost of a lot of false negatives [Tessens et al., 2009].

If we perform BG/FG detection on a scan line obtained from an image instead of on the full image, we also save computations and memory as in the line sensor case. This will be discussed in detail in Section 4.3.3.3 and Section 4.3.3.4. However, the advantage of using a light-integrating line sensor for this is that the acquisition of a scan line is performed completely optically without any need for processing or memory resources. Moreover, we can profit from the more interesting characteristics of a line sensor as compared to a regular camera, which were described in Section 4.3.1.

Table 4.2 Number of operations necessary to obtain a foreground mask, in general terms and for an example with CIF (352×288) resolution for the images and with 128 sensing elements for the line sensor.

	Full image	Scan lines from image	Scan lines from line sensor
General	$9lw$	$hw + 9w$	$9w_{\text{line}}$
Example	912384	104544	1152

4.3.3.3 Computational requirements

An update of the background of the method of [Petrovic et al., 2009] requires six arithmetic operations per pixels. If we make abstraction of the hysteresis thresholding, comparing the current image with the short- and long-term backgrounds requires another two operations. Foreground validation adds another operation per pixel, bringing the total number of arithmetic operations necessary for the generation of a foreground mask to nine per pixel or $9lw$ for an image of height l and width w (ignoring operations necessary for memory access).

Performing this algorithm on a scan line obviously reduces the number of operations necessary to obtain a foreground mask. Calculating the foreground mask of a scan line coming from a line sensor with w_{line} sensing elements requires $9n$ operations.

As explained in Section 4.3.3.2, one can also first obtain a scan line from an image and then perform BG/FG segmentation to save memory and computations for the BG/FG segmentation. The BG/FG segmentation of the full image can then be approximated by Eq. 4.9. To obtain a scan line from an image, hw additions are necessary, which means that in total $lw + 9w$ operations are performed to obtain the foreground mask of both vertical and horizontal scan lines.

These expressions for computational complexity are summarized on the first line of Table 4.2.

As an illustration, let us assume that $w = 352$, $l = 288$ (CIF resolution) and $w_{\text{line}} = 128$ (e.g. for the line sensor TSL1401CS-LF of TAOS). The second line of Table 4.2 indicates the number of operations required in this case.

For foreground detection on scan lines from light-integrating line sensors, this number is about a factor 100 of foreground detection on scan lines obtained from full images, and a factor 800 of the full image BG/FG segmentation. For foreground detection on scan lines obtained from full images, this number is about a factor 8 of the full image BG/FG segmentation.

Let us now assume the typical parameters $l = 3w/4$ and $w_{\text{line}} \approx w/2$. These are the parameters for CIF resolution and the line sensor TSL1401CS-LF of TAOS). Moreover, let us assume the number of bits for representing a pixel of the

Figure 4.8 Base 10 logarithm of the number of operations as a function of image width w. Method of [Petrovic et al., 2009] on full image (dash dotted line), on scan lines (full line) and on scan lines from light-integrating line sensors (dashed line).

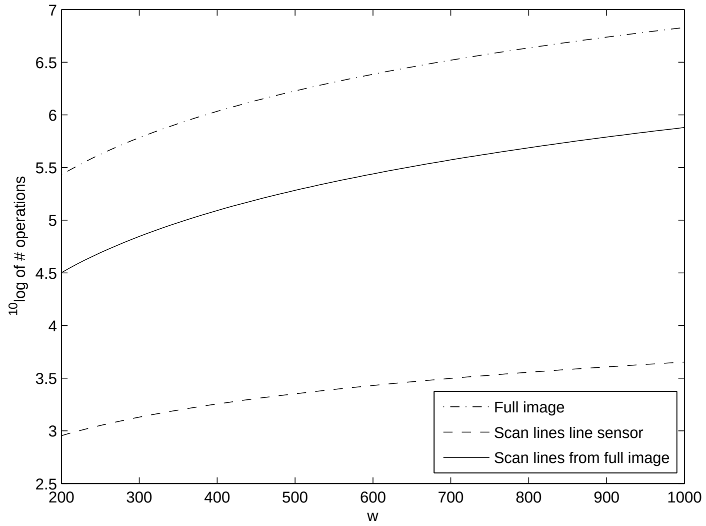


image is 8 ($b_{\text{image}} = 8$), and the number of bits for representing a pixel of the scan line is 12 ($b_{\text{line}} = 12$, as for line sensor TSL1401CS-LF of TAOS). Note that due to the high bit depth of line sensor data b_{line} is usually larger than b_{image}.

In Fig. 4.8, we plot on a logarithmic scale how the number of operations needed for foreground detection evolves as a function of the image width w. Approximately, we can assume that l and w_{line} scale proportionally with w through $l = 3w/4$ and $w_{\text{line}} \approx w/2$ (as indicated above). As a result, we conclude that calculating a FG/BG segmentation on a scan line of a full image saves computational resources, but the largest decrease in computational power is obtained by the use of light-integrating line sensors.

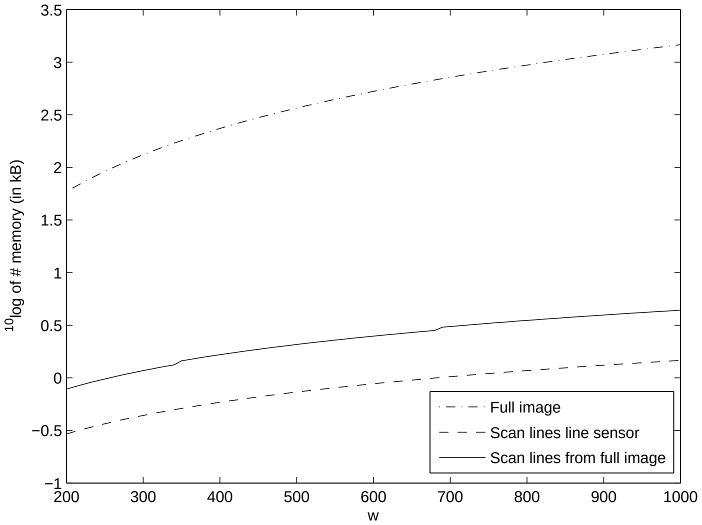
4.3.3.4 Memory requirements

Concerning the memory requirements, storing a static background requires one frame of memory. The method of [Petrovic et al., 2009] needs two frames. For an image of size $l \times w$ and with b_{image} bits per pixel, this means $2lwb_{\text{image}}$ bits of storage space.

For a line sensor of length n and with b_{line} bits per pixel, this means $2nb_{\text{line}}$ bits of storage space.

For the case of foreground detection on scan lines from full images, the mem-

Figure 4.9 Base 10 logarithm of the amount of memory (in kB) as a function of image width w. Method of [Petrovic et al., 2009] on full image (dash dotted line), on scan lines (full line) and on scan lines from light-integrating line sensors (dashed line).



ory requirements are as follows. To represent the sum of all pixel values along a horizontal line, with b bits per pixel, one needs $\lceil {}^2\log(l2^b)\rceil$ bits. The storage of a horizontal scan line thus requires $\left(\lceil {}^2\log(l)\rceil + b\right)w$ bits, and twice this number is necessary for foreground detection.

Again, let us consider as in Section 4.3.3.3 the typical parameters for CIF resolution and the line sensor TSL1401CS-LF of TAOS: $l = 3w/4$, $w_{\text{line}} \approx w/2$, $b_{\text{image}} = 8$ and $b_{\text{line}} = 12$. In Fig. 4.9, we plot on a logarithmic scale how the number of kilobytes of memory needed for foreground detection evolves as a function of the image width w. Approximately, we can assume that l and w_{line} scale proportionally with w through $l = 3w/4$ and $w_{\text{line}} \approx w/2$ (as indicated above). First, we can see from this figure that the reduction of required memory for the full image foreground detection as compared to foreground detection with line sensors, is up to a factor of 1000. Second, performing the foreground detection on a scan line of the image decreases the required memory by up to a factor of more than 300 as compared to operating on the full image. Between scan lines from full images and scan lines from light integrating light-sensors the reduction of required memory is up to a factor of around 3.

Therefore, it is clear that foreground detection on full images is very expensive memory-wise. Performing the detection on a scan line alleviates the memory problem. If we choose to use light-integrating line sensors, the memory requirements are even more reduced.

4.3.4 Occupancy maps from line sensor scan lines

With the method described in the previous section (Section 4.3.3), we segment the scan lines from the light-integrating line sensors into FG and BG. From these FG/BG scan lines we obtain a 2D occupancy map with a method related to the method described for multi-camera setups in Section 4.2.3. However, this method needs to be adapted to take the specificities of this setup into account. In particular, instead of having FG/BG segmentations of full images from all camera views at our disposal, we have in this scenario FG/BG segmentations of scan lines from all light-integrating line sensors. How we will deal with this difference, will be explained throughout the remainder of this section.

Similarly to Section 4.2.3, let us consider a network of N' light-integrating line sensors and let the ground plane of the observed scene be discretized in resolution cells $\mathbf{x}$. We wish to assign a real value to each cell $\mathbf{x}$ that expresses our confidence that the cell $\mathbf{x}$ is occupied.

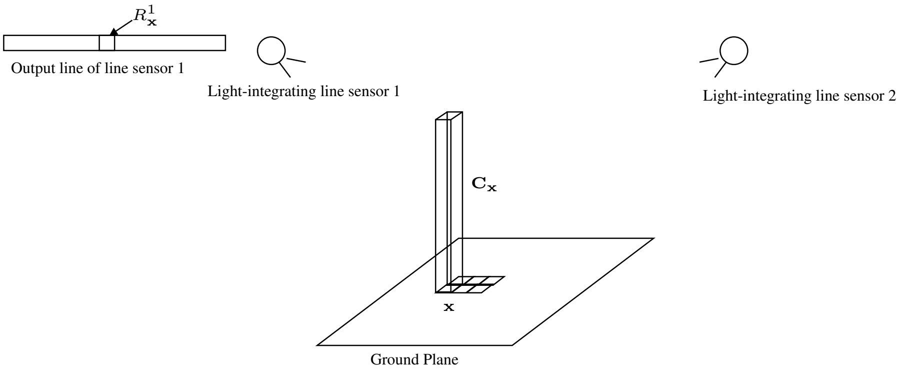
Also as in Section 4.2.3, for each cell $\mathbf{x}$ the mutually exclusive and exhaustive hypotheses that $\mathbf{x}$ is either occupied ($\{occ_{\mathbf{x}}\}$) or not ($\{nocc_{\mathbf{x}}\}$) constitute the frame of discernment $\theta_{\mathbf{x}} = \{occ_{\mathbf{x}}, nocc_{\mathbf{x}}\}$ [Dempster, 1968]. The information from each line sensor n', $1 \leq n' \leq N'$, is considered a distinct piece of evidence and we denote the BBA representing this evidence by $m_{n'}$. We now explain how we define the BBA for the case of light-integrating line sensors.

Let us consider a rectangular cuboid $\mathbf{C}_{\mathbf{x}}$ with height H and cell $\mathbf{x}$ as base, where H is the height along which the light-integrating slit can integrate light. If this 3D box lies completely outside the viewing frustum of line sensor n', this line sensor cannot provide any information about the occupancy of $\mathbf{x}$. The BBA is then $m_{n'}(\{occ_{\mathbf{x}}\}) = 0$, $m_{n'}(\{nocc_{\mathbf{x}}\}) = 0$ and $m_{n'}(\theta_{\mathbf{x}}) = 1$.

Otherwise, if this 3D box lies inside the viewing frustum of line sensor n', this cuboid corresponds to one or more sensing elements of line sensor n' and these line sensor elements define a line sensor region $R_{\mathbf{x}}^{n'}$ (see Fig. 4.10). The scan line from each line sensor is segmented into background and foreground, e.g. with a method as we presented in Section 4.3.3. Note that in Section 4.2.3, the region $R_{\mathbf{x}}^{n}$ was a rectangular region or a line-shaped region (the latter for $\mathbf{x}$ with sufficiently small area, see Figure 4.1). In this case $R_{\mathbf{x}}^{n'}$ is a horizontal line of pixels or simply one pixel (the latter for $\mathbf{x}$ with sufficiently small area).

Then, we determine in each region $R_{\mathbf{x}}^{n'}$ the fraction of background pixels $b_{\mathbf{x}}^{n'}$ and of foreground pixels $f_{\mathbf{x}}^{n'}$. Of course $b_{\mathbf{x}}^{n'} + f_{\mathbf{x}}^{n'} = 1$. Note, as in Section 4.2.3, that in most cases the resolution of the ground plane discretization is chosen such that the rectangular cuboid with as base a cell $\mathbf{x}$ is projected onto a region in the line sensor not wider than one sensing element of the line sensor. In this scenario,

Figure 4.10 The projection of a rectangular cuboid $\mathbf{C_x}$ with height H and cell $\mathbf{x}$ as base into light-integrating line sensor 1 defines a region $R^1_{\mathbf{x}}$ on the output line of line sensor 1.


$R^1_{\mathbf{x}}$
Output line of line sensor 1
Light-integrating line sensor 1
Light-integrating line sensor 2
$\mathbf{C_x}$
$\mathbf{x}$
Ground Plane

$f^{n'}_{\mathbf{x}}$ either equals 0 when the line sensing element corresponding to cell $\mathbf{x}$ is part of the background (and $b^{n'}_{\mathbf{x}}$ equals 1 then), or equals 1 when the line sensing element corresponding to cell $\mathbf{x}$ is part of the foreground (and $b^{n'}_{\mathbf{x}}$ equals 0 then).

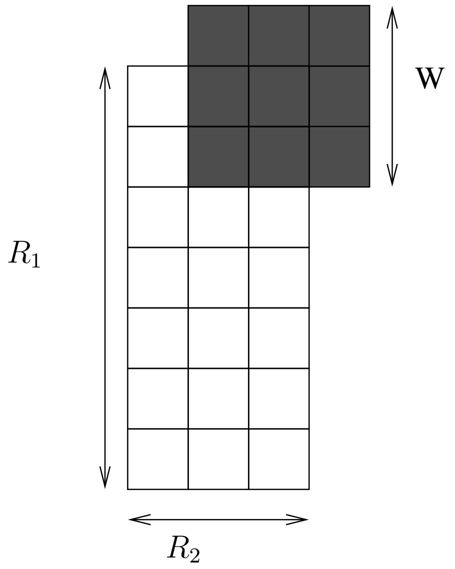
For the evidential approach, the evidence $m_{n'}(\{nocc_{\mathbf{x}}\})$ of line sensor n' for the hypothesis $\{nocc_{\mathbf{x}}\}$ is $b^{n'}_{\mathbf{x}} = 1 - f^{n'}_{\mathbf{x}}$ (as in Section 4.2.3):

$$m_{n'}(\{nocc_{\mathbf{x}}\}) = b^{n'}_{\mathbf{x}}. \tag{4.22}$$

For $m_{n'}(\{occ_{\mathbf{x}}\})$, the situation is more complicated and we have to deal carefully with the specific characteristics of the light-integrating line sensors. $f^{n'}_{\mathbf{x}} = 1$ may be attributable to a person occupying any of the resolution cells $\mathbf{x}$ which are sensed by the same sensing element of the line sensor. Let $G^{n'}_{\mathbf{x}}$ be the total number of cells being projected on the line sensor pixel corresponding to cell $\mathbf{x}$, for line sensor n' of the network of N' line sensors. Because of the line sensor projection geometry, these $G^{n'}_{\mathbf{x}}$ positions will be approximately laid out in a trapezoid, which we approximate by a rectangle $\mathbf{R}$ with dimensions $R_1 \times R_2$ (see Figure 4.11). Note that in the multi-camera setup of Section 4.2.3, these positions were laid out in an approximately *square*-shaped trapezoid. The reason for this shape difference is the difference between the projection geometry of a camera and a light-integrating line sensor (see Section 4.3.2).

Assuming a person occupies a square of W^2 cells, this person can be in $(R_1 + W - 1)(R_2 + W - 1)$ different positions with respect to the rectangle $\mathbf{R}$. A particular cell $\mathbf{x}$ in the rectangle $\mathbf{R}$ is only occupied in W^2 of all these positions.

Figure 4.11 Example of a rectangular approximation $\mathbf{R}$ of 7×3 resolution cells $\mathbf{x}$ for which the cuboid $\mathbf{C}_\mathbf{x}$ is projected onto the same region $R_\mathbf{x}^{n'}$. A person, represented here by the gray square with $W^2 = 9$ resolution cells, can assume $(R_1+W-1)(R_2+W-1)$ different positions such that it overlaps with $\mathbf{R}$. Hence, if $R_\mathbf{x}^{n'}$ is completely part of the foreground, there is a probability of $W^2/\big((R_1 + W - 1)(R_2 + W - 1)\big)$ that a particular cell is actually occupied by a foreground object.



Hence, the evidence is scaled with

$$g_{\mathbf{x}}^{n'} = W^2 / \big((R_1 + W - 1)(R_2 + W - 1)\big) \tag{4.23}$$

and $m_{n'}(\{occ_{\mathbf{x}}\})$ is obtained as

$$m_{n'}(\{occ_{\mathbf{x}}\}) = g_{\mathbf{x}}^{n'} f_{\mathbf{x}}^{n'}. \tag{4.24}$$

With $m_{n'}(\{occ_{\mathbf{x}}\})$ and $m_{n'}(\{nocc_{\mathbf{x}}\})$ defined, $m_{n'}(\theta_{\mathbf{x}}) = 1 - m_{n'}(\{occ_{\mathbf{x}}\}) - m_{n'}(\{nocc_{\mathbf{x}}\})$.

The pieces of evidence collected by the N' line sensors about each cell $\mathbf{x}$ are fused using Dempster's rule of combination (see Eq. 4.1). As mentioned in Section 4.2.2 and 4.2.3, Dempster's rule assumes that the sources that produced the evidence are uncorrelated. This is a valid assumption in a scenario in which the line sensors are not mounted very closely to each other. This is the case for the experimental setups discussed in Section 4.4, and hence Dempster's rule can be applied in these scenarios. If this assumption is not valid, the cautious conjunctive rule [Denoeux, 2008] should be used instead, as discussed in Section 4.2.2 and 4.2.3.

As in Section 4.2.3, this fusion process must be performed for each resolution cell. The occupancy map, which contains the fused evidence of occupancy for all occupancy map cells, is denoted by $m(\{occ\})$.

4.4 Results

In this section, we show the results of occupancy sensing with multiple cameras (from Section 4.2) and occupancy sensing with multiple line sensors (Section 4.3). We compare the results with the results of the state-of-the-art occupancy detection methods with multiple cameras from [Delannay et al., 2009] and [Fleuret et al., 2008].

Since we do not dispose of an actual light-integrating line sensor and to make the comparison with camera networks possible, we will make a simulation of the data coming from a light-integrating line sensor network from the data of a multi-camera network. To this end, we integrate the intensity functions $I(x, y)$ of the camera images over vertical lines to obtain the horizontal scan lines $H(x)$ (as in Eq. 4.21). These calculated scan lines simulate the data coming from the light-integrating line sensors, and hence on these scan lines we perform occupancy detection with the method as described in Section 4.3.

In Section 4.4.1, we compare the different proposed multi-camera methods with state-of-the-art multi-camera methods. In this section, we do not discuss the line sensor method since the used test basket ball data set [De Vleeschouwer and Delannay, 2009] is not suitable for a line sensor network and produces irrelevant results. In brief, the reason for this is that, due to the geometry of the light-integrating line sensor, we cannot deduce from a top view image the corresponding light-integrating line sensor data, and if we simply leave out the information from the top views, the remaining number of side views is too low to calculate an accurate occupancy map that distinguishes between the high number of players in this scene. This will be explained in Section 4.4.1. In Section 4.4.2, we then make an overall comparison of the different occupancy methods, including the light-integrating line sensor method, with a different camera network environment. Finally, in Section 4.5, we discuss a demo environment we developed from the method of Section 4.2.3.

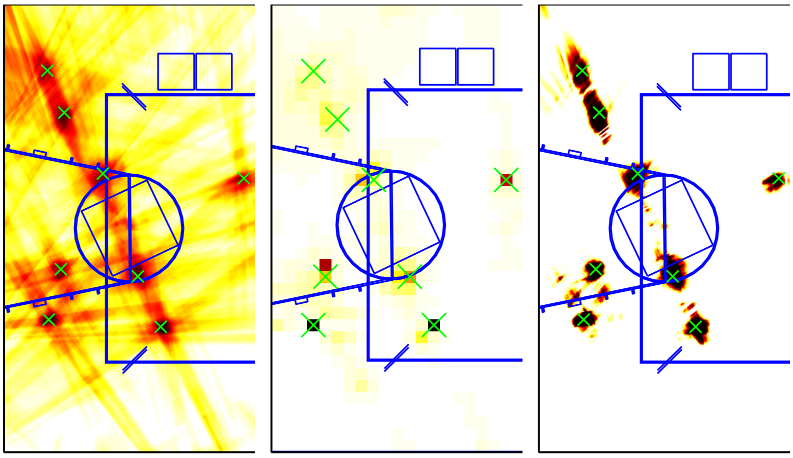
4.4.1 Comparison of multi-camera methods

To compare the three methods with multiple cameras, namely the method described in Section 4.2, the method from [Delannay et al., 2009] and the method from [Fleuret et al., 2008], we use the publicly available basketball dataset from the European project APIDIS [De Vleeschouwer and Delannay, 2009]. It consists of seven synchronized and calibrated video streams from five cameras with partially overlapping views distributed around the court, and two top-mounted cameras with fish eye lenses. The size of the field is $15m \times 28m$. There are on average 12 targets on the field. This camera setup is not suited for the extraction of scan lines to simulate a light-integrating line sensor network for two reasons. First, for the top views, integrating along the vertical direction does not yield the desired integration along the vertical direction in the scene. However, if we leave out the information from the top view cameras, the remaining number of side view cameras is too low to distinguish between the high number of players. Hence, the too low number of side views in relation to the number of players in the scene is a second reason why this setup is not suitable for the simulation of a light-integrating line sensor setup.

The videos are captured at 2 megapixel resolution and 25 fps. The average height of a player is set to $2m$, as in [Delannay et al., 2009]. We consider square resolution cells with an area of $(0.02m)^2$, also as in [Delannay et al., 2009]. In the rare case of conflicts in the fusion process, all evidence is transferred to $m(\theta_{\mathbf{x}})$. W is chosen to be $0.66m$. The foreground is detected using an algorithm based on mixture of Gaussians modelling [Stauffer and Grimson, 2000] with elementary shadow removal [Kaewtrakulpong and Bowden, 2001]. We have chosen these algorithms since they are well-known in the art and give satisfactory foreground detection results.

Ground truth target positions have been made available for 60 frames recorded at $1s$ intervals within the time interval 18:47 until 18:48 [De Vleeschouwer and

Figure 4.12 Example of the occupancy map results for the dataset of [De Vleeschouwer and Delannay, 2009], with (a) the aggregated occupancy map from [Delannay et al., 2009], (b) the probabilistic occupancy map from [Fleuret et al., 2008], (c) the proposed evidential occupancy map. White corresponds to low confidence/probability/evidence of occupancy, black to high. The crosses indicate the ground truth player positions.



(a) (b) (c)

Delannay, 2009]. As most cameras point to the left half of the court, only positions in that half are considered for the evaluation.

