Synthesis, Coding, and Evaluation of 3D Images Based on Integral Imaging

Roger Olsson

Department of Information Technology and Media
Mid Sweden University

Doctoral Thesis No. 55
Sundsvall, Sweden
2008

Mittuniversitetet
Informationsteknologi och medier
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-85317-98-1
ISSN 1652-893X

Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av Teknologie Doktorsexamen torsdagen den 12 juni 2008 i L111, Mittuniversitetet, Holmgatan 10, Sundsvall.

Tryck: Kopieringen, Mittuniversitetet, Sundsvall.

Always in motion is the future

Yoda

Sammanfattning

De senaste åren har kameraprototyper som kan fånga tredimensionella (3D) bilder presenterats, baserade på 3D-tekniken Integral Imaging (II). När dessa II-bilder betraktas på en 3D-skärm, delger de både ett djup och ett innehåll som på ett realistiskt sätt ändrar perspektiv när tittaren ändrar sin betraktningsposition.

Avhandlingen koncentrerar sig på tre hämmande faktorer gällande II-bilder. För det första finns det en mycket begränsad allmän tillgång till II-bilder för jämförande forskning och utveckling av kodningsmetoder. Det finns heller inga objektiva kvalitetsmått som uttryckligen mäter distorsion med avseende på II-bildens egenskaper: djup och betraktningsvinkelberoende. Slutligen uppnår nuvarande standarder för bildkodning låg kodningseffektivitet när de appliceras på II-bilder.

En metod baserad på datorrendrering har utvecklats som tillåter produktion av olika typer av II-bilder. En II-kameramodell ingår som bas, kombinerad med ett scenbeskrivningsspråk som möjligör att godtydligt komplexa virtuella scener definieras. Ljustransporten inom scenen och fram till II-kameran simuleras med strålföljning och geometrisk optik. Den presenterade metoden används för att skapa ett antal II-kameramodeller, scendefinitioner och II-bilder.

Två kvalitetmått har tagits fram för att objektivt kvantifiera distorsion som kan uppträda i en II-bild med avseende på dess specifika egenskaper. Det första måttet modellerar hur distortionen uppfattas av en tittare som betraktar en 3D-skärm ur olika betraktningsvinklar. Det andra måttet beräknar distorsionens djupdistribution inom II-bilden. Nya aspekter av kodningsinducerade artefakter påvisas med de föreslagna kvalitetsmåtten.

Slutligen har en kodningsmetod för II-bilder utarbetats som bland annat utnyttjar videokodningsstandarden H.264 / AVC genom att först transformera II-bilden till en pseudovideosekvens (PVS). Kodningsmetodens egenskaper har studerats i detalj och jämförts med andra kodningsmetoder, bland annat med hjälp av de föreslagna kvalitetsmåtten. Den föreslagna kodningsmetoden åstadkommer samma kvalitet som JPEG2000 till ungefärligen 1 / 60-del av kraven på lagring och distribution.

Abstract

In recent years camera prototypes based on Integral Imaging (II) have emerged that are capable of capturing three-dimensional (3D) images. When being viewed on a 3D display, these II-pictures convey depth and content that realistically change perspective as the viewer changes the viewing position.

The dissertation concentrates on three restraining factors concerning II-picture progress. Firstly, there is a lack of digital II-pictures available for inter alia comparative research and coding scheme development. Secondly, there is an absence of objective quality metrics that explicitly measure distortion with respect to the II-picture properties: depth and view-angle dependency. Thirdly, low coding efficiencies are achieved when present image coding standards are applied to II-pictures.

A computer synthesis method has been developed, which enables the production of different II-picture types. An II-camera model forms a basis and is combined with a scene description language that allows for the describing of arbitrary complex virtual scenes. The light transport within the scene and into the II-camera is simulated using ray-tracing and geometrical optics. A number of II-camera models, scene descriptions, and II-pictures are produced using the presented method.

Two quality evaluation metrics have been constructed to objectively quantify the distortion contained in an II-picture with respect to its specific properties. The first metric models how the distortion is perceived by a viewer watching an II-display from different viewing-angles. The second metric estimates the depth-distribution of the distortion. New aspects of coding-induced artifacts within the II-picture are revealed using the proposed metrics.

Finally, a coding scheme for II-pictures has been developed that inter alia utilizes the video coding standard H.264/AVC by firstly transforming the II-picture into a pseudo video sequence. The properties of the coding scheme have been studied in detail and compared with other coding schemes using the proposed evaluation metrics. The proposed coding scheme achieves the same quality as JPEG2000 at approximately 1/60th of the storage- or distribution requirements.

Acknowledgements

I would like to thank my main supervisor Professor Youzhi Xu for his confidence in providing me with the opportunity to become a PhD-student at Mid Sweden University. Thanks also to my supervisor Docent Tingting Zhang who, together with Youzhi, allowed me to independently explore and define the topic of this work. In addition I would like to specially acknowledge the help provided by my supervisor Docent Mårten Sjöström. His guidance and helpful comments on scientific character and writing has helped me transform my text into a form that is far more understandable than it was initially. Thanks also to Professor Theo Kanter for his suggestions on how to improve the main theme of the dissertation, and Fiona Wait for proofreading the text.

My colleagues at the Division of Information and Communication Systems are worthy of great appreciation. The interesting discussions at coffee breaks have provided rewarding pauses from struggling with subordinate clauses or something similarly intriguing. Better still have been the (more or less speculative) lunch debates on various topics well outside the realm of 3D images that I have had the opportunity of participating in together with Magnus Eriksson, Stefan Pettersson, Mårten, and others. Thanks also to all other PhD-students on the Department of Information Technology and Media and the Swedish Graduate School of Telecommunication (GST) with whom I have crossed paths during these years and learned a lot. As a GST-student I also greatly appreciate the financial support provided by GST as well as project funding from the EU Objective 1-programme Södra Skogslän region.

Finally, I want to give special thanks to my family who have cheered me through both floods and droughts in motivation and creativity. To my wife Sara I want to express my love; this thesis would not have been written without your love, support and encouragement.

Roger Olsson
Sundsvall, May 2008

Contents

List of selected papers

This dissertation is mainly based on the following papers:

I Roger Olsson and Youzhi Xu. A ray-tracing based simulation environment for generating integral imaging source material. In *Proceedings of RadioVetenskap och Kommunikation*, pp. 663 – 666, Linköping, Sweden, June, 2005.

II Roger Olsson and Youzhi Xu. An interactive ray-tracing based simulation environment for generating integral imaging video sequences. In *Proceedings of Optics East*, SPIE, Vol. 6016, pp. 150 – 157, Boston (MA), USA, October, 2005.

III Roger Olsson, Mårten Sjöström, and Youzhi Xu. A combined pre-processing and H.264-compression scheme for 3D integral images. In *Proceedings of International Conference on Image Processing*, IEEE, pp. 513 – 516, Atlanta (GA), USA, October, 2006.

IV Roger Olsson, Mårten Sjöström, and Youzhi Xu. Evaluation of combined pre-processing and H.264-compression schemes for 3D integral images. In *Proceedings of Visual Communications and Image Processing*, SPIE, Vol. 6508, pp. 65082C-1 – 65082C-12, San Jose (CA), USA, January, 2007.

V Roger Olsson and Mårten Sjöström. A Depth Dependent Quality Metric for Evaluation of Coded Integral Imaging based 3D-images. In *Proceedings of 3DTV-Conference*, IEEE / EURASIP / MPEG-IF, Kos, Greece, May, 2007.

VI Roger Olsson and Mårten Sjöström. A novel quality metric for evaluating depth distribution of artifacts in coded still 3D images. In *Proceedings of Stereoscopic Display and Application XCIX*, SPIE, Vol. 6803, San Jose (CA), USA, January, 2008.

VII Roger Olsson. Empirical rate-distortion analysis of JPEG 2000 3D and H.264 / AVC coded integral imaging based 3D-images. *Accepted for 3DTV-Conference*, IEEE / EURASIP / MPEG-IF, Istanbul, Turkey, May, 2008.

VIII Roger Olsson, Mårten Sjöström, and Youzhi Xu. A Coding Scheme for Integral-Imaging-Based 3D Images Using H.264 / AVC and JPEG2000 3D. Submitted to *IEEE Transactions on Multimedia*, May, 2008.

Hereafter these papers are referred to by their Roman numerals.

Terminology

Abbreviations and Acronyms

3DAV	3D Audio and Video
3DTV	Three-dimensional TV
BMP	Bitmap Image
bpc	bits per color channel
B-picture	Bi-directional prediction coded picture
bpp	bits per pixel
bps	bits per second
CAD/CAE	Computer Aided Design / Computer Aided Engineering
CBR	Constant bitrate
CCD	Charge-Coupled Device
CI	Component Image
CISO	Component Image Selection Order
CQ	Constant quantizer
CT	Computer Tomography
CRT	Cathode Ray Tube
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
EI	Elementary Image
EPI	Epipolar Plane Image
FOV	Field of view
GRIN	GRadient INdex of refraction
H.264/AVC	H.264/MPEG-4 AVC
HDTV	High Definition TV
HPO	Horizontal Parallax Only
HVS	Human Visual System
IBR	Image Based Rendering
II	Integral Imaging
II-picture	Integral Imaging picture
IoR	Index-of-Refraction
IP	Integral Photography
I-picture	Intra coded picture
JPEG	Joint Photographic Experts Group

JPEG2000	JPEG2000 Part 1
JP3D	JPEG2000 Part 10
JVT	Joint Video Team
LCD	Liquid Crystal Display
LUT	Look-Up Table
MCP	Motion-Compensated Prediction
MPEG	Moving Picture Experts Group
Mpixels	Mega pixels (10^6 pixels)
MRI	Magnetic Resonance Imaging
MSSIM	Mean Structural SIMilarity index
PNG	Portable Network Graphics
P-picture	Prediction coded picture
PSNR	Peak Signal to Noise Ratio
PVI	Pseudo Volumetric Image
PVI-slice	Slice in the PVI
PVS	Pseudo Video Sequence
PVS-frame	Picture in the PVS
RI	Ray-space Image
SDL	Scene Description Language
SDTV	Standard Definition TV
SI	Sub Image
VBR	Variable bitrate
VCEG	Video Coding Experts Group
VI	View Image
VZ	Viewing Zone

Mathematical Notation

Notation related to Integral Imaging

P	Plenoptic function
λ	Wavelength of light
t	Time
$\mathbf{P}$	RGB-triplet from vector-form of P
$\mathbf{I}_{2D}$	2D image sampled from P
$\mathbf{I}_{3D}$	3D image sampled from P
α	Viewing angle of lenslet and thereby II-system
Λ	Distance between pixel- and lens array in II-system
δ^L	Pitch of symmetric lenslet ($\delta^L_x = \delta^L_y$)
A	Lenslet magnification factor
f	Lenslet focal length
$\delta_x^{P_I}$	Horizontal pixel pitch at image plane
$\delta_y^{P_I}$	Vertical pixel pitch at image plane
R^I	Spatial resolution at image plane
R	Resolution of pixel array

II	Integral Imaging picture
$II(m,n)$	RGB-pixel at row m and column n in II-picture
M, N	Number of pixels in pixel array horizontally and vertically
U, V	Number of pixels in Elementary Image horizontally and vertically
K, L	Number of Elementary Images horizontally and vertically
Ξ, Ψ	Number of Component Images horizontally and vertically
CI	Component Image
EI	Elementary Image
SI	Sub Image
RI	Ray-space Image

Notation related to Synthesis

$\mathcal{I}$	Set of pixel arrays in a generic ideal II-camera
$\mathcal{O}$	Set of optical elements in a generic ideal II-camera
$\mathcal{C}$	II-camera model
$\mathcal{L}^{C_k}$	Set of $\mathbf{L}$ describing the II-camera C
$\mathcal{D}^{C_k}$	Set of $\vec{D}$ describing the II-camera C
$\mathcal{L}^{I_k}$	Set of $\mathbf{L}$ corresponding to pixels of pixel array k
$\mathcal{D}^{I_k}$	Set of $\vec{D}$ corresponding to pixels of pixel array k
G^P	Generating matrix defining location point of pixel (m,n)
G^L	Generating matrix defining location point of lenslet (k,l)
Δ_x^P	Horizontal size of pixel array
Δ_y^P	Vertical size of pixel array
δ_x^P	Horizontal pixel pitch of pixel array
δ_y^P	Vertical pixel pitch of pixel array
δ_x^L	Horizontal lenslet pitch
δ_y^L	Vertical lenslet pitch
M	Horizontal resolution of each pixel array
N	Vertical resolution of each pixel array
K	Number of pixel arrays used in the II-camera
P	Plenoptic function
$\mathbf{P}$	RGB-triplet from vector-form of P
$\mathbf{P}^C$	Plenoptic function sampled by the II-camera C
$\vec{D}$	Direction vector
θ	Latitudinal rotation angle
ϕ	Longitudinal rotation angle
$\mathbf{L}$	Location point
$f()$	Location point translation function
$g()$	Direction vector rotation function
$\mathbf{B}_k^L$	Bounding box encompassing the space of $\mathcal{L}^{C_k}$
$\mathbf{B}_k^D$	Bounding box encompassing the space of $\mathcal{D}^{C_k}$
$[X,Y,Z]^T$	3D coordinate in the II-camera coordinate system
$[x,y,z]^T$	3D coordinate in the pixel format's color coordinate system
Λ	Distance between pixel- and lens array in II-system

If $k = 1$, index k is omitted from the notation.

Notation related to Evaluation

$PSNR$	Peak Signal to Noise Ratio in dB
Q_{global}	Global quality metric
$\mathbf{Q}_{angle}$	Angle-dependent quality metric
$SI_{u,v}$	SI corresponding to row u and column v
$\mathbf{VI_E}$	The View Image seen from viewpoint $\mathbf{E}$
Δu	Horizontal pixel offset from EI-center
Δv	Vertical pixel offset from EI-center
Λ	Distance between pixel- and lens array in II-camera
$\mathbf{Q}_{view}$	View image quality metric
$\mathbf{V}$	Location of virtual camera
$\mathbf{D}$	Depth map constructed using virtual camera
d	Depth layer of depth map $\mathbf{D}$
$\mathbf{T}^d$	Binary mask-image selecting which pixels belong to objects at d
$\mathbf{B}^d$	Base image constructed from densely located EIs
$\mathbf{R}^d$	Reference image constructed from dispersedly located EIs
$\mathbf{Q}_{depth}$	Depth-dependent quality metric
$\mathbf{E}_i$	View point location where $i = front, up, down, left, right$
I	Image synthesized using virtual camera

The $\widehat{\ }$ -operator denotes an image with coding-induced distortion.

Notation related to Coding

PVS	Pseudo Video Sequence
PVS	PVS-frame within the PVS
S, T	Number of pixels in PVS-frame horizontally and vertically
J	Number of PVS-frames
j	Index of PVS-framewithin the PVS
Γ	Permutation function defining a CISO
c_j	Cross-correlation coefficient between PVS-frame j and $j-1$
$\overline{c}$	Average c_j from all J PVS-frames
σ_c	Standard deviation of c_j
e	Difference residual
B	Size of II-picture in bits
r	Bitrate of II-picture in bits per pixel
b	Average size of PVS-frame in bits
f	Frame rate of PVS
$\bar{b}_h$	Portion of average size b corresponding to headers
$\overline{b}_{h,slice}$	Portion of $\bar{b}_h$ corresponding to slice headers

$\overline{b}_{h,macroblock}$	Portion of $\bar{b}_h$ corresponding to macroblock headers
MB	Macroblock size
$\bar{b}_d$	Portion of average size b corresponding to data
H	Header portion metric
T_c	Coding time
T_e	Encoding time
T_d	Decoding time
QC	Quality cost
ΔQ_{global}	Difference in Q_{global} between two coding schemes
C_T	Encoding time quotient between two coding schemes

Chapter 1

Introduction

An approach towards real life quality has been the aim for many telecommunication applications where image, video, or audio presentation form a part. In an ideal communication system, the participants could be separated by a large distance yet still perceive each other as if they were in the same room. Different applications have over the years evolved from being simple and rudimentary to providing almost lifelike presentations. Today we enjoy high definition color imaging contrary to the early days of video where quality was restricted to grainy black-and-white images. Despite the ongoing engineering efforts to enhance the quality of video, *depth* is a scene property that has been difficult to incorporate and reproduce in a satisfactory manner. Three-dimensional (3D) video has been pursued for decades as being the future of video communication.

Many 3D techniques have attempted to become a general 2D replacement but have failed. The main reasons for their lack of success have included the following:

- Not all of the requirements for natural depth perception have been fulfilled, causing eyestrain after extended periods of viewing.
- The introduced depth has negatively affected the quality of other properties such as image resolution and viewing angle.

As a compromise, different applications have used different 3D techniques for which the accompanying limitations have been acceptable within that particular context.

However, there are techniques that aspire to be *the* next 3D technique and thus are to be able to provide both an eyestrain-free lifelike depth-sensation while still retaining the qualities present in 2D techniques. Integral Imaging (II) is such a 3D technique, which is an extension of the photographic invention *integral photography*, made by Lippmann [1] in 1908. The idea was to place an array of tiny lenses – much like those in a fly's eye – on top of a photographic plate. Different perspectives of the scene are thereby decomposed and captured within the same photo. When this photo is later viewed through a similar lens array the different perspectives are

combined into a single 3D image that unfolds the depth of the depicted scene. It is mainly research and development on image capturing sensors and display panels throughout the past few decades that has made integral photography evolve from a photographic technique of the early 20th century into a potential solution for digital 3D images and -video applications of today. The photographic plate could now be replaced by an image capturing sensor containing a high resolution pixel array, e.g. a Charge-Coupled Device (CCD). The captured image could then be transmitted or stored for later presentation on a high resolution display, e.g. an Liquid Crystal Display (LCD)-panel.

Integral Imaging fulfills one of the major requirements placed on a future 3D technique: it is autostereoscopic. That is, the viewer is not required to wear any type of glasses in order to perceive the depth in the 3D image. The 3D effect is instead provided by the display itself. In addition, II also provides motion parallax, which means that the perspective of the depicted scene changes correctly according to the position of the viewer. Standing up while watching a scene with a coffee mug would actually allow the viewer to see if it is empty or not. Another consequence of the large set of perspectives of the depicted scene is that multiple users can share the viewing experience with the same ease as when viewing 2D video. These two properties differ considerably from the *stereoscopic* 3D techniques that require each user to wear special glasses that provides a single fixed viewing position, established at the time of capture. That is, regardless of how the viewer moves relative to the display, the same perspective of the depicted scene is perceived. This unnatural property of viewing is believed to be one of the reasons behind the induced eyestrain that may appear after long term use of stereoscopic techniques [2].

A number of modified II-techniques have emerged that enhance one or several of the II-properties, e.g., maximum viewing angle, depth fidelity, and spatial resolution of the 3D image. These techniques are relatively different from each other yet share the same fundamentals. However, comparing them against each other has proved to be somewhat difficult. Implementing an additional II-technique that is merely used for comparison purposes has not been practically feasible, due to the large amount of time and resources required in developing II-camera and -display prototypes. Instead, new II-techniques are often compared to an original form of II instead of to other similar II-techniques. This approach is not valid since the properties of new II-technique are dependent on each other, i.e. improving one degrades others. Hence, the only way to investigate such property-dependencies – and thereby the necessary trade-offs – is to compare similar II-techniques to each other.

1.1 Motivation

When studying the development of the II-research field from a signal and system perspective, an observation can be made. Despite its favorable properties, II does not come without a cost. Compared to present 2D images and 2D video, II requires an increased number of pixels, in both the image capturing sensor and the display panel, to retain the same spatial resolution in the reproduced 3D image. These ad-

ditional pixels are required to store the scene depth that is implicitly produced by the camera's lens array. Certainly this is a fact for all 3D techniques, to varying extents. However, to achieve good 3D fidelity – i.e. high resolution and precision and lack of eye fatigue – an increase in the number of pixels by a factor of 50 or more is suggested for II [2, 3]. Considering this in the light of the presently used video distribution channels for 2D, efficient coding and compression of both II-based 3D images and 3D video becomes important. Fiber optic networks with 40 Gbps links may even require coding, despite being considered bandwidth-abundant for today's multimedia applications. For example, the raw bitrate of a single II-based 3D video sequence with HD-resolution 1920×1080 for sharp images, 50 views for good 3D fidelity, and 25 frames per second for smooth playback would require ≈ 62 Gbps. Coding may also be viable for applications with less requirements in 3D fidelity. In order to provide a smooth transition from 2D to 3D, legacy equipment must also be able to extract suitable subsets of the 3D data. For example, set-top boxes that are only capable of decoding standard definition 2D video, would access and decode the base set of the 3D video. Fully 3D-enabled set-top boxes, would decode the complete sequence enabling the 3D video to be displayed on a 3D-enabled display. Moreover, the images captured by 3D camera systems might in some cases only be viewed on 2D presentation devices or on paper. Instead of storing one 2D version and a full 3D image, coding can efficiently combine the two and thereby more efficiently use the available storage. Hence, introducing coding for II-based 3D images enables a trade-off between storage requirements and computational complexity. This trade-off will be differently balanced depending on the application it applies to.

However, before any 3D images coding scheme can be developed either

- all relevant 3D images properties must not be subject to change or
- the coding scheme must be adaptable to a wide range of available 3D images produced by different 3D techniques.

Unfortunately, neither of these demands is fulfilled in the case of II. New and enhanced II-techniques continue to emerge and no standardized image format exists at present. It is also uncommon for proposed II-techniques to be accompanied by numerous 3D images, which are available for general widespread use. Nevertheless, easy access to different 3D images is vital to make research in II-based 3D images processing feasible.

1.2 Overall aim and problem definitions

The work presented in this dissertation is based on two defined goals, each further divided into two verifiable problem statements:

G1 Provide means for simple production of II-signals, depicting strictly defined scenes that can be easily adapted to different II-techniques.

P1a How can the scene, the II-system, and the II-signal be decoupled to aid the comparison of II-signals produced by different II-techniques?

P1b Can such a decoupling be used to provide a supply of II-signals, which for example would facilitate research on coding methods for II-signals?

G2 Propose a coding scheme for II-signals that allows for a variable trade-off between coding efficiency and coding introduced distortion.

P2a How can the II-signal be coded such that a more compression efficient representation is achieved than that possible with existing coding methods?

P2b What consequences will a proposed coding method have on objective quality?

The questions posed in P1a – P2b are somewhat generally formulated. The following clarifications make them more specific.

- The II-signal will, for this study, refer to a static Integral Imaging picture, and not an II-based 3D video sequence. 3D video is – analog to its 2D counterpart – merely a set of consecutive pictures shown one after another at a sufficiently high pace. The knowledge about static Integral Imaging pictures is therefore a vital prerequisite that can be extended at a later stage to II-based 3D video by also considering time.
- The term *coding efficient* refers to the quantity of data required to represent the coded Integral Imaging picture relative to the coding-induced distortion. A coding scheme that produces a coded Integral Imaging picture that requires small amounts of data, while still inducing only minor distortion, has a high coding efficiency. Compression efficient is a synonymous term.

1.3 Solution approach

At a first glance it might appear that one way to fulfill Goal G1 would be to formalize a test procedure. This is only true if either each new II-camera prototype is transported to a set of commonly decided and defined scenes; or that *all* properties of these scenes are strictly defined such that they can be rebuilt at the location of the II-camera prototype. Unfortunately, both alternatives require too large an investment in both time and physical resources to be feasible in real-life. However, this would not prove to be infeasible if the transport of an II-camera was to be free of cost or if defining all aspects of a scene was possible. Computer simulation makes these somewhat utopian ideas possible, as the physical objects become virtual and the transfer of atoms is changed into setting electrons in motion. All aspects could be explicitly and exactly defined and research on each subsystem could be performed simultaneously by separately modeling the II-based 3D image, the II-system, and the scene to be depicted. For example, new 3D images depicting a given scene could

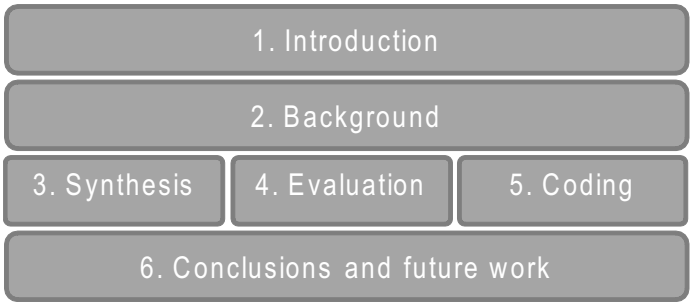


Figure 1.1: A graphical representation of the dissertation's outline and its constituting parts

be rendered by merely interchanging the II-system model while retaining the scene model. Or, II-based 3D images could be produced with the characteristics of any II-technique as soon as an II-system model is defined, which facilitates comparative research where similar II-techniques are evaluated on pre-defined scene models.

The manner in which to approach problems P2a – P2b, is to a large extent, universal with reference to the coding of signals with the intent to compress:

- identify the signal characteristics
- find a form in which the identified signal redundancy is efficiently revealed
- reduce this redundancy while limiting the introduced perceived distortion.

I will attack both problem pairs (P1a – P1b and P2a – P2b) using the same approach: to an as large extent as possible leverage on achievements in open-source software projects and state-of-the-art multimedia coding standards.

In the chapters of this dissertation I will answer the above stated questions and present solutions that achieve the two goals.

1.4 Outline

The dissertation is divided according to the two goals in Section 1.2. Figure 1.1 shows a graphical illustration of the content and the constituent chapters' interrelationships. Following this introduction, Chapter 2 provides a more detailed theoretical background, which fulfills three purposes. Firstly, a basis regarding the concepts and properties of 3D images is provided, coupled with an outline of different 3D techniques and their pros and cons. Secondly, the specific 3D technique Integral Imaging is studied in detail. Thirdly, related works are presented with respect to the three areas of the thesis, namely synthesis, evaluation, and coding.

After the background in Chapter 2, the specific problem statements are addressed in detail. Problem statement P1 is discussed in Chapter 3. Section 3.3 addresses P1a

by proposing a II-camera model as a means of separating scene, II-based 3D images, and II-camera. Within the context of the problem statement P1b, Section 3.4 presents a ray-tracing based software application that uses the II-camera model for synthesizing II-based 3D images. Chapter 4 bridges the two problem areas synthesis and coding together with a discussion concerning the evaluation of II-based 3D images. Chapter 5 addresses problem statements P2a and P2b. A coding scheme for II-based 3D images is proposed that utilizes 2D video coding algorithms to uncover and reduce the redundancy of the 3D image. The II-coding scheme is parameterized, which results in different variants that are evaluated and compared against existing coding methods. Finally, Chapter 6, concludes the work presented and discusses areas of future study.

1.5 Contributions

The content of this dissertation is based mainly on the previously listed papers I to VIII – in which I have performed the greater part of the development, simulation, evaluation, analysis and presentation. The contributions can be divided into three main parts:

1. A synthesis tool that can generate II-based 3D images and 3D video using an II-camera model capable of describing different II-techniques.
2. Two quality evaluation metrics that are capable of quantifying coding-induced distortion present in an II-based 3D image.
3. A coding scheme for II-based 3D images that provides a coding efficiency vastly exceeding previous works.

The first part, addressed in Paper I and Paper II, describes an II-camera description model and a ray-tracing based synthesis tool. Taken together they enable the easy generation of II-based 3D images and 3D video, which assists in the development, evaluation and quick adoption of the new emerging II-techniques.

The second part describes two novel quality evaluation metrics and has in part been presented in Papers IV – VI. The first metric models the view perceived by a viewer watching an II-based 3D display from different angles. The second metric quantifies the 3D image's quality at different depths. The metrics are evaluated on a set of coded 3D images and the results are compared both with previously proposed quality metrics and with a visual inspection of the introduced coding artifacts.

The third part discusses a coding scheme that was first presented in paper III and further expanded in Paper IV. An II-based 3D images is transformed into a form that essentially resembles a 2D video sequence. This sequence is then coded using the 2D video coding standard H.264/MPEG-4 AVC. Different parameterizations of the scheme are studied in addition to its effect on the quality of the decoded II-picture. The scheme's relation to current state-of-the-art 2D images coding approaches and similar Integral Imaging picture coding schemes are studied, which has been presented in Papers VII – VIII.

1.5.1 Condensed abstracts from the selected papers

1.5.1.1 Paper I

The next evolutionary step in enhancing video communication fidelity over wired and wireless networks is taken by adding scene depth. Three-dimensional video using integral imaging based capture and display subsystems have shown promising results and are now in the early prototype stage. To assist in the development, evaluation and adoption of these new emerging techniques an effort to create a ray-tracing based interactive simulation environment to generate integral imaging source material has been initiated and is described in this paper. As a base for the simulation environment a generic integral imaging description model is presented. This model is designed to facilitate optically accurate rendering using the open-source ray tracing package MegaPOV, which fully incorporates the POV-Ray scene description language to exactly define all scene properties compared to experimental research. The simulation environment's potential for easy deployment of three-dimensional source material, adhering to a few different integral imaging variants, is demonstrated.

1.5.1.2 Paper II

Three-dimensional video using integral imaging (II) based capture and display subsystems are now in the early prototype stage. We have created a ray-tracing based simulation tool to generate II-based 3D video sequences. Such a tool would assist in the development, evaluation and quick adoption of these new emerging techniques into the whole communication chain. A generic 3D camera description model, which is the base for the II-synthesis tool, is also described. This description model allows for optically accurate synthesis of II-signals using MegaPOV, which is a customized version of the open-source ray tracing package POV-Ray. The scene description language of POV-Ray can then be used to exactly define a virtual scene. The initial development of the II-synthesis tool is focused on producing and visualizing II-signals, which adheres to the optical properties of different II-techniques published in the literature. Both temporally static as well as dynamic systems are considered.

1.5.1.3 Paper III

The next evolutionary step in enhancing video communication fidelity is taken by adding scene depth. 3D video using integral imaging (II) is widely considered as the technique able to take this step. However, an increase in spatial resolution of several orders of magnitude from today's 2D video is required to provide a sufficient depth fidelity, which includes motion parallax. In this paper we propose a pre-processing and compression scheme that aims to enhance the compression efficiency of integral images. We first transform a still integral image into a pseudo video sequence consisting of sub-images, which is then compressed using an H.264 video encoder. The improvement in compression efficiency of using this scheme is evaluated and presented. An average PSNR increase of 5.7 dB or more, compared to JPEG 2000, is observed on a set of reference images.

1.5.1.4 Paper IV

To provide sufficient 3D depth fidelity, integral imaging (II) requires an increase in spatial resolution of several orders of magnitude from today's 2D images. We have recently proposed a pre-processing and compression scheme for still II-frames based on forming a pseudo video sequence (PVS) from sub images (SI), which is later coded using the H.264/MPEG-4 AVC video coding standard. The scheme has shown good performance on a set of reference images. In this paper we first investigate and present how five different ways to select the SIs when forming the PVS affect the schemes compression efficiency. We also study how the II-frame structure relates to the performance of a PVS coding scheme. Finally we examine the nature of the coding artifacts which are specific to the evaluated PVS-schemes. We can conclude that for all except the most complex reference image, all evaluated SI selection orders significantly outperforms JPEG 2000 where compression ratios of up to 342:1, while still keeping PSNR > 30 dB, is achieved. We can also confirm that when selecting PVS-scheme, the scheme which results in a higher PVS-picture resolution should be preferred to maximize compression efficiency. Our study of the coded II-frames also indicates that the SI-based PVS, contrary to other PVS schemes, tends to distribute its coding artifacts more homogenously over all 3D scene depths.

1.5.1.5 Paper V

The two-dimensional quality metric Peak-Signal-To-Noise-Ratio (PSNR) is often used to evaluate the quality of coding schemes for 3D images based on integral imaging (II). The PSNR may be applied to the full II resulting in single accumulate quality metric covering all possible views. Alternatively, it may be applied to each view results in a metric depending on viewing angle. However, both of these approaches fail to capture a coding scheme's distribution of artifacts at different depths within the 3D image. In this paper we propose a metric that determines the 3D images quality at different depths. First we introduce this 1D measure, and the operations that it is based on, followed by the experimental setup used to evaluate it. Finally, the metric is evaluated on a set of 3D images; each coded using four different coding schemes and compared with visual inspection of the introduced coding distortion. The results indicate a good correlation with the coding artifacts and their distribution over different depths.

1.5.1.6 Paper VI

The two-dimensional quality metric Peak-Signal-To-Noise-Ratio (PSNR) is often used to evaluate the quality of coding schemes for different types of light field based 3D images, e.g. integral imaging or multi-view. The metric results in a single accumulated quality-value for the whole 3D image. Evaluating single views – seen from specific viewing angles – gives a quality matrix that present the 3D images quality as a function of viewing angle. These two approaches do not capture all aspects of the induced distortion in a coded 3D image. We have previously shown coding schemes of similar kind for which coding artifacts are distributed differently with respect to the 3D image's depth. In this paper we propose a metric that captures the depth distribution of coding-induced distortion. Each element in the resulting quality vector corresponds to the quality at a specific depth. First we introduce the pro-

posed full-reference metric and the operations on which it is based. Second, the experimental setup is presented. Finally, the metric is evaluated on a set of differently coded 3D images and the results are compared, both with previously proposed quality metrics and with visual inspection.

1.5.1.7 Paper VII

Novel camera systems producing 3D images containing light direction in addition to light intensity is emerging. Integral imaging (II) is a technique on which many of these systems rely. The pictures produced by these cameras (II-pictures) are space-requiring in terms of data storage compared to their 2D counterparts. This paper investigates how coding the II-pictures using H.264 / AVC and JPEG 2000 Part 10 (JP3D) affect the images in terms of rate-distortion as well as introduced coding artifacts. A set of four reference images are coded using a number of pre-processing and encoding variants, so called coding schemes. For low bitrates (<0.5bpp) the H.264 / AVC-based coding schemes have higher coding efficiency, which asymptotically level of at higher bitrates in favor of JP3D. The JP3D coded 3D images show less spread in quality than H.264 / AVC, when PSNR as a function of viewing angle is evaluated. However, the distortion induced by H.264 / AVC is primarily localized to object boarders within the 3D image, which in initial tests appear less visible than the JP3D coding artifacts that spread out evenly over the image. Extensive subjective tests will be performed in future work to further support the presented results.

1.5.1.8 Paper VIII

Revolutionary camera systems based on the 3D imaging technique Integral Imaging (II) have been presented in recent years, which exceed the capture possibilities of conventional cameras. The properties of the captured II-pictures mean that they are restricted in their ability to be efficiently compressed using 2D images compression methods. In this paper we propose a pre-processing and coding scheme that compresses the II-pictures using state-of-the-art video or volumetric image coding standards H.264 / AVC and JPEG2000 Part 10 3D (JP3D). The II-picture is firstly transformed into a pseudo video sequence (PVS) or a pseudo volumetric image (PVI), which is later encoded using H.264 / AVC or JP3D. We compare the proposed coding scheme with the 2D images coding standard JPEG 2000 and other coding schemes for II-pictures. Our results show that the proposed solution vastly outperforms JPEG2000 by more than 10 dB in Peak-Signal-to-Noise-Ratio (PSNR) when applied to a set of II-pictures. In addition, the proposed scheme produces equal distortion levels at a 60-th of the bitrate required by JPEG2000 at similar cost in CPU-time. The paper further elaborates on the parameterizations of the proposed coding schemes and how it affects image quality.

Chapter 2

Background

2.1 Three-dimensional images

2.1.1 History

Humans have always wanted to depict their surroundings as accurately as possible, ever since the first man-made image was engraved and painted on a cave wall more than 30 000 years ago. Over time – as different theories about the world have emerged – artists have been provided with tools to make the colored 2D surface appear more and more like a window into the depicted 3D scene. The technique behind properly using perspective was the result of several centuries of thought, beginning as early as the 11th century when the Arabian mathematician and philosopher Alhazen published his book *Optics* [4, 5]. However, it did not become applicable to the world of arts until 400 years later, when the scientific knowledge concerning linear perspective was rediscovered and formulated. The Italian architect and Renaissance man Filippo Brunelleschi formulated a set of drawing principles, e.g. the vanishing point where all parallel lines on a plane in 3D converge to a specific point when projected to a 2D surface [6]. Before that time, different object depths were mainly illustrated using occlusion, i.e. near objects occlude objects further away when positioned in the line of sight.

With the invention of photography at the beginning of the 19th century, a correctly depicted 3D scene now became a matter of pressing the shutter button of the camera and developing the photo. However, even though this method allows the viewer to see an exact snapshot of a real-life scene, one significant difference exists. Both eyes have the same perspective of the depicted scene when viewing the photo. In real-life, the left and the right eye view the scene from a slightly different perspective due to the distance between the pupils. This discrepancy was already recognized by the French painter Bois-Clair in 1692 [7, 8]. His solution to the problem was to combine two paintings, interlaced in thin vertical stripes, and attach them to a set of vertically aligned opaque rods. A particular painting could only be seen

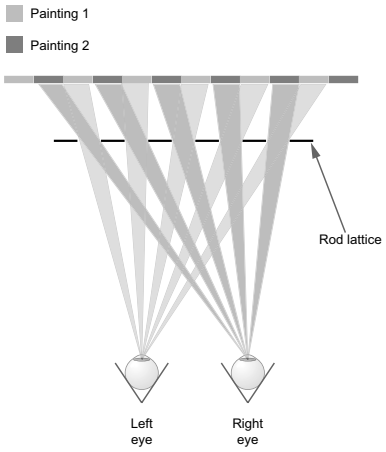


Figure 2.1: Interlaced paintings combined with a lattice of opaque rods.

from certain viewing points when the combined painting is seen through this lattice of rods. From other view points the painting is occluded by the opaque rods. Two different perspectives of the depicted scene could be viewed simultaneously by the left and right eye by carefully selecting the viewing position based on the interlacing and lattice characteristics. An example of a combined painting and a rod lattice is illustrated in Figure 2.1. Sir Charles Wheatstone later formalized the knowledge for binocular vision in 1838 [9]. At the same time he also presented the stereoscope, the first hand-held device to direct two separate images to the left and right eye respectively. This invention was followed by a number of similar devices during the latter half of the 20th century [10]. Some of those techniques are described in Section 2.1.3.1. More advanced techniques were presented in the first half of the 20th century, in which integral photography and holography are two prominent examples. The latter is briefly described in Section 2.1.3.2, whereas the former is discussed in detail in Section 2.2. Hence, several steps have been taken towards enhancing man-made depictions of the real world. However, despite the many efforts made, work still remains before the depiction will contain the ideal characteristics that makes it resemble a plain glass window into the depicted 3D scene.

2.1.2 Human visual system requirements

Thus, it is evident that there are several aspects in the Human Visual System (HVS) that must interact to provide a solid depth perception. These so-called depth cues, can be categorized into either psychological high-level cognitive cues or physiological low-level sensory cues depending on where their primary site of operation is within the HVS [11]. In the following two sections, an overview is given for the depth cues widely considered to be most important [2, 12–14].

2.1.2.1 Psychological depth cues

A large number of depth cues can be triggered even in a 2D painting, by using higher cognitive processes:

Occlusion From experience we know that if object B is partially covered by object A, object A is closer to us than object B. This is one of the strongest depth cues that, if not fulfilled, makes a correct depth perception very difficult. Occlusion is illustrated in Figure 2.2 (a), in which the only difference between the left and right image is which object's circumference that is broken.

Light and shadowing The squared fall-off rate of light power might not be commonly dwelled upon when viewing the world. However, the properties of light and the characteristics of the shadows cast by lit objects are two very important depth cues. In Figure 2.2 (b) the objects from Figure 2.2 (a) are lit and shadows are cast in the left and right image respectively. As a result, the shape and position of the scene objects appear more clearly.

Linear perspective Our intuition – relating to the knowledge about venturing points, converging parallel lines etc. – assists us in determining scene depth. Compare the left and right image in Figure 2.2 (c). The left looks less natural, which is the result of it not being produced with linear perspective but is instead using orthographic projection.

Texture gradient As the depth of a textured object is increased, the texture of its projection becomes of increasingly high-frequency, i.e. the texture gradient increases. In Figure 2.2 (c), this is evident in the floor tiling.

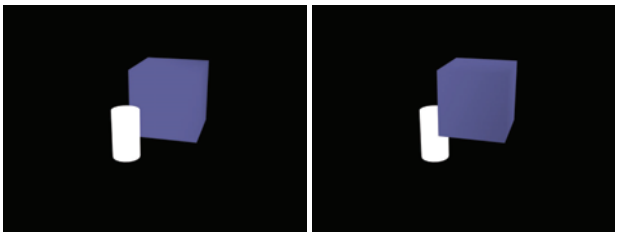
Retinal size Knowing an object's size relative to other objects, coupled with linear perspective, aids in determining the distance to the scene objects. Figure 2.2 (d) illustrates the change in the perceived distance to the scene as a result of known objects sized.

Air perspective contrast As light travels through an environment it is not affected in a homogeneous manner. For example, different wavelengths are attenuated more than others. If objects are located at a long distance from the viewer, the colors become desaturated and the contrast is reduced. An example of this is the gray or blue tint of distant mountains, which is evident in many landscape paintings or photos. Figure 3.7 (d) on page 66 gives an example of decaying contrast for distant objects.

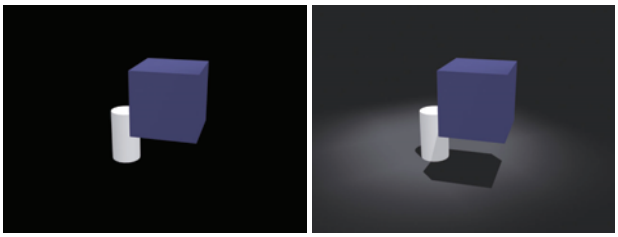
These high-level cues are also referred to as *monocular* depth cues [11], since they provide partial depth perception even from viewing a 3D scene with only one eye, or when viewing a 2D depiction with both eyes.

2.1.2.2 Physiological depth cues

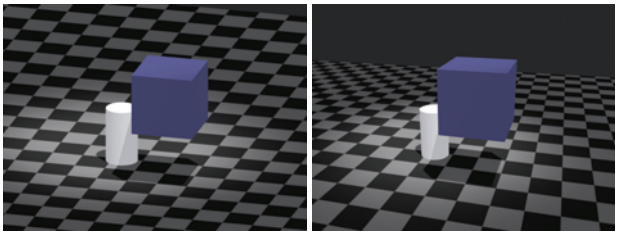
The low-level cues enumerated below act on a more fundamental level of the HVS, i.e. they are directly related to the physical aspects of the eye's optical system. This



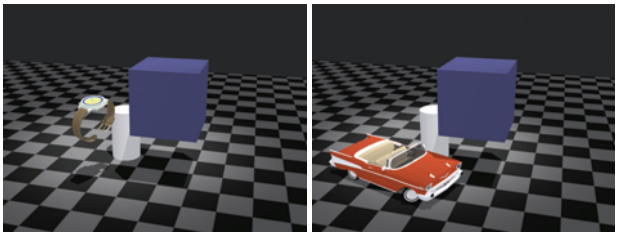
(a)



(b)



(c)



(d)

Figure 2.2: Examples of high-level depth cues illustrating (a) occlusion, (b) light and shadowing, (c) linear perspective and texture gradient and (d) retinal size.

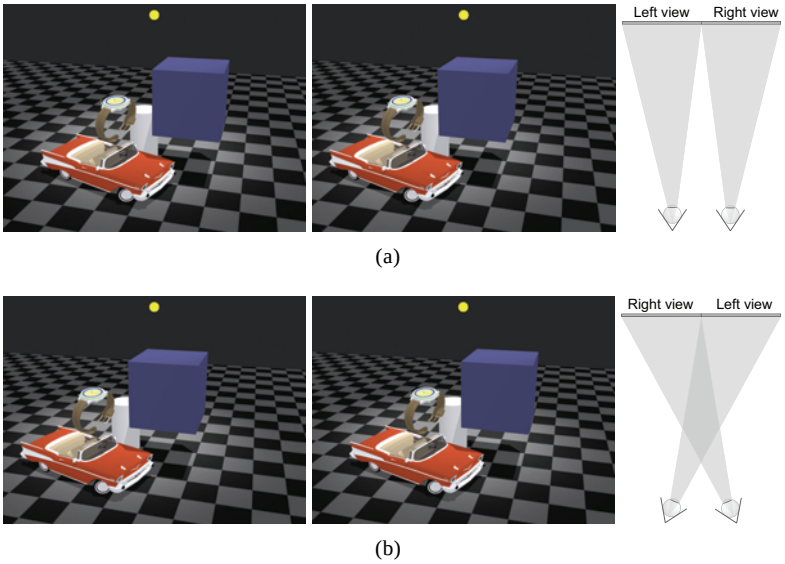


Figure 2.3: Examples of binocular parallax for (a) parallel viewing and (b) cross-eyed viewing.

makes them inaccessible when producing 2D depictions. However, they are considered vital for providing an exact depiction of a 3D scene [10].

Binocular parallax This *binocular* depth cue is triggered by the disparity introduced in the two captured projections of the left and right retina respectively. The HVS processes these two slightly different images for which a smaller disparity causes the perception of greater depth. Figure 2.3 shows two views from slightly different perspectives. The views can be perceived without any aid by merely fixating each view pair such that the correct view is seen by the corresponding eye. The left and right views in Figure 2.3 (a) allow for parallel free-viewing whereas the right and left views of Figure 2.3 (b) are arranged to be viewed using cross-eyed viewing.