Fig. 4.12c shows an example of $m(\{occ\})$ as obtained with the Dempster-Shafer based method of Section 4.2.3, in part of the left half of the court. Fig. 4.12a shows the aggregated occupancy map obtained as in [Delannay et al., 2009] (see Section 4.2.1), Fig. 4.12b the probabilistic occupancy map of [Fleuret et al., 2008] (see Section 4.2.1) with cell width set to $0.4m$ (other widths yield less accurate results). The map obtained by DS fusion is very representative of the actual occupancy of the field because it shows very clearly defined peaks at the target positions, and very few ghost objects or interference strokes between objects. This is less the case for the methods of [Delannay et al., 2009] and [Fleuret et al., 2008].

Let the total mass (TM) be the sum over all cells of the occupancy evidence for the proposed method ($\mathrm{TM} = \sum_{\forall \mathbf{x}} \mathrm{m}(\{\mathrm{occ}_{\mathbf{x}}\})$), of the aggregated occupancy confidence for the method of [Delannay et al., 2009], and of the occupancy probability for the method of [Fleuret et al., 2008]. In Fig. 4.13, we plot for our method and the method of [Delannay et al., 2009] the percentage of TM that lies within a disc with diameter d around a ground truth target position as a function of d. For the method of [Fleuret et al., 2008] this evaluation method yields poor results since in this method the correlation between the occupancy probability of adjacent cells is explicitly ignored. Hence, to obtain good results with this method the size of the resolution cells should approximate the expected size of the objects to detect and this cell size is significantly larger than in our method and the method of [Delannay et al., 2009]. Therefore, for fair comparison we plot for the method of [Fleuret et al., 2008] for different cell widths d the percentage of TM that is generated in cells that are actually occupied by a target.

From Fig. 4.13 we conclude that in the proposed method the mass of occupancy evidence is more concentrated around the ground truth positions than the mass of occupancy confidence of the method of [Delannay et al., 2009] and the mass of occupancy probability of the method of [Fleuret et al., 2008]. This is obvious from the ratio between the percentage of total mass of our method and the method of [Delannay et al., 2009] and [Fleuret et al., 2008]. For [Delannay et al., 2009], this ratio ranges from $19.38\%/2.38\% = 8.13$ for $d = 40\mathrm{cm}$ to $94.18\%/43.96\% = 2.14$ for $d = 340\mathrm{cm}$, and reaches 6.66 for a typical diameter of $1m$ for sports players. For [Fleuret et al., 2008], it ranges from $4.65\%/0.65\% = 7.13$ for $d = 20\mathrm{cm}$ to $94.18\%/87.15\% = 1.08$ for $d = 340\mathrm{cm}$, and reaches 1.24 for $d = 1m$. In other words, the ground occupancy map obtained using the proposed method is more accurate than using the methods of [Delannay et al., 2009] and [Fleuret et al., 2008]. This is beneficial for direct use or for further analysis of the map.

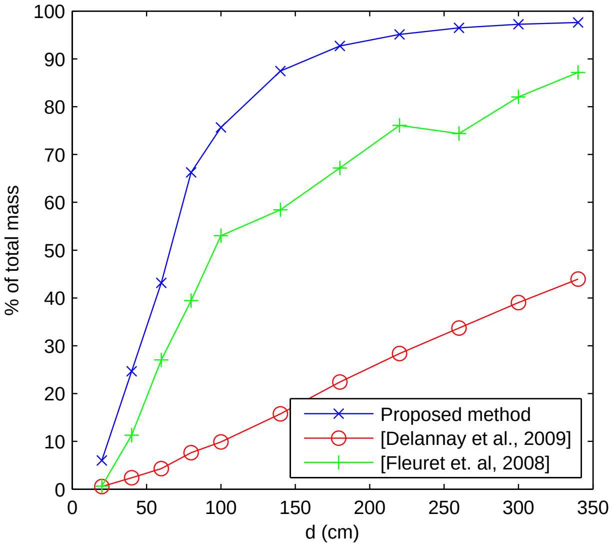
4.4.2 Comparison of all methods

The test environment we will use is an indoor scene of $5m$ by $4m$ observed by $N = 10$ web cameras. Approximately 8 minutes of footage (2400 frames) in which two, three and four persons appear have been recorded at 5 frames per second and at CIF resolution (352×288). Only the starting points of these recordings have been synchronized.

Fig. 4.14 and Fig. 4.15 show example occupancy maps with respectively two and three persons in the scene. Fig. 4.14c and Fig. 4.15c show $m(\{occ\})$ as obtained with the Dempster-Shafer based method of Section 4.2.3. Fig. 4.14d and Fig. 4.15d show $m(\{occ\})$ obtained by using BG/FG images that are the extension of the FG/BG scan lines (Eq. 4.9). The BG/FG scan lines were obtained from the BG/FG images on full images by applying the Radon transform on these BG/FG images. We will further refer to this method as *scan lines after FG/BG*.

Fig. 4.14e and Fig. 4.15e show for the same frame $m(\{occ\})$ as obtained with the light-integrating line sensors method of Section 4.3.4. For this curve, we again use BG/FG images that are the extension of the FG/BG scan lines (Eq. 4.9). However, compared to Fig. 4.14d and Fig. 4.15d, the BG/FG scan lines were obtained by applying the FG/BG subtraction methods on scan lines directly with the method as described in Section 4.3.3. These scan lines were obtained from the original full

Figure 4.13 The percentage of the total mass within a disc with diameter d around a ground truth target position (for the proposed methods and the method of [Delannay et al., 2009]), or within cells with width d actually occupied by a target (for method [Fleuret et al., 2008]).



images by applying the Radon transform on the images themselves (and not on the BG/BG images as is the case for Fig. 4.14d and Fig. 4.15d). The latter scenario is a simulation of the scenario with light-integrating line sensors. We will refer to this method as *scan lines before FG/BG*. In the following, we will jointly refer to the latter two methods (used for Fig. 4.14/4.15d and Fig. 4.14/4.15e) as *scan line methods*.

To compare with the state-of-the-art, we show in Fig. 4.12a the aggregated occupancy map obtained as in [Delannay et al., 2009], and in Fig. 4.12b the probabilistic occupancy map of [Fleuret et al., 2008] with cell width set to $0.4m$ (other widths yield less accurate results). As in Section 4.4.1, the map obtained by DS fusion with full images is very representative of the actual occupancy of the field and shows very clearly defined peaks at the target positions. The same holds for the scan line methods, both when we obtain the BG/FG scan lines directly from the full BG/FG images or with the light-integrating line sensor method. With the scan line methods, however, we sporadically observe ghost objects, which are objects that appear in the map, without having any person occupying that space. This happens more often when more persons appear in the scene (compare e.g. Fig. 4.14 with two persons to Fig. 4.15 with three persons) and in zones that are observed by fewer cameras (e.g. at the borders of the map). The latter problems can be alleviated by using more sensors. If we compare our methods to [Delannay et al., 2009], we observe fewer ghost objects or interference strokes between objects, both for our full image method and for the scan line methods. Compared to [Fleuret et al., 2008], the occupancy map as obtained with our methods is more concentrated around the actual ground truth positions, indicated by the crosses. For instance, in Fig. 4.14a, the occupancy mass belonging to the person that is situated most right in the map, is moved to the right compared to the actual ground truth position. The same happened in Fig. 4.15a for the two most right persons.

Similar to Section 4.4.1, we plot in Fig. 4.16 for our methods and the method of [Delannay et al., 2009] the percentage of TM that lies within a disc with diameter d around a ground truth target position as a function of d. As in Section 4.4.1, for fair comparison with the method of [Fleuret et al., 2008] we plot for this method for different cell widths d the percentage of TM that is generated in cells that are actually occupied by a target.

From Fig. 4.16 we conclude that for the full image DS method the mass of occupancy evidence is most concentrated around the ground truth positions. For d smaller than $0.8m$, the *scan line after FG/BG* outperforms the method of [Fleuret et al., 2008]. This is due to the fact that the *scan line after FG/BG* method estimates more occupancy around the actual target position than [Fleuret et al., 2008], as discussed above. However, *scan line after FG/BG* method has more ghost objects that are far away from actual targets and hence for larger d, this method does not reach the ideal 100% on the figure as opposed to [Fleuret et al., 2008]. The light-integrating line sensors method has a performance that is comparable to [Fleuret et al., 2008], despite the fact that only scan lines are observed in this case. Note, however, that the light-integrating line sensors have many advantages compared to the multi-camera method. For instance, their cost price is much lower, which would allow us to easily increase the number of sensors and in this way boost the performance. The method of [Delannay et al., 2009] performs worst, due to the large number of ghost objects and interference strokes between the objects.

4.5 Real-time demonstrator

We have implemented the method of Section 4.2 in a camera network installed at Hogeschool Gent to calculate ground occupancy in real time. The network consists of four progressive CCD color cameras with a resolution of 1024×768, each connected to an Intel Core 2 Duo/1.86GHz processor. The cameras are mounted at a height of around $3m$. Each camera plus computer simulates a smart camera. A base station with the same processor completes the network. The cameras observe an indoor scene of $6m$ by $4m$. The resolution cells $\mathbf{x}$ have a size of $0.5cm$ by $0.5cm$.

Each camera $n = 1 \ldots 4$ performs foreground detection based on mixture of

Gaussians modeling [Stauffer and Grimson, 2000] with elementary shadow removal [Kaewtrakulpong and Bowden, 2001] and calculates for all $\mathbf{x}$ $m_n(\{occ_{\mathbf{x}}\})$, where $g^n_{\mathbf{x}}$ is always set to one. $m_n(\{occ\})$ is transmitted over an Ethernet cable to the base station.

As $g^n_{\mathbf{x}}$ is always one, $m_n(\theta_{\mathbf{x}})$ depends only on the calibration parameters of the camera (i.e., the viewing range) and is stored at the base station. The base station calculates $m_n(\{nocc_{\mathbf{x}}\})$ as $m_n(\{nocc_{\mathbf{x}}\}) = 1 - m_n(\{occ_{\mathbf{x}}\}) - m_n(\theta_{\mathbf{x}})$. The occupancy maps of the single cameras $m_n(\{occ\})$ are fused using Dempster's rule of combination to obtain the final occupancy map $m(\{occ\})$.

The base station starts calculating the occupancy map $m(\{occ\})$ as soon as it has received a new $m_n(\{occ\})$ from all four cameras since the last time $m(\{occ\})$ was calculated. However, to make the system resilient against transmissions getting lost or the occupancy calculation of single cameras being delayed, the base station is also programmed to operate at a minimal frame rate fps$_{\min}$. If after a time 1/fps$_{\min}$ it has not received data from all cameras yet, the last received $m_n(\{occ\})$ is used as the current one for all cameras n from which no data was received. In our system, fps$_{\min}$ is set to 2 fps, since the lower bound of the system frame rate is currently 2 fps. The system frame rate will be discussed in more detail below.

In this demonstrator additionally some area of the ground plane is marked as a forbidden zone. People walking in the forbidden zone trigger an alert. The alert is triggered as soon as one third of the total mass TM is in the forbidden zone.

The system currently operates at 2 to 3 fps. The bottleneck is the calculation of $m_n(\{occ\})$ at the camera side. A more efficient implementation with integral images to calculate $b^n_{\mathbf{x}}$ and $f^n_{\mathbf{x}}$ would be a straightforward way to speed up calculations. Another way to speed up the calculations would be a reduction of the size of the resolution cells $\mathbf{x}$ which have a size of $0.5cm$ by $0.5cm$ and which could be set to $5cm$ by $5cm$ or even larger depending on the desired accuracy. These speed-up possibilities would also help to reduce the latency of the system, which currently amounts to about $1s$.

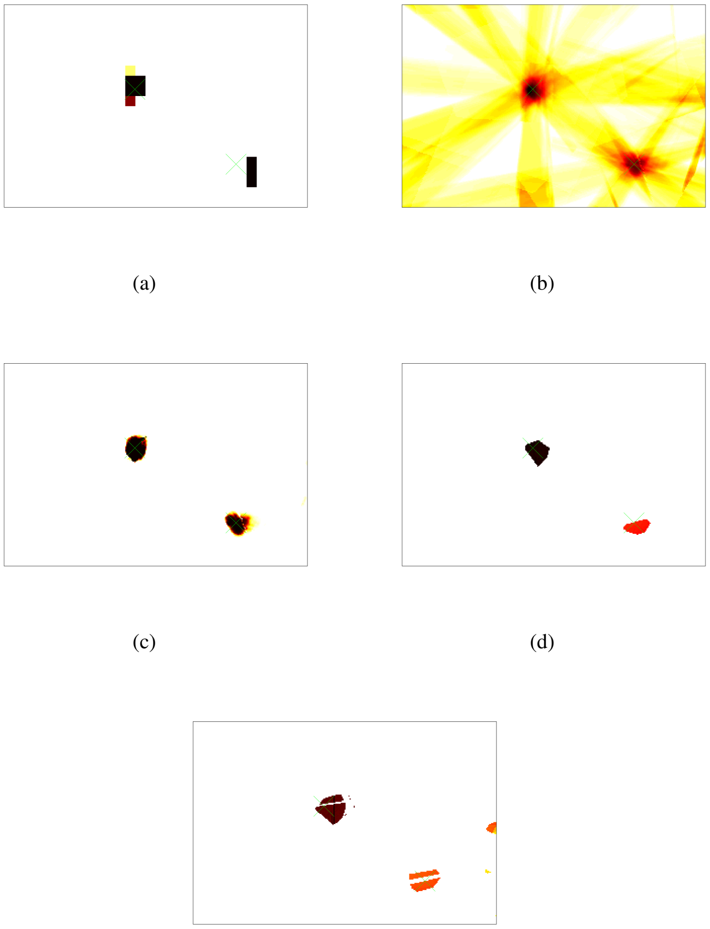
Fig. 4.17 shows a picture of the demonstrator in use. A video explaining its operation is also available [Tessens and Morbee, 2010]. The carpet marks the observed scene. People are allowed to walk on the light gray track and the dark gray carpet is the forbidden zone. The projector screen on the right shows the alert level on the left - green means no alert at this moment - and the occupancy map on the right. Yellow indicates high evidence of occupancy. The black track represents the allowed zone and the blue regions the forbidden area. The system latency clearly shows up in Fig. 4.17. Indeed, the region of high occupancy evidence corresponding to the left person on the projector screen matches the location where the person was standing about 1 s prior to the current scene. The right person has been stationary for the past second and is therefore shown at the correct location.

4.6 Conclusion

In this chapter, we have described new methods to calculate occupancy maps. The first method makes use of the data from multiple cameras. In particular, we have shown how the performance of a method requiring only forward projections from the images to the ground plane can be significantly improved by Dempster-Shafer based fusion of the single view ground occupancy maps. Experiments show clear improvements in the occupancy maps obtained with our multi-camera method compared to the other state-of-the-art multi-camera methods in terms of concentration of the occupancy evidence around ground truth person positions. However, the use of multiple cameras for occupancy reasoning has, notwithstanding the very accurate occupancy results, several disadvantages: the possibility of privacy breach, the expensive changes to the infrastructure, the high-complexity processing, and the large power consumption. To overcome these drawbacks, we proposed a second method which makes use of a network of light-integrating line sensors. This novel system is privacy-friendly, can be installed with minimal changes to the infrastructure, is cheap, requires only low computational power, and can be battery-operated. In terms of the accuracy of the occupancy results obtained by this method, there is a small loss in performance compared to our multi-camera method if we use the same number of sensors. Compared to the other state-of-art multi-camera methods, however, comparable or better occupancy results are obtained.

The research work presented in this chapter has been published in one journal paper [Morbee et al., 2010]. Two patent applications about this work have been filed [Morbee and Tessens, 2010, Morbee and Tessens, 2011]. Initial results have been published in one chapter of Lecture Notes of Computer Science [Lee et al., 2008], and at several international conferences [Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008]. Furthermore, we developed a real-time 2D occupancy demonstrator that shows the applicability and accuracy of our method in real-life applications.

Figure 4.14 Example of the occupancy map with two persons in the scene, with (a) the aggregated occupancy map from [Delannay et al., 2009], (b) the probabilistic occupancy map from [Fleuret et al., 2008], (c) the proposed evidential occupancy map from full images, (d) the proposed evidential occupancy map from scan lines obtained after FG/BG extraction on full images and (e) the proposed evidential occupancy map from light-integrating line sensors (scan lines obtained before BG/FG extraction). White corresponds to low confidence/probability/evidence of occupancy, black to high. The crosses indicate the ground truth person positions.



(a)

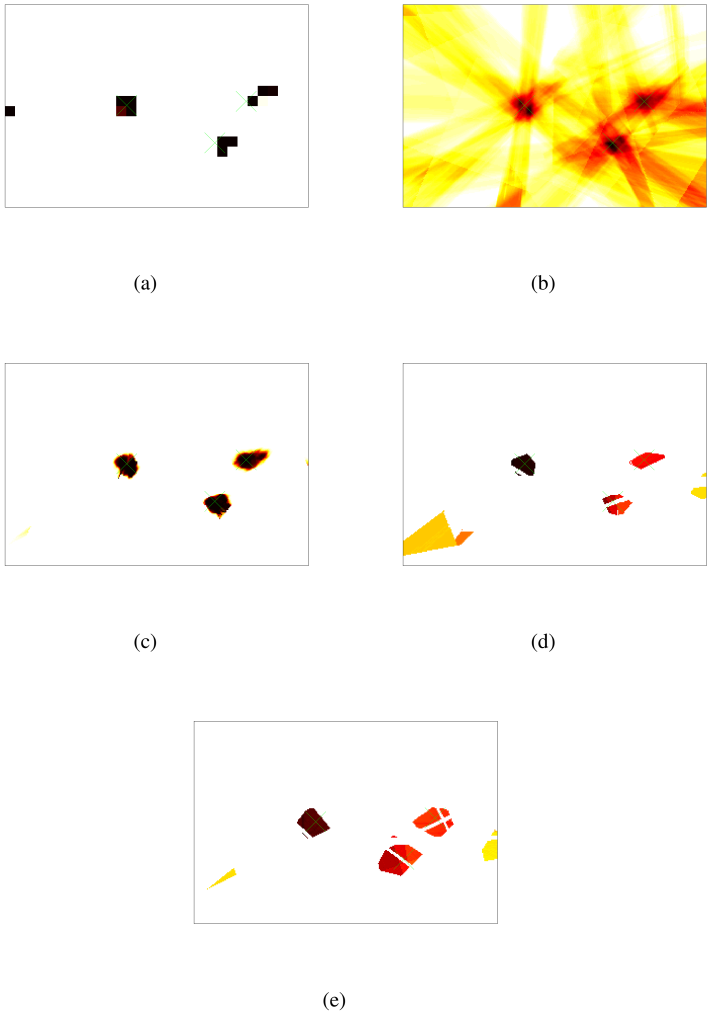
(b)

(c)

(d)

(e)

Figure 4.15 Example of the occupancy map with three persons in the scene, with (a) the aggregated occupancy map from [Delannay et al., 2009], (b) the probabilistic occupancy map from [Fleuret et al., 2008], (c) the proposed evidential occupancy map from full images, (d) the proposed evidential occupancy map from scan lines obtained after FG/BG extraction on full images and (e) the proposed evidential occupancy map from light-integrating line sensors (scan lines obtained before BG/FG extraction). White corresponds to low confidence/probability/evidence of occupancy, black to high. The crosses indicate the ground truth person positions.



(a)

(b)

(c)

(d)

(e)

Figure 4.16 The percentage of the total mass within a disc with diameter d around a ground truth target position for the different occupancy sensing methods.

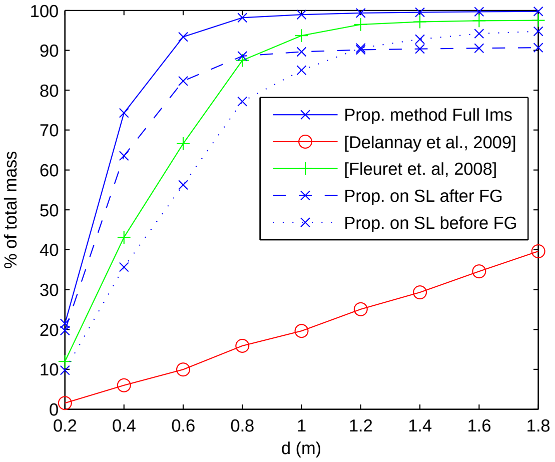
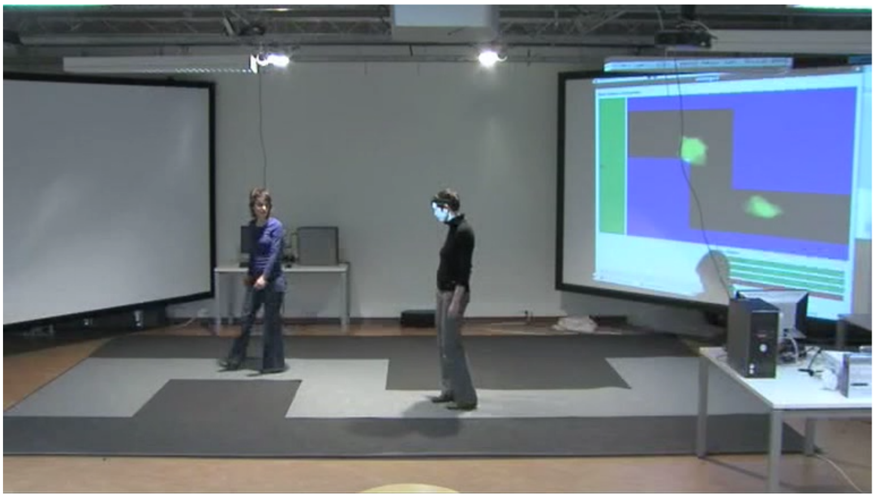


Figure 4.17 Real-time demonstrator in use. The carpet marks the observed scene. People are allowed to walk on the light gray track and the dark gray carpet is the forbidden zone. The projector screen on the right shows the alert level on the left - green means no alert at this moment - and the occupancy map on the right. Yellow on the screen indicates high evidence of occupancy. The black track on the screen represents the allowed zone and the blue regions the forbidden area.



5
Task Assignment in Vision Networks

5.1 Introduction

Vision networks can perform many types of tasks, e.g. tracking a person in the scene, recognizing the gesture of a person, detecting abnormal behavior. For many scene-related tasks, a camera network with overlapping fields of view provides substantial advantages over a single fixed viewpoint camera because sensing data from different nodes can be fused to perform the task. For example, in a tracking application, camera networks can alleviate occlusion problems; in gesture recognition and abnormal behavior detection, cues from different viewpoints can lead to a more robust decision.

A practical multi-camera network often takes care of several of these tasks simultaneously. For example, in a room in which multiple persons are present, the tracking of each of these persons is a different network task which the camera network should take care of.

The number and the type of tasks a camera network can deal with is of course limited by the network resources. The most important camera network restrictions are the limited computational power of the cameras and the communication constraints. The communication constraints are the maximal amount of data that can be transmitted over the network.

In a practical multi-camera network charged with multiple tasks and with restricted network resources the aim is to achieve the best overall task performance (an accurate definition of this concept will be given in Section 5.3) by distributing the tasks in an efficient way among the sensors in accordance with the given restrictions. This distribution of tasks among the sensors is called task assignment. In this work we present a novel, general solution to task assignment in practical vision networks (i.e. vision networks with network restrictions) with overlapping fields of view.