Accommodation When focusing the eyes on different distances within a 3D scene, accommodation is triggered. Muscles in the eye pull the lens, which causes it to change its thickness and thereby its optical power. This is illustrated in Figure 2.4 (a), where two objects at different distances cause the lens to change thickness and thereby focus.

Convergence Convergence is the second binocular depth cue. It is triggered by the rotation of the left and right eyes when fixing upon a specific point in the 3D scene. Distant objects introduce less convergence while close objects introduce more. Figure 2.4 (b) shows two examples of this.

Motion parallax When changing the view point – relative to the 3D scene – the object projections translate on the retina. For distant objects the speed of this

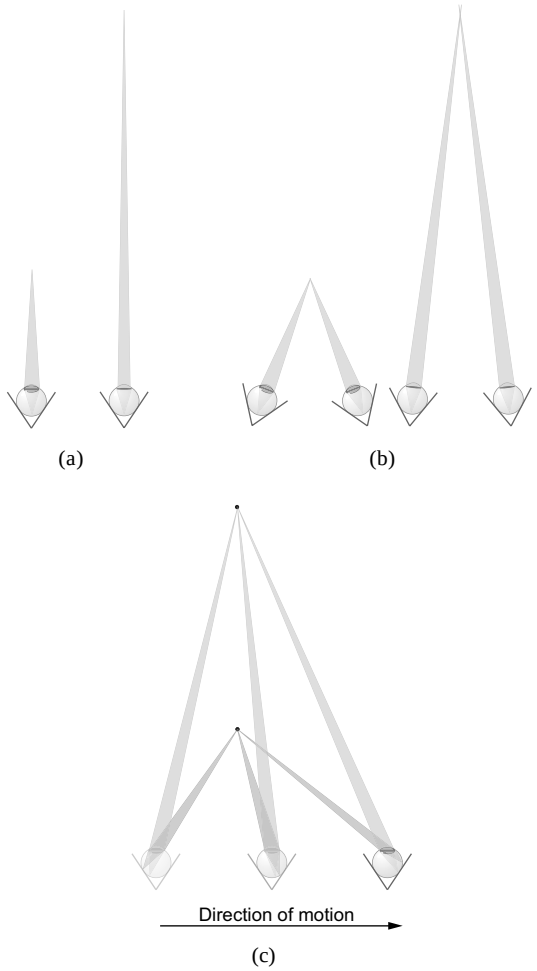


Figure 2.4: Examples of low-level depth cues illustrating (a) accommodation, (b) convergence and (c) motion parallax.

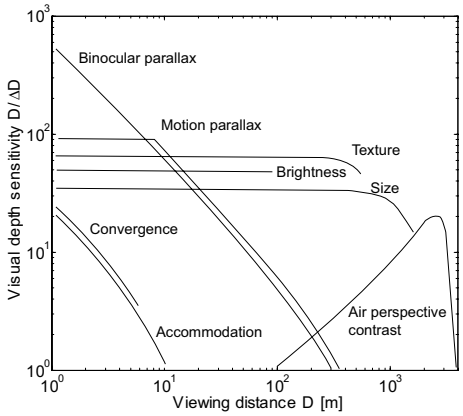


Figure 2.5: Depth-cue sensitivity over different viewing ranges [11].

translation is slower than for objects that are close to the viewer. An example of motion parallax is shown in Figure 2.4 (c), where the a single eye is looking at two objects while moving to the right. Note the difference in distance that the two objects' projections travel on the retina as a function of their distance to the viewer.

The individual importance of each depth cue or how strongly each cue contributes to the total depth perception, depends on the distance at which the objects are located. Figure 2.5, adopted from the work of Siegel and Nagata [11], illustrates this by plotting the visual depth sensitivity for some of the described depth cues. The visual depth sensitivity of a specific cue is defined as $\frac{D}{\Delta D}$, where ΔD is the minimum change in distance that can be visually perceived for the cue at distance D. A cue with low visual depth sensitivity contributes less to the depth perception of an object at a specific distance than a cue with high visual sensitivity. The low-level cues are evidently important for the majority of 3D applications since the distance between display and viewer rarely exceeds the range of 0–10 meters in which these cues dominate. Several of the mentioned depth cues also interact due to an inherent interdependency. For example, when fixing an object in 3D space by converging the eyes to that point, accommodation simultaneously puts the object in focus and vice versa. The importance of providing the two binocular depth cues has also been shown in subjective evaluations.

Motoki et al. [15] have shown that 3D presentation techniques clearly outperform their 2D counterparts, by measuring factors such as sensation of depth, sharpness, naturalness and total image quality. The increased power of 3D presentation techniques compared to 2D is further confirmed by objective evaluation of psychological effects such as induced body sway [2]. Fulfilling only a subset of all depth cues might be sufficient for certain applications. However, the main reason for the induced eye fatigue caused by non-ideal 3D techniques is believed to be the inability to fully provide all depth cues and their interrelations [2].

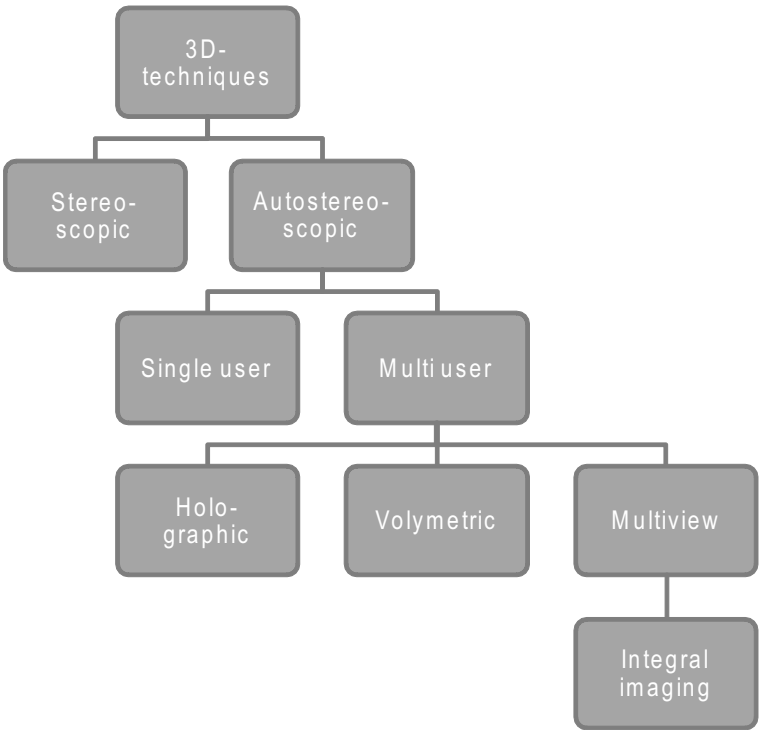


Figure 2.6: A non-exhaustive taxonomy of 3D techniques, which defines integral imaging as a multi-view, multi-user autostereoscopic 3D technique.

2.1.3 3D-techniques

Even though the problem that a successful 3D technique must address is clearly defined – provide as many of the depth cues, and their interdependencies, as possible – the methods used for solving the problem differ. However, a few specific approaches exist into which different methods can be categorized. Figure 2.6 presents a simple 3D technique taxonomy, which will be briefly described in Section 2.1.3.1 and 2.1.3.2. For additional information on different 3D techniques the reader is referred to the surveys presented in [16, 17, 10].

2.1.3.1 Stereoscopic

Stereoscopic 3D techniques rely on some kind of user worn equipment for providing the perception of depth. Different means exist to relay the separate views to each eye:

- Anaglyph
- Polarization

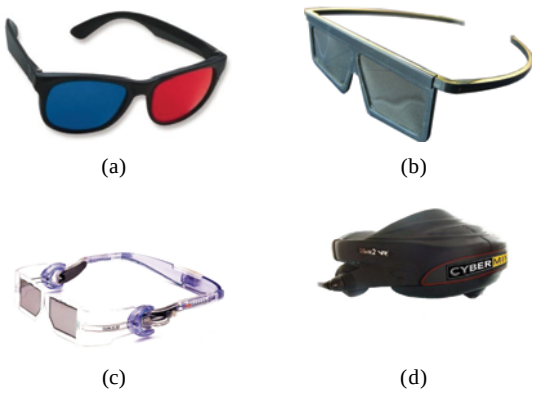


Figure 2.7: Sterescopic glasses for (a) anaglyph [18], (b) polarization [19], (c) time-sequential [20] and (d) head-mounted display [21].

- Time multiplexing
- Spatial multiplexing

Anaglyph techniques use complementary colors to separate the two different views, as illustrated in Figure 2.7 (a). Red is often used for the left view, which makes cyan the color used for the right view. Often one of the cyan components, green or blue, is used instead. The colors of the depiction are naturally affected by the use of color to separate the views.

If perpendicular polarization is used to separate the views instead of color, each view can preserve its original color. The two views are demultiplexed using low cost passive glasses. An example of polarized glasses is shown in Figure 2.7 (b).

The anaglyph and polarization techniques provide the two views simultaneously. Time sequential techniques, on the other hand, display the views in succession, one after the other at a rate higher than the flicker fusion rate. This requires glasses that actively alter the transparency of the lenses in synchronization with the display. The left eye is occluded when the right view is presented and vice versa. Figure 2.7 (c) shows an example of such a pair, which uses LCD lenses to occlude each eye.

Spatially separating the two views is also a possibility as shown in Figure 2.7 (d) where the glasses or goggles contain one LCD-display a few centimeters in front of each eye.

There are four main problems with the stereoscopic techniques, which make them unfavorable for general 3D use:

1. The user is required to wear some kind of goggles to perceive 3D.
2. There is a discrepancy introduced between convergence and accommodation depth cues. For example, even though the binocular parallax and convergence

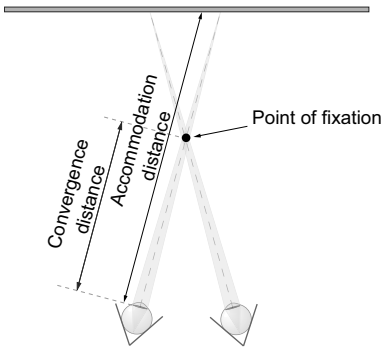


Figure 2.8: Convergence and accommodation discrepancy.

might position a depicted object in front of the display, the eyes must focus on the display to make the object sharp. Figure 2.8 illustrates this.

3. Various degrees of cross-talk is produced. That is, parts of the left view leaks over into the right view and vice versa. Cross talk can however be completely removed by separating the views spatially. Still, this would require the troublesome setup of having to wear a head-mounted display such as that in Figure 2.7 (d).

4. The motion parallax depth cue is not provided as there are only two views of the depicted 3D scene shown. Even if an extensive look-around capability might only be required in specific 3D applications, the fact that the slight body sway of the viewer is not reflected in the scene perceptive can become annoying and tiresome.

Motion parallax can be supported in stereoscopic techniques by using active tracking of the eyes' position and gaze direction. Based on this gathered information the two views are updated accordingly. This enhancement is referred to as active stereoscopy and for it to be effective the views must be updated quickly. If not, the motion parallax effect will be subjected to a delay that still could cause nausea. Hence, active stereoscopy requires solving two relatively costly and complex problems while still not being able to provide all the required physiological depth cues.

2.1.3.2 Autostereoscopic

Instead of requiring goggles, autostereoscopic techniques rely on added complexity in the display to provide a 3D image.

Single user autostereoscopic techniques are very similar to the active stereoscopic approach. The main difference is that the complexity is transferred from the glasses to the display. Hence, the user is not required to wear any goggles. However, for passive single user autostereoscopic techniques the user is instead restricted to a

specific position in order to perceive the 3D effect. An active display with additional complexity can solve this by tracking the viewer and guide the two views toward the user's current position. With a sufficient optical guiding system, more than one viewer can be tracked. However, similar to active stereoscopy the cost of this solution does not scale well when the number of viewers is increased.

Multi-user autostereoscopic techniques are designed to allow for any number of viewers to see 3D simultaneously. Holography is perhaps the most well known technique because of its flawless reproduction of all of the four physiological depth cues. However, due to its high demand on pixel resolution it will, for the foreseeable future, remain a photographic technique [10]. This, together with the inherent requirement of having the 3D scene set up in 1:1 scale in a darkened studio lit only with one ore more monochromatic lasers – if the scene is not computer generated – further prohibits it from being the next 3D technique [22]. Figure 2.9 (a) shows a "camera setup" using a single laser for a monochrome capture, where the banana in the middle is the depicted object.

Volumetric techniques construct a display volume in which the depicted scene is shown. The level of eye fatigue for volumetric techniques is also low given that all of the four low-level depth cues are provided. Numerous methods exist to create this display volume [10]. One method is to stack semi-transparent 2D display panels next to each other and displaying slices of the scene on them [23]. Another method is to alter the rotation angle of a single display panel which has the obvious disadvantage of high speed [24]. The inherent discrete nature of the volume slicing can be somewhat mitigated by proper anti-aliasing of the scene space. Examples of these two methods are shown in Figure 2.9 (b). Despite the apparently advantageous property of providing all low-level depth cues, volumetric techniques are unable to become a general 2D replacement in their present form. This is because the occlusion depth cue can not be supported from an arbitrary view point, a consequence of the semi-transparent sub-components of the display. That is, a scene depicted by a volumetric autostereoscopic technique is always somewhat transparent, regardless of the extent of transparency in the real 3D scene [12].

Multi-view techniques, like the stereographic techniques, rely on disparity between images in order to provide depth cues. However, for multi-view techniques more than two views are combined and distributed into the viewing space using different types of view-forming optics attached to the display panel. Several views corresponding to different scene perspectives allow for varying degrees of look-around capability, as Figure 2.9 (c) illustrates. Multi-view is considered to be the technique with the greatest prospects [10]. In part because, given proper parametrization, multi-view promises to provide *all* high- and low-level depth cues without the restrictions on scene setup implied by holography. It also lends itself to an easier migration path from present 2D techniques, compared to holographic and volumetric techniques, since the capturing stage consists of a set of 2D cameras. Multi-view techniques that provide Horizontal Parallax Only (HPO) arrange this set of cameras in a horizontal line with parallel optical axes. The set of cameras for full parallax form an often planar surface. A thorough presentation of the majority of aspects regarding multi-view techniques is given by Son and Javidi [25]. Given its close re-

lationship to II this autostereoscopic technique will be further discussed in Section 2.2.

2.1.4 Applications

There is a broad range of data visualization fields with applications that would benefit from 3D imaging and 3D images.

- Medical imaging
- Marketing
- Design and construction
- Entertainment
- Complex visualization
- Enhanced 2D images

One envisioned application in the field of medicine is a system to aid surgery that overlays data from different sensors such as video, x-ray, Magnetic Resonance Imaging (MRI) and ultrasound [27]. The surgical operation is greatly facilitated by having the compounded data presented in 3D.

Digital posters, or digital signs, are becoming an important advertising channel. In marketing, every means of catching the attention of shoppers is of interest. Adding 3D to the digital signs allows for the sign content to not only update but literally extend out from the poster.

The gap between design/construction and manufacture/operation of a product could be greatly reduced when the whole development process is performed in 3D. Many mistakes or miscalculations could be avoided early in the design process if designers could study, walk around and in detail investigate a 3D model of a future product as if it was real using a full 3D Computer Aided Design / Computer Aided Engineering (CAD/CAE) application.

The applications with the broadest user base are those within the consumer market. Entertainment applications such as 3DTV and 3D computer games are all important driving forces for the development of 3D techniques. Companies such as Philips, Sharp, LG and Samsung and more are starting to release autostereoscopic 3D displays for the whole spectrum of the consumer market [26, 28, 29].

Visualization applications that would benefit from 3D are many. Studying the high dimensional simulation results of various complex phenomena is one. Another might be to present meteorological or oil prospecting data as accurately as possible. An essential overview might become lost when the data sets are projected down to fit a 2D display. Hence, in the same way that color visualization provides more information than monochrome, a 3D display allows a more truthful visualization than 2D.

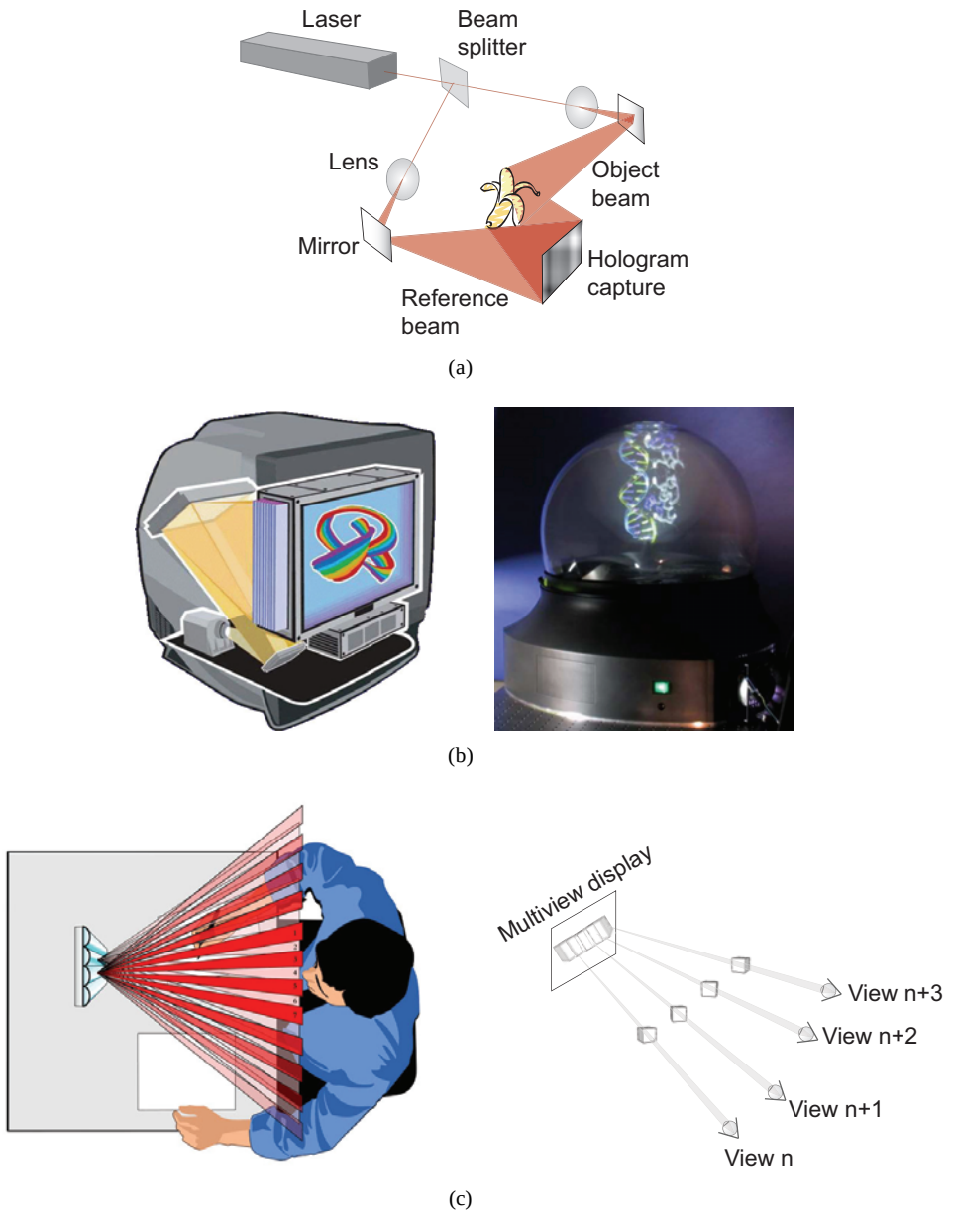


Figure 2.9: Autostereoscopic techniques showing (a) a holographic capture setup, (b) rotating and stacked volumetric displays [23, 24] and (c) viewing zones from a multi-view display [26].

Another implicit indicator of growing number of future 3D applications is the ongoing standardization work with the Joint Video Team (JVT)of ISO / IEC (Moving Picture Experts Group (MPEG)) and ITU-T (Video Coding Experts Group (VCEG)). Standards for both 3D video and free-view video are being worked on, which lays the groundwork for enhancing today's 2D video applications with depth and also introducing new novel 3D applications.

The comprehensive information captured within a 3D images of the type presented in the next section may also be used for purposes other than to show the depicted scene in 3D. Computational imaging is an emerging field where 2D images are synthesized in digital post-processing producing 2D images that could not have been captured by conventional 2D cameras [30]. Virtual optical systems may be simulated that allow for the altering of the view point, the focal plane, the aperture size etc., *after* the image has been captured.

2.2 Integral Imaging (II)

2.2.1 History

When Gabriel Lippmann proposed Integral Photography (IP) in 1908 it was another step in the quest to depict a 3D scene as accurately as possible [1]. It surpassed Frederick E. Ives' parallax stereogram – which was presented five years earlier – by providing depictions which captured both horizontal *and* vertical parallax [8]. The increased number of views also allowed for motion parallax and by adopting lenses in the capture and display process – instead of barriers – an increased optical efficiency was achieved. Herbert E. Ives' – who shared father with the parallax stereogram – simplified Lippmann's work in 1931 by replacing the lens array with vertically aligned cylindrical lenses that sacrifice vertical parallax [31]. An approach which could be seen as a more optically efficient extension of the parallax panoramagram patented in 1918 by Clarence W. Kanolt [8, 32]. Then the focus on stereoscopic techniques postponed theoretical studies of IP until the late 1960s [33, 34, 16]. The recent decade's success in digitizing the photographic process led to the necessity to generalize the concept of IP and resulted in the adoption of the term II [32]. The activity rate within the field of IP and II is shown in Figure 2.10, which extrapolates the curve presented by Okoshi [16] in 1980. As shown, the theoretical activities are now being coupled with practical advancements in inter alia image sensor resolution and lens array optics, which when combined, suggest that a large scale break through for II is imminent. Novel II-based camera systems have, in recent years, been presented that enables the capture of a larger portion of scene-reflected light than that possible using conventional cameras [35–38]. Images captured using these camera systems are shown in Figure 2.11.

Integral Imaging allows for the capture of both light intensity *and* direction as the single large aperture is replaced by a multitude of smaller apertures. The 3D image's extended information may be used in a wide range of post-processing operations to synthetically re-locate the focal plane, re-design the aperture, re-position

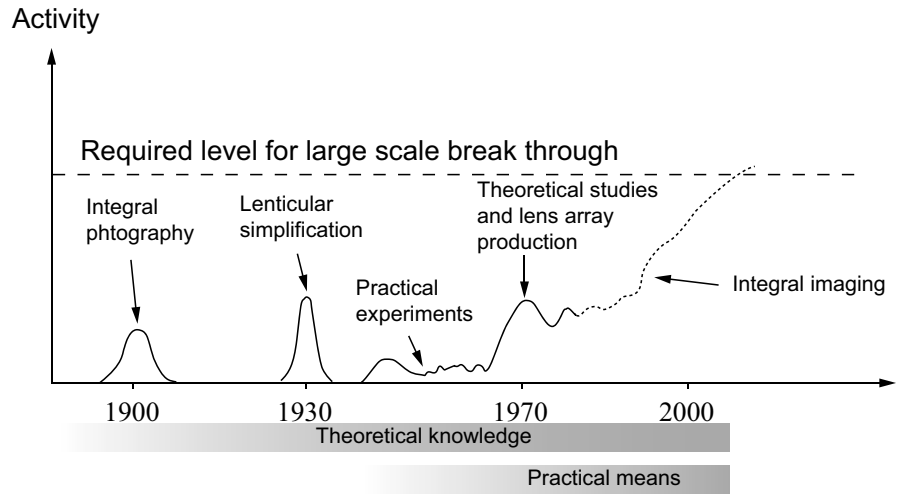


Figure 2.10: Level of activity within the field of IP and II during the last century. The figure is an extrapolation of a figure published in [16].

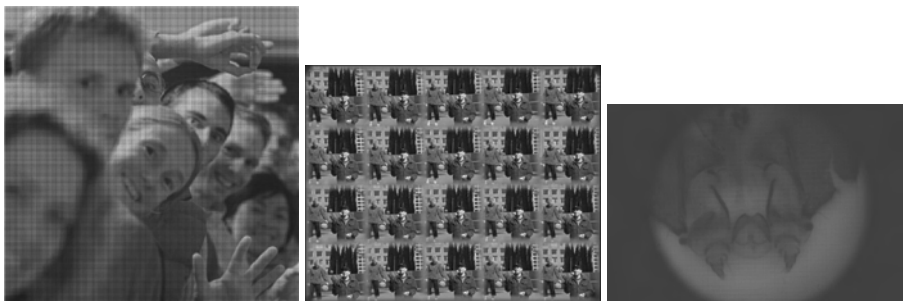


Figure 2.11: Examples of images taken by II-based camera systems [35, 38, 36]. The two leftmost images stem from hand-held cameras whereas the right (depicting a silkworm mouth) originates from a microscope.

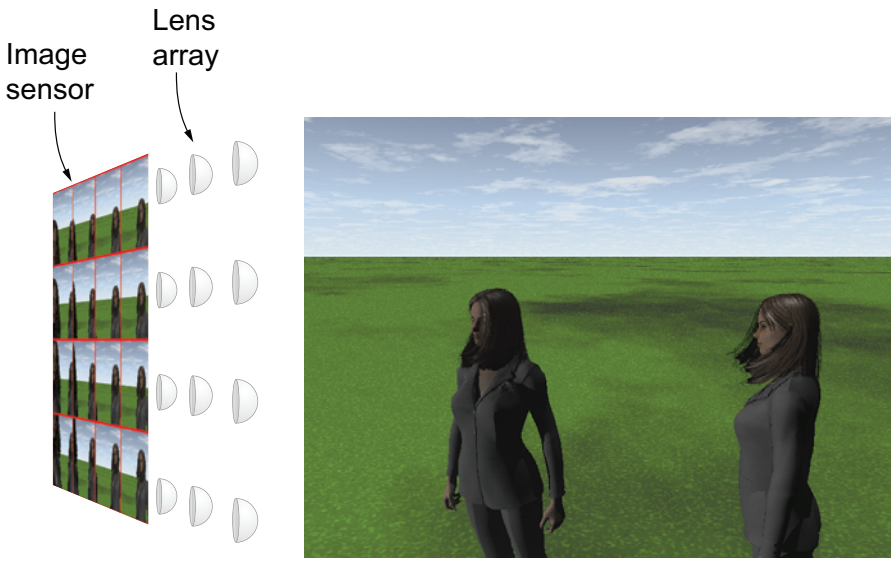


Figure 2.12: An II-camera capturing the scene Twins from 4×4 different perspectives, which implicitly stores scene depth. The projections on the image sensor are framed using red markings to elucidate their boarders.

the viewpoint, extract scene depth et cetera [30]. An example of an II-based camera is shown in Figure 2.12 where for illustration purposes an extremely small number of apertures was used.

2.2.2 The dimension taxonomy

Before proceeding with an in-depth study regarding how II works, and its pros and cons, the key term *dimension* will be explicitly defined. A reader who is familiar with how dimension is used in the context of autostereoscopic 3D techniques – and who has no problem in grasping the slightly exaggerated yet completely correct description of II as *a 3D scene depiction technique that takes samples from a subset of the plenoptic function and reorganizes them into a 2D image* – should feel free to miss both this and the next section.

In 1991, Adelson and Bergen [39] proposed a model that describes the entire visible space that surrounds us. Their model, *the plenoptic function*, is a seven-dimensional function $P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$, which describes the intensity of light with wavelength λ that intersects space in position $[V_x, V_y, V_z]^T$ at time t from a directional angle (θ, ϕ). The plenoptic function "serves as the sole communication link between physical objects and their corresponding retinal images" and is the "intermediary between the [3D] world and the eye" [39]. That is, when we view the world around us it can be argued that we do not perceive the objects as such. Instead we sam-

Figure 2.13: Three samples of the plenoptic function P. Dark arrows correspond to directions from which visible light is present whereas bright arrows correspond to directions from which no light is arriving.

ple the plenoptic function that the objects collectively produce. Hence, this model is a powerful tool in describing different image capture systems and the images they produce. Figure 2.13 illustrates a set of samples taken from the plenoptic function P in a scene containing a number of objects.

The process of sampling the plenoptic function is attributed to all image capturing devices, including our eyes. What differs is the way in which the sampling is performed. A conventional digital camera captures an image at a specific time t and place $[V_x, V_y, V_z]^T$ by sampling the plenoptic function, producing the image $\mathbf{I}_{2D}(u, v) = P(\theta, \phi)$. The horizontal and vertical pixel positions u and v are implicitly related to the directional angles by a polar to Cartesian transformation. Similar to the operation performed by the color receptive cones in the retina, the visible range of wavelengths λ is integrated in each sensor pixel and divided into the red, green and blue components respectively. Furthermore, camera optics restrict the field of view, or range of directional angles, and the sensor's pixel resolution defines the sampling rate within this range. A video camera on the other hand results in an output that contains one more dimension, time. Hence, a 3D slice is taken, which results in the video sequence $\mathbf{I}_{2D}(u, v, t) = P(\theta(t), \phi(t), V_x(t), V_y(t), V_z(t))$. Note that even though this is a signal with three dimensions, it merely contains a set of 2D images sampled at different points in time. It does not contain sufficient information in its general form to reconstruct a 3D view of the world at any given time t.

For a depiction that in all essentials resembles a view into the 3D scene through a plain glass window, an image capturing device that produces an output with a higher dimensionality than three is required. Depth cues necessary for perceiving the image as 3D are otherwise lost. Four of the seven variables in the plenoptic func-

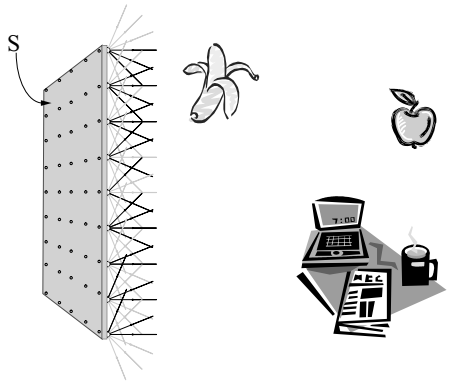


Figure 2.14: The plenoptic function sampled on the rectangular surface S. Coloring of arrows are the same as for Figure 2.13.

tion suffice to depict a static 3D scene. Given that the window may be modeled as a 2D surface S, the four variables would be a pair of position coordinates (k, l) together with a pair of directional angles for each position. Using the Cartesian form the image output then becomes $\mathbf{I}_{3D}(u, v, k, l) = P(\theta, \phi, \mathcal{V} \in S)$, where $\mathcal{V}$ is a set of sampling points located on the surface S. Figure 2.14 illustrates a planar rectangular image and its "directional" pixels, each capturing the incoming light from all directions in the positive hemisphere, i.e. into the scene. For presentation purposes only the set of directional pixels in the nearest column is shown.

This subset of the plenoptic function was concurrently defined – albeit slightly differently – in 1996 as the Light field and the Lumigraph [40, 41]. Hereafter the term light field will be used to describe this subset. Thus, the light field is sufficient to depict a 3D scene as through a plain glass window. The next section will address how II can be described in terms of sampling the light field.

2.2.3 II - a way to sample the light field

The surface S in Figure 2.14 bears a strong resemblance to a camera with a multitude of regularly spaced apertures in front of the film or image sensor. The main difference is that the light field definition states that the directional information in each point (k, l) is independent from all other points. In the analogy with a multiple-aperture-single-image-sensor-camera, the integrity of the directional information from all points (k, l) can *not* be guaranteed due to the potential overlap when each so called Elementary Image (EI) is stored onto the same image sensor. Figure 2.15 shows an II-camera, which uses a lens array to sample the kl-plane together with an image sensor that collectively stores the directional information from all (k, l) samples. Only the nearest column of lenses is shown with object-indicating arrows, to make the illustration more clear. The set of uv-planes – corresponding to each (k, l) sample – integrates into the same 2D plane constructing an integral image.

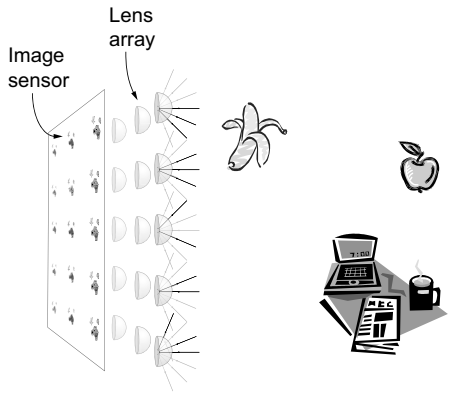


Figure 2.15: The plenoptic function sampled by an II-direct pickup camera. The image sensor stores the directional information sampled by the set of lenses in the lens array.

Hereafter, the pixels stored in the image array is referred to as an Integral Imaging picture (II-picture).

The micro-lenses or lenslets in the lens array can take different forms: spherical, rectangular, cylindrical etc. Their position pattern can be both rectangular and hexagonal. Instead of lenslets the optical equivalent pinhole and point light may also be used. For example, vertically aligned cylindrical lenses or lenticulars result in an HPO II-technique that sacrifices vertical parallax for a reduced requirement in pixel resolution [31, 42]. Consequently, the lenslet size and position strongly affect how the light field is sampled and thus the II-properties.

Not all scenes are as well behaved as that in Figure 2.15 where there is no overlap between neighboring lenslets' projections. Hence, it is the II-camera's responsibility to sample the light field in such a way that perfect reconstruction is possible at the display side. There are many trade-offs to be made in order to accomplish this and the following section will present the II-properties that must be considered when designing an II-capture and display system.

In addition to the fully optical II-capture and display systems, systems exist where either the camera or the display is replaced by computer simulations – computer generated II and computational II respectively [32]. In all essentials these are also covered by the following discussions. However, computational II particularly allows for improved reconstruction properties due to the ideal – and even non-physical – characteristics of the virtual display optics. Chapter 3 further discusses computer generated II, which is used in this work both as a substitute for II-camera prototyping and as a tool to perform comparative studies. Computational II is used in Chapter 4 where it constitutes a part of a proposed quality metric.

2.2.4 II-properties

The high degree of geometrical symmetry between the II-camera and II-display enables one of them to be studied and to transfer the conclusions drawn to the other. This will be exploited in this section, where the perspective of the study will switch between the camera and the display when required.

There are a number of II-properties that are vital for the final experience of viewing the depicted 3D scene. The most important are:

1. Pseudoscopic image.
2. Real and virtual images.
3. Viewing angle.
4. Image resolution.
5. Depth range.

The following overview will show the effect of these properties and how they interact. It is based on a geometrical optic analysis due to its beneficially low complexity while still providing sufficient accuracy from which system level conclusions can be drawn. For exact studies of physical phenomenon such as aberrations and diffraction, wave optics are required. This topic is outside the scope of the work and the reader is referred to [32] for a comprehensive survey and [43] for in-depth studies.

The II-properties are an effect of the geometry of the II-system. Therefore Figure 2.16 illustrates a geometry model that subsequent discussions will relate to. The parameters in Figure 2.16 are δ^L (lens array pitch), δ^P (pixel array pitch), Λ (gap between pixel and lens array), α (lenslet opening angle), z (distance to image plane), Δz (depth range) and δ^{P_I} (image pixel pitch). These will be further discussed in the subsequent sections. To simplify the illustration, the II-system is seen from the side and is thus exposing only three lenslets from the first column.

2.2.4.1 Pseudoscopic image

In its original form the II-display relays the captured II-picture using a lens array similar to that used in the II-camera. Figure 2.17 (a) shows the sampling and Figure 2.17 (b) illustrates the reconstruction respectively. A disadvantage associated with this approach is that a pseudoscopic image of the depicted scene is produced, i.e. all objects are depth-inverted; convex becomes concave and vice versa.

Different methods, optical and electrical, have been proposed to reconstruct an orthoscopic or depth-correct image. One optical method is to add another optical subsystem that inverts the object depths *prior* to the capturing lens array [31, 44, 45]. Figure 2.17 (c) shows one implementation of such a subsystem. A less bulky and more simple approach is to adopt GRadient INdex of refraction (GRIN) lenses in the lens array, i.e. optic fibers with an index of refraction that changes as a function

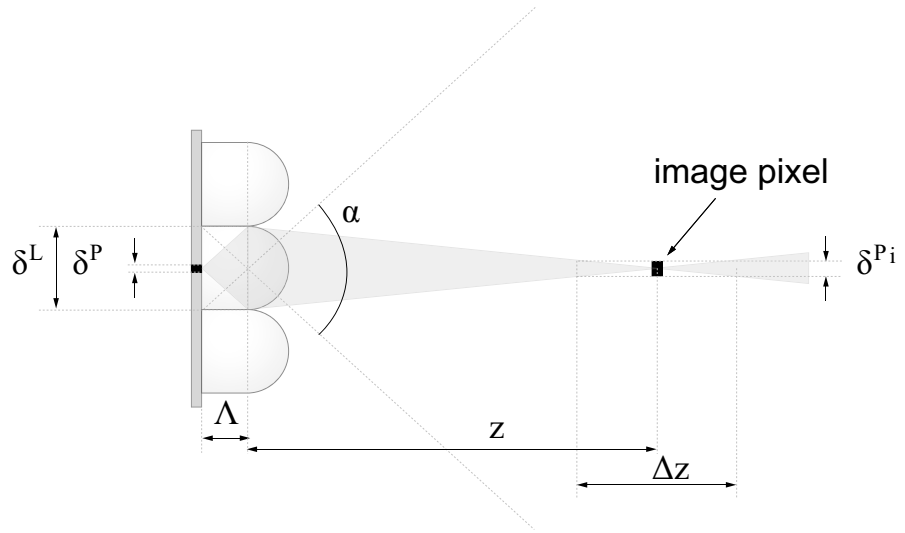


Figure 2.16: Geometry model of a II-display, from which the various II-properties are derived and defined.

of radius Okano et al. [46]. Compared to the original lenslets, the GRIN counterparts can produce erect images, which when projected in reconstruction produce orthoscopic images. This 180°rotation of each lenslet projection, with respect to the lenslet's optical axis, is shown in Figure 2.17 (d) – showing only the vertical component of the rotation due to presentation simplifications. A drawback associated with many of these optical solutions is the reduced reconstructed image resolution due to inter alia diffraction effects caused by the additional optical components [47]. Performing an 180°rotation of each lenslet projection is a straightforward operation from a digital signal processing standpoint and has also been utilized as a conversion method [48]. However, there is still with the drawback of transforming real images into virtual and thus restricting all depicted objects to be confined *inside* the display. Martínez-Corral et al. [47] combined the method proposed by Ives [31] (which lacks this disadvantage as Figure 2.17 (c) points out) with digital signal processing such that a part of the II-camera optics is replaced by simulated ideal components without any distortion factors.

2.2.4.2 Real and virtual images

Virtual images prove to be not the only unwanted by-product of some methods which are converting from pseudoscopic images to orthoscopic. Even though a real image that floats in mid-air might be considered more striking it is also more affected by window violation [49]. This involves a real object being seen without distortion in a smaller portion of the depicted 3D space than a similarly sized virtual object. Figure 2.18 illustrates this property. Thus, using real *and* virtual images is advantageous

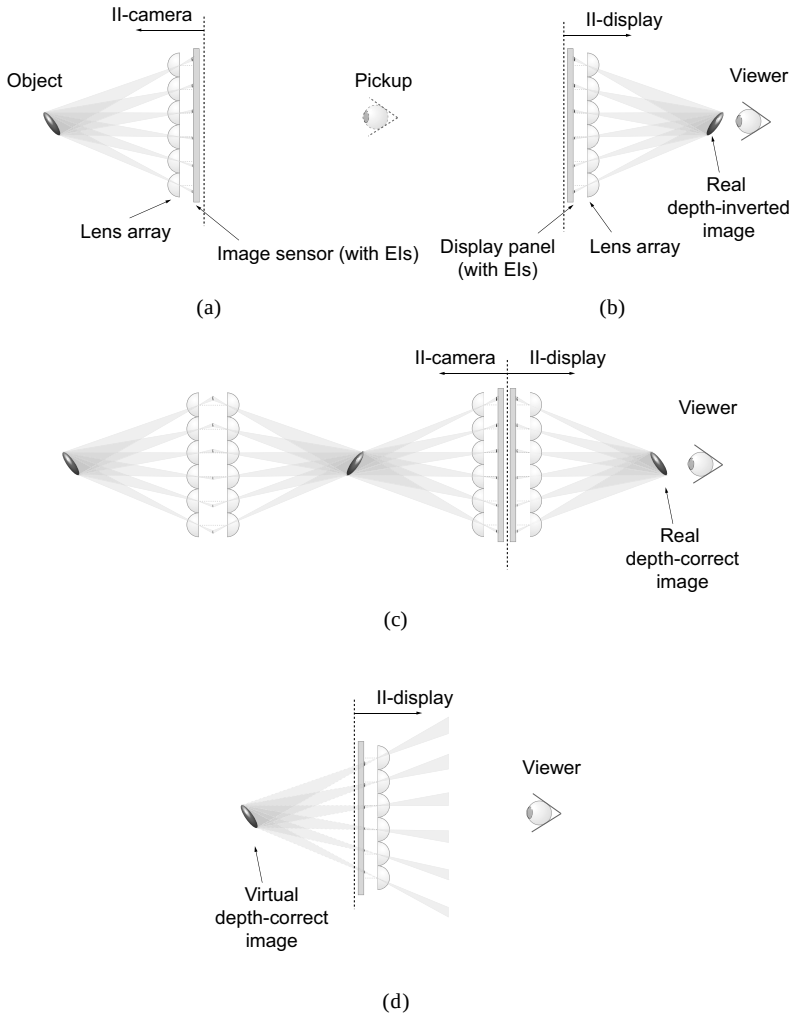


Figure 2.17: Pseudoscopic to orthoscopic transformation methods. (a) Direct capture with single lens array (b) Direct reconstruction rendering depth-inverted pseudoscopic objects (c) Two-tier capture lens array with reconstructed depth-correct orthoscopic images (d) Virtual depth correct image by rotating the EIs 180 °around their individual centers.

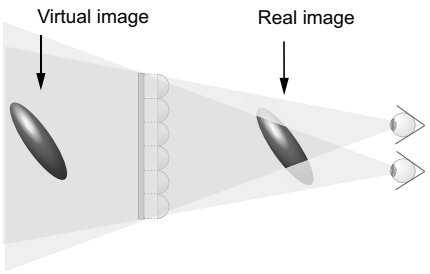


Figure 2.18: Window violation affecting the real object causing parts of the object to be perceived without the binocular depth cue.

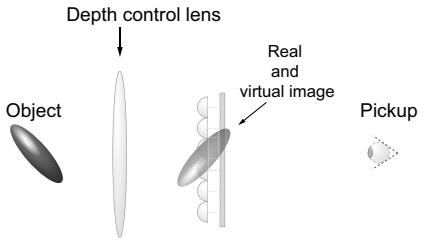


Figure 2.19: The II-camera uses a depth-control lens to allow for simultaneous real and virtual object reconstruction.

when depicting a 3D scene.

In the original direct capture version of II, all objects within the 3D scene are real as a result of the physical boundary of the II-camera's lens array. Hence, all reconstructed objects are either real or virtual depending on the pseudoscopic conversion method. A depth controlling lens can be used to remedy this constraint, which projects the scene such that it straddles the lens array. Some imaged objects then remain real whereas others become virtual. Figure 2.19 shows a depth control lens implemented using a simple convex lens. GRIN-lens arrays have also been used as a depth control lens to reduce the different degrees of depth distortion that a convex lens introduces [50]. Varying the position of the depth control lens, relative to the lens array, allows for different parts of the depicted scene to be reconstructed at the display plane.

2.2.4.3 Viewing angle

A weakness of II is the relatively small viewing angle in which orthoscopic 3D is perceived. A typical value of the viewing angle α is approximately 20°[2, 37, 51]. In

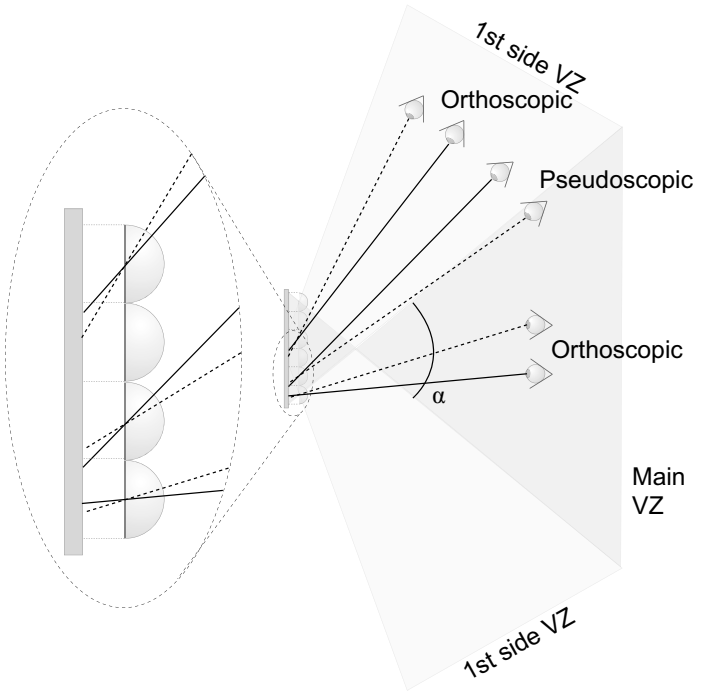


Figure 2.20: VZs that are formed by a II-display. A user positioned with both eyes within a given VZ (the main VZ, the 1st side VZ, 2nd side VZ, etc.) perceives the 3D images as orthoscopic. However, a depth-inverted pseudoscopic 3D images is perceived if the eyes are positioned in two adjacent VZs.