A first crucial component in such a task assignment system is quantifying the contribution of one or more cameras to the accomplishment of a task. The contribution of a camera set depends on the view point of the camera(s) and on the scene configuration, which is subject to change over time. In the case of view-correlated nodes, the event of interest may be simultaneously observed by several sensors, but not all cameras are equally suited to perform the task at hand. In [Tessens et al., 2011], we proposed a unifying approach to integrate quality of view measures in a criterion for the contribution of a sensor to a task founded on generalized information theory. This criterion is treated extensively in the PhD of Linda Tessens [Tessens, 2010].

In [Tessens et al., 2011] and in the PhD of Linda Tessens [Tessens, 2010], however, the focus lies on determining, for *one* task, which cameras are most suited to perform it. In this chapter, we go a step further. We investigate how we can efficiently distribute *a plurality* of tasks among the network cameras, taking into account that the camera network is constrained in terms of computational power and communication capabilities. To solve the constrained task assignment optimization problem, we propose two greedy optimization methods and study their complexity and performance.

As a proof of concept, we apply our general task assignment method to a camera network that is charged with the tracking of multiple persons. We evaluate how the tracking performance is influenced by computational constraints in the network. We test our method on extensive real data from different camera network environments.

The remainder of this chapter is organized as follows. In Section 5.2, we give an overview of the related literature. A formal problem formulation is provided in Section 5.3. In Section 5.4, we discuss how network constraints are integrated into the problem formulation by using cost functions. Section 5.5 briefly describes the camera set suitability value to quantify the contribution of one or more cameras to the accomplishment of a task, which was discussed in detail in the PhD of Linda Tessens [Tessens, 2010]. The greedy optimization solutions and their complexity compaired to the non-greedy solution, are described in Section 5.6. In Section 5.7, our task assignment framework is applied to a multi-camera multi-person tracking scenario. The experimental results are discussed in Section 5.8, and the conclusions are presented in Section 5.9.

5.2 Related work

Task assignment has received ample attention in literature. [Xiong and Svensson, 2002] and [Rowaihy et al., 2007] provide interesting overviews of this field. One approach is to model the network as a multi-agent system in which the sensors (the agents) seek to increase their personal utility by performing (parts of) one or more tasks [Kraus et al., 1995, Zlotkin and Rosenschein, 1991, Shehory and Kraus, 1998]. Market-oriented approaches are popular in this context [Dash et al., 2007, Mullen et al., 2006, Ostwald et al., 2005]. Alternatively, the agents may also be modeled to display cooperative behavior [Talukdar et al., 1998, Bowyer and Bogner, 1999, Hager and Durrant-Whyte, 1986, Luo et al., 1998]. Sensor assignment can also be modeled using a bipartite matching problem, where two node sets (the sensors and the tasks) need to be matched with largest possible weight under some constraint on the multiple usage of nodes [Kwok et al., 2002]. Alternatively, it can be formulated as a partially observed Markov decision problem [Boutilier et al., 1999, Castanon, 1997] or sensor resources can be managed based on fuzzy logic [Gonsalves and Rinkus, 1998, Smith III, 2004]. The task assignment problem can also be solved by formulating a search objective and then performing either a global [Molina Lopez et al., 1995] or a heuristicly or optimally reduced search [Kalandros and Pao, 1999, Bian et al., 2006, Dang et al., 2006]. However, all these papers treat the sensors as abstract agents which operate on artificial data and incur ideal costs.

Next to these theoretical approaches, there also exist practical task assignment approaches. In [He et al., 2003, Lee and Zomaya, 2007], the problem of task scheduling for a grid computing environment is studied, in which computers from multiple administrative domains combine their resources to reach a common goal. Some other approaches focus in particular on vision networks. Bakhtari et al. [Bakhtari and Benhabib, 2007] propose an active vision system for sensor selection, positioning and orientation with the aim of imaging and recognizing multiple targets in surveillance environments. In [Yao et al., 2010], a task assignment algorithm is proposed for camera networks to uniformly distribute computational load over the cameras. In [Soro and Heinzelman, 2007], energy limitations of wireless camera nodes are integrated in the camera assignment in order to reduce the overall network power consumption. Camera assignment for tracking and localization has been studied in [Isler and Bajcsy, 2005, Isler et al., 2005, Pahalawatta and Katsaggelos, 2004, Denzler et al., 2003, Sommerlade and Reid, 2008, Snidaro et al., 2003, Gupta et al., 2007, Ercan et al., 2006]. These methods will be discussed in more detail in Section 5.7.

Most mentioned camera assignment approaches either mainly deal with limited computational and energy capacities [Yao et al., 2010, Soro and Heinzelman, 2007, Pahalawatta and Katsaggelos, 2004], or choose task quality as the primary criterion under the assumption of unlimited computational power [Gupta et al., 2007, Denzler et al., 2003, Sommerlade and Reid, 2008, Snidaro et al., 2003, Ercan et al., 2006]. Other approaches focus on specific active vision systems [Bakhtari and Benhabib, 2007, Davis et al., 2007], while we aim at formulating a general solution. The novelty of our work compared to these works is that we propose a task assignment framework that is general and that measures and controls the quality of multiple tasks, while simultaneously coping with the computational and communication constraints of the camera network. To prove the applicability of our method, we apply our scheme in Section 5.7 to a multi-camera multi-person tracking scenario and test it on several real-life environments (see Section 5.8). First, we start with a general formulation of the problem in the next section.

5.3 Problem formulation

Consider a network of N cameras n, $1 \leq n \leq N$, charged with K tasks k, $1 \leq k \leq K$. Each task is assigned to a set of cameras. Let S^k denote the set of cameras selected for task k. $\mathcal{S}$ is the set of the sets selected for all tasks: $\mathcal{S} = \{S^1, \ldots, S^k, \ldots, S^K\}$. In other words, $\mathcal{S}$ represents a specific assignment of all tasks k to camera sets S^k. Note that not all S^k contain the same number of cameras.

An assignment $\mathcal{S}$ puts each camera n in the network in a certain sensing state $\mathbf{s}_n$. The state of a camera indicates which tasks are assigned to it. Let $s_n^k, k = 1, \ldots, K$ be binary numbers indicating if the task k uses camera n or not. Then,

$$\mathbf{s}_n = \left[\, s_n^1 \ldots\ s_n^k\ \ldots\ s_n^K \right] \tag{5.1}$$

If camera n is charged with task k, then $s_n^k = 1$. If camera n is not charged with task k, then $s_n^k = 0$. In a scenario in which multiple persons are tracked (this case will be discussed in detail in Section 5.7), the tracking of each person can be considered a different task k. In this example, the state of the camera indicates which persons in the scene are tracked by this particular camera. If there are e.g. five persons in the scene ($K = 5$) and camera 3 tracks the persons that correspond to tasks $k = 2, 3$ and 5, then the state of camera 3 is

$$\mathbf{s}_3 = \left[\, 0\ 1\ 1\ 0\ 1 \right] \tag{5.2}$$

The number of ones in state vector $\mathbf{s}_n$ (denoted by $|\mathbf{s}_n|$) is the total number of tasks assigned to camera n in the assignment $\mathcal{S}$. Obviously, $|\mathbf{s}_n| \leq K$.

Note that $\mathcal{S}$ and $\mathbf{s}_n$ are not independent of each other. In particular, if $\mathbf{s}_n^k = 1$, then $n \in S^k$. If $\mathbf{s}_n^k = 0$, then $n \notin S^k$.

The sets $S^k \in \mathcal{S}$ may or may not overlap. We call a camera that is selected for at least one task an *active* camera. In other words, for an active camera n holds that $|\mathbf{s}_n| \geq 1$. The number of active cameras of a set $\mathcal{S}$ is denoted by $N_{\mathcal{S}} \leq N$.

Some camera sets are more suited for certain tasks than others. For example, if a person is mostly occluded in one camera view, that camera may be less useful in determining the person's position. However, the situation is more complicated, because tasks are not performed by a single camera but by a set of cameras. Therefore, we associate a *suitability value* $v(S^k, k)$ with each possible camera selection set S^k. The total suitability value V for a specific assignment $\mathcal{S}$ of the tasks to the cameras, is a function of the values $v(S^k, k)$ of the K tasks. This suitability value function is defined as

$$V(\mathcal{S}) = g(v(S^1, 1), \ldots, v(S^K, K)) \tag{5.3}$$

Operating a camera n in a state $\mathbf{s}_n$ incurs a *cost*, denoted by $c(\mathbf{s}_n, n)$. This cost function returns the cost of taking measurements from camera n for the assigned tasks, and can represent different physical properties, such as the communication cost associated with using a camera or the computational cost of executing its assigned tasks on it. In Section 5.4, we will show how we will model the costs. The total cost C incurred in the considered network of N cameras for a specific assignment $\mathcal{S}$ of the tasks to the cameras, is a function of the costs $c(\mathbf{s}_n, n)$ of the N cameras. This cost function is defined as

$$C(\mathcal{S}) = h(c(\mathbf{s}_1, 1), \ldots, c(\mathbf{s}_N, N)) \tag{5.4}$$

To combine the concept of suitability value (expressed by Eq. 5.3) with the concept of cost (expressed by Eq. 5.4), we introduce the *welfare* W of our system, which is defined as[1]

$$W(\mathcal{S}) = V(\mathcal{S}) - C(\mathcal{S}). \tag{5.5}$$

A camera network aims at maximizing its welfare W, or in other words aims at maximizing the values and minimizing the costs. Hence, solving the task assignment problem comes down to finding the assignment $\mathcal{S}^*$ that maximizes the system welfare:

$$\mathcal{S}^* = \arg\max_{\mathcal{S} \in \Gamma} W(\mathcal{S}) = \arg\max_{\mathcal{S} \in \Gamma} (V(\mathcal{S}) - C(\mathcal{S})) \tag{5.6}$$

[1] To be able to use this expression for the welfare, the suitability value function $V(\mathcal{S})$ (Eq. 5.3) and the cost function $C(\mathcal{S})$ (Eq. 5.4) should include an adequate weighting factor, such that the subtraction is performed between appropriately weighted terms.

where Γ denotes the set of all possible assignments. Performing the joint task assignment optimization of Eq. 5.6 offers the possibility of controlling the quality with which tasks are performed (through the suitability value function Eq. 5.3), while distributing the tasks among the cameras according to practical criteria (through the cost function Eq. 5.4). Unfortunately, Eq. 5.6 is NP-hard for arbitrary cost and value functions [Dang et al., 2006]. In specific cases, this optimization problem can be solved in polynomial time. In the following Sections 5.4 and 5.5, we will propose cost and value functions that allow us to simplify the solution of Eq. 5.6. In particular, we will use the cost function (Eq. 5.4) to model the practical criteria which influence how tasks should be assigned (Section 5.4). To monitor the quality of the executed tasks, we use the suitability value function (Eq. 5.3). The latter is described in Section 5.5.

5.4 Constraints in smart camera networks

In this section, we will study the cost function part (Eq. 5.4) of the task assignment optimization problem (Eq. 5.6). As mentioned in Section 5.3, this cost function allows us to cope with the practical limitations of a camera network. Examples of these practical constraints are the limited computational power of the cameras, the bandwidth restrictions of the network, the maximally allowable number of active cameras etc.

In Section 5.4.1, we explain more concretely how the cost function (Eq. 5.4) is formulated. In particular, we study in detail one important constraint, namely the one imposed by the limited computational power of the cameras. In Section 5.4.2, we reformulate the cost function for this particular constraint in a formulation that is more suitable for this constraint. Finally, in Section 5.4.3, we will show how we can use the formulations of Sections 5.4.1 and 5.4.2 to optimize the operation of the network. As an example, we show how the frame rate of the system is optimized under the computational power constraint, by limiting the peak computational load of the cameras.

Note that other constraints than the limited computational power (or combinations of constraints) will not be treated in detail since they can be dealt with in a very similar way.

5.4.1 Cost function

A camera network can operate normally as long as its practical limits are not reached. E.g., if a camera of the network is charged with too many tasks, such that the necessary calculations cannot all be executed in real-time, this will introduce delays in the network operation, and might even lead to an operation failure of the whole network. Similar situations occur when the required network band width for all tasks exceeds the available bandwidth, or when the number of active cameras exceeds a practical limit which depends on e.g. the available number of transmission channels.

In order to allow the network to operate normally, it should be avoided that the camera network reaches its operational limits when assigning tasks to the cameras.

In this work, we focus on the computational power constraint. A camera can only perform a restricted number of operations in a certain time span. If all tasks are similar[2], we can assume they require the same number of operations O_{task} per second. Therefore, if we assume that all cameras have similar computational capabilities, the maximal number of concurrent tasks on a camera is

$$K' = \lfloor \frac{O_{\text{cam}}}{O_{\text{task}}} \rfloor \tag{5.7}$$

with O_{cam} the number of operations a camera can perform in one second.

Then, the cost $c(\mathbf{s}_n, n)$ of camera n is defined as

$$c(\mathbf{s}_n, n) = \begin{cases} 0, & |\mathbf{s}_n| \leq K' \\ +\infty, & |\mathbf{s}_n| > K' \end{cases} \tag{5.8}$$

and the cost function of Eq. 5.4 is then

$$C(\mathcal{S}) = \sum_{n=1}^{N} c(\mathbf{s}_n, n) \tag{5.9}$$

with $c(\mathbf{s}_n, n)$ as in Eq. 5.8.

A similar cost function can be drawn up for the case of a network with restricted communication resources. In this case, the network will only be able to make a certain number of camera queries (i.e. asking and receiving information from a camera for a certain task) in a certain time slot. Once this number is achieved, the cost for transmission becomes infinitely high. In the case of limited battery power, the cost function can be a function of the remaining battery power. Other constraints, such as a maximal number of active cameras, can be handled in a very similar way.

5.4.2 Practically admissible assignments

In the particular case of the cost function of Eqs. 5.8 and 5.9, $C(\mathcal{S})$ equals zero for admissible assignment sets $\mathcal{S}$, and equals infinity for non-admissible assignment sets $\mathcal{S}$. Hence, this constraint on computational load can be reformulated

[2] This is for example the case for the multiple person tracking scenario discussed in Section 5.7.

as limiting the admissible assignments. More specifically, if we denote by Γ' the restricted set of practically admissible assignments, the task assignment problem from Eq. 5.6 becomes

$$\mathcal{S}^* = \arg\max_{\mathcal{S}\in\Gamma'} W(\mathcal{S}) = \arg\max_{\mathcal{S}\in\Gamma'} V(\mathcal{S}). \tag{5.10}$$

Indeed, since $C(\mathcal{S})$ is zero for the admissible sets $\mathcal{S} \in \Gamma'$, $W(\mathcal{S})$ equals $V(\mathcal{S})$ for these sets. The non-admissible sets will never maximize the system welfare, as their system welfare equals $-\infty$. This is due to the infinite cost they incur and the fact that $V(\mathcal{S})$ will never equal infinity for a finite number of tasks. Therefore, one does not need to search for the optimal set among these non-admissible sets, and hence Γ of Eq. 5.6 can be reduced to Γ'.

In the case of limited computational power, we limit the maximal number of tasks that a camera can take care of, called K'. Then, the set Γ' of practically admissible assignments of Eq. 5.10 is the collection of all $\mathcal{S}$ for which $|\mathbf{s}_n| \leq K'$, $\forall n$:

$$\Gamma' = \{\, \mathcal{S} \mid |\mathbf{s}_n| \leq K',\, \forall n\} \tag{5.11}$$

This restricted set of assignments avoids computational overload of the network cameras. In the next section, we will show how the network can benefit from this limitation of the peak computational load of the cameras.

5.4.3 Limitation of peak computational load

A limitation of the peak computational load of the cameras is of major concern in smart camera networks. In particular, by minimizing the peak computational load, we can maximize the frame rate of the system. These topics will be explained in more detail in this section.

Let us consider a task k that uses observations from different cameras. Let us denote by t^k the computation time needed to compute the output data of task k for one set of images (consisting of one image from each camera of the camera set S^k, all the images in this set are taken at (approximately) the same time instant). This computation time t^k is determined by the computation time for the fusion of data from these cameras, t^k_{fuse}, and the *task bottleneck time* t^k_b. The latter is the maximum computation time needed by a camera (called *bottleneck node*) for the processing of a frame to determine the data needed to perform task k. To quantify this more specifically, we denote by t^k_n the computation time needed for camera n to process a frame in order to obtain the information related to task k. In other words, t^k_n is the time between the actual image sensing and the end time of the transformation of this image sensing data into task-related information available at some central point of processing. To keep the formulation general, we assume that the processing of each task is independent of the other tasks, and also that the time needed to communicate this task-related information to the central point of processing is negligible[3]. The bottleneck time t_b^k is then

$$t_b^k = \arg\max_{n \in S^k} t_n^k \tag{5.12}$$

and the total processing time for a task (per image set captured by camera set S^k) is

$$t^k = t_b^k + t_{\text{fuse}}^k. \tag{5.13}$$

Since the observation gathering on the cameras (often pixel-based operations) involves video processing, we assume that it is much more computationally expensive than the fusion process, which operates on higher-level and lower-volume data. Therefore, we assume $t_n^k >> t_{\text{fuse}}^k$ and hence we can approximate t^k by

$$t^k \approx t_b^k. \tag{5.14}$$

If a certain node is charged with too many tasks at the same time, it might become the task bottleneck node of a task k and increase the processing time for that task significantly.

If the peak computational load is limited, the task bottleneck time is reduced to a minimum. The frame rate f_k at which this task can be performed increases accordingly. To estimate the latter, we also need to consider the time needed by the task assignment algorithm, which we will denote by t_{TA}:

$$f^k = \frac{1}{t_b^k + t_{\text{TA}}}. \tag{5.15}$$

The maximally achievable frame rate f for the entire network charged with K tasks is

$$f = \arg\min_{k \in [1,K]} f^k. \tag{5.16}$$

5.5 Suitability value for smart camera networks

In this section, we will study the suitability value function (Eq. 5.3) of the task assignment optimization problem (Eq. 5.6). As mentioned in Section 5.3, this suitability value function allows us to monitor the quality of the executed tasks.

[3] Note that, as in Chapter 3, additional delays are possible due to networking effects. The study of these networking effects depends on the application and the network setup, and falls out of the scope of this PhD. To make abstraction of these effects, we assume in the remainder of this chapter that the network is a *perfect* network, and consequently, the delay introduced by networking effects on the transmission of the bits is set to 0.

In Section 5.5.1, we explain how we can quantify how well a vision task is performed by a set of cameras. For this, we use concepts from information theory, in the context of imprecise probability theory. This theory provides an extension to the classical probability theory and is able to explicitly represent the absence or incompleteness of information, which often occurs when performing vision tasks (e.g. when there is occlusion, or when something happens outside of the camera viewing frustum). More specifically, the Dempster-Shafer (DS) theory of evidence [Dempster, 1968, Shafer, 1976], which was introduced in Section 4.2.2, will be used for this purpose.

In Section 5.5.2, we will apply this theory to obtain the desired suitability value $v(S^k, k)$. This suitability value has been extensively studied in the PhD of Linda Tessens [Tessens, 2010] and will therefore only be discussed briefly in this dissertation. For more details, we refer the reader to the PhD of Linda Tessens [Tessens, 2010].

5.5.1 Quantification of task-related information

In a camera sensor network all tasks basically involve information gathering. The more information relevant to a task a camera set can acquire, the more suited it is to perform this task. Quantifying the task-related information contained in the observations of a camera set is thus a key issue in designing a value $v(S^k, k)$ which reflects the suitability of the set S^k for the task k.

In information theory, information is specified in terms of the entropy associated with a random variable. In this work we therefore define a camera network task more precisely as discovering the value of a realization of a random variable X using a subset of cameras. E.g. in the tracking example treated in Section 5.7, X designates within which range of ground positions the target is located.

In a camera network it frequently occurs that a sensor can only yield partial information or even no information at all about a task. This happens when all or part of the events relevant to the task are occluded or occur outside of the camera viewing frustum. In these cases classical probability theory has to resort to priors which can be difficult to obtain, and if badly modeled, introduce misleading information in the system.

Imprecise probability theory provides an extension of its classical counterpart and is able to explicitly represent the absence or incompleteness of information using lower and upper probabilities. A well known mathematical theory that implements the concept of imprecise probabilities through belief functions is the Dempster-Shafer theory of evidence [Dempster, 1968, Shafer, 1976], which was introduced in Section 4.2.2. In the next section, we explain how we use this theory to obtain the desired suitability value $v(S^k, k)$.

5.5.2 Generalized information-theoretic approach

The concepts from information theory as they were introduced for classical probability theory cannot be straightforwardly transferred to imprecise probability theory. To this end, generalized information theory was developed [Klir, 1991]. In generalized information theory, information is defined in terms of uncertainty reduction.

Uncertainty comprises several aspects: *probabilistic uncertainty* is generated by the randomness of a system, whereas *unspecificity* arises when there is evidence for a proposition that aggregates several elementary propositions, but no or little information about the elementary propositions individually. The latter can be mathematically expressed by the generalized Hartley (GH) measure [Abellan and Moral, 2000]: $GH(m) = \sum_{A \subseteq \Omega} m(A) \log_2 |A|$, where $|A|$ denotes the cardinality (number of elements) of the set A.

The generalization of Shannon (GS) entropy to characterize probabilistic uncertainty is defined through an *aggregated uncertainty*, AU, which unites both unspecificity and probabilistic uncertainty: $GS(m) = AU(m) - GH(m)$. To define the aggregated uncertainty present in a BBA m, we first define $\mathcal{D}$, a set of probability distribution functions $p(\omega)$ on the finite set Ω that are *consistent* with m, as follows [Klir and Wierman, 1999]:

$$\begin{aligned}\mathcal{D} = \{p(\omega) | \omega \in \Omega, p(\omega) \in [0,1], \sum_{\omega \in \Omega} p(\omega) = 1, \\ \sum_{B \subseteq A} m(B) \leq \sum_{\omega \in A} p(\omega) \text{ for all } A \subseteq \Omega\}. \quad (5.17)\end{aligned}$$

The aggregated uncertainty is defined as [Klir and Wierman, 1999]

$$AU(m) = \max_{p \in \mathcal{D}} \left[- \sum_{\omega \in \Omega} p(\omega) \log_2 p(\omega) \right]. \quad (5.18)$$

It is the maximal Shannon entropy within $\mathcal{D}$. An efficient algorithm for computing Eq. 5.18 is available in [Klir and Wierman, 1999].

In what follows we will use the aggregated uncertainty, which joins probabilistic uncertainty and unspecificity, to characterize the uncertainty in a BBA m.