Figure 2.20 the viewing space of a II-display is shown – the leftmost part of the figure is a zoomed in section of the lens array. A viewer located in the main Viewing Zone (VZ) is able to see both the real and the virtual objects with the correct depth, i.e. orthoscopic. The analogy between II-display and -camera geometry, translates the main VZ of the display into the subset of the depicted 3D scene in the camera that can be captured by all lenslets. The perceived image becomes pseudoscopic when the viewer is moving out of the main VZ and sees pixels confined to the lenslet projections as well as pixels belonging to neighboring lenslets. Moving even further translates the viewer into the first side VZ where again orthoscopic images can be seen. This repetitive pattern continues throughout the 180° around the display plane's normal. The viewing angle of a II-system is restricted by the lenslet viewing

angle α in Figure 2.16, defined as:

$$\alpha = 2arctan\left(\frac{\delta^L}{2 \cdot \Lambda}\right), \tag{2.1}$$

where δ^L is the pitch of the lenslets (horizontal and vertical pitch set equal) and Λ is the gap between the lens array and the display panel. Thus, increasing δ^L or decreasing Λ, both have a positive effect on increasing the viewing angle α. However, increasing α has a negative effect on both the image resolution and the depth range as the following sections will show.

A method of extending the viewing angle is to utilize temporal multiplexing at a rate surpassing the critical flicker frequency of the HVS. Moving the lens array in synchrony with a high speed update of the pixel array content can increase the viewing angle [52]. A broader viewing angle can also be accomplished by introducing a dynamic barrier array between the pixel sensor or display panel, and the lens array [53]. In the display, these barriers act as opaque tubes that restrict what pixel subsets are seen through the lenslets. At one time instant the pixel subset directly under the lenslets is directed for view using the tubes. In the next instant, the tubes have tilted to show other pixel subsets. Alternating fast between the tube tilt angles and synchronously updating the pixel content removes the side views in Figure 2.20. The side views are instead filled with novel information extending the viewing angle. The complexity of using techniques based on mechanical motion increases when the operation frequency is increased. A somewhat modest frequency of 25-30 Hz is sufficient to provide a still image, but for 3D video a multiple of the used frame rate is necessary [54]. Lee et al. [55] propose a system, which does not posses this inertia problem. Limiting the motion to be discrete, allows for the use of a liquid-crystal shutter as an on-off-mask located behind the lens array. Covering up neighboring lenslets in turn allows for a larger pixel subset to be associated with each lenslet and thus increasing the viewing angle. Despite being theoretically attractive, these methods have a practical obstacle to overcome: the update frequency of the pixel arrays must comply with the operation frequency used. This is a requirement that may cause problems for certain display technologies which are limited in their operation frequency as a result of physical properties [56].

More about the viewing angle of II-systems can be found in [57] and additional multiplexing approaches are collected in [32].

2.2.4.4 Image resolution

A reconstructed image pixel size δ^{P_I} (as shown in Figure 2.16) can be defined at depth z using the lenslet magnification factor $A = \frac{z}{\Lambda}$ as

$$\delta^{P_I} = \min\left(\frac{z \cdot \delta^P}{\Lambda}, \delta^L\right), \tag{2.2}$$

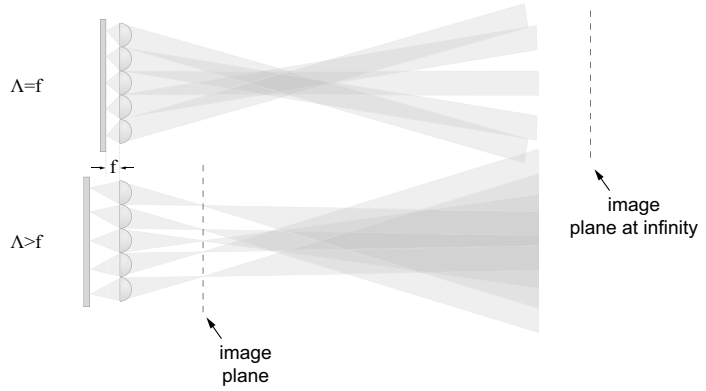


Figure 2.21: Two II-systems, with different gap-distance between pixel- and lens array.

where δ^P is the pixel size of the display panel and δ^L is the lenslet pitch [58]. The location of the image plane is calculated using the simple lens law

$$z = \frac{f \cdot \Lambda}{\Lambda - f}, \tag{2.3}$$

where f is the focal length of each lenslet. Two examples of image plane location due to different gaps Λ, are shown in Figure 2.21. Substituting Equation (2.3) in Equation (2.2) results in the size of an imaged pixel at located depth z. Thus, the image plane where the objects are reconstructed has the spatial resolution $R^I = \frac{1}{\delta^{P_I}}$. Increasing the gap, or decreasing the lenslet pitch, both increase R^I. However, increasing the gap reduces the viewing angle and also translates the image plane closer to the lenslet array. The lowest image resolution ($R^I = \frac{1}{\delta^L}$), is achieved when the gap is equal to the lenslet focal length $\Lambda = f$ [58]. This setup places the image plane at $z = \infty$, so called depth-priority II [59]. Resolution-priority II emphasizes image resolution before object depth by placing the image plane at a finite depth. This is achieved by setting $\Lambda > l$. Depth- and resolution-priority II is shown in the top and bottom part of Figure 2.21 respectively.

Temporal multiplexing can also be used to provide an increased image resolution. The moving lenslet array technique described in the previous section allows for an increased number of pixel subsets per lenslet, thus increasing the display panel resolution in a virtual manner [52]. The image resolution is then increased according to Equation (2.2). Other spatial- and spatiotemporal-multiplexing methods are described by Stern and Javidi [32].

2.2.4.5 Depth range

A consequence of an image plane is that objects *outside* the plane are out of focus or represented by reduced image resolution. The range in which objects are accurately

reconstructed is denoted by the II-system's marginal image depth Δz and is shown in Figure 2.16. The marginal image depth is in [58] defined as

$$\Delta z = 2z\frac{\delta^{P_I}}{\delta^L} \tag{2.4}$$

for $\Lambda \neq f$, whereas for $\Lambda = f$

$$\Delta z = 2\Lambda\frac{\delta^L}{\delta^P}. \tag{2.5}$$

Objects are considered out of focus when the focus error becomes larger than the image pixel. This occurs for objects that are located at a larger distance from z than Δz. For resolution-priority II, Equation (2.4) indicates that the depth range is reduced if the image resolution is increased. Moving the image plane close to the lens array would remedy this. However, it is not possible to achieve an infinite marginal image depth. The resolution of image planes close to the lens array would become constrained to the lens pitch according to Equation (2.5) [58]. Increasing gap Λ or lenslet pitch δ^L and decreasing pixel size δ^P will increase the depth range. Unfortunately, the viewing angle α and image resolution R are instead reduced by increasing Λ or δ^L.

Increasing the depth range is advantageously achieved by optical methods. Instead of using a homogenous set of lenses in the lens array, a repetitive pattern of lenses with different focal lengths within the lens array allows for a set of image planes located at different depths [60]. The resulting reduction in image resolution – due to the introduction of lenslets with larger pitch – is remedied by temporal multiplexing as described previously. Another approach, which does not require temporal multiplex at all, is to stack additional display panels with different gaps together [61]. The resulting image planes are then also stacked together and combined into a full 3D images with a broader range of depths in focus.

2.2.5 Constraints in property trade-off

As the previous sections have shown there is a strong inter-relation between the different II-properties, viewing angle, image resolution and depth range. Thus, to design an optimum II-system, trade-offs are required. To assist in this trade-off process, Min et al. [58] have proposed a characteristic equation that combines Equation (2.1), (2.2) and (2.4) into

$$R^{I^2}\Delta z \cdot tan\left(\frac{\alpha}{2}\right) = R, \tag{2.6}$$

where R is the resolution of the image sensor / display panel. The equation clearly states that there is only one single method by which all the properties can be improved, without sacrificing any other: increasing the resolution of the pixel sensor and display panel. All other approaches will merely emphasize one property at the expense of the others.

The characteristic equation stems from a geometrical optics analysis, which disregards diffraction. However, there is a relationship proposed by Stern and Javidi

[32], which also considers diffraction. Writing their "product of depth of focus and resolution squared" using this dissertation's notation gives

$$\Delta z R^{I^2} = \frac{1}{\lambda}, \tag{2.7}$$

where λ is the wavelength of incident or transmitted light. This relationship holds for the diffraction limited case as well, i.e. for II-systems where $\Lambda = f$. For II-systems that have an image pixel resolution that is a function of lenslet magnification ($\Lambda \neq f$), the viewing angle and pixel size also become factors in the relationship according to

$$\Delta z R^{I^2} \alpha^2 = \frac{\lambda}{(\delta^P)^2}. \tag{2.8}$$

Despite geometrical optics there are other ways to analyze II-systems, e.g. using wave optics [43]. Moreover, optical transfer function analysis has been adopted by approximating the II-system as a linear time-invariant system [62]. The interested reader is referred to these references for in-depth and extended studies.

2.2.6 The Component Images of the II-picture

The II-camera stores the light field samples on a 2D images sensor, which allows for viewing the fixed-time 3D information directly as a 2D image. That is, the intensity and directional information from the slightly different perspectives of the scene are spatially multiplexed into a single image similar to those shown in Figure 2.11 on page 25. Subsequent discussions will assume II-cameras using rectangular lenslets arranged in a rectangular pattern. This weak constraint simplifies the presentation and can easily be met by II-cameras with other lenslet shapes and positioning patterns by applying cropping or zero-padding of the captured 3D image.

We start with defining a color II-picture $\mathbf{II}$ as

$$\begin{aligned} \mathbf{II} &= [II(m,n)]_{\substack{m=0,1,\ldots,M-1 \\ n=0,1,\ldots,N-1}} \\ &= \begin{bmatrix} II(0,0) & \cdots & II(M-1,0) \\ \vdots & \ddots & \\ II(0,N-1) & & II(M-1,N-1) \end{bmatrix}, \end{aligned} \tag{2.9}$$

where $m = 0,1,...,M-1$ and $n = 0,1,...,N-1$ are the horizontal and vertical positions of an II-picture pixel respectively. The RGB-color pixel at n_0-th column and m_0-th row is then

$$II(m,n) = \begin{bmatrix} II_R(m,n) & II_G(m,n) & II_B(m,n) \end{bmatrix}^T. \tag{2.10}$$

The II-picture can be transformed without loss between different representational forms. These forms can be considered to be composed from different types of Component Image (CI). I then define a CI as

$$\mathbf{CI} = [CI_{\xi,\psi}(s,t)]_{\substack{\xi=0,1,\ldots,\Xi-1 \\ \psi=0,1,\ldots,\Psi-1}}, \tag{2.11}$$

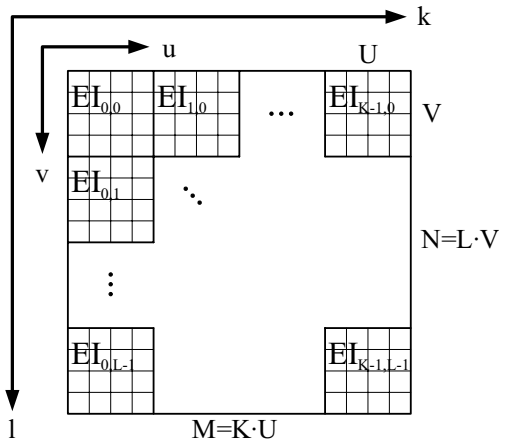


Figure 2.22: Spatial multiplex structure of the II-picture showing the number of EIs ($K \times L$) and their resolution ($U \times V$).

where each of the $\Xi \cdot \Psi$ CIs has a spatial resolution of $S \times T$. Note that the CI set is constructed from the II-picture pixels exhaustively, i.e. $M = \Xi \cdot S$ and $N = \Psi \cdot T$.

2.2.6.1 Elementary Image (EI)

The subset of pixels beneath each lenslet, in which a low resolution projection of the depicted 3D scene is stored, is called EI. Hence, the simplest transformation is to set CI equal to EI, which in practice means that the II-picture is not transformed at all. However, the EI becomes

$$\begin{aligned} \mathbf{EI} &= [EI_{k,l}(u,v)]_{\substack{k=0,1,...,K-1 \\ l=0,1,...,L-1}} \\ &= [II(k \cdot U + u, l \cdot V + v)]_{\substack{k=0,1,...,K-1 \\ l=0,1,...,L-1}} , \end{aligned} \tag{2.12}$$

where $u = 0, 1, ..., U-1$ and $v = 0, 1, ..., V-1$ are the horizontal and vertical pixel positions within each EI. Thus, each of the $K \cdot L$ EIs has a resolution of $U \times V$ pixels. Figure 2.22 shows how the II-picture is subdivided into EIs. Note that each individual EI has its own u,v-coordinate system. From a light field perspective the EI can also be defined as $EI_{k,l}(u,v) = \mathbf{I}_{3D}(u,v,k,l)$. The number of pixels in each EI corresponds to the number of views distributed, or the angular resolution in which the 3D scene can be represented. In Figure 2.23, pixels that are intersected by lines crossing the center of the same lenslet belong to the same EI.

2.2.6.2 Sub Image (SI)

The complete Sub Image (SI) is the next CI-type and is formed from II-picture pixels sharing the same relative horizontal and vertical offset to the EI centers. The SI is an

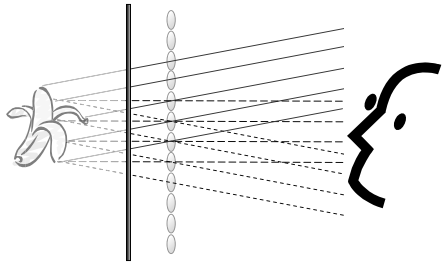


Figure 2.23: II-display with display panel and lens array. Rays passing a given lens corresponds to an EI. Parallel rays from different lenses correspond to an SI.

extension of a concept used in the field of II-based depth estimation [63, 64]. I define a complete sub-image SI as

$$\begin{aligned}\mathbf{SI} &= [SI_{u,v}(k,l)]_{\substack{u=0,1,...,U-1\\v=0,1,...,V-1}} \\ &= [II(k\cdot U+u, l\cdot V+v)]_{\substack{u=0,1,...,U-1\\v=0,1,...,V-1}} ,\end{aligned} \tag{2.13}$$

where $k = 0, 1, ..., K-1$ and $l = 0, 1, ..., L-1$ are the horizontal and vertical pixel positions within each SI. Thus, each of the $U \cdot V$ SIs has a resolution of $K \times L$. Again, using light field terminology $\mathbf{SI}_{u,v}(k,l) = \mathbf{I}_{3D}(u,v,k,l)$. In Figure 2.23, pixels that are intersected by parallel lines of the same line-style belong to the same SI. The fact that the SI is formed from parallel light rays results in its characteristic *orthographic* projection property, i.e. contrary to the *perspective* projection of the EI a change in object depth does not result in a size change of the object's projection onto the SI.

2.2.6.3 Ray-space Image (RI)

The final CI-type is the Ray-space Image (RI), also known as Epipolar Plane Image (EPI) [65, 66]. There are different definitions of RI for full parallax II-pictures, as RI was originally defined for one-dimensional lens positioning and HPO II-pictures. In this work RI is defined as

$$\begin{aligned}\mathbf{RI} &= [RI_{v,L}(u,k)]_{\substack{v=0,1,...,V-1\\l=0,1,...,L-1}} \\ &= [II(k\cdot U+u, l\cdot V+v)]_{\substack{v=0,1,...,V-1\\l=0,1,...,L-1}} .\end{aligned} \tag{2.14}$$

This means that rows of pixels are selected from the II-picture (top to bottom) and folded into images with a resolution of $U \times K$ pixels. A characteristic property of RI is that the slant angle of a line segment is proportional to the depth of the object giving rise to the segment.

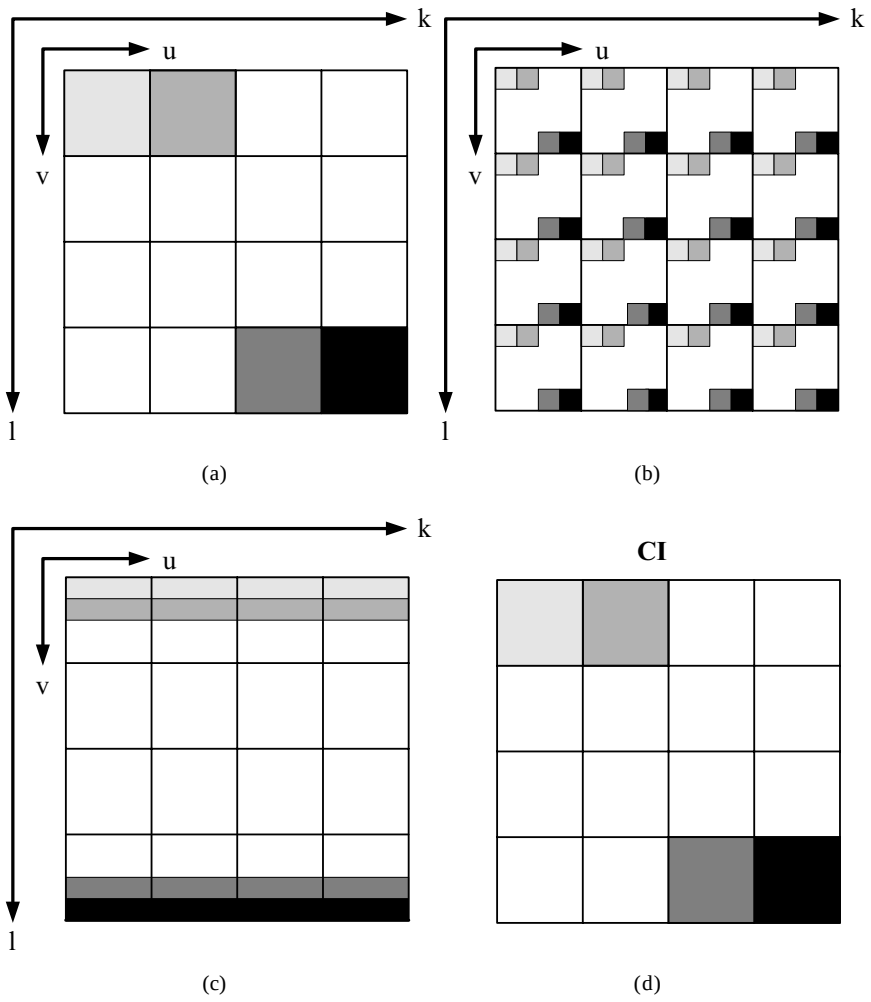


Figure 2.24: The set of CIs in (d) and the II-picture pixels that constructs (a) $CI = EI$, (b) $CI = SI$ and (c) $CI = RI$.

2.2.6.4 Comparison of CIs

The definitions in Equations (2.12) – (2.14) construct different sets with different characteristics. Figure 2.24 shows graphically how the II-picture pixels are selected for the different types of CI. A specific CI (identified using a certain gray scale in Figure 2.24 (d)) is composed from II-picture pixels in (a) – (c) with the same corresponding gray scale. Note that even though the figure implies a squared shaped II-picture, this is not a requirement. An example of content for the three CIs (taken from the II-picture Twins) is shown in Figure 2.25. In this particular case a square shaped II-picture was used ($U = V = K = L = 64$) to clearly illustrate the different CIs prop-

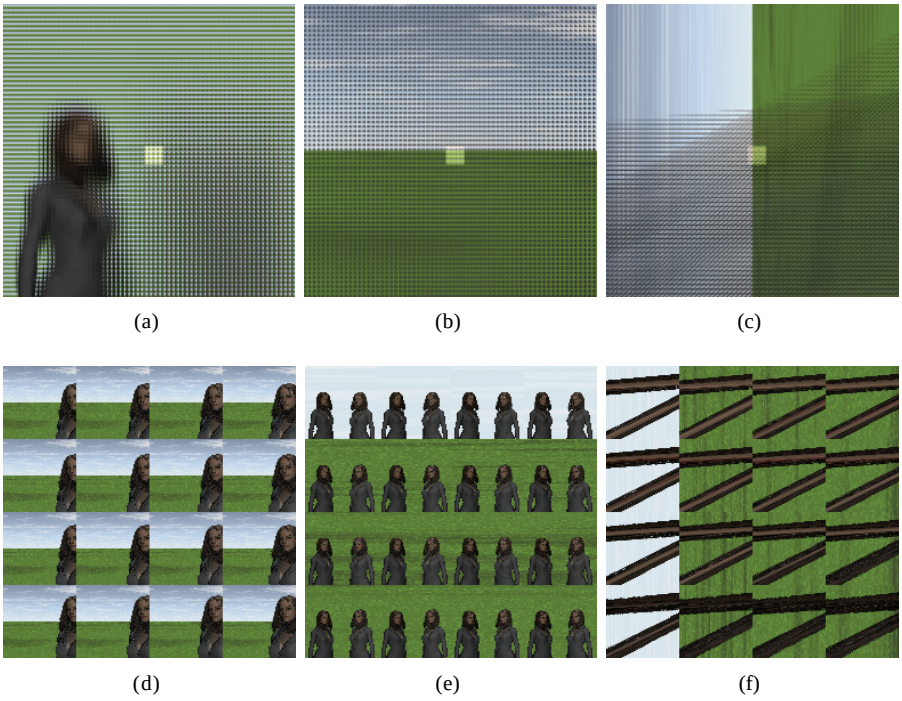


Figure 2.25: The II-picture Twins transformed into different CIs. (a) $CI = EI$, (b) $CI = SI$ and (c) $CI = RI$. (d) – (f) shows the middlemost highlighted 4×4.

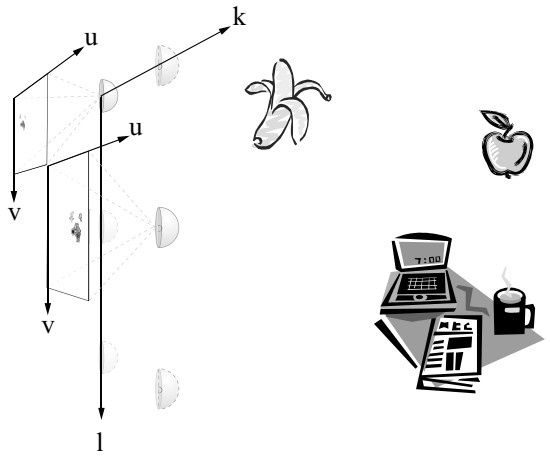


Figure 2.26: The plenoptic function sampled by a multi-view camera setup. Only two camera projections are shown to make the presentation less cluttered.

erties without being constrained by unequal resolution ($S \times T$). The orthographic property of the SI are manifested by the equal size of the two women, despite them being at different distances from the II-camera. The RI's correspondence between the slant angle and the object depth is also a prominent property. The top color patch, corresponding to nearest woman, has a smaller slant angle as she is closer to the camera than the other woman. Hence, a steeper slant corresponds to a more distant object.

2.2.7 Multi-view - another way to sample the light field

There are other ways than II to sample the light field for 3D depiction. Multi-view is a common technique that uses a set of ordinary 2D cameras in the sampling process. This results in a lower resolution of the kl-plane whereas the uv-plane is sampled using a higher resolution. Figure 2.26 illustrates a sparsely populated kl-plane in which each camera is located relatively distantly from the others. Contrary to the II-direct pickup camera the uv-plane is more densely sampled whereas the kl-plane are sparsely populated. Thus, each camera has a relatively high spatial resolution, but there are a relatively small number of cameras. This relationship is often reversed for II even though there are II-systems where the set of lenses ($K \cdot L$) is less than the EI resolution ($U \times V$) [38]. The maximum base line, i.e. the distance between the most distant (k, l) samples, is also larger for multi-view as compared to II. Cameras can naturally not be placed as tight as lenslets. A larger maximum base line allows for a larger parallax.

The common notion about II and multi-view is that the former relies on a very high number of views – in both the horizontal and vertical directions. However,

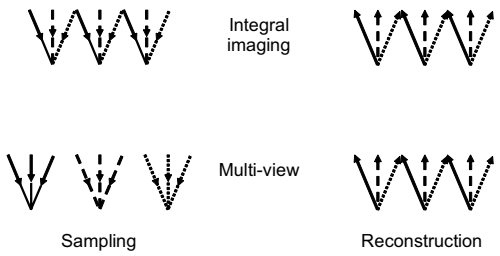


Figure 2.27: Difference in sampling and reconstruction for II and multi-view.

several different kinds of II exists (prioritizing depth over spatial resolution or vice versa, sacrificing vertical parallax for bandwidth etc). From a signal format perspective there is a strong resemblance between the HPO II (with 8 pixels per lenticular lens width) studied by Forman et al. [67] and the 9-view multi-view format used in the 3D display from Philips [26]. Thus, the main difference between II and multi-view is not the number of views but rather how the light field is sampled and reconstructed. For II, the reconstruction strives to be an exact inverse of the sampling, which is not the case for multi-view as Figure 2.27 shows. Despite the similar reconstruction optics, the information distributed orthographically differs between the two techniques. Where the II-display reconstructs an orthographic projection into a specific direction relative to the display plane normal, the multi-view display reconstructs a downsampled perspective projected image from one of the cameras. This fundamental difference is independent of the resolution used to sample the kl- and uv-plane.

2.3 Related works

This section summarizes the related works conducted within the three areas of the dissertations: synthesis, evaluation and coding. This overview acts as the basis for the subsequent three chapters.

2.3.1 Synthesis

There have been relatively few efforts in addressing II-picture synthesis in the literature. However, the work that has been presented presented has shared a common goal despite different objectives: produce II-pictures with significantly good correspondence to real-life II-pictures. In addition, the majority of contributions in the literature also share the common principle of concentrating on a single specific II-techniques.

A number of contributions in the field of basic II-research use simple synthesis methods, which may be sufficient for evaluating certain properties [68, 47]. Al-

though being perfectly adequate for their task, none of these synthesis methods lend themselves to a more general use. This would involve transforming these synthesis methods to outside the scope of the work in which they have been defined and this would prove to be rather difficult.

A somewhat more general approach was presented as a sub-system of a surgical navigation system, i.e. the synthesis was restricted to producing II-pictures from volumetric data [69, 70]. Volumetric data – captured from MRI or Computer Tomography (CT) scans – was transformed into II-pictures using geometrical optics and ray-tracing. The synthesized II-pictures were later displayed by an II-display and superimposed onto a patient as a visual aid during operation.

The contribution of Milnthorpe et al. [71] instead focused on rendering wire-frame models into II-pictures. This approach addressed a broader range of applications as it could be considered an update of the fixed rendering pipeline, which was the dominating graphics card architecture at that time. The proposed software model updated important pipeline sections such as projection, polygon culling, rasterization, and interpolative shading. The results were also used to visualize so-called "cyber-sculptures", which were 3D sculptures displayed as part of an art installation using a projector-based II-display [72].

An even more general way to synthesize multi-view images was proposed by Halle [12] – thereby also lending itself to II-picture synthesis. The approach elegantly focused on enhancing the rendering speed by significantly redesigning the rendering pipeline. The proposed method rendered directly into the space of EPI instead of rendering all scene objects for each and every view (EI or SI). The EPIs bear a strong resemblance to the RIs defined in Section 2.2.6. Hence, a large (if not complete) set of views of the scene was produced concurrently when rendering to this ray-space. Although showing significant rendering speedups for multi-view images with many views, the high complexity associated with redesigning the display chain prevented this work from being adopted. In addition, the lack of availability of 3D displays meant that there were few incentives to work in this area. Another major obstacle to its adoption was the fact that the rendering speed of the approach on single view rendering (i.e. when used on 2D displays) was of a magnitude slower than the predominant 2D rendering approaches at that time. However, the ongoing efforts in making current graphic cards programmable, coupled with the advent of autostereoscopic displays with more and more views, might very well alter the chances of success for this achievement.

A few contributions have focused on generic and flexible synthesis, in contrast to the above mentioned works where simplicity or speed have been the target. Athineos et al. [73] based their synthesis method on an open-source ray-tracing software package where the II-camera was considered being a part of the scene and thereby modeled as any other scene object.

2.3.2 Evaluation

A common approach to evaluating distortion in II-pictures is to apply a metric that aggregates the quality of the whole II-picture into a single scalar distortion-value. One of the most common objective quality metric that is used in 2D images and 2D video coding is the Peak Signal to Noise Ratio (PSNR), mainly due to its computational tractability and acceptable correlation with subjective test results on image- and video quality. In addition, its widespread use simplifies comparative studies. The PSNR in dB for full color RGB-images is defined as

$$PSNR\left(X,\hat{X}\right) = 20 \cdot log_{10}\left(\frac{255}{\sqrt{MSE}}\right), \tag{2.15}$$

where

$$MSE = \frac{1}{3 \cdot M \cdot N} \cdot \left\|X - \widehat{X}\right\|_F^2 \tag{2.16}$$

where $\left\|X - \widehat{X}\right\|_F$ denotes the Frobenius norm of the difference between matrix X and $\widehat{X}$, which are full color images of size $M \times N$. The metric commonly assumes that $\widehat{X}$ has undergone some kind of distortion-inducing operation making it different from the original X, e.g. lossy encoding and subsequent decoding. Applying (2.15) to the II-picture's complete set of pixels gives a global quality metric that I in this dissertation define as

$$Q_{global} = PSNR(II, \widehat{II}), \tag{2.17}$$

which produce a single scalar quality value for the whole 3D image. The advantage of a scalar value is that a large-scale overview of the degree of distortion present in the II-picture is achieved, which is why it is commonly used in the literature [74–77].

However, the quality aggregation produced by (2.17) is not only an advantage. No detailed insight into the distribution of the distortion can be inferred from a scalar. Hence, a more informative evaluation result was achieved by extending the PSNR to be viewing-angle-dependent [78]. Only II-pictures generated using a lenticular HPO II-technique were considered, which gave a 1D quality metric. Extending their work to include full-parallax II-pictures is a straightforward operation and produces a 2D quality matrix as a result. Hence, I define the PSNR dependent on viewing-angle for a full-parallax II-picture as

$$\begin{aligned}\mathbf{Q}_{angle} &= \left[Q_{angle}(u,v)\right]_{\substack{u=0,1,\ldots,U-1\\v=0,1,\ldots,V-1}} \\ &= \left[PSNR\left(SI_{u,v}, \widehat{SI}_{u,v}\right)\right]_{\substack{u=0,1,\ldots,U-1\\v=0,1,\ldots,V-1}}.\end{aligned} \tag{2.18}$$

The pixel indices u and v correspond to a horizontal and vertical angle, which depends on the geometry of the II-display. Note though that an SI is an orthographic projection and this affects the position of the virtual viewer, which is implicitly assumed by the quality metric $\mathbf{Q}_{angle}$. For the II-display to be perceived as an SI, the virtual viewer must be located at an infinite distance from the display plane. This constraint on viewer location has both an advantage and a disadvantage. Using SIs

implies that no interpolation is required, which is good as this could otherwise introduce additional distortion within the II-picture at the time of measurement. The downside is obviously that implicitly locating the viewer at infinity is a physically flawed assumption.

2.3.3 Coding

There are a few different coding approaches that have been proposed for II-based 3D images in the literature. To a large extent these different coding schemes are linked by being strongly related to the underlying II-picture-structure.

Forman [45] addressed HPO II-pictures, i.e. II-picture-structures which stem from the capture and display using vertical lenticular lens arrays. The resulting EIs cover the whole II-picture vertically ($L = 1$). As a first pre-processing step, each EI was further divided into blocks of 8×8 pixels in the vertical direction. Two different approaches were then proposed to code the EI subsets: a hybrid coding approach known from various MPEG-standards and a 3D DCT. In the first technique, the residual from the difference between two horizontally adjacent EI-subsets was transformed using the Discrete Cosine Transform (DCT). The transform coefficients were then quantized and entropy coded. The second technique replaced the described hybrid-operation of prediction and transformation with an $8 \times 8 \times 8$ pixel 3D DCT, which was applied to EI-subsets that were combined horizontally from eight consecutive EIs.

Sgouros et al. [74] extended the concept of coding EIs using video coding techniques. A full parallax II-picture-structure was used with EIs of hexagonal shape and position pattern, as opposed to an HPO-structure. The minimum rectangular shape encompassing the hexagonal EI was selected for further processing instead of the EIs themselves, to comply with the rectangular pictures that are prevalent in the context of image and video coding. These pictures were then coded using a combination of JPEG-like intra-coding and MPEG-like inter-coding techniques. A disparity estimation and compensation stage were also adopted, which allowed the horizontal and vertical redundancy to be addressed more efficiently. Yeom et al. [75] later proposed a similar approach for an II-picture-structure with a low number of high resolution rectangular EIs. The MPEG2 video coding standard was used for coding; three different ways to order the set of EIs prior to coding were studied and the coding efficiency was evaluated using objective quality measurements on each individual EI.

Similar ideas on coding arose within the computer graphics community, in parallel to the work performed within the field of II. The 3D images parameterizations light field and lumigraph are similar in character to II-pictures but with an increased number of projections and an increased resolution of each projection; each projection originally stemming from a 2D camera [40, 41]. It was identified that vector quantization, Lempel-Ziv entropy coding, linear prediction, and transformation were all useful tools to reduce the multidimensional redundancy inherent in these large data sets. In line with the research performed within the II-community, an MPEG-like

approach to code the light field was initially proposed [79]. This was later followed by the use of the MPEG-1 video coding standard itself, albeit without disparity compensation in order to enable low-complex random access within the light field [80]. An approach to coding the light field using wavelet transformation was proposed by Girod et al. [81], which contrary to the previous methods used the disparity information within the light field to enhance coding efficiency. Closely related to the light field is the ray-space representation of a 3D scene [65], which have been discussed as a format for free-viewpoint TV where the user is free to arbitrarily change the camera's position and aim within the scene [82]. Ray-space images coded using H.264 / MPEG-4 AVC (H.264 / AVC) was proposed by Shao et al. [76].

2.4 Concluding remarks

The desire to depict the world in three-dimensions has resulted in numerous 3D techniques. Out of the numerous attempts at finding the optimal imaging solution, only a few 3D techniques have the potential to provide all the depth cues as required by the HVS to perceive a 3D image. Integral Imaging is such a technique.

The plenoptic function describes the whole visible 3D space. An II-camera can capture a portion of this data set and is able to allow it to be completely reconstructed using a II-display. The recent years research progress within the field of imaging sensors and display panels has allowed the II-technique to be applied to a larger field of applications. The continuing increase in resolution of image sensors and display panels allows for enhancement of the 3D properties of II-cameras and displays.

Computer simulation of the capturing process of II-cameras has in the literature mainly been aimed at low complexity models for verification of laboratory prototype setups. Measuring distortion in II-pictures has traditionally been conducted using 2D images quality metrics extended for II-picture-use by the addition of two dimensions, without giving due consideration to the additional 3D images properties. More effort has been put into investigating how to reduce the inherent redundancy of the II-picture. The next three chapters will present methods and techniques to synthesize, evaluate and code II-pictures that extend and supersede the approaches described in the related works.

2.4.1 Problem definitions

A conclusion can be made after re-examining the problem definitions of Section 1.2 in light of the related works. Neither of the synthesis methods presented in the literature solves problems P1a – P1b. No work has shown how to decouple the constituting parts of the synthesis process: the II-camera, the scene and the II-picture. The previous works with regards to solving problem P2a is almost as easily dismissed. Any novel coding scheme that is presented as coding efficient *must* supersede the coding efficiency of previous methods. Thus, the presented works on coding could be considered fulfilling problem P2a to the degree that each method supersedes the

methods preceding it. However, Chapter 5 will present a coding scheme that improves on the coding efficiency compared to what is presented in previous works with regards to coding. The problem P2b is a consequence of evaluating a presented coding scheme. Still, not many proposed coding schemes have evaluated the objective quality in any other way than from a global perspective. Thus, previous works have discussed how objective quality is affected by a proposed coding scheme but only from a very general point of view. The work presented in Chapter 4 extends on this discussion and gives tools that can provide a more balanced view on coding induced distortion in II-pictures.

Chapter 3

Synthesis

The previous chapter showed the II-camera to be a powerful tool for sampling the plenoptic function. It samples a sufficient portion to allow for the reconstruction of a 3D depiction of the captured scene. Different modifications of the original form strive to enhance different sampling aspects, which lead to trade-offs due to the resolution limitation given by Equation (2.6). These trade-offs are difficult to assess due to the lack of a static frame of reference. Thus, there is at present no easy way of comparing newly evolving II-techniques with each other, since there is no explicit and well defined II-system or II-picture references. In other research fields such as the signal processing of 2D images and 2D video, reference signals have been defined on which novel algorithms and systems are evaluated using well defined quality metrics. These test-songs, -images and -videos are constructed to have specific and complementary characteristics such that they represent a sufficiently large selection of all possible signals. The metrics are then used to measure how a system affects these characteristics under given constraints. In the field of image processing and compression such comparison operations are facilitated by:

1. A well defined quality metric.
2. A set of widely used reference images chosen as input signals.

Transferring these requisites to the field of II would entail the definition of at least one quality metric and a set of reference *scenes*. The definition of quality metrics and evaluation methods are further discussed in Chapter 4. However at present there have been no efforts made to define reference scenes and the most likely reason involves the complexity of the task. Gathering exact knowledge of object size, position, color and texture as well as optics, lighting and environment properties is a very complex if not impossible procedure. Without this knowledge the depicted scene cannot be reproduced exactly in order to test a novel II-technique. However, it is possible to achieve full control over the parameters mentioned by defining *virtual* reference scenes. Repeatability of experiments and accessibility of results is greatly enhanced by having an explicitly defined scene and an equally explicitly defined

virtual camera model.

There is another objective that synthesis (computer generated II) fulfils in addition to aiding comparative research between II- techniques. Once the design of an II-technique is finalized and an II-picture format is settled, the necessity then exist to provide a large set of reference II-pictures for further signal processing research, such as source and channel coding. Real-life captured II-pictures are vital and will dominate once real-life II-cameras become available. However, synthesized II-pictures have a particular quality that is very valuable in aiding reproducibility: the possibility of exactly defining and reproducing scenes.

3.1 Chapter outline

In this chapter, a ray-tracing-based means of synthesizing II-picture will be presented. The approach allows for easy definition of arbitrary complex reference scenes and the synthesis of II-pictures, which adheres to different II-techniques. In Section 3.3, a basic generic II-camera model is proposed that provides a common basis from which different II-techniques can be transformed. Section 3.4 presents how this II-camera model is used to render II-pictures from descriptions of simple reference scenes. Results are presented in Section 3.6, including different II-camera models, scene descriptions and synthesized II-pictures. Finally, conclusions are made in Section 3.7.

3.2 Methodology

The synthesis method presented in this chapter will be a supplement to physically producing II-camera prototypes and capturing real-life II-pictures. The defined II-camera model and the properties of the synthesized II-pictures will be qualitatively evaluated. From an engineering point of view a quantitative evaluation of the II-camera model would have been preferred, e.g. by comparing synthesized II-pictures with real-life produced II-pictures. However, there is one major reason why this is not performed in this work. Without exact knowledge about the depicted real-life 3D scene properties it would be impossible to conclude whether differences between the synthesized II-picture and its physical reference were caused by limitations in the II-camera model, or simply as the result of different scene properties. Hence, objectively evaluating the II-camera model is considered outside the scope of this work.

A useful II-camera model must copy all the essential properties of a real world II-camera. If not, the value of the resulting II-pictures could be questioned. However, defining "essential properties" is a process that is strongly correlated to the intended application, as shown in the previous works of Section 2.3. The essential properties of the II-camera model presented in this dissertation can be derived by the problem statements P1a and P1b in Section 1.2. The properties are

- Flexible – the model must be able to describe cameras from a large set of different II-technologies.
- Self-contained – the camera must be decoupled from the depicted 3D scenes.
- Non-complex – the II-camera model representation should be easy to use, store and distribute.
- Scalable – the II-camera model should be possible to extend with new functionality.

These properties are evaluated for the different synthesis approaches at the end of this chapter.

3.3 II-camera model

A generic ideal II-camera may be discerned when different types of II-cameras are studied. Based on such a study, I define an II-camera model as consisting of two sets of components:

1. A set of K pixel arrays $\mathcal{I} = \{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_{K-1}\}$.
2. A set of optical elements $\mathcal{O}$.

This broad definition are narrowed down by two constraints in order to reduce complexity:

1. The spatial resolutions are equal for all of the planar pixel arrays in $\mathcal{I}$ and set to $M \times N$ pixels.
2. The interaction between the set of pixel arrays $\mathcal{I}$ and the optical elements $\mathcal{O}$ is described using geometrical optics.

The two sets of components allow for both time-static and time-dynamic II-techniques to be modeled. Different subsets of the model would be active at different times for dynamic systems. The following presentation of the II-camera model focuses on static II-techniques but it is relatively straightforward to extend it to dynamic techniques.

It was stated in Section 2.2.2 that any type of camera could be modeled by properly sampling the plenoptic function. Hence, this is the starting-point for deriving the II-camera model. The plenoptic function is rewritten here for clarity:

$$P(\theta, \phi, \lambda, t, V_x, V_y, V_z). \tag{3.1}$$

Equation (3.1) can without loss of generality be simplified, which is the goal of the following derivation steps. A more compact expression can be achieved if the function arguments are vectorized.

Firstly, the direction coordinates are rewritten using Cartesian to spherical transformation. This increases the dimensionality of the function by one but facilitates the later use of the II-camera model. Thus, I define a direction vector

$$\vec{D} = [D_x, D_y, D_z]^T \tag{3.2}$$

that relates to the direction angles as

$$\begin{aligned} \theta &= acos\left(\frac{D_z}{\left|\vec{D}\right|}\right) \\ \phi &= atan\left(\frac{D_y}{D_x}\right), \end{aligned} \tag{3.3}$$

where $\left|\vec{D}\right| = \sqrt{D_x^2 + D_y^2 + D_z^2}$, i.e. the length of the direction vector $\vec{D}$. Constraining the direction vector to be of length one ($\left|\vec{D}\right| = 1$) gives us $\theta = acos(D_z)$.

Secondly, combining the position coordinates V_x, V_y and V_z into a location point gives us

$$\mathbf{L} = [V_x, V_y, V_z]^T. \tag{3.4}$$

Thirdly, the intensity of all visible wavelengths λ is integrated into an approximate RGB-triplet $\mathbf{P} = [P_R, P_G, P_B]$ according to an RGB color model of choice [83]. Using the format of Equation (3.2) and Equation (3.4) in Equation (3.1) and rearranging the arguments, allows for the plenoptic function to be expressed as a compact vector function

$$\mathbf{P} = P(\mathbf{L}, \vec{D}, t), \tag{3.5}$$

where the RGB-triplet $\mathbf{P}$ corresponds to the RGB-color of the light ray passing through point $\mathbf{L}$ from direction $\vec{D}$ at any time t. Equation (3.5) can be further condensed for a static world, i.e. a non-moving II-camera and stationary objects

$$\mathbf{P} = P(\mathbf{L}, \vec{D}). \tag{3.6}$$

This vector function is still a continuous and generic description of the visible world and must be appropriately sampled to represent what is captured by a specific camera type. There is an infinite number of location points $\mathcal{L}$ and for each location point $\mathbf{L} \in \mathcal{L}$ there are an infinite number of direction vectors $\mathcal{D}$. Reducing these sets is the next step in deriving the II-camera model. Based on the second constraint above regarding geometrical optics, the following assumption is formulated: the interior components of the II-camera do not alter the captured light but merely transports it from the exterior (the depicted scene) to the pixel arrays I, via the optical elements O. Hence, it is sufficient to study $\mathbf{P}$ on the boundary between the interior of the II-camera and the exterior scene. The location points on this boundary S are defined as $\mathcal{L}^S$ and are those that could be considered for sampling Equation (3.5). In addition, only a subset of $\mathcal{L}^S$ possess corresponding direction vectors $\mathcal{D}$ that lead to the light finally being captured by the set of pixel arrays $\mathcal{I}$. A portion hit the inner casing of

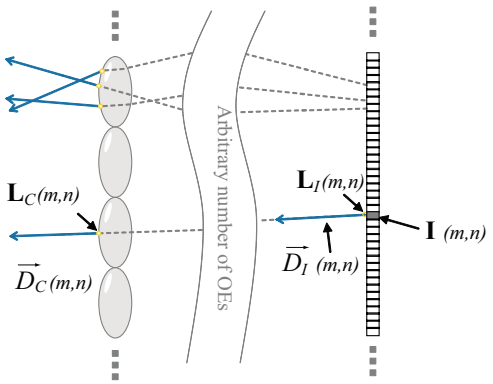


Figure 3.1: Active light rays captured by the k-th pixel array I_k. Location points $\mathbf{L}_C \in \mathcal{L}^C$ and direction vector $\vec{D}_C \in \mathcal{D}^C$ are derived from $\mathbf{L}_I \in \mathcal{L}^{I_k}$ and $\vec{D}_I \in \mathcal{D}^{I_k}$ respectively

the II-camera before reaching any pixel array and therefore do not contribute to the captured II-picture. Thus, for each pixel array $\mathbf{I}_k$ a set of location points $\mathcal{L}^C \subset \mathcal{L}^S$ exists, which has a set of direction vectors $\mathcal{D}^C$ that align with the light rays that intersect $\mathcal{L}^S$ and later becomes captured by an image sensor. We calculate these two sets, for each k, as

$$\begin{aligned} \mathcal{L}^{C_k} &= f\left(\mathcal{L}^{I_k}, \mathcal{D}^{I_k}\right), \\ \mathcal{D}^{C_k} &= g\left(\mathcal{L}^{I_k}, \mathcal{D}^{I_k}\right), \end{aligned} \tag{3.7}$$

where $\mathcal{L}^{I_k}$ and $\mathcal{D}^{I_k}$ are the location points and direction vectors for the pixel array k. The functions f and g describe the operation of the II-camera, i.e. they defines the light transport within the the II-camera's interior. Hence, we can define what is captured by the camera as

$$\mathbf{P}^C = P(\mathcal{L}^C, \mathcal{D}^C) = \left\{P(\mathcal{L}^{I_k}, \mathcal{D}^{I_k})\right\}_{\forall k}. \tag{3.8}$$

That is, the light being captured by the set of pixel arrays $\mathcal{I}$ is fully described by evaluating the plenoptic function at $\mathcal{L}^C$ and $\mathcal{D}^C$. Hence, to fully describe the image $\mathbf{P}_C$ that the II-camera captures on its image sensors does not require an explicit knowledge of $\mathcal{L}^{I_k}$, $\mathcal{D}^{I_k}$, $f()$ and $g()$. Knowing $\mathcal{L}^C$ and $\mathcal{D}^C$) is sufficient. An example of the relationship between the set of $\mathcal{L}^{I_k}$, $\mathcal{D}^{I_k}$ and $\mathcal{L}^C$, $\mathcal{D}^C$ is shown in Figure 3.1. Four light rays are traced through the optical elements from the exterior of the II-camera to the pixel array I_k. Note that evaluating $\mathbf{P}$ at the surface of the pixel array is equivalent to evaluating $\mathbf{P}$ at the outer boundary of the set of optical elements $\mathcal{O}$.