Applying our definition of a network task k of Section 5.5.1 to the DS theory, we formulate each task as assessing the validity of a set of elementary propositions that form a frame of discernment Ω. Each possible camera set S^k gathers evidence about the propositions within the power set 2^Ω, leading to a BBA m_{S^k}. The smaller the aggregated uncertainty in m_{S^k}, the more informative the observations of the set and the better suited this set is for the task. Let $|\Omega|$ denote the number of elementary propositions in the frame of discernment Ω. The maximal possible aggregated uncertainty in a BBA m_{S^k} equals $\log_2 |\Omega|$. It is for example obtained when $m_{S^k}(\omega) = 1/|\Omega|, \forall\omega \in \Omega$. We define our camera set suitability value for task k as

$$v(S^k, k) = 1 - \frac{AU(m_{S^k})}{\log_2 |\Omega|}. \tag{5.19}$$

A camera set that is very suitable for task k will thus have a suitability value close to one, whereas unsuitable sets will have a value of zero. The value function of Eq. 5.3 is then

$$V(\mathcal{S}) = \sum_{k=1}^{K} v(S^k, k) \tag{5.20}$$

with $v(S^k, k)$ as in Eq. 5.19.

With the value function as in Eq. 5.20 and the cost function as in Eq. 5.9, the constrained optimization problem of Eq. 5.10 is

$$\mathcal{S}^* = \arg\max_{\mathcal{S}\in\Gamma'} \left[\sum_{k=1}^{K} v(S^k, k) \right] \tag{5.21}$$

with Γ' as in Eq. 5.11.

In Section 5.7 we solve this optimization problem for the case of task assignment in a network in which multiple persons are tracked. First we discuss how we can simplify the solution of the optimization problem of Eq. 5.10 or Eq. 5.21.

5.6 Solving the optimization problem

As there is only a discrete number of possible sets $\mathcal{S}$, Eq. 5.10 is a discrete constrained optimization problem. An exhaustive search over all possibilities, guarantees that the optimal solution of Eq. 5.10 is found. In a network of N cameras, 2^N camera subsets are possible for each task, which means Γ contains 2^{NK} possible assignments. Only assignments in $\Gamma' \subseteq \Gamma$ need to be evaluated. The nature of the imposed constraints will dictate the exact number of elements of Γ', but for many cameras and tasks, an exhaustive search quickly takes up more computation time than available.

Let us now consider the particular optimization problem of Eq. 5.21. In this case, each camera n (of a total of N cameras) can be maximally charged with K' (of a total of K) tasks. Then, the number of admissible subsets is $(\frac{K!}{K'!(K-K')!})^N$. An exhaustive search over all these possible assignments is intractable, even for quite small N and K'. For instance, for typical $N = 10$, $K = 10$ and $K' = 3$, the number of sets is 6.2 10^{20}.

To simplify the search for the optimal assignment, we discuss in the next section two greedy optimization approaches. In Section 5.6.2, we compare their complexity and performance.

5.6.1 Greedy heuristics

The first greedy heuristic is an extension of the fast polynomial, approximate coalition formation algorithm described in [Dang et al., 2006]. We have summarized the pseudo-code of this first proposed optimization heuristic in Algorithm 1 (see end of this chapter).

The goal of the algorithm is to find the task assignment set $\mathcal{S}^*$. In this algorithm, we start from a task assignment set $\mathcal{S}^*$, for which all the elements, denoted by $S^{*1}, \ldots, S^{*K}$, are empty sets. In a first step, we search for each camera $n \in \mathcal{C} = \{1, \ldots, N\}$ (line 10 of Algorithm 1) the K'' ($K'' \leq K'$) tasks this camera n should take care of such that the system welfare of Eq. 5.10 is maximized (lines 14-26 of Algorithm 1). This set of tasks is called the camera's *best state* and is denoted by $\mathbf{s}_n^* = [s_n^{*1}, \ldots, s_n^{*K}]$ with $|\mathbf{s}_n^*| = K'' \leq K'$. To find a camera's *best state* $\mathbf{s}_n^*$, let us introduce Γ_n, which is the set of all possible task assignment sets, in which camera n takes care of K'' (i.e., K' or fewer) tasks, and in which all the other cameras $\bar{n}$ ($\bar{n} \in \mathcal{C}\backslash\{n\}$) do not take care of any tasks, i.e.

$$\Gamma_n = \{\ \mathcal{S} \mid |\mathbf{s}_n| \leq K',\ \text{and}\ |\mathbf{s}_{\bar{n}}| = 0\ \forall \bar{n} \in \mathcal{C}\backslash\{n\}\} \tag{5.22}$$

Among all the sets $\mathcal{S} \in \Gamma_n$, we look for the set that maximizes the system welfare. This task assignment set is denoted by $\mathcal{S}_n$. For $\mathcal{S}_n$ holds

$$\mathcal{S}_n = \arg\max_{\mathcal{S} \in \Gamma_n} W(\mathcal{S}) \tag{5.23}$$

Camera n's best state $\mathbf{s}_n^*$ is the state that corresponds to the assignment $\mathcal{S}_n$. In other words, if we denote by $S_n^1, \ldots, S_n^K$ the elements of the set $\mathcal{S}_n$, then holds

$$\forall k : s_n^{*k} = \begin{cases} 1, & \text{if } S_n^k = \{n\} \\ 0, & \text{if } S_n^k = \emptyset \end{cases} \tag{5.24}$$

Note that for calculating a camera's best state in this step, only the camera itself (and not the other cameras) is taken into consideration. The camera among all N cameras that yields with its best state the highest system welfare is chosen. We denote this camera by n_1. For n_1 holds

$$n_1 = \arg\max_{n=1,\ldots,N} W(\mathcal{S}_n). \tag{5.25}$$

For those tasks k for which holds that $s_{n_1}^{*k} = 1$, camera n_1 is added to the subset S^{*k} of the final assignment $\mathcal{S}^*$ (line 33 of Algorithm 1). n_1 is the value of variable $n_{\max}$ the first time the lines 33-35 in Algorithm 1 are run through. Camera n_1 is then removed from $\mathcal{C}$ (line 35 in Algorithm 1).

In a second step, among all cameras $n \in \mathcal{C}$, i.e. all cameras except camera n_1 (line 10 of Algorithm 1), we look again for each camera's best state s_n^* (lines 14-26 of Algorithm 1). The difference with the first step, is that now for the calculation of a camera's best state not only the camera itself, but also the assignment of the previously selected camera n_1 is considered. Hence, a camera's best state are the K'' (i.e., K' or fewer) tasks this camera should take care of such that, considering that camera n_1 already takes care of the tasks for which $s_{n_1}^{*k} = 1$ (selected in the first step), the system welfare of Eq. 5.10 is maximized.

To find a camera's *best state* $\mathbf{s}_n^*$ in this step, let us again introduce Γ_n, which is the set of all possible task assignment sets, in which camera n takes care of K'' (i.e., K' or fewer) tasks, in which camera n_1 already takes care of the tasks for which $s_{n_1}^{*k} = 1$, and in which all the other cameras $\bar{n}$ ($\bar{n} \in \mathcal{C}\backslash\{n\}$, i.e. $\bar{n} \neq n \neq n_1$) do not take care of any tasks, i.e.

$$\Gamma_n = \{\, \mathcal{S} \mid |\mathbf{s}_n| \leq K', \mathbf{s}_{n_1} = \mathbf{s}_{n_1}^*, \text{ and } |\mathbf{s}_{\bar{n}}| = 0 \, \forall \bar{n} \in \mathcal{C}\backslash\{n\}\} \tag{5.26}$$

Among all the sets $\mathcal{S} \in \Gamma_n$, we look for the set that maximizes the system welfare. This task assignment set is denoted by $\mathcal{S}_n$. For $\mathcal{S}_n$ holds

$$\mathcal{S}_n = \arg\max_{\mathcal{S} \in \Gamma_n} W(\mathcal{S}) \tag{5.27}$$

Camera n's best state $\mathbf{s}_n^*$ is the state that corresponds to the assignment $\mathcal{S}_n$. In other words, if we denote by $S_n^1, \ldots, S_n^K$ the elements of the set $\mathcal{S}_n$, then holds

$$\forall k : s_n^{*k} = \begin{cases} 1, & \text{if } n \in S_n^k \\ 0, & \text{if } n \notin S_n^k \end{cases} \tag{5.28}$$

We select the camera n_2 that yields the highest system welfare with its best state. Hence, for n_2 holds

$$n_2 = \arg\max_{n=1,\ldots,N(n \neq n_1)} W(\mathcal{S}_n) \tag{5.29}$$

For those tasks k for which holds that $s_{n_2}^{*k} = 1$, camera n_2 is added to the subset S^{*k} of the final assignment $\mathcal{S}^*$ (line 33 of Algorithm 1). n_2 is the value of variable $n_{\max}$ the second time the lines 33-35 in Algorithm 1 are run through. Camera n_2 is then removed from $\mathcal{C}$ (line 35 in Algorithm 1).

In the next steps, we repeat this process: we determine for all cameras $n \in \mathcal{C}$ (line 10 of Algorithm 1) their best state considering the assignment of the previously selected cameras n_1, n_2, etc. These steps are repeated until we have assigned tasks to all cameras. The complexity of this algorithm will be discussed in Section 5.6.2.

To reduce the number of loops that need to be run through with Algorithm 1, we propose a second heuristic. The pseudo-code of this optimization heuristic is summarized in Algorithm 2 (see end of this chapter).

As for Algorithm 1, the goal of this algorithm is to find the task assignment set $\mathcal{S}^*$. In this algorithm, we also start from a task assignment set $\mathcal{S}^*$, for which all the elements, denoted by $S^{*1}, \ldots, S^{*K}$, are empty sets. In each step we randomly select one camera $n \in \mathcal{C}$ (line 8 of Algorithm 2). For this camera n, we find its best state s_n^* (lines 10-22 of Algorithm 2) in the same way as we did in Algorithm 1 (see Eqs. 5.23, 5.24, 5.27, and 5.28). In other words, we look for the K'' (i.e., K' or fewer) tasks this camera n should take care of such that the system welfare of Eq. 5.10 is maximized, taking the assignment of previously selected cameras into consideration (except in the first step where only the camera itself is considered, see Eqs. 5.23 and 5.24). The difference with the first heuristic is that we do not choose among all cameras the one that yields the highest system welfare (as in Eqs. 5.25 and 5.29). Instead, in this heuristic we randomly select in each step one of the cameras, and determine for this camera its best state (taking previous assignments into consideration, except in the first step). This simplification reduces the number of operations significantly at the cost of a reduction in task assignment quality. In Section 5.6.2 we will compare in detail the complexity of this algorithm with that of Algorithm 1. In Section 5.8, we will show that despite the lower complexity of Algorithm 2, its performance is comparable to that of Algorithm 1.

5.6.2 Complexity reduction

As explained at the beginning of this section, the number of admissible subsets when each camera n (of a total of N cameras) can be maximally charged with K' (of a total of K) tasks is $(\frac{K!}{K'!(K-K')!})^N$. Hence, to optimally assign tasks one needs to evaluate the argument of Eq. 5.10 for each of these admissible subsets. This means that the welfare needs to be evaluated $(\frac{K!}{K'!(K-K')!})^N$ times.

If we adopt one of the greedy approaches, the number of assignments that needs to be assessed is reduced. Each iteration step in both heuristics (i.e. lines 14-26 of Algorithm 1 and lines 10-22 of Algorithm 2) includes as computationally most expensive step the welfare calculation ($W(\mathcal{S}_{\text{task}})$ on line 19 for Algorithm 1 and on line 15 for Algorithm 2). If we adopt again the cost function as described in Section 5.4.1, this means that this welfare calculation is equal to the sum of the values of all tasks (see Eqs. 5.10 and 5.20). The other algorithm steps are negligible compared to this step. For Algorithm 1, the number of loop iterations and hence the number of welfare evaluations is $\sum_{j=0}^{N-1}\left[(N-j)\sum_{i=0}^{K'-1}(K-i)\right]$. For instance, for typical $N = 10$, $K = 10$ and $K' = 3$, the number of loop iterations is 1485 (as opposed to 6.2 10^{20} for the exhaustive search). For Algorithm 2, the number of loop iterations is $N \sum_{i=0}^{K'-1}(K-i)$. For typical $N = 10$, $K = 10$ and $K' = 3$, this means 270 iterations.

To assess the reduction in complexity of the greedy approaches of Algorithm 1 and Algorithm 2 compared to the exhaustive search, we define the complexity reduction factor (CRF) as the ratio of the number of iterations (each involving a welfare calculation) for the exhaustive search to the number of iterations (each involving a welfare calculation) for the greedy search. This will give a very good approximation of the real complexity reduction of the greedy approaches, since the welfare calculation is far more computationally expensive than the other steps in the greedy algorithms (mainly comparisons and assignments of values to variables). In the PhD of Linda Tessens [Tessens, 2010], one can find a detailed study of the number of operations needed for suitability value calculation (and by extension, welfare calculation, see Eq. 5.20) in the specific case of person tracking.

For Algorithm 1, the CRF is

$$\mathrm{CRF}_{\mathrm{Alg.1}} = \frac{\left(\frac{K!}{K'!(K-K')!}\right)^N}{\sum_{j=0}^{N-1}\left[(N-j)\sum_{i=0}^{K'-1}(K-i)\right]} \tag{5.30}$$

For Algorithm 2, the CRF is

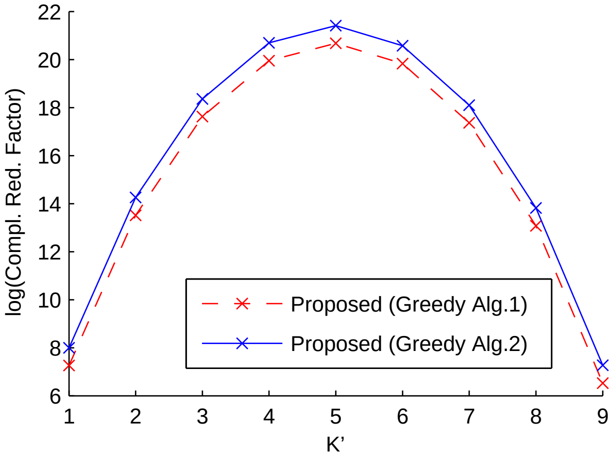
$$\mathrm{CRF}_{\mathrm{Alg.2}} = \frac{\left(\frac{K!}{K'!(K-K')!}\right)^N}{N\sum_{i=0}^{K'-1}(K-i)} \tag{5.31}$$

In Figure 5.1, we plotted the complexity reduction factor as a function of K' for both greedy approaches on a logarithmic scale for the typical values $N = 10$ and $K = 10$. We observe that the complexity is heavily reduced by the greedy approaches. Moreover, the greedy method of Algorithm 2 is a factor 5 to 6 less complex than the greedy method of Algorithm 1.

5.7 Application to task assignment for multiple object tracking

In this application example we consider a multi-camera system that observes a scene containing multiple persons. The goal of the system is to track the persons, i.e. to determine their position on the ground plane at each time instant. We consider the tracking of each person as a different network task. The final goal is to determine the best possible assignment of cameras to the multiple tracking tasks, under the computational constraints described in Section 5.4. We do this by

Figure 5.1 Complexity reduction factor for the greedy heuristics of Algorithm 1 and Algorithm 2 (Eqs. 5.30 and 5.31) as a function of the maximal number of tasks per camera (K'). The number of nodes $N = 10$ and the number of tasks $K = 10$.



solving Eq. 5.10 with the greedy heuristics of Algorithm 1 and 2 using the camera set suitability value of Eq. 5.19 (applied to a tracking task, see Section 5.7.2) and designing the admissible subsets through Eq. 5.11.

In this section, we first provide an overview of the state-of-the-art for task assignment in a tracking scenario. We compare this related literature with our task assignment approach. This is the subject of Section 5.7.1. After that, we explain how the suitability value can be designed for a tracking scenario (Section 5.7.2). Then, we comment on practical issues that need to be considered when applying our task assignment algorithm to a tracking application. In particular, we illustrate how we integrate the algorithm of [Munoz-Salinas et al., 2009] into our framework. Note that the subjects of Sections 5.7.2 and 5.7.3 are explained and tested in detail in the PhD of Linda Tessens [Tessens, 2010]. We discuss in this dissertation only these aspects that are beneficial for a good understanding of the remainder of this chapter.

5.7.1 Related work

Our approach differs from the existing literature in several ways. The task assignment methods of [Pahalawatta and Katsaggelos, 2004, Yao et al., 2010, Soro and Heinzelman, 2007] deal with limited computational and energy capacities with minor consideration for actual tracking accuracy in occluded and confusing environments (in contrast to our approach where we monitor the quality of the tracking through the suitability functions). In particular, [Yao et al., 2010] studies the uniform distribution of computational load among the cameras such that the number of dropped objects is minimized and the system frame rate maintained. In [Soro and Heinzelman, 2007], energy limitations of wireless camera nodes are integrated in the camera assignment in order to reduce the overall network power consumption. Pahalawatta et al. [Pahalawatta and Katsaggelos, 2004] select a limited number of active sensors nodes for target tracking in order to maximize the network lifetime.

Other approaches choose the observability of the targets as primary criterion under the assumption of unlimited computational power [Gupta et al., 2007, Denzler et al., 2003, Sommerlade and Reid, 2008, Snidaro et al., 2003, Ercan et al., 2006]. The authors of [Denzler et al., 2003] adopt an information-theoretic approach to control the focal length of a camera based on the uncertainty associated with the target position. This is done by minimizing the expected entropy of the state conditioned on the observation. In [Sommerlade and Reid, 2008] this approach is extended to account for the appearance of new targets, leading to an active scene exploration system. The authors in [Snidaro et al., 2003] base their view selection on a quality measure for the appearance of a tracking target in an image.

The methods of [Pahalawatta and Katsaggelos, 2004,Denzler et al., 2003,Sommerlade and Reid, 2008,Snidaro et al., 2003] cannot effectively take occlusion into account - a frequent problem in tracking - without significant reformulation of the algorithms. In [Gupta et al., 2007] cameras are selected especially to avoid occlusion (and confusion - people being visible behind the target) in a localization task. This is achieved by determining the probability of visibility of each part of a person model in each camera based on probabilistic estimates of the poses of other people in the scene. This determines the order in which the object positions and poses should be inferred. [Ercan et al., 2006] handles occlusion in a similar way, albeit in 2D, by weighting error contributions with the probability of occlusion, calculated from the prior of the occluding object. Furthermore an essentially geometric approach is followed to minimize the localization error of an object given its prior position distribution and the camera noise parameters.

We propose a task assignment method for static cameras that controls the quality of the multiple target tracking tasks, while simultaneously coping with computational constraints. Our approach links a generalized information-theoretic criterion for camera selection (similar to [Denzler et al., 2003, Sommerlade and Reid, 2008]) with taking the impact of occlusion and confusion of multiple targets on the localization into account (similar to [Gupta et al., 2007]). At the same time, the computational load is distributed uniformly among the cameras (as in [Yao et al., 2010]). The task assignment method is suited to be used in combination with a tracker based on particle filtering. Particle filters are powerful tools that can model multiple hypotheses, making them robust, and that can handle non-linear motion and noise models. As our tracking quality criterion is founded on the Dempster-Shafer theory of evidence, problems of absent or incomplete information (partial or complete invisibility due to limited fields of view, occlusions) are naturally handled.

5.7.2 Suitability value for tracking

In this section, we show how the suitability value of Section 5.5 is applied to a tracking task. Tracking a person basically involves gathering information about this person's location. As explained in Section 5.5.1, in information theory, information is specified in terms of the entropy associated with a random variable. A tracking task is therefore defined as discovering the value of a realization of a random variable X using the observations of a subset of cameras. More particularly, the random variable X designates within which range of ground positions the target is located.

To define these ranges, we divide the ground plane in the vicinity of the tracked person in $G-1$ discretization cells X_g, $1 \leq g \leq G-1$ (we will explain how in Section 5.7.3.3). There is also a part of the ground plane area in which we cannot gather observations (the area outside the viewing range of all cameras in the network), or in which we do not expect the tracked person to be. This part of the ground plane area makes up another cell X_G.[4]

The equivalent of this tracking task in the DS formulation of Section 5.5.2 is assessing the validity of the propositions in the frame of discernment Ω, where Ω is made up of elementary propositions that express the hypothesis that the position x of the tracked person lies in the discretization cell X_g. In other words, $\omega_g = \{x \in X_g\} \in \Omega$, $g = 1 \dots G$, where X_g designates a discretization cell corresponding to a range of ground positions. The inclusion of the cell X_G in Ω makes the set of hypotheses exhaustive.

For each camera set $S \in \Gamma'$ we extract a BBA m_{S^k} on this frame of discernment from the observations made by the cameras in the set as follows. The observations about a single cell X_g can provide direct evidence for only two hypotheses: the target is in this cell (ω_g) or it is not ($\Omega \backslash \omega_g$). Combining evidence from different cells using one of the combination rules of Section 4.2.2 allows us to draw indirect conclusions about some hypotheses for which no direct evidence can be gathered because, as explained in Section 4.2.2, applying these rules leads to a specialization of the basic belief (i.e. basic belief is redistributed over the subsets of each proposition). Indeed, if there is evidence supporting the hypothesis that the target is not in cell X_g and other evidence that it is not in $X_{g'}$, then the hypothesis that it

[4] Note that the discretization of the ground positions is only necessary in the proposed method to determine a suitable camera set to perform the tracking task. For the tracking as such, one of the many existing multi-camera multi-person tracking algorithms can be used, as will be discussed in Section 5.7.3.1. This tracking does not need to operate on discretized ground positions.

is in any of the other cells becomes more likely. To model this intuitively plausible evidence gathering process, we consider the assessment of the hypotheses in Ω based on the observations about a single cell X_g as a separate body of evidence, denoted as $m^g_{S^k}$.

Thus $m^g_{S^k}(A) = 0$ for all proper subsets of Ω except for ω_g and $\Omega \backslash \omega_g$. By the definition of a basic probability assignment then $m^g_{S^k}(\Omega) = 1 - m^g_{S^k}(\omega_g) - m^g_{S^k}(\Omega \backslash \omega_g)$.

The cell X_G never contains any particles because it is in the part of the ground plane area in which we do not gather observations, either because we cannot or because we do not expect the tracking target to be there. Because no direct evidence about the presence or absence of the target in X_G can be gathered $m^G_{S^k}(\Omega) = 1$ and $m^G_{S^k}(A) = 0, \forall A \subset \Omega$.

The body of evidence m_{S^k} is obtained by fusing the bodies of evidence $m^g_{S^k}$ from all cells. Then, the suitability value $v(S^k, k)$ is obtained through Eq. 5.19. The distinct pieces of evidence $m^g_{S^k}$ can be combined to obtain m_{S^k} using Dempster's rule of combination (Eq. 4.1). This is not possible if the evidence is not independent. Non-distinct pieces of evidence should be combined using the cautious conjunctive rule of [Denoeux, 2008]. Unfortunately, as mentioned in Section 4.2.2, the result of fusing distinct bodies of evidence with this rule is less informative than if Dempster's rule (Eq. 4.1) is used. It is therefore important to establish to what extent the possible dependence between the evidence sources of different cells actually manifests itself in a practical scenario. This is discussed and tested in the PhD of Linda Tessens [Tessens, 2010] and in [Tessens et al., 2011]. The conclusion that is drawn from this analysis, is that the dependence of evidence is considerable when S^k contains only one camera and it almost disappears as soon as S^k contains at least two cameras. This justifies using Dempster's rule (Eq. 4.1) for combining the bodies of evidences $m^g_{S^k}$ of all cells to obtain m_{S^k} if the evidence stems from at least two cameras. If S^k contains only one camera, the cautious conjunctive rule of [Denoeux, 2008] must be used.

5.7.3 Practical choices

In this section, we present some practical choices we make to apply our task assignment method to a network in which multiple persons are tracked. The goal is to efficiently distribute tasks among the cameras. These practical choices are particularly important for obtaining the suitability value. Since this suitability value was extensively discussed in the PhD of Linda Tessens [Tessens, 2010], we will not give a detailed analysis here, but only give an overview of the most important aspects.

In Section 5.7.3.1, we explain what multi-camera multi-person tracking is, and how an existing multi-camera multi-person tracking algorithm fits into our scheme.

In particular, we choose to integrate the algorithm of [Munoz-Salinas et al., 2009] into our framework to test the validity of our method. The basics of this method that are essential for the further understanding of this chapter, are also given. Subsequently, in Section 5.7.3.2, we show how m_{S^k} is obtained, which is needed for the calculation of $v(S^k, k)$ (Eq. 5.19). In Section 5.7.3.3, we comment on how we divide the ground plane in the vicinity of the tracked person in $G-1$ discretization cells X_g ($1 \leq g \leq G-1$). In Section 5.7.3.4, we introduce a final practical issue, which is the use of simulated observations to avoid costly data transmissions.

5.7.3.1 Multi-camera multi-person tracking

A multi-camera multi-person tracking system is able to determine the position on the ground plane of each person in the scene at each time instant. Many such systems have been proposed in literature, for example [Fleuret et al., 2008, Mittal and Davis, 2003]. The goal of this work is *not* to introduce a new multi-camera multi-person tracker, but to cleverly select cameras to track each person in the scene while taking the camera network constraints on computation into account. In this chapter, we have proposed such a task assignment method. The exact choice of the used multi-camera multi-person tracking system is therefore not a key issue in this work.

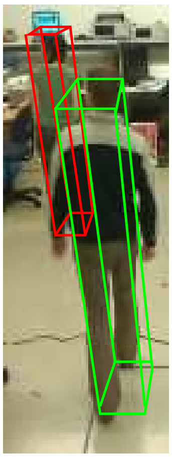
In this work we opt for the tracker in [Munoz-Salinas et al., 2009], which extends the Bayesian particle filter to the DS theory of evidence. It combines the strength of a classical particle filter to handle non-linear and non-Gaussian motion and error models with the power of the DS theory to elegantly model uncertainty and absence of knowledge without having to specify any priors or conditionals. This latter property is particularly advantageous in camera networks, where limited fields of view and occlusions frequently pose problems.

In the following, we briefly present the main aspects of the algorithm of [Munoz-Salinas et al., 2009]. Only the elements used in our task assignment method for tracking (in particular, for the calculation of the suitability value) are highlighted. For a comprehensive description of its operation, we refer the reader to [Munoz-Salinas et al., 2009].

Consider for each tracking target at time t a set of positions x_l, $1 \leq l \leq L$, on the ground plane of a 3D scene. These positions are called particles. For each particle the hypotheses that the tracked person is present at this position ($\{$*present*$\}$) or not ($\{\neg$*present*$\}$) are investigated. These two hypotheses constitute the frame of discernment Θ associated with this particle. To gather evidence, each camera n makes an observation and translates it for each particle into a body of evidence m_n^l.

By analyzing an image region R where a tracked person is expected to be observed when standing at the position x_l, evidence about the presence of the

Figure 5.2 Projection of the wire frame of two 3D models into a camera image.



tracked person at this position is collected. This image region R is derived by assuming that a 3D model of the person is standing at position x_l. The 3D model is a cuboid with ground plane centered at x_l and with the dimensions of an average adult (see Fig. 5.2 for two examples).

Observations that are considered evidence for the hypothesis that the tracked person is present at position x_l are

- the presence of pixels that are part of the foreground within the image region R;
- a distance between the center of mass of the image region R and of the detected foreground region within the image region R;
- a difference in color within the foreground within the image region R to the color histogram model of the tracked person kept by the camera.

For more details on the tracking algorithm, the reader is referred to [Munoz-Salinas et al., 2009].

5.7.3.2 Obtaining the BBA m_{S^k}

To gather evidence for the propositions in the frame of discernment $\Omega = \{\omega_g | 1 \leq g \leq G\}$, we have to perform observations and extract evidence from these. For the tracking algorithm we also need to collect observations. For reasons of efficiency, we make use of the observations performed for the tracking algorithm to extract evidence for the propositions in our frame of discernment Ω. We take

$$m^g_{S^k}(\omega_g) = \max_{\forall l \in [1,L] | x_l \in X_g} m^l_{S^k}(\{\textit{present}\}), \tag{5.32}$$

where $m^l_{S^k}$ is obtained by fusing the bodies of evidence m^l_n for all cameras in the considered set: $n \in S^k$. Eq. 5.32 expresses that the basic belief that the tracked person is in cell X_g equals the highest evidence of presence measured in the particles that lie in this cell. The quality of this approximation depends on the sampling density in the cell. The basic belief for the hypothesis that the target being tracked is not in cell X_g is defined as the minimal evidence of absence measured in any of the particles that lie in this cell:

$$m^g_{S^k}(\Omega \backslash \omega_g) = \min_{\forall l \in [1,L] | x_l \in X_g} m^l_{S^k}(\{\neg \textit{present}\}). \tag{5.33}$$

Note that this is equivalent with

$$m^g_{S^k}(\Omega \backslash \omega_g) = 1 - \max_{\forall l \in [1,L] | x_l \in X_g} [m^l_{S^k}(\{\textit{present}\}) + m^l_{S^k}(\Theta)].$$

This implies that when we have full information about all particles in the cell (i.e. $m^l_{S^k}(\Theta) = 0$ for all $l \in [1, L]$ for which $x_l \in X_g$), $m^g_{S^k}(\Omega \backslash \omega_g) = 1 - m^g_{S^k}(\omega_g)$. For example if there is no evidence that the target is in this cell because $m^l_{S^k}(\{\textit{present}\}) = 0$ for all particles in the cell, i.e. $m^g_{S^k}(\omega_g) = 0$, then we are sure that the target is not in this cell: $m^g_{S^k}(\Omega \backslash \omega_g) = 1$. If nothing is known about the presence or absence of the target at all particles in the cell (i.e. $m^l_{S^k}(\Theta) = 1$ for all $l \in [1, L]$ for which $x_l \in X_g$), $m^g_{S^k}(\Omega \backslash \omega_g) = 0$.

As discussed in the PhD of Linda Tessens [Tessens, 2010] and in [Tessens et al., 2011] it is shown that Dempster's rule can be safely used to combine the bodies of evidence $m^g_{S^k}$ from all cells if S^k contains at least two cameras. If S^k contains only one camera, the cautious conjunctive rule of [Denoeux, 2008] must be used.

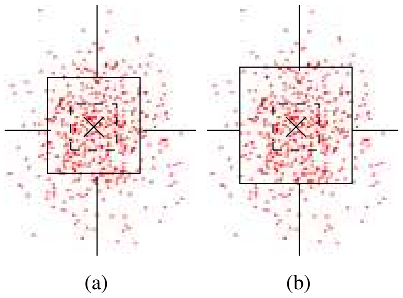
When we have obtained m_{S^k}, we can use Eq. 5.19 to calculate the suitability $v(S^k, k)$ of a camera set S^k for tracking target k (corresponding to task k).

5.7.3.3 Discretization scheme

A design choice which influences the suitability value of a camera set is the discretization scheme of the ground positions. This discretization is only used to assess if a set of cameras can make a sound estimate of the position of the tracking target and not for the tracking as such.

We assume the target is at its estimated position (or a prediction thereof, as

Figure 5.3 Discretization schemes of the ground positions. The dots are the particle positions, the cross indicates the estimated target position, the dashed line delineates the contour of the person 3D model centered at the estimated target position and the full lines indicate the discretization cell borders. (a) shows a possible division of the space around the center cell, which has the minimal allowed side length of twice the 3D model side length. In (b) the center cell is larger than this minimal allowed size.



will be explained in Section 5.7.3.4). Around this position we center a cell which we call the center cell. Its center is the estimated target position. It is shaped and sized such that the 3D model placed at the estimated target position is completely disjunct with a 3D model placed in any particle in another cell. Hence, the minimal allowed side length of the center cell is twice the 3D model side length. The rationale is that camera sets that localize the target in the center cell and also clearly observe that the target is not present in the other cells are very suitable to perform the tracking. Various possible divisions of the space around the center cell are proposed in the PhD of Linda Tessens [Tessens, 2010], and the influence of the scheme choice on the performance of the suitability value is studied.

Considering the reasoning and experimental results presented in the PhD of Linda Tessens [Tessens, 2010], we will use in this chapter a discretization scheme with a center cell with side length at least twice the 3D model side length and four other cells (see Fig. 5.3a or b) for our task assignment algorithm. For a detailed study on this subject, we refer the reader to the PhD of Linda Tessens [Tessens, 2010] and to [Tessens et al., 2011].

5.7.3.4 Avoiding costly data transmissions

Calculating the suitability of a camera set S_k as described in Section 5.5.2, requires observations from all cameras in the set to be collected at some point of central processing. Comparing the sum of suitability values of the sets S_k of the assignment $\mathcal{S} \in \Gamma'$ eventually requires observations from all cameras in the network to be collected at a central point. However, we wish to save camera and network resources by making and transmitting fewer observations.

To this end, we determine an assignment $\mathcal{S}^*$ (i.e. assign tracking tasks k to the cameras n) and the cameras only make and transmit observations about the $|\mathbf{s}_n|$ targets they are charged with. This assignment $\mathcal{S}^*$ is not based on observations of the current time instant. Instead, as will be explained below, it is based on *simulated* observations. The base station broadcasts the task assignment decision to all cameras. Only the cameras $n \in S_k$ actually make and transmit real observations about the tracking target k.

To make the camera selection decision at the base station, the camera images are not available, nor any of the observations of the current frame (in fact, no observations have been made yet, also not on the cameras). Of course, these images or observations could be transmitted by the cameras, but it is exactly these costly observations and transmissions that we wish to avoid.

Therefore, the task assignment decision is based on simulated observations of person models placed at *predicted* target positions. Alternatively, the observations of the previous time frame could be used to base the selection on. However, the observations for tracked object k are only available for the cameras that were selected at the previous time instant to perform this task k, since only these cameras have transmitted their observations about task k to the base station. To keep the input data of the task assignment algorithm homogeneous for all cameras, we prefer to use simulated observations.

To predict a tracking target position we assume that the target does not move appreciably between subsequent frames. The higher the frame rate and the lower the target's speed, the more reasonable this assumption is.

The task assignment decision is broadcast to all cameras and each camera makes and transmits real observations of the scene for its assigned task(s). Based on these observations, the tracking algorithm estimates the targets' current position. Based on this position, the task assignment decision for the following frame is calculated, and so on.

5.8 Results

In this section we discuss the performance of the task assignment method for tracking as proposed in Section 5.7.

5.8.1 Test data

We use natural video sequences recorded in two different environments for our evaluation.

The first environment is the one from the publicly available basketball dataset from the European project APIDIS [De Vleeschouwer and Delannay, 2009]. In these sequences a basketball court is observed by seven synchronized and calibrated cameras (see Fig. 5.7). The videos are captured at 25 fps and at a resolution of 800×600. The size of the field is $15m \times 28m$. There are on average 12 targets on the field. We have used the images recorded in the time interval 18:47 until 18:50 (4500 frames). As most cameras point to the left half of the court, only positions in that half are considered for the evaluation.

The second environment is an indoor scene of $5m$ by $4m$ observed by $N = 10$ web cameras. The camera views are shown in Fig. 5.9. Approximately 8 minutes of footage (2400 frames) in which two, three and four persons appear have been recorded at 5 frames per second and at CIF resolution (352×288). Only the starting points of these recordings have been synchronized.

Foreground detection is done using an algorithm based on mixture of Gaussians modeling [Stauffer and Grimson, 2000] with elementary shadow removal [Kaewtrakulpong and Bowden, 2001]. The size of the 3D model box is set to $0.5m \times 0.5m \times 1.7m$.

For the sequences of the second environment ground truth ground plane positions of the tracked persons have been generated for every fifth frame (1 s intervals). This has been done by manually checking the output of the multi-camera person detection algorithm of [Delannay et al., 2009] and correcting it where necessary. For the APIDIS sequence, ground truth target positions have been made available at 1 s intervals.

5.8.2 Evaluation metrics

For each frame for which ground truth target positions are available, we determine the root mean squared error (RMSE) of the estimated target positions with respect to the ground truth positions and average them over all tracked targets and all frames. We also count the number of times a tracker loses its target. The error of a lost person's position does not contribute to the average RMSE. After each loss the tracking is reinitialized at the correct position and tracking resumes.

A person is considered lost if none of the particles of its tracker is closer to the ground truth position than twice the maximal standard deviation σ_{prop} of the propagation of the particles, plus half the side length of the 3D person model box. The idea is that in this case the target is not likely to be recovered anymore by a propagation of the particles. In [Munoz-Salinas et al., 2009] the maximal $\sigma_{\text{prop}} = 2s/\text{fps}$, where s is the speed with which the targets are assumed to move and fps is the frame rate at which the system operates. In our second environment the frame rate is 5 fps and the speed is assumed 1m/s. In the APIDIS environment the frame rate is 25 fps and the moving speed of the basket ball players is assumed 5m/s. Both scenarios lead to $2\sigma_{\text{prop}} + \text{sidelength\_3Dbox}/2 = 1.05\text{m}$.

5.8.3 Tracking performance

Using the task assignment scheme described in Section 5.7, we now track persons using a varying maximum number of tracking tasks K' per camera. We use the discretization scheme of Fig. 5.3b and choose the number of particles $L = 50$. We compare the tracking performance of this method with the tracking performance when each camera is charged with K' tasks that remain fixed throughout the sequence. These fixed tasks have been chosen as the best performing ones among all possible fixed sets, but using common sense to reduce the number of sets. E.g. a person that is not visible during the whole sequence in a camera with close-up view, is discarded a priori as fixed task for this camera. We also compare with tracking using for each camera a set of K' tasks that is randomly chosen in each frame.

Numerical results for the first environment are shown in Fig. 5.4. Overall the proposed method outperforms the fixed and random task assignment schemes. The performance gain is larger for smaller sets. For larger sets all methods perform equally well. With our method, when each camera tracks only six out of twelve targets, the tracking performance is already very close to when tracking all targets with all cameras. The performance of the two greedy heuristics of Algorithm 1 and Algorithm 2 is very similar, despite the significantly lower complexity of Algorithm 2 (see Section 5.6 and Section 5.8.4).

In Fig. 5.5a, we have displayed a visual tracking result of the proposed method with $K' = 4$ by plotting the wire frame of the 3D model placed at the estimated target position in one of the top views of this environment. We observe that all targets are tracked well despite the fact that each cameras tracks only one third of the twelve targets. To compare our tracking results with those of the fixed assignment approach, we show the tracking results for the fixed assignment approach in Fig. 5.5b. We observe much poorer tracking performance for the fixed approach than for our approach: for this frame five tracking targets are lost or inaccurately tracked (i.e. the targets with red, medium aquamarine, purple, light pink and light blue wire frames) due to the inadequate task assignment.

The fixed task assignment method performs poorly because some targets are permanently assigned to cameras which have a close-up view or point at a specific part of the scene, and therefore often have a bad view on the assigned tracking target due to occlusions or limited fields of view. These problems are illustrated in Fig. 5.6, in which for two views at the same time instant as in Fig. 5.5 the targets assigned to this view by the fixed assignment approach are indicated with wire frames that have a different color per target.

In view Fig. 5.6a, one of the assigned targets (the one with the dark blue wire frame) is occluded, which hampers the tracking. In view Fig. 5.6b, the assigned object with the medium aquamarine wire frame is hardly visible in this view.

The detailed assignments of the proposed task assignment method with a max-

Figure 5.4 Number of target losses (upper panel) and average RMSE (lower panel) for different task assignment schemes as a function of the maximum number of tasks for each camera for the first environment.

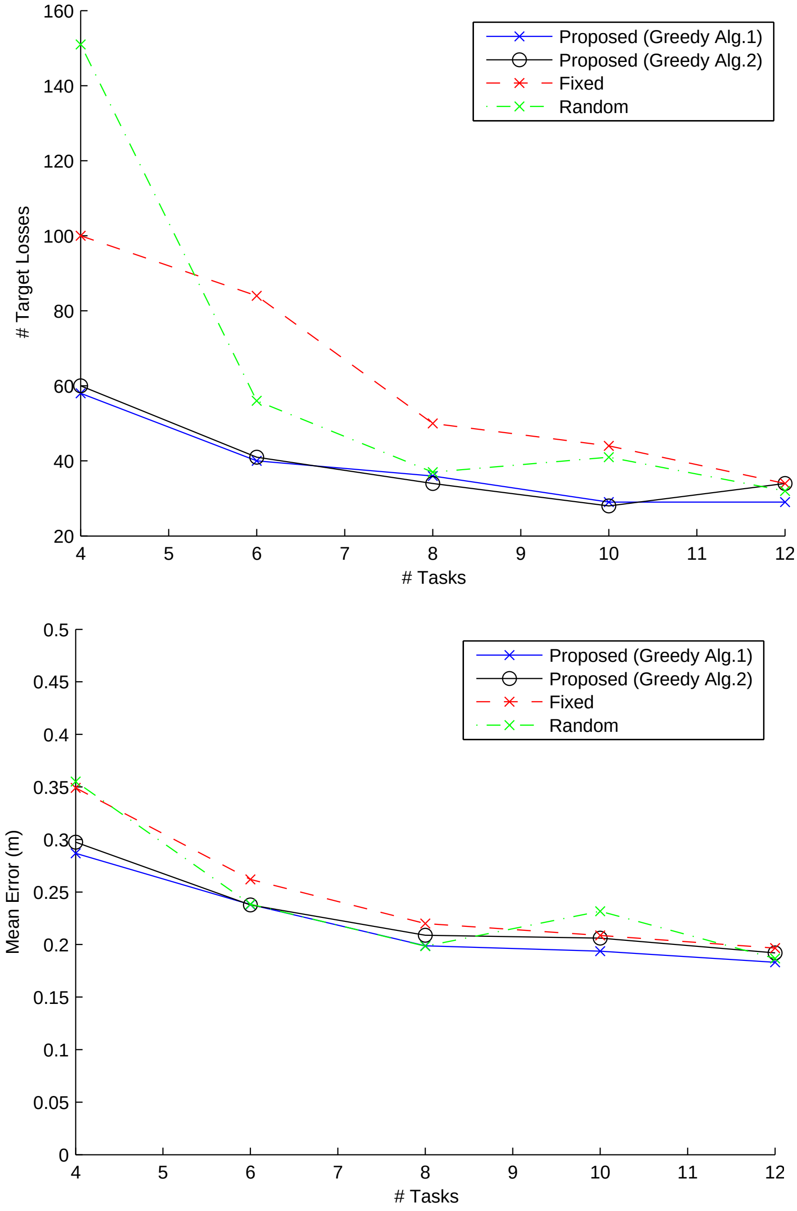
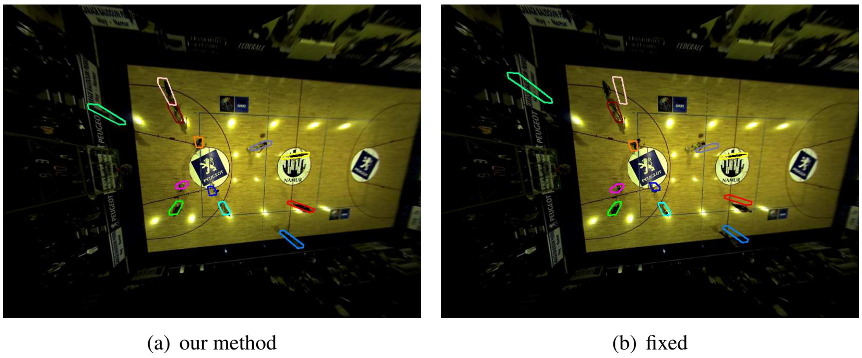


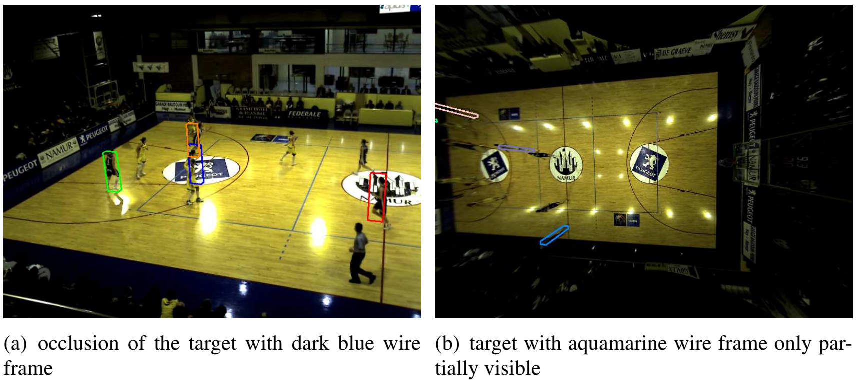
Figure 5.5 Tracking results for all targets shown plotted on one of the top views. Comparison between (a) our assignment method and the (b) fixed assignment approach.



(a) our method

(b) fixed

Figure 5.6 Two examples in the first environment of the problems of a fixed (or random) task assignment. The targets assigned to a particular view using the fixed assignment approach are indicated with wire frames that have a different color per target. The number of assigned targets per view is 4.



(a) occlusion of the target with dark blue wire frame

(b) target with aquamarine wire frame only partially visible imum number of tasks per camera $K' = 4$ are depicted in Fig. 5.7, again for the same time instant as in Fig. 5.5. In contrast with the fixed assignment method, our approach tracks the target with the medium aquamarine wire frame with the adequate views of Fig. 5.7a and e. A similar situation is observed for the person with the red and the light pink wire frame.