The II-camera model allows for an arbitrary number of pixel arrays or image sensors to be used. However in the subsequent discussion – for clarity – only a single pixel array will be considered, i.e. $K = 1$. However, the extension to $K > 1$ is straightforward. The discrete nature of the pixel arrays implies that the components in $\mathcal{L}^{I_k}$ and $\mathcal{D}^{I_k}$ also pack together into discrete clusters. Even though $\mathcal{L}^{I_k}$ and $\mathcal{D}^{I_k}$

are often uniformly spread due to the periodic structure of the pixel array, $f()$ and $g()$ will most likely cause $\mathcal{L}^C$, $\mathcal{D}^c$ to have non-uniform distribution characteristics.

Hence, defining the sets $\mathcal{L}^C$ and $\mathcal{D}^c$ of the II-camera model requires specifying

- $\mathcal{L}^{I_k}$ and $\mathcal{D}^{I_k}$
- $f()$ and $g()$.

The set $\mathcal{L}^{I_k}$ (which correspond to the positions of the pixels on the k-th pixel array) may simply be constructed using hexagonal or rectangular lattices depending on the structure of the pixel positioning within the array. Moreover, super-sampling gives a more physically accurate II-camera model and may be achieved by considering several location points and direction vectors per pixel (m, n) and averaging the resulting colors. Calculating the two functions $f()$ and $g()$ might range in complexity from piecewise linear transformations of $\mathcal{L}^{I_k}$ and $\mathcal{D}^{I_k}$ to full ray-tracing solutions depending on the II-technique being modeled. The II-camera model does not contain information about *how* the $\mathcal{L}^C$, $\mathcal{D}^c$ are produced by $f()$ and $g()$ from $\mathcal{L}^{I_k}$ and $\mathcal{D}^{I_k}$. The end result is sufficient and therefore defines the resulting discrete II-camera model, which is here described as

$$\mathcal{C} = \left\{ \mathbf{L} \in \mathcal{L}^C, \mathbf{D} \in \mathcal{D}^C \right\}, \tag{3.9}$$

i.e. the sets of location points and accompanying direction vectors that when used to evaluate the plenoptic function $\mathbf{P}$, results in the captured II-picture. If a single pixel array is used and each pixel is considered to capture a single light ray, a one-to-one mapping between pixel (m, n) and location point and direction vector pair is achieved. Equation (3.9) then translates into

$$\mathcal{C} = \left\{ \mathbf{L}(m,n) \in \mathcal{L}^C, \vec{D}(m,n) \in \mathcal{D}^C \right\}. \tag{3.10}$$

A set of presumptions are made to aid in the parametrization of the II-camera model

- The primary pixel array in $\mathcal{I}$ is presumed to be located in the xy-plane, with its center at the origin and with the camera looking down the positive z-axis, i.e. a left-handed coordinate system is used.
- The arbitrary position and orientation of the camera within the virtual scene is outside the scope of the model and is instead handled by transforming $\mathbf{L}$ and $\vec{D}$ into the scene's coordinate system.
- Any additional information required, but not contained in the II-camera model $\mathcal{C}$, is placed in accompanying metadata.

The following subsection will describe how to represent the II-camera model in a form which allows for its practical use. An accompanying metadata structure will also be briefly discussed. Section 3.5 will give a simple example of how to parameterize the II-camera model of Equation (3.10).

3.3.1 II-camera model representation

The II-camera model $\mathcal{C}$, with its finite sets $\mathcal{L}^C$ and $\mathcal{D}^C$, can be arranged in numerous ways and be contained in various data structures. However, by arranging $\mathcal{L}^C$ and $\mathcal{D}^C$ in pixel maps, the II-camera model becomes easily accessible using generic imaging software. With $2 \cdot K$ pixel maps, each having a spatial resolution of $M \times N$, the necessary location points and direction vectors are conveniently stored. The k-th pixel map pair carries the x-, y- and z-components of $\mathbf{L}_k$ and $\vec{D}_k$ in the red, green and blue channels of the RGB-image respectively. A new coordinate system is implicitly used when storing $\mathcal{C}$ in pixel maps, which is the coordinate system that the colors of the pixel map are defined in. A set of $2k$ bounding boxes $\mathbf{B}_k^L$ and $\mathbf{B}_k^{\vec{D}}$ are constructed to enable this. Given that the transformation between the coordinate systems is handled identically for $\mathcal{L}^C$ and $\mathcal{D}^C$, the following description will omit the dependence of $\mathbf{L}$, $\vec{D}$ and k. Hence, we define a bounding box as

$$\mathbf{B} = \begin{bmatrix} x_{\min} & x_{\max} \\ y_{\min} & y_{\max} \\ z_{\min} & z_{\max} \end{bmatrix}, \tag{3.11}$$

where each row corresponds to the bounding box' limits in each dimension, defined in the II-camera model's coordinate system. Maximum precision is achieved by not constraining the bounding box to be square, i.e. by not setting $x_{\min} = y_{\min} = z_{\min}$ and $x_{\max} = y_{\max} = z_{\max}$. $\mathbf{B}$ scales the color-value range of the pixel map format ($[x, y, z]^T$) into the coordinate system range of the II-camera model ($[X, Y, Z]^T$) using

$$[X, Y, Z]^T = \begin{bmatrix} \frac{x}{C_{bpc}} (x_{\max} - x_{\min}) + x_{\min} \\ \frac{y}{C_{bpc}} (y_{\max} - y_{\min}) + y_{\min} \\ \frac{z}{C_{bpc}} (z_{\max} - z_{\min}) + z_{\min} \end{bmatrix}, \tag{3.12}$$

where the normalizing constant C_{bpc} corresponds to the bits per channel (bpc) used by the image format storing the pixel maps. Thus, image formats using 8 bpc to represent a pixel's color results in $C_{bpc} = 2^8 = 256$. Choosing an image format is based on the required accuracy for the application in which the II-camera model is to be used. Hence, the precision and dynamic of the format's color representation must be sufficient to describe all the essential variations in $\mathcal{L}^C$ and $\mathcal{D}^C$. Selecting an image format is outside the scope of the II-camera model. However, for the II-camera model to be practically usable the synthesis application using it must be able to read the selected image format.

The representational form of the II-camera model contains a set of pixel maps and accompanying bounding boxes, where the latter constitute metadata vital for the representational form but not suitable for storing in pixel maps. The small amount of metadata makes the storage in binary data-structures unnecessary. Plain text was instead adopted as it provides easy access to the information using generic software. If the selected image format supports metadata could be stored in the same file as the either of the pixel maps. If not a plain text file could accompany the two pixel maps.

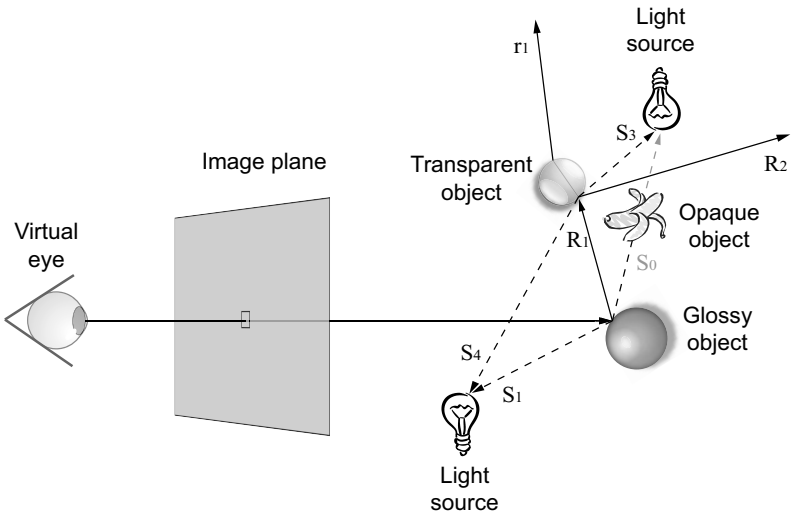


Figure 3.2: An overview of the recursive operation of ray-tracing.

A separate text file was adopted in this work to reduce complexity in accessing the information. Supplementary information (in addition to the $2k$ bounding boxes) could also be included in the metadata, e.g. the pixel and lenslet pitch of the II-camera, the Index-of-Refraction (IoR) of the lenslets etc.. Such information might not be required for synthesizing II-pictures per se but may be useful for various signal processing operations, for example synthesizing novel 2D views from the II-picture.

3.4 II-picture synthesis

After defining the II-camera model $\mathcal{C}$ and representing it using a set of pixel map pairs with accompanying metadata, a way to evaluate the plenoptic function $\mathbf{P}$ in Equation (3.5) still remains to be provided. A synthesis software application is required to make use of the model. Such an application is presented in the following two subsections. First the core of the application is described – the open source ray-tracing software package MegaPOV . After that the general structure of the application itself is discussed.

3.4.1 Ray-tracing using MegaPOV

Rays are traced from the virtual eye, through the image plane, and into the scene to be depicted when a synthetic image is rendered using ray-tracing, as illustrated by Figure 3.2. For each pixel a set of primary rays is traced into the scene, which is in the opposite direction with regards to how physical light is captured by an image sensor.

Ray-tracing is a recursive process in which each primary ray that hits a virtual object generates a set of new secondary rays, which is similarly traced within the scene. Hence, it is of vital importance that the exponentially growing set of rays (tertiary, quaternary etc.) is limited. Two conditions stop the recursion: (a) if a ray leaves the scene or (b) if a pre-defined number of recursion steps have been performed. The type of rays generated at each ray-object intersection point, differ based on the physical property of the object and can be characterized into:

- Shadow rays - S; these are traced to the scene's light sources to determine to what degree they contribute to the point's color. Occluding objects are identified using shadow rays.

- Reflection rays - R; a glossy surface gives rise to a reflection which alters the intersection point's color.

- Refraction rays - r; a transparent object bends light based on the relation between the refraction indices of the object and the surroundings.

These three ray types are all represented in Figure 3.2. The colors calculated at each intersection point are traversed back and summed until the first intersection point is reached. All color terms from all intersection points add together and result in the color of the image pixel through which the primary ray was cast. An increased correspondence with a real-life camera is achieved by using several rays per pixel, as mentioned in Section 3.3. Increasing the number of rays traced per pixel increases the resulting image quality, but at the expense of increased rendering time.

MegaPOV is "a collection of unofficial extensions of POV-Ray" [84], which is an open-source multi-platform ray-tracing software application [85]. Three major benefits are achieved as a result of basing the synthesis application on MegaPOV:

1. Optically accurate synthesis provided by POV-Ray, which includes refraction, refraction, caustics etc.

2. Possibility to define scenes of an arbitrary choice, complexity and precision using POV-Ray's generic Scene Description Language.

3. Access to a user defined camera projection type, which is a feature present in MegaPOV that allows a ray to be traced from any position in any direction within the virtual scene.

The user defined camera projection type in MegaPOV enables the plenoptic function $\mathbf{P}$ to be evaluated using the II-camera model $\mathcal{C}$. Hence, this is a key feature for which MegaPOV was selected. The large community supporting POV-Ray with scenes, objects, enhanced functionality, etc. is other features that is an advantage for MegaPOV.

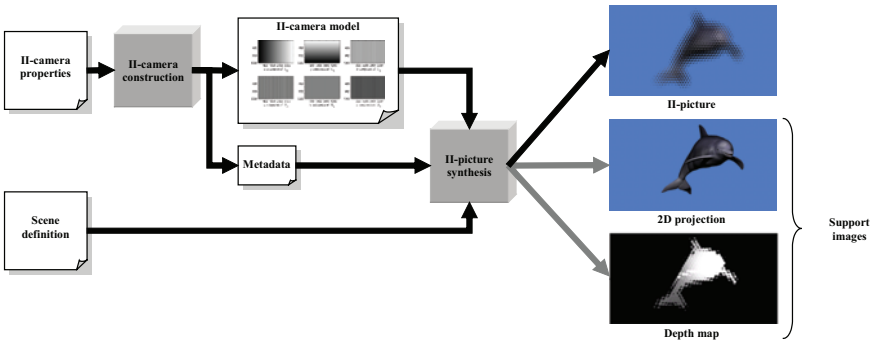


Figure 3.3: II-camera properties combine with scene description to form a II-camera model, which is used to synthesize an II-picture and additional support images.

3.4.2 Integrating II-camera model and MegaPOV

MegaPOV allow for rays to be shot into the scene from any position and in any direction. The II-camera model describes location points $\mathcal{L}^C$ and direction vectors $\mathcal{D}^C$, which produce an II-picture when used to evaluate the plenoptic function $\mathbf{P}$ in Equation (3.5). Hence, the ray tracer could be used to evaluate $\mathbf{P}$ by adapting the representation of the II-camera model to fit MegaPOV. Changing representation format was accomplished by a set of adaptation and extension macros, which was developed in POV-Ray's Scene Description Language (SDL). Despite incorporating $\mathcal{C}$ into the framework of the ray-tracer, additional functions such as metadata handling, II-camera positioning and orientation were also developed.

Figure 3.3 shows the information flow that starts with the II-camera properties and scene description and ends with a synthesized II-picture. The model with accompanying metadata is parameterized in the II- camera construction and the II-camera model is adapted to MegaPOV, which then performs the rendering of the II-picture. The resulting II-picture can be viewed or saved for later use. Note the additional information, or support images, that are able to be produced concurrently with the rendering of the II-picture. One example is a perspective projection overview of the virtual scene, which might be used to visually verify the accuracy of the scene setup. Another example is a depth map that provides scene-depth on a pixel per pixel basis. Having access to a so called ground truth regarding the scene's depth is vital when for example evaluating II-based depth-extraction algorithms or for post-processing of synthesized II-pictures.

If the proposed II-camera model is constrained to only be parameterized as a pinhole 2D camera and the set of optical elements $\mathcal{O}$ are considered to be objects of the virtual scene the proposed approach becomes similar to that presented in [73]. However, at this point there is no longer a clear separation between the camera- and scene-model, which results in a synthesis solution that lacks the required self-contained property of the II-camera model.

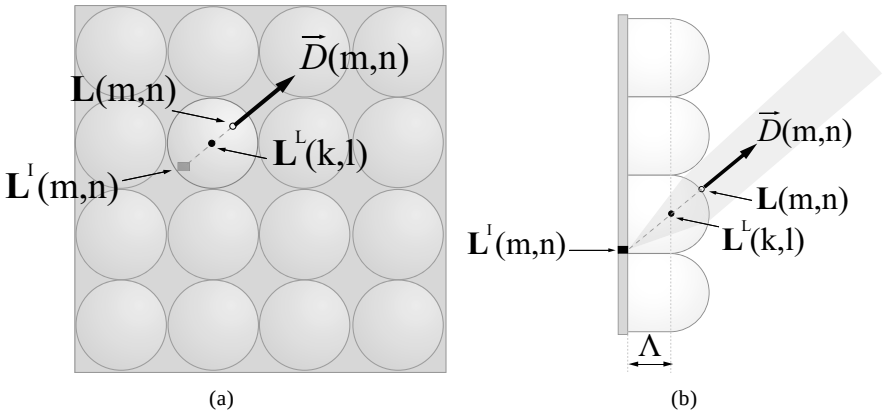


Figure 3.4: Geometry of a II-camera exemplifying how to parameterize the proposed II-camera model

3.5 Example of II-camera model parametrization

The II-camera to be modeled has lenslets packed in a rectangular pattern directly at a gap distance Λ from the pixel array. Note that the naive II-camera to be modeled is chosen to elucidate the II-camera parametrization process and not to be representative of what could be modeled using the II-camera model.

Initially, the two sets $\mathcal{L}^I$ and $\mathcal{D}^I$ must be defined, as they provide the input to the functions $f()$ and $g()$. The location $\mathbf{L}^I(m,n)$ of each of the $M \cdot N$ pixels may be defined in many different ways. In this example we use a generating matrix G^P defined as

$$G^P = \begin{bmatrix} \delta_x^p & 0 & 0 & -\frac{\Delta_x^p}{2} \\ 0 & -\delta_y^p & 0 & \frac{\Delta_y^p}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.13}$$

where Δ_x^p and Δ_y^p are the horizontal and vertical sizes of the pixel array respectively and $\delta_x^p = \frac{\Delta_x^p}{M}$ and $\delta_y^p = \frac{\Delta_y^p}{N}$ are the horizontal and vertical pixel pitch. A location point is then calculated according to

$$\mathbf{L}^I(m,n) = G^P \cdot \begin{bmatrix} m \\ n \\ 0 \\ 1 \end{bmatrix}. \tag{3.14}$$

A few approximations are utilize in the process of defining the direction vector $\vec{D}^I(m,n)$. These approximations are inferred from Figure 3.4 and enables the calculation of $\mathbf{L}(m,n)$ and $\vec{D}(m,n)$ *without* having to explicitly derive $\vec{D}^I(m,n)$ or defining analytical expressions for the two functions $f()$ and $g()$.

Note that light that enters a lenslet perpendicular to the lenslet's surface transfers straight through the lenslet without undergoing refraction. This implies for a spherical lenslet that light rays parallel to this normal-ray will refract to a focal point located *on* the normal-ray at a distance from the surface corresponding to the lenslet's focal length. Setting the gap Λ (see Figure 3.4) equal to the focal length of the lenslets coincides the focal point with the point where the normal-ray intersects with the pixel array plane. Hence, two simplifications can be made. Firstly, the direction vector $\vec{D}(m,n)$ is calculated as

$$\vec{D}(m,n) = \mathbf{L}^L(k,l) - \mathbf{L}^I(m,n), \tag{3.15}$$

where $\mathbf{L}^L(k,l)$ is the center point of the lenslet under which $\mathbf{L}^I(m,n)$ is located. Secondly, evaluating the plenoptic function $\mathbf{P}$ at $\mathbf{L}(m,n)$, $\mathbf{L}^L(k,l)$ or $\mathbf{L}^I(m,n)$ will produce the same results using the direction vector given by Equation (3.15), as Figure 3.4 also shows. For simplicity the lenslet center is therefore selected as the location point, i.e $\mathbf{L}(m,n) = \mathbf{L}^L(k,l)$. This selection translates the location points from the surface of the optical element to their interior. However, this is a feasible operation since the translation does not affect the II-picture produced by the II-camera model. Thus, what remains is to define $\mathbf{L}^L(k,l)$ as a function of the pixel (m,n). Again, a generating matrix G^L is used according to

$$G^L = \begin{bmatrix} \delta_x^L & 0 & 0 & -\frac{\Delta_x^P}{2} \\ 0 & -\delta_y^L & 0 & \frac{\Delta_y^P}{2} \\ 0 & 0 & 1 & \Lambda \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.16}$$

where δ_x^L and δ_y^L are the horizontal and vertical lenslet pitch respectively. Equation 3.16 now provides the means to find the center point of the (k,l)th lenslet using

$$\mathbf{L}^L(k,l) = G^L \cdot \begin{bmatrix} k \\ l \\ 0 \\ 1 \end{bmatrix}. \tag{3.17}$$

Now, what remains to determine is which lenset (k,l) corresponds to the current pixel (m,n). Given that the pixel and lenslet arrays are located on parallel planes differing only by a translation Λ along the z-axis, Equations (3.13) and (3.16) are combined into

$$G^{P\to L} = (G^L)^{-1} \cdot G^P. \tag{3.18}$$

From Equation (3.18) the lenslet nearest to the pixel (m,n) is found using

$$[k,l]^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot round\left(G^{P\to L} \cdot \begin{bmatrix} m \\ n \\ 0 \\ 1 \end{bmatrix} \right), \tag{3.19}$$

where $round()$ is the nearest integer function rounding the elements of the argument vector into the nearest integers [86]. Substituting k and l from Equation (3.19) into

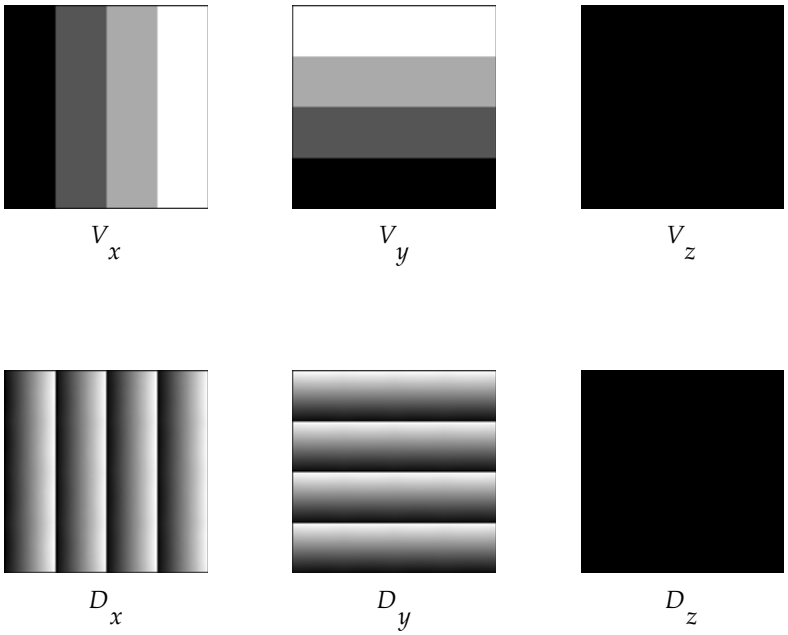


Figure 3.5: Example sets $\mathcal{L}^C$ and $\mathcal{D}^C$ decomposed into x-, y- and z-coordinates.

Equation (3.17) finally gives the lenslet position $\mathbf{L}^L(k,l)$ that is required to determine the direction vector $\vec{D}(m,n)$ in Equation 3.15 by

$$\vec{D}(m,n) = G^L \cdot [k,l,0,1]^T - G^P \cdot \begin{bmatrix} m \\ n \\ 0 \\ 1 \end{bmatrix}. \tag{3.20}$$

The bounding boxes $\mathbf{B}^l$ and $\mathbf{B}^d$ defined by Equation (3.11) are now easily constructed.

The II-camera model derived using Equations (3.17) and (3.19) is shown in Figure 3.5 where the resulting two sets $\mathcal{L}^C$ and $\mathcal{D}^C$ have been separated into their RGB-components in order for the x-, y- and z-coordinates of location points $\mathbf{L}(k,l)$ and direction vectors $\vec{D}(m,n)$ to be more clearly visible. Note the monotonically stepwise increasing gray scale in the x- and y-component of $\mathcal{L}^C$. This corresponds to the uniform placement of location points and the fact that the II-picture pixels corresponding to an EI all share the same location point equal to a specific lenslet center. As a result of selecting the lenslet centers as location points, there are no variations in the z-component. The pixels in V_z all correspond to the gap distance Γ. If $\mathcal{L}^C$ had been located on the surface instead of in the centers of the lenslets, a variation corresponding to the height profile of the lens array had appeared in V_z.

With regards to the direction vectors of the II-camera model, the x- and y-components of $\mathcal{D}^C$ makes the ray directions span from left to right and bottom to top within the range of the lenslet's viewing angle α. Similar to $\mathcal{L}^C$, no variation occur in D_z for this II-camera model. When defining $\vec{D}(m,n)$ as in Equation 3.15, no variations exist within either of the two terms z-components. That is, $\mathbf{L}^L(k,l) = \Gamma \forall \{k,l\}$ and $\mathbf{L}^I(m,n) = 0 \forall \{m,n\}$ and thus, all z-components in $\mathcal{D}^C$ correspond to the gap distance Γ. Note that $\mathcal{D}^C$ is sufficiently defined using only two components corresponding to the latitudinal and longitudinal angles ϕ and θ respectively. Hence, any of the three components of $\mathcal{D}^C$ may be set to a constant value as long as this normalization is reflected in the other two components as well, keeping the direction of the vector unchanged.

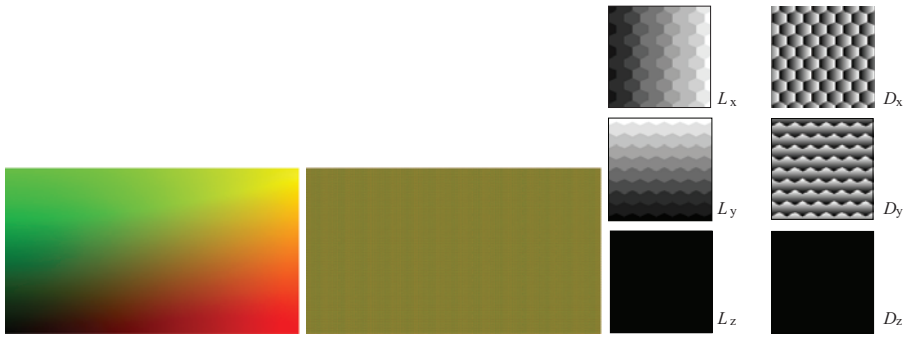
When $\mathcal{L}^C$ and $\mathcal{D}^C$ are to be stored, both integer-based formats and floating point based formats can be used as discussed in Section 3.3.1. For this example (as for the subsequent synthesis in Chapter 5) the Portable Network Graphics (PNG)-format with 16 bpc ($C_{bpc} = 65536$ in Equation (3.12)) was used [87]. Using a floating point based image format was not necessary for producing reference II-pictures for the II-picture coding, as will be discussed in Chapter 5. However, the high-dynamic range image format RGBE is preferably used for applications or II-techniques requiring large dynamics in specifying $\mathcal{L}^C$ and $\mathcal{D}^C$. RGBE uses one byte per channel together with a one byte shared exponent, i.e. 32 bit per pixel. MegaPOV supports RBGE and thereby enables high dynamic II-camera models to be used for any application that so requires.

3.6 Results

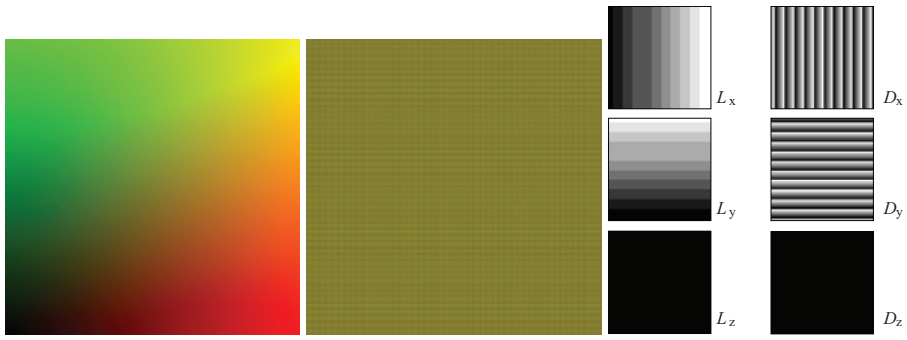
3.6.1 II-camera models

Three different models are presented here, showing that II-based camera systems proposed in the literature may be described using the II-camera model. The first is based on the properties of an II-based high resolution video system presented by Okano et al. [37]. The characteristics of the second II-camera model are set to be similar to those for the plenoptic camera in [35]. The third model is presented showing that the II-camera model also has the ability to describe other camera systems, including conventional 2D projection cameras.

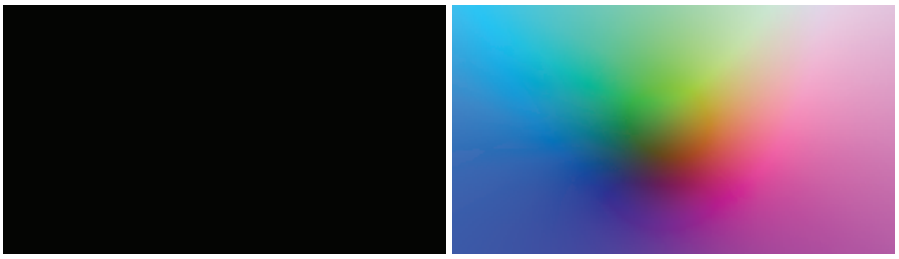
A summary of the properties used to produce the II-camera models is presented in Table 3.1 and Figure 3.6 shows the accompanying II-camera models. Note the gradient from black to yellow (red+green) in the two top II-camera models. This indicates that the location points are distributed over the xy-plane. The reason for that the end of the gradient is not white (red+green+white), is because of the lack of variation in the models z-components. The apparent lack of information in D_z is a result of the normalization causing the angular information to be carried solely in D_x and D_y. The 2D pinhole approximation has the set of $\mathbf{L}$ set to a single point (the pinhole) and thus do not contain any variation at all in either of the x-, y-, and z-components.



(a)



(b)

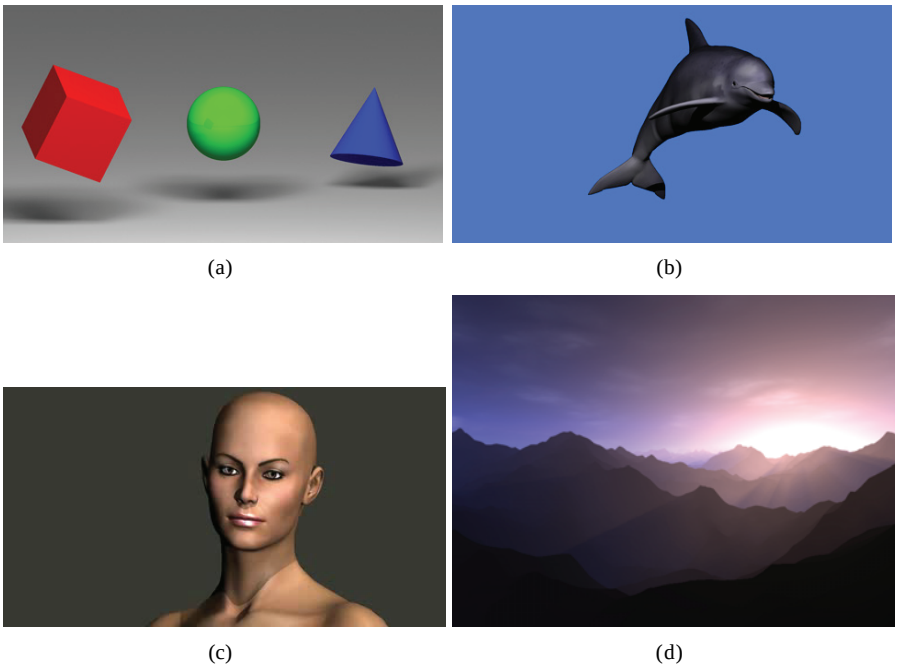


(c)

Figure 3.6: II-camera models parameterized to describe a set of different cameras: (a) a II-based high definition video camera [37], (b) a still plenoptic camera [35] and (c) a 2D pinhole approximation. The two first columns from left to right show $\mathcal{L}^C$ and $\mathcal{D}^C$. Subfigure (a) and (b) also show a zoomed in portion of the top left corner of $\mathcal{L}^C$ and $\mathcal{D}^C$ respectively.

Table 3.1: II-system parameters.

Parameter	II HD video	Plenoptic camera	Pinhole 2D
Pixel array resolution	3840×2160	4096×4096	8192×4608
No. of lenslets	160×125	296×296	1×1
EI resolution	$\sim 24 \times 17$	$\sim 14 \times 14$	n/a
Lenslet positioning	hexagonal	rectangular	n/a



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

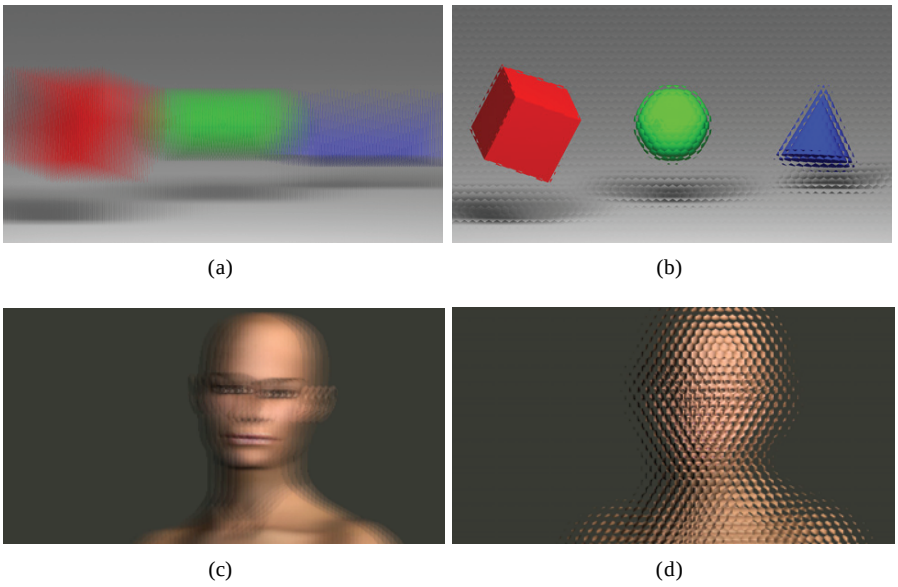
Figure 3.7: Example scenes (a) Objects, (b) Dolphin, (c) Hairdo and (d) Mountains [88].

3.6.2 Virtual scenes

A set of virtual scenes is presented, using 2D projection, in Figure 3.7. An additional set of scenes, used as reference input in the subsequent coding chapter, is shown as perspective projections in Figure 5.9 on page 110. Note that the complexity of the scenes described by the SDL can range from a simple geometrical primitive (Objects) to fully textured meshes with a large number of vertices (Hairdo).

3.6.3 Synthesized II-pictures

Finally, Figure 3.8 shows a few example II-pictures that illustrate the possibility in interchangeably altering the II-camera and the scene to be depicted. The II-pictures



(a) (b)

(c) (d)

Figure 3.8: II-pictures synthesized with different II-cameras and virtual scene definitions.

in the first column are synthesized using an HPO II-camera. The second column's II-pictures are from a GRIN-based II-camera. From top to bottom, the two rows depict the scenes Objects and Hairdo respectively. An observed II-property is that a larger object-camera distance leads to an object being projected into a larger set of EIs. This becomes clear when comparing the red cube with the blue cone in Figure 3.8 (b). In Figure 3.8 (a) the HPO II-technique's ability to only provide horizontal parallax can also be observed as the lack of vertical spreading caused by any of the objects. It is still the case that the horizontal spreading increases with increasing distance between the object and the II-camera.

3.6.4 Comparison between synthesis approaches

The presented results show that the proposed II-camera model is capable of producing II-pictures from different II-techniques. In the following table a comparison is made between the proposed synthesis approach and previous work with respect to the four properties defined in Section 3.2. Previous sections have shown that the proposed generic II-camera model can be used as a flexible tool to describe II-cameras, adhering to various II-techniques. II-pictures from different II-techniques can easily be obtained by parameterizing the II-camera model. Combined with the metadata, it is self-contained as a result of being separated from the scene and can hence be interchanged in a non-complex way. Arbitrary complex scenes can be designed for stressing different II-properties due to the use of an open SDL. The framework of a pixel map based II-camera model with metadata allows for additional pixel maps to

Table 3.2: II-picture synthesis comparison chart

	Flexible	Self-contained	Non-complex	Scalable
Basic synthesis [47, 68]	Low	Low	Low	Low
Volumetric synthesis [69, 70]	Low	Medium	Low	Medium
Wireframe synthesis [71]	Medium	High	Medium	Medium
EPI synthesis [12]	High	High	Low	Medium
Raytrace camera+scene [73]	High	Low	Low	High
Proposed II-camera model	High	High	High	High

be added in the future including new aspects of the II-camera processes, e.g. modeling point spread functions and color aberrations. The level of the other methods fulfillment of these properties was set based on their characteristics as presented in Section 2.3.1 on page 44.

3.7 Concluding remarks

A flexible, self-contained, low-complex and scalable approach for synthesizing II-pictures is a valuable complement to experimental research with the II-field.

In this chapter I presented a novel general II-camera model that can be parameterized to represent various different II-technologies. Conventional 2D cameras, camera array systems and other constructs with planar pixel arrays may also be modeled. The important contribution of this model is that it encompasses a multitude of camera system (2D camera arrays and II-based 3D cameras) in the well defined and easily manageable form of a pair of 2D pixel maps. By combining the II-camera model with MegaPOV and its SDL and ray-tracing functionality, gives us a flexible and scalable synthesizing method capable of producing II-pictures of arbitrary size, complexity, and other properties.

Virtual scenes have been defined using the open SDL of MegaPOV, which allows for the design of arbitrary complex reference II-pictures. Combining the II-camera model with the open SDL allows for the defining of reference II-pictures, which is an essential part of inter alia research in coding schemes for II-pictures.

Thus, synthesizing II-pictures allows for the generation of II-pictures in a simple and cost effective way compared to experimental research. Comparing the described synthesis method with the most closely related work, presented by Athineos et al. [73], reveals an important conceptual difference. When they model the II-camera as a part of the virtual scene, two major disadvantages are introduced:

1. The time for synthesizing is made unnecessary long.
2. There is no flexibility in changing II-camera model or 3D scene.

The light rays going through the interior of the II-camera must be re-traced for every II-picture, even when their paths never change as is the case for a time-static

II-camera. In the synthesizing method I have described, the light transport with regards to the II-camera is only calculated once, when the II-camera model is initially constructed. Athineos et al. [73] introduce a dependency that may prove very difficult to break when combining the II-camera model and the scene model into a global world model. Separating the two models is a prerequisite for providing the flexibility and scalability of a synthesis method that the research presented in this dissertation set out to achieve.

The presented approach with a independently modeled II-camera and a virtual scene, offer the most flexible solution to II-picture synthesis from the compared approaches. Thus, the described synthesis method is extensively used in Chapter 5 to synthesize reference II-pictures used for evaluating coding schemes.

3.7.1 Authors contributions

With regards to this chapter my main contributions are:

- A practically useable II-camera model with accompanying meta data, designed to be capable of describing cameras adhering to various different II-techniques.
- A mathematical description of the II-camera model, using the framework of the plenoptic function as a basis.
- A modular synthesis method that utilizes the defined II-camera model in combination with ray-tracing and a Scene Description Language capable of synthesizing II-pictures of arbitrary type and complexity.
- An example set of II-camera models, virtual scenes and II-pictures.

I have presented large parts of this work in Papers I and II.

3.7.2 Problem definitions – P1a and P1b

How can the scene, the II-system, and the II-based 3D image be decoupled to aid the comparison of 3D images produced by different II-techniques? The chapter has presented a modular synthesis approach where the virtual scene is explicitly decoupled from a proposed II-camera model. This division allows for the production of different II-pictures depicting a given scene, yet adhering to numerous different II-techniques. In addition, scenes may easily be designed to explicitly stress specific II-properties.

Can such a decoupling be used to provide a supply of II-based 3D images, which for example would facilitate research on coding methods? Given the modularity of the presented synthesis method, it may be easily used to produce II-pictures as the input in developing coding schemes for II-pictures. Chapter 5 uses the work presented in this chapter extensively, both for producing II-pictures of a specific II-technique as well as for providing a basis for comparative studies of similar II-techniques. Complementary information about the II-picture can be produced, e.g. ground-truth pixel-wise

depth information that may be used to validate the results from subsequent image processing algorithms.

Chapter 4

Evaluation

Having the means to evaluate a coded II-pictureis as important as having II-pictures to code. Without proper evaluation metrics, the performance of any constructed coding scheme is undefined.

4.1 Chapter outline

This chapter discusses the means to objectively evaluate distorted II-pictures caused by for example lossy coding. Firstly an overview of presently used metrics is given in Section 2.3.2. This is then followed by Section 4.3, which presents two novel objective quality metrics constructed to reveal distortion effects not possible with the metrics proposed in the literature. A qualitative discussion of the metrics characteristics is given in 4.4. Finally, concluding remarks are given in Section 4.5, which also summarizes the authors contributions.

4.2 Methodology

Evaluating the quality of a 3D image, which has been subjected to any type of lossy coding, may be performed principally in two different ways:

1. Using a strictly defined algorithm to calculate the extent of the difference between the original and coded image.
2. Allowing a group of people, specifically chosen to be typical users of the intended application, to asses the quality of the coded image.

The first approach is also known as objective testing and generates strictly reproducible results, which is essential in experimental and comparative research. The purpose of the objective test metric may be to extract the distortion with regards to

specific properties of the image and thereby provide a deeper insight into the distortion inducing process. The purpose might also be to predict user tests with regards to subjective quality. However, finding an algorithm that captures all the essential properties of the complex processes taking place in the HVS is a yet unresolved research problem [89]. If a sufficiently good model can ever be found is open to debate.

Regardless of which aim the objective tests have, the second approach of conducting subjective tests is an important complementary quality evaluation method. More so if the goal of the objective test metric is to predict a viewer's conception of quality. By definition, the subjective method will produce quality results adhering strongly to what an average viewer perceives in terms of 3D image quality; as long as there are sufficiently high numbers of viewers, the 3D images to be viewed are appropriately designed, the environment where the tests are conducted are explicitly controlled, etc.. Thus, conducting a trustworthy subjective test requires that a multitude of aspects with respect to designing the test is properly considered. If any of these factors are not met, the subjective test results are invalid.

The two novel quality metrics that will be presented in this chapter will adhere to the described objective approach. Contrary to the metrics in the related work they will strive to explicitly measure the distortion with respect to the II-picture properties. As a result, they are also likely to better correspond to subjective test results as they indicate differences in quality with regards to explicit II-picture properties such as depth; properties that a viewer may prove to consider important as they convey the special character of the 3D image and thereby strongly contributing to the subjective quality. However, no formal and extensive subjective test will be conducted to verify the correctness of this hypothesis.

The two quality metrics will instead be evaluated using a qualitative discussion about their properties compared to present metrics. In addition, a quantitative empirical analysis will be performed in the next chapter on coding, where the metrics are utilized in evaluating coded II-pictures. The main reasons for excluding the certainly valid subjective evaluation method are time and resources. If proper subjective tests are to be conducted, an experiment must be designed such that sufficient information can be inferred and well-founded conclusions can be made about the evaluated quality. A vital part of this design is to have access to a physical II-display with the necessary properties such that it corresponds to the II-technology evaluated. No such II-display was available at the time when the work presented in this dissertation was conducted. Emulating the II-display using stereoscopic display was considered but the idea was postponed. The quality of the evaluation would be affected by the limited properties of the stereoscopic technique, e.g. the cross-talk introduced by imperfect time synchronization of shutter glasses. Furthermore, the extent of this contamination with regards to measured quality would be unknown. However, small scale qualitative studies were performed by visually inspecting synthesized stereoscopic views of coded II-pictures using cross-eyed viewing. These studies were mainly conducted as a form of validity check in the design of coding schemes for II-pictures.

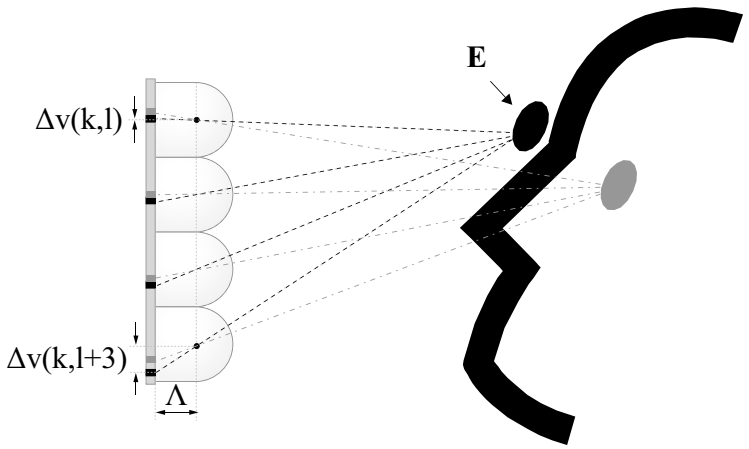


Figure 4.1: II-display with pixel and lens array. Rays entering a viewpoint $\mathbf{E}$, after intersecting all lens centers, construct a VI.