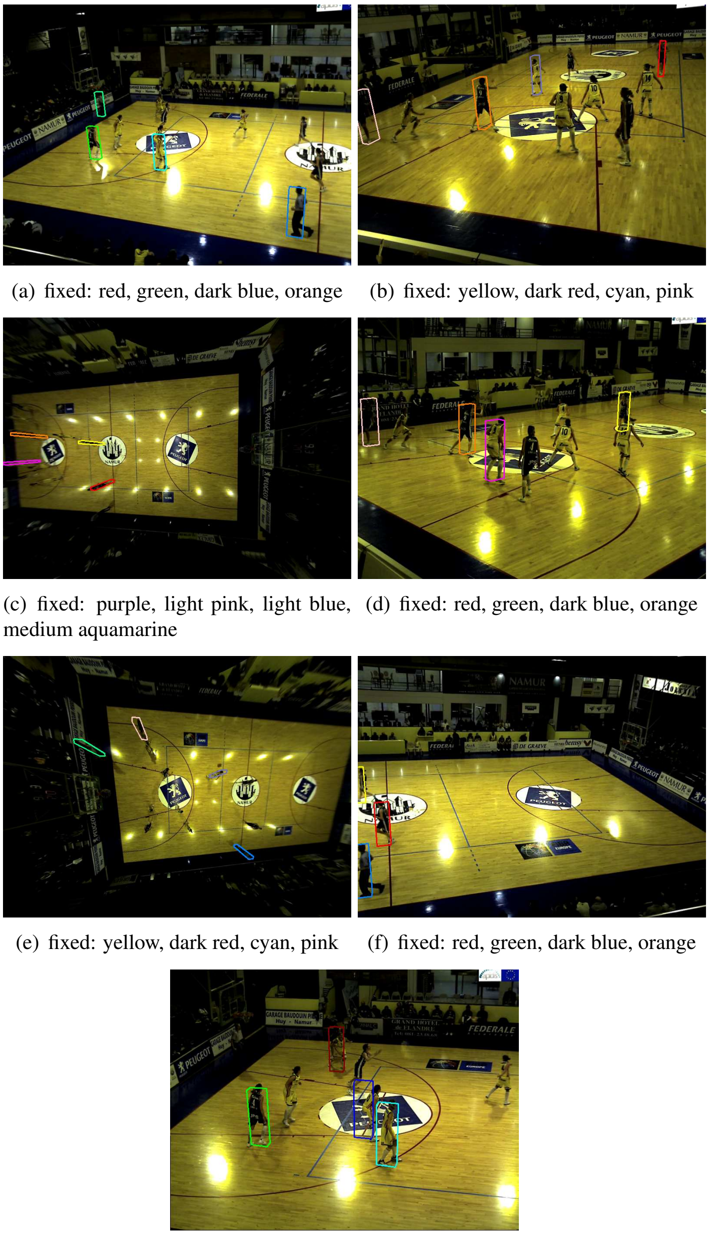
Targets assigned by the fixed assignment approach to cameras that have a nearly complete view of the court are generally tracked well. In particular, three cameras, the views of Fig. 5.7a, c and e perform generally best. Unfortunately, these views can only take care of a maximum of four tracking tasks each, and hence they should be involved with the targets that are not well visible in the other views. For example in our approach the view of Fig. 5.7e takes care of three targets that are not well visible in the other views. We can also observe that with our assignment method cameras get assigned targets that are not occluded in their view and are mostly close to the camera, in order to achieve good tracking performance. E.g., for the views of Fig. 5.7a and g, three of the four chosen targets are the targets that are closest to the camera and none of the targets is occluded. The view of Fig. 5.7f only gets assigned three targets instead of the maximum of four, as the assignment algorithm judges that it cannot provide any useful extra information for additional targets. Indeed, only three targets are visible in this view. In this way, computational power and time are saved.

With the random assignment method, similar problems occur as for the fixed assignment approach. Therefore, results are not shown here.

Numerical results for the second environment are shown in Fig. 5.8. We have plotted the results for the frames in which four persons appear. For the average RMSE the same conclusion as for the previous environment can be drawn, namely that the proposed method outperforms the others especially for lower numbers of assigned targets. For larger sets, the different methods display equal performance. The performance of the two greedy heuristics of Algorithm 1 and Algorithm 2 is again very similar. When each camera tracks only half of the targets, the tracking performance is already very close to when tracking all targets with all cameras. Note from Fig. 5.9 that the cameras in this setup have narrow viewing frustums. This increases the importance of dynamic task assignment, as it is not possible to select a fixed or random set with an overview of the scene.

In Fig. 5.9, a tracking result in this environment is visualized for tracking with a maximum of two tracking tasks per camera, assigned using the proposed method. The camera views in this environment are very diverse. Some cameras provide an overview of the scene, e.g. Fig. 5.9d and i, whereas some focus on a small part of it, e.g. Fig. 5.9a, b and f. A random assignment of cameras in such a setup often leads to poor tracking results. In the best performing fixed task assignment, we distribute tasks among the cameras such that each object is tracked by cameras that are geometrically spread over the whole scene.

Figure 5.7 Undistorted camera views in the first environment. The targets assigned to a camera view with the proposed assignment method are indicated with wire frames that have a different color per target. For comparison, the assigned targets in the fixed task assignment approach are indicated in the caption of each view by the color of their wire frame. The maximal number of assigned targets per view is four.



(a) fixed: red, green, dark blue, orange

(b) fixed: yellow, dark red, cyan, pink

(c) fixed: purple, light pink, light blue, medium aquamarine

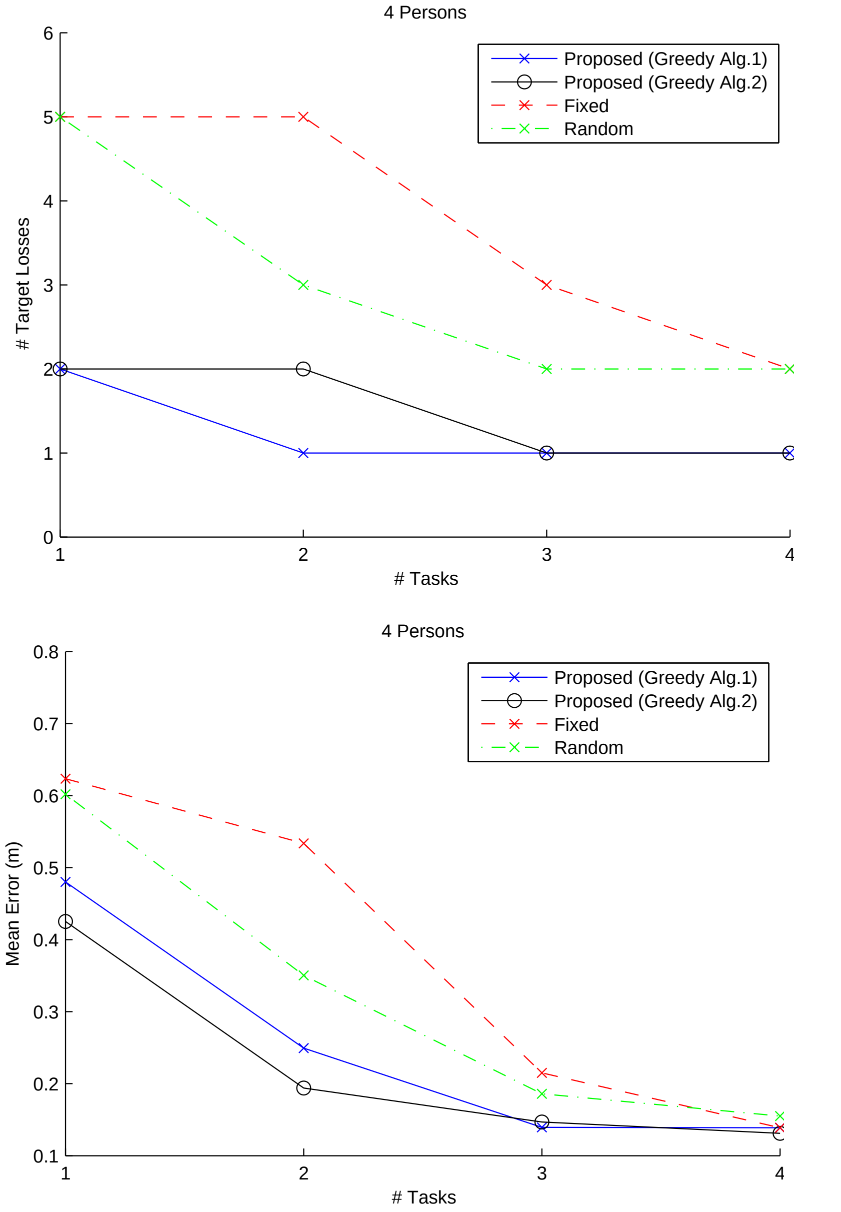
(d) fixed: red, green, dark blue, orange

(e) fixed: yellow, dark red, cyan, pink

(f) fixed: red, green, dark blue, orange

(g) fixed: purple, light pink, light blue, medium aquamarine

Figure 5.8 Number of target losses (upper panel) and average RMSE (lower panel) for different task assignment schemes as a function of the maximum number of tasks for each camera for the second environment.



The proposed method is more flexible. First, our method can handle occlusions. E.g. in the fixed assignment approach the view of Fig. 5.9i tracks the person with the blue shirt (orange wire frame), which is occluded by the person with the yellow shirt (blue wire frame). The same happens with the person with the brown shirt (red wire frame) in the view of Fig. 5.9j, which is occluded but unfortunately selected for tracking by this view in the fixed assignment approach. Second, our method can take advantage of close-up views. E.g., in the fixed assignment approach the person with the blue shirt (and the orange wire frame) is not tracked in the view of Fig. 5.9f, which displays a very good close-up view of this person. Third, for the fixed approach it regularly happens that targets outside the camera viewing frustum are assigned to it. E.g., the person with the white sweater (green wire frame) is supposed to be tracked by the view of Fig. 5.9a, but she is unfortunately not visible in this view. The same holds for the person in the yellow shirt (blue wire frame) which does not appear in the view of Fig. 5.9f. The latter problem occurs more often for the close-up views.

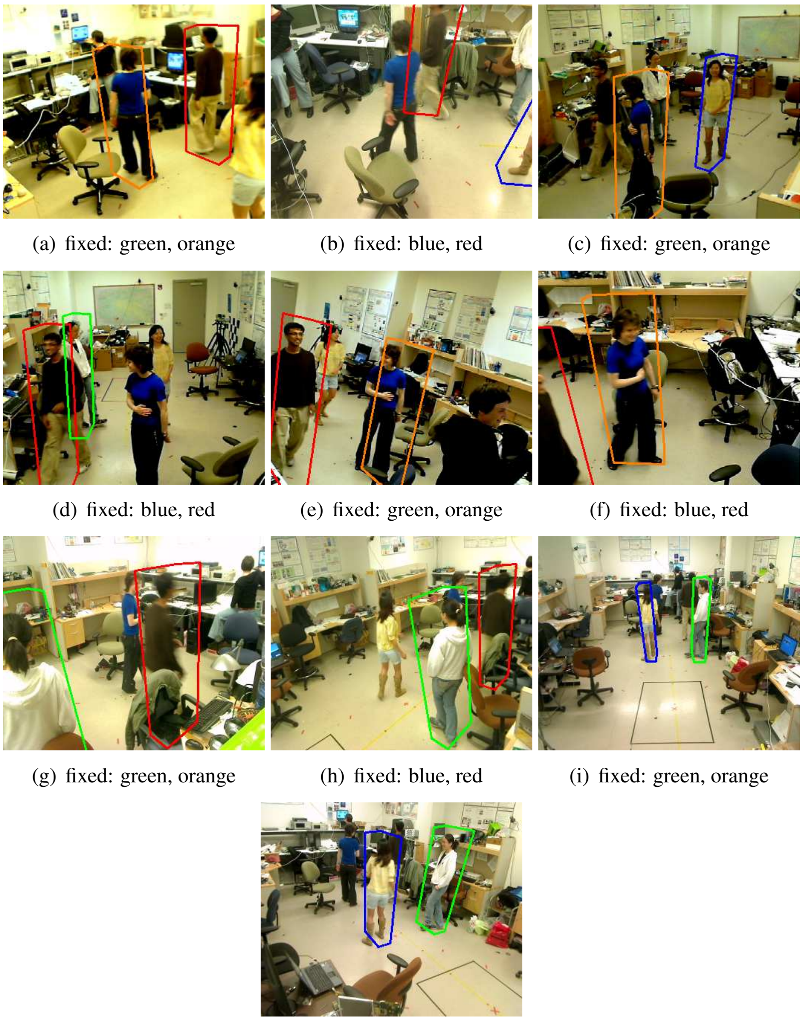
5.8.4 Computation time and frame rate

Since each camera only tracks K' out of K targets, the computational load on each camera is reduced by a factor K'/K.

To assess the influence of our task assignment algorithm on the system frame rate, we additionally need to consider the time needed for the task assignment algorithm (see Eq. 5.15). In this discussion, we assume that the task assignment algorithm is executed on a central base station. For each task k, the task assignment algorithm needs to determine for L particles in which of the G discretization cells they lie. Then, Eq. 5.10 is solved by one of the proposed greedy optimization algorithms (see Section 5.6). Each iteration step adds a camera n to the camera set S^k for task k. The major computation step is the update of $v(S^k, k)$ (to calculate the welfare of the corresponding task assignment $\mathcal{S}$, see Section 5.6). Note that this value is only computed if with the new S^k an admissible $\mathcal{S}$ is produced. The value calculation involves the following major steps:

1. fusing evidence of selected cameras for the selected task;
2. for G cells: obtaining the maximal evidence of presence and the minimal evidence of absence to construct m_S^g;
3. fusing the m_S^g from all G cells to obtain m_S;
4. determining the aggregated uncertainty in the obtained body of evidence m_S;
5. calculating the value for the selected task and the selected camera set as in Eq. 5.19 and adding it to the system welfare 5.10.

Figure 5.9 Camera views in the second environment. We show the task assignment and tracking results for the proposed method. For each view, the persons tracked by that view are indicated by a wire on the estimated target position. In the fixed task assignment scenario, camera views a, c, e, g, i track the persons indicated by the orange and green wire frame, and camera views b, d, f, h, j track the persons in the blue and red wire frame.



(a) fixed: green, orange

(b) fixed: blue, red

(c) fixed: green, orange

(d) fixed: blue, red

(e) fixed: green, orange

(f) fixed: blue, red

(g) fixed: green, orange

(h) fixed: blue, red

(i) fixed: green, orange

(j) fixed: blue, red

In a non-optimized implementation, the number of operations exponentially rises in steps 3 and 4 with the number of cells G because there are 2^G hypotheses in the power set of Ω. It is therefore important to keep G low (we advise that G is smaller than 10). The number of loops with value update that has to be executed depends on the chosen greedy optimization heuristic (see Section 5.6). Assume for example that each camera can be charged with a maximum of K' tasks. For typical parameters $G = 6$, $L = 50$, $N = 10$, $K = 4$ and $K' = 2$ (second environment), the task assignment algorithm took on average 489 ms during 10000 executions in a non-optimized c++ implementation on an Intel Core i7 920/2.67GHz processor for greedy Algorithm 1. For the speeded-up greedy heuristic of Algorithm 2, the task assignment took on average 112 ms under the same conditions. The theoretically expected reduction in complexity of the speeded-up heuristic is indeed around a factor 5, as was discussed in Section 5.6.2.

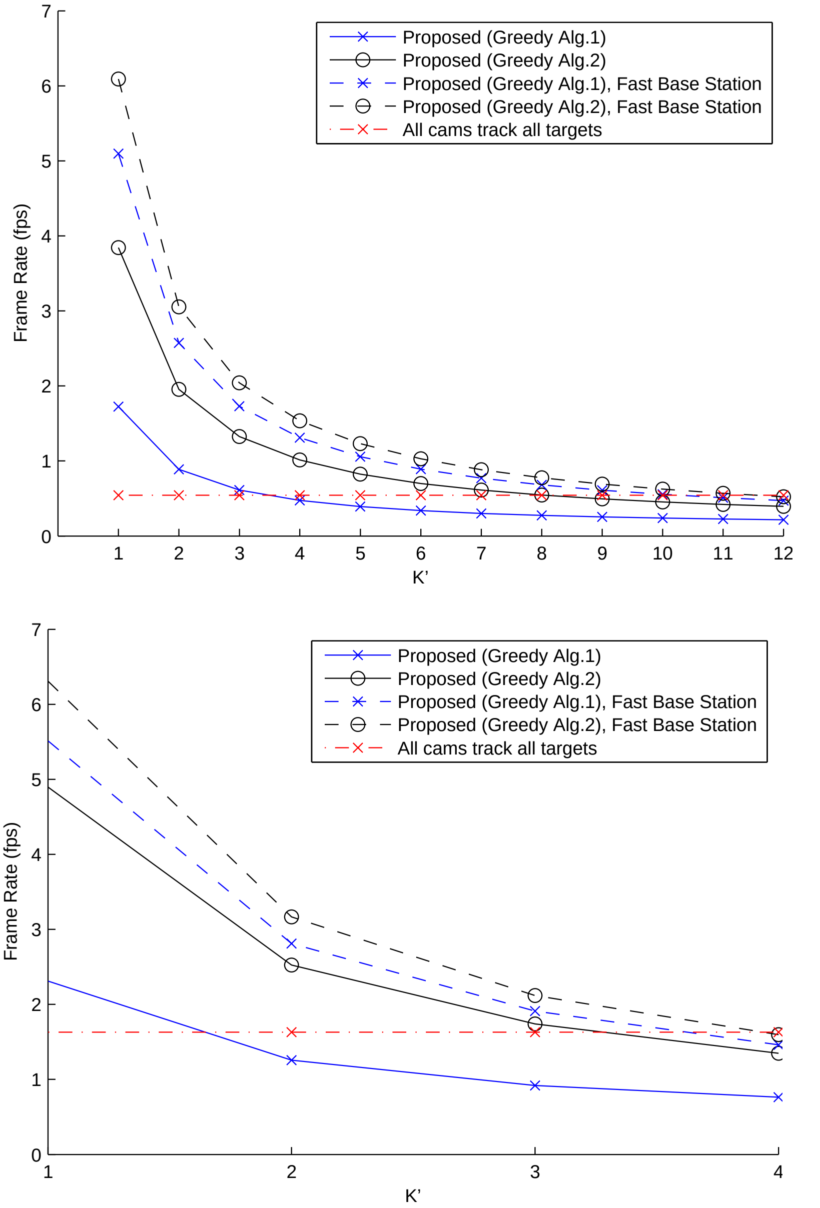
In Fig. 5.10, we show for both environments the achievable frame rate with our framework (full and dashed lines) as compared to the scenario in which all cameras track all targets (dash-dotted line). For the full lines, we assume that the processing speed of the cameras is equal to that of the central base station, which is usually not the case in practice. For this scenario, we observe that the achievable frame rate for the greedy approach of Algorithm 1 is smaller than for the scenario in which all cameras track all targets. The speeded-up heuristic of Algorithm 2 is in this case essential to avoid a frame drop. With this greedy heuristic we gain frame rate compared to the scenario in which all cameras track all targets as long as $K' < \frac{3K}{4}$. To make a fairer comparison with the scenario in which all cameras track all targets, we also plot the frame rate of our method if we assume that the processor which executes the task assignment algorithm is a factor 10 more powerful than the the CPU of the cameras (dashed lines). We base this assumption on typical processing power capacities of the present-day available smart camera networks. For this scenario, we gain frame rate for all values of K' and in particular when $K' < \frac{3K}{4}$. The frame rate gain for the greedy heuristic of Algorithm 1 is always a bit smaller than for the greedy heuristic of Algorithm 2, due to the higher complexity of its calculation of the task assignment.

5.9 Conclusion

An important challenge in smart camera networks charged with multiple tasks is achieving the best overall task performance, while distributing the tasks in an efficient way among the sensors according to the limited network resources. We have proposed a novel, general framework to do this in practical vision networks.

We have presented an optimization scheme for task assignment that aims at maximizing the system welfare. We have also discussed how the constraints in a

Figure 5.10 Average frame rate as a function of the number of assigned tasks for the proposed method for the first (upper panel) and second environment (lower panel).



network can be modeled in this framework. In particular, we have studied how to distribute the different network tasks among the cameras such that the peak load is minimized and the frame rate maximized. Another crucial component in a task assignment system is quantifying the contribution of one or more cameras to the accomplishment of a task. In this work, we have used a general set suitability value which evaluates the quality with which a subset of cameras accomplishes a network task. This set suitability value is derived from the Dempster-Shafer theory of evidence and can be applied to a wide range of vision problems. To solve the system welfare optimization problem, we have proposed two greedy approaches, which allow to reduce the complexity significantly compared to an exhaustive search.

As a proof of concept, we have applied our framework to task assignment in a camera network in which multiple targets are tracked. The method has been tested on thousands of frames in different environments. When assigning only half of the tracking tasks to each camera, the method allows to track persons with only a minor performance drop compared to tracking all targets with all cameras. Compared to the scenario in which all cameras are tasked for all tasks, we achieve frame rates that are up to two times as high, while maintaining similar tracking performance. The proposed method clearly outperforms other task assignment schemes for tracking.

The work on task assignment has led to one submitted journal paper [Tessens et al., 2011] and one journal paper that is currently in preparation for submission [Morbee et al., 2011]. Initial results have been published in one chapter of Lecture Notes of Computer Science [Lee et al., 2008], and at several international conferences [Morbee et al., 2009a, Morbee et al., 2008b, Tessens et al., 2008].

Algorithm 1 Greedy Optimization

Input: O (observations or simulated observations of all cameras), K' (maximal number of tasks per camera)

Output: $\mathcal{S}^*$ (sub-optimal assignment to perform the network tasks)

```
$S^k, S^{*k}, S^k_{\text{task}} \leftarrow \emptyset, k \in [1, K]$ (no cameras are selected for any task)
$\mathcal{S} = \{S^1, \ldots, S^k, \ldots, S^K\}$
$\mathcal{S}^* = \{S^{*1}, \ldots, S^{*k}, \ldots, S^{*K}\}$
$\mathcal{S}_{\text{task}} = \{S^1_{\text{task}}, \ldots, S^k_{\text{task}}, \ldots, S^K_{\text{task}}\}$
$\mathcal{C} \leftarrow \{1, \ldots, n, \ldots, N\}$ (set of cameras)
$W^* \leftarrow 0$ (total system welfare)
$W_{\max} \leftarrow 0$
$k_{\max} \leftarrow 1$
while $|\mathcal{C}| > 0$ do
    for $n \in \mathcal{C}$ do
        $\mathcal{S} \leftarrow \mathcal{S}^*$
        $W^{\text{cam}}_{\max} \leftarrow W^*$
        $\mathcal{K} \leftarrow \{1, \ldots, k, \ldots, K\}$ (set of tasks)
        while $|\mathcal{K}| > (K - K')$ and $k_{\max} > 0$ do (based on $O$, calculate $n$'s best state $\mathrm{s}^*_n$)
            $k_{\max} \leftarrow 0$
            for $k \in \mathcal{K}$ do
                $\mathcal{S}_{\text{task}} \leftarrow \mathcal{S}$
                $S^k_{\text{task}} \leftarrow S^k_{\text{task}} \cup \{n\}$
                if $W(\mathcal{S}_{\text{task}}) > W^{\text{cam}}_{\max}$ then
                    $W^{\text{cam}}_{\max} \leftarrow W(\mathcal{S}_{\text{task}})$
                    $k_{\max} \leftarrow k$
                end if
            end for
            $\mathcal{K} \leftarrow \mathcal{K} \backslash \{k_{\max}\}$
            $S^{k_{\max}} \leftarrow S^{k_{\max}} \cup \{n\}$
        end while (end of calculation of $n$'s best state $\mathrm{s}^*_n$)
        if $W^{\text{cam}}_{\max} > W_{\max}$ then
            $n_{\max} \leftarrow n$
            $W_{\max} \leftarrow W^{\text{cam}}_{\max}$
            $\mathcal{S}_{\max} \leftarrow \mathcal{S}$
        end if
    end for
    $\mathcal{S}^* \leftarrow \mathcal{S}_{\max}$
    $W^* \leftarrow W_{\max}$
    $\mathcal{C} \leftarrow \mathcal{C} \backslash \{n_{\max}\}$
end while
```

Algorithm 2 Speeded-up greedy Optimization

Input: O (observations or simulated observations of all cameras), K' (maximal number of tasks per camera)

Output: $\mathcal{S}^*$ (the sub-optimal assignment to perform the network tasks)

```
$S^{*k}, S^k_{\text{task}} \leftarrow \emptyset, k \in [1, K]$ (no cameras are selected for any task)
$\mathcal{S}^* = \{S^{*1}, \ldots, S^{*k}, \ldots, S^{*K}\}$
$\mathcal{S}_{\text{task}} = \{S^1_{\text{task}}, \ldots, S^k_{\text{task}}, \ldots, S^K_{\text{task}}\}$
$\mathcal{C} \leftarrow \{1, \ldots, n, \ldots, N\}$ (set of cameras)
$W^* \leftarrow 0$ (total system welfare)
$k_{\max} \leftarrow 1$
while $|\mathcal{C}| > 0$ do
    randomly select $n \in \mathcal{C}$
    $\mathcal{K} \leftarrow \{1, \ldots, k, \ldots, K\}$ (set of tasks)
    while $|\mathcal{K}| > (K - K')$ and $k_{\max} > 0$ do (based on $O$, calculate $n$'s best state $\mathbf{s}^*_n$)
        $k_{\max} \leftarrow 0$
        for $k \in \mathcal{K}$ do
            $\mathcal{S}_{\text{task}} \leftarrow \mathcal{S}$
            $S^k_{\text{task}} \leftarrow S^k_{\text{task}} \cup \{n\}$
            if $W(\mathcal{S}_{\text{task}}) > W^*$ then
                $W^* \leftarrow W(\mathcal{S}_{\text{task}})$
                $k_{\max} \leftarrow k$
            end if
        end for
        $\mathcal{K} \leftarrow \mathcal{K} \backslash \{k_{\max}\}$
        $S^{*k_{\max}} \leftarrow S^{*k_{\max}} \cup \{n\}$
    end while (end of calculation of $n$'s best state $\mathbf{s}^*_n$)
    $\mathcal{C} \leftarrow \mathcal{C} \backslash \{n\}$
end while
```

6
Overall Conclusion

In the past chapters, we have made a thorough study of a series of intelligent vision systems, that process sensor data in a way that is appropriate for the application, and that take practical constraints into account.