4.3 Metrics for II-picture evaluation

Note that neither Q_{global} nor $\mathbf{Q}_{angle}$ are applied to images that are directly viewed when watching a II-display. The main property of a II-display is to present only a subset of the II-picture for the viewer using the demultiplexing lens array. Hence, a viewer merely sees a subset of all II-picture pixels at any given viewing position. Moreover, a physical viewer is never positioned at an infinite distance to the II-display.

The concept of View Image (VI) is defined based on these observations. Within the viewing space are an indefinite number of VIs; each derived based on the viewer's position relative to the display. Figure 4.1 shows how a viewpoint $\mathbf{E}$ determines a VI and how it is constructed from a subset of II-picture pixels. Note that two different VIs perceived by a viewer at a specific location are constructed from different subsets of the II-picture pixels. The disparity between the two causes the perceived depth of the 3D image.

The set of VIs are explicitly defined in this work as

$$\begin{aligned}\mathbf{VI} &= [VI_{\mathbf{E}}(k,l)]_{\substack{k=0,1,\ldots,K-1\\ l=0,1,\ldots,L-1\\ \forall \mathbf{E}}} \\ &= II\left(k\cdot U+\frac{U}{2}+\Delta u(k,l), l\cdot V+\frac{V}{2}+\Delta v(k,l)\right),\end{aligned} \tag{4.1}$$

where $\mathbf{E}=[E_x,E_y,E_z]^T$ is the 3D coordinate of any given viewpoint [90]. This definition implies that the pixel panel of the II-display is positioned to coincide with the xy-plane (see Figure 4.2) and a single EI-pixel is seen through each EI concurrently,

as shown in Figure 4.1. We calculate the pixel-offsets relative to each EI-center using

$$\begin{aligned}\Delta u(k,l) &= \Lambda \cdot \frac{(L_x(k,l) - E_x)}{L_z(k,l) - E_z} \\ \Delta v(k,l) &= \Lambda \cdot \frac{(L_y(k,l) - E_y)}{L_z(k,t) - E_z},\end{aligned} \tag{4.2}$$

where Λ is the gap distance between the pixel and lens array and

$$\mathbf{L}^L(k,l) = [L_x(k,l), L_y(k,l), L_z(k,l)]^T$$

is the position of the (k,l)-th lens center. Linear interpolation is adopted to handle non-integer pixel offsets from Equation (4.2). Two examples of how different lenslets produce different Δv are shown in Figure 4.1.

4.3.1 Sparse angle dependent quality

The first proposed quality metric is directly based on the definition of VIs and aims to evaluate the coding induced distortion as seen by a viewer located within the II-display's viewing space.

4.3.1.1 Constructing a representative VI set

The indefinite number of VIs makes it necessary to limit the set in order for the quality metric to be computationally tractable. A sparse set of five VIs is used when forming a quality metric that provides a sampled yet informative view of the perceived quality. The view points selected in order to produce the VI sets are: $\mathbf{E}_{front}$, $\mathbf{E}_{up}$, $\mathbf{E}_{down}$, $\mathbf{E}_{left}$, $\mathbf{E}_{right}$. All views are equidistant to, and aimed at, the II-display at a distance r giving rise to similarly sized VIs. $\mathbf{E}_{front}$ corresponds to a view point located on the normal to the center of the II-display whereas $\mathbf{E}_{left}$, $\mathbf{E}_{right}$ are viewpoints rotated a longitudinal angle $\pm\phi$ with respect to the normal. Analogously, up and down are defined with a latitudinal rotation angle of $\pm\theta$. The distance (r) and angles (ϕ and θ) are parameters of the metric and should be set such that the resulting view points are evenly distributed within the viewing space and the II-techniques designed viewing distance. Hence, the three parameters will differ for different II-techniques. See Figure 4.2 for a geometrical overview regarding how the five VIs are positioned relative to each other. Figure 4.3 shows the corresponding VIs for the example. Note the horizontal and vertical motion parallax inherent in the five VIs, where the two women's position relative to each other changes from left to right; their position relative to the horizon also changes from top to bottom.

4.3.1.2 Assessing the quality of a VI

The selected VIs correspond much more to what is physically seen when watching an II-display than an II-picture or any SI. This warrant the use of a distortion assessment

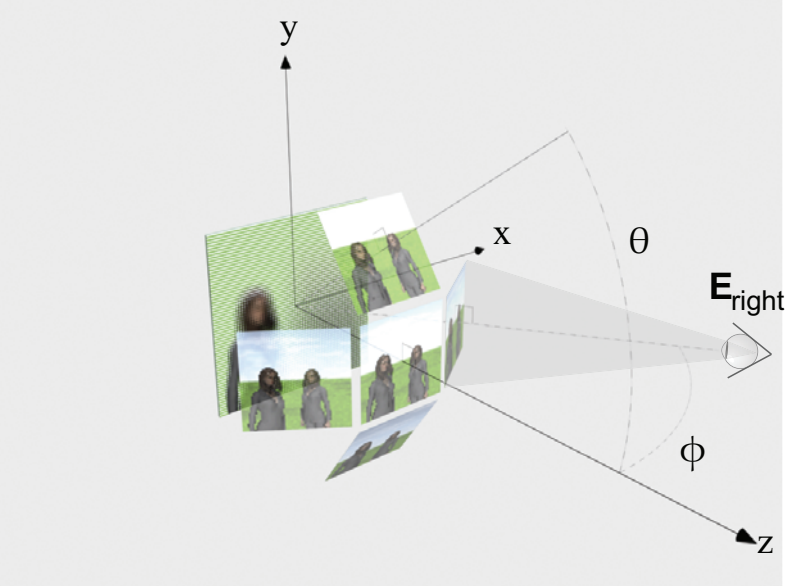


Figure 4.2: Schematic representation of the five VIs used to evaluate the coding artifacts. Right view point $\mathbf{E}_{right}$ is explicitly marked.

scheme that offers high correlation with subjective user-tests. Due to the prevalence of PSNR in coding research, it has a distinct place as a reference metric. However, there are other metrics that provides a stronger correlation with subjective tests than PSNR[91–93]. The gray scale Mean Structural SIMilarity index (MSSIM) proposed by Wang et al. [94] has shown good correspondence with subjective tests on 2D images coding quality [95]. As a result, it is selected to be applied to the set of VIs defined in the previous section.

In the following a brief description of MSSIM is presented. Figure 4.4 illustrates the main steps in calculating MSSIM. Signal x correspond to the original image and $\hat{x}$ the distorted counterpart. Three properties of the two images are compared for similarity:

- Luminance
- Contrast
- Structure

A set of operations is performed on each image hierarchically such that the properties are extracted and made available for the three comparisons. The luminance of an image is estimated using the mean intensity of the image. Contrast is estimated by calculating the standard deviation of the image after its luminance from the previous step has been removed. Both the luminance- and the contrast estimation are

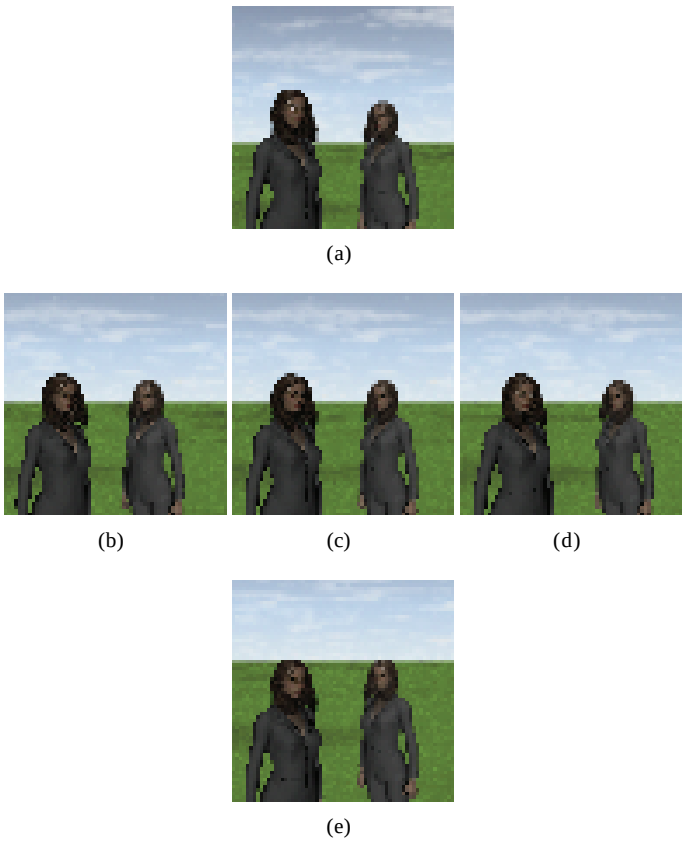


Figure 4.3: Reference II-picture Twins decoded from view point (a) $\mathbf{E}_{up}$, (b) $\mathbf{E}_{left}$, (c) $\mathbf{E}_{front}$, (d) $\mathbf{E}_{right}$ and (e) $\mathbf{E}_{down}$. The coarseness of the VIs (64×64 pixels) are due to the used II-picture structure ($K \times L = 64 \times 64$).

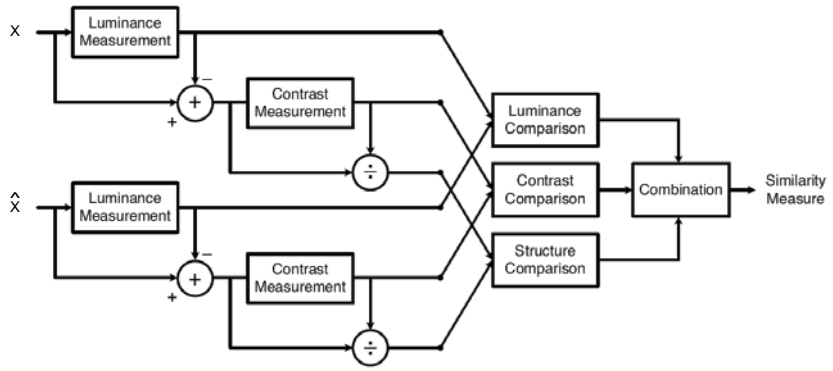


Figure 4.4: The operations used to evaluating MSSIM on a distorted 2D images $\hat{x}$ and its original x [94]

performed using a sliding-window operation to capture their local variations. Removing the contrast from the luminance-normalized image gives a residual image in which the structure is now uncovered. Hence, the two images x and $\hat{x}$ each give rise to three images that isolates the luminance, contrast and structure part of the image respectively. The three image pairs are mutually compared using three comparison functions and the results are weighted and combined into a pixel map with an SSIM index value for each pixel. A single quality-value MSSIM is achieved by computing the average value of all the SSIM index values. The fact that MSSIM is based on a sliding-window principle allows it to capture inter-pixel effects that the pixel-to-pixel difference approach of PSNR is unable to do. In this work the default parameters of MSSIM were adopted using the MATLAB-function provided by Wang et al. [94], on the web page accompanying the paper. The interested reader is referred to the paper for more explicit details regarding MSSIM.

4.3.1.3 Combining operations

The original and distorted II-pictures give rise to five VIs each, according to Section 4.3.1.1. Evaluating the MSSIM on this set gives rise to a quality vector that I define as

$$\begin{aligned}\mathbf{Q}_{view} &= [Q_{view}(\mathbf{E})]_{\mathbf{E}=\{\mathbf{E}_{top},\mathbf{E}_{front},\mathbf{E}_{bottom},\mathbf{E}_{left},\mathbf{E}_{right}\}} \\ &= \left[\mathrm{MSSIM}(\mathbf{VI}_{\mathbf{E}}, \widehat{\mathbf{VI}_{\mathbf{E}}})\right]_{\mathbf{E}=\{\mathbf{E}_{top},\ldots,\mathbf{E}_{right}\}},\end{aligned} \tag{4.3}$$

where $VI_{\mathbf{E}_n}$ and $\widehat{VI}_{\mathbf{E}_n}$ are the VIs constructed from the original and distorted II-picture respectively. In this form Equation (4.3) resembles Q_{angle} in Equation (2.18) but with a more physically founded location of the metrics implied virtual viewer. In addition the VI is a perspective projections, which the SI used in Equation (2.18)

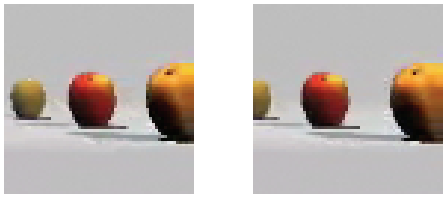
is not. If a scalar quality-value is required, for example in rate-distortion analysis, the arithmetic mean $\overline{\mathbf{Q}}_{view}$ may be employed. Using $\overline{\mathbf{Q}}_{view}$, contrary to Q_{global}, would produce a result that considers the spatial demultiplexing performed by the II-display. However it should be noted that the quality assessment provided by Equation (4.3) passes a verdict on a subset of all pixels from the II-picture, contrary to Q_{global} and $\mathbf{Q}_{angle}$ that include all pixels in the II-picture in their calculations. Section 4.4 will qualitatively evaluate this sparse angle-dependent MSSIM-based quality metric, which will be used at a later stage to evaluate the coding-induced distortion in Chapter 5.

4.3.2 Sparse pseudo-depth dependent quality

The previously described metric addresses the motion parallax of the II-picture with its angle-dependent characteristic, while this second proposed metric focuses on the inherent depth of the II-picture. Neither a global nor a viewing-angle-dependent metric can explicitly reveal how distortion is distributed within the 3D image. This lack of explainability of the two previously presented approaches is exemplified by applying Q_{global} and $\mathbf{Q}_{angle}$ to an II-picture coded using two different coding schemes. Figure 4.5 shows three VIs produced from the two coded II-pictures, corresponding to what is perceived by a user viewing these coded 3D images on an II-display. The images have been positioned for cross-eyed free viewing, i.e., the VI seen by the right eye is positioned to the left and vice versa as described in Section 2.1.2.2 on page 13. Note that although the coded II-pictures have the same Q_{global} (28 dB), they show very different depth-distributions of coding-induced distortion. Furthermore, Figure 4.5 (a) shows less distortion for nearby objects whereas Figure 4.5 (b) presents a more uniform distribution with respect to the depth of the 3D image. It is also not possible for these properties to be explicitly revealed by an angle-dependent metric; although depth is a property that may be inferred from motion parallax. This can be shown by applying $\mathbf{Q}_{angle}$ to the same II-pictures and presenting the resulting graphs as in Figure 4.6. Only the horizontal component of the angle-dependent quality metric is shown to simplify the figures. That is, only the middlemost row of the 2D quality metric is presented ($\mathbf{Q}_{angle}\left(u, \frac{V}{2}\right)$). Note that the variations in $\mathbf{Q}_{angle}$ over the viewing range are significant as compared to the Q_{global}, which is represented by the dotted lines. Hence, there is a potential of greater understanding when using a viewing-angle-dependent quality metric. In addition, by comparing the sub-figures a lack of tendency over the evaluated viewing angles is revealed, which again is due to the different properties of the two coding schemes. It still remains impossible to discern the apparent difference in depth-distribution of coding artifacts in the images in Figure 4.6

Thus, the above quality metrics evaluate the quality of an II-picture from a global, angle and viewpoint aspect. The important 3D property *depth* is, however, not addressed. Hence, coding schemes that distribute equal amounts of distortion, only differing in depth distribution, may not be easily distinguished by using Q_{global}, $\mathbf{Q}_{angle}$ or $\mathbf{Q}_{view}$. Therefore, the second quality metric that is proposed aims to explicitly reveal the distortion's distribution with respect to the depth of the II-picture.

(a) 0.08 bpp



(b) 0.14 bpp

(c) 0.35 bpp

Figure 4.5: II-picture Apples coded to Q_{global} (28 dB) using (a) EI-based PVS, (b) a SI-based PVS, and (c) a JPEG2000. The bitrate r required for each coding scheme is presented beneath each subfigure.

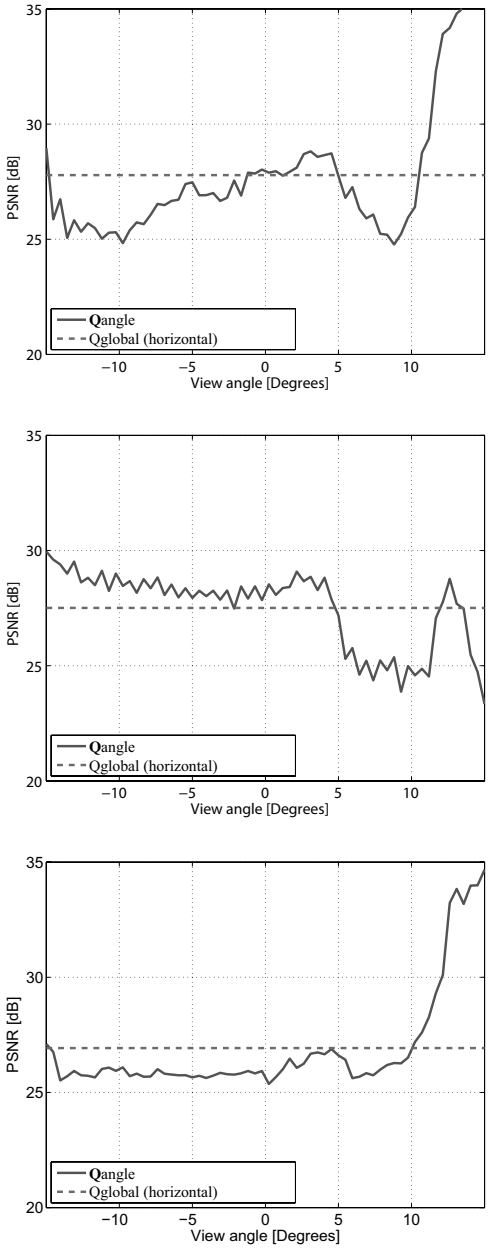


Figure 4.6: $\mathbf{Q}_{angle}$ evaluated on II-picture Apples coded using (a) EI-based PVS, (b) EI-based PVS, and (c) JPEG2000 coded II-picture.

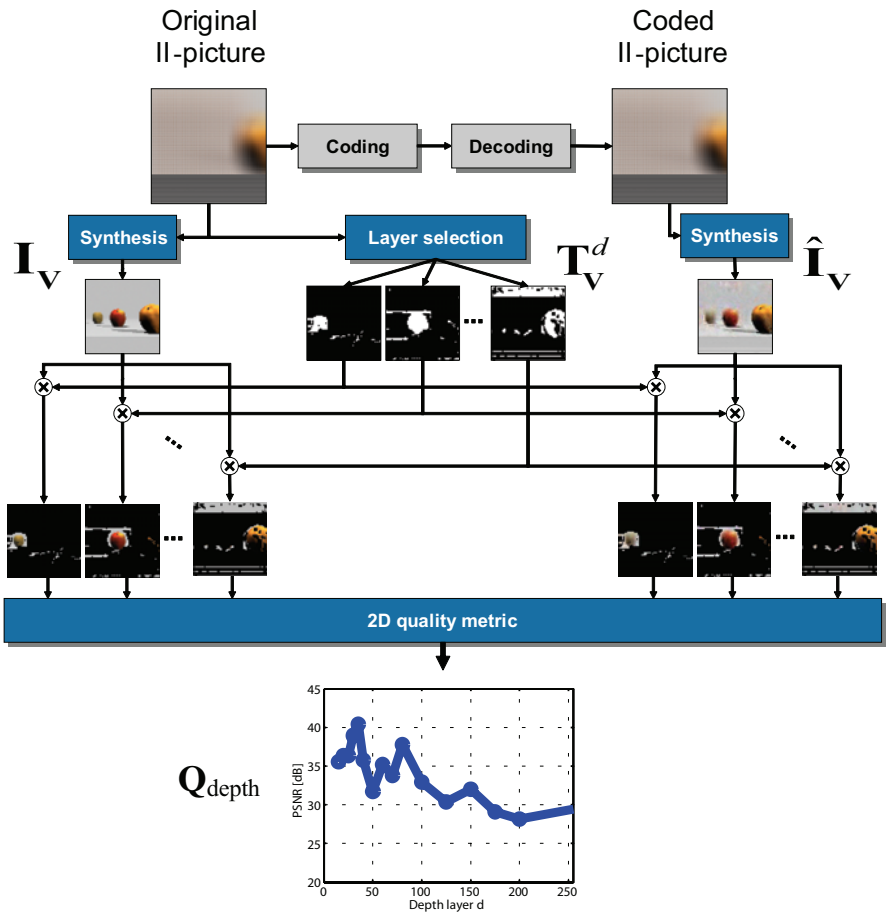


Figure 4.7: The operations constituting the sparse pseudo-depth dependent quality metric.

The metric is composed of three sequential steps:

- a VI-pair is synthesized originating from the original and the distorted II-picture respectively;
- pixels within the VIs, which belongs to objects at a specific depth are identified and
- a 2D quality metric is applied to all pixel subsets sharing the same depth layer.

These steps result in a quality metric taking the form of a 1D vector, with elements representing the distortion at each evaluated depth layer. Figure 4.7 illustrates the operations constituting the metric. The operations are:

1. Synthesis using image based rendering

2. Depth layer identification using depth estimation

3. Quality assessment using a 2D quality metric

The modularity of the metric allows it to be parameterized in many ways. The following description of the metric is based on a single parametrization but alternative methods for performing each operation will be discussed when appropriate.

4.3.2.1 Synthesis

The location of the virtual camera $\mathbf{V}$, from where the views are rendered, could be placed arbitrarily. However, two favorable properties are achieved by constraining the position to

$$\mathbf{V} = \left[0, 0, \frac{-f}{\delta^L}\right], \tag{4.4}$$

where f and δ^L are the focal length and pitch of each lens in the II-camera lens array respectively. Firstly, any angle-dependency is eliminated from the metric; it is primarily a depth-dependent metric. Secondly, the specific distance to the image plane of the II-camera ($\frac{-f}{\delta^L}$) ensures that the virtual camera's field-of-view β equals the II-camera's viewing angle α. This results in an efficient use of the 3D images data as all EIs contribute to the synthesized views. Figure 4.8 gives a geometrical overview of the model used to synthesize each VI.

When synthesizing the image pair, it is vital not to destroy any coding artifacts present as they will be used as inputs for the 2D quality metric. All interpolation must thus be avoided as it would have a low-pass filtering affect on the distortion, i.e. it would smear away any coding artifacts and consequently influence the measurements. Therefore, only one EI-pixel contributes to the color when calculating the pixel color values for the image pairs $\mathbf{I}$ and $\hat{\mathbf{I}}$, originating from the original and compressed II-picture respectively. Furthermore, nearest neighbor interpolation is used when evaluating Equation (4.1) for the same reason. This implies that in Figure 4.8, the color of a image pixel marked red is only taken to be the color of the pixel in the EI beneath lenslet $\mathbf{L}^L(1, 0)$.

4.3.2.2 Depth layer identification

Only a subset of the VI-pixels will represent projections of objects located at a specific depth layer. Hence, identification must be made as to which pixels correspond to which depth layers prior to applying the 2D quality metric. Different methods exist to estimate depth within a 3D image. We derive a depth map from the uncoded original II-picture using the depth-from-focus technique focus measure [96] due to it being a favorable combination of low complexity and high quality. The following discussion assumes that the virtual camera used to construct the depth map is located at position $\mathbf{V}$.

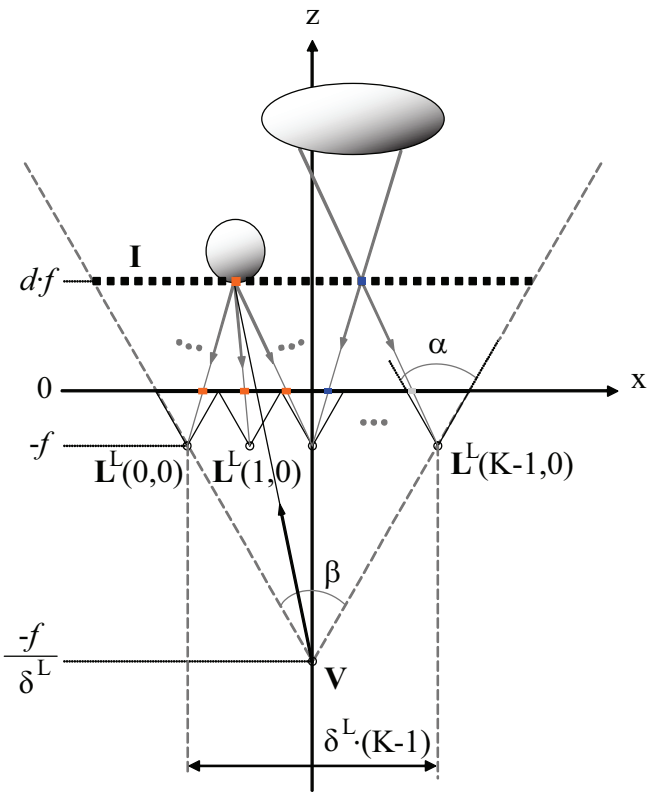


Figure 4.8: Geometry of the virtual camera placement relative to the II-camera's lenslets. Only the xz-plane with the first row of lenslets ($\mathbf{L}^L(k,0)$) is shown.

A set of EIs is used when synthesizing an image pixel, instead of only one as was the case for synthesizing the VI-pair. A large set of EIs corresponds to a large aperture of the virtual camera which thereby gives a shallow depth of field. The focus measure utilizes different patterns of EIs in order to synthesize an image at a specific focus plane or depth layer. Image pixels that correspond to a diffuse reflecting object located *at* the depth layer (e.g. the red pixel in Figure 4.8) will have similar pixel color values from contributing EIs. Other image pixels (e.g. the blue pixel in Figure 4.8) will be an average of objects *outside* the depth layer and the colors will therefore vary. By combining a base image $\mathbf{B}^d$ with a reference image $\mathbf{R}^d$ – each with pixels synthesized using different relative sets of EIs – allows for the calculating of a measure of how likely it is for an image pixel to belong to depth layer d. A pixel-resolution depth map is constructed by:

$$\mathbf{D} = \arg\min_d \left(\left| \mathbf{B}^d - \mathbf{R}^d \right| * h \right), \tag{4.5}$$

where h is a filter kernel that is used in a convolution step (operator $*$) to enhance the result. The interested reader is referred to [96] for more details.

Contrary to the rendering stage, this layer selection process benefits from including a *larger* set of EIs when rendering the base and reference images. Averaging over a larger set increases the reliability of the focus measure as the depth of field is reduced, which consequently enhances the accuracy of the depth map. The number of EIs contributing to each image pixel is in this work increased from 2 to 25 compared to the original focus measure definition. Increasing the set of EIs has the cost of increasing the synthesis time and hence also the time to calculate the proposed quality metric, which is the reason for not including all EIs in the calculation. Figure 4.9 shows the EI-patterns used to select which neighboring EIs that contribute to the pixel color values of the base and reference image. Based on the depth map $\mathbf{D}$, a mask-image $\mathbf{T}^d$ is derived according to

$$\mathbf{T}^d_{\mathbf{V}} = \begin{cases} 1 & \text{if} \quad \mathbf{D}_{\mathbf{V}}(m,n) = d \\ 0 & \text{otherwise} \end{cases}, \tag{4.6}$$

which is used to extract pixels and produce the set of masked depth layer images

$$\mathbf{X} = \mathbf{I}_{\mathbf{V}} \cdot \mathbf{T}^d_{\mathbf{V}} \tag{4.7}$$

and

$$\widehat{\mathbf{X}} = \hat{\mathbf{I}}_{\mathbf{V}} \cdot \mathbf{T}^d_{\mathbf{V}}, \tag{4.8}$$

which correspond to the original and coded II-picture respectively. The multiplication in Equation (4.7) and Equation (4.8) is performed pixel-by-pixel.

4.3.2.3 Quality assessment

The set of masked depth layer images $\mathbf{X}$ and $\widehat{\mathbf{X}}$ are (contrary to the VIs) in a form that will not be viewed per se; the viewer will see the combined set of all depth layer images when watching the II-display. Therefore, applying a 2D quality metric consistent with a subjective test of 2D image quality is unnecessary. PSNR is instead

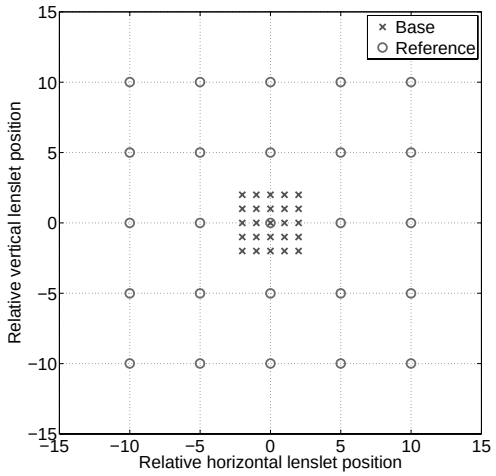


Figure 4.9: Relative lenslet positions that are used to derive a specific image pixel for the base- and reference virtual image respectively.

used and Equation (2.15) is applied to the two masked depth layer images defined in Equation (4.7) and Equation (4.8). The average operation performed when calculating Equation (2.16) presumes that all pixels in the two images contain valid color values. However, this is not true for the masked depth layer images. Only $K^d \cdot L^d = \sum_{\forall k,l} \mathbf{T}^d(k,l)$ of the pixels belong to depth layer d. Normalizing using $K \cdot L$ gives equal weight to each layer, regardless of how many pixels that correspond to the depth layer. That is, a layer with a small number of severely distorted pixels may give rise to a PSNR-value equal to a layer with a large number of pixels showing minor coding artifacts. Normalizing using $K_d \cdot L_d$ instead gives equal importance to each depth layer, regardless of size. The latter normalization results in an increased ability to discern the distortion and is therefore used.

4.3.2.4 Combined operations

The proposed quality metric $\mathbf{Q}_{depth}$ is the combination of the described operations, which I here define as

$$\begin{aligned} \mathbf{Q}_{depth} &= [Q_{depth}(d)]_{\forall d} \\ &= \left[PSNR\left(\mathbf{I} \cdot \mathbf{T}^d, \hat{\mathbf{I}} \cdot \mathbf{T}^d\right)\right]_{\forall d}. \end{aligned} \tag{4.9}$$

The selection with regards to which depth layer is to be evaluated is outside the scope of the metric design. A linear or a logarithmic distribution of d can be adopted if no a priori knowledge is available about the 3D image's depth content. An approximate depth distribution gained from a pre-processing depth estimation step is another option.

With the full metric defined it time to provide an explanation with regards to the prefix *pseudo* that is a part of the proposed metric's name. The metric produces a vector explicitly defining the II-picture's quality as a function of its depth seen from a specifically defined viewpoint according to Equation (4.4). However, viewpoints differing only in lateral position (the xy-coordinates of Equation (4.4)) might give rise to different amounts of distortion at the same depth layer. This can not be solely explained by the fact that the respective depth layers are also translated relative to each other. Instead, it is a consequence of basing the metric on a subset of II-picture pixels. For example, from one viewpoint an object might be more distorted than from another even though its distance from the two originating viewpoints is the same. This might occur when distortion is induced by a coding scheme that does not explicitly operate based on the depth of a II-picture. Therefore, the prefix pseudo is used to indicate that the distortion measured does not correspond to an absolute depth within the 3D image, albeit it being correct with respect to the depth perceived from the viewpoint defined in Equation (4.4).

4.4 Results

The qualitative evaluation of the proposed metrics aims to describe their advantages in a more explicit manner. The two quality metrics will be evaluated empirically in the next chapter where they will be used to indicate coding-induced distortion properties, which is not possible to discern using present metrics.

4.4.1 Sparse angle-dependent MSSIM

Using $\overline{\mathbf{Q}}_{view}$ significantly differs from using Q_{global} even though both calculations results in a single scalar value. This is because $\overline{\mathbf{Q}}_{view}$ is based on

- the use of VIs, which models the spatial demultiplexing performed by the II-displays lens array and
- the use of MSSIM that warrants a stronger agreement with subjective testing results.

Q_{global} on the other hand is only based on the difference between pixels in the II-picture and does not in any way consider the process of viewing an II-display.

Furthermore, $\mathbf{Q}_{view}$ with its angle-dependency might also at a quick glance seem similar to a $\mathbf{Q}_{angle}$ evaluated at a subset of the $U \times V$. However, the difference is significant and is mainly based on three properties of $\mathbf{Q}_{view}$:

- it presumes a physically founded location of the viewer,
- the evaluated VIs are perspective projections close to what is perceived on a II-display, and

- the evaluation method (MSSIM) have a higher correlation with subjective tests than PSNR.

Neither of these properties can be attributed to $\mathbf{Q}_{angle}$. Note though that the quality assessment provided by Equation 4.3 passes verdict on a subset of all pixels from the II-picture. Q_{global} and $\mathbf{Q}_{angle}$ include all II-picture pixels in their calculations.

A quantitative analysis of $\overline{\mathbf{Q}}_{view}$ is implicitly performed in Section 5.8.5 of the subsequent chapter on coding II-pictures, where it is utilized to estimate how coding-induced distortion is perceived.

4.4.2 Sparse pseudo-depth-dependent PSNR

With its different approach, $\mathbf{Q}_{depth}$ is not directly comparable to Q_{global}, $\mathbf{Q}_{angle}$ or $\mathbf{Q}_{view}$. Being able to discern how distortion is distributed with respect to depth is a property that is not present in any of these metrics. Hence, directly comparing them with $\mathbf{Q}_{depth}$ is not appropriate. An evaluation of the constituting operations is instead performed, studying how different parameterizations affect the end result.

The modular construction of the metric allows for different parameterizations than those presented. For example, the metric may easily be extended to more densely sample the quality within the viewing space by adding more viewpoints. However, it may not be feasible to extend the metric such that it would be comparable to Q_{global} and $\mathbf{Q}_{angle}$, which exhaustively incorporates all pixels in the II-picture. The computational requirements would then prohibit the use of the metric in other than extreme measurement scenarios. Even more so if the resulting 1D quality metric was condensed into a few scalars for each measurement instance, e.g., mean, minimum, maximum, standard deviation etc..

The serial structure of the metric, with output from one operator being the input of another, renders the quality of the end result no better than its constituent parts. A simple sensitivity analysis shows that in order for the metric to provide reliable results about the depth distribution of distortion, the depth estimation operation must perform with equal reliability. Estimating depth from a set of two images or more is an active research field that as yet has no final and absolute solution. Highly complex off-line solutions compete with resource efficient real-time algorithms. The depth-from-focus techniques (which the focus measure adheres to) constitute one approach, but a number of others exist for estimating depth from two or more images [97]. However, not much has been discussed about depth estimation within the II-community, which is partly explained by the course depth maps that are produced if the II-picture is merely considered to be a large set of low resolution images. The works on depth estimation that have been presented in the literature have for this particular reason all been based on other CIs rather than EI [63, 98, 64]. The interested reader is referred to the Middlebury Stereo Vision Page, which holds an updated top-list containing the present state-of-the-art algorithms for stereo-based depth estimation [99]. Future work within II-based depth estimation might very well contain elements from these algorithms but adopted for SIs for example. Any

such technique can be directly incorporated into the proposed sparse pseudo-depth dependent quality metric.

The empirical results from applying $\mathbf{Q}_{depth}$ to coded II-pictures are presented in the coding chapter's Section 5.8.5.3.

4.5 Concluding remarks

An obvious property of quality metrics designed for 2D images is that they fail to capture all aspects of a 3D images format such as II. This means that there is the necessity for objective quality metrics to be explicitly designed to quantify distortion present in II-pictures; in addition, quantifying it in ways with specific relevance to important II-picture properties. This chapter presented two quality metrics, which I constructed explicitly for II-pictures and their specific properties.

The first metric models how distortion is perceived by a viewer watching an II-display. For this a set of View Image is synthesized, which simulates how the optics of the II-display demultiplexes the 3D images stored in the II-picture. A quality metric with low dimensionality is achieved while still sampling the quality within the viewing space, by sparsely selecting from which viewpoint the VIs are synthesized. The state-of-the-art quality metric for 2D images MSSIM is applied to the VIs in order for the proposed metric to have potentially high correlation with subjective test results.

The second metric aims to discern the depth distribution of coding induced distortion, acting as a supplementary tool when evaluating distortion in addition global and angle-dependent metrics. New aspects of coding artifacts in 3D images may be revealed using the metric, both in the original form of the metric as well as after post-processing of the result producing moments such as mean and standard deviation or extrema such as min and max. Furthermore, a rate-distortion image may easily be produced by displaying $\mathbf{Q}_{depth}$ as a function of both depth and rate. Each pixel in the rate-distortion image then corresponds to the quality at a given combination of depth and rate. Such a 2D function gives a broader understanding of the effect of coding and how it affects the perceived depth within a II-picture.

Despite the favorable properties of objective quality metrics, there is as yet no algorithmic method that can capture all aspects of the processes taking place in the HVS. As a consequence, an exact knowledge about how the quality will be perceived is unknown using only objective metrics. The only way to achieve this knowledge is through subjective tests, which allows a set of observers to view and grade the test material produced by the system under study. However, the logistic requirement required to set up such tests becomes complicated as a consequence of the human participation. In addition external parameters such as ambient room lighting, viewing distance, display luminance etc. must be carefully defined and exactly conformed to. Unfortunately, there is no such thing as an ideal test setup. Thus, the final results from subjective tests are to some degree affected and tainted by the test setup itself, making comparability of the achieved test results less simple than when

using objective quality metrics. For this reason, and due to the lack of a proper II-based 3D display being available at the time of performing these tests, no subjective tests where performed as a part of this work. Instead the focus of my research was placed on extending the range of *objective* evaluation metrics to also include tools specifically designed for measuring distortion with regards to II-picture properties.

4.5.1 Authors contributions

My main contributions with regards to the topic of this chapter are the construction of:

- An angle-dependent quality metric that aims to model how a II-picture is perceived by a viewer watching a II-display.
- A depth-dependent quality metric, which gives a view with regards to how distortion is distributed within a II-pictures with respect to its depth.

Empirical studies of coding-induced distortion using the proposed metrics are conducted in Chapter 5 but can also be considered as contributions to this chapter. The novel quality metrics that this chapter has described, revealed distortion characteristics in Chapter 5, which would have been impossible to show otherwise.

This chapter's content has been published in parts in Papers IV – VI and VIII.

4.5.2 Problem definition – P2b

What consequences will a proposed coding method have on objective quality? During the work on coding it has become evident that the traditional global objective metric are useful for comparing overall performance but fails to provide a detailed insight into the characteristics of II-picture distortion. The II-picture contains additional properties such as view-angle dependence and depth, which is not present in 2D images. Transferring objective quality metrics from the field of 2D imaging into the realm of II-pictures does not imply that II-picture properties are explicitly evaluated. As a result this chapter has presented two metrics that model how distortion manifests itself in properties specific to II-pictures. Hence, developing coding methods for II-pictures affects objective quality to such an extent that new metrics to measure objective quality are required.

Chapter 5

Coding

The high pixel resolution requirement for II-based 3D images compared to 2D images results in an increased raw data rate requirement. To reduce this data rate requirement gap relative to 2D images, some type of compressive coding must be used. Otherwise the 3D image's demand for storage space or transmission rate will limit the rate at which this more lifelike presentation format is adopted. Lossless coding – where the original image can be reconstructed without distortion - is often a requirement if the decoded image is to be further processed or analyzed. Medical and forensic applications are typical examples where any distortion introduced by the coding scheme would be highly undesirable since it would affect the analysis and the conclusions drawn. However, in applications where the sole purpose of the decoded image is for it to be looked at, the properties of the HVS could be favorably utilized. A specifically important HVS property in this context is the tolerance for different types of distortion in the color, spatial and spatial frequency domains. A lossy coding scheme can achieve significantly higher coding efficiency than a lossless approach as it allows for a certain amount of distortion to be introduced during coding. Various signal processing approaches such as prediction and transformation are used to uncover the parts of the image that the HVS is less sensitive to. Variably quantizing these parts of the image allows for a gradual trade-off between rate and distortion. As a result of the standardization of 2D images coding, in which the two standards Joint Photographic Experts Group (JPEG) and JPEG2000 Part 1 (JPEG2000) are the most prominent examples, many applications have adopted digital imagery in different forms. Both consumer digital cameras and images on the World-Wide-Web are a direct consequence of lossy 2D images coding. For 3D images the arguments for coding are even stronger. The work presented in this dissertation focuses on 3D images that are viewed directly on a II-display, i.e. no subsequent analysis will be performed after the II-picture has been decoded. This allows for the more coding efficient lossy approach to be used and therefore, when the term coding is used in subsequent discussions, lossy coding is implied.

5.1 Chapter outline

This II-picture coding chapter is structured as follows. In the next section, the methodology used to assess the conducted work is presented. Section 5.3 gives a brief presentation about why applying 2D coding schemes on II-pictures is an inefficient coding approach. The subsequent Section 5.4 and Section 5.5 then presents a coding scheme for time static II-pictures, which utilizes the efficiency of standards for 2D video and volumetric image compression. How the coding standards are parameterized to fit the presented coding scheme is discussed in Section 5.6. The setup used in when performing evaluation experiments is presented in Section 5.7 and the result from the experimental studies are summarized in Section 5.8. The coding chapter is finally concluded in Section 5.9, where also the author's contributions are explicitly summarized.

5.2 Methodology

This chapter will present a coding scheme that utilized 2D video coding tools to compress 3D images in the form of II-pictures. Different forms of the scheme will be produced as a consequence of parameterizing the basic scheme. The scheme will be studied theoretically with respect to this parametrization. To be able to test the coding efficiency a set of reference scenes, with adherent II-pictures, will be defined. Using these II-pictures, the coding efficiency of the scheme will be objectively tested by evaluating:

- Coding efficiency
- Coding quality
- Coding cost in terms of CPU-time

The proposed II-picture coding scheme will be compared against other coding schemes presented in the literature. Finally, a qualitative evaluation will be performed based on simulated visualization that investigates the nature and extent of the coding artifacts produced by the different coding schemes. Details concerning coding artifacts are presented in Section 5.8.5 where the sparse angle-dependent quality metric $\mathbf{Q}_{view}$ from Section 4.3 is used. In addition, the depth distribution of the coding-induced distortion will be studied using $\mathbf{Q}_{depth}$ from Section 4.3 and the results are presented in Section 5.8.5.3

5.3 II-picture characteristics

The first step in constructing a coding scheme is to characterize the signal to be coded. At first glance the II-based 3D images bears a close resemblance to the images which are captured by an ordinary 2D camera. This is particularly the case

when a subset of the II-picture is enlarged such that the individual EIs are revealed. However, after a more careful examination there are, in the main, two properties of the II-picture that prohibits basing the coding scheme design directly on 2D images coding arguments:

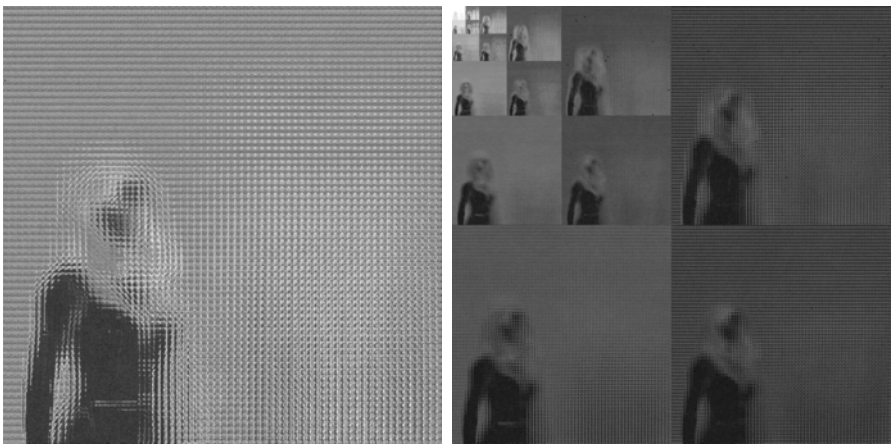
1. The increased spread of the spatial redundancy in the II-picture which is imposed by the periodic nature of the lens array.

2. The II-picture will be decoded by a lens array prior to viewing.

High correlation between neighboring pixels is the main advantageous 2D images property used when designing 2D images coding schemes. This characteristic is exploited by either predicting a specific pixel-value from nearby pixels or by transforming a block of pixels and giving priority to the transform coefficients with high energy. Unfortunately, this property does not transfer in an unchanged manner to the context of II-pictures. On the one hand, spatial redundancy is even stronger in II-pictures where it spans an even larger number of pixels than for the 2D images case. This is the result of the similarities between neighboring EIs. The majority of the depicted scene is not captured in a single EI, but in several. This results in a spatial redundancy between neighboring EI pixels but also between EIs, which increases the spatial redundancy spread. An object that is further away from the II-camera is captured by a larger number of EIs and introduces a larger spatial redundancy spread. However, the pixel-correlation between EIs is not smooth as it is within each EI. Instead it is broken up due to the periodic II-picture structure, which consequently introduces a repetitive characteristic in the II-based 3D image.

Thus, a portion of the bitrate required for coding an II-picture using 2D images coding schemes will be required to retain this periodic pattern induced by the II-picture structure. Figure 5.1 shows the coefficients that results from applying a 8x8 blockwise DCT and a Cohen-Daubechies-Feauveau (CDF) 9/7 wavelet transform on the transforming the II-picture Twins. These two approaches are the main building blocks used to uncover spatial redundancy in JPEG and JPEG2000 respectively. The blockwise DCT, with its lack of handling redundancy between blocks, fails to reduce the strong pixel-correlation between EIs. In Figure 5.1 (a) this is manifested as high energy in a significant portion of the DCT-blocks. The recursive subdivision of the wavelet approach is particularly adapted to address widespread similarities within the image. However, when applied to II-pictures a significant part of the high frequency coefficients receive energy from the periodic II-picture-structure and not the captured content. This is shown in Figure 5.1 (b) as high energy in the right part of the image, which corresponds to high frequencies that should contain relatively small amounts of energy. Hence, in a transform-based 2D images coding scheme the II-picture structure appears as a set of coefficients containing high energy based solely on the II-picture structure. Apart from being necessary for decoding a valid II-picture, the bitrate portion required for this set of coefficients does not contribute to the quality of the actual 3D content.

The second II-picture property that prevents the use of 2D images coding scheme relates to the HVS shortcomings, which proved themselves able to provide useful



(a) (b)

Figure 5.1: Transform coefficients as a result of (a) 8x8 block-wise DCT (JPEG) and (b) 5-stage decomposition using CDF 9/7 wavelet transform (JPEG2000).

information when developing 2D images coding schemes. However, in the context of II-picture coding it is important to bear in mind that the II-picture – even though it is stored in its original form as 2D data – will be decoded by a lens array before viewing. Thus, the HVS information is applicable to the pixels forming the VIs but not to the pixels in the II-picture itself. Coding schemes for 2D images assume that the order in which the pixels are stored in the 2D data is also the order in which they will be viewed. To alleviate this discrepancy two options are avaiable: either the HVS knowledge is transformed such that it is applicable to the pixels in the II-picture; or the pixels in the II-picture are transformed into a form that is more suited for the HVS knowledge. The following sections will describe a coding method that adopts the latter approach.

5.4 The proposed coding scheme - an overview

The coding scheme presented in this chapter is built using state-of-the-art coding standards originally designed for video and volumetric images, more specifically H.264 / AVC and JPEG2000 Part 10 (JP3D) [100, 101]. This enables the vast knowledge in coding on which these standards have been built to be utilized. Coding an II-picture with coding standards adopted for other signal types requires a pre-processing step. The coding scheme first transforms the II-picture into a form that resembles a video sequence or a volumetric image, which is then encoded using a H.264 / AVC- or a JP3D-encoder. The scheme is generically composed of three operations:

1. transform the II-picture into a CI set,

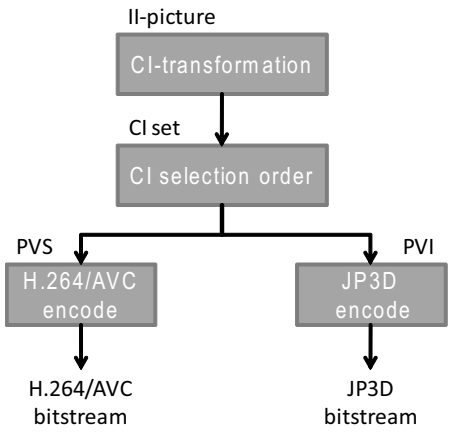


Figure 5.2: The constituting operations of the proposed coding scheme.

2. select from the set of CIs such that a Pseudo Video Sequence or Pseudo Volumetric Image is formed and
3. encode the Pseudo Video Sequence or Pseudo Volumetric Image using the corresponding coding standard.

Figure 5.2 shows the the coding scheme as a block-diagram.

In the following, this generic description of the scheme will be formalized by providing greater detail with reference to the choice of the constituent parts.

5.5 Pseudo Video Sequence (PVS) and Pseudo Volumetric Image (PVI)

The first step in the proposed coding scheme is forming a Pseudo Video Sequence (PVS) according to

$$\begin{aligned} \mathbf{PVS} &= [PVS_j(s,t)]_{j=0,1,\ldots,J-1} \\ &= \left[CI_{\Gamma(j)}(s,t)\right]_{j=0,1,\ldots,J-1}, \end{aligned} \tag{5.1}$$

where PVS_j refers to the j-th picture in the PVS (PVS-frame). J is the total number of PVS-frames constructed by exhaustively selecting all CIs ($J = \Xi \cdot \Psi$). The permutation function $\Gamma()$ controls the order in which a specific CI index (ξ, ψ) is selected to form the PVS-frame j according to

$$\Gamma : j \rightarrow \{\xi, \psi\}. \tag{5.2}$$

The II-picture is transformed into a Pseudo Volumetric Image (PVI) if each CI is designated to be a slice in a volumetric stack instead of a frame in a PVS. The difference

is conceptual and merely indicates that a volumetric image encoder is to be used on the transformed II-picture. Hence, Equation (5.1) is also applicable to define the PVI after changing the abbreviation PVS into PVI. We use the prefix *pseudo* in PVS and PVI to indicates that these forms of the II-picture are not a video sequences or volumetric image per se, albeit the characteristics are similar for certain CI-types. The PVS is a function of a spatial variable instead of time as is the case for a video sequence. For the PVI index j does not correspond to the z-axis of a 3D volume as is the case for volumetric data constructed using in medical examinations such as CT, MRI or ultrasound.

5.5.1 Choosing type of Component Image

The most evident way of constructing the PVS is by setting $CI = EI$, i.e. defining the EIs to be the PVS-frames [45, 74, 75]. The resulting PVS is of length $J = K \cdot L$ where each PVS-frame has a resolution of $S \times T = U \times V$ pixels. The EI-based PVS closely resembles a video sequence, albeit with video frames of relatively low resolution.

Constructing the PVS using RIs results in a PVS with $J = V \cdot L$ PVS-frames that each have a resolution of $S \times T = U \times K$ pixels [76]. The RI-based PVS-frames bears little resemblance to the pictures of a 2D video sequence, as was shown in Figure 2.25 (c) and (f) on page 42. This is a disadvantage as an efficient 2D video encoder would utilize the shortcomings of the HVS to enhance the coding efficiency. Thus, the more the encoder input deviates from the purpose of the HVS-based encoding algorithms, the less likely it is that the encoder will perform optimally.

Constructing the PVS out of the SI set was first proposed by Olsson et al. [77]. This choice of PVS-frames brings favorable properties compared to using EI or RI. Greater coding efficiency is achieved for II-picture structures with a multitude of EIs where each EI has a lower spatial resolution, i.e. when $K \cdot L >> U \cdot V$. An additional advantage is that the coding induced distortion distributes more evenly within the depth of the coded II-picture. These statements are dealt with in future sections of this chapter as this particular PVS is studied in more detail. Hence, a SI- based PVS will have $J = U \cdot V$ PVS-frames, each with a resolution of $S \times T = K \times L$ pixels.

Given the orthographic characteristic of the SI it is worth noting how this implies two properties of the SI-based PVS:

1. The projections of distant objects will translate a larger distance between consecutive PVS-frames than nearby objects.

2. An object will have similar spatial frequency content within its projection regardless of the object's distance to the II-camera.

These two properties will have a specific effect on the coded image's quality.

On the one hand, the first property suggests that distant objects will be more difficult to code by the following 2D video coding tool. For example, the motion compensation step will more often fail to find matches for distant objects as their

projection falls outside the given motion vector search area, contrary to nearby objects whose projections translate shorter distances. This will in turn result in a less efficient intra-coding of the portion of the PVS-frame that contains these more distant objects. On the other hand, the second property suggests that the intra-coding of these distant objects is the main cause of distortion, compared to EI where the perspective projection also low-pass filters distant objects due to the shape of the lenslets' frustum. That is, the orthographic property of the SI will transform an object's textured face into the projection's textured surface, approximately without influencing the spatial frequency of the texture. The EI's linear perspective projection averages the texture of a distant object into a smaller number of pixels than for a nearby object. This reduces the higher spatial frequencies of a distant object's texture projection in comparison to that of a nearby object with an identical texture. Section 5.8.5 further discusses how these properties manifest into coding artifacts that appear when viewing the coded II-picture using an II-display.

An informative overview of the PVS can be achieved by studying the CIset's 2D images form shown in 2.25, since the PVS is constructed by exhaustively selecting all CIs. Naturally, all aspects of the three types of PVS cannot be covered by a single example. Nevertheless, the example Twins is a good example on which to base a further discussion. Two conclusions can be drawn by examining the three PVS types. Firstly, the different transforms results in data sets that are not equally homogenous on the large scale. The EI data set has a much more straight relationship between 2D projection content and scene objects and their distance to the camera. The orthographic property of the SI data set instead manifests itself in a much more homogenous 2D projections since object projection size is independent of object distance. The second conclusion is that the permutation function is an important factor to consider when forming a PVS, regardless of transform.

5.5.2 Component Image Selection Order (CISO)

Choosing the order in which the CIs are selected to form the PVS will affect how efficiently the II-picture can be compressed using H.264/AVC. This Component Image Selection Order (CISO) is a one-to-one mapping, where the permutation function $\Gamma(j)$ selects a CI (ξ, ψ) within the CI set to become the j-th PVS-frame. Hence, different CISOs are implemented using different permutation functions $\Gamma(j)$. This is shown in Figure 5.3 where two different $\Gamma(j)$ produce two different PVSs from the same CI set; dashed arrows correspond to one CISO whereas dotted arrows correspond to another.

Each permutation function will result in a PVS with slightly different coding characteristics. Hence, a fundamental parameter in the design of a PVS coding scheme is how to choose the permutation function or the CISO regardless of which CI is used to form the PVS-frames. Five different selection orders are studied in this work: row, column, parallel, zig-zag and spiral [90]. In Figure 5.4, four of these are illustrated for a CI set with $\Xi \cdot \Psi = 8 \cdot 8 = 256$; column being excluded since it is merely the row CISO rotated 90°.

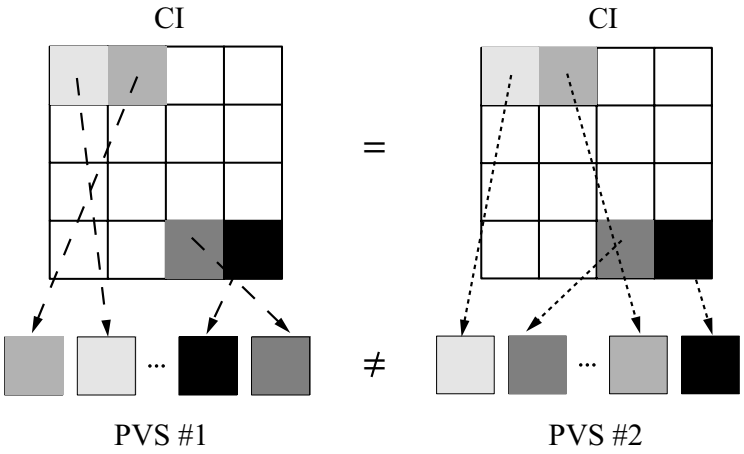


Figure 5.3: The CISO process used to construct a PVS.

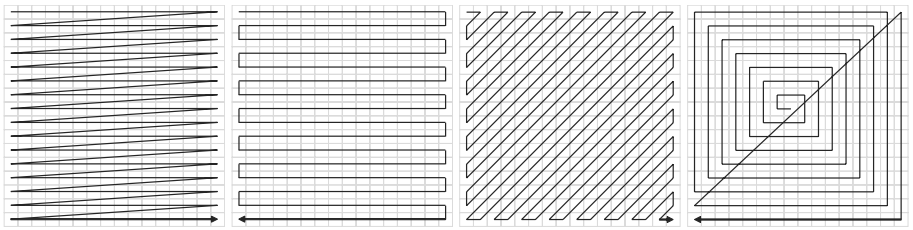


Figure 5.4: Selection pattern of CISO (a) row, (b) parallel, (c) zig-zag and (d) spiral.

It is possible to assess each CISO's potential in producing a more easily compressible input for the subsequent encoder using:

- the cross-correlation coefficient c and
- the difference residual energy e

of the reordered CI set. PVSs with high correlation between consecutive PVS-frames (corresponding to CIs) are more likely to produce small residuals as a result of the future motion compensation. Thus, the larger number of PVS-frames resulting in high values of c, the more efficient the subsequent encoder should perform. In this assessment, the cross-correlation coefficient $c(j)$ for two consecutive PVS-frames PVS_j and PVS_{j-1} is defined for $j = 1, 2, \ldots, J-1$ as

$$c_j = \frac{\sum_s \sum_t \left|PVS_j - \overline{PVS_j}\right| \left|PVS_{j-1} - \overline{PVS_{j-1}}\right|}{\sqrt{\sum_s \sum_t \left|PVS_j - \overline{PVS_j}\right|^2} \sqrt{\sum_s \sum_t \left|PVS_{j-1} - \overline{PVS_{j-1}}\right|^2}}, \tag{5.3}$$

where $\overline{X}$ denotes the arithmetic mean value of all components in matrix X. The arithmetic mean $\overline{c}$ and standard deviation σ_c are then calculated for the complete PVS according to

$$\overline{c} = \frac{1}{J-1} \sum_{j=1}^{J-1} c_j \tag{5.4}$$

and

$$\sigma_c = \sqrt{\frac{1}{J-2} \sum_{j=1}^{J-1} (c_j - \overline{c})^2}. \tag{5.5}$$

A high $\overline{c}$, accompanied by a low σ_c, would indicate that a high portion of the CIs are similar in content.

An additional assessment on how efficiently a motion compensation operation or wavelet transform decomposition would be able to reduce the inter-frame redundancy is defined as

$$e = \frac{1}{J-1} \sum_{j=1}^{J-1} e_j^2, \tag{5.6}$$

where

$$e_j = \|PVS_j - PVS_{j-1}\|_F \tag{5.7}$$

and $\|X\|_F$ denotes the Frobenius norm of matrix X. The difference residual energy in (5.6) provides a restrictive assessment, which for the case of PVS estimates the DPCM-residual *without* motion compensation. A low e indicates that the residual contains less energy for the transform to compact, which increases the possibility of an efficient compression.

In Section 5.8.2, $\overline{c}$, σ_c and e are calculated for the four reference II-picture 's PVSs, which are described in Section 5.7.1. Since both the cross-correlation and the residual energy only gives an assessment of the selection orders characteristics, and the

final coding quality provides the correct answer, objective evaluation of the coding efficiency is also presented in the experimental results of Section 5.8.2.

5.5.3 Bit rate penalties from coding structure

Associated with the resulting bitstreams of the proposed coding schemes there is also a bitrate penalty due to the header information required. The bitstream describing a coded II-picture is composed of two parts, regardless of the used coding scheme. Part of the bitstream syntax is directly related to the uncoded II-picture pixels whereas the other part is merely required to describe the used coding structure or the bitstream semantics. The header information present in e.g. a H.264 / AVC bitstream is vital for making the bitstream possible to decode. However, the header information will not directly benefit the quality of the 3D image. To achieve maximum coding efficiency, a minimum of bits should be spent on conveying bitstream semantics.

The header overhead required for a coded PVS is mainly due to the H.264 / AVC semantic constructs sequence, picture, slice, macroblock and block. Depending on how the PVS is formed, the bits set aside for headers can have more or less influence on the resulting 3D images quality. For CIs with a high resolution ($S \times T$), the header portion of the bitrate becomes negligible. However, the negative impact on the image quality is increased as the resolution is reduce. Thus, it is not feasible to construct an EI-based PVSfor II-picture structures where there is a large number of relatively low resolution EIs for example. For such II-picture structures, a SI-based PVS is more beneficial [77].

A formal definition of the PVS bitrate is required in order to study this intuitively derived PVS property. Firstly, I define the size B of the II-picture (in bits) as

$$B = M \cdot N \cdot r, \tag{5.8}$$

where r is the desired number of bits per pixel (bpp) for the coded II-picture. These bits are distributed over the J number of PVS-frames in such a way that on average, each PVS-frame $PVS(x, y, j)$ will be of size

$$\bar{b} = \frac{B}{J} = \frac{B}{\frac{M \cdot N}{S \cdot T}} = S \cdot T \cdot r. \tag{5.9}$$

Thus, to achieve a specific compression ratio for the II-picture, a bitrate R in bps should be set for the PVS according to

$$R = S \cdot T \cdot r \cdot f, \tag{5.10}$$

where f is the PVS's pseudo frame rate. Again, time is not an explicit variable in a PVS. Thus, the frame rate within the context of a PVS is not the inverse of time. Note that albeit the bitrate R in Equation (5.10) might imply that Constant bitrate (CBR) must be used to code the PVSThis is not the case. Different types of Variable bitrate (VBR) -schemes may be utilized as long as the resulting data size of the coded PVS complies with Equation (5.8).

The number of bits required for a coded PVS-frame (Equation 5.9) can be separated into two terms: header bits $\bar{b}_h$ and data bits $\bar{b}_d$. The latter term corresponds directly to the pixels of the II-picture and the quality of the resulting 3D image. Hence, studying the relation between $\bar{b}_h$ and $\bar{b}_d$ gives some idea concerning how much of the bitrate will explicitly contribute to the 3D images quality. For this I define the relation of headers as

$$\frac{J \cdot \overline{b}_h}{B} = \frac{\frac{M \cdot N}{S \cdot T} \cdot \overline{b}_h}{M \cdot N \cdot r} = \frac{1}{S \cdot T \cdot r} \cdot \overline{b}_h. \tag{5.11}$$

The header bits can be further decomposed into headers relating to the different semantic constructs according to

$$\overline{b}_h > \overline{b}_{h,slice} + \frac{S \cdot T}{\mathrm{MB}_{\mathrm{size}}} \cdot \overline{b}_{h,MB}. \tag{5.12}$$

Additional header terms corresponding to the semantic constructs sequence, picture, and block are excluded and are instead implicitly contained in the inequality. The second header term's factor conveys the number of macroblocks within each PVS-frame. For H.264/AVC, $\mathrm{MB}_{\mathrm{size}} = 16 \times 16$ pixels. In Figure 5.5, the relationship in Equation (5.11) is plotted for $S \times T = 1$ and $S \times T = M \times N$, which shows the limits within which any PVS-scheme must operate, regardless of the chosen CI and CISO. As expected, the PVS-frames should be as large as possible to reduce the impact on image quality due to header data, especially for low bitrates. Inserting Equation (5.12) in Equation (5.11) gives the header portion H that is defined here as

$$\begin{aligned} H\left(S \cdot T, r\right) &= \frac{1}{S \cdot T \cdot r} \cdot \overline{b}_h \\ &> \frac{1}{r} \cdot \left(\frac{\overline{b}_{h,slice}}{S \cdot T} + \frac{\overline{b}_{h,macroblock}}{\mathrm{MB}^2} \right). \end{aligned} \tag{5.13}$$

The header portion h enables a comparison of different PVSs with respect to the portion of the bitrate they require to convey header data. Hence, a more favorable coding efficiency is achieved by selecting as large an CI resolution as possible when constructing the PVS, as this gives a smaller header portion H.

5.5.4 Working range for the SI-based PVS

Evaluating Equation (5.13) for all CIs and comparing the results will indicate the II-picture structures for which the proposed SI-based PVS is favorable. Comparing SI with EI gives

$$\begin{aligned} H_{EI} &> H_{SI} \\ H\left(U \cdot V, r\right) &> H\left(K \cdot L, r\right) \\ \frac{1}{U \cdot V} &> \frac{1}{K \cdot L} \\ \frac{M \cdot N}{U \cdot V} &> U \cdot V \\ \sqrt{M \cdot N} &> U \cdot V. \end{aligned} \tag{5.14}$$

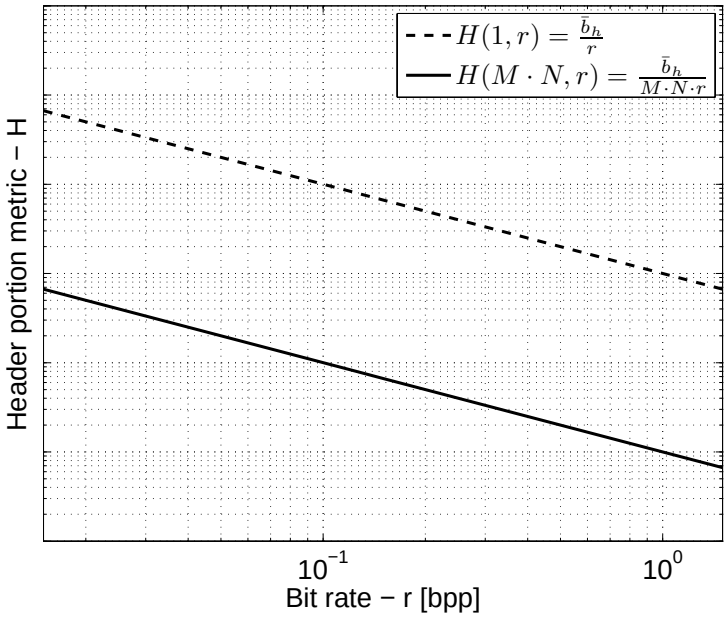


Figure 5.5: Maximum and minimum header portion within which any PVS-scheme's header portion is located.

Hence, the SI-based PVS should be preferred over EI-based PVS if the number of EI-pixels is less than the square root of the number of pixel in the II-picture. Analogously, comparing SI with RI gives

$$\begin{aligned} H_{RI} &> H_{SI} \\ H\left(U \cdot L, r\right) &> H\left(K \cdot L, r\right) \\ \frac{1}{U \cdot L} &> \frac{1}{K \cdot L} \\ \frac{M \cdot N}{U \cdot V} &> U \cdot \frac{N}{V} \\ \sqrt{M} &> U, \end{aligned} \tag{5.15}$$

which indicate that an SI-based PVS is better qualified to provide high coding efficiency than a RI-based PVS if the horizontal EI-resolution $U < \sqrt{M}$. The relation only contains horizontal resolution variables, because the PVS-frame's vertical resolution is $T = L$ for both SI and RI and thus canceled out.

Adding the constraint of quadratic EIs allows Equation (5.13) to produce H_{EI}, H_{SI} and H_{RI} as a function of $U \cdot V = U^2$. These three functions are shown in Figure 5.6 and show that SI provides less header overhead than both EI and RI if $U^2 < \sqrt{M \cdot N}$.

The dividing lines between CI-types shown in Figure 5.6 (where the benefit of one

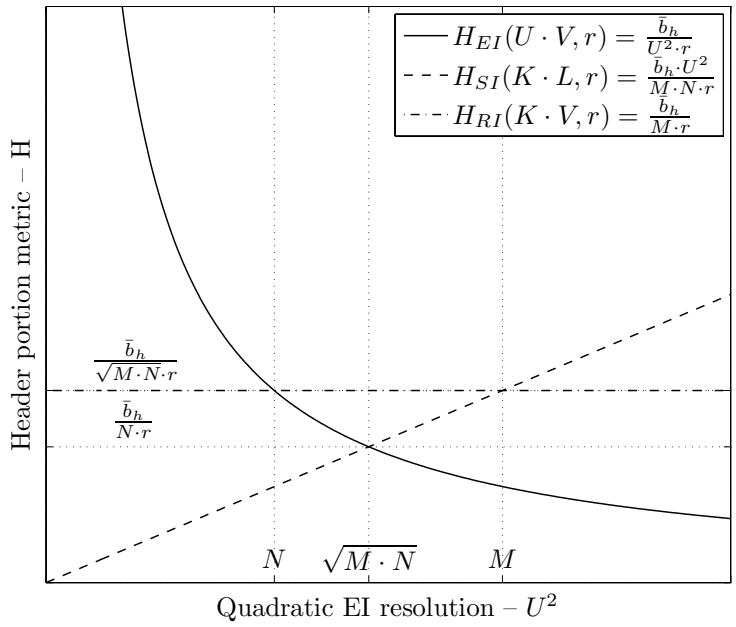


Figure 5.6: Header portion for the EI-, SI- and RI-based PVS-schemes with accompanying dividing lines for when each CI-type is more beneficial.

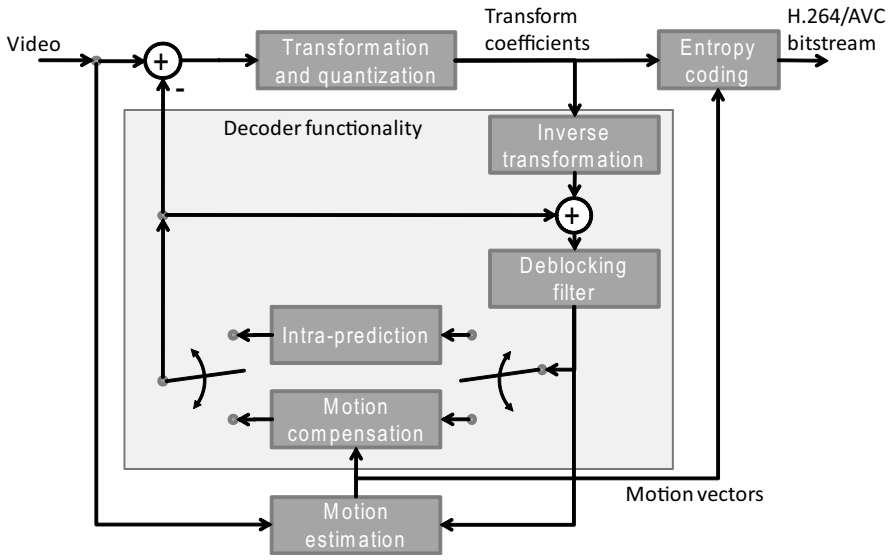


Figure 5.7: The subsystems of a H.264/AVC-encoder.

supersedes the others) are also verified experimentally in Section 5.8.3 by comparing coding efficiency.

5.6 Coding the PVS or PVI

Forming the PVS or PVI is only the first step in coding the II-picture. The intermediate form that the PVS or PVI constitutes, aims to be a format well suited for 2D video and volumetric image coding tools. Thereby a large portion of the redundancy inherent in the static II-based 3D images can be exposed and reduced.

5.6.1 H.264/MPEG-4 AVC

The H.264 / AVC is the current state-of-the-art 2D video coding standard, which is the reason for adopting it as the video coding part of the proposed SI-based PVS coding scheme. It obtains its performance from a block-based hybrid coding approach, just as its precursors MPEG-2, MPEG-4 part 2, H.262 and H.263. These coding approaches reduces the inherent spatial and temporal redundancy of the video sequence by combining both prediction and transformation tools. The main difference which sets aside H.264 / AVC is the greatly increased number of ways in which this reduction can be done, i.e. the variety of coding tools that are available for use [102]. Figure 5.7 shows a block diagram of a typical H.264 / AVC-encoder structure. [103].

In previous video coding standards, the spatial redundancy has been exposed for reduction by transforming a block of pixels using DCT. This decorrelating transform concentrates the pixels' energy into a smaller number of transform coefficients thereby facilitating a reduction in bitrate. However, H.264/AVC first predicts the pixel block from neighboring pixels before applying its DCT-like integer transform to the residual. The actual bitrate reduction is accomplished by scalar quantizing the transform coefficients corresponding to the intra-prediction residual. Knowledge about the HVS and its reduced sensitivity for distortion in high spatial frequencies, is considered when the quantizer step-size is chosen for the individual transform coefficients. As Figure 5.7 illustrates, this integer transform is also applied to the residual from the temporal prediction. Temporal redundancy, i.e. similarities between neighboring pictures, is reduced using block-based Motion-Compensated Prediction (MCP). For each block in a picture, similar blocks are searched for in one or several reference pictures and thus motion is estimated. The change in position between the blocks is described using a motion vector. The residual after subtracting the two blocks is transformed, quantized and combined with the motion vector. The decision to use intra-prediction or motion estimation is made on a block by block basis. A picture with only intra-predicted blocks is denoted an Intra coded picture (I-picture)and acts as a starting point for the MCP. A Prediction coded picture (P-picture) can on the other hand contain inter-predicted blocks, i.e. blocks predicted from other previous I-pictures or P-pictures. Furthermore, Bi-direction coded picture (B-picture)s can be predicted from previous *and* subsequent I-pictures and B-pictures. B-pictures in H.264/AVC differ in characteristics compared to previous standards as they themselves can be used as reference pictures. Additionally, more than two reference pictures can be used to predict a single B-picture. The abundance of variably selected block sizes allows for a more adaptive MCP. The enhanced entropy coding stage, which uses variable length coding, is also a great contributor to the improved coding efficiency of H.264/AVC. The de-blocking post-processing filter is also a contributor, which increases the subjective quality of the coded video at low bitrates. Despite the spatial and temporal redundancies there is also a chromatic redundancy that is utilized in H.264/AVC, much the same as in other video coding standards. Changing the color space of the sequence from RGB to YCbCr benefits coding because of the lower correlation between components in YCbCr. In addition, the HVS is less sensitive to the color difference components Cb and Cr, which makes it possible to low-pass filter and sub-sample these without any significant perceivable distortion. To control the vast number of coding alternatives, rate-distortion optimization algorithms of various complexities are almost mandatorily used in H.264/AVC encoders. Further information about the H.264/AVC standard can be found in numerous overviews and introductions [102–104].

There is one main aspect that affects how the coding tools of H.264/AVC is interpreted when they are applied to a PVS as opposed to a 2D video sequence. Pseudo time corresponds to different spatial variables and not to time. For example in the SI-based PVS an increased pseudo time index j refers to different viewing directions of an orthographic camera. For the EI-based PVS j is equivalent to a change in position of perspective camera (with its center restricted to a plane and the viewing direction normal to this plane). As a result of this change in basis, video coding terminology

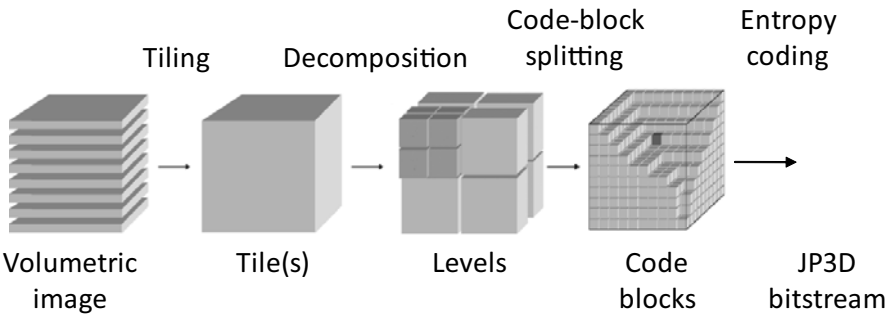


Figure 5.8: The JP3D-process: tiling, decomposing and code-block splitting. The figure extends on an image from [105].

also changes meaning. MCP for example actually refers to disparity compensation when H.264 / AVC is used for EI- and SI- based PVS coding. That is, what the encoder believes to be temporal redundancy, and addresses by P-pictures or B-pictures, are actually different kinds of inter-view redundancies that stem from viewing the captured scene from different perspectives, angles or both. This also leads to other implications concerning video sequence properties such as scene-change, fade-in / fade-out, camera zoom and rotation etc. These properties are difficult for a video encoder to handle efficiently, including those adhering to H.264 / AVC. As far as the PVS is concerned these properties do not exist explicitly, even though similar characteristics can be seen depending on the PVS transform. For example in the SI-based PVS a type of scene-change can occur when a II-picture depicts a scene with few objects and a horizon. The orthographic nature could then result in one PVS-frame containing all sky and no ground whereas the next could contain all ground and no sky; thereby causing a large residual after motion compensation straining the encoder. Using the column CISO when producing a PVS from the set of SIs in Figure 2.25 (b) would result in this scene-change property.

5.6.2 JPEG2000 Part 10 (JP3D)

Having the II-picture in the form of a PVI enables the use of JP3D, which is a 3D extension to JPEG2000 and was recently accepted as an official ISO / ITU-standard [101]. In a similar manner to that for JPEG2000 Part 1, JP3D tiles the image as the first step. The volumetric image is tiled into cuboid shaped subsets of arbitrary but equal size. A tile can be coded and decoded independently, which allows for random access within the volume. Larger tiles improves coding efficiency as they cumulate similar frequency content from a larger portion of the signal at the cost of requiring more memory to encode and decode. Each tile is decomposed using a discrete wavelet transform (integer or floating point) into a arbitrary number of levels per dimension. The transform coefficients from each level or sub-band are then combined into code-blocks as shown in Figure 5.8. Each code-blocks is entropy coded bitplane

by bitplane which allows for a progressive decode. Furthermore, the entropy code uses truncation to allow for scalable coding with a finer granularity than a twofold increase or decrease in bitrate.

The JP3D-features adopted for volumetric imaging applications (e.g., scalability, random access, and progressive decoding) cost in terms of coding efficiency. For II-pictures that are intended for direct viewing on an II-display these features are of little use. More importantly, their cost in terms of reduced coding efficiency puts a JP3D-coded PVI in a disadvantageous position compared to an H.264 / AVC-coded PVS as will be shown later.

5.6.3 Coding cost

The PVS coding schemes rely on the use of two complex compression standards, which prohibits the explicit specification or measurement of the number of mathematical operations involved in coding. The next best option is to implicitly measure the computational cost by explicitly timing the CPU-time for encoding (T_e) and decoding time (T_d) of the PVS or PVI coding schemes. The coding time may be further factorized into pre-processing / coding and post-processing / decoding according to

$$T_c = T_e + T_d = T_{LUT} + T_{enc} + T_{dec} + T_{LUT^{-1}} \tag{5.16}$$

to determine the influence forming PVS or PVI has on coding time when a Look-Up Table (LUT) is used. Further factorizing the time into operations such as CI-transformation and CISO; or coding operations such as color space conversion (RGB to / from YCbCr), motion / disparity estimation, DCT calculations etc. is outside the scope of this work. Moreover, forming a PVS or PVI is a one-to-one mapping that could be performed in constant time using the LUTand is therefore also excluded from explicit CPU-time measuring. Thus, the coding cost evaluated subsequently is T_{enc} and T_{dec} only.

Note that the measured CPU-time should mainly be read as indications of relative complexity between the evaluated coding schemes. The CPU-time is very much a consequence of the amount of optimizations each software encoder has undergone with respect to the platform on which the software is running.

5.7 Experimental setup

The II-pictures used in the following experiments were synthesized using the approach presented in Chapter 3. Using simulation instead of physical experimentation offers two advantages: exact knowledge about 3D scene and II-camera is achieved and can be exploited to verify results; comparative research is easily performed inexpensively and quickly, e.g. evaluating how different II-camera properties affect the efficiency of the proposed coding scheme. All experiments were conducted on a PC-system with a 3 GHz Pentium 4 and 3 GB RAM running Windows

Table 5.1: II-camera parameters

Parameter	Value
Lenslet focal length - f [mm]	0.73
Lenslet pitch - δ^l [mm]	0.39
Pixel sensor resolution - $M \times N$ [pixels]	4096×4096
	8192×4096
Pixel sensor size [mm^2]	25×25
	50×25
Pixel pitch - δ^p [μm]	6.1

XP SP2. All disk I/O was conducted against a 256 MB ram disk to remove any dependencies on hard drive architecture. The measurement of CPU-time is used for relative comparisons between the coding schemes. By not considering absolute timing the experiments aim to eliminate the major impact on CPU-time caused by RAM speed, CPU cache size, number of CPU cores etc..

5.7.1 II -camera model

A pinhole model was used to simulate the II-camera's optics; approximating lenses with pinholes is a feasible simplification within this context since optical power efficiency is not an issue in computer simulation. The II-camera's pixel sensor noise sources (amplification noise, photon noise etc.) were jointly modeled using normally distributed additive monochromatic noise $N(\mu = 0, \sigma = 1)$. With regard to resolution the pixel sensor was set to be on par with current state-of-the-art sensors, e.g. the 16 Mpixels and >30 Mpixels sensors used in the Canon EOS-1Ds Mark II and the Hasselblad H3DI respectively. Details about the II-camera model is summarized in Table 5.1. Note that different experimental setups were employed when conducting the different experiments, which is the reason for different pixel sensor resolution and size. The main difference between the setups was the II-picture structure used. Table 5.2 present the three setups.

Setup 1 is used for defines an II-picture structure that corresponds to the ratio $\frac{K \cdot L}{U \cdot V}$, which the coding schemes are designed to address. The number of EIs ($512 \cdot 256$) is set such that the spatial resolution of the 3D images is similar to the spatial resolution on Standard Definition TV (704×576). The EI-resolution (16×16) is set to be on par with state-of-the-art II-cameras [35–37].

Setup 2 varies the II-camera characteristics enabling an evaluation of how the different PVS schemes are affected by different II-picture structures. Each II-picture structure has a constant II-picture-resolution $M \times N$ but with different EI- and SI-resolutions.

Setup 3 sets $U \times V = K \times L = 64 \times 64$, thereby achieving an II-picture structure that is neutral from a PVS-picture resolution standpoint. That is, the resulting PVSs have equal resolution and length regardless of chosen CI-type. This setting results in a square II-picture with ≈ 16.7 Mpixels, contrary to the other setups that employ

Table 5.2: II-picture structures used in experimental setups

	Setup 1	Setup 2	Setup 3
II-picture resolution – $M \times N$	8192×4096	8192×4096	4096×4096
SI resolution – $K \times L$	512×256	512×256 256×128 128×64 64×32 32×16	64×64
EI resolution – $U \times V$	16×16	16×16 32×32 64×64 128×128 256×256	64×64

an aspect ratio of 2 and ≈ 33.5 Mpixels.

For Setup 1 – 3, the H.264 / AVC macro-block size of 16×16 pixels imposes a principle restriction on the II-picture structure. The resulting PVS-frame resolutions, regardless of chosen PVS-approach, must be an integer multiple of this macro-block size for the PVS to be possible to code without padding with zeros pixels or mirroring the edge pixels.

5.7.2 II-pictures

In order to experimentally support and evaluate the previous theoretical discussions, four reference 3D scenes were defined as origins from which the II-pictures used in the experiments were synthesized. Each 3D scene was designed to differently stress three specific scene characteristics:

- Detail – the amount of high frequency content produced by inter alia fine texture.
- Depth – the variation of the scene objects in the z-direction.
- Fill factor – the amount of the scene containing objects.

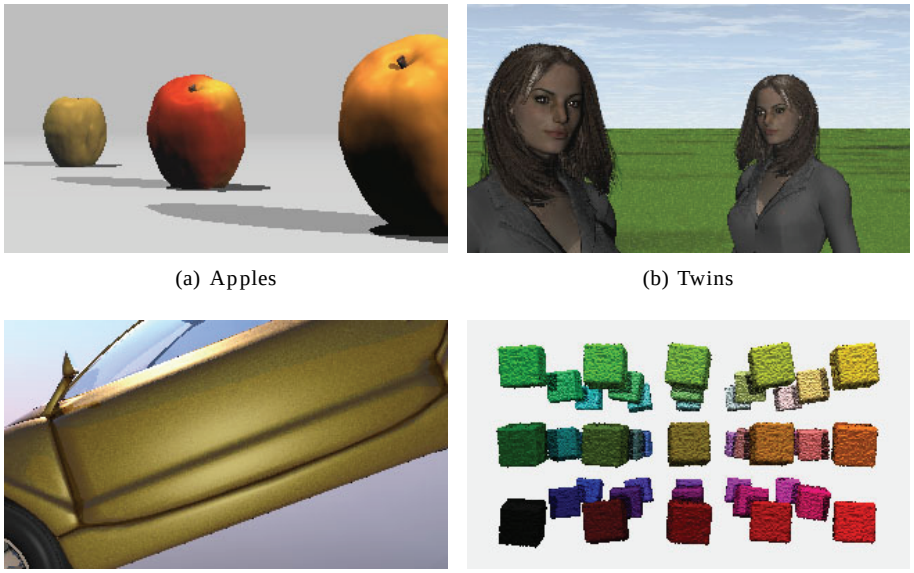
The reference 3D scenes characteristics' are summarized in Table 5.3 and a perspective projection of each scene is shown in Figure 5.9.

5.7.3 Coding parameters

All coding schemes were evaluated in the bitrate-range $r = \{0.015, \ldots, 1.5\}$ bpp. For the PVS -schemes this is transformed into a PVS bitrate, which the H.264 / AVC-encoder must adhere to. The JPEG2000- and JP3D-encoders operate on the bitrate r

Table 5.3: Reference 3D scene characteristics

Scene	Degree of detail	Depth range	Fill factor
Apples	low	long	low
Twins	high	short	low
Car	low	short	high
Cuboid	high	long	high



(a) Apples (b) Twins

(c) Car (d) Car

Figure 5.9: Two dimensional perspective projections of the four II-picture s (a) Apples, (b) Twins, (c) Car and (d) Cuboid.

directly. When using these encoders there is a vast number of parameters that influence the final quality of the coded video sequence or volumetric image. Parameterizing these optimally is partially the responsibility of the encoder's rate-distortion optimization algorithms and partially the task of the user. Selecting which parameters to consider is also a consequence of the intended application.

5.7.3.1 H.264/AVC

The following H.264 / AVC parameters were selected to be studied with respect to their effect on coding efficiency: picture type distribution (GOP-structure), motion vector search area and bitrate control technique.

The distribution of I-pictures, P-pictures and B-pictures over the set of PVS-frames is an important parameter to consider. A larger portion of predicted pictures (B-pictures and P-pictures) in the PVS will allow for a more efficient reduction of the redundancy in the pseudo time domain. Setting the GOP-length to J PVS-frames maximizes the temporal prediction length. The integer-based transform used in H.264 / AVC makes the transform and inverse transform a lossless operation, which removes the concern about error-drift for long prediction runs. Hence, the default value for the image type distribution was set to $IP\ldots$ and compared with $IBP\ldots$, $IBBP\ldots$, and $IBBBP\ldots$.

The maximum search area for motion estimation / compensation is a parameter that is mainly restricted by coding time. A large search area will be able to reduce larger translational motion occurring between consecutive PVS-frames. However, a larger search area comes at the price of increased coding time. The default search area was set to 16×16 pixels and compared with 32×32, 64×64 and 128×128.

A H.264 / AVC-encoder can comply with a requested final size B (see Equation (5.8)) in many different ways. The majority of techniques require a feedback control system to insure that the resulting size does not deviate from the desired value. CBR controls the quantization such that a constant bitrate is achieved at the expense of a varying quality of the PVS-frames. VBR instead varies the bitrate while attempting o produce constant quality. The default technique used in the experiments was CBR and two additional VBR-techniques was also studied: Constant quantizer (CQ) and 2-pass coding. CQ keep the quantization parameter QP fixed throughout the whole encoding. The main disadvantage with this approach is the difficulty in complying with a requested size B as no feedback is being used. The 2-pass encoding allows the encoder to gather statistics about the video sequence in a first pass, which it can later exploit in the second encoding pass.

The H.264 / AVC-encoder used for the experiments was x264, a free library for encoding H.264 / AVC video streams [106].

5.7.3.2 JPEG2000 and JP3D

The coding parameters for JPEG2000 and JP3D were set as follows. Both approaches were set to utilize the maximum tile-size, which maximizes the coding efficiency. However, tiling still had to be conducted for PVIs where $J > 32768$ as these exceeded the maximum tile size allowed in the z-dimension of JP3D. For example, the EI-based PVI constructed using Setup 1 was divided into $\frac{8192 \cdot 4096}{32768} = 1024$ tiles in the z-dimension. In both cases the CDF $5/3$ wavelet kernel was used to decompose each tile in 5 levels per dimension. The CDF $5/3$ kernels integer coefficients makes the transformation a lossless operation. Each tile was decomposed in 5 levels per dimension. However, for CI-types EI and RI – where either or both of the S- and T resolutions became 16 pixels – no decomposition (level = 0) was performed for that dimension enabling the PVI to be at all possible to code.

The Kakadu software framework was used as a JPEG2000-encoder, whereas the JPEG2000 Part 10 - Verification Model JP3D was used to encode JP3D [107, 108].

5.8 Results

This section will present the results from a number of empirical rate-distortion analyzes, which examine different factors influencing the coding efficiency. Firstly, the PVS coding approach is compared with 2D images coding where the effects of CI-transform and CISOs are studied. Secondly, the effect of bitstream headers is empirically evaluated followed by a comparison of different CISOs. Thirdly, a study is conducted aiming to show how the coding schemes operated with respect to view- and depth quality and what visible coding artifacts they introduce. Finally the time complexity of the coding schemes is investigated.