The main conclusions of this PhD are summarized in Section 6.1. The scientific output of this PhD is summed up in Section 6.2. In Section 6.3, we comment on possible areas for future research.

6.1 Conclusions

6.1.1 Modulo-PCM based coding algorithm for very low complexity coding of images

As explained in Chapter 2, this algorithm is useful in applications that need to preserve some of the desirable properties of PCM coding such as direct processing, random access and rate scalability. Our coding scheme combines three well-known simple coding techniques: PCM, binning and interpolative coding. The encoder first analyzes the signal statistics in a very simple way. Then, based on these signal statistics, the encoder simply discards a number of bits of each sample. The decoder recovers the discarded bits by using the received bits and side information generated from previously decoded samples. This algorithm is especially appropriate for image coding since it introduces larger coding errors in those regions where it is less visible. Experimental results obtained in the encoding of several digital images showed that this algorithm has a better objective and subjective performance than PCM at low rates. At high rates, our Modulo-PCM method and PCM provide similar results. Other source coding techniques, such as modulo-PCM coding with side information and pixel-domain Wyner-Ziv coding, perform slightly better in terms of rate-distortion, but at the expense of a significant increase in encoder and decoder complexity.

6.1.2 Thorough study and improvement of pixel-domain distributed video coding algorithms

In contrast to conventional video coding, distributed video coders perform simple intra-frame *encoding* and complex inter-frame *decoding*. This feature makes this type of coding suitable for applications that require low-complexity encoders.

In Chapter 3, we first developed a model of the coding distortion introduced by pixel-domain distributed video coders. Our distortion model can be used to determine the value of coding parameters under certain coding constraints. In particular, we showed how our model can be used to select the quantization step size of each video frame so that a target distortion limit can be approximately met. Experimental results showed that, even though the accuracy of the distortion predictions is limited by the restricted computational capacity of distributed encoders, the described distortion constraints can be approximately fulfilled by using our model.

Second, to allocate a proper number of bits to each frame, most distributed video coding algorithms use a feedback channel, which allows the decoder to request additional bits when needed. However, in some cases, a feedback channel does not exist. We therefore proposed a rate allocation algorithm for pixel-domain distributed video coders without feedback channel. Our algorithm estimates at the encoder the number of bits for every frame without significantly increasing the encoder complexity. Compared to the pixel-domain distributed video coder with feedback channel, the pixel-domain distributed video coder without feedback channel has only a small loss in rate-distortion performance, especially at low rates.

Third, we focused on the pixel-domain distributed video coder with feedback channel. We utilized this feedback channel to improve the rate allocation and to achieve very near-to-optimal rate allocation while at the same time eliminating the main feedback channel inconveniences, i.e., its negative impact on latency and decoder complexity. The method estimates at the encoder the number of bits needed for the decoding of every frame while still keeping the encoder complexity low. Experimental results showed that, by using our algorithm, the number of bit requests over the feedback channel - and hence, the decoder complexity and the latency - are significantly reduced. Meanwhile, a very near-to-optimal rate-distortion performance is maintained.

6.1.3 Novel vision systems for calculating accurate 2D occupancy maps

Two novel systems for 2D occupancy sensing were treated in Chapter 4. A first system consists of a set of calibrated and synchronized cameras. Each camera calculates a foreground (FG)/background (BG) silhouette and transfers this silhouette to a reference plane using its camera image-floor homographies. We proposed Dempster-Shafer based fusion of the ground occupancies computed from each view. This method yields very accurate occupancy detection results and in terms of concentration of the occupancy evidence around ground truth person positions it outperforms the state-of-the-art probabilistic occupancy map method and fusion by summing.

A second system uses instead of cameras a more specific device consisting of a linear array of optical sensing elements, called a line sensor. We proposed to use the line sensor together with a light-integrating optical system, which ensures that each sensing element integrates all light within a certain range of incidence angles. The data coming from multiple light-integrating line sensors is used for accurate 2D occupancy calculation algorithm. To this end, we first developed FG/BG subtraction algorithms for scan lines, that determine the probability that a pixel of the line sensor is a foreground pixel. Using the reprojections of these scan lines to the 2D scene map and our Dempster-Shafer occupancy data fusion method, we obtain a 2D occupancy map. In terms of concentration of the occupancy evidence around ground truth person positions, the results are quite close to the results obtained with multiple cameras, especially for setups where the line sensors view the scene from aside and not from above. Additional advantages of the line sensor compared to other types of 2D occupancy sensing systems such as multiple cameras, PIR sensors, radar/radio beacons, pressure carpets etc, are its low price, its low power consumption, its high data rates, its high bit depth (and hence, high accuracy) and its privacy-friendly nature.

6.1.4 Task assignment framework for intelligent vision networks

In smart camera networks with overlapping fields of views, intelligent multi-sensor management or task assignment is an important tool to limit data redundancy and to save computational and communication resources without discarding useful information. In Chapter 5, we presented a novel, general framework to quantify the quality with which a subset of cameras accomplishes a network task. The proposed set suitability value is derived from the Dempster-Shafer theory of evidence and can be applied to a wide range of vision problems. This suitability value allows us to appropriately assign tasks to the cameras in the network under computational and communication constraints.

As a proof of concept, we use the task assignment framework in camera networks in which multiple targets are tracked. With the proposed task assignment method, we can dynamically assign cameras to each tracking target under peak load and bandwidth constraints. Experimental results showed that the targets are tracked in difficult circumstances of occlusions and limited fields of view with as little as three targets per camera or with a maximum of four active cameras with the same accuracy as when using seven, eight or ten cameras. The proposed method clearly outperforms other camera task assignment schemes for tracking in terms of average position error and number of target losses.

6.2 Contributions

The research during this PhD resulted in a number of scientific contributions which we have summed up in the section.

The work on modulo-PCM based coding for very low complexity coding of images has led to one journal publication, that is currently under review [Prades-Nebot et al., 2010].

The research work on distributed video coding was published as one journal article [Morbee et al., 2008a], one chapter in Lecture Notes on Computer Science [Morbee et al., 2007a], and several international conference publications [Roca et al., 2008, Roca et al., 2007, Morbee et al., 2007d, Morbee et al., 2007c, Morbee et al., 2007b, Morbee et al., 2006a].

The research on 2D occupancy sensing has been published in one journal paper [Morbee et al., 2010]. Two patent applications about this work have been filed [Morbee and Tessens, 2010, Morbee and Tessens, 2011]. Initial results have been published in one chapter of Lecture Notes of Computer Science [Lee et al., 2008], and at several international conferences [Tessens et al., 2009, Morbee et al., 2008b, Tessens et al., 2008]. Furthermore, we developed a real-time 2D occupancy demonstrator that shows the applicability and accuracy of our method in real-life applications [Tessens and Morbee, 2010].

The work on task assignment has led to one submitted journal paper [Tessens et al., 2011] and one journal paper that is currently in preparation for submission [Morbee et al., 2011]. Initial results have been published in one chapter of

Lecture Notes of Computer Science [Lee et al., 2008], and at several international conferences [Morbee et al., 2009a, Morbee et al., 2008b, Tessens et al., 2008].

6.3 Future work

The most notable areas with potential for future research are the following.

- In Chapter 2: improving the assignment table for the coding parameters, by encoding a larger set of images and studying how the PSNR varies as a function of R and $\mathrm{PSNR}_{\mathrm{SI}}$.
- In Chapter 4 and 5: improving the robustness of the methods against calibration inaccuracy, synchronization errors, and failure of the foreground detection methods.
- In Chapter 4: including temporal information to improve the occupancy detection accuracy, and to overcome the robustness problem mentioned under point 2.
- In Chapter 4: the occupancy sensing algorithm could be improved by including an algorithm for detecting the position of the persons in the scene. The positions of the detected persons will define occluded zones for each camera n. Then, we can take this additional occlusion information into account in the calculation of the occupancy map $m_n(\{occ\})$ by setting $m_n(\theta_{\mathbf{x}}) = 1$ for all occluded cells. In a similar way, other (possibly permanent) occluders, such as furniture in the scene or objects blocking part of the view of cameras (e.g. cables), can be taken into account as well.
- In Chapter 4: building a light-integrating line sensor network, and testing it possibilities in real-life environments. Testing the applicability of the modulo-PCM algorithm of Chapter 2 to the high data rate output of the light-integrating line sensors.
- In Chapter 4: implementing scan line foreground detection algorithms in hardware and testing the algorithms in connection with a working prototype of a light-integrating line sensor.
- In Chapter 5: integrating various types of vision tasks in the task assignment framework and testing its performance in different camera network setups.

A

Appendices

A.1 Derivation of (2.16)

The term $D_{u,i,i}$ is

$$D_{u,i,i} = \frac{1}{\Delta} \int_{\mathcal{I}_{u,i}} \int_{\mathcal{L}_{u,i}} (x - \hat{x})^2 f_{Y|X}(y|x)\, dy\, dx. \tag{A.1}$$

If $m = 0$, there is only one decision interval $\mathcal{L}_{u,0} = (-\infty, \infty)$. If $m = 1$, each set $\mathcal{M}_u$ has two associated intervals $\mathcal{L}_{u,0} = (-\infty, t_u)$ and $\mathcal{L}_{u,1} = (t_u, \infty)$ where t_u is the value of y, which is equidistant to $\mathcal{I}_{u,0}$ and $\mathcal{I}_{u,1}$. If $m > 1$, all the decision intervals of set $\mathcal{M}_u$ have length d except for the first and the last ones, which are of the form $\mathcal{L}_{u,0} = (-\infty, t_{u,0})$ and $\mathcal{L}_{u,2^m-1} = (t_{u,2^m-1}, \infty)$. In order to obtain a simple expression for the solution to (A.1), we assume that all intervals $\mathcal{L}_{u,i}$ have the same length d, irrespectively of u and i. Hence,

$$D_{u,i,i} \approx \frac{1}{\Delta} \int_{c_{u,i}-\Delta/2}^{c_{u,i}+\Delta/2} \int_{c_{u,i}-d/2}^{c_{u,i}+d/2} (x - \hat{x})^2 f_{Y|X}(y|x)\, dy\, dx \tag{A.2}$$

where $c_{u,i}$ is the midpoint of $\mathcal{I}_{u,i}$. If we substitute

$$\hat{x} = \begin{cases} c_{u,i} - \frac{\Delta}{2}, & c_{u,i} - \frac{d}{2} < y < c_{u,i} - \frac{\Delta}{2} \\ y, & c_{u,i} - \frac{\Delta}{2} < y < c_{u,i} + \frac{\Delta}{2} \\ c_{u,i} + \frac{\Delta}{2}, & c_{u,i} + \frac{\Delta}{2} < y < c_{u,i} + \frac{d}{2} \end{cases} \tag{A.3}$$

and $f_{Y|X}(y|x) = \frac{\alpha}{2}\, e^{-\alpha|x-y|}$ into (A.2) and solve the integrals, we obtain

$$\begin{aligned} D_{u,i,i} &\approx \frac{2}{\alpha^2}\left(1 + e^{-\alpha\Delta}\right) + \frac{4}{\alpha^3}\left(e^{-\alpha\Delta} - 1\right) \\ &+ \frac{e^{-\alpha d/2}}{\Delta\alpha^3}\left[e^{-\alpha\Delta/2}(\alpha^2\Delta^2 + 2\alpha\Delta) - 4\sinh(\alpha\Delta/2)\right]. \end{aligned} \tag{A.4}$$

For high enough α values (i.e. for high enough accurate SI), we can neglect the third term of (A.4) with respect to the first two terms, and we have

$$D_{u,i,i} \approx \frac{2}{\alpha^2}\left(1 + e^{-\alpha\Delta}\right) + \frac{4}{\alpha^3}\left(e^{-\alpha\Delta} - 1\right). \tag{A.5}$$

This is the expression we would have derived if we had initially assumed that $\mathcal{L}_{u,i,} = (-\infty, \infty)$, which is true when $m = 0$ (i.e., when there are not any decision errors).

Note that in Section 3.3.1 the same expression is obtained for the distortion of pixel-domain Wyner-Ziv coders where it is assumed that the number of transmitted bits is high enough to completely avoid decision errors.

A.2 Derivation of (2.17)

The term $D_{u,i,j}$ is

$$D_{u,i,j} = \frac{1}{\Delta}\int_{x\in\mathcal{I}_{u,i}}\int_{y\in\mathcal{L}_{u,j}} (x - \hat{x})^2 f_{Y|X}(y|x)\, dy\, dx. \tag{A.6}$$

Although the length of $\mathcal{L}_{u,j}$ depends on j (Appendix A.1), in order to obtain a simple expression for the solution to (A.6), we assume that $\mathcal{L}_{u,j}$ is an interval of length d centered at c_j; hence

$$D_{u,i,j} = \frac{1}{\Delta}\int_{c_i - \frac{\Delta}{2}}^{c_j + \frac{\Delta}{2}}\int_{c_j - \frac{d}{2}}^{c_j + \frac{d}{2}} (x - \hat{x})^2 f_{Y|X}(y|x)\, dy\, dx \tag{A.7}$$

where c_i is the midpoint of $\mathcal{I}_{u,i}$ and $c_j = c_i + (j - i)d$. To further simplify the solution to (A.7), we assume that $\hat{x} \approx c_j$ and that the distortion for every x in $\mathcal{I}_{u,i}$

is approximately equal to the distortion when $x = c_i$. With these assumptions, we can rewrite (A.7) as

$$D_{u,i,j} \approx \frac{1}{\Delta} \int_{c_i - \Delta/2}^{c_i + \Delta/2} \int_{c_j - d/2}^{c_j + d/2} (c_i - c_j)^2 f_{Y|X}(y|c_i) \, dy \, dx \qquad \text{(A.8)}$$

and, substituting $f_{Y|X}(y|c_i) = \frac{\alpha}{2} \, e^{-\alpha|c_i - y|}$ into (A.8) and solving the integrals, we finally obtain

$$D_{u,i,j} \approx e^{-\alpha|j-i|d} \, (j - i)^2 d^2 \, \sinh \frac{\alpha d}{2}. \qquad \text{(A.9)}$$

References

[Aaron and Girod, 2002] Aaron, A. and Girod, B. (2002). Compression with side information using turbo codes. In *IEEE Data Compression Conference*, pages 252–261.

[Aaron et al., 2004] Aaron, A., Rane, S., Setton, E., and Girod, B. (2004). Transform-domain Wyner-Ziv codec for video. In *SPIE Visual Communications and Image Processing*, San Jose, CA, USA.

[Aaron et al., 2003] Aaron, A., Setton, E., and Girod, B. (2003). Towards practical wyber-ziv coding of video. In *IEEE International Conference on Image Processing (ICIP)*, pages 869–872, Barcelona.

[Aaron et al., 2002] Aaron, A., Zhang, R., and Girod, B. (2002). Wyner-Ziv coding of motion video. In *Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 240–244, Pacific Grove, CA, USA.

[Abellan and Moral, 2000] Abellan, J. and Moral, S. (2000). A non-specificity measure for convex sets of probability distributions. *Intern. Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, 8(3):357 – 367.

[Alahi et al., 2009] Alahi, A., Boursier, Y., Jacques, L., and Vandergheynst, P. (2009). Sport players detection and tracking with a mixed network of planar and omnidirectional cameras. In *Proceedings of ACM/IEEE ICDSC*, pages 1–8, Como, Italy.

[Ascenso et al., 2005a] Ascenso, J., Brites, C., and Pereira, F. (2005a). Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In *5th EURASIP Conference*, Smolenice, Slovak Republic.

[Ascenso et al., 2005b] Ascenso, J., Brites, C., and Pereira, F. (2005b). Motion compensated refinement for low complexity pixel distributed video coding. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Como, Italy.

[Bahl and Padmanabhan, 2000] Bahl, P. and Padmanabhan, V. N. (2000). RADAR: an in-building RF-based user location and tracking system. pages 775–784.

[Baird et al., 1992] Baird, R. K., Turcheck, J., P., S., and Martin, J. P. (1992). High resolution camera sensor having a linear pixel array. US Patent, Patent Number 5157486.

[Bakhtari and Benhabib, 2007] Bakhtari, A. and Benhabib, B. (2007). An active vision system for multitarget surveillance in dynamic environments. *IEEE Trans Syst Man Cybern B Cybern*, 37(1):190–8.

[Belkoura and Sikora, 2006a] Belkoura, Z. and Sikora, T. (2006a). Towards rate-decoder complexity optimisation in turbo-coder based distributed video coding. In *Picture Coding Symposium*, Beijing, China.

[Belkoura and Sikora, 2006b] Belkoura, Z. M. and Sikora, T. (2006b). Improving Wyner-Ziv video coding by block-based distortion estimation. In *European Signal Processing Conference*, Florence, Italy.

[Bian et al., 2006] Bian, F., Kempe, D., and Govindan, R. (2006). Utility-based sensor selection. In *IPSN 2006: Fifth International Conference on Information Processing in Sensor Networks 2006*, volume 2006, pages 11 – 18, Nashville, TN, United states.

[Boutilier et al., 1999] Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11(0):1 – 94.

[Bowyer and Bogner, 1999] Bowyer, R. and Bogner, R. (1999). Cooperative behaviour in multi-sensor systems. *ICONIP99 6th International Conference on Neural Information Processing. Proceedings*, page 10.1109/ICONIP.1999.845699.

[Brites et al., 2006a] Brites, C., Ascenso, J., and Pereira, F. (2006a). Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities. In *14th EUSIPCO Conference*, Florence, Italy.

[Brites et al., 2006b] Brites, C., Ascenso, J., and Pereira, F. (2006b). Improving transform domain Wyner-Ziv video coding performance. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II–525–II–528, Toulouse, France.

[Brites et al., 2006c] Brites, C., Ascenso, J., and Pereira, F. (2006c). Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding. In *Picture Coding Symposium*, Beijing, China.

[Bruckstein et al., 2003] Bruckstein, A., Elad, M., and Kimmel, R. (2003). Down-scaling for better transform compression. *IEEE Trans. Image Processing*, 12(9):1132–1144.

[Casares and Velipasalar, 2008] Casares, M. and Velipasalar, S. (2008). Light-weight salient foreground detection for embedded smart cameras. In *Proc. of the ACM/IEEE International Conference on Distributed Smart Cameras*.

[Castanon, 1997] Castanon, D. A. (1997). Approximate dynamic programming for sensor management. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2, pages 1202 – 1207, San Diego, CA, USA.

[Cheung and Ortega, 2006] Cheung, N.-M. and Ortega, A. (2006). A model-based approach to correlation estimation in wavelet-based distributed source coding with application to hyperspectral imagery. In *IEEE International Conference on Image Processing (ICIP)*, pages 613–616, Atlanta, USA.

[Cheung et al., 2005] Cheung, N.-M., Wang, H., and Ortega, A. (2005). Correlation estimation for distributed source coding under information exchange constraints. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages II–682–II–685, Genova, Italy.

[Clos et al., 2004] Clos, H., Federspiel, L., and Schoos, A. (2004). System for detecting seat occupancy. USPTO Patent Office, Publication Number US20070029768A1.

[Comer et al., 1996] Comer, M. L., Shen, K., and Delp, E. J. (1996). Rate-scalable video coding using a zerotree wavelet approach. In *Proc. Image and Multidimensional Digital Signal Processing Workshop*, volume III, pages 162–163, Belize City, Belize.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

[Dalai et al., 2006a] Dalai, M., Leonardi, R., and Pereira, F. (2006a). Improving turbo codec integration in pixel-domain distributed video coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II–537–II–540, Toulouse, France.

[Dalai et al., 2006b] Dalai, M., Leonardi, R., and Pereira, F. (2006b). Intra mode decision based on spatio-temporal cues in pixel domain Wyner-Ziv video coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France.

[Dang et al., 2006] Dang, V. D., Dash, R. K., Rogers, A., and Jennings, N. R. (2006). Overlapping coalition formation for efficient data fusion in multi-sensor networks. volume 1, pages 635 – 640, Boston, MA, United states.

[Dash et al., 2007] Dash, R., Vytelingum, P., Rogers, A., David, E., and Jennings, N. (2007). Market-based task allocation mechanisms for limited-capacity suppliers. *IEEE Transactions On Systems Man And Cybernetics Part A-Systems And Humans*, 37(3):391–405.

[Davis et al., 2007] Davis, J. W., Morison, A. M., and Woods, D. D. (2007). An adaptive focus-of-attention model for video surveillance and monitoring. *Mach. Vision Appl.*, 18(1):41–64.

[De Vleeschouwer and Delannay, 2009] De Vleeschouwer, C. and Delannay, D. (2009). Basket ball dataset from the European project APIDIS. `http://www.apidis.org/Dataset/`.

[Delannay et al., 2009] Delannay, D., Danhier, N., and Vleeschouwer, C. D. (2009). Detection and recognition of sports(wo)men from multiple views. In *Proceedings of ACM/IEEE ICDSC*, pages 1–7, Como, Italy.

[Dempster, 1968] Dempster, A. P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society, Series B*, 30:205–247.

[Denoeux, 2008] Denoeux, T. (2008). Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172(2-3):234 – 264.

[Denzler et al., 2003] Denzler, J., Zobel, M., and Niemann, H. (2003). Information theoretic focal length selection for real-time active 3-d object tracking. In *Proc. IEEE Intern. Conf. on Computer Vision*, page 400.

[El-Desouki et al., 2009] El-Desouki, M., Deen, M. J., Fang, Q., Liu, L., Tse, F., and Armstrong, D. (2009). CMOS image sensors for high speed applications. *Sensors*, 9(1):430–444.