5.8.1 PVS vs 2D images coding

Figure 5.10 shows the Q_{global} for the four II-pictures coded using EI-, SI-, and RI-based PVS, SI-based PVI and the JPEG2000 coded II-picture. By using Q_{global}, the complete effect that the coding schemes have on the II-pictures is captured. This experiment verifies that PVS-coding provides significantly higher coding efficiencies than applying state-of-the-art 2D images coding to II-pictures adhering to the II-camera category defined in Setup 1. The SI-based PVS achieves an average increase of 17.9, 12.6, 14.2 and 10.9 dB in Q_{global} compared to JPEG2000, for the four II-pictures. Furthermore, the SI-based PVS requires less than approximately $1/60$-th of the bitrate to produce a quality corresponding to JPEG2000 (see Figure 5.10 (b) and (c)). For example, Twins in Figure 5.10 coded at 0.015 bpp gives a Q_{global} 25 dB that JPEG2000 requires 1 bpp to produce. Note the significantly worse results for the EI-based PVS which is in-line with the expected penalty from a large required header portion. The EI-based PVS results in a Q_{global} that on average is 13.4, 8.2, 9.1 and 5.8 dB below the SI-based PVS. For all but a few bitrate ranges, the proposed

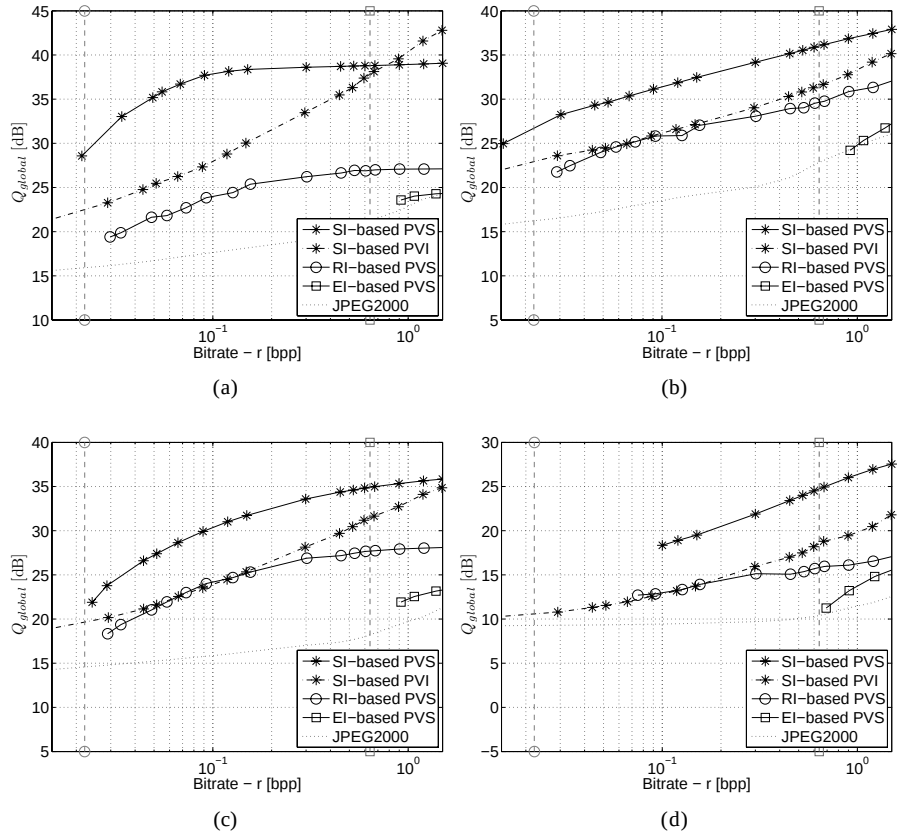


Figure 5.10: Q_{global} for Setup 1 and reference II-picture (a) Apples, (b) Twins, (c) Car and (d) Cuboid. The vertical lines from left to right mark the minimum achievable bitrate for the RI- and EI-based PVS respectively.

SI-based PVS also outperforms the corresponding SI-based PVI, albeit not as significantly as the EI- and RI-based PVSs. An average difference of 5.1, 4.5, 4.3 and 6.0 dB in favor of the PVS is measured. Stated differently, the PVS can produce the same medium-level distortion ($\approx$ 30 dB) as the PVI at approximately a fourth of the bitrate for all evaluated II-pictures.

The reference scenes effect on the PVS coding quality can also be studied in Figure 5.10. Cuboid has the least, but yet significant, PSNR improvement of the four scenes when comparing the SI-based PVS with the 2D images coding approach. For this scene, there exists less redundancy between views that can be exposed and reduced; a property caused by the scene's high detail and long depth. The low complex Apples and and Car on the other hand have a strong inter-view redundancy that is efficiently reduced using PVS coding.

Note that even though the average results for EI are superior to JPEG2000, they are not averaged over the same bitrate range and are therefore not directly comparable. The EI-based PVS is not possible to code below a certain bitrate corresponding to the bitrate that is approximately required for carrying the header data alone. The major part of the bitrate is required by the header portion of the bitstream for a PVS with $J = 512 \cdot 256 = 131072$ low resolution PVS-frames. A lower bound on bitrate may be calculated by coding an all black II-picture (with zero-valued pixels) using the coding schemes. An all black video sequence coded using x264 with maximized quantization parameter QP=51 results in a bitstream containing the header information alone, approximately. No video sequence with similar properties in resolution and length can thus be coded at a bitrate lower than the minimum bitrate required for this header-only bitstream. The two vertical lines in Figure 5.10 correspond to the minimum bitrates for the RI- and EI-based PVS respectively. That is, an EI-based PVS may not be coded at a bitrate lower than 0.63 bpp. Hence, the EI-based PVS is only superior to JPEG2000 for high bitrates; it cannot be used at all for lower bitrates, as Figure 5.10 illustrates. Vertical lines corresponding to minimum bitrate of the other coding schemes are located below the range of bitrates shown in the figure.

The incomplete curves of the SI-based PVS are contrary to the other transforms, not a consequence of a significant header bitrate portion. Instead they are the result of scene complexity. For complex scenes with high detail, depth and fill factor, the introduced disparity between neighboring PVS-pictures is high. This strains the MCP such that even if the H.264 / AVC-quantization parameter is set to its maximum value ($QP = 51$), there still remains significant energy in the prediction residual. In Figure 5.10, this appears as incomplete curves for the SI-based PVS scheme for all but Twins caused by a lower bitrate being impossible to achieve unless a nonstandard quantization parameter ($QP > 51$) is used. If it would be possible to set a value of $QP > 51$, and thereby thus introduce a harder quantization of the residual from the motion compensation stage, a lower bitrate than that shown would be achievable. Of course this would come at the price of a lower Q_{global}.

Using JP3D instead of H.264 / AVC, given the same CI-type and CISO, is only an option for high bitrates and certain II-pictures where Q_{global} levels away for the PVS approach, e.g. Figure 5.10 (a). The reason for the asymptotic behavior of the PVS is a combination of low complex II-pictures and the compression methods of

H.264 / AVC. An H.264 / AVC-encoder introduces a certain amount of distortion even when its quantization parameter QP is set to a minimum (QP=1). The HVS-weighted quantization matrix that QP modulates, always causes measurable distortion albeit not necessarily perceivable. Thus, lossless coding can not be achieved using H.264 / AVC by merely reducing the parameter QP to its minimum. This is contrary to JP3D, which allows for a gradual transition from lossy to lossless coding.

As a result of the large advantage of using PVS compared to PVI, the subsequent evaluation will focus solely on the PVS. However, the use of PVI may still be justified for certain combinations of bitrates, contents, and applications requirements.

5.8.2 SI-based PVS selection orders

The previous section showed that the SI-based PVS applied to the Setup 1 II-picture structure, is superior in coding efficiency compared to the other CI-types. This section investigates the importance of choosing a specific CISO for the SI-based PVS. Table 5.4 presents $\overline{c}$, σ_c and e for all combinations of CISOs. The best results are marked in bold font in the table. The CISO holding the majority of high $\overline{c}$, low σ_c and low e is the parallel, which suggests that forming a SI-based PVS using parallel CISO should provide the highest coding efficiency. To verify this, Figure 5.11 presents Q_{global} for the different CISOs. The graphs show that the parallel CISO is the most efficient, even if the difference between row and parallel is very small. The worse performance of the column CISO is mainly due to the view discontinuities introduced when going from the bottom of one column, to the top of the next. This occurs with a period of 16 images, straining the motion compensation of H.264 / AVC. There is a noticeable performance increase for horizontal CISOs (row and parallel) over those containing a stronger vertical component (column, zigzag and spiral). This is largely an effect of the foremost horizontal object positioning in the used reference scenes. For scenes containing an extensively random object positioning, the difference between CISOs is anticipated to be low.

5.8.3 Working range for the SI-based PVS

Section 5.5.4 showed that it is important to consider the bitrate portion required for header data when selecting CI-types for the PVS. This finding is also validated in the following experiment where the coding efficiency of PVS-schemes based on EI and SI is compared. Setup 2 with its variable II-picture structures is used for this comparative evaluation. The difference in Q_{global} is used to collectively capture all aspects that affect the coding efficiency. I calculate the difference as

$$\Delta Q_{global}(r) = Q^{SI}_{global}(r) - Q^{EI}_{global}(r), \tag{5.17}$$

where a positive difference speaks in favor of using SI- instead of EI-based PVS for the specific II-picture structure. The result is presented in Figure 5.12 for Apples and Car (similar results are produced for Twins and Cuboid). Using SI instead of EI is beneficial for II-picture structures ($K \times L \times U \times V$) $512 \times 256 \times 16 \times 16$,

Table 5.4: Mean and standard deviation ($\overline{c}$,σ_c) of cross-correlation coefficient c and difference residual energy e

	Apples			Twins			Car			Cuboid		
	$\overline{c}$	σ_c	e	$\overline{c}$	σ_c	e	$\overline{c}$	σ_c	e	$\overline{c}$	σ_c	e
SI row	0.624	0.282	1.66	0.812	0.139	1.1	0.677	0.136	1.89	-0.012	0.279	15.7
SI parallel	**0.649**	**0.265**	1.48	0.844	**0.071**	**0.872**	**0.699**	**0.072**	**1.74**	-0.007	0.279	**15.6**
SI spiral	0.632	0.275	**1.46**	**0.845**	0.115	0.946	0.607	0.137	2.28	**-0.151**	**0.163**	17.8
SI col	0.593	0.308	1.57	0.839	0.188	0.995	0.454	0.22	3.1	-0.017	0.272	15.8
SI zigzag	0.552	0.257	2.1	0.814	0.107	1.22	0.592	0.084	2.34	-0.023	0.257	15.7

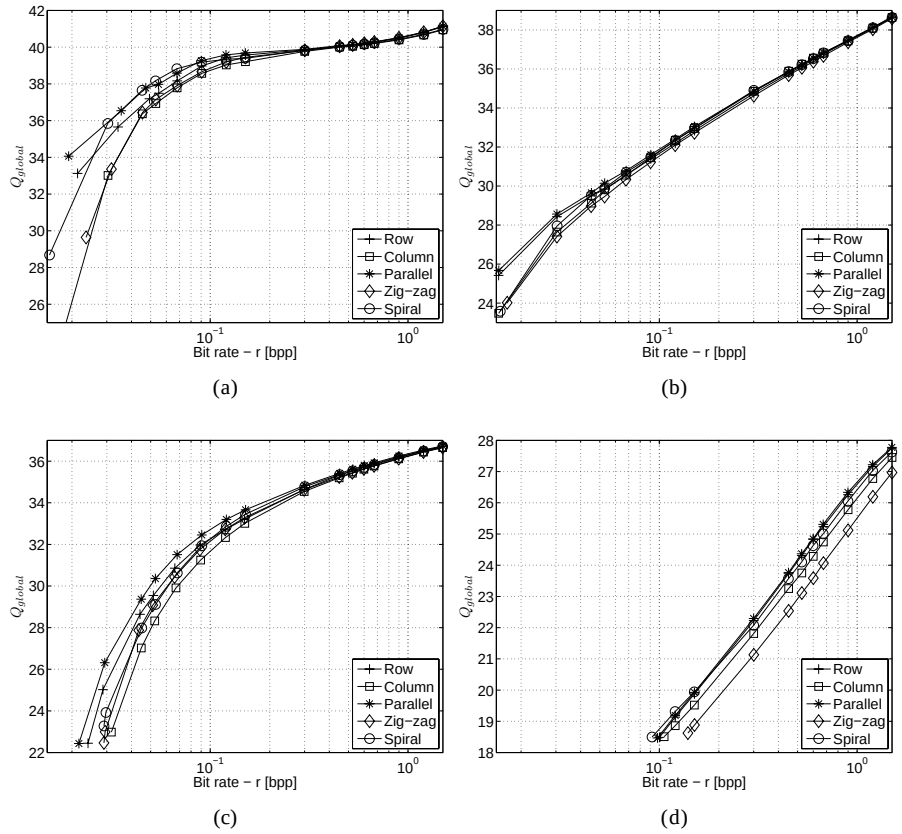


Figure 5.11: CISO evaluated using Q_{global} for Setup 1 where (a) Apples, (b) Twins, (c) Car and (d) Cuboid.

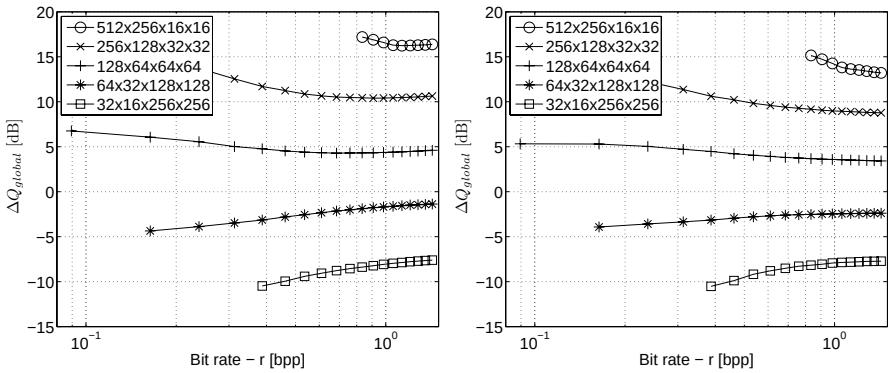


Figure 5.12: Comparison of SI- and EI-based PVS-schemes using ΔQ_{global} for Setup 2. (a) Apples and (b) Car.

Table 5.5: The effect of picture type distribution and GOP-length on Q_{global} averaged over all r for Apples adhering to the Setup 1 II-picture structure. Default parameters where $IP\ldots$ and GOP-length equal to $J = 256$.

	$IP\ldots$	$IBP\ldots$	$IBBP\ldots$	$IBBBP\ldots$
256	0.00	-0.57	-0.75	-0.96
128	-0.05	-0.67	-0.82	-1.02
32	-0.23	-0.99	-1.13	-1.35
16	-0.42	-1.25	-1.48	-1.67

$256\times128\times32\times32$ and $128\times64\times64\times64$. This is in line with the theoretical results that indicate that SI should be used over EI when constructing the PVS for EI-resolutions less than 76×76 ($U^2 < \sqrt{8192 \cdot 4096} \approx 76^2$).

5.8.4 Coding parameters

Table 5.5 summarizes the effects on coding efficiency that results from changing the picture type distribution and GOP-length when coding a SI-based PVS from Apple adhering to Setup 1. The other reference II-pictures give similar results and are therefore excluded. Contrary to the previous study, the difference in Q_{global} is here defined according to

$$\Delta Q_{global}(r) = Q_{global}(r) - Q^{ref}_{global}(r), \tag{5.18}$$

where Q^{ref}_{global} is the reference result from evaluating the PVS with default parametrization and Q_{global} is a placeholder for the other parameterizations. Hence, when Equa-

tion (5.18) is negative the default values (picture type distribution $IP\ldots$ and GOP-length $J = 256$) are preferred over the evaluated parametrization.

Neither reducing the GOP-length nor introducing B-pictures have a positive effect on coding efficiency. Reducing the GOP-length from the default length J allows for random access within the PVS at the cost of reduced coding efficiency. However, such functionality is superfluous when the II-picture's intended use is to be presented on a II-display in its entirety. Splitting the PVS into 16 GOPs with 16 PVS-frames each reduces the coding efficiency with 0.42 dB for Apple. For Twins, Car and Cuboid the corresponding reduction is 0.54, 0.51 and 0.12 dB respectively. Decreasing the GOP-length to $< J$ would facilitate random access within the PVS but at the expense of a reduced coding efficiency. Incorporating B-pictures are often considered to imply an improved coding efficiency when coding 2D video. A B-picture could utilize the information from a subsequent as well as a preceding picture, which would reduce the residual energy after motion compensation compared to a corresponding P-picture. The total number of bits for a B-picture, including the additional motion vector and header bits, would be lower than for a P-picture while still giving equal or better quality. Bits spent on a B-picture have in previous standards been non-reusable since B-pictures have not been allowed to act as references themselves. In H.264 / AVC however, B-pictures are permitted to act as references for further prediction, which should make their means to increase the coding efficiency more clear.

Incorporating B-pictures would allow for predicting in both front- and back directions of the PVS. B-pictures are considered especially advantageous when new scene content is revealed in a manner that cannot be compensated efficiently using P-pictures. A B-picture enables prediction from subsequent frames, which is an advantage if those have a stronger correlation with the revealed content than preceding frames. Hence, the B-pictures chance to increase the coding efficiency relies on CISOs that reveal content that cannot be efficiently predicted from previous frames. For the selected CISO (parallel), a minimum number of extensive changes are introduced for the reference II-pictures. Thus, the advantage of B-pictures is reduced and instead their cost in terms of reduced quality dominates. If a picture distribution of $IBBBP\ldots$ is used the coding efficiency is reduced by 0.96 dB compared to the default $IP\ldots$. For Twins, Car and Cuboid the corresponding reduction is 0.78, 0.53 and 0.74 dB respectively.

Increasing the motion vector search range allows for the compensation of larger translations between consecutive PVS-frames, which is beneficial for the coding efficiency. However, increasing the search range implies an increased number of comparisons and thus a longer encoding time. In Table 5.6 the coding efficiency in terms of difference in Q_{global} is summarized for Twins together with the corresponding encoding time T_{enc}. The encoding time T_{enc} has been normalized with respect to the default parametrization, resulting in the unit-free relation

$$C_T = \frac{T_{enc}}{T_{enc}^{ref}}, \tag{5.19}$$

where T_{enc}^{ref} is the encoding time for the default reference parametrization that uses

Table 5.6: The effect of motion vector search area on Q_{global} averaged over all r for Apples and Twins adhering to Setup 1 II-picture structure. Default parameter was 16×16 pixels.

	Apples		Twins	
	ΔQ_{global} [dB]	C_T	ΔQ_{global} [dB]	C_T
16×16	0	1	0	1
32×32	0.12	2.8	0.01	3.3
64×64	0.18	9.2	0.05	11
128×128	0.21	28	0.11	31

Table 5.7: The effect of bitrate control techniques on Q_{global} averaged over all r for Apples and Twins adhering to Setup 1 II-picture structure. Default parameter was CBR.

	Apples		Twins	
	ΔQ_{global} [dB]	C_T	ΔQ_{global} [dB]	C_T
CBR	0	1	0	1
VBR CQUANT	0.39	0.95	0.29	0.96
VBR 2-pass	0.15	1.12	0.18	1.30

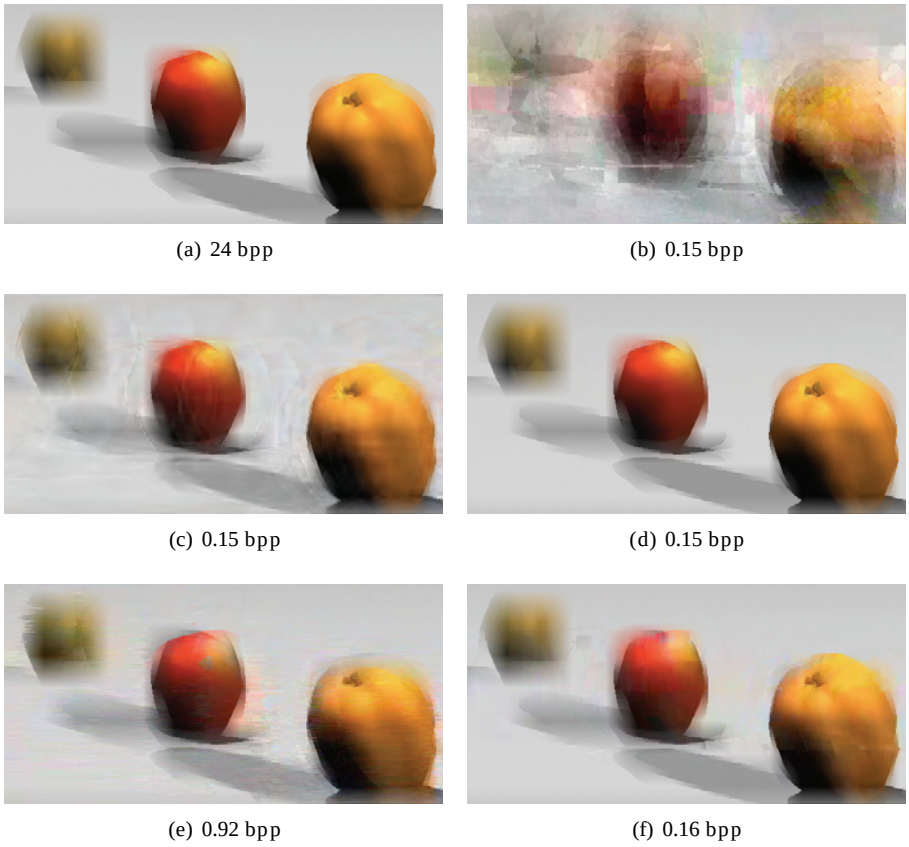
CBR. Twins, with its low depth range, shows no significant improvement in coding efficiency when the motion vector search range is increased. Apples on the other hand benefits from increasing the search range all the way up to 128×128, with an improved Q_{global} of 0.21 dB compared to the default range of 16×16. Both II-pictures show an exponential increase in encoding time, as expected. Hence, increasing the motion vector search area only benefits the coding efficiency for 3D scenes that have long depth ranges due to the orthographic nature of the SI. The more distant an object is from the II-camera, the farther the object's projection is translated between neighboring PVS-frames. For example, the low depth range of Twins only induces translations between PVS-frames that can be captured within the default 16×16 search area. Hence, increasing the search area when coding Twins has no benefit with respect to coding efficiency.

Finally, the three different bitrate control techniques are evaluated. The result in ΔQ_{global} and C_T is summarized in Table 5.7 for Apples and Twins. On average, using VBR CQ gives the best results of the three bitrate control techniques. Moreover, the encoding takes approximately 5 % less time than CBR as a result of not explicitly controlling the bitrate at all. Note however that when VBR CQ is used the requested size B is never closely matched since QP is the control parameters and not r. If a low deviation in the resulting B is required, then VBR 2-pass is a good compromise between CBR and VBR CQ. However, the additional first pass gathering statistics comes at the price of an increased encoding time compared to CBR. This corresponds to an $<= 30$ % increase in T_{enc} relative to CBR, when encoding to the experimental setup's ram disk.

5.8.5 Coding artifacts

The results presented until this time have been based on evaluating the complete II-picture using Q_{global}. Albeit suitable for comprehensive analysis, Q_{global} provides no detailed information about how the coding induced distortion may be perceived. This experiment fulfills this requirement by presenting VIs for subjective evaluation followed by $\overline{\mathbf{Q}}_{view}$ for the PVS-approaches. Setup 3 is used to examine the coding artifacts that are introduced from the PVS schemes, despite the fact that the proposed coding schemes are designed for II-picture structures corresponding to Setup 1. The amount of bitstream header penalty for EI- and RI-based PVS using Setup 1 results in such poor coding efficiency that these selections of CI-type and CISO would produce unusable 3D images. The EI-based PVS is not even capable of producing II-pictures with a bitrate $r < 0.63$ bpp as discussed previously. Figure 5.13 shows the significant coding artifacts introduced by some of the coding schemes when Apple using Setup 1 is coded with a requested bitrate $r = 0.15$ bpp. The lack of a depth-control lens in the used II-camera model is the reason for the blurriness of the more distant apples, even in the original VIÀs discussed in Section 2.2.4, a depth-control lens would allow for both real and virtual objects and hence approximately double the useable depth range. The absolute difference images with respect to the original uncoded image are also shown in Figure to more clearly uncover the distortion within the images of Figure 5.13. Figure 5.14 (a), which represent a front view of the uncoded II-picture, are completely white as a natural consequence of not containing any coding artifacts, which results in an absolute difference of zero. The other images are calculated with respect to the uncoded front view and the absolute difference have then been inverted for presentation purposes. Thus, white pixels correspond to no difference between the two front views of a coded II-picture and the original uncoded II-picture. From studying the coding artifacts presented in Figure 5.14 (b) – (f) is is evident that the SI-based PVS coding scheme induces the least amount of distortion of the evaluated coding schemes. Comparing Figure 5.14 (b) – (c) with Figure 5.14 (d) – (f) reveals a difference in coding artifact character between the wavelet approach of JPEG2000 and JP3D, and the hybrid approach of H.264 / AVC. The artifacts caused by quantizing a wavelet decomposition (be it 2D or 3D) are relatively evenly distributed within the 3D image. The distortion induced by H.264 / AVC is located to the scene objects in general and localized to the boarders of the objects in particular.

The above figures have made clear that the SI-based PVS is superior with regards to coding efficiency for the Setup 1, for which it was designed. The reason for this being mainly its low header portion H, which allows more bits to be spend on increasing the 3D image quality than to maintain the necessary bitstream structure. Hence, in order to further study the coding artifacts of the different coding schemes a setup that does not favor a particular scheme must be used. Thus, this section thus present the induced coding artifacts of the PVS-coding schemes when applied to a II-picture structure that favors no particular CI-type and thereby investigates how the SI-based PVS-coding scheme behaves outside its intended working range.

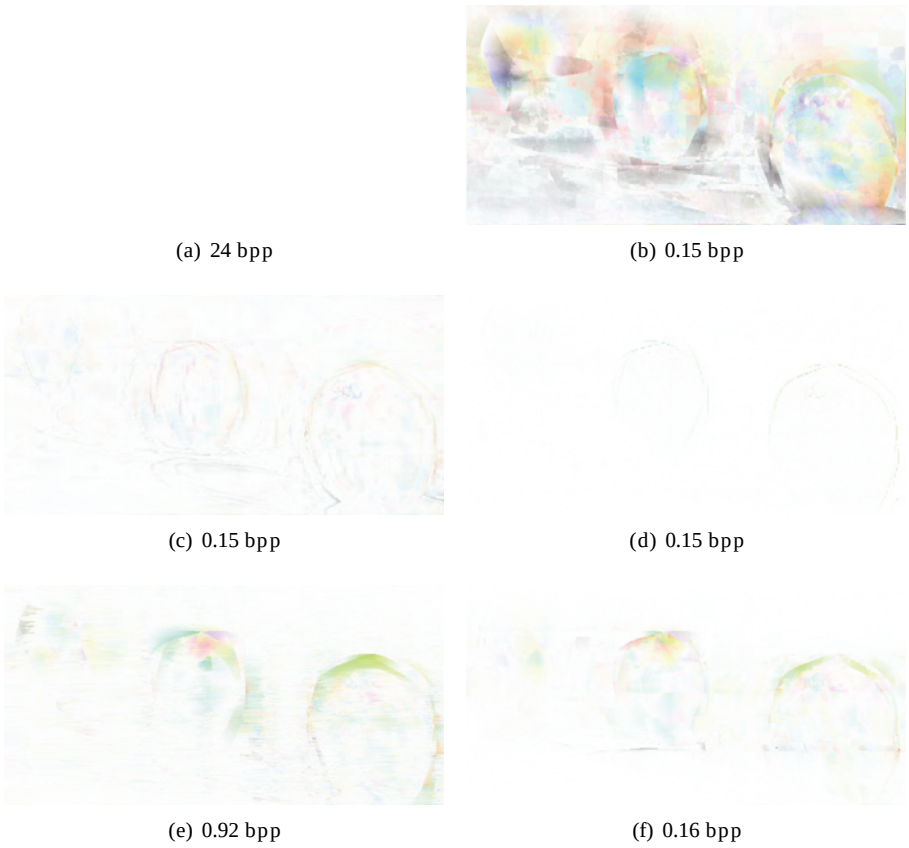


(a) 24 bpp (b) 0.15 bpp

(c) 0.15 bpp (d) 0.15 bpp

(e) 0.92 bpp (f) 0.16 bpp

Figure 5.13: Coding artifacts in Apples adhering to Setup 1 at a requested $r = 0.15$ bpp for (b) JPEG2000 coded II-picture, (c) SI-based PVI, (d) SI-based PVS, (e) EI-based PVS and (f) RI-based PVS. The actual bitrate produced is presented beneath each subfigure. Subfgure (a) shows the uncoded front view for comparison.



(a) 24 bpp (b) 0.15 bpp

(c) 0.15 bpp (d) 0.15 bpp

(e) 0.92 bpp (f) 0.16 bpp

Figure 5.14: Absolute difference images corresponding to the images shown in Figure 5.13 for (a) Uncoded front view (b) JPEG2000 coded II-picture, (c) SI-based PVI, (d) SI-based PVS, (e) EI-based PVS and (f) RI-based PVS.

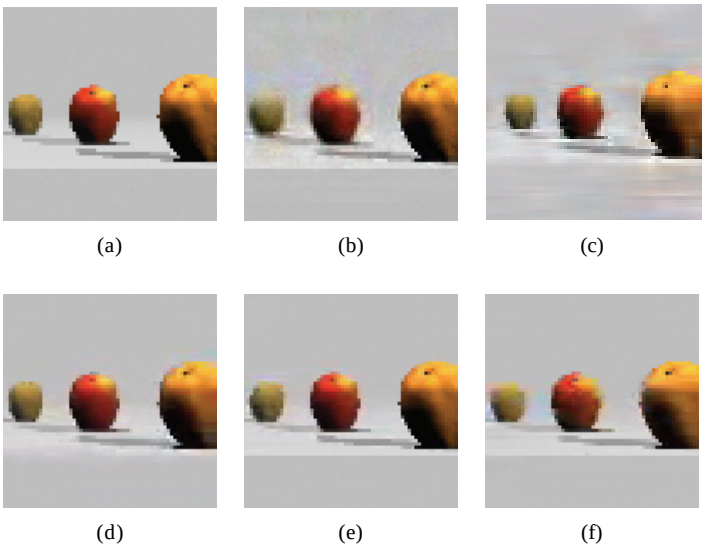
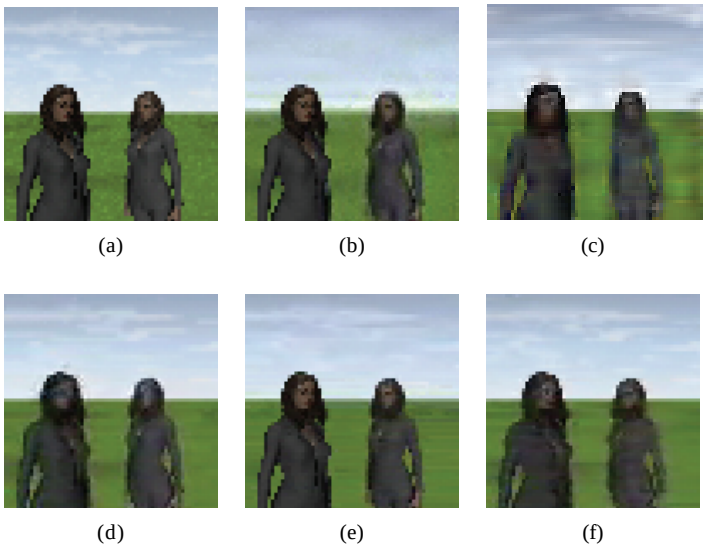


Figure 5.15: Characteristic coding artifacts in Apples adhering to Setup 3 at $r = 0.15$ bpp for (b) JPEG2000 coded II-picture, (c) SI-based PVI, (d) SI-based PVS, (e) EI-based PVS and (f) RI-based PVS. Subfigure (a) shows the uncoded front view for comparison.

5.8.5.1 Subjective evaluation

In Figures 5.15 – 5.18, the specific artifact characteristics of the coding schemes are illustrated for a bitrate $r = 0.15$ bpp selected to exaggerate the distortion. JPEG2000 coding of the II-picture results in the worst quality, which could expected after examining Figure 5.10. This is due to the lack of directional prediction in JPEG2000, which the motion compensation stage in the PVS-approaches allows for. The character of the coding artifacts in all VIs corresponding to JPEG2000, resembles additive Gaussian noise. The standard deviation of the noise increases slightly with increasing object depth. The difficulty in coding distant objects is because these objects, due to the lens array, spread out over a larger portion of the II-picture than close objects. This makes the introduced redundancy, which spans EIs that are further apart, more difficult to address for JPEG2000.

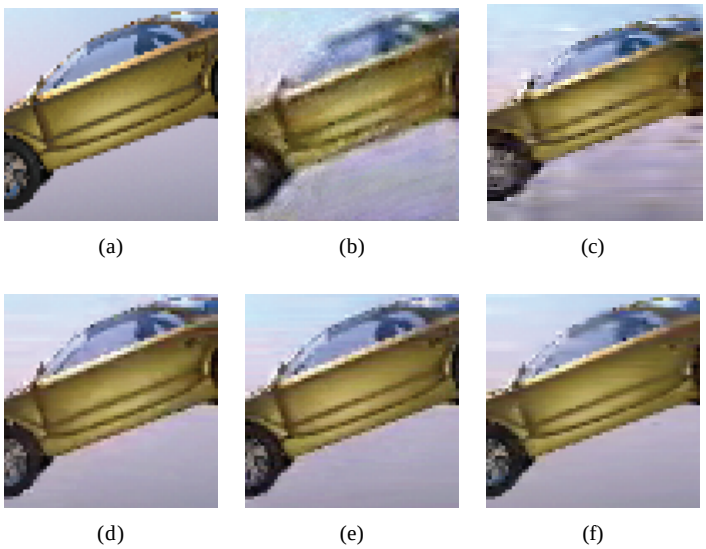
The SI-based PVI also results in quite significant coding artifacts, albeit with a different smearing characteristic than JPEG2000. The multidimensional wavelet decomposition of JP3D fails to rival the MCP of H.264 / AVC even when being to II-pictures adhering to Setup 3. Furthermore, the distortion induced in the low frequency decomposition levels are spread throughout the whole PVI if a single tile is used. As a result, the coding induces low frequency distortion in all views at all depths of the II-picture. Comparatively, the MCP of H.264 / AVC produce more high frequency distortion. This can be verified by the absence of distortion in the smooth areas of the VIs corresponding to PVS coding schemes.



(a) (b) (c)

(d) (e) (f)

Figure 5.16: Characteristic coding artifacts in Twins adhering to Setup 3 at $r = 0.15$ bpp for (b) JPEG2000 coded II-picture, (c) SI-based PVI, (d) SI-based PVS, (e) EI-based PVS and (f) RI-based PVS. Subfigure (a) shows the uncoded front view for comparison.



(a) (b) (c)

(d) (e) (f)

Figure 5.17: Characteristic coding artifacts in Car adhering to Setup 3 at $r = 0.15$ bpp for (b) JPEG2000 coded II-picture, (c) SI-based PVI, (d) SI-based PVS, (e) EI-based PVS and (f) RI-based PVS. Subfigure (a) shows the uncoded front view for comparison.

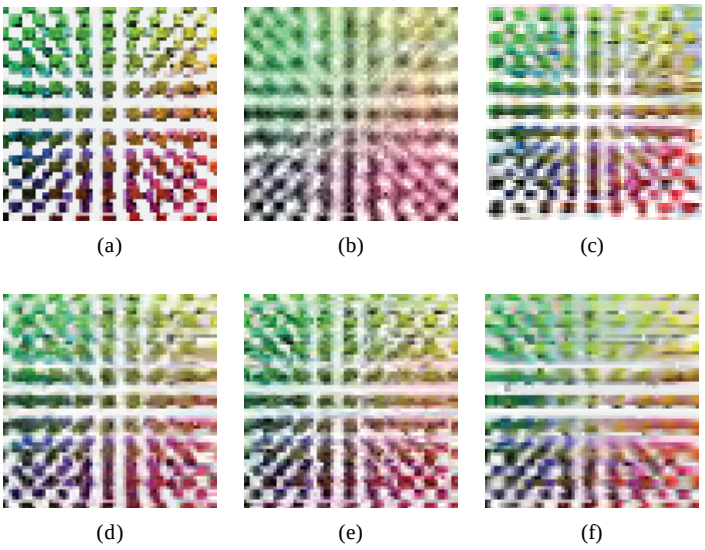


Figure 5.18: Characteristic coding artifacts in Cuboid adhering to Setup 3 at $r = 0.15$ bpp for (b) JPEG2000 coded II-picture, (c) SI-based PVI, (d) SI-based PVS, (e) EI-based PVS and (f) RI-based PVS. Subfigure (a) shows the uncoded front view for comparison.

When studying the artifacts caused by the SI-based PVS coding scheme in subfigures (d), a blurring over the whole depth range can be seen. In Twins this is especially evident in the face of the woman closest to the II-camera. As could be expected, the blurring effect is less pronounced for images with less detail. Apples and Car for example, contain objects with low frequency textures and thus show less coding artifacts. There is also a spread in colors around object edges, much like the mosquito noise that is well-known from many DCT-based coding schemes. The level of this noise tends to decrease with increased depth – compare the farthest and closest objects in Apples and Woman respectively.

In the EI-based PVS scheme, the noise has slightly different characteristics as subfigures (e) illustrates. The coding artifacts have a significant horizontal component, which is not the case for the mosquito-like artifacts introduced by the SI-based PVS. The level of the EI-based PVS noise also increases in level with increasing depth, as can be seen when comparing the edges of distant and close objects in Apples and Twins. In the complex Cuboid, the horizontal smearing combines with the penalizing of distant objects, which results in a coded image that looks "unwashed" compared to the SI-based version.

In the RI-based PVS scheme an irregular smearing effect appears. This is especially evident in the middle apple in Apples and in the face and suit of the closest woman in Twins. As a result of using RIs as PVS-pictures the introduced artifacts have different horizontal and vertical properties. Horizontally, a slight depth related mosquito-like noise can be noticed as in the case of SI-based PVS. Vertically, on the

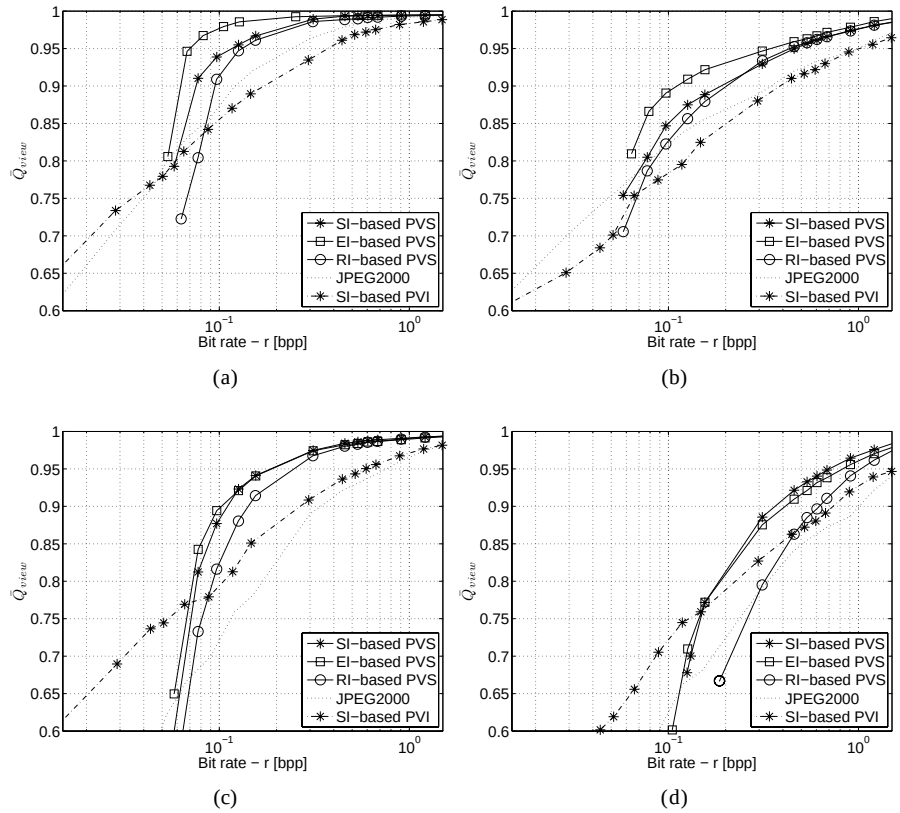


Figure 5.19: $\overline{\mathbf{Q}}_{view}$ evaluation of Setup 3 using Equation (4.3). (a) Apples, (b) Twins, (c) Car and (d) Cuboid.

other hand, there is no evident smoothing, which is especially evident in the sharp vertical edges of the sub-cube rows in Cuboid (see Figure 5.18 (f)).

5.8.5.2 Sparse angle dependent quality

Figure 5.19 presents $\overline{\mathbf{Q}}_{view}$ for the four II-pictures. The EI-based PVS scheme gives the best $\overline{\mathbf{Q}}_{view}$ for both Apples and Twins, which are two II-pictures with low fill factors and long depth ranges. This result is partially verified by examining Figure 5.16, where the VI from Twins show the least amount of distortion. However, the visual inspection of Apples in Figure 5.15 does not give the same unchallenged correspondence between $\overline{\mathbf{Q}}_{view}$ and subjective evaluation. The EI-based PVS introduces more visible distortion than the SI-based PVS if only the apples are considered. Figure 5.15 (e) shows that the furthest apple are severely distorted by the EI -based PVS whereas the shadowed floor closer to the II-camera is less affected by coding artifacts. Hence, $\overline{\mathbf{Q}}_{view}$ assesses the quality of the whole VI and does not consider whether the aver-

age viewer assesses the quality of a VI based on some region of interest.

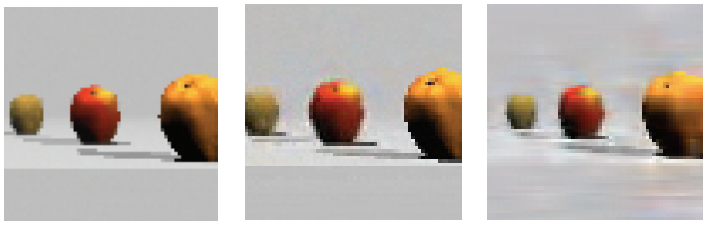
The SI-based PVS scheme performs equally well to that for the EI-based PVS for Car and Cuboid. The short depth range of Car and high fill factor of Cuboid render similar projection sizes and translation speeds in consecutive PVS-frames for both PVS-approaches. This makes the EI- and SI-based approaches produce similar PVSs and hence, similar coding efficiencies. When comparing the $\overline{\mathbf{Q}}_{view}$ graphs with the subjective quality of the VIs in subfigure (d) of Figure 5.15 – Figure 5.18 an agreement between the two can be seen. The smoothing effect of the SI-based PVS results in a relatively low $\overline{\mathbf{Q}}_{view}$ for II-pictures that have large regions of high frequency content such as Apples (the shadows on the floor) and Twins (the two women's features).

The RI-based PVS gives consistently the lowest $\overline{\mathbf{Q}}_{view}$ out of the three PVS s. Horizontally it shares the smoothing characteristic of the SI-based PVS. Vertically, it distorts farther objects more severely, similar to the EI-based PVSThe combination of these two distortion characteristics is not favorable for $\overline{\mathbf{Q}}_{view}$ or the subjective quality of the VIs.

Both JPEG2000 and the SI-based PVI result in the lowest $\overline{\mathbf{Q}}_{view}$ for the bitrate under test ($r = 0.15$ bpp). They take turns in being the worst of the evaluated coding schemes depending on which II-picture is being examined. However, Cuboid is an exception where the SI-based PVI is on par with the PVS approaches. The SI- and EI-based PVS gather together with the SI-based PVI at a $\overline{\mathbf{Q}}_{view} \approx 0.77$ when being coded at $r = 0.15$ bpp. The slight advantage in quality of the VIs from the PVS coding schemes compared to the SI-based PVI, is in part explained by their slightly higher bitrate caused by a non-optimal bitrate control of the H.264 / AVC-encoder.

5.8.5.3 Sparse pseudo-depth dependent quality

Figure 5.20 presents center views of the II-picture Apples coded using EI- and SI-based PVS, EI-based PVI and JPEG2000-coded II-picture. These center views will be used for visually validating that which can be inferred from the proposed quality metric $\mathbf{Q}_{depth}$. The II-pictures have been coded using bitrates resulting in $Q_{global} = 28$ dB for all coding schemes and thereby making them equal in distortion from a global perspective. $\mathbf{Q}_{depth}$ for all coding schemes are presented in Figure 5.21. Figure 5.21 show peaks in $\mathbf{Q}_{depth}$ at low depth layers d for the EI-based PVS and JPEG2000-coded II-picture but rapidly falls off as d is increased. This corresponds to the common property of the EI-based PVS and the JPEG2000-coded II-picture, i.e. to distort distant objects more than nearby objects. A property that also is seen in Figures 5.20 (b) and (e). The SI-based PVS results in a less steep decline in $\mathbf{Q}_{depth}$ for increasing depth, and the fluctuation over all evaluated depths is also lower. Figure 5.20 shows artifacts that are in line with this measurement: the distortion introduces an evenly distributed low-pass filtering throughout the depth, contrary to the other schemes that show a more heterogenous distribution of coding artifacts. The $\mathbf{Q}_{depth}$ graphs stemming from the SI-based PVI tend to have the least fluctuations, but at a low absolute level. The patchy-like artifacts in Figure 5.20 (c) are randomly distributed at different depths and combine into a quality level that is relatively low.



(a) 24 bpp (b) 0.35 bpp (c) 0.26 bpp

(d) 0.14 bpp (e) 0.08 bpp

Figure 5.20: Center views from the II-picture Apples coded to $Q_{global} = 28$ dB using (b) JPEG2000-coded II-picture (c) SI-based PVI, (d) SI-based PVS and (e) EI-based PVS. The bitrate r required for each coding scheme is presented beneath each subfigure. Subfigure (a) shows the uncoded front view for comparison.

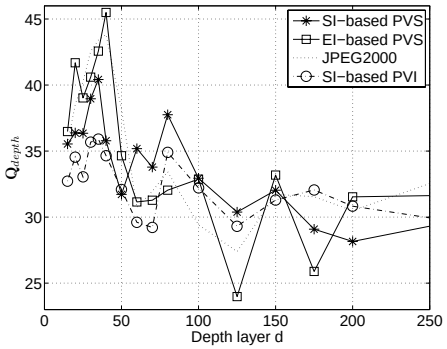


Figure 5.21: Sparse pseudo depth-dependent metric $\mathbf{Q}_{depth}$ applied to Apples coded using the four coding methods shown in Figure 5.20.

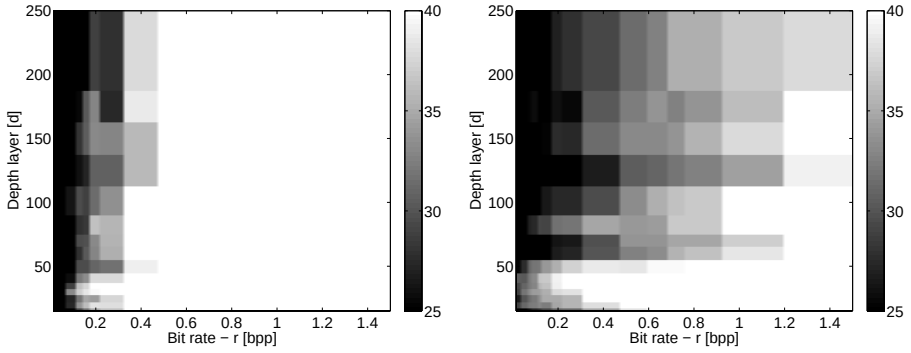


Figure 5.22: Rate-distortion images $\mathbf{Q}_{depth}(r)$ applied to Apples coded with (a) SI-based PVS and (b) JPEG2000-coded II-picture.

Analyzing rate-distortion properties is a vital aspect when coding schemes are evaluated. A good coding scheme should degrade the quality gracefully when the rate is reduced. $\mathbf{Q}_{depth}$ gives rise to a graph per bitrate evaluated, similar to those shown in Figure 5.21. If the rate is to be incorporated into the metric without reducing the information contained in it, the results may be presented as an image. Each pixel of the resulting image contains a specific $\mathbf{Q}_{depth}(r)$, i.e. a PSNR-value for a particular combination of rate and depth. Figure 5.22 shows such depth quality images for two coded versions of Apples. Presenting $\mathbf{Q}_{depth}(r)$ as an image gives a general view while still allowing for detailed analysis. An example is the result of the SI-based PVS in Figure 5.22 (a), which succeeds in producing high Q_d throughout the whole depth range for high bitrates. It also shows a smooth degradation when the rate is reduced. The JPEG2000-coded II-picture in Figure 5.22 (b) does not share this characteristic. Instead, objects further away are subject to coding artifacts even at high bitrates, and even more so when the rate is reduced.

Classical one-dimensional rate-distortion graphs can be obtained by reducing the dimensionality of $\mathbf{Q}_{depth}$ to moments such as the mean and standard deviation or extrema such as the minimum and maximum. Sometimes a measurement result in the form of a scalar value is more feasible for evaluation than a vector of values. Figure 5.23 shows for each bitrate the mean, minimum and maximum value, i.e. $\overline{\mathbf{Q}}_{depth}$, $\min \mathbf{Q}_{depth}$ and $\max \mathbf{Q}_{depth}$. Note the difference in spread of distortion levels that is given rise to by the coding schemes. The maximum quality level $\max \mathbf{Q}_{depth}$ is similar for the two coding schemes. However, the SI-based PVS shows both a higher average quality ($\overline{\mathbf{Q}}_{depth}$) and a higher minimum quality ($\min \mathbf{Q}_{depth}$) than the JPEG2000-coded II-picture. The evident advantage of reducing $\mathbf{Q}_{depth}$ to a single moment and two extrema is that graphs corresponding to traditional rate-distortion graphs may be produced. However, the disadvantage is that this reduction constitutes a considerable information loss with respect to what the metric produces.

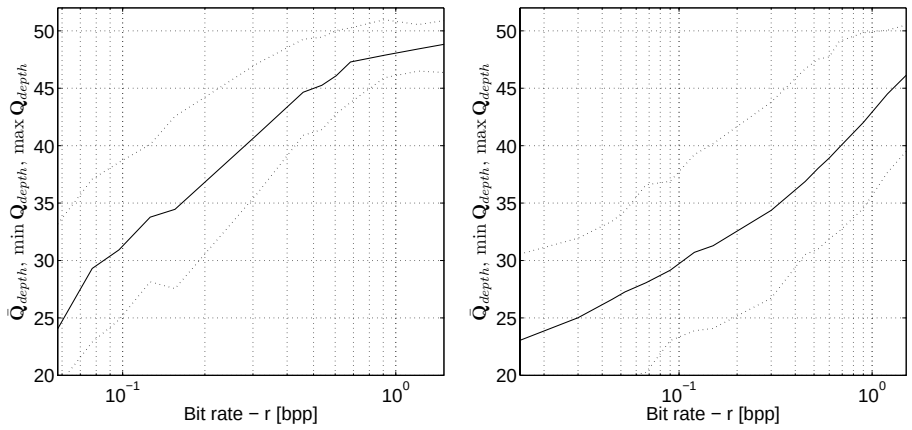


Figure 5.23: $\overline{\mathbf{Q}}_{depth}$ (solid line) , $\min \mathbf{Q}_{depth}$ (lower dashed line) and $\max \mathbf{Q}_{depth}$ (upper dashed line) applied to Apples coded with (a) SI-based PVS and (b) JPEG2000-coded II-picture.

5.8.6 Coding cost

Table 5.8 summarizes the CPU-time cost for the SI-based PVS coding scheme in comparison to a subset of the evaluated reference schemes. The CPU-time is averaged over all bitrates $r = \{0.015, \ldots, 1.5\}$ bpp. The proposed SI-based PVS achieves similar decoding times to those for the JPEG2000, despite the large difference in coding efficiency. Using EI as CI-type is again shown to be unfavorable for the II-picture structure of Setup 1. The low resolution of the PVS-frames does not compensate for the large number of frames that the H.264 / AVC encoder has to inspect. As a result, the CPU-time required for encoding an EI-based PVS is approximately two orders of magnitude larger than for encoding an SI-based PVS. Encoding an SI-based PVI is also approximately 10 times slower than for the corresponding SI-based PVS. The variations between II-pictures' CPU-times are small, which is anticipated as the actual 3D images content should only weakly affect the number of coding decisions the encoder has to make.

Depending on the regularity of the LUT used to construct the PVS different cache hit rates may be experienced as the LUT accesses II-picture pixels. The transfer of pixels from a Setup 1 II-picture to PVS took approximately 2 seconds for a completely random LUT on a the experimental setup's PC-system. However, a monotonically increasing LUT (where e.g. the pixels of the II-pictures are merely shifted into the PVS row-by-row) gives a transformation time of 200 ms. Hence, the combined transformation and CISO operation time lies in-between these bounds excluding the time required to construct or load the LUT.

Table 5.8: Measured CPU-time for encoding and decoding operations of the evaluated coding schemes; II-pictures adhering to Setup 1 was used.

	Apples		Twins		Car		Cuboid	
	Enc [s]	Dec [s]	Enc [s]	Dec [s]	Enc [s]	Dec [s]	Enc [s]	Dec [s]
SI-based PVS	9.1	2.0	10.7	2.8	10.9	2.7	12.0	2.9
SI-based PVI	53.4	16.5	67.2	15.3	74.2	15.9	93.4	14.5
RI-based PVS	15.9	2.2	17.8	2.6	18.9	2.6	17.0	2.5
EI-based PVS	1405.3	7.6	1400.3	7.8	1441.0	7.9	1421.9	9.0
JPEG2000	6.2	3.1	6.2	2.9	6.8	3.1	7.2	2.9

5.9 Concluding remarks

Coding an II-picture using 2D images coding tools fails to capture the inherent redundancy sufficiently well. I have shown that a much larger portion of this redundancy can be exposed if the II-picture is first transformed into a PVS. The coding scheme I have presented, which codes the PVS using the video coding standard H.264 / AVC, has shown to achieve a significant gain of up to 17.9 dB in coding efficiency compared to the 2D image coding standard JPEG2000. For a given quality level, the PVS scheme requires approximately 1 / 60-th of the bitrate necessary for JPEG2000. Transforming the II-picture into a PVI and coding it using the volumetric image coding standard JP3D was also investigated. Based on the lower coding efficiency of JP3D compared to H.264 / AVC I found this coding standard alternative to be a solution feasible mainly for applications that benefits from a partial decode of the II-picture, e.g. to produce enhanced 2D images from a subset of the II-picture.

Moreover, I introduced the concept of the header portion H of a PVS and showed it to be an important factor to consider when parameterizing the PVS-scheme. That is, how much of the allocated bitrate for a II-picture is required for storing the essential bitstream structure or semantics is essential for how much of the bitrate that is left for the 3D image quality. The magnitude of H is related to which CI-transform the PVS is constructed from. The SI-based PVS coding scheme that I have proposed, provides a significantly better coding efficiency for II-picture structures with a large number of low resolution EI than other PVS schemes. An increase of 5.8–13.4 dB in Q_{global} is measured when the SI-based PVS is compared with the EI-based PVS. This significant improvement in coding efficiency is partially explained by the low header portion H of the SI-based PVS.

The second operation when constructing a PVS after choosing a CI-transform is to define the CISO. After evaluating a set of CISOs, I concluded that the parallel CISO gives the highest coding efficiency for the investigated II-picture structure of Setup 1. Still, the difference in coding efficiency caused by different CISOs is small compared to that caused by different CI-transformations. The final parametrization step of the PVS coding schemes is to set the encoder parameters. The conducted study shows that the highest coding efficiency is achieved using a single I-picture followed by subsequent P-pictures as the combination of picture coding types and GOP-length. Incorporating B-pictures, shows a deterioration in the coding efficiency, despite the fact that they are being enabled for use as references.

Measuring the coding induced distortion with respect to II-picture properties such as viewing angle and depth is made possible by means of the quality metrics, which I described in Chapter 4. With these metrics I could show that the characteristics of the introduced coding artifacts differ depending on the selected PVS scheme. The SI transform introduces a smoothing effect homogenously over the II-pictures depth range, contrary to the other coding schemes that distribute the distortion more unevenly. Determining to what degree depth distribution of distortion contributes to the subjective quality of a II-based 3D images is the aim of a future study.

Coding an II-picture using a PVS scheme is approximately a factor of two more

expensive, in terms of coding time, compared to coding the II-picture using JPEG2000. For the 33 Mpixel II-picture in Setup 1 this corresponds to an approximate 10 second encoding time, compared to 6 seconds for JPEG2000. However, the improvement in objective quality is significant with a vast improvement of up to 17.9 dB in Q_{global}. Moreover, for a given quality an SI-based PVS scheme require less than 10% of the bitrate of JPEG2000. For some II-pictures it is sufficient with as little as approximately a 1 / 60-th of the bitrate of JPEG2000. The implementations of a coding scheme can to various degrees take advantage of and optimize for a certain system setup. Hence, the results that I have presented with regards to coding cost or CPU-time should be read with this in mind. A different experimental setup may prove to change the relationships between the evaluated coding schemes, albeit most likely only to a smaller degree.

5.9.1 Author contributions

In this chapter on coding my contributions lay mainly in:

- Developing an SI-based PVS scheme adopted for II-picture structures with a large set of relatively low resolution EIs, which when evaluated showed a vast improvement in coding efficiency compared to existing coding schemes.
- Evaluating theoretically and experimentally the effects of parameterizing a generic PVS scheme and the factors to consider when constructing an efficient coding scheme.
- Performing a qualitative study of the PVS-introduced coding artifacts and how they manifest in the viewing domain using objective quality evaluation metrics.

The work presented in this chapter has been published in parts in Papers III and IV.

5.9.2 Problem definitions – P2a and P2b

How can the II-based 3D images be coded such that a more compression efficient representation is achieved than what is possible with existing coding methods? This chapter has shown that it is possible to construct a coding scheme vastly more efficient than existing coding methods for II-pictures. The SI-based PVS significantly outperforms both the 2D images coding standard JPEG2000 as well as other previously proposed coding methods for II-pictures for II-picture structures with a multitude of low resolution EIs. Moreover, the coding scheme presented in this chapter is capable of uncovering a relatively larger portion if the II-pictures inherent redundancy by means of utilizing on state-of-the-art coding standards such as H.264 / AVC and JP3DÅs a result, higher coding efficiency and better visual quality are obtained.

What consequences will a proposed coding method have on objective quality? The coding scheme presented has shown large improvements in traditional objective quality

metrics such as globally applying PSNR to the complete data set. However, the studies have also shown that the II-pictures coding schemes introduce distortion characteristics that cannot be captured using global metrics. The previous chapter presented quality metrics that allows measuring more of II-picture-specific characteristics. Using these metrics on coded II-pictures have shown that the evaluated coding schemes differ in how they distribute artifacts within the depth range of the II-picture. The SI-based PVS spreads the coding artifacts homogenously through-out the depth, contrary to the majority of other coding schemes.

Chapter 6

Conclusions and future work

This chapter will give an overview of the work presented in this thesis, discuss what conclusions can be drawn from the presented results and give suggestions on what conceivable future works.

6.1 Overview

Chapter 1 presented the motivation for the research presented in this dissertation, the overall aim of the work and the concrete problems to be solved. The solution approach was also generally presented. Chapter 2 then presented a background on 3D images and techniques in general, Integral Imaging in particular, and related work that combined forms the basis for the rest of the thesis. First, Integral Imaging was presented with regards to the all-embracing description of the visible world: the plenoptic function. I then suggested a nomenclature to describe the content of an II-based 3D image or II-picture using the concept of Component Image(CI). Decomposing the II-picture into different CI types enables specific inherent characteristics to be revealed. The basic properties of an II-picture was also summarized based on a literature survey, which showed inter alia a strong relationship existing between the viewing angle, the image resolution, and the depth range. The only manner in which all three properties could be improved was shown to be by increasing the pixel resolution of the II-picture.

Beginning with Chapter 3, the research conducted to address the identified problems with regards to synthesis, coding, and evaluation is presented. Chapter 3 first describe an II-camera model in Chapter 3 which I constructed to be capable of characterizing numerous camera systems based on II. The II-camera model was combined with a open-source ray-tracing application (MegaPOV), for which I developed a set of macros that allows the II-camera model to be used for synthesizing II-pictures, and the Scene Description Language of MegaPOV to be used for defining virtual scenes of arbitrary complexity. The feasibility of the modular synthesiz-

ing method was demonstrated by producing a set of II-pictures adhering to different II techniques. The synthesis method provides a decoupling between II-camera, scene, and II-picture contrary to previous works. Chapter 4 focused on the lack of objective quality metrics specifically addressing how distortion manifests itself in properties specific to II-picture such as 3D image depth. Within the scope of the research conducted on II-picture evaluation I constructed and described two quality metrics, which explicitly measures distortion with regards to the depth and the viewing-angle dependent content of a coded II-picture. The metrics were used in the distortion analysis performed within the realms of II-picture coding schemes.

Chapter 5 initially presented why coding an II-picture with a coding standard for conventional 2D images (JPEG2000) is not a feasible approach, even though the II-picture is represented as a 2D image in its basic form. As a result I developed a coding scheme that explicitly aims at compressing II-pictures. The coding scheme is formed from two constituting parts, constructing a Pseudo Video Sequence or Pseudo Volumetric Image and coding these constructs using the state-of-the-art standards for video (H.264/AVC) or volumetric images (JP3D). A reduction in required bitrate by more than 90% (compared to JPEG2000) was shown to be achievable when applying the coding scheme to a set of II-pictures. The increase in encoding time for the coding scheme using H.264/AVC was measured to be approximately two-fold compared to JPEG2000, whereas the decoding time was almost exactly the same for the two coding approaches. When the H.264/AVC approach was compared with using JP3Das the encoding part of the coding scheme, the former provided the same quality as the latter at a forth of the bitrate and a tenth of the encoding time. Thus, the II-picture redundancies is more efficiently revealed and reduced if attacked using the 2D video coding tools of H.264/AVC than if JP3D is utilized. The parametrization of the coding scheme was analyzed and the conclusion was made that any PVS construction does not suffice, but instead that it is important to consider the II-picture structure for the coding scheme to be efficient. Moreover, the chapter also shows that forming the PVS-frame from different CIs has consequences for the character of the coding artifacts, which are inevitably induced by a lossy coding scheme.

6.2 Goal outcome

The work presented in this dissertation set out to achieve two goals, which was formulated in Chapter 1 and is restated here for convenient reading:

G1. Produce II-based 3D images that could depict strictly defined scenes while still being adaptable to new emerging II-techniques.

G2. Propose a coding scheme for II-based 3D images that provides a variable trade-off between coding efficiency and coding introduced distortion

Each of the two goals was split into specific problem definitions, which the previous chapters have addressed in their concluding remarks. Those discussions are con-

densed into two concluding remarks about the outcome with regards to goals G1 and G2.

6.2.1 Goal G1 - Easily produce II-based 3D images

A modular method to synthesizing II-based 3D images has been presented as an alternative to physical prototyping of II-camera systems. An II-camera model forms the basis and is easily constructed, modified, and stored using any software application that can load and save images, as it is represented in the form of common full-color 2D images with the use of the PNG image format. The method allows for virtual scenes of arbitrary complexity to be defined using a well defined SDL. Thus, the synthesis method provides a flexible and low cost production of II-pictures with exact knowledge about the properties of the scene, the II-camera, and the resulting 3D image. Having access to II-pictures with exactly known properties is a valuable tool for quick plausibility-checks of design ideas as well as thorough analysis of accuracy and precision in numerous algorithms operating on II-pictures. Hence, the presented synthesis solution achieves Goal G1 in a more flexible and generic way than previous works.

6.2.2 Goal G2 - A coding efficient coding scheme for II-pictures

This thesis has investigated the feasibility of using state-of-the-art coding standards for video and volumetric images to compress II-based 3 images. It has been shown that transforming the II-picture into a PVS and coding it using the video coding standard H.264/AVC provides a significantly higher coding efficiency compared to applying the state-of-the-art image coding standard JPEG2000 on the II-picture directly. Approximately 1/60-th of the bitrate required by JPEG2000 was sufficient to provide the same objective quality when the PVS-based coding scheme was used. The increase in cost, with regards to measured CPU-time during II-picture encoding, for the PVS scheme was modest when compared to the significant improvement in coding efficiency. A two-fold increase in coding time was measured in comparison with JPEG2000, whereas the decoding times were approximately equal. Hence, the coding scheme for II-pictures that I have presented shows a significantly higher coding efficiency than previous schemes at a modest increase in cost with regards to CPU-time.

6.3 Future work

The world of imaging is on the verge of a revolutionary change. The increasing resolution of image sensors and display panels has enabled a photographic technology presented a century year ago to become the next step in image and video applications. Integral Imaging enables novel camera systems to capture a larger portion of scene-reflected light than that which is possible with conventional cameras. Spatially

multiplexing both light intensity and light direction onto the camera's image sensor allows for the captured II-picture to produce 3D images, refocusable 2D images, and more.

Integral Imaging is merely in its infancy, despite its 100 year old history. Numerous imaging application areas will benefit from capturing a larger portion of light properties than what is possible with conventional cameras. I believe that the II-picture may prove to become one of the pillars in a new era in image processing, computational photography, and 3D applications. The future work discussed next will extend on the work this dissertation has presented, despite the many other interesting topics related to II. Thus, the ideas on future projects will relate to synthesis, evaluation, and coding of II-pictures.

6.3.1 Synthesis

The light transport in the synthesis method is currently simulated using ray-tracing and geometrical optics. This approach covers a large part of all relevant properties of real-life light transport, but not all. For example, when the physical dimensions of scene objects or II-camera components are reduced, diffraction might occur. Diffraction and other properties cannot be simulated using geometrical optics directly. Wave optic methods must instead be applied. To some degree this can be solved by additional post-processing and extensions to the II-camera model, e.g. by adding pixel maps that defines the point spread functionality of the model. Discerning the degree of distortion that such a model would induce compared to exact wave optics, and measure the effects such an extended II-camera model would have on the final II-picture quality remains to be answered.

Simulating image capture using ray-tracing is inherently slow. Hence, there is a potential for speeding up the rendering-time. Adapting the presented synthesis method to include real-time ray-tracing methods is potentially a solution. The fast development in programmable graphic cards are beginning to form the basis for many computational problems that may be performed in parallel using relatively cheap and powerful computational resources. Algorithmically solving the rendering problem is an alternative to tackle the problem with more computational power. A rendering pipeline especially designed for multiview rendering was briefly described in the previous works of Chapter 1. Investigating to what extent the simulation method presented can be coupled with this rendering approach would be interesting. Developing an efficient multiview rendering architecture is likely a very rewarding task, with the aid of increasingly powerful graphic card that allow themselves to be redesigned by software alone; rewarding not only from a scientific but also from a commercial perspective. An increasing number of 3D displays require an equally increasing number of 3D-enabled computers, game consoles, media players etc., all of which would benefit from an efficient rendering engine to present different types of 3D images and 3D video

6.3.2 Evaluation

The sparse angle dependent quality metric uses a simple approximation of an II-display when performing its calculations. So do the sparse depth dependent quality metric. For the purpose of these metrics, this elementary model is sufficient. However, for other more comprehensive studies a more detailed II-display model would facilitate a more versatile display simulation. Such a model could be based on the concepts of the II-camera model, given the similarities in operation between the two.

The greater part of the evaluations performed have focused on objective quality. However, subjective tests are important in order to relate the numerical results to perceived quality. Hence, testing the subjective quality of various coding schemes for II-pictures is an interesting work that will be conducted in an imminent project using two different types of autostereoscopic multiview displays. These tests are likely to focus on how the II-picture specific properties are affected by distortion. Thus, also enabling a study of how perceived quality correlates with the quality-estimates provided by the proposed metrics.

In addition to performing evaluation using empirical studies with objective or subjective methods, a more theoretical approach is also possible. The work initiated by Ramanathan and Girod [109] in formalizing a rate-distortion framework for light field images, is of special interest. With access to real-life II-pictures it becomes increasingly compelling to study the conformity between a statistical model and the real-life signal and with what means the two can converge.

6.3.3 Coding

A coding scheme is strengthened in its validity as an increasing number of II-pictures confirm its performance and properties. Evaluations of how real-life II-pictures are handled by the studied coding schemes is an interesting project, or more an ongoing process. Formalizing a commonly accepted test procedure with a set of commonly accepted reference II-pictures would benefit the field of II-picture coding. The work presented in this dissertation constitutes one step in such an endeavor.

Extending the coding scheme to also include time, i.e. investigate how to efficiently code II-based 3D video, is the next evolutionary step. The work that is actively being pursued for multi-view video in various standardization efforts within the JVT of MPEG and ITU (ISO / IEC MPEG-C Part 3[110], Multiview Video Coding – MVC, and Multiview Video plus Depth – MVD) should form the base for such future works combined with the research presented with regards to II-based 3D images.

The coding scheme I have presented uses standardized coding schemes as a basis for coding II-pictures. Other II-picture coding approaches that does not re-use coding tools of state-of-the-art standards to the same extent, may prove to be even more adaptable to the characteristics of the II-picture. If so, only the sky (or the theoretical upper bound) is the limit with regards to how high the quality can become for any given bitrate.

Bibliography

[1] G. Lippmann, "Epreuves reversibles," *Comptes rendus hebdomadaires des Séances de l´Acadéemie des Sciences,* vol. 146, pp. 446 – 451, 1908.

[2] B. Javidi and F. Okano, Eds., *Three-Dimensional Television, Video, and Display Technologies*. Springer, 2002.

[3] M. C. Forman, N. Davies, and M. McCormick, "Continuous parallax in discrete pixelated integral 3D displays," *Optical Society of America*, 2002.

[4] "Ibn al-haytham," Wikipedia, December 2006. [Online]. Available: en.wikipedia.org / wiki / Alhazen

[5] "Alhazen," The Swedish National Encyclopedia, 2006. [Online]. Available: www.ne.se / jsp / search / article.jsp?i_art_id=111380

[6] "Perspective," Encyclopedia Britannica, 2006. [Online]. Available: search.eb.com / eb / article-9059357

[7] "LENSSTAR.org – a site committed to bringing you the latest on lenticular," 2006. [Online]. Available: www.lenstar.org

[8] D. E. Roberts, "History of lenticular and related autostereoscopic methods," Leap Technologies, LCC, Tech. Rep., 2003. [Online]. Available: www.microlens.com / HistoryofLenticular.pdf

[9] C. Wheatstone, "On some remarkable, and hitherto unobserved, phenomena of binocular vision," *Philosophical Transactions of the Royal Society of London*, vol. 128, pp. 371 – 394, 1838.

[10] J.-Y. Son, B. Javidi, and K.-D. Kwack, "Methods for displaying three-dimensional images," *Proceedings of the IEEE,* vol. 94, no. 3, pp. 502 – 523, March 2006.

[11] M. Siegel and S. Nagata, "Just enough reality: Comfortable 3-D viewing via microstereopsis," *IEEE Transactions on Cicuit and Systems for Video Technology*, vol. 10, no. 3, pp. 387 – 396, April 2000.

[12] M. W. Halle, "Multiple viewpoint rendering for three-dimensional displays," Ph.D. dissertation, MIT, 1997.

[13] J. D. Pfautz, "Depth perception in computer graphics," Ph.D. dissertation, University of Cambridge, 2002.

[14] O. Schreer, P. Kauff, and T. Sikora, Eds., *3D Videocommunications: Algorithms, concepts and real-time systems in human centered communication*. Wiley, 2005.

[15] T. Motoki, H. Isono, and I. Yuyama, "Present status of three-dimensional television research," *Proceedings of the IEEE*, vol. 83, no. 7, pp. 1009 – 1021, 1995.

[16] T. Okoshi, "Three-dimensional displays," *Proceedings of the IEEE*, vol. 68, no. 5, pp. 548–564, May 1980.

[17] N. A. Dodgson, "Autostereoscopic 3D displays," *IEEE Computer Society*, pp. 31 – 36, August 2005.

[18] "Proview anaglyph glasses," 3Dstereo-com, Inc., August 2006. [Online]. Available: www.3dstereo.com

[19] "Polarization glasses," Screen-Tech - the screen company, August 2006. [Online]. Available: www.screen-tech.de

[20] "Eye 2000 shutter glasses," Another World Inc., August 2006. [Online]. Available: www.anotherworld.to

[21] "Visette 45 SXVGA head mounted display," www.cybermindnl.com, Tech. Rep., August 2006. [Online]. Available: www.cybermindnl.com

[22] D. Pizzanelli, "The development of direct - write digital holography," Holographer.org, Tech. Rep., 2004.

[23] "Depthcube," LightSpace Technologies Inc., August 2006. [Online]. Available: lightspacetech.com

[24] "Perspecta display," Actuality Systems Inc., August 2006. [Online]. Available: www.actuality-systems.com

[25] J.-Y. Son and B. Javidi, "Three-dimensional imaging methods based on multiview images," *IEEE/OSA Journal of Display Technology*, vol. 1, no. 1, pp. 125 – 140, September 2005.

[26] "WoWvx Technology," Philips 3D Solutions, August 2006. [Online]. Available: www.business-sites.philips.com/3dsolutions/about/Index.html

[27] H. Liao, D. Tamura, M. Iwahara, N. Hata, and T. Dohi, "High quality autostereoscopic surgical display using anti-aliased integral videography imaging," in *Proc. MICCAI 2004, LNCS*, C. Barillot and D. Haynor, Eds. Springer-Verlag Berlin Heidelberg, 2004, pp. 462–469.

[28] "Sharp3d," Sharp Systems of America, August 2006. [Online]. Available: www.sharp3d.com

[29] "3D Display," Samsung SDI.CO., LTD, August 2006. [Online]. Available: www.samsungsdi.com/contents/en/product/3d/3d.html

[30] M. Levoy, "Light fields and computational imaging," *IEEE Computer*, vol. 39, no. 8, pp. 46–55, August 2006.

[31] H. E. Ives, "Optical properties of a Lippmann lenticulated sheet," *Journal of the Optical Society of America*, vol. 21, no. 3, pp. 171 – 176, March 1931.

[32] A. Stern and B. Javidi, "Three-dimensional image sensing, visualization, and processing using integral imaging," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 591 – 607, March 2006.

[33] C. B. Burckhardt, "Optimium parameters and resolution limitation of integral photography," *Journal of the Optical Society of America*, vol. 58, no. 1, pp. 71 – 76, January 1968.

[34] T. Okoshi, "Optimum design and depth resolution of lens-sheet and projection-type three-dimensional displays," vol. 10, no. 10, pp. 2284–2291, October 1971.

[35] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Stanford University Computer Science Tech Report 2005-02, Tech. Rep., 2005.

[36] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light field microscopy," in *ACM Transactions on Graphics 25(3)Proceedings of SIGGRAPH 2006*, vol. 25, no. 3, 2006.

[37] F. Okano, J. Arai, K. Mitani, and M. Okui, "Real-time integral imaging based on extremely high resolution video system," in *Proceedings of the IEEE*, vol. 94, no. 3. IEEE, March 2006.

[38] T. Georgeiv, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, , and C. Intwala, "Spatio-angular resolution tradeoff in integral photography," in *17th Eurographics Symposium on Rendering*, T. Akenine-Möller and W. Heidrich, Eds., Cyprus, June 2006.

[39] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. Cambridge, MA: MIT Press, 1991, pp. 3–20.

[40] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of SIGGRAPH '96*, no. 23. New Orleans (LA), USA: ACM, August 1996.

[41] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," 1996.

[42] M. McCormick and N. Davies, "Full natural colour 3D optical models by integral imaging," in *Fourth International Conference on Holographic Systems, Components and Applications*, September 1993, pp. 237 – 242.

[43] J.-H. Park, S.-W. Min, S. Jung, and B. Lee, "Analysis of viewing parameters for two display methods based on integral photography," *Applied Optics*, vol. 40, no. 29, pp. 5217 – 5232, October 2001.

[44] C. B. Burckhardt, R. J. Collier, and E. T. Doherty, "Formation and inversion of pseudoscopic images," *Journal of Applied Optics*, vol. 7, no. 3, pp. 627–632, April 1968.

[45] M. C. Forman, "Compression of integral three-dimensional television pictures," Ph.D. dissertation, De Montfort University, September 1999.

[46] F. Okano, J. Arai, H. Hoshino, and I. Yuyama, "Real-time three-dimensional pickup and display system based on integral photography," in *Novel Optical Systems and Large-Aperture ImaginNovel Optical Systems and Large-Aperture Imaging*, vol. 3430, no. 1. SPIE, 1998, pp. 70–79.

[47] M. Martínez-Corral, B. Javidi, R. Martínez-Cuenca, and G. Saavedra, "Formation of real, orthoscopic integral images by smart pixel mapping," *Optics Express*, vol. 13, no. 23, pp. 9175–9180p, November 2005.

[48] J.-S. Jang and B. Javidi, "Three-dimensional projection integral imaging using micro-convex-mirror arrays," *Optics Express*, vol. 12, no. 6, pp. 1077–1083, March 2004.

[49] M. Halle, "Autostereoscopic displays and computer graphics," *ACM SIGGRAPH Computer Graphics*, vol. 31, no. 2, pp. 58–62, May 1997.

[50] F. Okano, M. Kobayashi, J. Arai, and M. Okui, "Depth control GRIN lens array for integral photography," in *Three-Dimensional TV, Video, and Display II*, vol. 5243. SPIE, November 2003, pp. 30–41.

[51] B. Lee, S.-W. Min, and B. Javidi, "Theoretical analysis for three-dimensional integral imaging systems with double devices," *Applied Optics*, vol. 41, no. 23, pp. 4856 – 4865, August 2002.

[52] J.-S. Jang and B. Javidi, "Improved viewing resolution of three-dimensional integral imaging by use of nonstationary micro-optics," *Optics Letters*, vol. 27, no. 5, pp. 324 – 326, March 2002.

[53] H. Choi, S.-W. Min, S. Jung, J.-H. Park, and B. Lee, "Multiple-viewing-zone integral imaging using a dynamic barrier array for three-dimensional displays," *Optics Express*, vol. 11, no. 8, pp. 927– 932, 2003.

[54] Y. Wang, J. Osterman, and Y.-Q. Zhang, *Video Processing and Communications*, ser. Signal Processing Series. Prentice Hall, 2002.

[55] B. Lee, S. Jung, and J.-H. Park, "Viewing-angle-enhanced integral imaging by lens switching," *Optics Letters*, vol. 27, no. 10, pp. 818 – 820, May 2002.

[56] A. R. L. Travis, "The display of three-dimensional video images," *Proceedings of the IEEE*, vol. 85, no. 11, pp. 1817–1832, November 1997.

[57] H. Choi, Y. Kim, J.-H. Park, S. Jung, and B. Lee, "Improved analysis on the viewing angle of integral imaging," *Applied Optics*, vol. 44, no. 12, pp. 2311 – 2317, April 2005.

[58] S.-W. Min, J. Kim, and B. Lee, "New characteristic equation of three-dimensional integral imaging system and its applications," *Japanese Journal of Applied Physics*, vol. 44, no. 2, pp. 71 – 74, 2005.

[59] D.-H. Shin, M. Cho, K.-C. Park, and E.-S. Kim, "Computational technique of volumetric object reconstruction in integral imaging by use of real and virtual image fields," *ETRI Journal*, vol. 27, no. 6, pp. Dong–Hak Shin, Myungjin Cho, Kyu–Chil Park, and Eun–Soo Kim, December 2005.

[60] J.-S. Jang and B. Javidi, "Large depth-of-focus time-multiplexed three-dimensional integral imaging by use of lenslets with nonuniform focal lenghts and aperture sizes," *Optics Letters*, vol. 28, no. 20, pp. 1924 – 1926, October 2003.

[61] Y. Kim, J.-H. Park, H. Choi, J. Kim, S.-W. Cho, and B. Lee, "Depth-enhanced three-dimensional integral imaging by use of multilayered display devices," *Applied Optics*, vol. 45, no. 18, pp. 4334 – 4343, June 2006.

[62] H. Hoshino, F. Okano, H. Isono, and I. Yuyama, "Analysis of resolution limitation of integral photography," *Optical Society of America*, vol. 15, no. 8, pp. 2059 – 2065, August 1998.

[63] C. Wu, A. Aggoun, M. McCormick, and S. Kung, "Depth extraction from unidirectional integral image using a modified multi-baseline technique," *Stereoscopic Displays and Virtual Reality Systems IX*, vol. 4660, pp. 135 – 145, May 2002.

[64] J.-H. Park, S. Jung, H. Choi, Y. Kim, and B. Lee, "Depth extraction by use of a rectangular lens array and one-dimensional elemental image modification," *Applied Optics*, vol. 43, no. 25, pp. 4882– 4895, September 2004.

[65] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3D visual communication," in *Proceedings of International Picture Coding Symposium*, 1996.

[66] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, vol. 1, pp. 7–55, 1987.

[67] M. C. Forman, A. Aggoun, and M. McCormick, "A novel coding scheme for full parallax 3D-TV pictures," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, April 1997, pp. 2945–2947.

[68] R. Martínez-Cuenca, G. Saavedra, M. Martínez-Corral, and B. Javidi, "Enhanced depth of field integral imaging with sensor resolution constraints," *Optics Express*, vol. 12, no. 21, October 2004.

[69] H. Liao, S. Nakajima, M. Iwahara, N. Hata, I. Sakuma, and T. Dohi, "Real-time 3D image-guided navigation system based on integral videography," in *Proceedings of SPIE*, vol. 4615. SPIE, 2002, pp. 36–44.

[70] H. Liao, M. Iwahara, N. Hata, and T. Dohi, "High-quality integral videography using a multiprojector," in *Optics Express*, vol. 12, no. 6, March 2004, pp. 1067–1076.

[71] G. Milnthorpe, M. McCormick, and N. Davies, "Computer modeling of lens arrays for integral image rendering," in *Proceedings of the 20th Eurographics UK Conference*. IEEE Computer Society, 2002.

[72] K. Brown, M. McCormick, N. Davies, M. C. Forman, G. Milnthorpe, and R. Kotecha, "The use of computer generated integral images to visualise cyber-sculpture," in *Proceedings of the 20th Eurographics UK Conference*, 2002.

[73] S. S. Athineos, N. P. Sgouros, P. G. Papageorgas, D. E. Maroulis, M. S. Sangriotis, and N. G. Theofanous, "Physical modeling of a microlens array setup for use in computer generated IP," in *Proc. of SPIE-IS&T Electronic Imaging*, A. J. Woods, Ed., vol. 5664, March 2005, pp. 472 – 479.

[74] N. P. Sgouros, A. G. Andreou, M. S. Sangriotis, P. G. Papageorgas, D. M. Maroulis, and N. G. Theofanous, "Compression of IP images for autostereoscopic imaging applications," in *Proceedings of ISPA03*, vol. 1, September 2003, pp. 223 – 227.

[75] S. Yeom, A. Stern, and B. Javidi, "Compression of 3D color integral images," *Optics Express*, vol. 12, no. 8, pp. 1632 – 1642, April 2004.

[76] F. Shao, G. Jiang, K. Chen, M. Yu, and T.-Y. Choi, "Ray-space data compression based on prediction technique," in *International Conference on Computer Graphics, Imaging and Vision: New Trends*, July 2005, pp. 347 – 350.

[77] R. Olsson, M. Sjöström, and Y. Xu, "A combined pre-processing and H.264-compression scheme for 3D integral images," in *Proceedings of ICIP 2006*. Atlanta (GA), USA: IEEE, October 2006, pp. 513 – 516.

[78] M. C. Forman, N. Davies, and M. McCormick, "Objective quality measurement of integral 3D images," in *Proceedings of SPIE Vol. 4660 Stereoscopic Displays and Virtual Reality Systems IX*, 2002.

[79] M. Magnor and B. Girod, "Data compression for light field rendering," *Transactions for Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 338–343, April 2000.

[80] J. V. der Linden and R. Lobb, "MPEG-encoded light fields," *Journal of Graphics Tools*, vol. 6, no. 2, pp. 1–15, 2001.

[81] B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, Hong Kong, China, April 2003, pp. 760 – 763.

[82] M. Tanimoto, "Free viewpoint television - FTV," in *Picture Coding Symposium 2004*, 2004.

[83] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[84] "MegaPOV – a custom and unofficial patched version of POV-Ray," August 2005. [Online]. Available: http://megapov.inetart.net/

[85] "The Persistence of Vision Raytracer (POV-Ray)," August 2006. [Online]. Available: www.povray.org

[86] E. W. Weisstein, "Nearest integer function," From MathWorld–A Wolfram Web Resource, May 2008. [Online]. Available: http://mathworld.wolfram.com/NearestIntegerFunction.html

[87] "Portable network graphics (PNG) specification (second edition): Functional specification.ISO/IEC 15948:2003 (E)," World Wide Web Consortium (W3C), Tech. Rep., November 2003. [Online]. Available: http://www.w3.org/TR/2003/REC-PNG-20031110

[88] T. Frannansa, "Dawn over the mountains," POVRay Short Code Contest - Round 4, 2006. [Online]. Available: local.wasp.uwa.edu.au/pbourke/exhibition/scc4/final/

[89] A. Chalmers, S. Daly, A. McNamara, K. Myszkowski, and T. Troscianko, "Image quality metrics," SIGGRAPH Course Nr.44, July 2000.

[90] R. Olsson, M. Sjöström, and Y. Xu, "Evaluation of combined pre-processing and H.264-compression schemes for 3D integral images," in *Proceedings of Electronic Imaging - VCIP*. San Jose (CA), USA: IS&T/SPIE, January 2007.

[91] A. B. Watson, J. Hu, J. F. M. III, and J. B. Mulligan, "Design and performance of a digital video quality metric," in *Human Vision, Visual Processing, and Digital Display IX*, vol. 3644. SPIE, 1999, pp. 168 – 174.

[92] S. Wolf and M. H. Pinson, "Spatial-temporal distortion metric for in-service quality monitoring of any digital video system," in *Multimedia Systems and Applications II*, vol. 3845. SPIE, November 1999, pp. 266 – 277.

[93] M. A. Masry and S. S. Hemami, "A metric for continuous quality evaluation of compressed video with severe distortions," *Signal Processing: Image Communication*, vol. 19, no. 19, pp. 133 – 146, 2004.

[94] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600 – 612, April 2004.

[95] D. Vatolin, A. Parshin, O. Petrov, and A. Titarenko, "MSU subjective comparison of modern video codecs," Video Group, Graphics & Media Lab, Department of Computer Science Moscow State University, Tech. Rep., January 2006.

[96] K. Takahashi and T. Naemura, "Unstructured light field rendering using on-the-fly focus measurements," in *IEEE International Conference on Multimedia and Expo*. IEEE, July 2005.

[97] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7 – 42, 2002.

[98] J. Ren, A. Aggoun, and M. McCormick, "A novel object depth estimation algorithm for integral 3D images," in *International Conference onVisual Information Engineering*, 2003.

[99] D. Scharstein and R. Szeliski, "Middlebury stereo vision page," 2008. [Online]. Available: vision.middlebury.edu / stereo /

[100] *ISO/IEC 14496-10:2005, Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*. ISO, Geneva, Switzerland, 2005.

[101] *ISO/IEC 15444-10:2007, Information Technology – JPEG 2000 Image Coding System: Part 10 – Extensions for three-dimensional data*. ISO / IEC JTC1 / SC29 / WG1 FDIS, 2007.

[102] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264 / AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," in *Conference on Applications of Digital Image Processing XXVII*, vol. 5558. SPIE, 2004, pp. 454 – 474.

[103] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264 / MPEG4 advanced video coding standard and its applications," *Communications Magazine*, vol. 44, no. 8, pp. 134–143, August 2006.

[104] G. Sullivan and T. Wiegand, "Video compression - from concepts to the H.264 / AVC standard," *Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 18–31, January 2005.

[105] T. Bruylants, A. Munteanu, A. Alecu, R. Deklerck, and P. Schelkens, "Volumetric image compression with JPEG2000," in *Biomedical Optics & Medical Imaging*. SPIE, 2007.

[106] L. Aimar, L. Merritt, E. Petit, M. Chen, J. Clay, M. Rullgård, R. Czyz, C. Heine, A. Izvorski, and A. Wright, "x264 - a free h264 / avc encoder. core: 38 svn-341," December 2005. [Online]. Available: http: / / developers.videolan.org / x264.html

[107] D. Taubman, "Kakadu software - a comprehensive framework for jpeg2000 developers, v5.2," 2006. [Online]. Available: www.kakadusoftware.com

[108] T. Bruylants and P. Schelkens, "JPEG2000 Part 10 - Verification Model JP3D - v1.0.5," Vrije Universiteit Brussel (VUB) - Interdisciplinary institute for BroadBand Technology (IBBT), ISO / IEC JTC1 / SC29 / WG1, N4194, Tech. Rep., 2007.

[109] P. Ramanathan and B. Girod, "Theoretical analysis of geometry inaccuracy for light field compression," in *International Conference on Image Processing*, Rochester (NY), USA, September 2002.

[110] "ISO / IEC FDIS 23002-3, MPEG-C Part 3: Representation of auxiliary video and supplemental information," August 2007.

Biography

Roger Olsson was born on the 29th of September 1973 in Härnösand, Sweden. He received Master of Science in Electrical Engineering from Mitthögskolan, Sweden in 1998. In late 1997 he was employed at Limt Technology AB where he conducted research within the field of MPEG-2 based broadcast television until 2000. Between 2000 and 2004 he taught signal processing courses as a junior lecturer at the Mid Sweden University (MIUN). Parallel to teaching he began pursuing a Ph.D. at MIUN on part-time and from 2004 this has been his full-time occupation.

Roger is a student of the Graduate School of Telecommunications (GST) and between June 2004 and June 2005 he was the PhD-student representative in the GST Program Council. During the autumn term of 2004 he was also an interim PhD-student representative in the board of the Faculty of Science,Technology and Media at MIUN. Roger is a also a student member of IEEE and SPIE.

His main research interest is 3D images based on integral imaging and how to enhance and extend conventional capture and display applications with the additional features that integral imaging brings. Further more, extending the working range of state-of-the-art coding standards such as H.264/AVC and JPEG 2000 3D to include new signal types have attracted Roger's interest during the last years. Currently he is participating in two projects, which focus on introducing realistic 3D visualization into medical and marketing applications. On his spare time he elopes to the weekend cottage in the High Coast where he among other things contemplates on life over a glass of single malt with his wife, family and friends.