[Elwell, 2009] Elwell, B. (2009). Occupancy sensor network. US Patent, Patent Number 7486193.

[Ercan et al., 2006] Ercan, A. O., Gamal, A. E., and Guibas, L. (2006). Camera network node selection for target localization in the presence of occlusions. In *In SenSys Workshop on Distributed Cameras*.

[Ericson and Ramamoorthy, 1979] Ericson, T. and Ramamoorthy, V. (1979). Modulo-PCM: A new source coding scheme. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*.

[Etoh et al., 2003] Etoh, T., Poggemann, D., Kreider, G., Mutoh, H., Theuwissen, A., Ruckelshausen, A., Kondo, Y., Maruno, H., Takubo, K., Soya, H., Takehara, K., Okinaka, T., and Takano, Y. (2003). An image sensor which captures 100 consecutive frames at 1000000 frames/s. *IEEE Trans. Electron Devices*, 50(1):144–151.

[Federspiel and Michael, 2005] Federspiel, L. and Michael, M. (2005). Method for manufacturing a floor carpet structure including a sensing function for an automotive vehicle. USPTO Patent Office, Publication Number US20070178274A1.

[Fleuret et al., 2008] Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):267–282.

[Fowler, 2000] Fowler, J. J. (2000). Occupancy sensor and method of operating same. US Patent, Patent Number 6078253.

[Gemeiner et al., 2007] Gemeiner, P., Ponweiser, W., Einramhof, P., and Vincze, M. (2007). Real-time SLAM with high-speed CMOS camera. In *Proceedings IEEE Int. Conf. Image Analysis and Processing*, pages 297–302.

[Gonsalves and Rinkus, 1998] Gonsalves, P. and Rinkus, G. (1998). Intelligent fusion and asset management processor. *1998 IEEE Information Technology Conference, Proceedings*, pages 15–18.

[Gray and Neuhoff, 1998] Gray, R. M. and Neuhoff, D. L. (1998). Quantization. *IEEE Trans. Inform. Theory*, 44(6):2325–2383.

[Gunduz and Erkip, 2006] Gunduz, D. and Erkip, E. (2006). Distortion exponent of parallel fading channels. In *IEEE International Symposium on Information Theory*, pages 694–698, Seattle, WA.

[Gupta et al., 2007] Gupta, A., Mittal, A., and Davis, L. S. (2007). Cost: An approach for camera selection and multi-object inference ordering in dynamic scenes. *IEEE Intern. Conf. on Computer Vision*, pages 1–8.

[Hager and Durrant-Whyte, 1986] Hager, G. D. and Durrant-Whyte, H. F. (1986). Information and multi-sensor coordination. In *UAI*, pages 381–394.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[He et al., 2003] He, X., Sun, X., and von Laszewski, G. (2003). QoS guided Min-Min heuristic for Grid task scheduling. *Journal of Computer Science and*

Technology, 18(4):442–451. 1st International Workshop on Grid and Cooperative Computing (GCC2002), Hainan, Peoples R China, December, 2002.

[Hoeher et al., 2000] Hoeher, P., Land, I., and Sorger, U. (2000). Log-likelihood values and monte carlo simulation– some fundamental results. In *Int. Symp. on Turbo Codes and Rel. Topics*, pages 43–46.

[Inada, 2006] Inada, M. (2006). Linear image sensor, image reading apparatus using the same, image reading method, and program for implementing the method. USPTO Patent Office, Publication Number US20060061835A1.

[Isler and Bajcsy, 2005] Isler, V. and Bajcsy, R. (2005). The sensor selection problem for bounded uncertainty sensing models. volume 2005, pages 151 – 158.

[Isler et al., 2005] Isler, V., Khanna, S., Spletzer, J., and Taylor, C. J. (2005). Target tracking with distributed sensors: The focus of attention problem. *Comput. Vis. Image Underst.*, 100(1-2):225–247.

[Jain, 1989] Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc.

[Javed et al., 2002] Javed, O., Shafique, K., and Shah, M. (2002). A hierarchical approach to robust background subtraction using color and gradient information. In *Proc. of IEEE Workshop on Motion and Video Computing*, pages 22–27.

[Jayant and Noll, 1984] Jayant, N. S. and Noll, P. (1984). *Digital coding of waverforms*. Prentice-Hall, Inc.

[Kaewtrakulpong and Bowden, 2001] Kaewtrakulpong, P. and Bowden, R. (2001). An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems*, volume 5308, pages 149–158.

[Kalandros and Pao, 1999] Kalandros, M. and Pao, L. Y. (1999). Randomization and super-heuristics in choosing sensor sets for target tracking applications. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2, pages 1803 – 1808, Phoenix, AZ, USA.

[Kawamoto and Narabu, 1999] Kawamoto, S. and Narabu, T. (1999). Image sensing device, linear sensor for use in the image sensing device, and method of driving the linear sensor. US Patent, Patent Number 5920063.

[Kay, 1993] Kay, S. (1993). *Fundamentals of Statistical Signal Processing, Estimation Theory*. Prentice Hall Englewood Cliffs, Inc.

[Klir, 1991] Klir, G. (1991). Generalized Information-Theory. *Fuzzy Sets and Systems*, 40(1):127–142.

[Klir and Wierman, 1999] Klir, G. and Wierman, M. J. (1999). *Uncertainty-based information: elements of generalized information theory*. Physica-Verlag/Springer-Verlag, Heidelberg and New York.

[Kraus et al., 1995] Kraus, S., Wilkenfeld, J., and Zlotkin, G. (1995). Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297 – 345.

[Kubasov et al., 2007] Kubasov, D., Nayak, J., and Guillemot, C. (2007). Optimal reconstruction in Wyner-Ziv video coding with multiple side information. In *Proc. of IEEE International Workshop on Multimedia Signal Processing*, pages 183–186.

[Kurita et al., 2005] Kurita, Y., Iida, Y., Kempf, R., Kaneko, M., Mishima, H. K., Tsukamoto, H., and Sugimoto, E. (2005). Dynamic sensing of human eye using a high speed camera. In *Proc. IEEE Int. Conf. Information Acquisition*, pages 338–343.

[Kwok et al., 2002] Kwok, K. S., Driessen, B. J., Phillips, C. A., and Tovey, C. A. (2002). Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 35(1):111 – 122.

[Lee et al., 2008] Lee, H., Tessens, L., Morbee, M., Aghajan, H., and Philips, W. (2008). Sub-optimal camera selection in practical vision networks through shape approximation. volume 5259 LNCS, pages 266 – 277, Juan-les-Pins, France.

[Lee and Zomaya, 2007] Lee, Y. C. and Zomaya, A. Y. (2007). Practical scheduling of bag-of-tasks applications on grids with dynamic resilience. *IEEE Transactions on Computers*, 56(6):815–825.

[Li et al., 2003] Li, L., Huang, W., Gu, I. Y. H., and Tian, Q. (2003). Foreground object detection from videos containing complex background. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 2–10, New York, NY, USA. ACM.

[Lin and Costello, 2004] Lin, S. and Costello, D. J. (2004). *Error Control Coding*. Pearson Prentice Hall.

[Liu et al., 2006] Liu, L., Li, Z., and Delp, E. J. (2006). Backward channel aware Wyner-Ziv video coding. In *IEEE International Conference on Image Processing (ICIP)*, pages 1677–1680, Atlanta, USA.

[Liveris et al., 2002] Liveris, A., Xiong, Z., and Georghiades, C. (2002). Compression of binary sources with side information using low-density parity-check codes. In *Global Telecommunications Conference*, volume 2, pages 1300–1304.

[Luo et al., 1998] Luo, R. C., Shr, A. M., and Hu, C.-Y. (1998). Multiagent based multisensor resource management system. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1034 – 1039, Victoria, Can.

[International Telecommunication Union, 1998]
International Telecommunication Union (1998). Video coding for low bit rate communication. ITU-T Recommendation H.263. http://www.itu.int/rec/T-REC-H.263/.

[McCarthy and Muller, 2005] McCarthy, M. and Muller, H. (2005). Positioning with independent ultrasonic beacons. Computing Science Technical Report CSTR-05-005, Department of Computer Science, University of Bristol, UK.

[Mittal and Davis, 2003] Mittal, A. and Davis, L. S. (2003). M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision*, 51(3):189–203.

[Molina Lopez et al., 1995] Molina Lopez, J., Rodriguez, F., and Corredera, J. (1995). Fuzzy reasoning for multisensor management. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1398 – 1403, Vancouver, BC, Can.

[Morbee et al., 2007a] Morbee, M., Prades-Nebot, J., Pižurica, A., and Philips, W. (2007a). Improved pixel-based rate allocation for pixel-domain distributed video coders without feedback channel. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Lecture Notes in Computer Science, pages 663–674, Delft, the Netherlands. Springer-Verlag.

[Morbee et al., 2007b] Morbee, M., Prades-Nebot, J., Pižurica, A., and Philips, W. (2007b). Rate allocation algorithm for pixel-domain distributed video coding without feedback channel. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I–521–I–524, Honolulu, HI, USA.

[Morbee et al., 2006a] Morbee, M., Prades-Nebot, J., Pizurica, A., and Philips, W. (2006a). Feedback channel suppression in pixel-domain distributed video coding. In *Proceedings of the 17th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC)*, pages 154–157, Eindhoven, The Netherlands. Technology Foundation/IEEE Benelux.

[Morbee et al., 2006b] Morbee, M., Prades-Nebot, J., Pizurica, A., and W., P. (2006b). Content-based MPEG-4 FGS video coding for video surveillance. In *Proc. of SPS-DARTS 2006 (the second annual IEEE Benelux/DSP Valley Signal Processing Symposium*, pages 135–138.

[Morbee et al., 2008a] Morbee, M., Roca, A., Prades-Nebot, J., Pizurica, A., and Philips, W. (2008a). Reduced decoder complexity and latency in pixel-domain Wyner-Ziv video coders. *Springer Journal on Signal, Image and Video Processing (SIViP)*, 2(2):129–140.

[Morbee and Tessens, 2010] Morbee, M. and Tessens, L. (2010). Multiple light-integrating line sensors for 2D occupancy sensing. EPO Patent Office, Application Number EP10164483.9.

[Morbee and Tessens, 2011] Morbee, M. and Tessens, L. (2011). Multiple light-integrating line sensors for 2D occupancy sensing. EPO Patent Office, Application Number EP11000138.5.

[Morbee et al., 2010] Morbee, M., Tessens, L., Aghajan, H., and Philips, W. (2010). Dempster-Shafer based multi-view occupancy maps. *Electronic Letters*, 46.

[Morbee et al., 2011] Morbee, M., Tessens, L., Aghajan, H., and Philips, W. (2011). Dempster-Shafer based task assignment in vision networks. *submitted to International Journal on Computer Vision*.

[Morbee et al., 2008b] Morbee, M., Tessens, L., Lee, H., Philips, W., and Aghajan, H. (2008b). Optimal camera selection in vision networks through shape approximation. In *Proceedings of the 2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 46 – 51, Cairns, Queensland, Australia. ISBN: 978-1-4244-2295-1.

[Morbee et al., 2009a] Morbee, M., Tessens, L., Philips, W., and Aghajan, H. (2009a). PhD forum: Dempster-Shafer based camera contribution evaluation for task assignment in vision networks. In *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, pages 1–2.

[Morbee et al., 2007c] Morbee, M., Tessens, L., Prades-Nebot, J., Pižurica, A., and Philips, W. (2007c). A distributed coding-based extension of a mono-view to a multi-view video system. In *3DTV-Conference*, Kos, Greece.

[Morbee et al., 2007d] Morbee, M., Tessens, L., Quang-Luong, H., Prades-Nebot, J., Pižurica, A., and Philips, W. (2007d). A distributed coding-based content-aware multi-view video system. In *International Conference on Distributed Smart Cameras (ICDSC)*, pages 355–362, Vienna, Austria.

[Morbee et al., 2009b] Morbee, M., Velisavljevic, V., Mrak, M., and Philips, W. (2009b). Scalable feature-based video retrieval for mobile devices. In *ACM International Conference on Internet Multimedia Computing and Service (ICIMCS)*, pages 1–7, Kunming, Yunnan, China.

[Muehlmann et al., 2004] Muehlmann, U., Ribo, M., Lang, P., and Pinz, A. (2004). A new high speed CMOS camera for real-time tracking applications. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 5195–5200.

[Mullen et al., 2006] Mullen, T., Avasarala, V., and Hall, D. L. (2006). Customer-driven sensor management. *IEEE Intelligent Systems*, 21(2):41 – 49.

[Munoz-Salinas et al., 2009] Munoz-Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F., and Carmona-Poyato, A. (2009). Multi-camera people tracking using evidential filters. *Intern. Journal of Approximate Reasoning*, 50(5):732 – 749.

[Myron et al., 1997] Myron, Douglas, D., Williams, Errol, R., Hardin, Charles, C., Woytek, Timothy, W., and Stephens, Michael, A. (1997). Occupancy sensor and method of operating same. EPO Patent Office, Publication Number EP0809922A1.

[Netravali and Limb, 1980] Netravali, A. and Limb, J. O. (1980). Picture coding: A review. *Proceedings of the IEEE*, 68(3):366–406.

[Nishiyama, 2004] Nishiyama, Y. (2004). Image processing device using line sensor. US Patent, Patent Number 6748124.

[Ostwald et al., 2005] Ostwald, J., Lesser, V., and Abdallah, S. (2005). Combinatorial auctions for resource allocation in a distributed sensor network. *RTSS 2005: 26th IEEE International Real-Time Systems Symposium, Proceedings*, pages 266–274.

[Pahalawatta and Katsaggelos, 2004] Pahalawatta, P. V. and Katsaggelos, A. K. (2004). Optimal sensor selection for video-based target tracking in a wireless sensor network. In *Proc. Intern. Conf. on Image Processing*, pages 3073–3076.

[Petrovic et al., 2009] Petrovic, N., Jovanov, L., Pizurica, A., and Philips, W. (2009). Efficient foreground detection for real-time surveillance applications. *IEEE Trans. On Circuits and Systems for Video Technology*. - submitted.

[Prades-Nebot et al., 2010] Prades-Nebot, J., Morbee, M., and Delp, E. J. (2010). Very low complexity coding of images using modulo-PCM. *submitted to IEEE Trans. Circuits Syst. Video Technol.*

[Puri et al., 2006] Puri, R., Majumdar, A., Ishwar, P., and Ramchandran, K. (2006). Distributed video coding in wireless sensor networks. *IEEE Signal Processing Mag.*, 23(4):94–106.

[Puri and Ramchandran, 2002] Puri, R. and Ramchandran, K. (2002). PRISM: A new robust video coding architecture based on distributed compression principles. In *Allerton Conference on Communication, Control, and Computing*, Allerton, IL, USA.

[Ramamoorthy, 1981] Ramamoorthy, V. (1981). Speech coding using Modulo-PCM with side information. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, volume 6, pages 832–834.

[Robertson et al., 1995] Robertson, P., Villebrun, E., and Hoeher, P. (1995). A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In *IEEE Int. Conf. on Communications*, volume 2, pages 1009–1013.

[Roca et al., 2008] Roca, A., Morbee, M., Prades-Nebot, J., and Delp, E. (2008). Rate control algorithm for pixel-domain Wyner-Ziv video coding. In *Proc. Visual Communications and Image Processing (VCIP)*, San Jose, CA, USA.

[Roca et al., 2007] Roca, A., Morbee, M., Prades-Nebot, J., and Delp, E. J. (2007). A distortion control algorithm for pixel-domain Wyner-Ziv video coding. In *Picture Coding Symposium*, Lisbon, Portugal.

[Rowaihy et al., 2007] Rowaihy, H., Eswaran, S., Johnson, M., Verma, D., Bar-Noy, A., Brown, T., and La Porta, T. (2007). A survey of sensor selection schemes in wireless sensor networks. In *Proc. of SPIE*, volume 6562.

[Rowitch and Milstein, 2000] Rowitch, D. and Milstein, L. (2000). On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes. *IEEE Transactions on Communications*, 48(6):948–959.

[Schiele and Crowley, 1994] Schiele, B. and Crowley, J. L. (1994). Comparison of position estimation techniques using occupancy grids. In *In Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 1628–1634.

[Sekikawa and Kubono, 2007] Sekikawa, J. and Kubono, T. (2007). Spectroscopic imaging observation of break arcs using a high-speed camera. In *Proc. IEEE Conf. Electrical Contacts*, pages 275–279.

[Shafer, 1976] Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

[Shehory and Kraus, 1998] Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165 – 200.

[Shum and Komura, 2005] Shum, H. and Komura, T. (2005). Tracking the translational and rotational movement of the ball using high-speed camera movies. In *Proc. IEEE Int. Conf. Image Processing*, pages 1084–1087, Genoa, Italy.

[Slepian and Wolf, 1973] Slepian, J. and Wolf, J. (1973). Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471– 480.

[Smith III, 2004] Smith III, J. F. (2004). Fuzzy logic resource manager: Real-time adaptation and self-organization. In *Proc. of SPIE*, volume 5429, pages 77 – 88, Orlando, FL, United states.

[Snidaro et al., 2003] Snidaro, L., Niu, R., Varshney, P. K., and Foresti, G. L. (2003). Automatic camera selection and fusion for outdoor surveillance under changing weather conditions. In *Proc. of the IEEE Conf. on Advanced Video and Signal Based Surveillance*, page 364.

[Sommerlade and Reid, 2008] Sommerlade, E. and Reid, I. (2008). Information theoretic active scene exploration. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1397–1403.

[Soro and Heinzelman, 2007] Soro, S. and Heinzelman, W. (2007). Camera selection in visual sensor networks. In *2007 IEEE Conf. on Advanced Video and Signal Based Surveillance, AVSS 2007 Proceedings*, pages 81 – 86, London, United kingdom.

[Stauffer and Grimson, 2000] Stauffer, C. and Grimson, W. (2000). Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747–757.

[Sun and Li, 2005] Sun, J. and Li, H. (2005). A Wyner-Ziv coding approach to transmission of interactive video over wireless channels. In *IEEE International Conference on Image Processing (ICIP)*, pages 686–689, Genoa, Italy.

[Tagliasacchi et al., 2006] Tagliasacchi, M., Trapanese, A., Tubaro, S., Ascenso, J., Brites, C., and Pereira, F. (2006). Intra mode decision based on spatio-temporal cues in pixel domain Wyner-Ziv video coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II–57–II–60, Toulouse, France.

[Talukdar et al., 1998] Talukdar, S., Baerentzen, L., Gove, A., and De Souza, P. (1998). Asynchronous teams: cooperation schemes for autonomous agents. *Journal of Heuristics*, 4(4):295 – 321.

[Tao et al., 2007] Tao, C., Zhang, Y., and Jiang, J. J. (2007). Extracting physiologically relevant parameters of vocal folds from high-speed video image series. *IEEE Trans. Biomed. Eng.*, 54(5):794–801.

[Tessens, 2010] Tessens, L. (2010). *Information selection and fusion in vision systems*. Doctoral Thesis, Ghent University.

[Tessens and Morbee, 2010] Tessens, L. and Morbee, M. (2010). Video of occupancy map demonstrator. `http://telin.ugent.be/~ltessens/demoISYSS/`.

[Tessens et al., 2011] Tessens, L., Morbee, M., Aghajan, H., and Philips, W. (2011). Camera selection for tracking in smart camera networks. *submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Tessens et al., 2008] Tessens, L., Morbee, M., Lee, H., Philips, W., and Aghajan, H. (2008). Principal view determination for camera selection in distributed smart camera networks. In *Proceedings of ACM/IEEE ICDSC*, pages 1–8, Stanford, CA, USA.

[Tessens et al., 2009] Tessens, L., Morbee, M., Philips, W., Kleihorst, R., and Aghajan, H. (2009). Efficient approximate foreground detection for low-resource devices. In *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, pages 1–8.

[Tian et al., 2005] Tian, Y., Lu, M., and Hampapur, A. (2005). Robust and efficient foreground analysis for real-time video surveillance. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings*, pages 1182–1187.

[Tipnis et al., 2001] Tipnis, S., Nagarkar, V., Gaysinskiy, V., Miller, S., and Shestakova, I. (2001). High speed X-ray imaging camera for time resolved diffraction studies. In *IEEE Nuclear Science Symposium Conference Record*, pages 164 – 167.

[Trapanese et al., 2005] Trapanese, A., Tagliasacchi, M., Tubaro, S., Ascenso, J., Brites, C., and Pereira, F. (2005). Improved correlation noise statistics modeling in frame-based pixel domain Wyner-Ziv video coding. In *International workshop on very low bit rate video*, Sardinia, Italy.

[Ueno et al., 2006] Ueno, A., Otani, Y., and Uchikawa, Y. (2006). A noncontact measurement of saccadic eye movement with two high-speed cameras. In *Proc. IEEE Int. Conf. of the Engineering in Medicine and Biology Society*, pages 5583–5586.

[Westerlaken et al., 2006] Westerlaken, R. P., Borchert, S., Gunnewiek, R. K., and Lagendijk, R. L. (2006). Dependency channel modeling for a LDPC-based Wyner-Ziv video compression scheme. In *IEEE International Conference on Image Processing (ICIP)*, pages 277–280, Atlanta, USA.

[Winkler, 2006] Winkler, S. (2006). *Digital Video Quality*. John Wiley & Sons.

[Wolberg, 1990] Wolberg, G. (1990). *Digital Image Warping*. Wiley-IEEE Computer Society.

[Wyner and Ziv, 1976] Wyner, A. and Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10.

[Xiong and Svensson, 2002] Xiong, N. and Svensson, P. (2002). Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–86.

[Xiong et al., 2004] Xiong, Z., Liveris, A., and Cheng, S. (2004). Distributed source coding for sensor networks. *IEEE Signal Processing Mag.*, 21(5):80–94.

[Xu and Xiong, 2004] Xu, Q. and Xiong, Z. (2004). Layered Wyner-Ziv video coding. In *SPIE Visual Communications and Image Processing: Special Session on Multimedia Technologies for Embedded Systems*, San Jose, CA, USA.

[Xu and Xiong, 2006] Xu, Q. and Xiong, Z. (2006). Layered Wyner-Ziv video coding. *IEEE Transactions on Image Processing*, 15(12):3791–3803.

[Yao et al., 2010] Yao, Y., Chen, C.-H., Koschan, A., and Abidi, M. (2010). Adaptive online camera coordination for multi-camera multi-target surveillance. *Comput. Vis. Image Underst.*, 114(4):463–474.

[Zeng and Venetsanopoulos, 1993] Zeng, B. and Venetsanopoulos, A. (1993). A JPEG-based interpolative image coding scheme. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pages 393–396, Minneapolis, MN.

[Zhevelev et al., 2009] Zhevelev, B., Kotlicki, Y., and Lahat, M. (2009). Passive infra-red detectors. US Patent, Patent Number 7573032.

[Zivkovic et al., 2008] Zivkovic, Z., Kleihorst, R., Danilin, A., Schueler, B., Chan, C., Aghajan, H., Arturi, G., and Kliger, V. (2008). Towards low latency gesture control using smart camera network. In *Proc. ECV/CVPR 2008*.

[Zlotkin and Rosenschein, 1991] Zlotkin, G. and Rosenschein, J. S. (1991). Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1317 – 1324.